FeTaQA: Free-form Table Question Answering

Linyong Nan[1] **Chiachun Hsieh**[3] **Ziming Mao**[1] **Xi Victoria Lin**[2*] **Neha Verma**[1]
Rui Zhang[4] **Wojciech Kryściński**[2] **Hailey Schoelkopf**[1] **Riley Kong**[5] **Xiangru Tang**[1]
Mutethia Mutuma[1] **Ben Rosand**[1] **Isabel Trindade**[1] **Renusree Bandaru**[4]
Jacob Cunningham[4] **Caiming Xiong**[2] **Dragomir Radev**[1,2]

[1] Yale University, USA [2] Salesforce Research, USA [3] The University of Hong Kong, China
[4] Penn State University, USA [5] Archbishop Mitty High School, USA
{linyong.nan, ziming.mao}@yale.edu, hsiehcc@connect.hku.hk

Abstract

Existing table question answering datasets contain abundant factual questions that primarily evaluate a QA system's comprehension of query and tabular data. However, restricted by their short-form answers, these datasets fail to include question–answer interactions that represent more advanced and naturally occurring information needs: questions that ask for reasoning and integration of information pieces retrieved from a structured knowledge source. To complement the existing datasets and to reveal the challenging nature of the table-based question answering task, we introduce FeTaQA, a new dataset with 10K Wikipedia-based {*table*, *question*, *free-form answer*, *supporting table cells*} pairs. FeTaQA is collected from noteworthy descriptions of Wikipedia tables that contain information people tend to seek; generation of these descriptions requires advanced processing that humans perform on a daily basis: Understand the question and table, retrieve, integrate, infer, and conduct text planning and surface realization to generate an answer. We provide two benchmark methods for the proposed task: a pipeline method based on semantic parsing-based QA systems and an end-to-end method based on large pretrained text generation models, and show that FeTaQA poses a challenge for both methods.

1 Introduction

Question Answering (QA) is the task of producing answers to natural language questions based on knowledge resources (Burke et al., 1997; Yao and Van Durme, 2014; Chen et al., 2017). One of the primary goals of QA is to allow users to directly and efficiently interact with large-scale and heterogeneous knowledge sources. In the real world, knowledge sources take a variety of forms, including unstructured texts (documents, passages, or conversations), structured knowledge bases, and semi-structured tables, each requiring dedicated modeling approaches.

For QA over text, a sequence modeling approach is usually adopted to encode the query and the context, and answers are either categorical (Lai et al., 2017), extractive (Rajpurkar et al., 2016; Yang et al., 2018), or abstractive/generative (Kociský et al., 2017; Nguyen et al., 2016; Fan et al., 2019; Kwiatkowski et al., 2019). For QA over table, a common approach is to apply semantic parsing on the query and the table schema to generate a logical form (e.g., a SQL-like database query) that can be executed to retrieve the answer from the relevant portion of the table (Pasupat and Liang, 2015; Iyyer et al., 2017; Zhong et al., 2017; Yu et al., 2018). The answers are extracted facts/entities in the table, therefore usually in short-form.

Though existing datasets have enabled significant progress for table QA, their limitations prevent them from reflecting the challenging nature of the task. The exchange of information between humans through interactions with questions and answers is different from the interactions presented in most of the existing QA datasets, in which questions are specific (sometimes contrived for testing multi-hop reasoning) and provide most of the information, while answers are in short-form and fill in the missing information piece. Nevertheless, in many cases, people tend to seek more structured information content, such as ''how'', ''why'', and some of the ''what'' questions that ask for general concepts. Therefore a QA system should also possess such structuring capability, evaluated by text generation tasks.

[*] Now at Facebook AI.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 35–49, 2022. https://doi.org/10.1162/tacl_a_00446
Action Editor: Radu Florian. Submission batch: 4/2021; Revision batch: 8/2021; Published 1/2022.

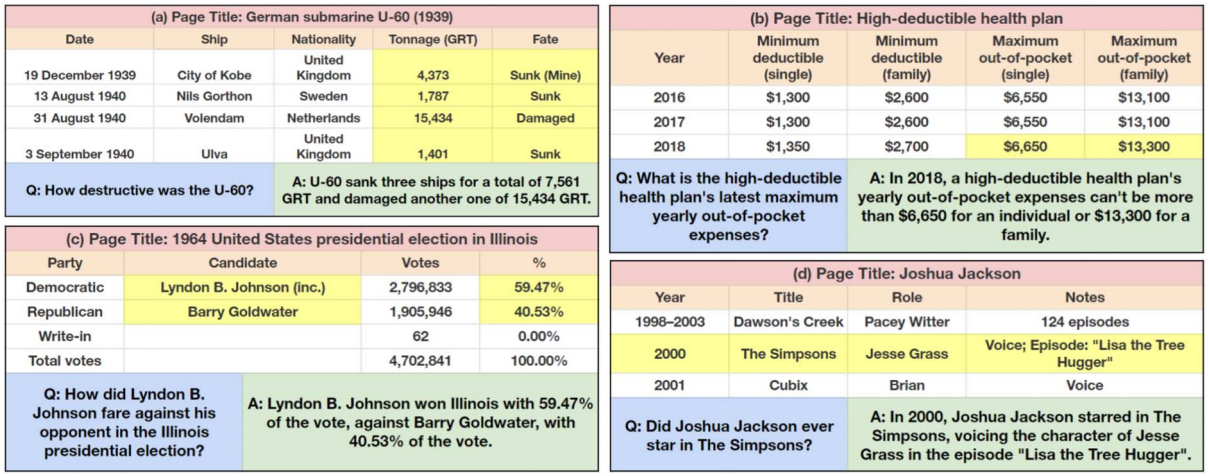

Figure 1: Examples of **FeTaQA** instances. Only part of the original table is shown for better visualization.

