REASTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples

Yilun Zhao[1] **Linyong Nan**[1] **Zhenting Qi**[2] **Rui Zhang**[3] **Dragomir Radev**[1]
[1]Yale University [2] Zhejiang University [3]Penn State University
{yilun.zhao, linyong.nan}@yale.edu

Abstract

Reasoning over tabular data requires both table structure understanding and a broad set of table reasoning skills. Current models with table-specific architectures and pre-training methods perform well on understanding table structures, but they still struggle with tasks that require various table reasoning skills. In this work, we develop REASTAP to show that high-level table reasoning skills can be injected into models during pre-training without a complex table-specific architecture design. We define 7 table reasoning skills, such as numerical operation, temporal comparison, and conjunction. Each reasoning skill is associated with one example generator, which synthesizes questions over semi-structured tables according to the sampled templates. We model the table pre-training task as a sequence generation task and pre-train REASTAP to generate precise answers to the synthetic examples. REASTAP is evaluated on four benchmarks covering three downstream tasks including: 1) WIKISQL-WEAK and WIKITQ for Table Question Answering; 2) TABFACT for Table Fact Verification; and 3) LOGICNLG for Faithful Table-to-Text Generation. Experimental results demonstrate that REASTAP achieves new state-of-the-art performance on all benchmarks and delivers a significant improvement on low-resource setting. Our code is publicly available at https://github.com/Yale-LILY/ReasTAP.

1 Introduction

Inspired by the massive success of pre-trained language models (LM) on free-form natural language (NL) tasks (Devlin et al., 2019; Dong et al., 2019; Raffel et al., 2020; Lewis et al., 2020), researchers have attempted to extend the pre-training to table data. Tables are a valuable form of data that organize information in a structured way. They often contain data that is organized in a more accessible manner than in unstructured texts. To adapt the pre-training paradigm on structured tabular data, previous works mainly focus on designing models with table-specific architectures and pre-training methods. This includes introducing a structure-aware attention mechanism (Yin et al., 2020; Deng et al., 2020; Zayats et al., 2021), adding auxiliary structure indicative embeddings (Herzig

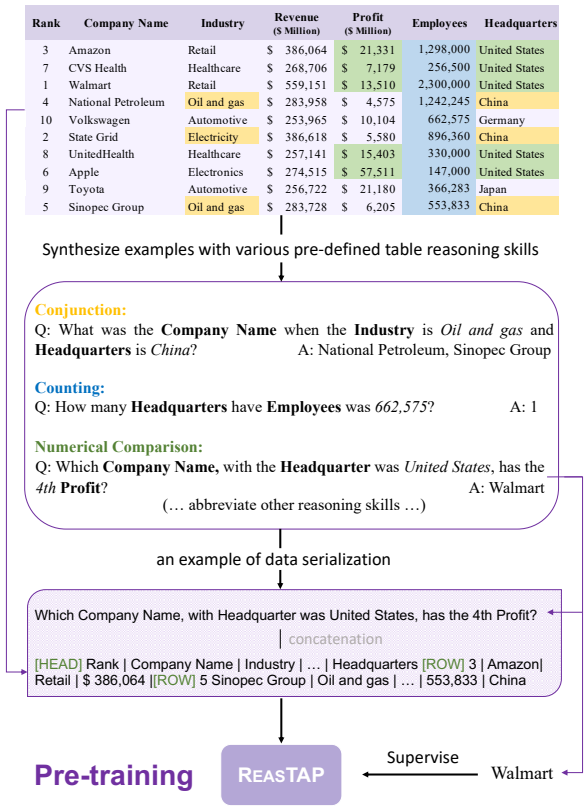

Figure 1: The illustration of REASTAP pre-training. The tables are crawled from Wikipedia. During pre-processing, we perturb the table row order to alleviate unwanted bias brought by table encoding. The colored cells are relevant facts necessary to answer the given question. Each color corresponds to a different table reasoning skill. And each reasoning skill corresponds to an example generator, which synthesizes QA pairs over tables according to the sampled templates. We model the pre-training task as a sequence generation task and pre-train REASTAP to generate correct answers given the flatten table and synthetic question.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9006–9018
December 7-11, 2022

et al., 2020; Eisenschlos et al., 2020; Wang et al., 2021b), and designing table-specific pre-training objectives (Yin et al., 2020; Yu et al., 2021a; Wang et al., 2021b; Liu et al., 2022b,a). While these methods are effective in understanding table structures, they increase the modeling complexity and lack interpretability on why models learns table reasoning skills during pre-training.

