Generation-focused Table-based Intermediate Pre-training for Free-form Question Answering

Peng Shi[1*], **Patrick Ng**[2], **Feng Nan**[2], **Henghui Zhu**[2], **Jun Wang**[2], **Jiarong Jiang**[2], **Alexander Hanbo Li**[2], **Rishav Chakravarti**[2], **Donald Weidner**[2], **Bing Xiang**[2], **Zhiguo Wang**[2]

[1] University of Waterloo, [2] AWS AI Labs
peng.shi@uwaterloo.ca, patricng@amazon.com

Abstract

Question answering over semi-structured tables has attracted significant attention in the NLP community. However, most of the existing work focus on questions that can be answered with short-form answer, i.e. the answer is often a table cell or aggregation of multiple cells. This can mismatch with the intents of users who want to ask more complex questions that require free-form answers such as explanations. To bridge the gap, most recently, pre-trained sequence-to-sequence language models such as T5 are used for generating free-form answers based on the question and table inputs. However, these pre-trained language models have weaker encoding abilities over table cells and schema. To mitigate this issue, in this work, we present an intermediate pre-training framework, Generation-focused Table-based Intermediate Pre-training (GENTAP), that jointly learns representations of natural language questions and tables. GENTAP learns to generate via two training objectives to enhance the question understanding and table representation abilities for complex questions. Based on experimental results, models that leverage GENTAP framework outperform the existing baselines on FETAQA benchmark. The pre-trained models are not only useful for free-form question answering, but also for few-shot data-to-text generation task, thus showing good transfer ability by obtaining new state-of-the-art results.

Introduction

Question Answering (QA) (Rajpurkar et al. 2016; Krishna, Roy, and Iyyer 2021) is an important natural language processing task that enables the interactions between the users and large-scale knowledge sources. Based on the different forms of the knowledge sources, the QA task is categorized into different sub-tasks, such as Text-based QA that answer questions based on the unstructured texts, Table-based QA where semi-structure tables are the knowledge source, and Semantic Parsing where logic-form is generated to answer question from structured knowledge graphs and databases.

For Text-based QA and Table-based QA, existing work primarily focused on extracting relevant portion of the text/table to answer the question, which are usually short-form facts or entities (Rajpurkar et al. 2016; Pasupat and Liang 2015; Iyyer, Yih, and Chang 2017). However, these QA systems may not meet the needs of the users, who tend to ask more complex questions[1] that require free-form answers (e.g. explanations) rather than short entities.

Efforts have been made in addressing the shortcoming of the QA systems. For the Text-based QA, Kočiskỳ et al. (2018); Fan et al. (2019); Krishna, Roy, and Iyyer (2021) proposed to leveraged sequence-to-sequence architectures to generate free-form answers based on the retrieved documents. However, the free-form Table-based QA remains largely unexplored. More recently, Nan et al. (2021) used pre-trained language model T5 (Raffel et al. 2019) — a sequence-to-sequence architecture — to generate long form answers from the table knowledge source.

However, the sequence-to-sequence pre-trained language models, such as BART (Lewis et al. 2019) or T5 (Raffel et al. 2019), have weaker encoding ability over table cells and schema. These language models usually employ long documents as the training corpus, obtaining impressive encoding ability over unstructured text. On the other hand, tabular data have their own structures to express the semantics, which are usually not captured by these language models.

Recently, several solutions are proposed for alleviating aforementioned issue by introducing pre-training or intermediate training strategies for tables. For example, Herzig et al. (2020) proposed TAPAS that used Masked Language Model (MLM) as pre-training objective for improving the contextual representation of BERT (Devlin et al. 2018) over table inputs. They showed the pre-trained model obtained state-of-the-art performance for Table-based QA where entities are extracted from the table. They achieved large improvements over the table entailment task. Albeit the improvements, these pre-training models were designed and evaluated for short-form answer, where the answer is often a table cell or aggregation of multiple cells. Thus pre-training strategies to solve complex questions that require long-form answers remain unexplored.

In this work, we present an intermediate language model pre-training framework, Generation-focused Table-based Intermediate Pre-training (GENTAP), that exploits different learning strategies, including short-form entities and long-form explanations. We demonstrate that our learning strate-

[*]Work done while at AWS AI Labs.

[1]Complex question in our work refers to the question that requires long-form explanation to answer.

gies enhance question understanding and table representation abilities of the pre-trained language models for complex questions. Instead of using bidirectional contextual encoder such as BERT to exploit the potential on the text generation task, our framework is based on the BART (Lewis et al. 2019) encoder-decoder architecture, which was trained with denoising training objectives. Specifically, our two different learning targets are designed for improving different aspects of the pre-trained language model, including, but not limited to, long-form answer generation augmentation (LongAug) and factual accurate answer generation augmentation (ShortAug). *LongAug* leverages table knowledge enriched long sentence as the learning target. *ShortAug* uses short entities that precisely answer the corresponding question as target; this learning target is to improve the model's accuracy in generating key facts based on the knowledge contained in the table.