Dataset	Knowledge Source: Wikipedia articles	Stories, books, movie scripts	Online forum texts	Wikipedia tables	Answer Format	Avg # Words in Answer
SQuAD (Rajpurkar et al., 2016)	✓				Text-span	3.2
HotpotQA (Yang et al., 2018)	✓				Short-form entity	2.2
NarrativeQA (Kociský et al., 2017)		✓			Free-form text	4.7
ELI5 (Fan et al., 2019)			✓		Free-form text	130.6
WikiTableQuestions (Pasupat and Liang, 2015)				✓	Short-form entity	1.7
SequenceQA (Saha et al., 2018)				✓	Short-form entity	1.2
HybridQA (Chen et al., 2020d)	✓			✓	Short-form entity	2.1
FeTaQA				✓	Free-form text	18.9

Table 1: Comparison of **FeTaQA** with other QA datasets.

To complement the existing datasets with the absent QA interactions, we present FeTaQA, a **Fre**e-form **Ta**ble Question Answering dataset that includes long, informative, and free-form answers. FeTaQA reveals the challenging nature of the table QA task: 1) retrieving multiple entities from tables based on the query; 2) aggregating and reasoning over relations of these entities; and 3) structuring surface information and inferences into a coherent answer that is faithful to the table. We collect question–answer pairs from noteworthy descriptions of Wikipedia tables that are high quality sentences rich in structured information contents. We annotate questions that elicit such descriptions, and we make efforts to ensure that the QA interaction is compatible, and question annotations are not contrived. In addition, the FeTaQA tables cover a diverse set of topics and contain un-normalized text, including numbers, dates, and phrases. FeTaQA examples are presented in Figure 1 and differences between FeTaQA and other QA datasets are described in Table 1.

We formulate generative table question answering as a Sequence-to-Sequence learning problem. We propose two benchmark methods and provide experimental results for them. The first one is an end-to-end model that integrates query and table comprehension, reasoning, and language generation by adapting T5 (Raffel et al., 2020). The other is a pipeline model that achieves content selection and surface realization in separate modules involving TAPAS (Herzig et al., 2020), which is a recently proposed pre-trained model that jointly processes text and tabular data for the usage of semantic parsing.

Through human studies, we evaluate answers generated by our proposed models as well as the reference answer based on fluency, correctness, adequacy (informativeness), and faithfulness. The results indicate the challenging nature of FeTaQA and that there is much room for improvement in QA systems. We make the dataset and code available online.[1]

2 Dataset

Here we introduce FeTaQA and describe the process and criteria for collecting the tables, questions, and answers. Some statistics of FeTaQA are shown in § 2.4.

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

2.1 Desiderata

We frame generative table question answering as a problem of generating an answer a to a question q based on a table T and its metadata m. Our goal was to construct a table QA dataset $\{(q_i, a_i, T_i, m_i)|i = 1 \dots n\}$ with a large number of instances and diverse topics. We want to collect questions that seek not just a specific fact, but more structured information: Desirably, they should require retrieving more and different facts and reasoning with diverse aggregations. Answers should be well structured information contents, faithful to the tables, and presented in natural utterances.

2.2 Data Collection Method

A natural way to collect a table-based QA pair is to ask annotators to first generate a question given a table, then provide the answer to it. However, we found that it usually takes more effort to ask about how multiple facts are related or share something in common than to ask about a specific fact in the table; annotators spend much more time finding out the relations between cell contents for question generation, and they also need to spend time writing an answer. We found that ToTTo (Parikh et al., 2020), a recently proposed large-scale Table-to-Text dataset, is a desirable resource to start with. It contains textual descriptions that are naturally written and fully grounded in Wikipedia tables. Additionally, ToTTo comes with annotations of table cells that support the sentences: A sentence is supported by the cell contents if it is directly stated or can be logically inferred by them. ToTTo applied several heuristics to sample the tables and the candidate sentences from Wikipedia pages, and their annotators are asked to revise sentences and highlight the corresponding table regions so that the sentences still have the varied language and structure found in natural sentences.

We want to first sample a subset of these sentences that already provide aggregation and reasoning on multiple facts in the table, which is the target content that annotators spend most of the time trying to come up with, so that we could largely reduce the time spent on annotation. More importantly, such sentences contain noteworthy information that users are more interested in and likely to ask given a table from Wikipedia. We sample ToTTo instances with the following considerations. First we found that ToTTo's annotation of highlighted cells is a reasonable indicator of how much information is required from the table to give the answer, which we aim to maximize. With this objective, we found by probing ToTTo that tables with extreme sizes (too large or too small number of rows, columns or both) are more similar to attribute–value pairs instead of tables with complicated structures, and they tend to have a small number of highlighted cells, which make them not ideal for our dataset. As shown by Figure 9 and 10 in the Appendix, we removed all tables whose sizes are above the 75th percentile of the number of rows or columns of all ToTTo tables, and also removed tables with a single row or column. We further select tables whose highlighted cells span more than a single row or column to ensure sentences contain several table entities. We provide a flowchart of this sampling process in Figure 7 in the Appendix. This process gave us sufficient {*table, metadata, highlighted region, sentence*} instances from ToTTo, on which we conducted the annotation procedure as described below.

We adopted these table-grounded sentences as the answers in our new QA dataset and exploited ToTTo's annotations of table cells (the highlighted table region) as the weak supervision labels (denotations) for training and evaluating the intermediate semantic parser. We processed each table (originally in HTML format) as a 2-dimensional array, where the first row corresponds to the table header. We also processed merged cells by copying the cell content and cell highlighted region to all the individual cells that compose the original merged cell.

2.2.1 Question Annotation

Question annotations were collected with the help of human judges in two phases: an internal phase conducted by on-site expert annotators, and an external phase conducted by crowd workers on Amazon Mechanical Turk. To streamline the process, we built a custom Web interface to visualize table HTML and metadata, augmented with Web widgets that allow table region highlighting, table content and sentence editing. A screenshot of the annotation interface is shown in Figure 8 in the Appendix.