This paper presents a new table pre-training approach, named REASTAP, which enables a model to efficiently learn table structure understanding and table reasoning skills during pre-training. We first defined 7 table reasoning skills, such as numerical operation and temporal comparison. As shown in Figure 1, for each reasoning skill, a corresponding example generator was applied to synthesize Question Answering (QA) examples over tables. We modeled the pre-training task as a sequence generation task and pre-trained a sequence-to-sequence (seq2seq) LM to generate the answer to the synthetic questions. REASTAP is theoretically applicable to any seq2seq LM without a table-specific architecture design. Our key insight is that if a language model can be pre-trained to generate the answers to synthetic questions, which require various table reasoning skills, it should have a great table structure understanding and table reasoning capacity, thereby conferring benefits to downstream tasks. The main contributions of our work can be summarized as follows:

- We develop a new table reasoning example generation pipeline, which produces a large-scale table QA corpus that requires various reasoning skills over semi-structured tables.
- We propose a new table pre-training method, REASTAP, which helps the model to learn table structure understanding and various table reasoning skills during pre-training without any table-specific architecture design.
- REASTAP is evaluated on four downstream benchmarks. Experimental results demonstrate that REASTAP achieves new state-of-the-art results on all of them, and delivers a great improvement on low-resource setting.

2 Pre-training Corpus

2.1 Table Source and Pre-processing

We chose publicly available semi-structured tables as the table source. Specifically, we extracted tables from English Wikipedia[1], which covered a wide range of domains including popular culture, geography, politics, and science. We kept tables with 8-30 rows and at least three columns, resulting in around 600K tables. For each extracted table, a pre-processing script was applied to automatically annotate table columns with their data types (i.e, string, number, and date), which allows us to generate questions that involve manipulating numbers and dates. Furthermore, recent work (Yang et al., 2022; Wang et al., 2022) demonstrates that existing table pre-training approaches might encode table row order as an unwanted bias. For example, the pre-trained model being aware of row order information is inclined to select the first or last row of tables when answering superlative-type questions without truly understanding the table content. To alleviate this problem, we randomly shuffled table rows during pre-processing.

2.2 Example Generation

We defined 7 types of table reasoning skills, with examples and explanations shown in Table 1. The example generation pipeline was adapted from Yoran et al. (2021). Each reasoning skill is associated with one example generator and several question templates. The example generator was implemented as a function that takes a table T and generates several reasoning examples (T, $\boldsymbol{q}$, $\boldsymbol{a}$) according to the template, where $\boldsymbol{q}$ denotes the question, and $\boldsymbol{a}$ denotes the answer.

Each template contains typed variables that are instantiated with content from the extracted table. Specifically, column `col` and cell value `val` are indexed to specify that `val:i` must be instantiated by a cell value from the `i`-th column. Some templates also regulate that the selected column and cell value must be date or number type. `OPERATOR` and `ORDINAL` correspond to operators and ordinal numerals that are instantiated according to the specific reasoning skill. And `CONDITION:i` can be 1) a cell value from the `i`-th column; or 2) a number/temporal comparison statement if the `i`-th column is date or number type. For example, the question from Figure 1 "Which Company Name, with Headquarter was United States, has the 4th Profit?" are generated from one of the "Numerical Comparison" templates: "Which `col:1`, with `col:2` was `CONDITION:2`, has

[1] We parsed the 02-20-2022 Wikipedia dump using WikiExtractor Tools from https://github.com/attardi/wikiextractor

Reasoning	Example Templates	Example Questions & Answers	%Data
Conjunction	What was the `col:1` when the `col:2` was `CONDITION:2` and the `col:3` was `CONDITION:3`?	**Q:** What was the **Television Service** when the **Country** was *Italy* and the **Content** was *Sport*? **A:** Sky OMC Sports, ESPN, Gazzetta TV, ...	21.6%
Quantifiers Only/Every	Does `OPERATOR` `col:1`, with `col:2` was `CONDTION:2`, have `col:3` `CONDITION:3`?	**Q:** Does *every* **Company**, with **Headquarter** was *Paris*, have **Industry** *Financials*? **A:** Yes **Q:** Does *only* **Company Name**, with **Founded Year** was *later than 1964*, have **Employee Number** *greater than 30,000*? **A:** No	10.3%
Temporal Comparison	Which `col:1`, with `col:2` was `CONDITION:2`, happened the `ORDINAL` according to `col:3`?	**Q:** Which **Romaji**, with **Sales** was *greater than 203,471*, happened the *4th* according to **Date**? **A:** Hepburn	14.5%
Date Difference	how much time had passed between when the `col:1` was `val:1` and when the `col:2` was `val:2`?	**Q:** how much time had passed between when the **Candidate** was *John Kufuor* and when the **Candidate** was *Paul McCartney*? **A:** 16 years	5.7%
Counting	How many `col:1` have `col:2` `CONDITION:2`?	**Q:** How many **Event Location** have **Attendance** *greater than 10,235*? **A:** 7	18.0%
Numerical Operation	What was the OPERATOR of `col:1` when the `col:2` was `CONDITION:2`?	**Q:** What was the *sum* of **GDP Estimate ($ US Million)** when the **GDP Estimate ($ US Million)** was *greater than 841,969*? **A:** 1,574,013	15.9%
Numerical Comparison	Which `col:1`, with `col:2` was `CONDITION:2`, has the `ORDINAL` `col:3`?	**Q:** Which **Franchise**, with **Owner(s)** was *Nintendo*, has the *5th* **Total revenue($ US Billion)**? **A:** Pokemon	14.0%

Table 1: 7 reasoning skills with example for pre-training REASTAP. Variable names indicate permissible instantiations. `col` denotes a column name, `val` denotes a cell value, and indices denote that a cell value must originate from the specified column. `OPERATOR` and `ORDINAL` correspond to operators and ordinal numeral that are instantiated according to the specific reasoning skill, e.g., for 'Temporal Comparison', `ORDINAL` is replaced with a reasonable ordinal numeral such as "4th". And `CONDITION:i` can be 1) a cell value from the `i`-th column, or 2) number/temporal comparison statement (e.g. "later than 1967") if the `i`-th column is of number or date type.

the `ORDINAL` `col:3`?"

