# RobuT: A Systematic Study of Table QA Robustness Against Human-Annotated Adversarial Perturbations

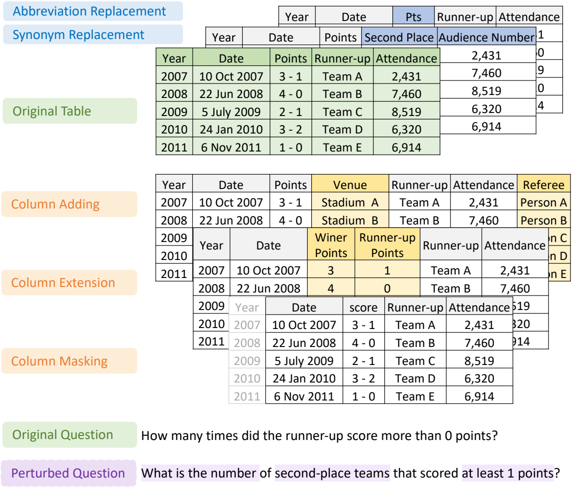
**Yilun Zhao**[1] **Chen Zhao**[2] **Linyong Nan**[1] **Zhenting Qi**[3] **Wenlin Zhang**[3]
**Boyu Mi**[3] **Xiangru Tang**[1] **Dragomir Radev**[1]
[1]Yale University [2] New York University [3] Zhejiang University
yilun.zhao@yale.edu cz1285@nyu.edu

## Abstract

Despite significant progress having been made in question answering on tabular data (Table QA), it's unclear whether, and to what extent existing Table QA models are robust to task-specific perturbations, e.g., replacing key question entities or shuffling table columns. To systematically study the robustness of Table QA models, we propose a benchmark called RobuT, which builds upon existing Table QA datasets (WTQ, WikiSQL-Weak, and SQA) and includes human-annotated adversarial perturbations in terms of table header, table content, and question. Our results indicate that both state-of-the-art Table QA models and large language models (e.g., GPT-3) with few-shot learning falter in these adversarial sets. We propose to address this problem by using large language models to generate adversarial examples to enhance training, which significantly improves the robustness of Table QA models. Our data and code is publicly available at `https://github.com/yilunzhao/RobuT`.

## 1 Introduction

Table QA uses structured table as world knowledge to answer questions. In recent years, Transformer-based models (Yin et al., 2020; Herzig et al., 2020; Yang et al., 2022; Jiang et al., 2022; Liu et al., 2022; Scao et al., 2022) achieve remarkable results on existing Table QA benchmark datasets (Pasupat and Liang, 2015; Zhong et al., 2017; Iyyer et al., 2017). Despite significant progress, state-of-the-art models are only evaluated within the same distribution, which does not provide insight into the model's robustness against out-of-domain distribution or adversarial data (Suhr et al., 2020), and recent studies (Cho et al., 2018; Zhu et al., 2020; Yang et al., 2022) revealed that existing models are vulnerable to adversarial perturbations. For example, Cho et al. (2018) observed significant performance degradation after a sentence-level question perturbation. Yang et al. (2022) showed that state-of-the-art Table QA models exhibited a dramatic performance drop after randomly shuffling the row or column order of the input table. However, previous works primarily focus on a single type of adversarial perturbation and rely on rule-based perturbation methods that are limited in linguistic richness. We fill this gap through a comprehensive evaluation of Table QA model robustness.



| Year | Date | Points | Runner-up | Attendance |
|---|---|---|---|---|
| 2007 | 10 Oct 2007 | 3 - 1 | Team A | 2,431 |
| 2008 | 22 Jun 2008 | 4 - 0 | Team B | 7,460 |
| 2009 | 5 July 2009 | 2 - 1 | Team C | 8,519 |
| 2010 | 24 Jan 2010 | 3 - 2 | Team D | 6,320 |
| 2011 | 6 Nov 2011 | 1 - 0 | Team E | 6,914 |

How many times did the runner-up score more than 0 points?

What is the number of second-place teams that scored at least 1 points?

Figure 1: Examples of adversarial perturbation over table header (blue), table content (orange), and question (purple). Table QA model predicts a correct answer on the original example but fails on perturbed ones.

In this paper, we constructed a new benchmark, **RobuT**, to systematically evaluate the **Robu**stness of **T**able QA models (Figure 1). RobuT was built upon the development set of WTQ (Pasupat and Liang, 2015), WikiSQL-Weak (Zhong et al., 2017), and SQA (Iyyer et al., 2017) datasets. Specifically, we designed 10 types of adversarial perturbations at three different levels (i.e., table header, table content, and natural language question), with a total number of 138,149 *human-annotated* perturbed examples.

We evaluated state-of-the-art Table QA models (Herzig et al., 2020; Chen et al., 2021; Liu et al., 2022; Yang et al., 2022; Jiang et al., 2022; Chen,

*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 6064–6081
July 9-14, 2023 ©2023 Association for Computational Linguistics

2022) and few-shot learning with large language models (LLMs) on ROBUT. The experiments revealed that all models significantly degrade performance in our adversarial sets, while large LLMs, such as GPT-3 (Brown et al., 2020; Wei et al., 2022b) and CodeX (Chen et al., 2021), are more robust. For example, GPT-3 outperforms all other Table QA models on both word-level and sentence-level question perturbations.

Motivated by the findings that LLMs are more robust against human-annotated adversarial perturbations, we developed **LETA**, a **L**LM-**E**nhanced **T**able QA **A**ugmentation framework that uses LLMs to generate adversarial examples to enhance model training. Specifically, we prompted GPT-3 or CodeX to simulate human annotation and generate adversarial training examples for all perturbation types. Experimental results showed that fine-tuning on these adversarial training examples significantly improves model robustness.

We summarize three major contributions:

- We constructed ROBUT, the first diagnostic evaluation benchmark for Table QA robustness. We applied rigid annotation quality control procedure to ensure the comprehensiveness, linguistic richness, and semantic association of the benchmark.

- Experimental results showed that state-of-the-art models exhibited significant performance drops on ROBUT benchmark, thus there is still large room to explore for Table QA tasks beyond high leaderboard scores.

- We designed LETA, an adversarial training example generation framework using LLM prompting methods. Experiments demonstrated that our methods effectively improves Table QA model robustness.

## 2 Related Work

**Table QA** Question answering over tables has received significant attention as it helps non-expert users interact with complex tabular data. This problem is originally framed as semantic parsing, also known as Text-to-SQL parsing (Yu et al., 2018, 2019; Wang et al., 2020b; Guo et al., 2021), in which the parser takes both question and table header as input, and predicts a SQL query that is directly executable to get the answer. However, training state-of-the-art Text-to-SQL parsers require large amounts of expensive SQL annotations, limiting its applicability to real scenarios; In addition, these Text-to-SQL parsers make a simplified assumption that only table headers are necessary while ignoring the value of table contents. To mitigate these issues, recent works ignore generating SQL queries, and instead follow *retrieve then reason* paradigm (Yin et al., 2020; Herzig et al., 2020; Eisenschlos et al., 2020; Yang et al., 2022; Liu et al., 2022; Jiang et al., 2022; Zhao et al., 2022b), which first retrieve information from the table, and conduct human-like reasoning to answer the question. With the help of pre-training on large scale table corpus, these approaches have achieved remarkable results on several Table QA benchmarks, including WikiTableQuestions (Pasupat and Liang, 2015), WIKISQL-WEAK (Zhong et al., 2017), and SQA (Iyyer et al., 2017). More recently, Chen (2022) found that LLMs (Brown et al., 2020; Chen et al., 2021) with few-shot in-context learning shows promise on the Table QA task.

**Robustness in Table-Relevant Task** Assessing model robustness is crucial for building trustworthy models (Wang et al., 2021; Chang et al., 2021; Goel et al., 2021; Wang et al., 2022a,b; Gupta et al., 2022). Recent work (Gan et al., 2021; Zeng et al., 2020; Chang et al., 2023) has focused on evaluating the robustness of text-to-SQL parsing models, and designed test sets with perturbations including NLQ input, table headers, and SQL queries. A major limitation is that these perturbations (e.g., lexical substitutions) are often targeted at a vulnerable key component that is specific to text-to-SQL parsing: schema linking (Wang et al., 2020a; Scholak et al., 2021), which matches table headers question keywords. Our study is focused on Table QA in general, and we make two key differences: First, in addition to existing perturbations, we also perturbed *table contents*, valuable information that is often dismissed by Text-to-SQL models. Second, unlike previous works that used human to verify perturbations generated from heuristics or models, we directly adopted human-annotated perturbations to ensure high data quality.