Provided the full context of ToTTo instances, the annotators were asked to write a question whose answer is the provided ToTTo sentence.

Highlighted Region	Cell Content	ToTTo Sentence	Percentage
✗	✗	✗	62.45%
✓	✗	✗	2.96%
✗	✓	✗	0.66%
✗	✗	✓	10.13%
✓	✗	✓	22.62%
✓	✓	✗	0.07%
✗	✓	✓	0.49%
✓	✓	✓	0.62%
	Total		100%

Table 2: Breakdown of modifications made by the annotators for generating more natural questions.

We found that such questions arise naturally when table cell contents are more semantically related. In addition, annotators were free to modify the sentence, the table cell content, and the highlighted region so that these contents could lead to a more natural question formulation and avoid any contrived effort. Table 2 provides measurements on how often annotators modified ToTTo resources for producing more compatible question-answer interactions.

Internal Annotations In the first phase of annotation, we enrolled 15 internal annotators who were provided with preliminary guidelines. In addition to the annotation task, they were asked to provide feedback regarding the task instructions and the user experience of the Web site, based on which we iteratively modified the guideline and the Web site design.

External Annotations For external annotations, we hired MTurk workers who have completed at least 500 HITs, have 97% approval rate, and are from English-speaking regions. To ensure that the MTurk annotators understand our task, we provided an instruction video for the interactive annotation tool usage, FAQs that clarify the annotations we desire, along with good vs. bad annotation examples. We also created a Slack channel for crowdsourced workers to ask questions and clarify doubts.

Annotation Evaluation To ensure that FeTaQA is of high quality, we evaluate crowdsourced annotations as follows. We built another Web interface for evaluation and asked internal evaluators to approve (with modification if necessary) based on grammatical correctness, relevancy to the highlighted table cells, and its compatibility with the answer. Evaluators modified question annotations if they are asking for only one of many facts in the answer sentence, or if a short-form answer is clearly adequate, as we discovered that most of the modifications that evaluators made are in this category. We reject when we couldn't modify the annotation to meet the above standards within a reasonable time frame. The breakdown of the evaluation result is shown in Table 3. We approved most of the annotations and rejected only 12%, for which we found the original ToTTo instances are hard to generate questions for. We found that these instances usually contain highlighted cells that do not have any clear relation, therefore making it difficult to come up with questions. Among the annotations we approved, only 16.7% of the original annotations were modified, so that the crowd-sourced annotations are not much affected by the internal evaluators' bias if there exist any.

Decision Type	Percentage
Reject	12.00%
Approve - *no modification*	73.30%
Approve - *only modify question*	7.66%
Approve - *only modify HR*	1.71%
Approve - *modify question and HR*	5.19%
Approve - *other modification*	0.14%
Total	100%

Table 3: MTurk annotation evaluation result breakdown. HR stands for highlighted region.

The annotator contributions to the final dataset are distributed as follows: We have 3,039 (30%) instances from internal annotators and 7,291 (70%) from MTurk workers. In total, our dataset contains 10,330 instances.

2.3 Dataset Split

Randomly splitting the dataset may make train, development, and test splits contain tables with similar contents (Finegan-Dollak et al., 2018; Lewis et al., 2021). Therefore, to increase the generalization challenge, we split FeTaQA to minimize the content/topic overlap (not necessarily question/answer type overlap) between train set and dev-test set, similar to ToTTo (Parikh et al., 2020). We calculate the Jaccard similarity of tokens shown in the question and the table column

Property	Value
Unique Tables	10,330
Question Length (Median/Avg)	12 / 13.2
Answer Length (Median/Avg)	18 / 18.9
Rows per Table (Median/Avg)	12 / 13.8
Columns per Table (Median/Avg)	5 / 5.9
No. of Highlighted Cell (Median/Avg)	6 / 8.0
Percentage of Cells Highlighted (Median/Avg)	10.7% / 16.2%
Page Title Length (Median/Avg)	2 / 3.3
Section Title Length (Median/Avg)	2 / 1.9
Training Set Size	7,326
Development Set Size	1,001
Test Set Size	2,003

Table 4: FeTaQA core statistics.

Annotation Quality	Score >= 4 (%)	% Agreement	Randolph's Kappa / 95% CI
Question Complexity	52.6	0.65	0.48 / [0.41, 0.55]
Denotation Correctness	89.0	0.88	0.82 / [0.76, 0.88]
Denotation Adequacy	91.6	0.89	0.83 / [0.77, 0.89]
Answer Fluency	95.0	0.92	0.89 / [0.84, 0.94]
Answer Correctness	92.4	0.91	0.86 / [0.80, 0.92]
Answer Adequacy	90.6	0.88	0.82 / [0.76, 0.88]
Answer Faithfulness	95.6	0.93	0.89 / [0.84, 0.94]

Table 5: Human evaluation over 100 samples of FeTaQA. Five internal evaluators are asked to rate the samples on a scale of 1 to 5. We report % of samples that have score ≥ 4 to show high quality of FeTaQA, and report percent agreement and Randolph's Kappa (Randolph, 2010) (with 95% CI) to show that our human evaluation has high inter-annotator agreement.

headers of two instances to measure their similarity. We first sampled 800 instances randomly as a seed set, then gradually add instances to it if an instance is similar to any instance in the seed set. When this seed set grows to take up 70% of all the instances, the remaining 30% instances are less similar to any instance in the seed set. The seed set then becomes the training set and the remaining instances are divided to form the development and test sets. This results in 7,326/1,001/2,003 instances in the train/dev/test splits, respectively.

