UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

FREB-TQA: a fine-grained robustness evaluation benchmark for Table Question Answering

Wei Zhou, Mohsen Mesgar, Heike Adel, Annemarie Friedrich

Angaben zur Veröffentlichung / Publication details:

Zhou, Wei, Mohsen Mesgar, Heike Adel, and Annemarie Friedrich. 2024. "FREB-TQA: a fine-grained robustness evaluation benchmark for Table Question Answering." In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 16-21 June 2024, Mexico City, Mexico*, edited by Kevin Duh, Helena Gomez, and Steven Bethard, 2479–97. Stroudsburg, PA: Association for Computational Linguistics (ACL).
https://doi.org/10.18653/v1/2024.naacl-long.137.

Nutzungsbedingungen / Terms of use:

CC BY 4.0

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
CC-BY 4.0: Creative Commons: Namensnennung
Weitere Informationen finden Sie unter: / For more information see:
https://creativecommons.org/licenses/by/4.0/deed.de

FREB-TQA: A Fine-Grained Robustness Evaluation Benchmark for Table Question Answering

Wei Zhou[1,3] **Mohsen Mesgar**[1] **Heike Adel**[2] **Annemarie Friedrich**[3]
[1]Bosch Center for Artificial Intelligence, Renningen, Germany
[2]Hochschule der Medien, Stuttgart, Germany [3]University of Augsburg, Germany
{wei.zhou|mohsen.mesgar}@de.bosch.com
annemarie.friedrich@informatik.uni-augsburg.de

Abstract

Table Question Answering (TQA) aims at composing an answer to a question based on tabular data. While prior research has shown that TQA models lack robustness, understanding the underlying cause and nature of this issue remains predominantly unclear, posing a significant obstacle to the development of robust TQA systems. In this paper, we formalize three major desiderata for a fine-grained evaluation of robustness of TQA systems. They should (i) answer questions regardless of alterations in table structure, (ii) base their responses on the content of relevant cells rather than on biases, and (iii) demonstrate robust numerical reasoning capabilities. To investigate these aspects, we create and publish a novel TQA evaluation benchmark in English. Our extensive experimental analysis reveals that none of the examined state-of-the-art TQA systems consistently excels in these three aspects. Our benchmark is a crucial instrument for monitoring the behavior of TQA systems and paves the way for the development of robust TQA systems. We release our benchmark publicly.[1]

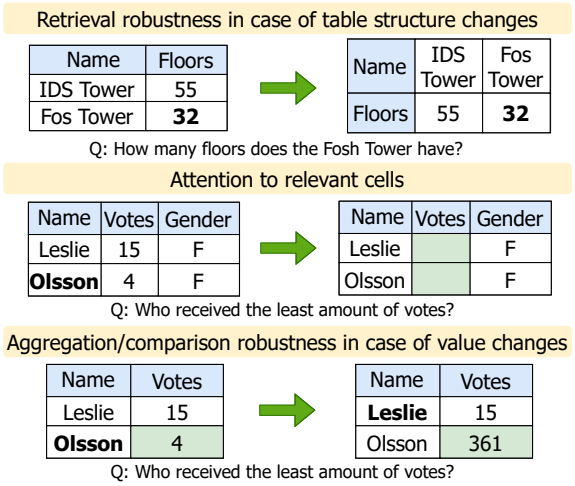

Figure 1: Our benchmark addresses three aspects of robustness shown in the yellow boxes. Answers are bold in tables. Original tables (left) are what exist in a TQA dataset and changed tables (right) show tables after perturbations. We demonstrate three perturbations in this figure (top to bottom): table transposing, removing relevant cells and modifying values to change answers.

1 Introduction

Table Question Answering (TQA) deals with answering natural language questions related to information organized in a table. TQA systems serve as a fundamental component for interacting with relational databases (Zhong et al., 2017; Yu et al., 2018) through natural language and for processing information across diverse domains, e.g., science (Desai et al., 2021) and finance (Zhu et al., 2021).

Processing tabular knowledge poses notable challenges. While tables are structured, there are no standard table layouts for representing a particular type of data. Table cells may contain various data types, including text and numbers. Moreover, tables can have nested structures. Thus, TQA systems should be able to robustly integrate textual commonsense understanding with numerical reasoning applied to structured data.

Recent benchmarks for TQA systems (Zhao et al., 2023; Yang et al., 2022) reveal that current TQA systems generate inconsistent responses by performing coarse-grained changes in tables and questions. However, these benchmarks contain a mix of questions requiring different levels of retrieval and aggregation capabilities.

As a result, these benchmarks fail to differentiate between various aspects of robustness, such as errors stemming from incorrect cell retrieval versus those arising from failures in value aggregation steps. Pinpointing the exact failures of TQA systems, however, is a necessary diagnostic step towards enhancing their robustness.

In this paper, we lay the groundwork for a more fine-grained systematic evaluation of TQA systems.

[1]https://github.com/boschresearch/FREB-TQA

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2479–2497
June 16-21, 2024 ©2024 Association for Computational Linguistics

To foster the development of robust TQA systems, we address three[2] important desiderata, as illustrated in Figure 1. With these aspects, we intend to separate several steps required when answering questions based on tabular data, i.e., identifying the relevant table cells and aggregating over or comparing them. In particular, we propose to evaluate TQA systems according to three aspects: (1) **Retrieval robustness in case of table structure changes**: Systems should retrieve the correct relevant cells regardless of table structure changes; (2) **Attention to relevant cells**: The system should use the relevant cell values from retrieval instead of exploiting shortcuts, such as model-internal knowledge or positional biases, when composing answers; and (3) **Aggregation/comparison robustness in case of value changes**: The system should aggregate the values of relevant cells correctly regardless of cell value changes. Aggregation can involve different types of reasoning. In this study, we focus on numerical operations, such as counting and comparisons, as they are the most common reasoning types in TQA (Zhu et al., 2021).

To evaluate TQA systems with regard to these aspects, we create **FREB-TQA**, a new benchmark based on four well-studied TQA datasets using seven novel automatic perturbations, one perturbation from previous work (Zhao et al., 2023), and extensive manual annotations.

In our extensive experiments, we study pipeline and end-to-end state-of-the-art TQA systems (Liu et al., 2021; Jiang et al., 2022a; Herzig et al., 2020; Cheng et al., 2022), as well as large language models (LLMs) (Touvron et al., 2023) using our benchmark. The fine-grained results uncover shortcomings of these systems that have not been previously explored. For example, we show that model performance diminishes substantially when column order changes, or when cells containing the answer are positioned at the bottom of the table. This finding indicates a strong positional bias in TQA systems, motivating more informative table encodings.

Our main contributions are: **(1)** We propose a novel benchmark of 8,590 selected and partially manually annotated questions and tables, resulting in a total of 75,205 instances using seven perturbation methods; **(2)** using several inter-annotator agreement studies, we demonstrate the solidness of our benchmark; and **(3)** using our benchmark, we experiment extensively with state-of-the-art TQA systems, discovering that for almost all robustness tests, system performances drop, demonstrating the difficulty of our benchmark for TQA systems.

[2]We acknowledge that there are additional aspects that future work should address.

2 Related Work

We provide an overview of previous research on TQA systems and their evaluation for robustness.

TQA systems. End-to-end TQA systems (Liu et al., 2021; Jiang et al., 2022b) compose an answer given a question and a serialized version of the table without intermediate steps or using additional tools. Pipeline systems convert questions into a command language, e.g., SQL (Cheng et al., 2022; Ni et al., 2023), filtered tables (Glass et al., 2021; Lei et al., 2022; Ye et al., 2023; Herzig et al., 2020), and then generate answers either via executing the commands (Ni et al., 2023; Zhang et al., 2023) or via using a trained neural network (Lei et al., 2023). Our benchmark reveals the key strengths and weaknesses of these types of systems.

Robustness evaluation. Several studies have shown that recently proposed TQA systems suffer from robustness issues (Yang et al., 2022; Zhao et al., 2023; Lin et al., 2023). Zhao et al. (2023) provide a robustness evaluation dataset for TQA, which includes header perturbations, content perturbations, and question perturbations. Their work is closely related to ours. However, they do not further disentangle the various aspects of robustness, thus not providing detailed insights into why systems are not robust. As we show in our experiments, the fine-grained aspects formulated in FREB-TQA contribute to a deeper understanding of TQA systems.