**Adversarial Data Generation** Existing works have proposed data augmentation and adversarial training techniques to improve model robustness. In the field of table-relevant tasks, Gan et al. (2021) applied the BERT-Attack model (Li et al., 2020) to generate adversarial training questions to im-

| Dataset | Type | # Tables | # Examples |
|---|---|---|---|
| WTQ (Pasupat and Liang, 2015) | Complex QA | 2,108 | 22,033 |
| WikiSQL-Weak (Zhong et al., 2017) | Simple QA | 24,241 | 80,654 |
| SQA (Iyyer et al., 2017) | Conversational QA | 982 | 6,066 |

Table 1: An overview of the WTQ, WikiSQL-Weak, and SQA datasets.

prove the Table QA model's robustness against synonym substitution. Pi et al. (2022) and Zhao et al. (2022a) proposed to train Table QA models over examples with perturbed database schema to defend schema-level adversarial attack. Recent approaches applied LLMs (Brown et al., 2020; Zhang et al., 2022) to generate adversarial data. For example, the evaluation data for NLQ-level perturbation in the Dr.Spider benchmark (Chang et al., 2023) were generated using LLM-prompting methods (Liu et al., 2021; Bach et al., 2022). In contrast, we created our test sets through human annotation, and applied LLMs to generate adversarial *training* examples to enhance training Table QA models.

## 3 RobuT Benchmark

We constructed RobuT to comprehensively evaluate the robustness of Table QA models against task-specific adversarial perturbations annotated by human experts. To ensure the high annotation quality of RobuT benchmark, we first designed the following three *annotation principles*:

- **Diagnostic Comprehensiveness:** To provide a comprehensive study, the benchmark should enumerate different diagnostic angles over multiple task-specific perturbation categories.
- **Phraseology Correctness and Richness:** The perturbations should follow linguistic phraseology conventions and are linguistically rich, which cannot be achieved by rule-based or model-based methods.
- **Semantic Association:** The perturbed part should still maintain the meanings of the original contexts, e.g., the new table should maintain the same domain after adding a few columns.

Following the aforementioned annotation principles, we curated RobuT based on the *development set*[1] of three mainstream Table QA datasets: WTQ (Pasupat and Liang, 2015), which contains human-annotated questions over Wikipedia tables and requires complex reasoning; Weakly-supervised WikiSQL (Zhong et al., 2017), which requires models to filter and optionally aggregate on table cell values to obtain the answer; and SQA (Iyyer et al., 2017), in which annotators decompose questions originally from WTQ to a sequence of questions (2.9 questions per sequence on average). The statistics of these three Table QA datasets are shown in Table 1.

We designed a total of 10 perturbation types on four different levels (i.e., table header, table content, natural language question, and mix). And as we have three subsets, our final dataset includes 30 *test sets* in total. Each test set contains parallel pre-perturbation and post-perturbation data to measure model robustness against the perturbation. In total, RobuT contains 138,149 pairs of examples, including 39,471 examples from RobuT-WTQ, 83,816 examples from RobuT-WikiSQL, and 14,862 examples from RobuT-SQA.

### 3.1 Table Header Perturbation

Table QA models often match the question segments to the table header in order to identify the relevant columns. However, most examples in existing Table QA datasets only consist of *exact match* scenarios (Suhr et al., 2020), leaving it unclear if models can handle table header variations. The goal of table header perturbation is to replace some column names of the table header with their *synonyms* or *abbreviations* that might mislead existing Table QA models.

**Header Synonym Replacement** Given a table, the annotators were asked to first identify the columns that can be renamed. For each candidate column, they were required to come up with a synonymous column name that maintains the same domain-relevancy. For example, the column “`runner-up`" in a table about sports can be renamed as “`second place`". The annotators were given full access to a public synonym website[2] as the reference of the synonymous names.

[1] For WTQ and SQA datasets that have multiple official train/dev splits for the purpose of cross-validation, we used the split of `random-split-1-{train/dev}` in our work.

[2] `https://www.thesaurus.com/`

**Header Abbreviation Replacement** For each table, we first collected abbreviation(s) of its column names, using APIs provided by a public abbreviation website[3]. The abbreviation would replace the original column name if the annotators decided that it is appropriate for the given table context.

## 3.2 Table Content Perturbation

To answer the given question, Table QA models should understand table contents, retrieve relevant cells, and reason over them. However, Yang et al. (2022) has found that existing Table QA models learn unwanted bias related to the table contents. In our preliminary work, we also found that questions in WTQ often use information from the first three or last two rows of the table as the answer. This finding suggests that the existing Table QA datasets actually contain annotation bias related to table content, as annotators are more likely to compose questions for the first or last few rows of the table. To evaluate the Table QA model robustness against table content variation, we designed five perturbation types to alter the table content in column-level or row-level that do not affect the final answers.

**Row Order or Column Order Shuffling** For each table, we randomly shuffled the order of its rows or columns. We excluded a small number of questions asking about the absolute table position since their answers will change after shuffling (e.g., “what is the last region listed on the table?").

**Column Extension** Column extension perturbation extends existing columns, including column name and column content, into multiple semantic-equivalent columns. Instead of using rule-based methods (Zhao et al., 2022a), we asked annotators to provide possible semantically equivalent substitutions for each column. Specifically, they were asked to decompose a compound column into multiple columns, such as replacing the column “`Score`” in a table about soccer games with “`Home Team Score`" and “`Away Team Score`".

**Column Masking** Some table columns are correlated to each other. For example, the column “`Ranking`” can be inferred by another column “`Total Points`”. We asked the annotators to mask the columns whose content could be inferred by other columns.

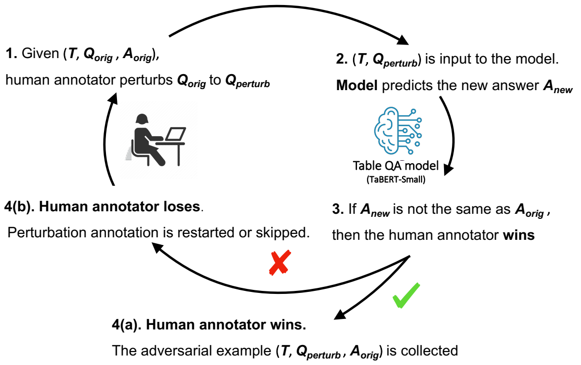

1. Given ($T$, $Q_{orig}$ , $A_{orig}$),
human annotator perturbs $Q_{orig}$ to $Q_{perturb}$
2. ($T$, $Q_{perturb}$) is input to the model.
Model predicts the new answer $A_{new}$
Table QA model
(TaBERT-Small)
3. If $A_{new}$ is not the same as $A_{orig}$ ,
then the human annotator wins
4(a). Human annotator wins.
The adversarial example ($T$, $Q_{perturb}$ , $A_{orig}$) is collected
4(b). Human annotator loses.
Perturbation annotation is restarted or skipped.

Figure 2: Overview of adversarial annotation process to collect perturbed NLQs in word-level and sentence-level using a model in the loop. $A_{orig}$ is the answer predicted by the Table QA model (i.e., TaBERT-small), given the table $T$ and pre-perturbed question $Q_{orig}$.

**Column Adding** Column adding perturbs table content by introducing new columns that are semantically associated with the original table context. Following Pi et al. (2022), for each table, we applied the TAPAS-based dense retriever (Herzig et al., 2020) to retrieve the most relevant tables from Web Data Commons database (Lehmberg et al., 2016). We collected the three most relevant tables for each source table. The annotators were then asked to follow the *semantic-association* annotation principle, and select some columns that can be randomly inserted into the original table.

## 3.3 NLQ Perturbation

In addition to table headers and contents, the input questions also affect model robustness. Our initial analysis found that questions from existing datasets contain annotation bias, causing models to learn shortcuts. For example, in WTQ, questions related to *counting* operation usually start with the phrase “how many”. And if we change the phase to “what is the quantity of", the fine-tuned models are likely to predict wrong, as they rely on the alignments between “how many” and *counting* operation.