2.4 Data Analysis and Statistics

Basic statistics of FeTaQA are shown in Table 4. We also conducted a human evaluation over 100 FeTaQA instances in 7 dimensions. Evaluation scores and inter-evaluator agreements are reported in Table 5. A quantitative and qualitative analysis of FeTaQA shows it contains lots of complex questions judged by human evaluators. Note that an ideal measurement of the question complexity is to quantify the structural complexity of the information contained in the answer, but since

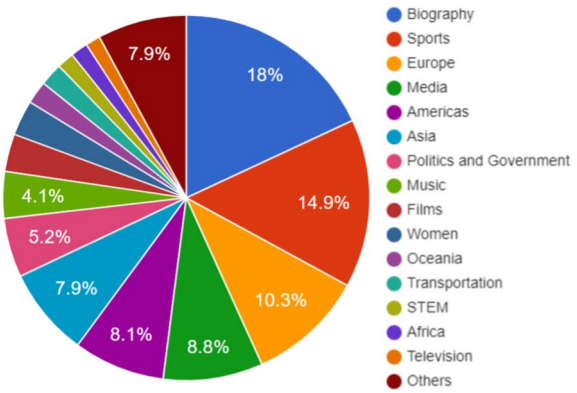

Figure 2: FeTaQA topics distribution.

this is a time-consuming process, we simply asked the evaluators to score based on their subjective judgement, which could have caused the relatively low agreement. The median number of highlighted cells (denotations) is 6, which is twice as much as the corresponding number for ToTTo, indicating that FeTaQA requires retrieval of multiple entities in the table. These denotations are correct and adequate as indicated by the corresponding high evaluation scores. The free-form answers have a median of 18 tokens in length, and are grounded to the table and the denotations, also suggested by the high evaluation scores.

Topics Similar to ToTTo, we use Wikimedia Foundation's topic categorization model (Asthana and Halfaker, 2018) to investigate the topic distribution of FeTaQA, as shown in Figure 2. We found that most of the instances are related to biography, sports, and geographical regions. There are also abundant instances related to media, politics, and government.

Question Types FeTaQA has diverse and complex questions, as illustrated in Figure 3. We found that in FeTaQA, a large percentage of *what* questions ask about entities in plural, or about abstract entities such as *outcome, result, margin, percentage*. In addition, there is a higher percentage of *how* questions that are not *how many/much*, compared to existing table QA datasets.

3 Models

To quantify the challenge posed by FeTaQA for state-of-the-art models, we used two modeling approaches that have shown to be effective for the

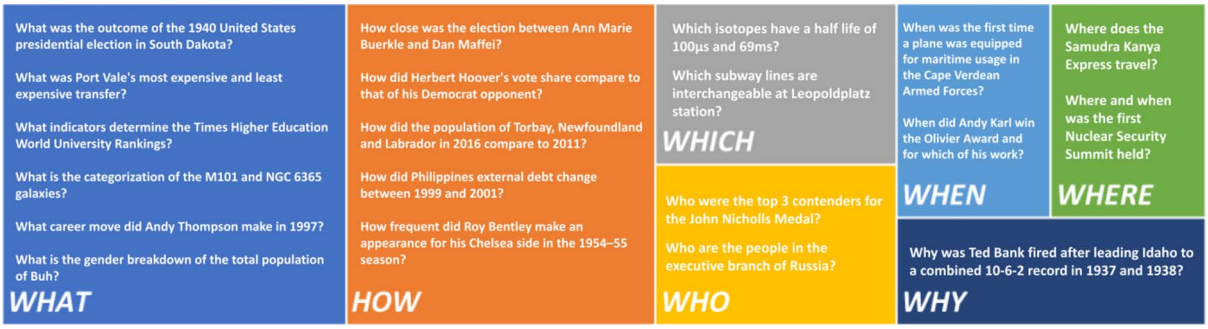

Figure 3: FeTaQA questions by most frequent starting words.

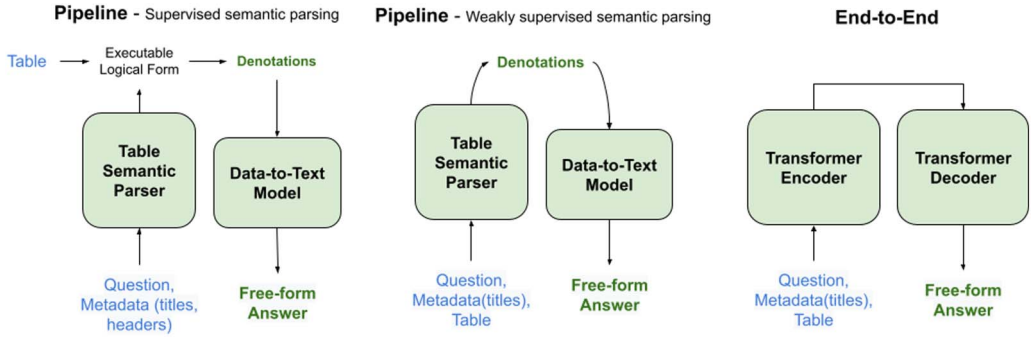

Figure 4: Pipeline model and End-to-End model diagrams.

existing table question answering datasets, with some modifications made to adjust to our task. Model configurations are shown in Figure 4.

3.1 Pipeline Model

Question answering over tables is usually seen as a semantic parsing task. A table semantic parser obtains representations of the question and the table schema, and uses these to generate database-like queries. These generated queries then get executed to give the final denotation(s), which are sufficient for answering the questions in the previous datasets. There are two possible settings for training or fine-tuning a table semantic parser, as shown by the two diagrams on the left in Figure 4. The first one is the supervised learning setting, which requires annotations of database-like queries. But due to their high annotation costs, people usually train semantic parsers with the latter: a weakly supervised setting, which requires label denotations, and semantic parsers learn to predict which table cells constitute the final answer (Note that we use ToTTo's highlighted table cells as these labels).

However, in our task, targets are generated texts instead of retrieved denotations, suggesting that we also need a generator to integrate the retrieved information into a cogent sentence. Therefore, we propose a pipeline model with two separately trained modules, described below.