To the best of our knowledge, no previous work on TQA provides analyses of models for our proposed aspects. However, there are several studies looking into those aspects for other tasks. In the context of tabular natural language inference, Gupta et al. (2022a,b) study to what extent models pay attention to relevant cells. The robustness of large language models in case of numerical value changes has primarily been studied in the context of solving math world problems (Stolfo et al., 2022) and tabular natural language inference (Akhtar et al., 2023).

Source Dataset	# EQs	# RQs	# ORI
WTQ (Pasupat and Liang, 2015)	205	1562	2831
WikiSQL (Zhong et al., 2017)	6013	0	8418
SQA (Iyyer et al., 2017)	157	0	2265
TAT (Zhu et al., 2021)	114	539	1668

Table 1: The first two columns show the total number of questions selected from the dev part of each source dataset for extraction questions (EQs) and reasoning questions (RQs). The last column shows the original number of questions in each source dev set.

3 Our Benchmark: FREB-TQA

FREB-TQA is a **F**ine-grained **R**obustness **E**valuation **B**enchmark for **T**able **Q**uestion **A**nswering. From four TQA datasets (Table 1), we first classify questions with regard to whether questions merely require cell value retrieval or further reasoning. We then generate perturbations for evaluating each aspect of robustness by making use of seven perturbation methods and collecting human annotations.

3.1 Source Datasets

For building FREB-TQA, we leverage the development sets of four well-studied TQA datasets: WikiTableQuestions (**WTQ**, Pasupat and Liang, 2015), **WikiSQL** (Zhong et al., 2017), Sequential Question Answering (**SQA**, Iyyer et al., 2017), and Tabular And Textual dataset for Question Answering (**TAT**, Zhu et al., 2021). The first three datasets feature tables from Wikipedia, TAT addresses the financial domain. See Appendix A.1 for detailed statistics. For all source datasets, we eliminate questions that relate to the table structure, such as "What is the name of the actor in the first row?" To identify such questions, we use a word list provided in Appendix A.3. Around 10% of the questions in WTQ and 3% of TAT questions are filtered out by this criterion.

3.2 Extraction and Reasoning Questions

To decouple the robustness aspects in TQA, we group questions from the source datasets into extraction questions (EQs) and reasoning questions (RQs) by applying heuristics and classification models. The answer to an EQ can be retrieved from a single cell of the table. The answer to an RQ additionally requires aggregating over several cell values. In Figure 1, the question "How many floors does the Fosh Tower have?" is an EQ since it only requires retrieving a cell value. The question "Who received the least amount of votes?" is an RQ as it requires comparing values of several cells.

Model	EQ	RQ
LLaMA2	74.27	57.02
Rule-based	75.23	**83.42**
Combined	**93.81**	60.17

Table 2: The precision of examined models for question type classification on 200 questions from WTQ.

Question type classification for WTQ. WTQ provides a diverse set of questions. We group them into EQs and RQs using two different methods. Each method is tuned to identify either EQs or RQs with high precision, as our final benchmark will only contain the set of questions identified as EQ or RQ by the respective method.[3]

Given a table, a question, and an answer, the **lexical rule-based** method conducts a string match between the answer and the table's cell values. If there is no match, the corresponding question is labeled as RQ. Otherwise, we detect if the question contains any comparative or superlative words using POS tags.[4] Since these words are signals for aggregation over table cells, we mark the question as RQ. Otherwise, the question's type is set to EQ. We also prompt **LLaMA2**-13b (Touvron et al., 2023) to label the question type (The prompt is provided in Appendix A.4). Finally, we combine the lexical rule-based and LLaMA2 models. If the lexical rule-based model labels a question as EQ, we obtain the LLaMA's prediction in addition. If they agree, the question type is set to EQ. Otherwise, it is set to RQ. To estimate the quality of these methods, we manually annotate questions sampled randomly from WTQ (100 EQs and 100 RQs). Details are given in Appendix A.2.

Table 2 shows the results achieved by these models. As we aim for a high-precision question-type classification, we apply the combined model for identifying EQs and the lexical rule-based method for identifying RQs on all questions in WTQ.

Question type classification for WikiSQL. Questions in WikiSQL do not require complex reasoning (Zhao et al., 2023; Lin et al., 2023). In WikiSQL, questions are labeled with regard to the operations required to answer them. We select the questions that can be answered without performing

[3]Around 20% of questions are selected by neither of the models and thus eliminated from our benchmark. We also found that the final set of EQ and RQ cases do not overlap.

[4]https://www.nltk.org

aggregation operations and label them as EQs.

Question type classification for SQA. The SQA dataset consists of dialogues in the form of questions and answers related to the information in a table. The answers to all questions except for the first question in a dialog rely on the dialog history. Thus, we select only the first question from each sequence, which usually asks for retrieving information from the table (Iyyer et al., 2017). Hence, we mark them as EQs.

Question type classification for TAT. TAT consists of questions about tables and text snippets and requires numerical reasoning. The dataset also features annotations about how an answer is derived, whether the answer is based on the table or the text, and which operations are required to answer a question. We leverage these annotations to identify EQs and RQs. More specifically, we first select questions that can be answered only based on tables. Next, if a question needs derivation or comparison, we classify them as RQ. Otherwise, we classify them as EQ.

3.3 Perturbations for Testing Retrieval Robustness against Table Structure Changes

A robust TQA system should answer an EQ by retrieving the answer from a table, regardless of table structure changes. We perturb the structure of tables associated with extraction questions. We replicate one perturbation type from previous work and introduce two new perturbation types for measuring this robustness aspect.

Shuffle all rows (columns). Following Zhao et al. (2023), we randomly shuffle all rows (columns) in a table. This perturbation allows us to study if TQA systems are robust against changes in re-arranging all rows (columns). However, it does not reveal which biases may impact the robustness of TQA systems. Thus, we introduce the following two new perturbation methods for this aspect.

Shift target rows (columns). For each EQ, the *target* row (column) in a table contains the cell that corresponds to the answer. This perturbation type shifts target rows (columns) either to the top, to the middle, or to the bottom part of a table. We identify the target cell by applying exact match between the answer and table cell value. For shifting target rows, we partition a table into three equal-length parts, referred to as top, middle, and bottom. For shifting target columns, we partition a table into two equal-length parts: front and back. We use more partitions for rows because on average tables include more rows than columns (Table 3). For our benchmark, we remove the target rows (columns) from the table and re-insert them at a random position in each partition. This perturbation method allows us to study whether TQA systems exhibit any positional biases.

Transpose. This perturbation type transposes the table, i.e., it rotates the table by 90 degrees and turns rows into columns and columns into rows. This perturbation allows us to study if TQA systems have a bias towards particular table layouts.

3.4 Perturbations for Testing Attention to Relevant Cells

TQA systems may answer a question by exploiting shortcuts (model-internal knowledge or positional biases) without paying attention to relevant cells (cells that are important to compose an answer). We propose three new perturbation methods for reasoning questions (RQs) to study this aspect of TQA robustness.

Remove relevant cells. We perturb a table by removing relevant cells from tables. The relevant cell annotations associated with RQs in our benchmark have been created by Zhu et al. (2021) and Ye et al. (2023) for TAT and WTQ, respectively. We test the validity of the latter since the annotations are gathered from LLMs (see Appendix A.2). This perturbation lets us investigate to what extent TQA systems bypass relevant cells to derive their answers. We observe that 70% of relevant cells contain non-numerical values for WTQ. For TAT, all relevant cells contain numerical values.

Remove table. TQA systems may bypass the whole table and use their internal knowledge to answer a question. To test for this behavior, for any RQs, we replace the table with a dummy table, consisting of one cell with a "None" value.

Shift relevant rows. This perturbation evaluates to what extent TQA systems bypass table cell values and rely on the position of relevant cells. For instance, to answer the question "Who received the least amount of votes?" in Figure 1, a TQA system may exploit a shortcut between the last row and the question since in most cases, rows in tables from TQA datasets are sorted. Because of the correlation between cell values and positions, this type of

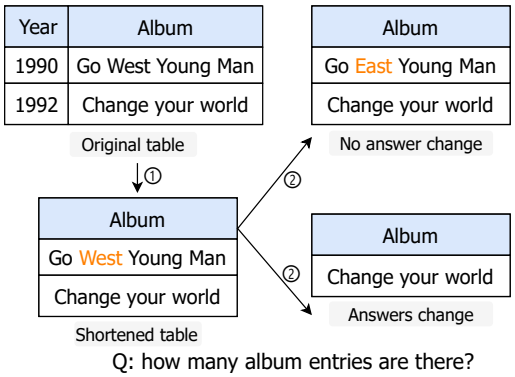

Figure 2: An example of aggregation/comparison robustness in case of string change. ① illustrates shortening an original table to table cells on which numerical aggregations or comparisons operate. ② illustrates modifications on a shortened table, leading either to a change in answer or not. The orange parts mark changed strings.

shortcut mostly occurs for RQs that require comparing cell values. In this perturbation type, for each RQ, we remove relevant rows and re-insert them at a random position of the table.