To systematically evaluate Table QA model robustness against NLQ perturbation, we applied a model-in-the-loop adversarial example annotation framework (Bartolo et al., 2020) to collect new questions perturbed in word-level and sentence-level. As shown in Figure 2, a finetuned TaBERT-small (Yin et al., 2020) model was integrated into the annotation process. The annotators could directly interact with the model predictions during the annotation process. They were required to perturb questions at the word-level or sentence-level

[3] `https://www.abbreviations.com/`

| Level | Perturbation Type | # Example | TAPAS | | TableFormer | | TAPEX | | OmniTab | | GPT-3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Acc | R-Acc | Acc | R-Acc | Acc | R-Acc | Acc | R-Acc | Acc | R-Acc |
| | Development Set | 2,831 | 48.3 | – | 51.3 | – | 57.3 | – | 61.0 | – | 42.9 | – |
| Table Header | Synonym Replacement | 4,185 | 44.7 / 38.5<br>(-6.2) | 81.1 | 47.0 / 41.1<br>(-5.9) | 83.2 | 54.3 / 48.4<br>(-5.9) | 84.6 | 58.5 / 54.0<br>(-4.5) | 88.0 | 41.7 / 39.9<br>(-1.8) | **90.7** |
| | Abbreviation Replacement | 2,878 | 43.4 / 35.1<br>(-8.3) | 76.1 | 45.3 / 37.1<br>(-8.2) | 76.9 | 50.4 / 44.3<br>(-6.1) | 83.7 | 54.8 / 52.0<br>(-2.8) | 89.5 | 41.5 / 39.2<br>(-2.3) | **93.8** |
| Table Content | Row Order Shuffling | 7,636 | 48.0 / 40.6<br>(-7.4) | 74.8 | 51.0 / 50.9<br>(-0.1) | **97.0** | 56.9 / 45.7<br>(-11.2) | 71.7 | 60.6 / 51.2<br>(-9.4) | 77.8 | 42.9 / 38.5<br>(-4.4) | 90.2 |
| | Column Order Shuffling | 6,508 | 45.7 / 42.5<br>(-3.2) | 86.5 | 51.2 / 51.0<br>(-0.2) | **99.1** | 54.4 / 48.5<br>(-5.9) | 81.4 | 58.4 / 56.0<br>(-2.4) | 89.2 | 40.9 / 40.0<br>(-0.9) | 93.3 |
| | Column Extension | 2,672 | 50.9 / 42.5<br>(-8.4) | 73.4 | 52.5 / 45.0<br>(-7.5) | 74.8 | 61.2 / 47.8<br>(-13.4) | 71.4 | 64.5 / 52.9<br>(-11.6) | 74.7 | 43.1 / 37.4<br>(-5.7) | **81.4** |
| | Column Masking | 425 | 47.9 / 45.2<br>(-2.7) | 91.0 | 51.0 / 47.7<br>(-3.3) | 87.2 | 56.7 / 54.4<br>(-2.3) | 94.6 | 60.4 / 58.0<br>(-2.4) | 94.9 | 42.4 / 41.9<br>(-0.5) | **97.0** |
| | Column Adding | 4,574 | 48.9 / 47.1<br>(-1.8) | **89.3** | 51.9 / 48.7<br>(-3.2) | 83.5 | 57.4 / 50.4<br>(-7.0) | 80.1 | 61.6 / 57.2<br>(-4.4) | 84.8 | 41.3 / 36.8<br>(-4.5) | 85.6 |
| NLQ | Word-Level Paraphrase | 2,346 | 45.6 / 38.6<br>(-7.0) | 77.8 | 49.5 / 42.7<br>(-6.8) | 78.5 | 54.7 / 49.2<br>(-5.5) | 84.3 | 58.0 / 54.1<br>(-3.9) | 86.8 | 41.2 / 40.3<br>(-0.9) | **93.7** |
| | Sentence-Level Paraphrase | 2,404 | 45.6 / 41.1<br>(-4.5) | 80.8 | 49.6 / 44.0<br>(-5.6) | 77.1 | 54.8 / 49.5<br>(-5.3) | 84.0 | 58.2 / 55.4<br>(-2.8) | 87.0 | 41.0 / 40.5<br>(-0.5) | **94.2** |
| Mix | – | 3,012 | 44.5 / 32.0<br>(-12.5) | 64.7 | 47.6 / 35.3<br>(-12.3) | 63.4 | 52.0 / 39.5<br>(-12.5) | 70.5 | 64.5 / 43.2<br>(-11.3) | 74.0 | 37.4 / 30.6<br>(-6.8) | **83.2** |

Table 2: Data statistics and robustness evaluation results of state-of-the-art Table QA models on ROBUT-WTQ. ACC represents the *Pre-* and *Post-perturbation Accuracy*; R-ACC represents the *Robustness Accuracy*. Bold numbers indicate the highest *Robustness Accuracy* in each perturbation type, and underscores denote the second best result. When evaluating GPT-3 (i.e., `text-davinci-003`) in a few-shot setting, we reported results on 200 randomly sampled examples for each perturbation type. The results of ROBUT-WIKISQL and ROBUT-SQA are shown in Table 7 and 8 in Appendix.

that could change the model's predictions.

**Word-level Perturbation** For word-level NLQ perturbation, we required annotators to focus on perturbing the key entities in the question, such as replacing the entity with its synonym.

**Sentence-level Perturbation** For sentence-level NLQ perturbation, we required annotators to focus on perturbing the sentence structure, while maintaining its overall meaning. We did not consider the adversarial type of adding noise to the original question as it would change question's meaning.

### 3.4 Mix Perturbation

In previous subsections, we isolated each adversarial perturbation type into a separate evaluation set so that researchers can diagnose the robustness of their developed models from different aspects. This will help researchers understand which aspects of robustness require further enhancement, and improve their models accordingly. We also added a mix-perturbation evaluation set by combining two or three different-level annotated perturbations for each example. This evaluation set provides insights about the overall robustness of Table QA models.

## 4 Diagnostic Experiments

In this section, we evaluate existing Table QA models on our constructed benchmark, ROBUT.

### 4.1 Experimental Setup

**Compared Table QA models** We evaluated the following four representative table QA models on ROBUT, which first pre-trained on the collected large table corpus and then fine-tuned on the downstream Table QA tasks.

- **TAPAS** (Herzig et al., 2020) is based on BERT's encoder with additional positional embeddings for encoding tabular structure and two classification layers for cell selection and aggregation operator predictions.

- **TableFormer** (Yang et al., 2022) adapts TAPAS by introducing a learnable attention biases to mitigate the unwanted bias brought from row and column encoding.

- **TAPEX** (Liu et al., 2022) models the Table QA as a sequence-to-sequence task, and uses BART (Lewis et al., 2020) as the backbone without any table-relevant architecture design.

- **OmniTab** (Jiang et al., 2022) uses the same backbone as TAPEX, and is further pre-trained on collected natural and synthetic Table QA examples.

We also evaluated the **GPT-3** (Brown et al., 2020) model in a few-shot setting.

**Implementation Details** Since ROBUT only includes evaluation data, we fine-tuned the `Large` version of each Table QA model using the original Table QA training set and obtained three variants for WTQ, WIKISQL-WEAK, and SQA. As WTQ and SQA datasets have multiple official train/dev splits for the purpose of cross-validation, we used the split of `random-split-1-train` for fine-tuning. Specifically, WTQ training set contains 11,321 examples, WIKISQL-WEAK training set contains 56,355 examples, and SQA training set contains 4,257 sequences. We randomly split each official training set into a train/dev set with a ratio of 8:2 for fine-tuning. We ran 20 epochs with a batch size of 128 for each fine-tuning experiments and selected the best fine-tuning checkpoint based on the validation loss on the splitted dev set.

In terms of GPT-3 few-shot experiments, we used `text-davinci-003` via the public OpenAI APIs[4] with *two-shot* prompting. Similar to Chen (2022), we used a temperature of 0.7 without any frequency penalty and without top-k truncation. An example of "chain-of-thought" prompt prefix is shown in Figure 6 in Appendix.

**Evaluation Metrics** We used *Exact Match Accuracy* as the evaluation metric, which checks whether the predicted answers are equal to the ground truth. For SQA, we reported the average accuracy for sequential questions. We used the following three metrics to evaluate model robustness: **Pre-perturbation Accuracy** over pre-perturbation data; **Post-perturbation Accuracy** over post-perturbation data; **Robustness Accuracy** as the ratio of the correct predictions on both pre- and post-perturbation data versus the correct predictions on pre-perturbation data.

### 4.2 Diagnostic Results

According to Table 2 and Table 7, 8 in Appendix, all examined Table QA models exhibited significant performance drops for each perturbation type, thus are not robust under adversarial attacks.