Once all variables in the sampled template were instantiated, we obtained question $\boldsymbol{q}$. Then the example generator would programmatically return the corresponding answer $\boldsymbol{a}$.

2.3 Example Sampling

After generating a vast number of QA examples for each reasoning skill, we had to sample pre-training data from these synthetic examples. In our setting, the portion of pre-training examples (Table 1) corresponding to each reasoning skill roughly matches the portion of logical operations defined in TabFact (Chen et al., 2020b). We raised the portion of numerical operation skill as numerical reasoning is more challenging for models to learn. To increase the diversity of pre-training corpus, for each reasoning skill, we also sampled {SQL query, execution result} pairs from TAPEX (Liu et al., 2022b) pre-training corpus as complementary QA examples. The sampled pairs were categorised according to their function (e.g., `COUNT`, `SUM`). As a result, we obtained a total of 4M pairs of reasoning examples as the pre-training corpus for REASTAP.

3 Pre-training REASTAP

Task Formulation Each example in the synthetic pre-training corpus contains a question $\boldsymbol{q}$ and a semi-structured table T as the model input. The task objective is to generate an accurate answer string $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ given the question $\boldsymbol{q}$ and input table T:

$$\boldsymbol{a} = \operatorname{argmax} \prod_{i=1}^{n} P(a_i | a_{<i}, \boldsymbol{q}, T; \theta), \quad (1)$$

where θ denotes the parameters of a seq2seq LM.

Model Architecture Our method is theoretically applicable to any seq2seq LM, such as T5 (Raf-

fel et al., 2020) and GPT3 (Brown et al., 2020). In our experiments, we implemented REASTAP based on BART (Lewis et al., 2020), a widely used Transformer-based pre-trained model (Vaswani et al., 2017) that has proved its effectiveness on various comprehension and text generation tasks. In our experiments, we chose BART-Large as a backbone, which has around 400M parameters and 12 layers in both encoder and decoder.

Data Serialization As illustrated in Figure 1, the input contains a question and its corresponding table. We flattened the table so that it can be fed directly into the encore-decoder model. Specifically, by inserting several special tokens to indicate the table boundaries, a flattened table is denoted as

$$\begin{aligned} T = & \texttt{[HEAD]}\, h_1 \mid h_2 \mid \cdots \mid h_m \\ & \texttt{[ROW]}\ c_{1,1} \mid c_{1,2} \mid \cdots \mid c_{1,m} \\ & \cdots \\ & \texttt{[ROW]}\ c_{n,1} \mid c_{n,2} \mid \cdots \mid c_{n,m} \end{aligned}$$

where [HEAD] and [ROW] are special tokens indicating the region of table headers and rows respectively. We prefixed the flattened table T with the question and feed them into the model encoder. The decoder is tasked to generate the answer(s), separated by commas, autoregressively.

4 Downstream Tasks

We evaluated REASTAP on three different types of downstream tasks to verify its effectiveness. The statistics and examples for each task are shown in Table 2, and Table 10 in the Appendix, respectively. The fine-tuning of REASTAP is similar to the procedure for pre-training discussed in section 3. We modeled both downstream tasks as sequence generation tasks and leverage generative LMs to generate the output autoregressively.

Table QA WIKISQL-WEAK (Zhong et al., 2017) and WIKITQ (Pasupat and Liang, 2015) were used to evaluate REASTAP performance on Table QA tasks. WIKISQL-WEAK (Zhong et al., 2017) requires the models to perform filtering and optional aggregation on table cell values to answer the given a question. WIKITQ (Pasupat and Liang, 2015) requires a broader set of reasoning skills, thus is more challenging. The Table QA task formulation is the same as the REASTAP pre-training task. We used the denotation accuracy, which checks whether the predicted answers are equal to the ground truths, as evaluation metric.

Table Facts Verification We chose TABFACT (Chen et al., 2020b) to evaluate REASTAP performance on Table Facts Verification tasks. Given a table and a statement, TABFACT (Chen et al., 2020b) tries to distinguish whether the statement is entailed or refuted by the table. TABFACT divides its test sets into $\text{Test}_{\text{simple}}$ and $\text{Test}_{\text{complex}}$ subsets, where $\text{Test}_{\text{complex}}$ contains examples requiring more complex table reasoning skills. Furthermore, it selects a small test set $\text{Test}_{\text{small}}$ with 2K samples for human evaluation. To fine-tune on TABFACT, following BART (Lewis et al., 2020), we applied a binary classifier upon the hidden state of the last token in the decoder for the output. The objective is to generate the verification label $L \in \{0, 1\}$ given the statement $\boldsymbol{s} = (s_1, s_2, \cdots, s_n)$ and the input table T:

$$L = \underset{i \in \{0,1\}}{\text{argmax}}\, P(i \mid \boldsymbol{s}, T\,; \theta) \tag{2}$$

We used the accuracy (i.e., percentage of correct predictions) as evaluation metric.