One key challenge to employ the aforementioned intermediate pre-training tasks is the training data. Although it is easy to obtain large scale tables from web sources such as Wikipedia Tables, it is difficult to obtain the questions and answers (long form or short form) pairs that are interrelated with the tables. Recent work used the surrounding text of the tables as a proxy for related natural language utterances (Herzig et al. 2020; Yin et al. 2020). However, this causes a mismatch between the intermediate pre-training and downstream tasks where questions are one essential component of the tasks. More recently, Shi et al. (2020) confirmed that the surrounding text is far from optimal because those texts are dissimilar to the natural language questions in terms of text length, composition and content. The surrounding text of the tables can be quite noisy and may be irrelevant to the tables. In this work, following Shi et al. (2020) and Eisenschlos, Krichene, and Müller (2020), we leverage both sequence-to-sequence generation model and synchronous context-free grammar to generate the question-answer pairs for intermediate pre-training.

The outcome of the GENTAP is a sequence-to-sequence pre-trained model that have the enhanced ability for generating long-form answers for complex questions from tabular knowledge sources. The experimental results show that the models outperform the state-of-the-art models on FeTaQA dataset. We also find that our models have transfer ability for the few-shot data-to-text generation task by outperforming existing baselines. In summary, our work shows the following contributions:

- We propose a new framework for table-based long-form answer generation that exploits two different learning targets with synthetic data.
- We leverage a novel strategy to overcome pre-training data challenges by leveraging generative model and synchronous context-free grammar to generate synthetic data for learning joint representations of textual data and table.
- Our pre-trained model obtains state-of-the-art performance on the table-based free-form question answering dataset FeTaQA .
- Our pre-trained model demonstrates good transfer ability by achieving better effectiveness than baselines on few-shot data-to-text (FSD2T) generation task .

Our code and synthetic data will be made publicly available upon publication.

Models

Baseline Models To answer complex questions based on tabular content, one of the two methods is usually exploited: pipeline model and end-to-end model. For the pipeline model, a semantic parser is first leveraged to generate denotations (which are usually entities from the table), and then a data-to-text generation model is used to compose a coherent and fluent sentence from the table schema and denotations. This pipeline model relies heavily on the semantic parser to produce accurate denotations; otherwise error propagation may lead to poor performance. The second method, an end-to-end model, is formulated as a sequence-to-sequence learning problem where free-form answers are directly generated conditioned on the question and table input, without producing intermediate results. Nan et al. (2021) showed the latter approach yielded significantly better performance.

Thus, in this work, we use the BART sequence-to-sequence pre-trained language model as our baseline architecture, by leveraging its potential on text generation. More specifically, the table is linearized into a sequence T by separating the rows with special token `[ROW]` and separating cell values with vertical bar. This linearized table is appended to the question tokens q with `[SEP]` in between. In addition, we provide the positional embeddings for each token, including the segment embedding (for question segment and table segment), row embedding and column embedding (Herzig et al. 2020). These embeddings are added on top of the token embeddings as model inputs and optimized during the training. The free-form answer is regarded as target sequence. The Data-to-Text generation task is similar to the Free-form Question Answering, just without the prepended question. The input of the sequence-to-sequence model is the linearized table and the learning target is the table summary. We can regard a hidden question *"What is the summary of the table?"* is prepended.

Intermediate Pre-training For the pre-training model, we use a similar architecture as the baseline systems. The questions and tables are fed into the transformer encoder; the tables are linearized with same strategy as the baseline systems.

Two types of augmentations are employed in the intermediate pre-training stages: LongAug and ShortAug. In the LongAug, table-enriched sentences are regarded as our learning target, where the sentences express some facts that are based on some parts of the table. This learning target is expected to improve (include but not limit to) the natural sentence generation ability in the context of table-based question answering scenario. In the ShortAug, short entities are the learning target. If multiple entities are generated, they are separated with vertical bars. This learning target is expected to help the model to improve the factual accuracy of the pre-trained models. Because the essential component in the long-form answer is still the key entities that

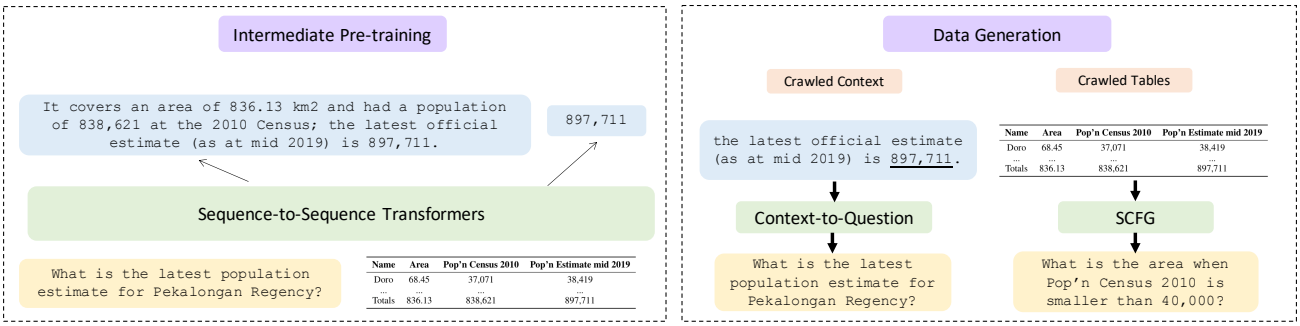

Figure 1: GenTaP Framework. The left figure shows our Intermediate Pre-training stages: LongAug and ShortAug. The right figure shows our synthetic training data generation methods: Context-to-Question for LongAug, and SCFG for ShortAug.

answer the questions. In terms of model architecture, we use same architecture as the baseline model, a positional embedding augmented sequence-to-sequence model. Note that during pre-training, we use two separate decoders for these two learning targets, and the model is trained with multitask learning fashion. Our preliminary experimental results show that two separate decoders outperformed unified decoder.