3.5 Perturbations for Testing Robustness when Aggregating/Comparing Values

TQA systems should compose a correct answer to an RQ by aggregating over or comparing values in the provided table independent of value changes. To evaluate this capability, for RQs, we manually modify cell values that should be aggregated to answer questions. We ask three human annotators to first identify the table cells that numerical aggregations or comparisons operate on (step ① in Figure 2). We refer to the part of the table relevant to answering the question as the *shortened table*. We use the shortened table versions in our experiments to minimize the effect of table length in this test. Then, annotators are asked to change one or two cell values per question, once resulting in an answer change and once not resulting in an answer change (step ② in Figure 2). We define two types of perturbations.

Modify values to change answers. This perturbation allows us to study to what extent TQA systems perform correct aggregations and adapt answers to the value changes accordingly. An example of numeric value changes is shown in Figure 1. The votes for *Olsson* is changed from *4* to *361*, resulting in a change of the answer (from *Olsson* to *Leslie*). Figure 2 shows an example of string value changes. The first entry *Go West Young Man* is removed, changing the answer from *2* to *1*.

Aspect	#P	#Q	#QT	#R	#C	#A
RR-TSC	64890	6489	10.05	12.14	7.18	1.02
Rel-Cel	6303	2101	14.75	21.44	5.77	1.05
ACR-VC	4012	2006	14.62	21.70	5.61	1.05
Total	75205	8590	10.28	14.42	6.85	1.03

Table 3: Benchmark statistics grouped by robustness aspects: retrieval robustness in case of table structure changes (RR-TSC), attention to relevant cells (Rel-Cel), and aggregation/comparison robustness in case of value changes (ACR-VC). #P and #Q show the number of perturbations and questions, respectively. # QT, #R, #C, and #A show the average question length in tokens, the number of table rows, the number of table columns and the number of cells for composing an answer.

Modify values without changing answers. This perturbation aims to study if systems articulate correct answers because of their biases to certain values. For instance, in Figure 1, systems might be capable of comparing *15* and *4*. However, if we change the votes for *Leslie* from *15* to *1500*, without changing the answer, systems might face difficulties as *1500* might not fall into the cell values distribution during pre-training. Figure 2 shows an example of string value changes: *West* is changed to *East* without changing the answer (2).

We filter out instances that annotators find unanswerable. In the case of TAT, all modified values are numeric. In the case of WTQ, 50% of the modified cell values are numeric, the others are string-based.

To assess the quality of the output of these perturbations, for each perturbation type, we randomly sample 50 instances created using this type from each dataset, resulting in 200 instances in total. Then, we ask two annotators who were not involved in the perturbation creation to provide the answers given the changed tables and questions. We compute the exact match accuracy of answers provided by the two annotators, which amount to 92.5% and 93.5%. We find that most wrong cases are related to questions asking for percentage changes: here, annotators sometimes neglected to add the minus symbol to negative percentage changes.

4 Experimental Settings

We use our benchmark to evaluate the robustness of state-of-the-art TQA systems with regard to our proposed aspects. Table 3 shows the main statistics of our benchmark grouped by the robustness aspects for which TQA systems are evaluated.

4.1 Examined TQA Systems

We analyze the robustness of three types of systems: end-to-end systems that are fine-tuned for the TQA task; pipeline systems that generate relevant cells or SQL queries which are then executed on a table, and off-the-shelf LLMs. In particular, we compare the following TQA systems. **TAPEX** (Liu et al., 2021) is an end-to-end TQA system based on BART (Lewis et al., 2019). **OmniTab** (Jiang et al., 2022b) further fine-tunes TAPEX on both more natural and synthetic data. **TaPas** (Herzig et al., 2020) first predicts relevant cells and an aggregation function, backboned by BERT (Devlin et al., 2019). Then, it articulates answers based on outputs from the previous step with a numeric tool. We categorize it as a pipeline model as answers are not directly generated. **Binder** (Cheng et al., 2022) is a pipeline model consisting of a parsing and an executing step. First, intermediate representations (e.g., SQL queries) are generated by GPT-3.5, and then the queries are executed by a program interpreter. **GPT-3.5** is an LLM and the backbone of various TQA models (Ye et al., 2023; Zhang et al., 2023). In the prompt to GPT-3.5, we use a three-shot demonstration to obtain answers (see Appendix A.7). LLaMA (Touvron et al., 2023) is an open-source LLM. We fine-tune its 7b chat version with LoRA (Hu et al., 2021). We describe details of fine-tuning in Appendix A.5.

4.2 Evaluation Metrics

To compare results on our benchmark, we use the following metrics.

Exact match accuracy (Em) checks if the predicted answers and the ground truth are the same. It is a widely used metric for evaluating TQA systems (Pasupat and Liang, 2015; Yang et al., 2022; Jiang et al., 2022b).

Exact match difference (**Emd**, Zhao et al., 2023) measures system performance change before and after perturbations (negative values indicate performances drop). Both Em and Emd focus on overall system performance.

Variation percentage (**VP**, Yang et al., 2022) measures to what extent predictions change before and after performing perturbations from an instance-level perspective. It is defined as follows:

$$\mathrm{VP} = \frac{\mathrm{C2W} + \mathrm{W2C}}{\mathrm{N}} \quad (1)$$

where C2W counts the number of instances whose predictions change from correct to wrong and W2C is the number of instances whose predictions change from wrong to correct. N is the total number of instances. For perturbations involving randomness, we report the mean and standard deviation of scores over five runs with different random seeds.

5 Experimental Results

In this section, we discuss the performance of TQA systems on our benchmark, and provide a detailed analysis of these models for each robustness aspect.

5.1 Retrieval robustness in case of table structure changes

To study this aspect of robustness, we use the first part of our benchmark (see Section 3.3) which consists of shuffling rows (columns), shifting target rows (columns), and transposing tables for EQs. To rule out the effect of different maximum input lengths of the models in this test, we use only instances that are within the maximum input length (512) of TaPas, the model accepting the smallest input. This means we make use of 91% of the EQs.

Figure 3 and Figure 4 show Emd and VP scores for row and column shuffling, respectively, averaged across all source datasets. We report the detailed results for each dataset in Appendix A.6. In addition, the first column of Table 4 provides the model performance in terms of Em on the original data (without perturbations).

Figure 3 shows that the performance of almost all systems drops when evaluating them on row (column) shuffling, which is consistent with prior results (Zhao et al., 2023). However, our fine-grained benchmark enables us to draw more conclusions beyond this general observation. First, systems are more vulnerable to column shuffling than to row shuffling. This is apparent for LLMs (GPT-3.5 and LLaMA2). When comparing model types, LLMs are most affected by row and column shuffling, followed by pipeline systems. The end-to-end TQA systems (TAPEX and OmniTab) are more robust in this regard. Second, regarding row perturbations, systems are highly impacted by the position of the target row. The more it is moved down the table (TM and TB), the more the performance drops. Notably, moving the target row to the top (TT) even leads to performance improvements for some of the systems, confirming that the systems have encoded some positional biases. It further reveals that systems are likely to fail on tables with many rows, where the target row can be

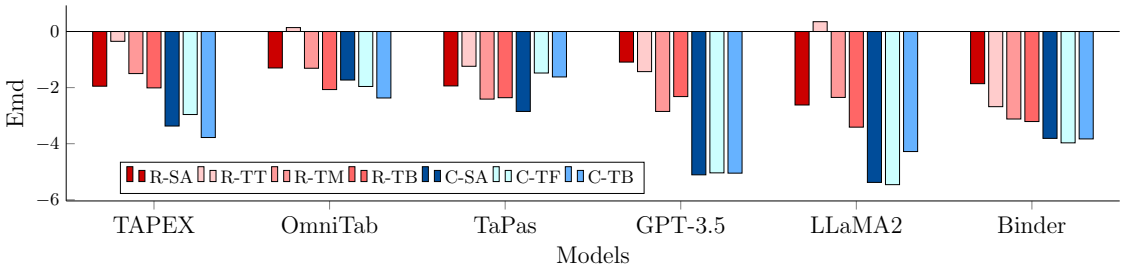

Figure 3: Exact match difference (Emd) on retrieval robustness against table structure changes perturbations **for extraction questions**, averaged across four datasets and seeds. R, C and stand for row and column. SA, TT,TM, TB and TF stand for shuffle all, target top, target middle, target bottom/back, target front, respectively.