Faithful Table-to-Text Generation We chose LOGICNLG (Chen et al., 2020a) to evaluate REASTAP performance on the Faithful Table-to-Text Generation task. Compared with previous Table-to-Text generation benchmarks (Wiseman et al., 2017; Balakrishnan et al., 2019; Parikh et al., 2020; Nan et al., 2022b), which primarily focus on surface-level realizations without much logical inference, LOGICNLG is tasked to generate statements that are logically entailed by the selected table region. Given the serialized input table with its selected columns as T, the objective is to generate a sentence $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ that is both fluent and factually correct:

$$\boldsymbol{y} = \text{argmax} \prod_{i=1}^{n} P(y_i | y_{<i}, T\,; \theta) \tag{3}$$

To evaluate the logical fidelity of generated sentences, Chen et al. (2020a) proposed two model-based evaluation methods: Parsing-based Evaluation (SP-Acc), and NLI-based Evaluation (NLI-Acc). SP-Acc directly extracts the meaning representation from the generated sentence and executes it against the table to verify the correctness. NLI-ACC uses a Natural Language Inference (NLI) model to predict entailment relationships. Following Chen et al. (2020a), we used SP-Acc, NLI-Acc as logical-fidelity evaluation metrics; and BLEU-1/2/3 as surface-level evaluation metrics. It is worth

Task	Dataset	# Examples	# Tables	Input	Output
Question Answering	WIKISQL-WEAK (Zhong et al., 2017)	80,654	24,241	Question	Answer
	WIKITQ (Pasupat and Liang, 2015)	22,033	2,108	Question	Answer
Fact Verification	TABFACT (Chen et al., 2020b)	118,275	16,573	Statement	Boolean
Faithful Text Generation	LogicNLG (Chen et al., 2020a)	37,015	7,392	Columns	Text

Table 2: Overview of downstream tasks used in this paper.

Model	Dev	Test
Previous Models		
MAPO (Liang et al., 2018)	71.8	72.4
MeRL (Agarwal et al., 2019)	74.9	74.8
LatentAlignment (Wang et al., 2019)	79.4	79.3
HardEM (Min et al., 2019)	84.4	83.9
Pre-trained LMs		
TaPas (Herzig et al., 2020)	85.1	83.6
GraPPa (Yu et al., 2021a)	85.9	84.7
T5-3B (Xie et al., 2022)	-	86.0
TAPEX (Liu et al., 2022b)	89.3	89.2
BART	86.9	86.1
REASTAP	**91.1**	**90.4**

Table 3: Denotation accuracies on WIKISQL-WEAK.

Model	Dev	Test
Previous Models		
MacroGrammer (Zhang et al., 2017)	40.6	43.7
MAPO (Liang et al., 2018)	42.7	43.8
MeRL (Agarwal et al., 2019)	43.2	44.1
LatentAlignment (Wang et al., 2019)	43.7	44.5
IterativeSearch (Dasigi et al., 2019)	43.1	44.7
Pre-trained LMs		
T5-3B (Xie et al., 2022)	-	49.3
TaPas (Herzig et al., 2020)	49.9	50.4
TableFormer (Yang et al., 2022)	51.3	52.6
TaBERT (Yin et al., 2020)	53.0	52.3
GraPPa (Yu et al., 2021a)	51.9	52.7
TAPEX (Liu et al., 2022b)	58.0	57.2
BART	37.1	37.5
REASTAP	**58.3**	**58.6**

Table 4: Denotation accuracies on WIKITQ.

noting that higher BLEU scores do not correlate with better logical fidelity (Nan et al., 2022a).

5 Experiments

5.1 Implementation Details

We implemented our models based on the fairseq library (Ott et al., 2019). We adopted BART-Large as the backbone model. For table pre-training, we synthesized and sampled 4M pairs of reasoning examples. In the following sections, unless specified explicitly, all the experimental results were evaluated under the default settings of 4M reasoning examples and BART-Large configuration. Our pre-training procedure ran 80,000 steps with a batch size of 256, which took about 34 hours on an 8 NVIDIA A5000 24GB cluster. For downstream tasks, the fine-tuning procedure ran 30,000 steps with a batch size of 256. The best pre-training and fine-tuning checkpoints were both selected according to the validation loss.

5.2 Main Results

Table QA On WIKISQL-WEAK, REASTAP outperforms all the baselines (Table 3). Specifically, on the test set of WIKISQL-WEAK, REASTAP achieves a denotation accuracy of 90.4%, which is 4.3% higher than BART and 1.2% higher than the previous best performance. On the more challenging WIKITQ, as shown in Table 4, REASTAP also surpasses the previous best system by 1.4%. It is worth noting that compared to WIKISQL-WEAK, WIKITQ contains much fewer tables and examples, which makes the adaptation of BART to tabular structures more challenging. Further REASTAP obtains an improvement of 21.1% over BART, indicating that in the low data regime, the improvements brought by REASTAP are more significant. We also evaluated REASTAP performance under low-resource settings (Section 6.1).