**Effect of Model Architecture** We found that TableFormer is the most robust against row and column order shuffling, with the help of its task-independent relative attention mechanism for table-text encoding. Despite that, for most perturbation types, TAPAS and TableFormer, even with specific table encoding designs, do not outperform TAPEX and OmniTab in robustness. Therefore, we conclude that model architectures may help defend specific but not all perturbation attacks.

**Large Language Model is more Robust** In context learning with GPT-3 is more robust than other models in most perturbation categories. First, the significantly larger pre-training corpus size and model parameters allow GPT-3 to better generalize to new data (Wei et al., 2022a). Second, as discussed in Section 3.2, existing Table QA datasets contain *annotation bias* related to both table contents and questions. And fine-tuned models, therefore, learn shortcuts to overfit to the training data, which limits their ability to defend against perturbations. To provide more insights into the robustness of in-context learning with large language models, we also evaluated various types of GPT series models (i.e., `text-davinci-002`, `text-davinci-003`, and `gpt-3.5-turbo`) on the ROBUT-WTQ set. As shown in Table 3, GPT series models with higher post-perturbation accuracy correlated with higher robustness accuracy in most cases.

## 5 LETA Framework

Motivated by the diagnostic results that LLMs are more robust against human-annotated perturbations, we adopted LLMs to enhance the robustness of *smaller* (i.e., less than 1B parameter) and *fine-tuned* Table QA models. Specifically, we introduced **L**LM-**E**nhanced **T**able QA **A**ugmentation (LETA) framework, which generates adversarial training examples at scale using the LLM prompting method, to improve model robustness against human-annotated adversarial perturbations.

Specifically, for each perturbation type in ROBUT, we designed task-specific "chain-of-thought" prompts (Wei et al., 2022b; Chen, 2022) to guide the GPT-3 (i.e., `text-davinci-003`) or CodeX (i.e., `code-davinci-002`) models to generate adversarial examples to enhance the training set. We repeated example generation three times to create diverse training data. We next discuss the details for each augmentation level.

[4] https://openai.com/api/

| Level | Perturbation Type | text-davinci-002 | | text-davinci-003 | | gpt-3.5-turbo | |
|---|---|---|---|---|---|---|---|
| | | Post-Acc | R-Acc | Post-Acc | R-Acc | Post-Acc | R-Acc |
| | Development Set | 40.3 | – | 42.9 | – | 43.7 | – |
| Table Header | Synonym Replace | 39.2 | 87.8 | 39.9 | 90.7 | **42.0** | **90.9** |
| | Abbrev Replace | 37.1 | 90.1 | 39.2 | 93.8 | **40.0** | **94.4** |
| Table Content | Row Shuffle | 35.7 | 87.4 | **38.5** | **90.2** | 36.6 | 88.7 |
| | Col Shuffle | 36.0 | 90.4 | **40.0** | **93.3** | 39.5 | 92.6 |
| | Col Extension | 35.2 | 79.5 | 37.4 | 81.4 | **38.0** | **82.0** |
| | Col Mask | 40.1 | 94.0 | 41.9 | **97.0** | **42.2** | 96.4 |
| | Col Add | 33.7 | 80.2 | 36.8 | 85.6 | **37.0** | **86.1** |
| NLQ | Word-Level | 37.9 | 93.3 | 40.3 | **93.7** | **40.7** | 93.5 |
| | Sentence-Level | 40.6 | 93.7 | 40.5 | **94.2** | **41.2** | 93.4 |
| Mix | – | 29.7 | 82.5 | 30.6 | 83.2 | **31.4** | **84.9** |

Table 3: *Post-perturbation Accuracy* and *Robustness Accuracy* of GPT series models on RobuT-WTQ. Models with higher post-perturbation accuracy correlated with higher robustness accuracy in most cases. Due to the budget constraints, we reported results on 200 randomly sampled examples for each perturbation type.

### 5.1 Table Header Augmentation

For both *header synonym* and *header abbreviation replacements* type, we randomly selected 10 examples from human-annotated perturbations as demonstrations. Each example includes the table header and first two rows as input and the perturbed table header as output (Figure 4 in Appendix).

### 5.2 Table Content Augmentation

For *column extension* and *column masking* types, we provided 8 demonstration examples. Each example includes the original table, the extended (or masked) column, and the corresponding explanations. For *column adding* type, we applied an existing table dense retriever to find the three most relevant tables (Section 3.2), and then prompted the CodeX model to added one or two new columns from the retrieved tables. Figure 5 in Appendix shows a prompt prefix example for *column adding*. For *row* or *column order shuffling*, we used heuristics to produce perturbed source table variants.

### 5.3 NLQ Augmentation

We analyzed the human-annotated perturbed questions and summarized three paraphrase categories at the word level, and two categories at the sentence level. Table 9 in Appendix shows examples for each category.

**Word-level NLQ** We focused on paraphrasing three types of question words: 1) *reasoning operation indicators* (e.g., "how many" - counting operation), to infer the reasoning type; 2) *table header indicators* (e.g., "who" - "athlete" column), to locate the relevant columns; and 3) *cell value indicators* (e.g., US - "USA" cell), to locate the relevant cells.

**Sentence-level NLQ** We designed two task-specific perturbations in terms of *sentence simplification* (e.g., "at the first place of" - "number one") and *interrogative transformation* (e.g., "when was" - "Please provide me with"). We also included *general syntactic perturbations* (e.g., "stock codes" - "ticker symbols") in sentence-level paraphrasing.

For each paraphrase category at word and sentence level, we designed five to eight demonstration examples to prompt GPT for paraphrased questions. Each example includes the original question, paraphrased question, and corresponding explanations.

## 6 Adversarial Training Experiments

In this section, we evaluate LeTA on our constructed benchmarks, RobuT.

### 6.1 Experiment Setup

**Baseline System** To compare with LeTA, we developed a competitive adversarial training data generation pipeline, RTA, which applied rule-based methods to generate adversarial augmentation data for each perturbation type in terms of table header and table content. It further used BERT-Attack (Li et al., 2020) to generate paraphrased questions.

**Implementation Details** We selected TaPas and TAPEX for experiments because they are the foun-

| Level | Perturbation Type | TAPAS | w/ RTA | w/ LeTA | TAPEX | w/ RTA | w/ LeTA |
|---|---|---|---|---|---|---|---|
| | Development Set | **48.3** | 45.3 (- 3.0) | 46.5 (- 1.8) | **57.3** | 53.6 (- 3.7) | 55.3 (- 2.0) |
| Table Header | Synonym Replace | 38.5 | 40.8 (+2.3) | **42.4** (+3.9) | 48.4 | 51.0 (+2.6) | **52.5** (+4.1) |
| | Abbrev Replace | 35.1 | 38.9 (+3.8) | **40.7** (+5.6) | 44.3 | 48.7 (+4.4) | **50.0** (+5.7) |
| Table Content | Row Shuffle | 40.6 | **42.3** (+1.7) | 42.2 (+1.6) | 45.7 | 48.1 (+2.4) | **48.2** (+2.5) |
| | Col Shuffle | 42.5 | **43.8** (+1.3) | 43.6 (+1.1) | 48.5 | **50.1** (+1.6) | **50.1** (+1.6) |
| | Col Extension | 42.5 | 44.2 (+1.7) | **46.3** (+3.8) | 47.8 | 50.0 (+2.2) | **51.3** (+3.5) |
| | Col Mask | 45.2 | 45.4 (+0.2) | **45.6** (+0.4) | 54.4 | 54.3 (- 0.1) | **54.6** (+0.2) |
| | Col Add | 47.1 | 47.6 (+0.5) | **47.9** (+0.8) | 50.4 | 53.1 (+2.7) | **54.2** (+3.8) |
| NLQ | Word-Level | 38.6 | 41.0 (+2.4) | **43.1** (+4.5) | 49.2 | 51.0 (+1.8) | **52.4** (+3.2) |
| | Sentence-Level | 41.1 | 41.7 (+0.6) | **43.6** (+2.5) | 49.5 | 50.7 (+1.2) | **52.9** (+3.4) |
| Mix | – | 32.0 | 33.1 (+1.1) | **35.2** (+3.2) | 39.5 | 41.0 (+1.5) | **42.3** (+2.8) |

Table 4: Accuracy of TaPas and TAPEX models on RobuT-WTQ before and after adversarial training. Compared with RTA, LeTA-augmented models have higher accuracy improvement across most types of perturbations.