Pre-training Data Synthesis

Data is one key part in this intermediate pre-training. As discussed, the question-answer (long or short form) pairs are expected in our pre-training stage, while they are not available in large scale for representation learning. In this work, we exploit two methods for synthesizing the pairs from large scale tables from Wikipedia: Context-to-Question Generation and Synchronous Context-free Grammar.

LongAug Synthetic Data: The target of Context-to-Question Generation is to synthesize *(Question, Long-form Answer)* pairs for intermediate pre-training stage LongAug. For each table we crawled from the Wikipedia page, we retrieve the statements that are relevant to the specific table from Wikipedia articles. We note these statements as *table knowledge enriched sentences* and these sentences are used as the proxy for long-form answers. Because the relevant statements usually come from the same article as the table appears in, we only consider each sentence in the specific Wikipedia page, without examining other articles. We compute the relevance level for each sentence and the table, by using the lexical matching strategy: if there are several cell values in the table appearing in the sentence (more than the threshold), we regard it as a relevant statement candidate. We note these overlapped entities as *key entities*. For each key entity, we generate a question for it by leveraging a context-to-question generator.

In particular, the input of the generator is the table knowledge enriched sentence and the key entity; the output of the generator is the corresponding question — see Figure 4 for an example. We use the BART model as the generator. To train the generator, we use the SQUAD (Rajpurkar et al. 2016) dataset. The SQUAD dataset is designed for reading comprehension task where (question, paragraph, short-form answer) triples are provided. We adapt the SQUAD dataset for our purpose: for each example, we first identify the sentence from the paragraph where the short-form answer is found; the input to train the generator is the concatenation of the article title, the identified sentence and short-form answer; the training target is the question. In this way, we generate large scale (question, table, long-form answer) triples by leveraging the alignment between the table and the context and context-to-question generator.

$$
\begin{aligned}
\text{[question]} &\rightarrow \text{What is [select] when [where]} \mid \\
&\quad\ \text{What is [select]} \\
\text{[select]} &\rightarrow \text{the [column]} \mid \\
&\quad\ \text{the [aggregation] of the [column]} \\
\text{[where]} &\rightarrow \text{[column] [comparison] [value]} \mid \\
&\quad\ \text{[where] and [where]} \\
\text{[aggregation]} &\rightarrow \text{smallest} \mid \text{largest} \mid \text{sum} \mid \text{average} \\
\text{[comparison]} &\rightarrow \text{is} \mid \text{is smaller than} \mid \text{is larger than}
\end{aligned}
$$

Figure 2: The SCFG for ShortAug Data Sampling.

ShortAug Synthetic Data: Similar to Eisenschlos, Krichene, and Müller (2020), we build table-dependent question that are SQL-like. We define a synchronous context-free grammar (SCFG) as shown in Figure 2 and questions are sampled from it. The corresponding answers can be easily obtained during the sampling process. These answers are all cell values from the table, or the numerical aggregation results such as `SUM`, `MAX` and `MIN`. As the example shown in right side of data generation in Figure 1, a question "What is the [area] when [Pop'n Census 2010] is smaller than 40,000" can be composed based on the table. In this way, we synthesize large scale (question, table, short-form answer) triples for the intermediate pre-training stage ShortAug.

Experiments

For all experiments, we train our GENTAP MODEL with underlying transformers initialized with BART-large model (Lewis et al. 2019). 250K LongAug examples are generated via Context-to-Question Generation and 250K ShortAug examples are generated via SCFG. The tables that are used in the downstream tasks are removed in the pre-training stage.

Tasks, Datasets and Baselines. We evaluate our model on the FETAQA (Nan et al. 2021) dataset. FETAQA is a table-

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
TAPAS + T5-large	11.00	0.40	0.22	0.35	0.24
T5-small (fine-tuned by Nan et al. (2021))	21.60	0.55	0.33	0.47	0.40
T5-base (fine-tuned by Nan et al. (2021))	28.14	0.61	0.39	0.51	0.47
T5-large (fine-tuned by Nan et al. (2021))	30.54	0.63	0.41	0.53	0.49
BART (fine-tuned by us)	32.14	0.658	0.432	0.551	0.512
Zero-shot (ours)	27.12	0.566	0.351	0.469	0.422
GenTaP (ours)	**36.74**	**0.689**	**0.476**	**0.587**	**0.545**
- ShortAug	36.07	0.683	0.470	0.582	0.541
- LongAug & ShortAug	33.87	0.668	0.443	0.563	0.520

Table 1: Results on the test split of FeTaQA dataset.