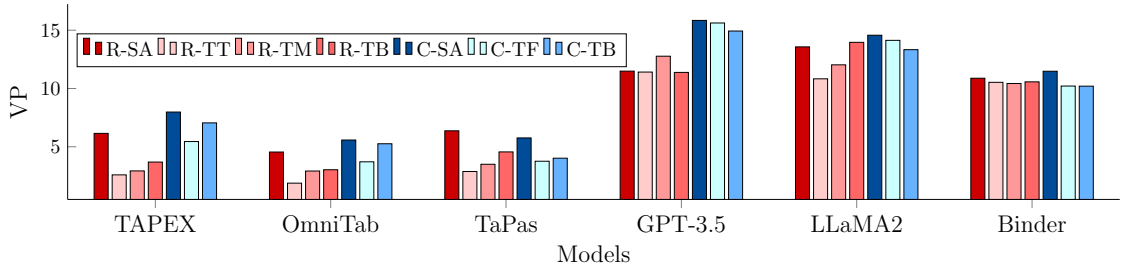

Figure 4: Variation percentage (VP) on retrieval robustness against table structure changes perturbations **for extraction questions**, averaged across four datasets and seeds. R, C and stand for row and column. SA, TT,TM, TB and TF stand for shuffle all, target top, target middle, target bottom/back, target front, respectively.

Model	Original Data	Emd	VP	Emd-Ft
TAPEX	79.83	-55.36	59.64	-43.11
OmniTab	**82.24**	-57.84	60.21	-41.71
TaPas	71.82	-60.98	65.59	-46.83
Binder	67.55	-51.86	62.70	-
GPT-3.5	71.77	**-17.32**	**26.15**	-
LLaMA2	61.10	-30.58	38.42	-

Table 4: Exact match difference (Emd) and variation percentage (VP) for **table transposing**. Emd-Ft stands for Emd after fine-tuned with transposed tables. Bold values suggest best performances.

located anywhere. Similar observations are found by Lin et al. (2023). However, we show that even within the maximum input window of a model, the positional bias exists during cell retrieval.

Third, Table 4 shows that systems fail on transposed tables as well (see columns Emd and VP). Only GPT-3.5 suffers less. We assume that this model has seen a larger variety of tables during its pre-training. Since our analysis of table transposing is new and especially the fine-tuned end-to-end TQA systems perform considerably worse on that perturbed data, we look into this aspect in more detail and investigate their ability to adapt to new table structures (here: transposed tables). For this, we further fine-tune them on transposed table data. The last column of Table 4 shows the results. By fine-tuning on transposed data, the performance of these models improves. However, the gap to GPT-3.5 is still large, showing TQA systems might need architectural changes or pre-training datasets that feature more diverse table structures.

5.2 Attention to relevant cells

To investigate if systems use relevant cell values to compose answers or rather use shortcuts, such as implicit knowledge or biases, we use the second part of our benchmark (see Section 3.4). To the best of our knowledge, this aspect has never been tested in prior work on TQA robustness. We split our analysis into two parts. First, we investigate system behavior when removing relevant information (relevant cells or the whole table). Second, we examine to what extent systems exploit linkages between questions and relevant cell positions rather than paying attention to cell content.

Table 5 shows the Em results for the first part, analysing the behavior of systems when removing relevant information. The column ORI shows the original performance of systems, the column RRel shows the performance when removing relevant cells and the column RT shows results when removing the whole table and replacing it with a

Model	WTQ			TAT		
	ORI	RT	RRel	ORI	RT	RRel
TAPEX*	52.26	8.84	9.57	9.65	3.53	5.38
OmniTab*	**58.12**	7.76	12.55	10.02	3.90	6.12
TaPas**	47.02	**2.80**	**8.21**	-	-	-
Binder**	45.67	6.86	11.62	**37.65**	**0**	**1.01**
GPT-3.5	42.06	6.95	16.52	36.17	**0**	1.73
LLaMA2	35.92	7.44	11.91	-	-	-

Table 5: The exact match (Em) of examined TQA systems regarding attention to relevant cells. ORI shows results on original tables. RT and RRel show the results when the removing table perturbation and removing relevant cells are used, respectively. We do not report results for TaPas and LLaMA2 on perturbations for the TAT subset as their ORI accuracy scores are too low (<4%) to draw conclusions from them.

dummy table. The results indicate that TaPas pays most attention to the cell values, i.e., its Em score is the lowest among the systems when removing the relevant information, which is the desired behavior.

In general, the pipeline models are more robust against changes in the position of the relevant cells than end-to-end models. This intuitively makes sense as they consist of an executing component that performs a function or query directly on the structured table. For a deeper investigation, we analyze instances which the end-to-end systems TAPEX and OmniTab still predict correctly after removing relevant information in tables for TAT. All of them feature the same answer "2019", indicating that answer values are not well distributed in the TQA benchmarks and systems learn this bias during fine-tuning.

For investigating the second question, i.e., to what extent systems exploit positional biases when answering questions, we analyze the variation percentage (VP) with and without shuffling relevant rows or columns. To account for the effect that shuffling data leads to challenges on its own (the first robustness aspect we analyzed before), we compare VP for comparison questions (involving cues, such as "least") with VP for non-comparison questions. Table 6 shows the results. For all systems, the gaps are larger than zero, indicating that they all use question-related shortcuts. Among them, TaPas features the largest gap, suggesting that among the tested systems, it exploits the most question-related shortcuts.

Model	Compare	Non_compare	Gap
TAPEX	10.24	9.70	0.54
OmniTab	7.37	6.02	1.35
TaPas	10.98	**4.04**	6.94
Binder	**9.47**	9.07	0.40
GPT-3.5	16.62	16.40	**0.22**
LLaMA2	17.11	12.89	4.22

Table 6: VP of instances (not) requiring value comparisons on WTQ. **Compare** refers to instances requiring comparison of cells to be solved (e.g., questions contain "highest", "lowest") and **Non_compare** refers to instances do not require comparisons. **Gap** refers to the gap between the two settings.

5.3 Aggregation/comparison robustness in case of value changes

We investigate the numerical reasoning abilities of TQA systems, i.e., their robustness against value changes. As many tables contain numeric data and questions might require the aggregation of several numeric values, this aspect is of utmost importance for real-world applications. While numerical reasoning abilities of models have been analyzed for other domains or tasks (Stolfo et al., 2022; Akhtar et al., 2023), a targeted analysis for TQA is missing in prior work. We analyze this aspect using the third part of our benchmark (Section 3.5). Table 7 shows the Em results for tables with original values (ORI) and shortened tables (ST) that contain only the part of the cell necessary for composing the answer. Additionally, the VP between the results before modifying the cell values and those after changing values is provided, once for the case where answers change (AC) and once where they do not (NC). For shortened tables, performance increases compared to original tables for all systems, again indicating that systems might perform a decent job on tables with a few rows but strongly fail on tables with many rows. When changing values, the pipeline systems TaPas and Binder are among the most robust for both AC and NC settings. We account this to the fact that they execute a predicted function or query on the table and, thus, directly involve the cell values when deriving the final answer. End-to-end systems (TAPEX and OmniTab) show small VP on TAT. However, they do not outperform LLMs, e.g., GPT-3.5.

6 Discussion

In our experiments, we use our new FREB-TQA benchmark to analyze end-to-end, pipeline, and LLM-based TQA systems. Our experimental re-

Model	WTQ				TAT			
	ORI	ST	AC-VP	NC-VP	ORI	ST	AC-VP	NC-VP
TAPEX	47.55	48.03	30.00	5.65	9.68	10.17	8.38	2.23
OmniTab	**52.79**	**54.69**	26.46	4.76	10.06	12.96	**6.33**	**2.05**
TaPas	41.09	48.57	16.39	**3.81**	-	-	-	-
Binder	42.99	50.88	**14.56**	7.96	**37.78**	**41.19**	9.50	8.19
GPT-3.5	38.91	51.70	23.20	8.78	36.30	46.37	26.82	14.71
LLAMA2	32.59	33.88	32.45	6.31	-	-	-	-

Table 7: System evaluation on aggregation/comparison robustness in case of value changes. ORI stands for original performance without perturbations. ST stands for passing short tables on which numerical aggregations operate. We report Em for the ORI and ST settings. AC-VP and NC-VP stand for variation percentage for the answers change and not change settings. We do not report results for TaPas and LLaMA2 on the TAT subset, as the performances are too low (<4%) to derive reasonable analysis from. Bold values suggest best performances.

sults show that all examined systems suffer from substantial issues when it comes to robustness. However, different system types show different patterns. **End-to-end TQA systems, for instance, seem to be more robust against changes in the row/column arrangement. However, they are more likely to fail on numerical reasoning questions.** This might be because the datasets these models are pre-trained on do not feature complex questions requiring numerical reasoning.