| Level | Type | %S $\geq$ 4 | | % win | | ≈ $ Cost (100 examples) | |
|---|---|---|---|---|---|---|---|
| | | Human | LeTA | Human | LeTA | Human | LeTA |
| Table Header | Synonym | 95.5 | 90.0 | 69 | 52 | 60.0 | 1.5 |
| | Abbreviation | 90.5 | 82.5 | 76 | 41 | 60.0 | 1.5 |
| Table Content | Col Extend | 90.0 | 63.5 | 90 | 22 | 100.0 | 6.0 |
| | Col Mask | 91.5 | 69.0 | 85 | 27 | 60.0 | 6.0 |
| | Col Add | 92.0 | 70.0 | 83 | 35 | 30.0 | 8.5 |
| NLQ | Word-level | 96.0 | 90.0 | 70 | 56 | 80.0 | 1.5 |
| | Sent-level | 94.0 | 92.0 | 74 | 50 | 80.0 | 1.5 |

Table 5: Comparison of adversarial data quality and cost of human annotation and LeTA. We report 1) percent of samples that have an average score $\geq$ 4, and 2) Percentage of times the examples are selected as better (may be tied). LeTA achieves comparable performance to human annotators for table header and NLQ perturbations, with a significantly lower annotation cost. We regard the pricing for `code-davinci-002` used in table content augmentation the same as `text-davinci-003` (i.e., $0.02/1K tokens).

dations of TableFormer and OmniTab, respectively. We evaluated the model performance on RobuT-WTQ before and after adversarial training. Models were fine-tuned from scratch on corresponding augmented training sets, which included both original and adversarial training data.

### 6.2 Results

According to Table 4, compared with RTA, LeTA-augmented models have higher post-perturbation accuracy across most types of RobuT perturbations. This result demonstrates that using LLM-prompting methods to generate adversarial training examples is more effective. In addition, despite the model's performance on the original development set decreasing with augmented data, the LeTA-augmented models are better on the original development set than the RTA-augmented models. This suggests that LeTA introduces less noise into the original training sets, as LLMs generate more natural adversarial examples. Such trade-off between robustness and accuracy (i.e., adversarial robustness comes at the cost of standard performance) has also been widely observed and discussed in different ML/NLP areas (Tsipras et al., 2019; Zhang et al., 2019; Zhao et al., 2022a). We will explore how to improve robustness without compromising accuracy in our future work.

### 6.3 Analysis

To evaluate the quality of adversarial example generation, we conducted human evaluations to compare the quality of the examples generated by the LeTA framework with those created by human annotators. We further provided case studies on common errors made by LeTA.

**Comparison with Human Annotation** For each perturbation type, we sampled 100 adversarial examples from both human annotation and LeTA.

| Error Type | Example |
|---|---|
| Change original meaning | **Original**: How many districts were created in the 1900's?<br>**Paraphrased**: How many districts were created in the nineteenth century?<br>**Explanation**: Should be *twentieth* |
| Mismatch with given prompt | For the prompt of replacing carrier phrase<br>**Original**: How many players scored more than 7 points?<br>**Paraphrased**: How many athletes scored more than 7 points?<br>**Explanation**: Should paraphrase the carrier phrase *How many* |
| Information missing | **Original**: What are the names and stock code of companies whose headquarters are located in the United States?<br>**Paraphrased**: Name some companies whose headquarters are located in the United States.<br>**Explanation**: *stock code* is missing |
| Hallucination | **Original**: What is the name and nation of the singer who have a song having "Hey" in its name?<br>**Paraphrased**: What is the name and nation of the singer having a song named "Hey Ya!"<br>**Explanation**: *"Hey Ya!"* does not appear in the given context |

Table 6: Case study for common errors made by the LETA framework for NLQ perturbation. The colored text highlights model errors.

Two evaluators were then asked to rate each sample on a scale of 1 (worst) to 5 (best) and determine which example was better, between the one created by human annotators and the framework. We also estimated the annotation cost of each perturbation type for both methods. The results in Table 5 demonstrate that LETA achieves comparable performance to human annotators for table header and NLQ perturbations, with much lower annotation cost. However, it still lags behind human annotators in terms of table content perturbations, we leave future work to design more effective prompting methods for table content augmentation.

**Error analysis of LETA generation** Table 6 shows examples of perturbed questions generated by the LETA framework. We identified the following common mistakes that LETA are likely to make: 1) changing the original meaning of the questions; 2) not consistent with the demonstration in the given prompt; 3) missing important information from the original question; and 4) hallucination.

## 7 Conclusion

This work proposes ROBUT, the first benchmark for Table QA robustness. ROBUT measures the robustness of Table QA models against different levels of human-annotated perturbations. Experimental results showed that state-of-the-art models exhibited significant performance drops on our ROBUT benchmark. To address this issue, we designed the LETA framework, which utilizes LLM-promoting methods to generate adversarial training examples to enhance Table QA model robustness. We believe that our work will raise awareness among researchers about the importance of robustness in Table QA models.

## Acknowledgements

We would like to dedicate this paper to the memory of Dr. Dragomir Radev. Dr. Radev's leadership, guidance, and expertise were instrumental in shaping the direction and quality of this project. His loss is deeply felt by all of us involved. We extend our heartfelt gratitude to Dr. Radev for his passion and dedication to the NLP community.

Chen Zhao is supported by the DARPA PTG program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

## Limitations

This work focuses on diagnosing and enhancing model robustness for Table QA tasks. However, there are other types of table reasoning benchmarks, such as table fact checking (Chen et al., 2020b; Gupta et al., 2020; Aly et al., 2021) and logical table-to-text generation (Chen et al., 2020a; Cheng et al., 2022), whose model robustness has not been well explored. We believe future work could extend the approaches for constructing ROBUT to these other table reasoning benchmarks, providing a more comprehensive understanding of model robustness for table understanding and reasoning tasks. Moreover, we did not consider those perturbations related to modifying the original cell values, which might change the final answer and thus will take a longer time for annotation. We believe future work could explore perturbations at the cell level.

## Ethical Consideration

RobuT were constructed upon the development set of WTQ (Pasupat and Liang, 2015), WikiSQL-Weak (Zhong et al., 2017), and SQA (Iyyer et al., 2017) datasets, which are publicly available under the licenses of CC BY-SA 4.0[5], BSD 3-Clause[6], and MIT[7], respectively. These licenses all permit us to compose, modify, publish, and distribute additional annotations upon the original dataset. All the experiments in this paper can be run on a single NVIDIA Tesla V100-32G GPU. Our benchmark and code will be released along with the paper.

For the RobuT annotation, we hired 15 graduate students (9 females and 6 males) majoring in STEM majors. The hourly rates are in the range of $10 and $12 based on the different working speed (above the local average wage of similar jobs). We recommended that annotators spend at most 4 hours per day for annotation in order to reduce pressure and maintain a comfortable pace. The whole annotation work lasted about 30 days.

## References

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

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

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

Kai-Wei Chang, He He, Robin Jia, and Sameer Singh. 2021. Robustness and adversarial examples in natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 22–26, Punta Cana, Dominican Republic & Online. Association for Computational Linguistics.

Shuaichen Chang, Jun Wang, Mingwen Dong, Lin Pan, Henghui Zhu, Alexander Hanbo Li, Wuwei Lan, Sheng Zhang, Jiarong Jiang, Joseph Lilien, et al. 2023. Dr.spider: A diagnostic evaluation benchmark towards text-to-SQL robustness. In *International Conference on Learning Representations*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wenhu Chen. 2022. Large language models are few(1)-shot table reasoners.

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

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

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

Minseok Cho, Reinald Kim Amplayo, Seung won Hwang, and Jonghyuck Park. 2018. Adversarial tableqa: Attention supervision for question answering on tables. In *Asian Conference on Machine Learning*.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association*

[5] https://creativecommons.org/licenses/by-sa/4.0/

[6] https://opensource.org/licenses/BSD-3-Clause

[7] https://opensource.org/licenses/MIT

*for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Yujian Gan, Xinyun Chen, Qiuping Huang, Matthew Purver, John R. Woodward, Jinxia Xie, and Pengsheng Huang. 2021. Towards robustness of text-to-SQL models against synonym substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2505–2515, Online. Association for Computational Linguistics.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55, Online. Association for Computational Linguistics.