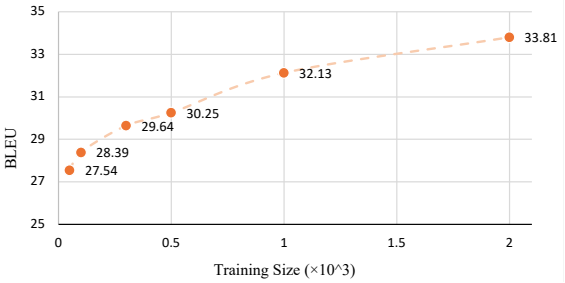

Figure 3: Low-data regimes. We finetuned GenTaP on 50, 100, 300, 500, 1000 and 2000 sampled training examples.

Model	Precision	Recall
T5-small	-2.8093	-2.3946
T5-base	-2.4989	-2.2686
T5-large	-2.3428	-2.1451
Zero-shot	-2.8333	-2.3555
GenTaP	**-2.0627**	**-1.8609**
- ShortAug	-2.0801	-1.8932
- LongAug & ShortAug	-2.1482	-1.9941

Table 2: BARTScore results on FeTaQA test split. Scores are shown in log probability. Higher is better.

based free-form question answering dataset that contains large scale *(question, table, long-form answer, supporting table cells)* pairs. Compared with WikiSQL (Zhong, Xiong, and Socher 2017) or WTQ (Pasupat and Liang 2015), the questions in FeTaQA are more complex — requiring elaborations and explanations. The state-of-the-art systems on FeTaQA are based on the T5-large end-to-end models that generate answers directly from the question and table inputs. We also compare our models with pipeline baselines that first leverage state-of-the-art weakly supervised parser TAPAS (Herzig et al. 2020) to generate denotations, and then leverage the T5-large as data-to-text generator.

We also evaluate transfer ability of our model by testing it on the few-shot Data-to-Text generation task. That is, we examine if our pre-training model is helpful on the related task of generating natural sentences based on the knowledge of table. We evaluate our model on Data-to-Text generation Dataset (FSD2T) (Chen et al. 2019). The FSD2T includes data in three different domains, including the Humans, Books and Songs. We experiment on different training size, including 50, 100, 200 and 500 training examples in each domain. The models are chosen based on the performance of the development set with 1000 examples. Test sets for Humans, Books and Songs consist of 13587, 5252, and 11879 examples. We compared our models with BASE (Chen et al. 2019), BASE+SWITCH+LM (Chen et al. 2019), and TABLEGPT (Gong et al. 2020) that are all based on GPT2 (Radford et al. 2019).

Results

FeTaQA Main Results. The main results of FeTaQA dataset are shown in Table 1. We evaluate the models with *unsupervised matching* in the *discrete string space* (Yuan, Neubig, and Liu 2021), such as BLEU, ROUGE-{1,2,L} and METEOR. The previous state-of-the-art performance is obtained by T5-large with 770M parameters, which achieves 30.54 BLEU score, outperforming other variants of T5 such as T5-base (220M parameters) and T5-small (60M parameters). For ROUGE-1, ROUGE-2, ROUGE-L and METEOR, the T5-large achieves 0.63, 0.41, 0.53 and 0.49 respectively. For the baseline that leverages the table-based pre-trained model such as TAPAS, the experimental results are obtained with the TAPAS + T5-large architecture. TAPAS + T5-large is a pipeline architecture that leverages the state-of-the-art models in two worlds: the weakly semantic parsing and the data-to-text generation. The model firstly extracts denotations (key entities) based on the questions and tables input. Then a trained T5-large model performs the data-to-text generation based on the produced denotations, together with other meta information of the tables. This baseline only obtains 11.00 BLEU score, due to imperfect parsing system and error propagation issue.

Our framework is based on the BART architecture with 406M parameters, that is smaller than the T5-large architecture. We finetune the BART model on the dataset, obtaining 32.14 BLEU score, exceeding the state-of-the-art T5-large model. For other metrics, our finetuned BART model also achieves new state-of-the-art performance. Augmenting with our pre-trained GENTAP model, the performance is further improved by large margins on different evaluate metrics, reaching 36.74 BLEU score, and 0.689, 0.476, 0.587, 0.545 on the ROUGE-{1,2,L} and METEOR, respectively.

Model	Lexical level F1	Tuple level F1
T5-large	0.722	0.509
BART fine-tuned	0.725	0.515
GenTaP	**0.767**	**0.558**
- ShortAug	0.755	0.554
- LongAug & ShortAug	0.746	0.538

Table 3: Factual Consistency Evaluation.

Zero-shot and Few-shot FeTaQA Results. Based on our intermediate pre-training objectives, our trained models already have the ability of answering the questions with free-form statements. Therefore, it is interesting to evaluate the zero-shot performance of the pre-trained models. Without finetuning, we directly feed the FeTaQA test set into the model and produce the answers. The results are shown in Zero-shot entry in Table 1, with 27.12 BLEU score and 0.566, 0.351, 0.469, 0.422 on the metrics of ROUGE-{1,2,L} and METEOR, respectively. Hence, the performance is on par with fully supervised T5-small model.

