Table QA on HiTab: A Comparative Study of TAPAS and TAPEX

Quang Nhat Truong[1,2*], Truc Tuan Do[1,2], Tai Duc Nguyen[1,2],
Ngan Luu-Thuy Nguyen[1,2],Kiet VanNguyen[1,2], Son Thanh Luu[1,2], Vu Duc Nguyen[1,2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{22521207*, 22521548, 22521277}@gm.uit.edu.vn
{ngannlt, kietnv, sonlt, vund}@uit.edu.vn

Abstract

Hierarchical tabular data in the HiTab Dataset poses significant challenges for Table-to-Text QA models, requiring the ability to handle complex multi-layered structures and perform deep semantic reasoning. In this study, we evaluate the performance of two pre-trained models, *tapas-finetuned-wtq* and *tapex-finetuned-wtq*, on the HiTab Dataset. Additionally, we fine-tune TAPEX to optimize its capability in processing complex tabular data. The results demonstrate that TAPEX, after fine-tuning, outperforms other approaches, establishing a new benchmark for hierarchical tabular data processing. This research sheds light on how these models approach and comprehend hierarchical table structures, maximizing the utility of tabular data in complex QA tasks.
Keywords **— Table-to-Text, Question Answering, Tapas, Tapex**

1 Introduction

Tables are an indispensable component in various fields such as statistics, finance, science, and academic research, playing a crucial role in data storage and presentation. Among them, **hierarchical tables** are commonly found in statistical reports and scientific documents due to their ability to organize multi-layered information. Most existing research primarily focuses on **flat tables**, while hierarchical tables pose unique challenges that require special attention.

Hierarchical tables come with three major challenges:

- **Hierarchical indexing:** Locating a data cell requires leveraging both row and column information, unlike flat tables, which rely on a single dimension.
- **Implicit relationships:** Computational relationships such as totals, ratios, or semantic links between entities are often not explicitly represented, making inference more complex.
- **Hierarchical structure:** Rows and columns in hierarchical tables may contain layered relationships or implicit meanings that are challenging to identify, requiring models to fully comprehend context for accurate processing.

To address these challenges, this study evaluates the performance of two existing Table-to-Text QA models, *tapas-large-finetuned-wtq* and *tapex-large-finetuned-wtq*, on the HiTab Dataset. Furthermore, we fine-tune TAPEX to optimize its ability to process complex tabular data. The HiTab Dataset, characterized by multi-layered questions and tables, not only presents significant challenges for existing baselines but also opens new research directions in processing hierarchical tabular data for QA tasks.

Input and Output of the Table QA

Input:

- **Tabular data:** A hierarchical table with multiple rows and columns containing information to be queried. The table may include numerical values, textual information, or a combination of both.
- **Question:** The question may require either directly querying a specific value or inferring an answer based on multiple sections of the table.

Output:

- **Answer:** A concise and accurate response based on the input question and tabular data. The answer can be a specific value (numerical or textual) or a result derived from computations over the data.

This study provides three key contributions:

1. **Model evaluation:** Analyze and compare the performance of existing Table-to-Text QA

models on the HiTab dataset, identifying their strengths and limitations.

2. **Performance improvement:** Fine-tune TAPEX to optimize its ability to handle hierarchical tables, expanding the applicability of Table-to-Text QA models.

3. **Foundation for future research:** Shed light on how models approach and learn hierarchical table structures, providing a basis for future studies in QA and NLG on tabular data.

The findings of this study not only offer deeper insights into how current models process hierarchical tables but also pave the way for new directions in leveraging complex tabular structures across various application domains. This research emphasizes the importance of developing Table-to-Text QA methods tailored for hierarchical data. **The structure of this paper is organized as follows: Section 2 (Related Work)** provides an overview of previous research in the field of Table QA, introducing the HiTab dataset and the referenced TAPAS and TAPEX models. **Section 3 (Dataset)** details the structure of the HiTab dataset, including its format and key characteristics. **Section 4 (Methodology)** describes the experimental setup, encompassing preprocessing steps, model configuration, data encoding, and evaluation metrics. **Section 5 (Results and Analysis)** presents the experimental outcomes, analyzing the limitations and errors of the TAPEX model when applied to the HiTab dataset. Finally, **Section 6 (Conclusion and Future Directions)** summarizes the main contributions of the study and proposes solutions and directions for future research.

Province	Farm operators			
	Immigrated between 2011 and		Other immigra	Non-immigra
	China	United States		
	percent			
Newfoundland and Labrador	0.0	0.0	0	0.2
Prince Edward Island	0.0	10.7	0.6	0.7
Nova Scotia	0.0	0.0	1.7	1.7
New Brunswick	8.9	0.0	0.9	1.1
Quebec	0.0	4.5	6.6	16.3
Ontario	58.7	13.1	34.6	25.2
Manitoba	0.0	4.6	6.7	7.4
Saskatchewan	8.2	11.2	4.0	17.9
Alberta	0.0	17.0	18.1	21.4
British Columbia	24.3	39.0	26.8	8.1

Figure 1: Hierarchical Tables

2 Related Work

2.1 Table Question Answering

Table Question Answering (QA) has been extensively studied, with numerous works focusing on processing simple flat tabular data. Pioneering studies, such as those by Herzig et al. [1] and Yin et al. [2], introduced Transformer-based methods to integrate tabular data and textual information within an end-to-end framework. Other studies, including Yu et al. [3] and Pasupat & Liang [4], provided foundational solutions for table-based question answering tasks but were primarily limited to handling flat tables and performed poorly on hierarchical tables. Furthermore, natural language generation (NLG) methods from tabular data, such as those by Lebret et al. [5] and Parikh et al. [6], have advanced techniques for generating textual content from tabular data.