Table Fact Verification As shown in Table 5, REASTAP also obtains a new state-of-the-art accuracy on all test subsets of TABFACT. For example, it surpasses the previous best system by 0.4% on $\text{Test}_{\text{simple}}$, and 1.0% on $\text{Test}_{\text{complex}}$.

Faithful Table-to-Text Generation Table 6 presents the results on LOGICNLG. It is observed that the BART backbone has already achieved competitive results in terms of both surface-level and logical-fidelity metrics. Compared with the BART backbone, REASTAP obtains slightly lower results on BLEU scores, which is reasonable since we con-

Model	Dev	Test	$\text{Test}_{\text{simple}}$	$\text{Test}_{\text{complex}}$	$\text{Test}_{\text{small}}$
LPA-Ranking (Chen et al., 2020b)	65.1	65.3	78.7	58.5	68.9
LFC (Zhong et al., 2020)	71.8	71.7	85.4	65.1	74.3
HeterTFV (Yang et al., 2020)	72.5	72.3	85.9	65.1	74.2
SAT (Zhang et al., 2020)	73.3	73.2	85.5	67.2	-
TaPas (Herzig et al., 2020)	81.0	81.0	92.3	75.6	83.9
TableFormer (Yang et al., 2022)	82.0	81.6	93.3	75.9	84.6
DecompTaPas (Yang and Zhu, 2021)	82.7	82.7	93.6	77.4	84.7
T5-3B (Xie et al., 2022)	-	83.7	-	-	-
TAPEX (Liu et al., 2022b)	84.2	84.0	93.7	79.1	85.5
BART	81.0	80.5	90.6	75.7	82.3
REASTAP	**85.1**	**84.7**	**94.1**	**80.1**	**86.2**

Table 5: Accuracies on TABFACT dataset.

Model	Surface-level			Logical-fidelity	
	BLEU-1	BLEU-2	BLEU-3	SP-Acc	NLI-Acc
GPT2-TabGen (Chen et al., 2020a)	49.6	28.2	14.2	44.7	74.6
GPT2-Coarse-to-Fine (Chen et al., 2020a)	49.0	28.3	14.6	45.3	76.4
DCVED (Chen et al., 2021a)	49.5	28.6	15.3	43.9	76.9
T5 (Liu et al., 2022a)	52.6	32.6	**19.3**	48.2	80.4
PLOG (Liu et al., 2022a)	51.7	32.3	18.9	48.9	85.5
R2D2 (Nan et al., 2022a)	51.8	32.4	18.6	50.8	85.6
BART	**53.0**	**32.9**	19.2	50.1	83.7
REASTAP	52.5	32.5	18.9	**54.8**	**89.2**

Table 6: Performance on LOGICNLG test set.

tinued pre-training REASTAP on our pre-training corpus that is irrelevant to the text generation task. However, REASTAP significantly improves the logical-fidelity scores, enhancing the SP-Acc and NLI-Acc by 4.7% and 5.5%, respectively. The results demonstrate that REASTAP can also help improve faithful text generation.

6 Analysis

Experimental results on three different kinds of downstream tasks show that REASTAP can broadly improve BART's generic table reasoning capabilities, which could be adapted to different downstream tasks, regardless of whether the tasks are highly similar to the REASTAP pre-training task or not. In this section, we further analyze our approach in terms of various aspects to provide researchers with a deeper insight for future work.

6.1 Low-resource Setting

To further understand how well REASTAP learns table reasoning skills during pre-training, we conducted experiments under the low-resource setting, where we fine-tuned REASTAP on 20% and 5% of downstream task training data. As shown in Table 7, in the low-resource setting, the improve-

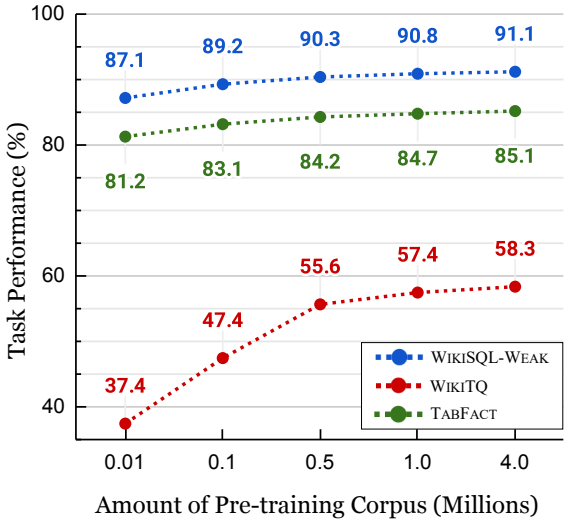

Figure 2: REASTAP downstream task performance on dev set with different scales of pre-training corpus.