Through experiments in low-data regimes, we find that our pre-trained GENTAP model is an efficient learner. We finetuned GENTAP on 50, 100, 300, 500, 1000 and 2000 sampled training examples. Experimental results are shown in Figure 3. Using just 100-300 training examples, the model can achieve comparable performance against the T5-base model; while with 1000-2000 training examples, the model can obtain the similar effectiveness against the supervised BART baseline.

Model-based Evaluation. Leveraging large scale pre-trained language model to evaluate the performance of generation models has become popular as its metric has been shown to have high correlation with human judgement. In this work, we further evaluate the models with the recent work, BARTScore (Yuan, Neubig, and Liu 2021). Instead of relying on *token-level matching* on the *discrete string space*, the BARTScore formulates evaluating generated text as a text generation task from pre-trained language models. The log probability of BART generator is used to evaluate the quality of hypothesises (h) based on the references (r). Based on different input-output pairs, the following metrics can be evaluated by using the BARTScore. 1) **Precision**: The encoder input is the reference text and the decoder input is the generated text. The $P(h|r)$ is calculated and it accesses how likely the hypothesis can be generated based on the reference input. 2) **Recall**: The encoder input is the generated text and the decoder input is the reference text. The $P(r|h)$ is evaluated and it calculates how many semantic content units are covered by the hypothesis. We use the BART finetuned on ParaBank2 as the evaluation checkpoint.[2]

We evaluate the predictions[3] of T5 models and compared against our models. As shown in the top section of Table 2, the T5-large obtains -2.3428 precision and -2.1451 recall. With FETAQA dataset finetuning, our GENTAP model obtains the best performance with -2.0627 precision and -1.8609 recall. Unsurprisingly, it also outperforms the zero-shot evaluation significantly, where the precision and recall scores are -2.8333 and -2.3555, respectively.

[2]https://github.com/neulab/BARTScore

[3]https://github.com/Yale-LILY/FeTaQA

Model	Average	#Score ≥ 4	Agreement
GenTaP	3.84	32	0.82
- ShortAug	3.70	30	0.81
- LongAug & ShortAug	3.42	27	0.85

Table 4: Human evaluation on 50 samples of FETAQA predicted instances on a 1-5 scale.

Are the generated free-form answers factually consistent? While metrics such as BLEU and ROUGE often serve as the primary metrics for assessing the quality of generated text, these metrics have been shown to be sometimes poorly correlated with answer correctness (Dhingra et al. 2019). As a result, we leverage an alternate evaluation criteria which leverages the highlighted cells from the FETAQA dataset's annotations. The highlighted cells are intended to capture key entities that the free-form answer should ideally make use of. So we measure the precision and recall of these key entities in the generated answer text. More specifically, we regard the highlighted cells that appear in the references as reference entity set; we extract the key entities from the generated text with string matching, denoted as hypothesis entity set. The precision, recall and F1 scores based on these two sets can be calculated; we call these scores are in lexical level. We can further regard the key entities that are from the same table row as a tuple; a tuple is correct only when all entities in the tuple are correct. Thus we can evaluate the tuple level precision, recall and F1 score. This is stricter evaluation for the models. These results are shown in Table 3 and demonstrate an improvement when GENTAP is used for pre-training. Our GENTAP obtains the 0.767 on the lexical level F1 score and 0.558 on the tuple level F1 score, outperforming the state-of-the-art T5-large model by large margin.

Human Evaluation. To further evaluate the quality of the answers generated by the models, we conducted human evaluation based on the following criteria. We asked internal annotators to evaluate 50 samples of FETAQA instances on a 1-5 scale. The results are shown in Table 4. The average score of the answers is 3.84, with 32 out of 50 answers obtaining 4 or 5. Cohen Kappa is calculated for showing the agreement of annotators.

Error analysis. To further understand the performance and behaviors of the models, we investigated the errors the models made. We classify the errors into the following types: lookup error and aggregation error. For the *lookup error*, the models fail to retrieve relevant rows/columns based on the header mentions or conditions. As shown in the Table 6, the two examples belong to this category. The question in the Example 1 requires the model to understand the condition "*between Barnyard and Grown Ups*" and retrieve the relevant rows in between from the table. The baseline model fails to understand the question and just extracts the information of movie "*Barnyard*" and "*Grown Ups*". Our GENTAP model is partially correct based on the answer it gener-

Domain	Humans				Books				Songs			
# of training instances	50	100	200	500	50	100	200	500	50	100	200	500
GPT (Switch + LM)	25.7	29.5	36.1	41.7	34.3	36.2	37.9	40.3	36.1	37.2	39.4	42.2
Table-GPT	29.8	34.5	40.6	45.6	35.1	37.3	38.5	41.6	36.7	37.8	39.3	42.3
GenTaP (ours)	**39.4**	**45.9**	**47.4**	**50.8**	**39.8**	**41.6**	**43.1**	**46.7**	**38.3**	**42.0**	**44.0**	**45.1**
- LongAug & ShortAug	37.5	44.1	46.5	50.1	37.9	40.8	40.4	46.6	36.7	40.7	42.7	43.6

Table 5: Few-Shot Data-to-Text Generation results on different domains.