Weakly Supervised Table Semantic Parsing The first module adopts a weakly supervised table semantic parser. Two recently proposed pre-trained models could help achieve this: TAPAS (Herzig et al., 2020) and TaBERT (Yin et al., 2020a). They are both pre-trained models for joint understanding of text and tabular data, and can be integrated into semantic parsers for solving table-based QA tasks. However, we did not include TaBERT in our experiment because it provides table column representations based on no more than 3 rows of the table, which are selected based on their *n*-gram overlap with the question. These representations are designed to help weakly supervised semantic parsers generate better database-like queries, therefore this method also depends on a reasonably designed domain-specific query language, as shown by TaBERT's use case of MAPO (Liang et al., 2018). In contrast, TAPAS provides representations for all table cells that help weakly supervised semantic parsers directly predict denotations in an end-to-end fashion, so it's easier to perform analysis for our pipeline models without considering any propagating error.

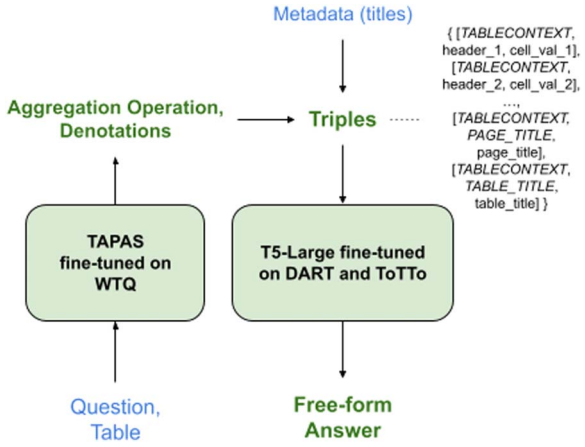

Figure 5: Weakly supervised fine-tuning of table semantic parser on FeTaQA. We choose a checkpoint of TAPAS-base fine-tuned on WikiTableQuestions to start with. After fine-tuning, the table semantic parser predicts denotations, which are then converted to triples and sent to the Data-to-Text module.

We fine-tune TAPAS with FeTaQA's label denotations (highlighted table regions). We believe fine-tuning is crucial for our task because TAPAS is pre-trained on questions that require retrieval of limited denotations (single entity or homogeneous entities that can be aggregated with `COUNT`, `SUM`, or `AVG` operation), while FeTaQA questions require retrieval of multiple entities and complex aggregations. Details of experiment results are provided in § 4.3. Note that besides denotations, TAPAS was pre-trained to explicitly predict an aggregation operation (choose from `COUNT`, `SUM`, `AVG`, `NONE`) applied to the predicted denotations to obtain the final answer. However, we argue that the aggregations required to solve FeTaQA instances are diverse and they are not covered by a small list of atomic operations pre-defined by humans. Instead, we use `NONE` as the aggregation operation label for fine-tuning TAPAS, and let the second module (described next) produce latent aggregations inferred from the question and the denotation predictions for generating the answer sentence.

Data-to-Text As shown in Figure 5, we fine-tune T5 (Raffel et al., 2020) on DART (Nan et al., 2021) to obtain a Data-to-Text model as the second module of the pipeline to perform inference of aggregation and surface realization of table cells (denotations in our case). We first convert the denotation prediction into the triple-set format with the following scheme: for each table cell in the highlighted region, we generate the following triple: `[[TABLECONTEXT], column_header, cell_value]`, where `column_header` is the cell's corresponding column name. Similar to DART, we use `[TABLECONTEXT]` as a special token for converting a table cell into a triple. We then incorporate the metadata into triples by replacing `column_header` with the field name (`TABLE_TITLE`, `PAGE_TITLE`) and `cell_value` with the metadata content (table title text, page title text). We end up with a triple-set containing all highlighted table cells and the metadata (table title and title of the Wikipedia page that includes the table). We further fine-tune the Data-to-Text model on ToTTo instances so that it adapts to our formation of triple-set inputs. To avoid exposure to FeTaQA test instances, we fine-tune with a sample of 8K ToTTo instances that are not used for creating FeTaQA.

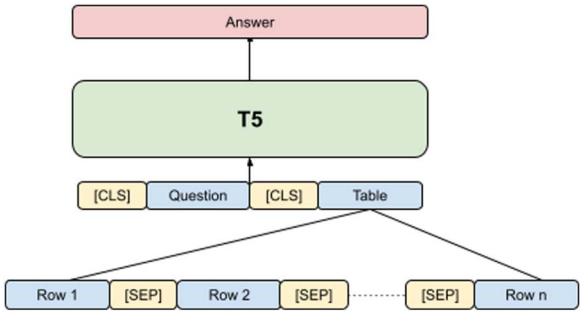

Figure 6: Table linearization in end-to-end model.

3.2 End-to-End Model

In this approach, we model the task as a sequence-to-sequence learning problem by linearizing table T appended to question q as the source sequence, and treating the free-form answer a as the target sequence. We propose a simple linearization scheme as a baseline: table rows are concatenated with `[SEP]` tokens in between, and cells in each row are separated by spaces. We prepend q to table linearization $\widetilde{T}$, and use `[CLS]` tokens as prefixes for separation. We fine-tune models from the T5-family on the FeTaQA train set. The linearization scheme is visualized in Figure 6. We considered an alternative option of integrating TaBERT into an end-to-end model but found it infeasible, since it provides contextual features for the question and table columns (instead of table cells, as in our table linearization). The decoder that generates the free-form answer does not have access to any

	sacreBLEU[2]	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore	BLEURT
Pipeline - zeroshot	9.16	0.38	0.20	0.33	0.22	0.88	−0.79
Pipeline - fine-tuned	11.00	0.40	0.22	0.35	0.24	0.91	−0.35
Pipeline - gold denotation	31.63	0.67	0.43	0.53	0.50	0.91	−0.23
End-to-End - T5-small	21.60	0.55	0.33	0.47	0.40	0.94	0.08
End-to-End - T5-base	28.14	0.61	0.39	0.51	0.47	0.96	0.31
End-to-End - T5-large	**30.54**	**0.63**	**0.41**	**0.53**	**0.49**	**0.96**	**0.57**

Table 6: Experiment results on the test split of FeTaQA.

table cell content. Therefore we did not include TaBERT as a baseline end-to-end model.