LLMs are more affected by row or column perturbations but much more robust against transposing tables. Their performance in performing aggregations or comparisons is highly dependent on the length of the serialized tables, i.e., they perform much better on tables that can be serialized to fewer tokens.

Finally, the pipeline models in our study are more robust against changes in relevant cells, including value changes. This is likely due to their symbolic execution component, which executes a predicted function or query on the given structured table. However, the prediction of the function or query itself still suffers from various perturbations of table data, including changes in table structure. Yet, another benefit of pipeline TQA systems is that the intermediate representation, i.e., the predicted function or query, makes the model explainable to a certain extent. We hence argue that **more research in the pipeline-based paradigm is a promising step towards more robust TQA.**

7 Conclusion and Outlook

In this paper, we have proposed to evaluate three aspects of robustness of TQA systems: retrieval robustness in case of table structure changes, attention to relevant cells, and aggregation/comparison robustness in case of value changes. We have presented a novel benchmark for a fine-grained analysis of those aspects. The main building blocks of our benchmark are targeted table perturbation methods and high-quality human annotations. Finally, we have evaluated a range of architecturally varied state-of-the-art TQA systems, as well as off-the-shelf LLMs. Our study has shown that while none of the systems was consistently robust, their weaknesses and strengths differ from each other. Systems trained in an end-to-end fashion are able to deal with changes in row/column arrangement but LLMs perform better when it comes to numeric operations. The answers of pipeline-based models, which use symbolic methods for part of their computations, are more faithful towards the table they are supposed to use when composing their output.

Our new benchmark constitutes an important first step for evaluating TQA systems in a fine-grained way, thereby directing research efforts towards building more robust TQA systems. As pipeline systems offer at least some explainability, we argue that research should concentrate on these types of systems. In particular, future work could explore the possibility of utilizing LLMs in a pipeline manner to build more robust TQA systems. Further, aggregating or extracting information from long tables is an important next step given the shortcomings of current systems in this regard which our analysis revealed.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback. We also thank Tianxing Liu, Xinyue Shi and Yidan Chen for their annotations.

Limitations

This work focuses on building a benchmark for analyzing the robustness of TQA systems in three fine-grained aspects. In terms of the reasoning type, we mainly discuss numerical reasoning. However, other types of reasoning, e.g., commonsense reasoning or temporal reasoning can also occur during the aggregation phase of TQA. Future studies can explore these aspects and extend our benchmark. We build our benchmark with English TQA datasets, this could also be extended by incorporating TQA datasets in other languages. Additionally, though we report statistics about percentages of numeric and non-numeric values when removing relevant cells, our benchmark does not distinguish these two value types explicitly. Future work could extend our benchmark on this aspect to explore how non-numeric value changes affect model performance.

Ethical Considerations

The development sets of the source datasets we use: WTQ (Pasupat and Liang, 2015), WikiSQL (Zhong et al., 2017), SQA (Iyyer et al., 2017) and TAT (Zhu et al., 2021) are publicly available under the licenses of CC-BY-SA-4.0[5], BSD-3 CLAUSE[6], MIT[7] and MIT, respectively. These licenses all permit us to compose, modify, publish, and distribute additional annotations upon the original dataset. Experiments in this paper are run on a single NVIDIA Tesla V100-32G GPU. Benchmark and code will be released along with the paper. For annotation tasks where humans are involved, we recruit 5 undergraduate students (3 females and 2 males) studying Linguistics in China. All 5 annotators voluntarily participate in the annotation tasks. Three out of four of our annotation tasks requires less than 2 hours to finish. The other one took two weeks and we suggest annotators to spend less than three hours per day to ensure enough rest.

References

Mubashara Akhtar, Abhilash Shankarampeta, Vivek Gupta, Arpit Patil, Oana Cocarascu, and Elena Simperl. 2023. Exploring the numerical reasoning capabilities of language models: A comprehensive analysis on tabular data. *ArXiv*, abs/2311.02216.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, R.K. Nadkarni, Yushi Hu, Caiming Xiong, Dragomir R. Radev, Marilyn Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2022. Binding language models in symbolic languages. *ArXiv*, abs/2210.02875.

Harsh Desai, Pratik Kayal, and Mayank Kumar Singh. 2021. Tablex: A benchmark dataset for structure and content information extraction from scientific tables. In *IEEE International Conference on Document Analysis and Recognition*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Michael R. Glass, Mustafa Canim, A. Gliozzo, Saneem A. Chemmengath, Rishav Chakravarti, Avirup Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. In *North American Chapter of the Association for Computational Linguistics*.

Vivek Gupta, Riyaz A. Bhat, Atreya Ghosal, Manish Shrivastava, Maneesh Singh, and Vivek Srikumar. 2022a. Is my model using the right evidence? systematic probes for examining evidence-based tabular reasoning. *Transactions of the Association for Computational Linguistics*, 10:659–679.

Vivek Gupta, Shuo Zhang, Alakananda Vempala, Yujie He, Temma Choji, and Vivek Srikumar. 2022b. Right for the right reason: Evidence extraction for trustworthy tabular reasoning. In *Annual Meeting of the Association for Computational Linguistics*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Annual Meeting of the Association for Computational Linguistics*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Mohit Iyyer, Wen tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022a. 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*

[5]https://creativecommons.org/licenses/by-sa/4.0/
[6]https://opensource.org/license/bsd-3-clause/
[7]https://opensource.org/license/mit/

the Association for Computational Linguistics: Human Language Technologies, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022b. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *North American Chapter of the Association for Computational Linguistics*.

J Richard Landis and Gary G Koch. 1977. An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, pages 363–374.

Fangyu Lei, Xiang Lorraine Li, Yifan Wei, Shizhu He, Yiming Huang, Jun Zhao, and Kang Liu. 2023. S3hqa: A three-stage approach for multi-hop text-table hybrid question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Weixian Lei, Difei Gao, Yuxuan Wang, Dongxing Mao, Zihan Liang, Lingmin Ran, and Mike Zheng Shou. 2022. AssistSR: Task-oriented video segment retrieval for personal AI assistant. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 319–338, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adrià de Gispert, and Gonzalo Iglesias. 2023. An inner table retriever for robust table question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-Guang Lou. 2021. Tapex: Table pre-training via learning a neural sql executor. *ArXiv*, abs/2107.07653.

Ansong Ni, Srini Iyer, Dragomir R. Radev, Ves Stoyanov, Wen tau Yih, Sida I. Wang, and Xi Victoria Lin. 2023. Lever: Learning to verify language-to-code generation with execution. *ArXiv*, abs/2302.08468.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics*.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Scholkopf, and Mrinmaya Sachan. 2022. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Annual Meeting of the Association for Computational Linguistics*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. Tableformer: Robust transformer modeling for table-text encoding. In *Annual Meeting of the Association for Computational Linguistics*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M. Patel. 2023. Reactable: Enhancing react for table question answering. *ArXiv*, abs/2310.00815.

Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir R. Radev. 2023. Robut: A systematic study of table qa robustness against human-annotated adversarial perturbations. In *Annual Meeting of the Association for Computational Linguistics*.

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

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *ArXiv*, abs/2105.07624.

A Appendix

A.1 Source Dataset Overview

Name	Domain	Feature	#Questions	#Table
WTQ	Wikipedia	complex QA	2831	346
WikiSQL	Wikipedia	simple QA	8418	2628
SQA	Wikipedia	conversational QA	2265	148
TAT	Finance	complex QA	1668	278

Table 8: Overview of development set of source data.

A.2 Human Annotations

Three annotators, all undergraduate students studying Linguistics in China, were involved in the annotation process.

To distinguish between extraction and reasoning questions. They are told to classify whether a question is of type extraction or reasoning after giving the definitions and examples of each question type. The instructions are listed as follow:

- Thank you for participating in this annotation task! In this task, you will be given a pair of question, answer and table and to determine if a question is extraction-based [1] or reasoning-based [0], or neither of them [-1]. We define extraction-based as: answers of the questions can be retrieved from table without numerical reasoning abilities. For instance, for the question: what is the sale number of 2018, given the rows showing the year and the column showing the sale number, the answer can be obtained by looking at certain cells without needing operations on numerical levels. We define reasoning-based as: answers of the questions should be firstly retrieved from table and then numerical reasoning abilities are needed to obtain the answer. For instance, for the question: how many countries have received 4 goals. Entries of countries receiving 4 goals should be retrieved, then, counting is needed to obtain the answer. Common numerical operations are: counting, comparing (date or number), summing, averaging, subtracting, etc. If you find the question unanswerable given the table information, please put a [-1] in the annotation field.