Example 1
Question: What films did Kevin James star in between Barnyard and Grown Ups?
Reference: James starred in I Now Pronounce You Chuck and Larry (2007) and Paul Blart: Mall Cop (2009) between Barnyard and Grown Ups.
Baseline: In 2006, Kevin James starred in Barnyard, and wrote, directed and starred in Grown Ups.
Our Model: Kevin James starred in Barnyard (2006) and I Now Pronounce You Chuck & Larry (2007).

Example 2
Question: Which animated characters were designed by Glen Keane in 1989 and 1990?
Reference: Glen Keane designed and animated the character of Ariel in the film The Little Mermaid (1989) and Marahute in The Rescuers Down Under (1990).
Baseline: Glen Keane designed the characters for The Little Mermaid (1989) and The Rescuers Down Under (1990).
Our Model: Glen Keane designed Ariel in The Little Mermaid (1989) and Marahute in The Rescuers Down Under (1990).

Table 6: Selected Examples for FeTaQA. Our Model refers to GENTAPwhile the Baseline refers to positional embedding augmented BART model without pre-training.

ates. It retrieves the movie "*I Now Pronounce You Chuck & Larry*" that is after the "*Barnyard*" but misses the other one. The question in the Example 2 asks the model to provide the information about the "*animated characters*". Our GENTAP model provides the corresponding information "*Ariel*" and "*Marahute*", however, the baseline does not answer with these key entities. On the other hand, the *aggregation type* questions are hard for the models. For example, the question "How much overall damage did the German submarine U-438 cause?" required the model to calculate the sum of the tonnage of the submarines and all the models failed. Further improving this type of questions is left for future work.

Few-Shot FSD2T Main Results. The results of few-shot data-to-text generation task are shown in Table 5. We can observe that our baseline models already achieve the state-of-the-art BLEU score all three domains under different training settings. For our GenTaP models, even it is not pre-trained for the question answering purpose, the models showed good transfer ability by further improving the performance. By comparing the different training size, we can observe that with fewer training examples, such as 50 or 100, the model has larger improvement margins. When the training size is larger such as 500, the improvements are less significant.

Aspect	#Counts	Agreement
Alignment	18.5	0.81
Correctness	30.5	0.84

Table 7: Human evaluation of LongAug synthetic data quality

Training Size	10K	50K	100k	250K
BLEU	34.58	35.01	35.49	36.07

Table 8: BLEU scores on different LongAug synthetic data sizes

Ablation Study for Pre-training

Data Synthesis Quality. The LongAug synthetic data generator — Context-to-Question Generation — obtains 21.52 BLEU score on the SQUAD validation set. To assess the quality of the pre-training data, we further sampled 50 examples from the generated *(question, table, context)* triples and ask NLP experts for judgement. We evaluate the data in the following aspects: 1) **Alignment**: whether the context is supported by the facts from the table. Because the contexts are aligned with tables automatically, false positive error will be introduced. 2) **Correctness**: whether the generated question is correct based on the context and sampled answer span. This evaluates the correctness aspect of the question generator. The human evaluation results are shown in Table 7. Out of 50 examples, there are 18.5 (averagely) context sentences aligning with the table, with 0.81 Cohen Kappa score. This indicates that the automatic alignment strategy imperfectly introduces errors for the data generation stage and can be further improved in the future work. For the Context-to-Question generator, 30.5 (averagely) out of 50 questions are in high quality based on the contexts and selected key entities, with 0.84 Cohen Kappa score. More alignment and generation examples are shown in Figure 4. First row shows a high-quality *(Question, Table, Context)* pair. For the second one, the generator makes mistakes with the subject, being confused "*Zhang*" with "*Jiangsu Suning*". For the third one, the error happens on the automatic alignment where the distance "*25*" in the context is matched with the number of village "*25*" in the table.

How does LongAug synthetic data size affect model performance? For LongAug, we analyze the effectiveness of the generation-based training data in terms of the scale. The Table 8 shows the performance of FeTaQA with different scales of pre-training corpus.

Context	Generated Question	Table
The final running of the Standard Stakes took place on June 9, 1908 and was won for the second straight time by owner James R. Keene.	In what year did Keene win for the second time?	Year | Winner | ... | Owner 1908 | Ballot | ... | James R. Keene 1907 | Peter Pan | ... | James R. Keene ... | ... | ... | ...
Zhang transferred to Chinese Super League side Jiangsu Suning on 28 February 2018.	What league did Jiangsu Suning join?	Club | Season | Division | ... Jiangsu Suning | 2018 | Chinese Super League | ... 2019 2020
… Kajen, which is located in the middle of the regency, about 25 km south of Pekalongan City.	About how many kilometers away from Pekalongan city is Kajen?	Name | Area in km^2 | ... | No. of vill. Kajen | 75.15 | ... | 25 ... | ... | ... | ...

Figure 4: Examples of our LongAug synthetic data. The Generated Questions were synthesized using our Context-to-Question method.

Model	BLEU
Random Token Masking	34.26
Key Entity Masking	34.85

Table 9: Training Task Exploration Results.