4 Experiments

In this section, we explain the experiment settings and report the automatic and human evaluations on model outputs.

4.1 Experiment Setup

We first experiment with the pipeline model in a zero-shot setting, that is, without any fine-tuning on FeTaQA. We use a checkpoint of TAPAS-base that is fine-tuned on WikiTableQuestions (Pasupat and Liang, 2015) to perform table semantic parsing implicitly in order to produce a set of denotations, which is then converted to a triple-set as described in § 3.1. We then employ a T5-large model (Raffel et al., 2020) that goes through two fine-tuning stages: in the first stage it is fine-tuned on the downstream Data-to-Text task with DART (Nan et al., 2021); in the second stage it is further fine-tuned on ToTTo instances to adapt to the triple-set formulation we proposed. We denote this setting as `Pipeline - zeroshot` in Table 6. Next we experiment with the pipeline model by fine-tuning the table semantic parser on FeTaQA. We further fine-tune the TAPAS-base checkpoint (WTQ fine-tuned) on FeTaQA train set and select models based on their performance on the development set. We use the same Data-to-Text model as described in the zero-shot setting.

For the End-to-End model, we adapt Hugging Face's implementation (Wolf et al., 2020) of T5 (Raffel et al., 2020) for our task. We use a standard T5-tokenizer with additional `[CLS]` and `[SEP]` tokens and the model vocabulary is resized accordingly. Since we expect the input sequence to be significantly longer than the target, we fine-tuned the models using T5's ''summarize:'' prefix. The motivation behind this is to avoid simple extraction from the table since abstractive summarization is supposed to rephrase important details in the source. T5-small is trained on 4 Tesla K80 GPUs with per-device batch size of 16 for 30 epochs (about 6,900 steps) which took less than an hour. T5-base is trained on 4 Tesla K80 with per-device batch size of 4 (due to GPU memory constraints) for 80 epochs (about 36,640 steps) and it took around 3 hours. As for T5-large, we distributed the layers across 8 Tesla K80 to train with a batch size of 4 for 80 epochs (about 80k steps) and it took 5 hours to train.

[2]SacreBLEU signature: BLEU+case.lc+numrefs.1+smooth.exp+tok.13a+version.1.3.7.

4.2 Automatic Evaluation Metrics

We use a variety of automatic metrics and human evaluation (§ 4.4) to evaluate the quality of the generated answers. We report sacreBLEU (Post, 2018), ROUGE-{1, 2, L} (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) that evaluate the *n*-gram match between generated and reference answers. Considering the limitations of these measures in evaluating the semantic meanings of sentences, we also report BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020) that incorporate semantics using contextual embeddings. To evaluate the retrieval competency of table semantic parsers, we applied various set similarity metrics to the predicted and reference denotation lists. Specifically, we report Jaccard similarity, Overlap, Cosine similarity, and Dice similarity.

4.3 Results and Discussions

Our experimental results on the FeTaQA test set are summarized in Table 6. The T5-large model using an End-to-End modeling approach achieves the highest performance scores in all evaluation metrics. Also, we observe a large performance gap

	Jaccard	Overlap Coff.	Cosine	Dice
Zeroshot	0.065	0.300	0.140	0.109
Fine-tuned	0.101	0.311	0.184	0.161

Table 7: Evaluation of denotation prediction on the test split of FeTaQA. We report performance of TAPAS in zero-shot and fine-tuned with weak supervision.

Source	Fluent (%)	Correct (%)	Adequate (%)	Faithful (%)
Pipeline	85.2	25.4	8.4	23.6
End-to-End	94.6	54.8	48.4	50.4
Reference	95.0	92.4	90.6	95.6

Table 8: Human evaluation over 100 samples of model outputs and references. We report the percentage of outputs that have scores of 4 or 5.

between pipeline models and End-to-End models, even though the latter only adopt a simple linearization strategy for encoding tables.

We also see that after fine-tuning on FeTaQA with denotations as weak supervisions, the pipeline model improves by almost 2 BLEU points. To further examine the source of this improvement, we report the evaluation of table semantic parser performance in Table 7, from which we also observe an improvement in retrieval capability. However, we note that compared with the gold denotations that have a median of six table cells being highlighted (shown in 4), our table semantic parser is only able to predict two table cells on average before fine-tuning on FeTaQA, and three table cells on average after. When gold annotations are used, the pipeline model is able to perform better than the End-to-End model. This indicates that the low performance of denotation predictions and the loss of relational information between denotations lead to the inadequate performance of pipeline models, and it also indicates that the table semantic parser has a large space for improvement. A final observation is that the End-to-End model is comparable to the model that has access to the gold denotations, suggesting that the End-to-End model is effective at extracting denotations latently.

4.4 Human Evaluation

To further evaluate the quality of the answers generated by different models comparing to the references, we conduct our human evaluation based on four criteria: (1) *fluency* if an answer is natural and grammatical; (2) *correctness* if an answer is correct; (3) *adequacy* if an answer contains all the information that is asked; (4) *faithfulness* if an answer is faithful and grounded to the contents of the table and the highlighted region. Each evaluator is asked to examine an answer given the question and the full context (table, highlighted region, and metadata) and give a score on a scale of 1 to 5 for each of the criteria. We ask five internal annotators to evaluate 100 samples of FeTaQA instances. Each sample is paired with 3 answers: the reference, the pipeline model result, and the End-to-End model result.

Table 8 attests to the high quality of our annotations and the challenging nature of FeTaQA. Similar to the evaluation result of the automatic metrics, we observe a large gap between the pipeline model and the End-to-End model, with the latter one significantly outperforming its counterpart in terms of answer correctness, adequacy, and faithfulness. Comparing the best performing End-to-End model outputs to human references, we see that there is room for improvement in the future.