Agreement on the binary question type classification of the 200 questions amounts to a κ-score (Fleiss, 1971) of 0.85, which shows good agreement among the annotators (Landis and Koch, 1977).

To test the validity of relevant cells annotations. We randomly sample 100 annotations for WTQ and ask the same 3 annotators to decide if removing the relevant cells prevents one from answering the questions or not. The instruction is listed as follows:

- Thank you to participate in this annotation task! In this task, you will be given a pair of question, answer, table and relevant cells from the table to determine if removing the relevant cells prevent one from answering the question (and deriving the same correct answers) or not. If you find removing the relevant cells prevent one from answering the question, please put [1] in the annotation field. Otherwise, please put [0]. If you find a question unanswerable or strange, please put [-1] in the annotation field.

The Fleiss' κ-score is 0.53, which shows moderate agreement.

A.3 Eliminating Questions Related to Table Structures

We collect common positional prepositions (suggesting positions) and ordinal numbers (suggesting order) in English. The list we use is as follows: ['first', 'second','third', 'last', 'top', 'bottom', 'before', 'previous', 'latter', 'after', 'next', 'below', 'above']. We eliminate questions containing words in the list.

A.4 LLMs prompt

```
You are a helpful, respectful and honest assistant and always follow the instructions.You are given a table and a question asking information from the table. To answer the question, first retrieve relevant cells from the table. If you need to count, compare, sum, average, or apply other operations to derive the answers from the retrieved cells, return the operations in a list: [operation:]. If answers are in the retrieved cells, return [retrieved]. For instance, for the question 'How many countries won 4 medals', you retrieve the countries won 4 medals and count the number of retrieved countries. So you return [operation: count]. For the question 'What is the total number of sales from 2015-2018?', you need to first retrieve the sales of the year 2015, 2016 and 2018, then sum the retrieved values. So you return [operation:sum].
Question:```question```
Table:```table```
```

Figure 5: LLaMA2 Prompt for classifying extraction and reasoning questions.

A.5 Training/Prompting Details

Similar as in Zhao et al. (2023), we fine-tuned the Large version of pre-trained end-to-end TQA systems for 20 epochs and use the random-split-1-train if multiple train-dev splits are provided. For TAT, the training epochs is increased to 50 epochs. We then split the training set further into training and development sets with the ratio of 80:20. The batch size we use are 32 and gradient accumulation is 4. The best model is selected based on validation loss. As for fine-tuning LLaMA2, we use the 7B chat-hf version. We use the code provided in LLaMA-Factory[8] and keeps the default settings of all parameters in the code. In terms of GPT-3.5, we use three-shots demonstrations. The demonstrations can be found in Appendix A.7. As for Binder, since the systems used in the original paper (Codex) is no longer provided by OpenAI, we use GPT-3.5 as the backbone. The performance might degrade due to the switch, as what is observed in Zhang et al. (2023).

A.6 Results for Robustness Against Table Structural Change for Different Datasources

The following four tables show the exact match difference and variation percentage for each subset on retrieval robustness against table structure changes perturbations.

A.7 LLM prompts for GPT3.5

Figure 6, Figure 7, Figure 8 and Figure 9 shows the prompts we used for prompting GPT-3.5 for WTQ, WikiSQL, SQA and TAT, respetively.

[8]https://github.com/hiyouga/LLaMA-Factory

Ps	Exact match difference						Variation percentage					
	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder
NP	84	84.57	69.71	77.71	63.43	72.57	0	0	0	0	0	0
R-SA	-3.09 ± 0.46	-1.03 ± 0.98	-0.91 ± 1.06	-0.57±1.20	-3.43 ± 2.20	-2.86 ± 1.30	7.43 ± 1.14	4.46±0.98	7.54 ± 1.11	11.31 ± 1.42	16.46 ± 2.03	9.03 ± 1.51
R-TT	-2.17 ± 1.17	0.00±0.36	0.57 ± 0.89	-0.69 ± 1.71	1.26 ± 1.27	-3.89 ± 1.75	3.31 ± 0.23	2.51±0.58	4.46 ± 0.23	12.80 ± 1.83	10.40 ± 0.91	11.43 ± 2.96
R-TM	-1.03 ± 0.56	0.00±0.81	-1.71 ± 0.36	-1.49 ± 0.78	-2.40 ± 0.56	-3.54 ± 2.21	1.71±0.00	3.43 ± 0.36	3.54 ± 1.11	11.09 ± 1.78	9.71 ± 0.72	10.86 ± 2.91
R-TB	-2.63 ± 0.69	-1.03 ± 0.84	-0.69 ± 0.43	-0.46±1.37	-2.97 ± 1.75	-3.43 ± 1.20	4.23 ± 1.43	3.09±0.28	4.57 ± 0.36	10.51 ± 0.86	14.40 ± 0.43	10.06 ± 2.19
C-SA	-4.91 ± 0.93	-0.91±1.28	-2.51 ± 1.18	-7.31 ± 1.51	-6.74 ± 2.92	-4.11 ± 1.75	11.09 ± 2.30	5.94±0.78	6.17 ± 0.84	18.51 ± 1.93	11.09 ± 2.30	10.97 ± 1.42
C-TF	-3.77 ± 0.46	-1.03±0.56	-1.37 ± 0.28	-5.60 ± 2.21	-5.46 ± 3.85	-2.86 ± 1.20	6.51 ± 1.00	3.77±1.12	4.34 ± 0.28	16.11 ± 2.79	15.77 ± 2.57	10.86 ± 2.48
C-TB	-6.97 ± 2.71	-1.26±0.56	-2.40 ± 0.91	-6.17 ± 1.11	-4.80 ± 1.47	-2.97 ± 2.41	9.49 ± 2.44	5.83 ± 0.56	4.46±1.37	16.23 ± 2.00	12.91 ± 2.00	10.97 ± 1.93
Tr	-65.14	-64.57	-54.86	-18.86	-34.29	-70.86	68.57	70.29	61.71	25.71	42.29	70.86

Table 9: WTQ: system performance on extraction data set. Ps stands for perturbations. NP stands for no perturbation. R, C and Tr stand for row, column and transpose perturbations. SA, TT, TF, TM and TB stand for shuffle all, target top, target front, target middle, target bottom/back respectively. We shade the best performances (minimal absolute values) with blue.

Ps	Exact match difference						Variation percentage					
	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder
NP	89.27	91.33	92.59	63.06	56.45	66.70	0	0	0	0	0	0
R-SA	-0.08 ± 0.04	0 ± 0.06	0.08 ± 0.04	-0.27 ± 0.13	-1.82 ± 0.22	-1.14 ± 0.11	1.08 ± 0.06	0.63 ± 0.02	0.63 ± 0.04	10.37 ± 1.11	12.26 ± 0.88	9.21 ± 0.63
R-TT	0.02 ± 0.01	0.04 ± 0.02	0.03 ± 0.03	0.12 ± 0.07	1.01 ± 0.52	-0.93 ± 0.05	0.57 ± 0.06	0.30 ± 0.01	0.42 ± 0.02	11.50 ± 0.56	13.75 ± 1.21	10.77 ± 0.94
R-TM	-0.08 ± 0.04	0.11 ± 0.01	0.04 ± 0.02	-1.04 ± 0.11	-1.24 ± 0.44	-1.23 ± 0.05	0.57 ± 0.06	0.30 ± 0.02	0.44 ± 0.02	13.61 ± 0.11	14.4 ± 0.42	12.84 ± 0.47
R-TB	-0.12 ± 0.03	-0.06 ± 0.04	-0.08 ± 0.03	-0.22 ± 0.13	-1.88 ± 0.79	-1.15 ± 0.07	0.68 ± 0.05	0.43 ± 0.02	0.51 ± 0.08	12.24 ± 0.77	14.61 ± 0.68	10.45 ± 0.79
C-SA	-0.45 ± 0.12	-0.12 ± 0.04	-0.06 ± 0.06	-2.22 ± 0.28	-1.53 ± 0.26	-2.06 ± 0.22	2.35 ± 0.10	1.73 ± 0.12	1.02 ± 0.02	19.09 ± 0.59	15.43 ± 0.90	16.72 ± 0.85
C-TF	-0.60 ± 0.03	0.06 ± 0.01	-0.01 ± 0.01	-3.26 ± 0.37	-4.01 ± 0.52	-3.03 ± 0.22	1.55 ± 0.14	0.79 ± 0.05	0.62 ± 0.07	18.26 ± 0.60	15.40 ± 0.37	13.11 ± 0.79
C-TB	-0.26 ± 0.10	-0.20 ± 0.07	-0.11 ± 0.04	-3.43 ± 0.33	-2.37 ± 0.77	-3.02 ± 0.21	1.17 ± 0.05	0.57 ± 0.09	0.43 ± 0.06	17.14 ± 0.52	15.26 ± 0.58	11.47 ± 0.61
Tr	-76.13	-78.72	-88.13	-15.95	-32.72	-56.07	76.99	79.34	88.17	30.15	40.72	73.69

Table 10: WikiSQL: system performance on extraction data set. Ps stands for perturbations. NP stands for no perturbation. R, C and Tr stand for row, column and transpose perturbations. SA, TT, TF, TM and TB stand for shuffle all, target top, target front, target middle, target bottom/back respectively. We shade the best performances (minimal absolute values) with blue.