Training Task Design. In this section, we show the ablation study of the training targets. All results are shown in Table 1, 2, 3, and 4. The –ShortAug denotes the intermediate pre-training without ShortAug. The –LongAug & ShortAug denotes the baseline model without intermediate pre-training — note that this model does include the positional, segment, column and row embeddings[4]. Based on the automatic evaluation metrics, the LongAug improve the BLEU score from 33.87 to 36.07 by large margin. The ShortAug can further improve the metric to 36.74. The effectiveness of the LongAug and ShortAug is also shown from the BARTScore, Lexical level F1, Tuple Level F1 and the human evaluation.

Instead of using the generated questions as the text for the model input in our proposed GenTaP framework, we also explored design choices for pre-training: 1) Random Token Masking, and 2) Key Entity Masking. **Random Token Masking** is analogous to the Masked Language Model and we randomly mask the token in the context as the model input. We keep the table unchanged and use the original context as the learning target. We expect the model to capture the alignments between the context and table by learning to recover the incomplete context. **Key Entity Masking:** Instead of masking random tokens which may be unimportant, we try to mask the key entities. More specifically, based on the context-table alignment aforementioned, we masked the co-occurrent entities in the context, making it a proxy of natural questions. Again, we use the unmasked context as the training target. In this way, we can enforce the model to learn to capture more alignments between context and table by recovering the context, because all missing tokens come from the table content. We pre-train the models in the same way as the GENTAP with the *(context, table)* pairs. The experimental results are shown in Table 9. Based on the BLEU score results, we find that using the generated question as text input is a better choice than these two proposals, thus we did not use them in our main experiments.

[4]BART (fine-tuned by us) in Table 1 and 3 do not leverage segment, column and row embeddings.

Related Work

Table-based Pre-training. Recently, table-based pre-training received a lot of attention (Herzig et al. 2020; Eisenschlos, Krichene, and Müller 2020; Shi et al. 2020; Deng et al. 2020; Yin et al. 2020; Yu et al. 2020; Iida et al. 2021; Liu et al. 2021). Large scale crawled tables are used for pre-training to enhance the table representation ability of language models. Different from these work, we focus on the pre-training for free-form question answering, by leveraging the context-table alignments and question generation model.

Generation-based Question Answering. By leveraging the powerful sequence-to-sequence pre-trained language model, several question answering tasks are formulated as the generation problem (Lewis et al. 2019; Raffel et al. 2019; Shakeri et al. 2020; Min et al. 2020; Izacard and Grave 2021; Gao et al. 2021; Lewis et al. 2021). More recently, free-form question answering have been received increasing attention (Fan et al. 2019; Krishna, Roy, and Iyyer 2021; Nan et al. 2021) as it can handle more complex questions.

Data-to-Text. Data-to-Text generation requires the model to produce precise and fluent description given the structured data input, such as tables (Lebret, Grangier, and Auli 2016; Parikh et al. 2020), triples (Gardent et al. 2017; Novikova, Dušek, and Rieser 2017; Nan et al. 2020), or logic forms (Damonte and Cohen 2019; Xu et al. 2018; Shu et al. 2021). Recently, large scale pre-trained models are actively applied on these tasks, obtaining new state of the art (Ribeiro et al. 2020; Chen et al. 2020; Li et al. 2021).

Conclusion

In this work, we present an intermediate pre-training framework, GENTAP, that improves the joint encoding ability of question and table for pre-trained sequence-to-sequence language model. With two different augmentation strategies, LongAug and ShortAug, our models achieve the state-of-the-art performance on the free-form table-based question answering task. Also, the GENTAP models show good transfer ability to the few-shot data-to-text generation task, by outperforming existing models on FSD2T dataset in various domains.

References

Chen, W.; Su, Y.; Yan, X.; and Wang, W. Y. 2020. KGPT: Knowledge-Grounded Pre-Training for Data-to-Text Generation. *arXiv preprint arXiv:2010.02307*.

Chen, Z.; Eavani, H.; Chen, W.; Liu, Y.; and Wang, W. Y. 2019. Few-shot NLG with pre-trained language model. *arXiv preprint arXiv:1904.09521*.

Damonte, M.; and Cohen, S. B. 2019. Structural neural encoders for AMR-to-text generation. *arXiv preprint arXiv:1903.11410*.

Deng, X.; Sun, H.; Lees, A.; Wu, Y.; and Yu, C. 2020. Turl: Table understanding through representation learning. *arXiv preprint arXiv:2006.14806*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhingra, B.; Faruqui, M.; Parikh, A.; Chang, M.-W.; Das, D.; and Cohen, W. W. 2019. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*.

Eisenschlos, J. M.; Krichene, S.; and Müller, T. 2020. Understanding tables with intermediate pre-training. *arXiv preprint arXiv:2010.00571*.

Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; and Auli, M. 2019. ELI5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Gao, Y.; Zhu, H.; Ng, P.; Nogueira dos Santos, C.; Wang, Z.; Nan, F.; Zhang, D.; Nallapati, R.; Arnold, A. O.; and Xiang, B. 2021. Answering Ambiguous Questions through Generative Evidence Fusion and Round-Trip Prediction. 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)*, 3263–3276. Online: Association for Computational Linguistics.