Jiaqi Guo, Ziliang Si, Yu Wang, Qian Liu, Ming Fan, Jian-Guang Lou, Zijiang Yang, and Ting Liu. 2021. Chase: A large-scale and pragmatic Chinese dataset for cross-database context-dependent text-to-SQL. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2316–2331, Online. Association for Computational Linguistics.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Vivek Gupta, Shuo Zhang, Alakananda Vempala, Yujie He, Temma Choji, and Vivek Srikumar. 2022. Right for the right reason: Evidence extraction for trustworthy tabular reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3268–3283, Dublin, Ireland. Association for Computational Linguistics.

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.

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

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

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 75–76, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

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

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

Xinyu Pi, Bing Wang, Yan Gao, Jiaqi Guo, Zhoujun Li, and Jian-Guang Lou. 2022. Towards robustness of text-to-SQL models against natural and realistic adversarial table perturbation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2022, Dublin, Ireland. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. 2020. Exploring unexplored generalization challenges for cross-database semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online. Association for Computational Linguistics.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020a. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Lijie Wang, Ao Zhang, Kun Wu, Ke Sun, Zhenghua Li, Hua Wu, Min Zhang, and Haifeng Wang. 2020b. DuSQL: A large-scale and pragmatic Chinese text-to-SQL dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6923–6935, Online. Association for Computational Linguistics.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022a. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1719–1729, Seattle, United States. Association for Computational Linguistics.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022b. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models.

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

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

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.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

Jichuan Zeng, Xi Victoria Lin, Steven C.H. Hoi, Richard Socher, Caiming Xiong, Michael Lyu, and Irwin King. 2020. Photon: A robust cross-domain text-to-SQL system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 204–214, Online. Association for Computational Linguistics.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

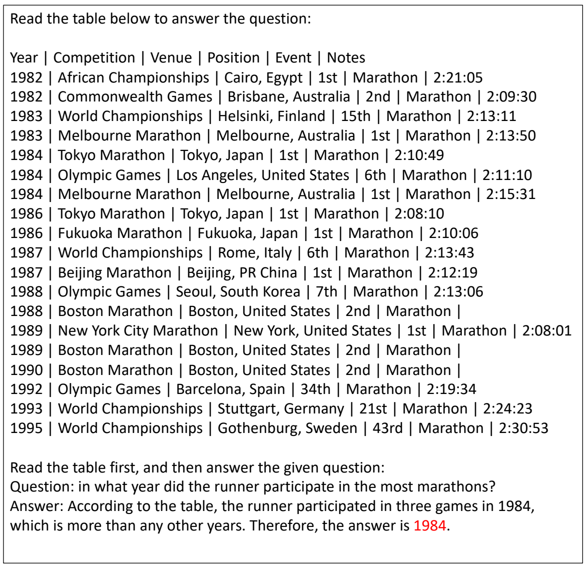
Chen Zhao, Yu Su, Adam Pauls, and Emmanouil Antonios Platanios. 2022a. Bridging the generalization gap in text-to-SQL parsing with schema expansion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5568–5578, Dublin, Ireland. Association for Computational Linguistics.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022b. ReasTAP: Injecting table reasoning skills during pre-training via synthetic reasoning examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9006–9018, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Yi Zhu, Yiwei Zhou, and Menglin Xia. 2020. Generating semantically valid adversarial questions for tableqa.

## A Appendix



```
Read the table below to answer the question:

Year | Competition | Venue | Position | Event | Notes
1982 | African Championships | Cairo, Egypt | 1st | Marathon | 2:21:05
1982 | Commonwealth Games | Brisbane, Australia | 2nd | Marathon | 2:09:30
1983 | World Championships | Helsinki, Finland | 15th | Marathon | 2:13:11
1983 | Melbourne Marathon | Melbourne, Australia | 1st | Marathon | 2:13:50
1984 | Tokyo Marathon | Tokyo, Japan | 1st | Marathon | 2:10:49
1984 | Olympic Games | Los Angeles, United States | 6th | Marathon | 2:11:10
1984 | Melbourne Marathon | Melbourne, Australia | 1st | Marathon | 2:15:31
1986 | Tokyo Marathon | Tokyo, Japan | 1st | Marathon | 2:08:10
1986 | Fukuoka Marathon | Fukuoka, Japan | 1st | Marathon | 2:10:06
1987 | World Championships | Rome, Italy | 6th | Marathon | 2:13:43
1987 | Beijing Marathon | Beijing, PR China | 1st | Marathon | 2:12:19
1988 | Olympic Games | Seoul, South Korea | 7th | Marathon | 2:13:06
1988 | Boston Marathon | Boston, United States | 2nd | Marathon |
1989 | New York City Marathon | New York, United States | 1st | Marathon | 2:08:01
1989 | Boston Marathon | Boston, United States | 2nd | Marathon |
1990 | Boston Marathon | Boston, United States | 2nd | Marathon |
1992 | Olympic Games | Barcelona, Spain | 34th | Marathon | 2:19:34
1993 | World Championships | Stuttgart, Germany | 21st | Marathon | 2:24:23
1995 | World Championships | Gothenburg, Sweden | 43rd | Marathon | 2:30:53

Read the table first, and then answer the given question:
Question: in what year did the runner participate in the most marathons?
Answer: According to the table, the runner participated in three games in 1984,
which is more than any other years. Therefore, the answer is 1984.
```

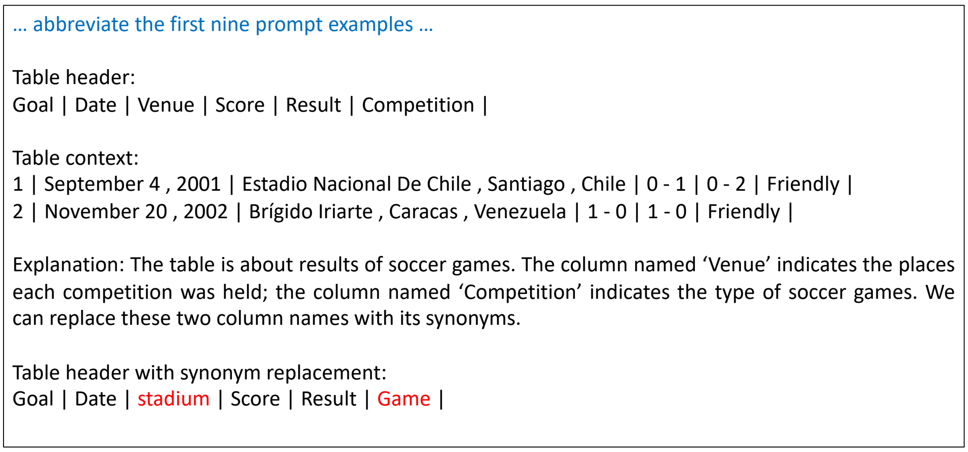
Figure 3: An example of GPT-3 “chain-of-thought” prompt prefix for the Table QA tasks.