%Train	Model	WikiSQL-Weak	WikiTQ	TabFact	LogicNLG		
					BLEU-1/2/3	SP-Acc	NLI-Acc
100%	BART	86.9	37.1	81.0	53.2/33.0/19.3	51.6	84.3
	ReasTAP	91.1 **(+4.2)**	58.3 **(+21.2)**	85.1 **(+4.1)**	52.8/32.7/19.0	55.7 **(+4.1)**	90.1 **(+5.8)**
20%	BART	76.4	21.9	77.4	52.4/32.5/19.1	48.0	80.9
	ReasTAP	86.8 **(+10.4)**	47.3 **(+25.4)**	82.6 **(+5.2)**	52.0/31.9/18.7	53.6 **(+5.6)**	87.2 **(+6.3)**
5%	BART	61.4	14.7	63.5	48.6/27.1/14.5	45.2	75.1
	ReasTAP	81.4 **(+20.0)**	36.1 **(+21.4)**	74.7 **(+11.2)**	47.9/26.3/14.0	50.2 **(+5.0)**	82.5 **(+7.4)**

Table 7: Performance on dev set under low-resource setting. Results show the average over 3 random seeds.

Variant	Dev
BART	37.1
ReasTAP	**57.0**
w/o Date Difference	56.7 (-0.3)
w/o Quantifiers Only/Every	56.4 (-0.6)
w/o Numerical Operation	55.6 (-1.4)
w/o Conjunction	55.4 (-1.6)
w/o Numerical Comparison	55.0 (-2.0)
w/o Temporal Comparison	54.8 (-2.2)
w/o Counting	54.3 (-2.7)

Table 8: WikiTQ dev set denotation accuracy with examples of different reasoning skills removed from the full model pre-training corpus. Each variant is trained using 900K pre-training examples.

Source Task	BART	ReasTAP
-	37.1	58.3
TabFact	42.3 (+5.2)	58.6 (+0.3)
WikiSQL-Weak	47.6 (+10.5)	58.4 (+0.1)

Table 9: Dev denotation accuracy of multi-task fine-tuning on the taget task WikiTQ.

ments introduced by ReasTAP are often more significant. For example, with only 5% training data of downstream tasks, ReasTAP delivers a dramatic improvement of 20.0%, 21.4%, and 11.2% over BART on WikiSQL-Weak, WikiTQ, and TabFact, respectively. The results from the low-resource setting show that ReasTAP endows BART with generic table reasoning capabilities.

6.2 The Scale of Pre-training Corpus

Figure 2 illustrates ReasTAP performance on downstream tasks with different pre-training corpus scales. We found that increasing the pre-training corpus generally brings positive effects for all downstream tasks. Furthermore, for simple tasks like WikiSQL-Weak, the gains by scaling up pre-training corpus are marginal, while for complex tasks like WikiTQ, it shows a positive trend by scaling up the pre-training corpus.

6.3 Necessity of Each Reasoning Skill

We investigated the contributions of the 7 reasoning skills to the downstream task performance of ReasTAP. We devised 8 variants of ReasTAP: one was trained with examples from all reasoning skills, while others were trained with examples without one reasoning skill. For each reasoning skill, we sampled 150K examples from the pre-training corpus. We kept the scale of pre-training corpus the same (i.e., 900K). We chose WikiTQ for experiments, on which BART does not perform well. Results shown in Table 8 demonstrate that all reasoning skills can benefit the model performance on WikiTQ. Furthermore, we find that some reasoning skills, such as counting and temporal comparison, bring more improvements to the model compared to others.

The analysis also helps us understand how the sets of pre-defined reasoning skills are injected during pre-training. When adopting ReasTAP to a new downstream task that requires new reasoning skills different from existing seven reasoning skills, one can also inject the new reasoning skill into model during the pre-training in a similar way. Specifically, once the templates for the new reasoning skill are designed, the synthesis pipeline will generate new examples for pre-training. Pre-training ReasTAP on these synthetic examples can help model learn the new reasoning skill.

6.4 Multi-Task Fine-tuning

We further conducted multi-task fine-tuning experiments to explore whether ReasTAP can benefit from the source task. We chose WikiSQL-Weak and TabFact for the source task, as their training datasets are relatively rich, and WikiTQ as

the target task. Models were first fine-tuned on the source task and then fine-tuned on the target task. As shown in Table 9, multi-task fine-tuning delivers a significant improvement to the target task when initialized by BART; while the improvements are marginal when initialized by REASTAP. This is reasonable because most table reasoning skills acquired by multi-task learning have been injected into the model during the pre-training.

7 Related Work

Reasoning Over Tables Reasoning over the input context is an important requirement for neural models to be applied in the real world, and especially when the input is structured knowledge such as a table. Several Table QA benchmarks (Pasupat and Liang, 2015; Zhong et al., 2017; Iyyer et al., 2017; Chen et al., 2020c) have been proposed to test systems' capability to conduct different types of reasoning, including numerical, logical or multi-hop reasoning. For Table Fact Verification tasks (Chen et al., 2020b; Aly et al., 2021), the models are required to perform logical inference to verify whether the given statement is entailed or refuted. Furthermore, Table-to-Text (Chen et al., 2020a; Parikh et al., 2020; Nan et al., 2022b) tasks to generate a natural language description of some part of the table based on inferences obtained from facts in the contexts. More recently, numerical reasoning over tabular data in financial domain has also raised increasing attention (Zhu et al., 2021; Chen et al., 2021b; Zhao et al., 2022; Cheng et al., 2022; Li et al., 2022; Zhou et al., 2022).