Ps	Exact match difference						Variation percentage					
	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder
NP	67.39	70.29	68.84	71.74	57.25	63.89	0	0	0	0	0	0
R-SA	-1.21 ± 1.46	-1.45 ± 0.57	-2.90 ± 1.18	-1.45 ± 0.78	-2.17 ± 0.98	-1.39 ± 0.90	5.56 ± 1.18	4.35 ± 1.17	7.25 ± 1.69	8.78 ± 1.40	9.98 ± 1.04	8.54 ± 1.14
R-TT	0.45 ± 1.12	0.24 ± 0.90	-1.45 ± 0.59	-0.72 ± 0.97	-2.17 ± 1.24	-1.43 ± 0.57	3.86 ± 1.37	2.66 ± 0.34	3.38 ± 0.34	10.82 ± 1.87	9.42 ± 1.24	9.43 ± 1.83
R-TM	-3.48 ± 0.49	-3.42 ± 0.34	-5.07 ± 0.59	-5.12 ± 0.88	-3.71 ± 1.53	-4.84 ± 1.01	3.86 ± 0.49	3.86 ± 0.34	5.56 ± 0.34	9.37 ± 1.22	10.22 ± 0.93	9.46 ± 1.27
R-TB	-4.66 ± 0.73	-4.59 ± 0.34	-4.83 ± 0.34	-5.80 ± 0.43	-4.90 ± 1.57	-6.12 ± 0.67	3.42 ± 1.70	4.34 ± 0.97	6.02 ± 0.35	11.39 ± 1.66	13.68 ± 2.54	10.87 ± 1.21
C-SA	-3.14 ± 1.26	-0.97 ± 1.07	-3.14 ± 0.86	-5.21 ± 1.34	-7.65 ± 1.70	-3.79 ± 1.05	12.32 ± 1.57	8.21 ± 1.37	7.97 ± 1.05	14.21 ± 1.87	16.21 ± 1.63	7.69±1.37
C-TF	-2.17 ± 1.02	-2.17 ± 0.96	-0.24 ± 0.31	-6.38 ± 1.54	-7.57 ± 1.39	-5.25 ± 0.77	7.00 ± 1.34	5.17 ± 1.02	5.24 ± 1.71	14.49 ± 1.92	13.14 ± 2.38	6.29 ± 1.51
C-TB	-2.90 ± 1.57	-3.14 ± 0.68	-1.21 ± 0.68	-5.11 ± 1.79	-4.38 ± 1.13	-3.58 ± 1.11	10.14 ± 1.05	7.49 ± 0.68	3.66 ± 0.68	11.62 ± 2.78	12.80 ± 2.25	5.57 ± 1.12
Tr	-51.45	-58.70	-63.04	-21.01	-34.78	-53.48	57.25	60.14	64.49	25.36	39.13	68.24

Table 11: SQA: system performance on extraction data set. Ps stands for perturbations. NP stands for no perturbation. R, C and Tr stand for row, column and transpose perturbations. SA, TT, TF, TM and TB stand for shuffle all, target top, target front, target middle, target bottom/back respectively. We shade the best performances (minimal absolute values) with blue.

Ps	Exact match difference						Variation percentage					
	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder	TAPEX	OmniTab	TaPas	GPT-3.5	LLaMA2	Binder
NP	78.65	82.75	56.14	74.56	67.25	67.02	0	0	0	0	0	0
R-SA	-3.43 ± 0.41	-2.71 ± 2.07	-4.02 ± 0.72	-2.06 ± 1.09	-3.05 ± 1.09	-2.05 ± 1.41	10.53 ± 1.89	8.77 ± 1.24	10.04 ± 1.89	15.50 ± 1.80	15.56 ± 1.80	16.73 ± 1.09
R-TT	0.29 ± 0.41	0.29 ± 0.83	-4.09 ± 0.83	-5.85 ± 2.19	-3.05 ± 2.07	-4.46 ± 2.41	2.63 ± 1.24	2.05 ± 1.65	4.09 ± 0.83	10.53 ± 3.79	10.97 ± 2.89	10.97 ± 1.49
R-TM	-3.39 ± 0.82	-2.72 ± 0.41	-2.88 ± 1.24	-3.74 ± 1.65	-2.85 ± 0.41	-2.88 ± 0.72	5.56 ± 1.65	4.09 ± 1.01	6.14 ± 1.24	9.01 ± 2.56	10.39 ± 1.89	8.56 ± 1.09
R-TB	-3.06 ± 2.73	-2.59 ± 0.34	-3.83 ± 0.34	-3.80 ± 1.43	-3.90 ± 2.57	-3.12 ± 1.67	6.42 ± 2.70	4.27 ± 1.97	7.13 ± 2.35	11.19 ± 2.37	13.68 ± 2.54	9.95 ± 2.20
C-SA	-4.97 ± 2.07	-4.92 ± 2.89	-4.97 ± 2.41	-5.70 ± 2.41	-5.58 ± 2.47	-5.29 ± 1.49	6.14 ± 1.24	6.43 ± 0.41	7.89 ± 1.24	11.71 ± 3.80	12.60 ± 2.07	10.56 ± 2.54
C-TF	-5.31 ± 1.24	-4.68 ± 1.65	-4.31 ± 0.41	-4.91 ± 1.24	-4.80 ± 1.65	-4.75 ± 0.79	6.73 ± 2.07	5.11 ± 1.65	4.82 ± 1.41	13.93 ± 2.24	12.89 ± 2.80	10.85 ± 0.83
C-TB	-4.97 ± 2.07	-4.88 ± 1.03	-4.75±0.82	-5.50 ± 1.09	-5.58 ± 2.41	-5.75 ± 1.24	7.31 ± 1.41	7.27 ± 0.81	6.43 ± 0.83	14.93 ± 3.72	12.34 ± 2.70	12.85 ± 2.41
Tr	-28.71	-29.36	-37.89	-13.45	-20.51	-27.02	35.73	31.05	47.98	23.39	31.53	38.02

Table 12: TAT: system performance on extraction data set. Ps stands for perturbations. NP stands for no perturbation. R, C and Tr stand for row, column and transpose perturbations. SA, TT, TF, TM and TB stand for shuffle all, target top, target front, target middle, target bottom/back respectively. We shade the best performances (minimal absolute values) with blue.