Gardent, C.; Shimorina, A.; Narayan, S.; and Perez-Beltrachini, L. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Gong, H.; Sun, Y.; Feng, X.; Qin, B.; Bi, W.; Liu, X.; and Liu, T. 2020. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, 1978–1988.

Herzig, J.; Nowak, P. K.; Müller, T.; Piccinno, F.; and Eisenschlos, J. M. 2020. TaPas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Iida, H.; Thai, D.; Manjunatha, V.; and Iyyer, M. 2021. TABBIE: Pretrained Representations of Tabular Data. *arXiv preprint arXiv:2105.02584*.

Iyyer, M.; Yih, W.-t.; and Chang, M.-W. 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)*, 1821–1831.

Izacard, G.; and Grave, E. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. arXiv:2007.01282.

Kočiskỳ, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K. M.; Melis, G.; and Grefenstette, E. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6: 317–328.

Krishna, K.; Roy, A.; and Iyyer, M. 2021. Hurdles to Progress in Long-form Question Answering. *arXiv preprint arXiv:2103.06332*.

Lebret, R.; Grangier, D.; and Auli, M. 2016. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

Li, J.; Tang, T.; Zhao, W. X.; Wei, Z.; Yuan, N. J.; and Wen, J.-R. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. *arXiv preprint arXiv:2106.01623*.

Liu, Q.; Chen, B.; Guo, J.; Lin, Z.; and Lou, J.-g. 2021. TAPEX: Table Pre-training via Learning a Neural SQL Executor. *arXiv preprint arXiv:2107.07653*.

Min, S.; Michael, J.; Hajishirzi, H.; and Zettlemoyer, L. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5783–5797. Online: Association for Computational Linguistics.

Nan, L.; Hsieh, C.; Mao, Z.; Lin, X. V.; Verma, N.; Zhang, R.; Kryściński, W.; Schoelkopf, N.; Kong, R.; Tang, X.; et al. 2021. FeTaQA: Free-form Table Question Answering. *arXiv preprint arXiv:2104.00369*.

Nan, L.; Radev, D.; Zhang, R.; Rau, A.; Sivaprasad, A.; Hsieh, C.; Tang, X.; Vyas, A.; Verma, N.; Krishna, P.; et al. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

Novikova, J.; Dušek, O.; and Rieser, V. 2017. The E2E dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Parikh, A. P.; Wang, X.; Gehrmann, S.; Faruqui, M.; Dhingra, B.; Yang, D.; and Das, D. 2020. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*.

Pasupat, P.; and Liang, P. 2015. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ribeiro, L. F.; Schmitt, M.; Schütze, H.; and Gurevych, I. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

Shakeri, S.; Nogueira dos Santos, C.; Zhu, H.; Ng, P.; Nan, F.; Wang, Z.; Nallapati, R.; and Xiang, B. 2020. End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5445–5460. Online: Association for Computational Linguistics.

Shi, P.; Ng, P.; Wang, Z.; Zhu, H.; Li, A. H.; Wang, J.; Santos, C. N. d.; and Xiang, B. 2020. Learning contextual representations for semantic parsing with generation-augmented pre-training. *arXiv preprint arXiv:2012.10309*.

Shu, C.; Zhang, Y.; Dong, X.; Shi, P.; Yu, T.; and Zhang, R. 2021. Logic-Consistency Text Generation from Semantic Parses. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4414–4426. Online: Association for Computational Linguistics.

Xu, K.; Wu, L.; Wang, Z.; Feng, Y.; and Sheinin, V. 2018. Sql-to-text generation with graph-to-sequence model. *arXiv preprint arXiv:1809.05255*.

Yin, P.; Neubig, G.; Yih, W.-t.; and Riedel, S. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.

Yu, T.; Wu, C.-S.; Lin, X. V.; Wang, B.; Tan, Y. C.; Yang, X.; Radev, D.; Socher, R.; and Xiong, C. 2020. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. *arXiv preprint arXiv:2009.13845*.

Yuan, W.; Neubig, G.; and Liu, P. 2021. BARTScore: Evaluating Generated Text as Text Generation. *arXiv preprint arXiv:2106.11520*.

Zhong, V.; Xiong, C.; and Socher, R. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR*, abs/1709.00103.

Appendices

Data Preprocessing

We leverage several heuristics to collect the tables and the contexts pairs. More specifically, for each sentence in the same page of the table, if one of the conditions is satisfied, then it is a valid *(table, context)* pair. **1)** A sentence is valid if it has tokens matching at least 3 key entities from the same row of the table. **2)** A sentence is valid if it has tokens matching with 2 key entities from the same row of the table for more than two times (two different rows). We sampled 250K examples from the collection obtained from the above heuristics for our experiments.

Training Details

For intermediate pre-training, we use 8 Tesla V100 GPUs to train at most 100K steps with initial learning rate of 2e-5 and batch size of 64. For FeTaQA dataset finetuning, 4 Tesla V100 GPUs are used to train the model, with initial learning rate of 1e-5 and batch size of 32. For FSD2T dataset finetuning, 1 Tesla V100 GPU is used to train with initial learning rate of 1e-5 and batch size of 8.