Table Pre-training Inspired by the huge success of pre-training in natural language, researchers have attempted to extend pre-training to structured tabular data (Yin et al., 2020; Herzig et al., 2020; Eisenschlos et al., 2020; Shi et al., 2021; Yu et al., 2021a; Wang et al., 2021b; Deng et al., 2020, 2021; Liu et al., 2022b) in recent years. Previous table pre-training work such as TABERT (Yin et al., 2020) and TAPAS (Herzig et al., 2020; Eisenschlos et al., 2020) took corrupted tables and NL sentences as input and tried to recover the corrupted parts. They had the intuition that such recovering processes can help strengthen the linking between sentences and structured tables. On the other hand, TAPEX (Liu et al., 2022b) learned from synthetic SQL programs. And Jiang et al. (2022) further pre-trained TAPEX over natural and synthetic QA examples to improve the few-shot performance over table QA tasks. Meanwhile, pre-training for Text-to-SQL tasks (Shi et al., 2021; Yu et al., 2021a,b; Deng et al., 2021) also attracted researchers' attention in recent years. Unlike previous work, we model the pre-training task as a sequence generation task, and inject various table reasoning skills into the model by tasking it to generate the precise answers of reasoning examples.

Synthetic Pre-training Corpus Generating a large-scale synthetic pre-training corpus is widely used in both natural language pre-training (Campagna et al., 2020; Geva et al., 2020; Yoran et al., 2021; Neeraja et al., 2021; Yue et al., 2022) and table pre-training (Yu et al., 2021a,b; Wang et al., 2021a; Liu et al., 2022b). For example, Geva et al. (2020) utilized automatically-generated numerical data to inject numerical reasoning skills during pre-training. And Yoran et al. (2021) leveraged large-scale WikiPedia resources to automatically generate examples that requires reasoning over multiple facts in the paragraph, and continue pre-training LM on this synthetic corpus. Furthermore, recent works (Liu et al., 2022b; Pi et al., 2022) showed that pre-training can be achieved by learning a program executor over synthetic corpus.

8 Conclusion

In this paper, we propose REASTAP, a new table pre-training approach, which injects various pre-defined table reasoning skills into models via learning to generate correct answers of synthetic questions. Compared to previous work which design table-specific architectures, REASTAP is easy to implement and is theoretically applicable to any sequence-to-sequence LM. REASTAP is evaluated over four downstream benchmarks. The experimental results demonstrate that REASTAP achieves new state-of-the-art results on each of them. This includes the improvements on WIKISQL-WEAK denotation accuracy to 90.4% (+1.2%); WIKITQ denotation accuracy to 58.6% (+1.4%); TABFACT accuracy to 84.7% (+0.7%); and LOGICNLG SP-Acc to 54.8% (+4.0%), NLI-Acc to 89.2% (+3.6%). Further analysis demonstrates that REASTAP delivers a significant improvement to BART on the low-resource setting, indicating that our proposed pre-training approach can effectively improve the model's generic table reasoning capabilities.

Limitations

The main limitation of our approash is that we utilized a template-based method to synthesize pre-training corpus. Although such template-based approach ensures the faithfulness of generated QA examples and the diversity of reasoning process required to answer the questions, it limits the semantic diversity of questions. We believe future work could exploit 1) more different types of reasoning skills, such as advanced numerical reasoning skills required in the finance domain (Zhu et al., 2021; Chen et al., 2021b); 2) a more universal synthetic example generation pipeline; 3) extending models to tables with hierarchical structures (e.g., more than one row or column header) (Cheng et al., 2022; Zhao et al., 2022); 4) a more efficient training framework (Biesialska et al., 2020; Yoran et al., 2021) that can update models to learn newly-defined reasoning skills effectively.

Ethical Consideration

Tables used in our synthetic pre-training corpus are collected and extracted from the 02-20-2022 Wikipedia dump[2], which is publicly available under the Creative Commons Attribution-ShareAlike 3.0 License and the GNU Free Documentation License. The licenses permit us to compose, modify, publish, and distribute additional annotations upon the original content.

Acknowledgements

We would like to thank the anonymous reviewers and action editors for their constructive feedback.

References

Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. 2019. Learning to generalize from sparse and underspecified rewards. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 130–140. PMLR.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Florence, Italy. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. 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 Wang. 2020c. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *Findings of EMNLP 2020*.

Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2021a. De-confounded variational encoder-decoder for logical table-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5532–5542, Online. Association for Computational Linguistics.

[2] https://archive.org/details/enwiki-20220220

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. Finqa: A dataset of numerical reasoning over financial data. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. 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.