| Level | Perturbation Type | # Example | TAPAS | | TableFormer | | TAPEX | | OmniTab | | GPT-3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ACC | R-ACC | ACC | R-ACC | ACC | R-ACC | ACC | R-ACC | ACC | R-ACC |
| | Development Set | 8,421 | 87.1 | – | 85.8 | – | 89.5 | – | 88.8 | – | 78.3 | – |
| Table Header | Synonym Replacement | 9,419 | 81.2 / 62.2<br>(-19.0) | 73.1 | 80.7 / 64.0<br>(-16.7) | 75.8 | 83.6 / 68.8<br>(-14.8) | 79.5 | 82.3/70.7<br>(-11.6) | 82.0 | 78.1 / 74.5<br>(-3.6) | **91.8** |
| | Abbreviation Replacement | 8,229 | 81.7 / 59.5<br>(-22.2) | 69.9 | 81.0 / 57.7<br>(-23.3) | 66.7 | 82.9/70.7<br>(-22.2) | 82.5 | 82.1 / 73.2<br>(-18.9) | 85.8 | 78.5 / 75.1<br>(-3.4) | **89.1** |
| Table Content | Row Order Shuffling | 17,490 | 84.8 / 80.1<br>(-4.7) | 91.1 | 85.7 / 85.2<br>(-0.5) | **96.9** | 88.5 / 83.0<br>(-5.5) | 86.2 | 87.6 / 82.4<br>(-5.2) | 87.9 | 78.2 / 76.5<br>(-1.7) | 92.3 |
| | Column Order Shuffling | 16,532 | 85.6 / 83.9<br>(-1.7) | 93.0 | 84.9 / 84.8<br>(-0.1) | **99.3** | 89.0 / 87.4<br>(-1.6) | 92.1 | 87.6 / 85.3<br>(-2.3) | 90.4 | 77.5 / 76.9<br>(-0.6) | 94.4 |
| | Column Extension | 2,626 | 89.8 / 51.9<br>(-37.9) | 56.4 | 86.0 / 50.8<br>(-35.2) | **58.8** | 92.0 / 53.2<br>(-38.8) | 57.1 | 91.2 / 53.8<br>(-37.4) | 57.6 | 80.2 / 55.5<br>(-34.7) | 56.4 |
| | Column Masking | 1,153 | 85.1 / 79.2<br>(-5.9) | **87.4** | 84.8 / 76.9<br>(-7.9) | 85.0 | 89.5 / 82.4<br>(-7.1) | 80.6 | 88.6 / 82.1<br>(-6.5) | 81.5 | 78.2 / 74.7<br>(-3.5) | 85.6 |
| | Column Adding | 6,444 | 77.8 / 69.6<br>(-8.2) | **81.6** | 75.4 / 67.3<br>(-8.1) | 80.1 | 81.4 / 64.9<br>(-16.5) | 71.0 | 79.7 / 66.7<br>(-13.0) | 71.3 | 78.3 / 70.5<br>(-7.8) | 81.0 |
| NLQ | Word-Level Paraphrase | 5,024 | 82.7 / 58.9<br>(-23.8) | 68.0 | 82.1 / 57.0<br>(-25.1) | 66.7 | 85.8 / 64.2<br>(-21.6) | 72.6 | 84.7 / 64.3<br>(-20.4) | 72.6 | 76.3 / 72.2<br>(-4.1) | **92.2** |
| | Sentence-Level Paraphrase | 3,726 | 79.3 / 66.7<br>(-12.6) | 78.6 | 76.8 / 64.5<br>(-12.3) | 79.1 | 81.6 / 68.7<br>(-12.9) | 80.8 | 80.6 / 70.1<br>(-10.5) | 81.3 | 75.0 / 72.6<br>(-2.4) | **95.1** |
| Mix | – | 4,752 | 70.8 / 52.9<br>(-17.9) | 69.9 | 70.1 / 51.2<br>(-18.9) | 67.5 | 80.1 / 60.3<br>(-19.8) | 70.7 | 79.2 / 64.2<br>(-18.0) | 71.2 | 69.5 / 60.1<br>(-9.4) | **80.2** |

Table 7: Data statistics and robustness evaluation results of state-of-the-art Table QA models on ROBUT-WIKISQL. ACC represents the *Pre-* and *Post-perturbation Accuracy*; R-ACC represents the *Robustness Accuracy*. Bold numbers indicate the highest *Robustness Accuracy* in each perturbation type, and underscores denote the second best result. When evaluating GPT-3 in a few-shot setting, we reported results on 200 randomly sampled examples for each perturbation type.

| Level | Perturbation Type | # Example | TAPAS | | TAPEX | | GPT-3 | |
|---|---|---|---|---|---|---|---|---|
| | | | ACC | R-ACC | ACC | R-ACC | ACC | R-ACC |
| | Development Set | 784 | 63.7 | – | 67.9 | – | 50.1 | – |
| Table Header | Synonym Replacement | 2,104 | 64.4 / 57.7<br>(-6.7) | 85.7 | 68.6 / 62.0<br>(-6.6) | 86.5 | 50.7 / 47.2<br>(-3.5) | **91.3** |
| | Abbreviation Replacement | 1,286 | 62.9 / 50.0<br>(-12.9) | 76.9 | 68.5 / 59.7<br>(-8.8) | 83.8 | 51.0 / 47.3<br>(-3.7) | **90.6** |
| Table Content | Row Order Shuffling | 2,356 | 60.9 / 55.3<br>(-5.6) | 85.0 | 64.1 / 60.2<br>(-3.9) | 88.9 | 49.2 / 47.4<br>(-1.8) | **93.7** |
| | Column Order Shuffling | 2,079 | 61.3 / 60.5<br>(-0.8) | **94.8** | 66.7 / 65.2<br>(-1.5) | 89.8 | 49.5 / 49.0<br>(-0.5) | 94.5 |
| | Column Extension | 1,540 | 62.4 / 40.8<br>(-21.6) | 62.1 | 66.8 / 42.0<br>(-24.8) | 58.9 | 49.5 / 34.9<br>(-14.6) | **60.8** |
| | Column Masking | 177 | 65.2 / 62.3<br>(-2.9) | **89.6** | 68.3 / 65.0<br>(-3.3) | 87.4 | 51.3 / 49.3<br>(-2.0) | **89.6** |
| | Column Adding | 2,254 | 62.7 / 60.8<br>(-1.9) | **92.7** | 67.3 / 58.9<br>(-8.4) | 81.9 | 50.2 / 48.5<br>(-1.7) | 91.6 |
| NLQ | Word-Level Paraphrase | 1,198 | 63.6 / 57.7<br>(-5.9) | 86.1 | 68.5 / 63.1<br>(-5.4) | 86.9 | 50.4 / 49.8<br>(-0.6) | **95.2** |
| | Sentence-Level Paraphrase | 1,084 | 62.8 / 57.5<br>(-5.3) | 85.7 | 68.0 / 61.9<br>(-6.1) | 86.0 | 49.8 / 49.5<br>(-0.3) | **96.3** |

Table 8: Data statistics and robustness evaluation results of state-of-the-art Table QA models on ROBUT-SQA. ACC represents the *Pre-* and *Post-perturbation Accuracy*; R-ACC represents the *Robustness Accuracy*. Due to the time constraint, we did not construct the *mix* set for ROBUT-SQA.



... abbreviate the first nine prompt examples ...

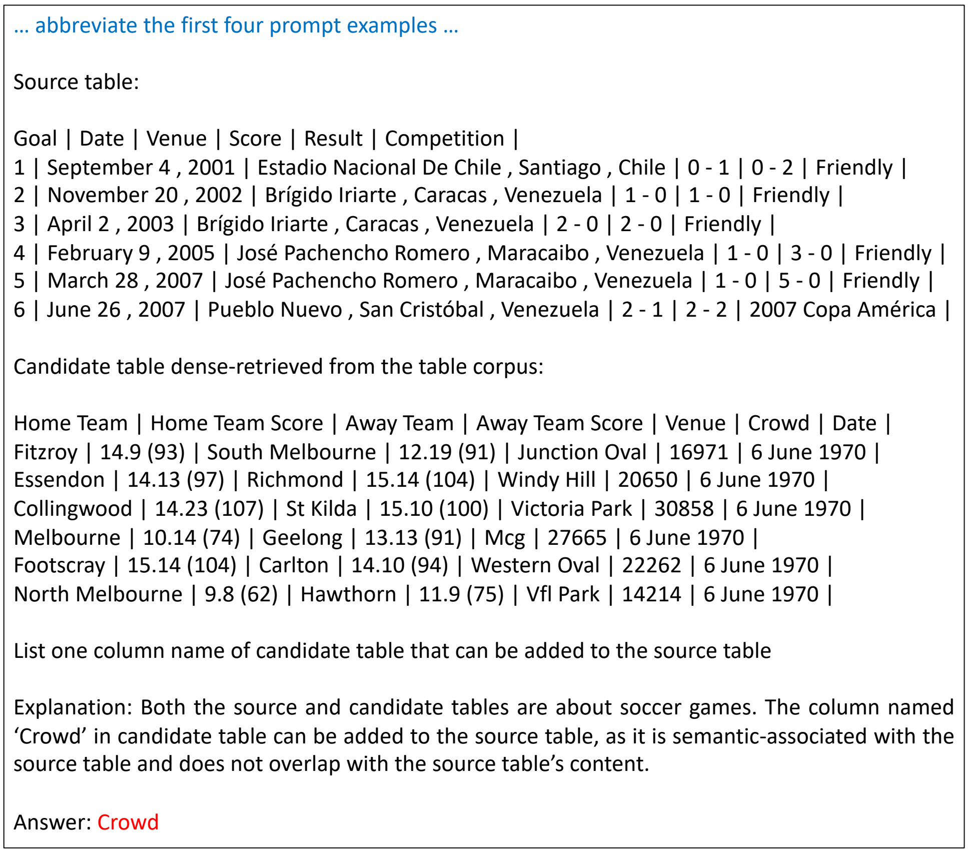
Table header:
Goal | Date | Venue | Score | Result | Competition |

Table context:
1 | September 4 , 2001 | Estadio Nacional De Chile , Santiago , Chile | 0 - 1 | 0 - 2 | Friendly |
2 | November 20 , 2002 | Brígido Iriarte , Caracas , Venezuela | 1 - 0 | 1 - 0 | Friendly |

Explanation: The table is about results of soccer games. The column named 'Venue' indicates the places each competition was held; the column named 'Competition' indicates the type of soccer games. We can replace these two column names with its synonyms.