5 Related Work

Generative QA Generative question answering datasets such as NarrativeQA (Kociský et al., 2017), CoQA (Reddy et al., 2019), TriviaQA (Joshi et al., 2017), and MS MARCO (Nguyen et al., 2016) all have free-form answers that are generated based on the contexts of Wikipedia articles, books, movie scripts, dialogues, or Web documents. These responses are mostly crowdsourced and are reported to mostly contain copies of short text spans from the source. By contrast, ELI5 (Fan et al., 2019) is a long form question answering dataset containing a diverse set of complex questions, each paired with a paragraph-long answer and 100 relevant *Web source* documents (Petroni et al., 2021; Krishna et al., 2021). FeTaQA is the first dataset for generative question answering over tables. Unlike the existing generative QA datasets that assess multi-documents retrieval and abstraction capability, FeTaQA poses new challenges in the reasoning and integration capability of a system given a structured knowledge source.

QA over Tables and Semantic Parsing Several datasets have been proposed to apply semantic

parsing on tables, including WikiTableQuestions (Pasupat and Liang, 2015), SequentialQA (Iyyer et al., 2017), WikiSQL (Zhong et al., 2017), and Spider (Yu et al., 2018). With the development of pre-trained language models, recent work (Yin et al., 2020b, Herzig et al., 2020; Eisenschlos et al., 2020; Iida et al., 2021) jointly learns representations for natural language sentences and structured tables, and Yu et al. (2021a,b) use pre-training approach for table semantic parsing. HybridQA (Chen et al., 2020d) and OTT-QA (Chen et al., 2021) have contexts of both structured tables and unstructured text. MultiModalQA (Talmor et al., 2021) contains complex questions over text, tables and images. These datasets define a table QA task that is extractive in nature by restricting their answers to be short-form, while FeTaQA frames table QA as a generation task.

Data-to-Text Generation Recent neural end-to-end models tested on the WebNLG 2017 dataset (Gardent et al., 2017) have focused on incorporating pre-training and fine-tuning for specific generation tasks (Chen et al., 2020b; Kale and Rastogi, 2020) to improve performance and strengthen generalization ability. However, recent models featuring separate content-planning and surface realization stages have exhibited improvements (Moryossef et al., 2019; Iso et al., 2020) over comparable baselines. TabFact (Chen et al., 2020c) is composed of Wikipedia tables coupled with statements labeled as either ''ENTAILED'' or ''REFUTED'' by the table. LogicNLG (Chen et al., 2020a) features statements logically entailed from tables. ToTTo (Parikh et al., 2020) is a large-scale open-domain dataset consisting of Wikipedia tables with a set of highlighted table cells and a sentence description of those highlighted cells. DART (Nan et al., 2021) is an open-domain Data-to-Text dataset that contains table-ontology-preserving data samples with a diverse predicate set occurring in Wikipedia tables.

6 Conclusion

In this paper, we introduced the task of generative table question answering with FeTaQA, a table QA dataset consisting of complex questions that require free-form, elaborate answers. We also proposed two modeling approaches: (1) a pipeline model that incorporates a table semantic parser and a Data-to-Text generator, and (2) an End-to-End model that integrates query comprehension, reasoning and text generation. Our experimental results indicate that the End-to-End model with a simple table encoding strategy achieves much higher scores than the pipeline model that requires table semantic parsing. Furthermore, we show that FeTaQA reveals the challenging nature of the table question answering task and calls for innovative model designs in the future.

Acknowledgments

The authors would like to thank the anonymous reviewers and the Action Editor for their valuable discussions and feedback.

References

Sumit Asthana and Aaron Halfaker. 2018. With few eyes, all hoaxes are deep. In *Proceedings of the **ACM** on Human Computer Interaction* 2(CSCW). https://doi.org/10.1145/3274290

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Robin D. Burke, Kristian J. Hammond, Vladimir Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57–57.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1171

Wenhu Chen, Ming-wei Chang, Eva Schlinger, William Wang, and William Cohen. 2021. Open question answering over tables and text. In *Proceedings of ICLR 2021*.

Wenhu Chen, Jianshu Chen, Y. Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *ACL*. https://doi.org/10.18653/v1/2020.acl-main.708

Wenhu Chen, Yu Su, X. Yan, and W. Wang. 2020b. Kgpt: Knowledge-grounded pre-training for data-to-text generation. In *EMNLP*. https://doi.org/10.18653/v1/2020.emnlp-main.697

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

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020d. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *Findings of EMNLP 2020*. https://doi.org/10.18653/v1/2020.findings-emnlp.91

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. https://doi.org/10.18653/v1/2020.findings-emnlp.27

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *ACL 2018*. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1033

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-3518

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.398

Hiroshi Iida, June Thai, Varun Manjunatha, and Mohit Iyyer. 2021. Tabbie: Pretrained representations of tabular data. In *NAACL*. https://doi.org/10.18653/v1/2021.naacl-main.270

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2020. Learning to select, track, and generate for data-to-text. *Journal of Natural Language Processing*, 27(3):599–626. https://doi.org/10.5715/jnlp.27.599

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. https://doi.org/10.18653/v1/P17-1167

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1147

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor

Melis, and Edward Grefenstette. 2017. The narrativeqa reading comprehension challenge. *CoRR*, abs/1712.07040.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *NAACL*. https://doi.org/10.18653/v1/2021.naacl-main.393

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association of Computational Linguistics*. https://doi.org/10.1162/tacl_a_00276

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1082

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.86

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V. Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10015–10027. Curran Associates, Inc.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. 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 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Murori Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *NAACL*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

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. https://doi.org/10.18653/v1/2020.emnlp-main.89

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. https://doi.org/10.3115/v1/P15-1142

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao,

James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: A benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.200

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-6319

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1264

Justus Randolph. 2010. Free-marginal multirater kappa (multirater kfree): An alternative to fleiss fixed-marginal multirater kappa. *Advances in Data Analysis and Classification*, 4.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266. https://doi.org/10.1162/tacl_a_00266

Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *AAAI 2018*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.704

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. MultimodalQA: complex question answering over text, tables and images. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.18653/v1/D18-1259

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with Freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland. Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1090

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020a. TaBERT: Pretraining for joint understanding of textual

and tabular data. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020b. 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.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021a. Grappa: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*.