Pradeep Dasigi, Matt Gardner, Shikhar Murty, Luke Zettlemoyer, and Eduard Hovy. 2019. Iterative search for weakly supervised semantic parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2669–2680, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online. Association for Computational Linguistics.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding through representation learning. *Proc. VLDB Endow.*, 14(3):307–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, 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.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Moxin Li, Fuli Feng, Hanwang Zhang, Xiangnan He, Fengbin Zhu, and Tat-Seng Chua. 2022. Learning to imagine: Integrating counterfactual thinking in neural discrete reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 57–69, Dublin, Ireland. Association for Computational Linguistics.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10015–10027, Red Hook, NY, USA. Curran Associates Inc.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022a. Plog: Table-to-logic pretraining for logical table-to-text generation. *arXiv preprint arXiv:2205.12697*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022b.

TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Linyong Nan, Lorenzo Jaime Yu Flores, Yilun Zhao, Yixin Liu, Luke Benson, Weijin Zou, and Dragomir Radev. 2022a. R2d2: Robust data-to-text with replacement detection. *arXiv preprint arXiv:2205.12467*.

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

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating external knowledge to enhance tabular reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2799–2809, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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.

Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin, Yan Gao, Qiang Fu, Jian-Guang Lou, and Weizhu Chen. 2022. Reasoning like program executors. *arXiv preprint arXiv:2201.11473*.

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

Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. 2021. Learning contextual representations for semantic parsing with generation-augmented pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13806–13814.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bailin Wang, Ivan Titov, and Mirella Lapata. 2019. Learning semantic parsers from denotations with latent structured alignments and abstract programs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3774–3785, Hong Kong, China. Association for Computational Linguistics.

Bailin Wang, Wenpeng Yin, Xi Victoria Lin, and Caiming Xiong. 2021a. Learning to synthesize data for semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2760–2766, Online. Association for Computational Linguistics.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021b. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. 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. *arXiv preprint arXiv:2201.05966*.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Xiaoyu Yang, Feng Nie, Yufei Feng, Quan Liu, Zhigang Chen, and Xiaodan Zhu. 2020. Program enhanced fact verification with verbalization and graph attention network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7810–7825, Online. Association for Computational Linguistics.

Xiaoyu Yang and Xiaodan Zhu. 2021. Exploring decomposition for table-based fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1045–1052, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Ori Yoran, Alon Talmor, and Jonathan Berant. 2021. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. *arXiv preprint arXiv:2107.07261*.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, bailin wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, richard socher, and Caiming Xiong. 2021a. Gra{pp}a: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*.

Tao Yu, Rui Zhang, Alex Polozov, Christopher Meek, and Ahmed Hassan Awadallah. 2021b. {SC}ore: Pre-training for context representation in conversational semantic parsing. In *International Conference on Learning Representations*.

Xiang Yue, Ziyu Yao, and Huan Sun. 2022. Synthetic question value estimation for domain adaptation of question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1340–1351, Dublin, Ireland. Association for Computational Linguistics.

Vicky Zayats, Kristina Toutanova, and Mari Ostendorf. 2021. Representations for question answering from documents with tables and text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2895–2906, Online. Association for Computational Linguistics.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1629, Online. Association for Computational Linguistics.

Yuchen Zhang, Panupong Pasupat, and Percy Liang. 2017. Macro grammars and holistic triggering for efficient semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, Copenhagen, Denmark. Association for Computational Linguistics.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

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

Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. LogicalFactChecker: Leveraging logical operations for fact checking with graph module network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6065, Online. Association for Computational Linguistics.

Fan Zhou, Mengkang Hu, Haoyu Dong, Zhoujun Cheng, Shi Han, and Dongmei Zhang. 2022. Tacube: Pre-computing data cubes for answering numerical-reasoning questions over tabular data. *arXiv preprint arXiv:2205.12682*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287.

Dataset	Example Input	Example Output
WIKISQL-WEAK	How many players played for adams state school? [HEAD] Pick # | CFL team | Player | Position | College [ROW] 145 | calgary stampeders | brett ralph | wr | alberta [ROW] 246 | ottawa renegades | lenard semajuste | fb | adam state . . .	3
WIKITQ	Which coach served previous to ardis smith? [HEAD] Tenure | Coach | Years | Record | Pct. [ROW] 11892 | Shelby Fletcher | 1 | 1–0 | 1.000 [ROW] 21893 | W. M. Walker | 1 | 4–6–1 | .409 . . .	F. C. Owen
TABFACT	John E. Moss and Phillip Burton are both re-elected in the house of representative election. [HEAD] District | Incumbent | Party | Result | Candidates [ROW] Clifornia 3 | John E. Moss | democratic | re-elected | JohnE. Moss (d) 69.9% John Rakus (r) 30.1% [ROW] California 5 | Phillip Burton | democratic | re-elected | Phillip Burton (d) 81.8% Edlo E. Powell (r) 18.2% . . .	1 (entailed)
LOGICNLG	Players for Jazz. [HEAD] Player ǀ School / Club Team ǀ Year [ROW] Adrian Dantley ǀ Notre Dame ǀ 1979 - 1986 [ROW] Brad Davis ǀ Maryland ǀ 1982 - 1984. . .	John Duren played for Utah Jazz for 2 years.

Table 10: The example inputs and outputs for our model on experimental datasets.