Table header with synonym replacement:
Goal | Date | stadium | Score | Result | Game |

Figure 4: An example of prompt prefix for *header synonym replacement* using GPT-3. The GPT-3 model is prompted to perturb the table header, given the table context (i.e., table header, and first two rows of the table).



... abbreviate the first four prompt examples ...

Source table:

Goal | Date | Venue | Score | Result | Competition |
1 | September 4 , 2001 | Estadio Nacional De Chile , Santiago , Chile | 0 - 1 | 0 - 2 | Friendly |
2 | November 20 , 2002 | Brígido Iriarte , Caracas , Venezuela | 1 - 0 | 1 - 0 | Friendly |
3 | April 2 , 2003 | Brígido Iriarte , Caracas , Venezuela | 2 - 0 | 2 - 0 | Friendly |
4 | February 9 , 2005 | José Pachencho Romero , Maracaibo , Venezuela | 1 - 0 | 3 - 0 | Friendly |
5 | March 28 , 2007 | José Pachencho Romero , Maracaibo , Venezuela | 1 - 0 | 5 - 0 | Friendly |
6 | June 26 , 2007 | Pueblo Nuevo , San Cristóbal , Venezuela | 2 - 1 | 2 - 2 | 2007 Copa América |

Candidate table dense-retrieved from the table corpus:

Home Team | Home Team Score | Away Team | Away Team Score | Venue | Crowd | Date |
Fitzroy | 14.9 (93) | South Melbourne | 12.19 (91) | Junction Oval | 16971 | 6 June 1970 |
Essendon | 14.13 (97) | Richmond | 15.14 (104) | Windy Hill | 20650 | 6 June 1970 |
Collingwood | 14.23 (107) | St Kilda | 15.10 (100) | Victoria Park | 30858 | 6 June 1970 |
Melbourne | 10.14 (74) | Geelong | 13.13 (91) | Mcg | 27665 | 6 June 1970 |
Footscray | 15.14 (104) | Carlton | 14.10 (94) | Western Oval | 22262 | 6 June 1970 |
North Melbourne | 9.8 (62) | Hawthorn | 11.9 (75) | Vfl Park | 14214 | 6 June 1970 |

List one column name of candidate table that can be added to the source table

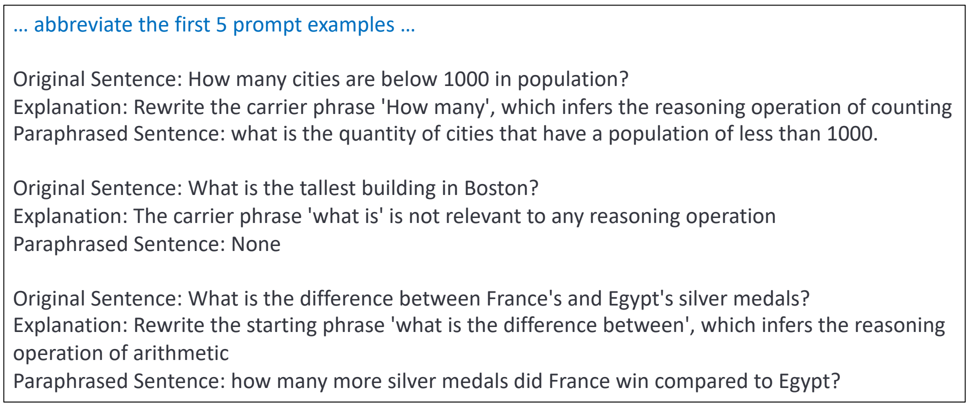
Explanation: Both the source and candidate tables are about soccer games. The column named 'Crowd' in candidate table can be added to the source table, as it is semantic-associated with the source table and does not overlap with the source table's content.

Answer: Crowd

Figure 5: An example of prompt prefix for *column adding* perturbation using CodeX. The candidate table is retrieved by the TAPAS-based dense retriever. The CodeX model is prompted to select one column from the candidate table that can be inserted into the source table.

| Level | Paraphrase Category | Paraphrased Example |
|---|---|---|
| Word | `Reasoning-synonym`<br>Paraphrase reasoning operation indicators with its synonyms | Original: Which was the first Chinese star map known to have been created?<br>Paraphrased: Which was the earliest Chinese star map known to have been created? |
| | `Reasoning-carrier`<br>Rewrite the carrier phrases that are used to infer the reasoning operation | Original: How many cities are above 1 million in population<br>Paraphrased: What is the quantity of cities that are above 1 million in population? |
| | `Header-synonym`<br>Paraphrase table header indicators with its synonyms | Original: Who had more points, Takaji Mori or Junji Kwano?<br>Paraphrased: Who performed better, Takaji Mori or Junji Kwano?<br>Explanation: `points` is the table header name. |
| | `Header-carrier`<br>Rewrite the carrier phrases used to infer the relevant table columns | Original: What are the names of players that scored more than 5 points.<br>Paraphrased: Which athletes scored more than 5 points?<br>Explanation: `Player Name` is the table header name. |
| | `Cell-Value-synonym`<br>Paraphrase cell value indicators with its synonyms | Original: How many districts were created in the 1900's?<br>Paraphrased: How many districts were created in the twentieth century? |
| Sentence | `Simplification`<br>Simplify the question and make it less redundant | Original: How many weeks did the song "Don't Cry for Me Argentina" written by Julie Covington spend at the first place of Australia's singles chart?<br>Paraphrased: How many weeks was Julie Covington's "Don't Cry for Me Argentina" number one in Australia's singles chart?? |
| | `Interrogative Transformation`<br>Convert the question between interrogative and imperative form | Original: When was the first game that Kansas State won by double digits?<br>Paraphrased: Please provide me with the date when Kansas State won the first game by double digits. |
| | `General`<br>Paraphrase the question in a general way, which might cover multiple paraphrased categories | Original: What are the names and stock codes of companies whose headquarters are located in the United States?<br>Paraphrased: List the names and ticker symbols of companies based in the United States? |

Table 9: Examples of paraphrase categories for LETA NLQ Augmentation. The red words in the original questions highlight the text that are paraphrased. The blue words in the paraphrases represent how the text are replaced.



... abbreviate the first 5 prompt examples ...

Original Sentence: How many cities are below 1000 in population?
Explanation: Rewrite the carrier phrase 'How many', which infers the reasoning operation of counting
Paraphrased Sentence: what is the quantity of cities that have a population of less than 1000.

Original Sentence: What is the tallest building in Boston?
Explanation: The carrier phrase 'what is' is not relevant to any reasoning operation
Paraphrased Sentence: None

Original Sentence: What is the difference between France's and Egypt's silver medals?
Explanation: Rewrite the starting phrase 'what is the difference between', which infers the reasoning operation of arithmetic
Paraphrased Sentence: how many more silver medals did France win compared to Egypt?

Figure 6: An example of prompt prefix for paraphrasing NLQ with `Reasoning-synonym` category. For each paraphrase category at the word or sentence level, we designed a demonstration with five to eight examples, where each example includes the original question, the paraphrased question, and corresponding explanations to prompt GPT-3 for generating new paraphrased questions.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*in "Limitations" section*

☑ A2. Did you discuss any potential risks of your work?
*In "Ethical Consideration" section*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*in "Abstract" and "Introduction" section*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*Section 3*

☑ B1. Did you cite the creators of artifacts you used?
*Section 3*

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Ethics section*

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Not applicable. Left blank.*

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Section 3*

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Section 3*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Section 3*

### C ☑ Did you run computational experiments?

*Section 4, 6*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*section 4, 6*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*section 4, 6*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*section 4, 6*

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Not applicable. section 4, 6*

**D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*section 3, and "Ethical Condideration" section*

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*section 3, and "Ethical Condideration" section*

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*section 3, and "Ethical Condideration" section*

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*section 3, and "Ethical Condideration" section*

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*"Ethical Condideration" section*

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*"Ethical Condideration" section*