Tao Yu, Rui Zhang, Oleksandr Polozov, Christopher Meek, and Ahmed Hassan Awadallah. 2021b. Score: Pre-training for context representation in conversational semantic parsing. In *ICLR*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1425

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

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

A Appendix

The Appendix contains the following contents:

- ToTTo instance sampling process. (Figure 7)
- FeTaQA annotation interface. (Figure 8)
- Distribution plots for number of rows and columns of ToTTo tables. (Figure 9 and 10)

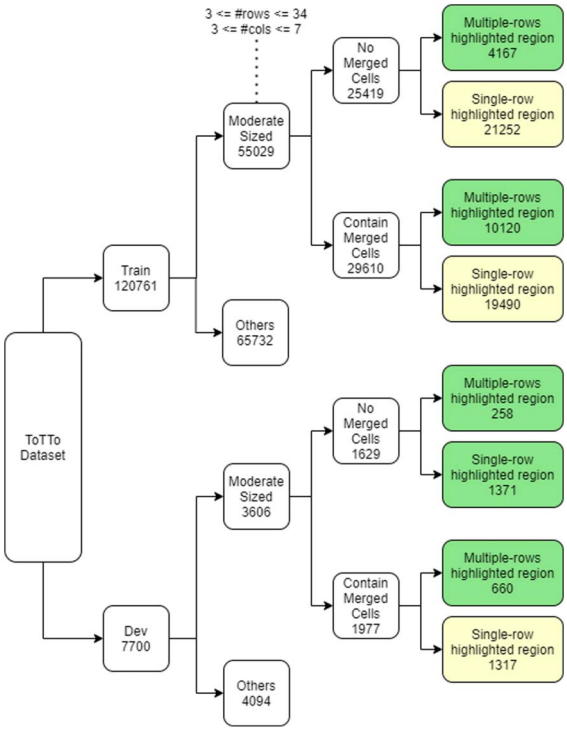

Figure 7: Flowchart of ToTTo sampling process.

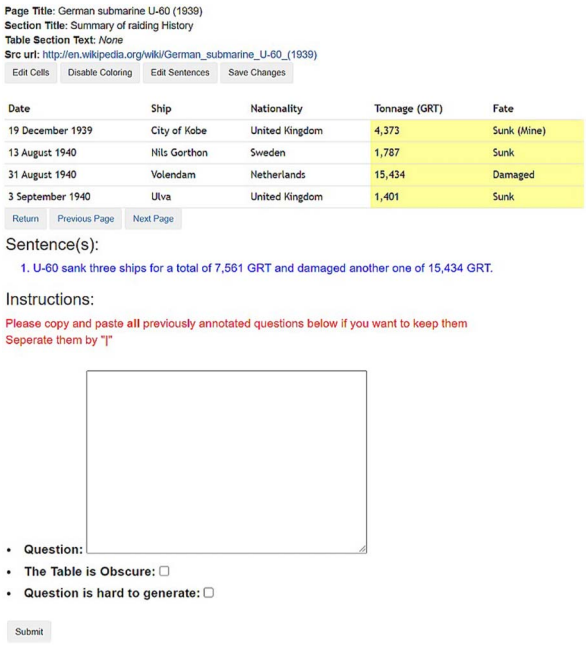

Figure 8: FeTaQA annotation interface.

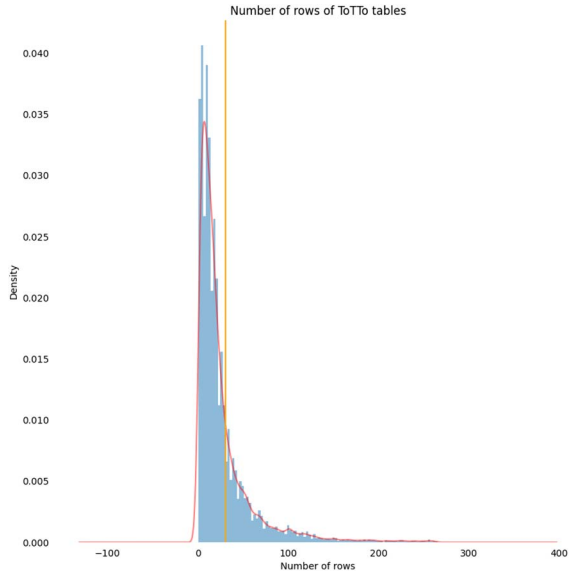

Figure 9: Number of rows distribution of ToTTo tables. Orange line indicates the 75th percentile. Outliers (3 standard deviations away) are removed for better visualization.

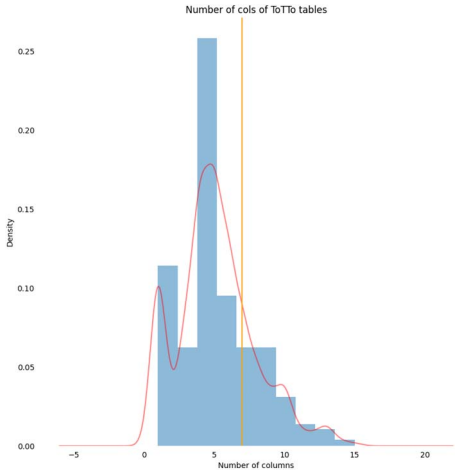

Figure 10: Number of columns distribution of ToTTo tables. Orange line indicates the 75th percentile. Outliers (3 standard deviations away) are removed for better visualization.