You are given a question and a table. Please answer the question with regards to the table and return the answer in the list format. Please return ONLY the answer as output in a list. In the table, | sperates columns and \n seperates rows. Below are the three examples:

Question: in how many games did the winning team score more than 4 points?
Table:
Home Team | Score | Away Team | Date | Agg
Aberdeen | 7–1 | Hamilton Academical | 11-10-1978 | 8–1
Airdrieonians | 1–2 | Arbroath | 10-10-1978 | 2–3
Ayr United | 1–1 | Falkirk | 11-10-1978 | 3–1
Clydebank | 1–1 | Hibernian | 11-10-1978 | 1–2
Morton | 5–2 | Kilmarnock | 11-10-1978 | 5–4
Montrose | 5–1 | Raith Rovers | 11-10-1978 | 5–4
Motherwell | 1–4 | Celtic | 11-10-1978 | 2–4
St. Mirren | 0–0 | Rangers | 11-10-1978 | 2–3
Answer: 3

Question: at the women's 200 meter individual medley sm10 event at the 2012 summer paralympics, how long did it take aurelie rivard to finish?
Table:
Rank | Lane | Name | Nationality | Time | Notes
 | 4 | Sophie Pascoe | New Zealand | 2:25.65 | WR
 | 5 | Summer Ashley Mortimer | Canada | 2:32.08 |
 | 3 | Zhang Meng | China | 2:33.95 | AS
4 | 6 | Katherine Downie | Australia | 2:34.64 |
5 | 2 | Nina Ryabova | Russia | 2:35.65 |
6 | 8 | Aurelie Rivard | Canada | 2:37.70 |
7 | 7 | Harriet Lee | Great Britain | 2:39.42 |
8 | 1 | Gemma Almond | Great Britain | 2:42.16 |
Answer: 2:37.70

Question: what is the difference in attendance in tie no 1 and 4?
Table:
Tie no | Home team | Score | Away team | Attendance
1 | IFK Västerås (D2) | 3–2 | IK Sleipner (D2) | 875
2 | Kramfors IF (N) | 2–4 (aet) | BK Kenty (D3) | 2,808
3 | Reymersholms IK (D2) | 5–2 | Åtvidabergs FF (D2) | 1,504
4 | Wifsta/Östrands IF (N) | 4–3 | Ludvika FfI (D2) | 974
5 | Råå IF (D3) | 3–2 | Sandvikens IF (D3) | 2,116
6 | Sandvikens AIK (D2) | 2–3 (aet) | Tidaholms GIF (D2) | 822
7 | Karlstads BIK (D2) | 2–5 | IF Friska Viljor (N) | 1,550
8 | Sandviks IK (N) | 2–3 | Surahammars IF (D2) |
Answer: 99
Now please answer this question:
Question:{question}
Table:
{table}
Answer:

Figure 6: Prompt for GPT-3.5 for WTQ.

You are given a question and a table. Please answer the question with regards to the table and return the answer in the list format. Please return ONLY the answer as output in a list. In the table, | sperates columns and \n seperates rows. Below are the three examples:

Question: How many number does Fordham school have?
Table:
Player | No. | Nationality | Position | Years in Toronto | School/Club Team
Patrick O'Bryant | 13 | United States | Center | 2009-10 | Bradley
Jermaine O'Neal | 6 | United States | Forward-Center | 2008-09 | Eau Claire High School
Dan O'Sullivan | 45 | United States | Center | 1995-96 | Fordham
Charles Oakley | 34 | United States | Forward | 1998-2001 | Virginia Union
Hakeem Olajuwon | 34 | Nigeria / United States | Center | 2001-02 | Houston
Answer: 45

Question: How many schools are in Bloomington, IN?
Table:
School | Location | Founded | Affiliation | Enrollment | Team Nickname | Primary conference
Indiana University | Bloomington, IN | 1820 | Public | 40354 | Hoosiers | Big Ten Conference (D-I)
Iowa State University | Ames, IA | 1858 | Public | 27945 | Cyclones | Big 12 Conference (D-I)
Lindenwood University | St. Charles, MO | 1827 | Private/Presbyterian | 11421 | Lions | MIAA (D-II)
Ohio University | Athens, OH | 1804 | Public | 20437 | Bobcats | Mid-American (D-I)
Robert Morris University | Chicago, IL | 1913 | Private/Non-Sectarian | 7277 | Eagles | Chicagoland (NAIA)
Answer: 1

Question: How many votes did Devil in a Hood receive in total?
Table:
Song | Mobiles | Northern Ireland | Northern England | Scotland | Southern England | Wales | Total
"Groovy Chick" | 10 | 3 | 2 | 3 | 2 | 3 | 23
"Clear the Air" | 5 | 5 | 10 | 8 | 3 | 4 | 35
"Devil in a Hood" | 4 | 1 | 3 | 4 | 4 | 1 | 17
"In My Life" | 2 | 6 | 8 | 5 | 5 | 10 | 36
"How Does It Feel" | 8 | 8 | 4 | 10 | 8 | 5 | 43
"The Girl" | 1 | 2 | 1 | 1 | 6 | 2 | 13
"About You" | 3 | 4 | 6 | 6 | 1 | 6 | 26
Answer: 17
Now please answer this question:
Question:{question}
Table:
{table}
Answer:

Figure 7: Prompt for GPT-3.5 for WikiSQL.

You are given a question and a table. Please answer the question with regards to the table and return the answer in the list format. Please return ONLY the answer as output in a list. In the table, | sperates columns and \n seperates rows. Below are the three examples:

Question: how many total deputies does Potosi have?
Table:
Department | Total Deputies | Uninominal Deputies | Plurinominal Deputies | Special Indigenous\nor Campesino Deputies | Senators
La Paz | 29 | 14 | 14 | 1 | 4
Santa Cruz | 28 | 14 | 13 | 1 | 4
Cochabamba | 19 | 9 | 9 | 1 | 4
Potosí | 13 | 7 | 6 | 0 | 4
Chuquisaca | 10 | 5 | 5 | 0 | 4
Oruro | 9 | 4 | 4 | 1 | 4
Tarija | 9 | 4 | 4 | 1 | 4
Beni | 8 | 4 | 3 | 1 | 4
Pando | 5 | 2 | 2 | 1 | 4
Total | 130 | 63 | 60 | 7 | 36
Answer: 13

Question: how many passengers arrived in 2011?
Table:
Year | Domestic passengers | International passengers | Total passengers | Change
2006 | 764,831 | 83,115 | 847,946 | +4.6%
2007 | 764,674 | 75,276 | 839,950 | −0.9%
2008 | 709,779 | 92,176 | 801,955 | −4.5%
2009 | 605,534 | 82,424 | 687,958 | −14.3%
2010 | 595,457 | 105,119 | 700,576 | +1.7%
2011 | 850,305 | 123,607 | 973,912 | +39.1%
2012 | 899 854 | 178,679 | 1,078,533 | +10.7%
2013 | 745,178 | 131,902 | 877,080 | -18.7%
Answer: 973,912

Question: how many silver medals did karine ruby win?
Table:
Athlete | Nation | Olympics | Gold | Silver | Bronze | Total
Philipp Schoch | Switzerland (SUI) | 2002–2014 | 2 | 0 | 0 | 2
Shaun White | United States (USA) | 2006–2014 | 2 | 0 | 0 | 2
Seth Wescott | United States (USA) | 2006–2010 | 2 | 0 | 0 | 2
Karine Ruby | France (FRA) | 1998–2002 | 1 | 1 | 0 | 2
Hannah Teter | United States (USA) | 2006–2014 | 1 | 1 | 0 | 2
Ross Powers | United States (USA) | 1998–2002 | 1 | 0 | 1 | 2
Kelly Clark | United States (USA) | 2002–2014 | 1 | 0 | 2 | 3
Danny Kass | United States (USA) | 2002–2006 | 0 | 2 | 0 | 2
Answer: 1
Now please answer this question:
Question:{question}
Table:
{table}
Answer:

Figure 8: Prompt for GPT-3.5 for SQA.

You are given a question and a table. Please answer the question with regards to the table and return the answer in the list format. Please return ONLY the answer as output in a list. In the table, | sperates columns and \n seperates rows. Below are the three examples:

Question: What is the percentage change in net sales from Frozen Kefir between 2018 and 2019?
Table:
| | 2019 | | 2018
In thousands | $ | % | $ | %
Drinkable Kefir other than ProBugs | $ 71,822 | 77% | $ 78,523 | 76%
Cheese | 11,459 | 12% | 11,486 | 11%
Cream and other | 4,228 | 4% | 5,276 | 5%
ProBugs Kefir | 2,780 | 3% | 2,795 | 3%
Other dairy | 1,756 | 2% | 3,836 | 4%
Frozen Kefir (a) | 1,617 | 2% | 1,434 | 1%
Net Sales | $ 93,662 | 100% | $ 103,350 | 100%
Answer: 12.76

Question: How much is the 2019 rate of inflation?
Table :
| 2019 % | 2018 % | 2017 %
Weighted average actuarial assumptions used at 31 March1: | | |
Rate of inflation2 | 2.9 | 2.9 | 3.0
Rate of increase in salaries | 2.7 | 2.7 | 2.6
Discount rate | 2.3 | 2.5 | 2.6
Answer: 2.9

Question: What was the change in raw materials between 2018 and 2019?
Table :
| March 31, |
| 2019 | 2018
Raw materials | $74.5 | $26.0
Work in process | 413.0 | 311.8
Finished goods | 224.2 | 138.4
Total inventories | $711.7 | $476.2
Answer: 48.5
Now please answer this question:
Question:{question}
Table:
{table}
Answer:

Figure 9: Prompt for GPT-3.5 for TAT.