2.2 Reference Dataset

The HiTab Dataset, introduced by Cheng et al. [7], is specifically designed to address the challenges posed by hierarchical tables, such as multi-level indexing, implicit relationships, and hierarchical structures. HiTab includes real-world questions paired with hierarchical tables, providing a robust foundation for research in QA and NLG on tabular data. Additionally, the dataset offers detailed annotations of entities and numerical values, significantly enhancing models' reasoning capabilities. Unlike previous datasets that primarily focus on flat tables, HiTab introduces more complex challenges, fostering the development of models capable of leveraging entity and numerical relationships within tabular data.

2.3 Reference Models

TAPAS, developed by Herzig et al. [1], and TAPEX, developed by Liu et al. [8], represent significant advancements in processing tabular data. TAPAS employs a Transformer-based framework to enable end-to-end QA by integrating textual and tabular data without the need for intermediate logical forms. TAPEX, on the other hand, focuses on learning a neural SQL executor to perform complex reasoning between tables and text, demonstrating superior performance in Table-to-Text QA tasks. Despite their notable achievements, both models face challenges in processing hierarchical tables, particularly in reasoning through implicit relationships and multi-level structures.

3 Dataset

3.1 Data Format

The HiTab dataset is organized in JSON format, containing detailed information about questions, answers, and their corresponding tables.
In conjunction with related table files, the dataset offers a diverse and comprehensive foundation for research on tabular data.
For better compatibility with the research, the dataset has been converted into a complete dataframe format. The original tables in JSON format were also transformed into CSV files.

3.2 Data Structure

HiTab Dataframe:
The dataframe contains **10,672 rows** with the following main columns:

- **id**: Identifier for each data sample.
- **table_id**: Identifier for each table.
- **table_source**: Source of the table data.
- **sentence_id** and **sub_sentence_id**: Labels for each sentence and sub-sentence.
- **question**: The question.
- **answer**: The corresponding answer.
- **aggregation**: Type of arithmetic or logical operation used.
- **linked_cells**: Cells linked to the question.
- **answer_formulas**: Mathematical or logical formulas generating the answer.
- **reference_cells_map**: Reference cell coordinates.
- **CSV Tables**: A total of **3,597 tables** with hierarchical structures.

Aggregation:

- A total of **7,536 rows** do not involve any aggregation (NONE).
- **3,136 rows** involve aggregation operations (OTHER THAN NONE), including:
  - **Arithmetic operations**: sum, diff, div, argmax, argmin.
  - **Logical operations**: opposite, pair-argmax, topk-argmax, and other types.

To support the research on Table Question Answering, the following columns were selected as key features:

- **table_id**: Links the question to the corresponding table.
- **question**: Input in the form of a question.
- **answer**: Expected output (ground truth).
- **reference_cells_map**: Coordinates of the answer within the table. Only (x, y) coordinates were used, while Excel-style coordinates were ignored.

4 Methodology

4.1 Preprocessing

During the preprocessing phase, the data is normalized and prepared to ensure compatibility with the input requirements of the TAPEX model. The specific steps include:

- **Table data processing**:
  - **Filling missing values**: Empty cells in the table are filled to prevent errors during encoding.
  - **Adding header rows**: A header row is added to the table to provide sufficient context for the model to understand the content.
- **Answer normalization**:
  - **Value separation**: Values are separated by commas when necessary.
  - **Integer formatting**: Large integers greater than 1,000 are formatted with commas (e.g., $1392.0 \rightarrow 1,392$).
  - **Real number conversion**: Real numbers without decimal places are converted to integers (e.g., $51.0 \rightarrow 51$).

4.2 Baseline Model

4.2.1 TAPAS

TAPAS (Tabular Pretrained Language Model) is a BERT-based architecture designed to perform Question Answering (QA) tasks on tabular data. TAPAS leverages the strong contextual representation capabilities of BERT while incorporating tabular information into its embeddings, enabling

Before	After
[51.0]	51
[8189.0]	8,189
['cameroon', 'cote d'ivoire', 'senegal']	cameroon, cote d'ivoire, senegal
[4.581946]	4.581946
[298.07]	298.07
[77.1, 69.8]	77.1, 69.8

Table 1: Example of answers before and after normalization

the model to understand table structures and the relationships between rows, columns, and cells.

Main Features:

- TAPAS combines tables and questions as input, using row, column, and value embeddings to model relationships within the table.
- The output of TAPAS can be either a textual answer or the coordinates of cells within the table.
- TAPAS is well-suited for basic QA tasks such as information retrieval and performing simple arithmetic operations.

4.2.2 TAPEX

TAPEX (Table Pretrained Language Model with Extrapolation) is a BART-based architecture designed to handle complex queries requiring logic and computations on tabular data. Compared to TAPAS, TAPEX is more optimized for tasks demanding advanced reasoning.

Main Features:

- TAPEX encodes tables and input questions as a sequence, utilizing the structure of BART.
- The output of TAPEX is textual and contains query results, which may include outcomes from arithmetic operations or logical reasoning.
- TAPEX supports complex tasks such as aggregation, data filtering, and logical querying.

4.2.3 Fine-tuned TAPAS & TAPEX on WTQ

WikiTableQuestions (WTQ) is a large and widely-used dataset comprising tables from Wikipedia and corresponding questions, designed to evaluate the reasoning and querying capabilities of Question Answering (QA) models on tabular data. In this study, we employ two models, TAPAS and TAPEX, fine-tuned on WTQ to perform direct inference on the HiTab dataset. This choice is motivated by several main factors:

- **Generalization of QA models:** TAPAS and TAPEX, trained on WTQ (a standard tabular dataset), represent table-based QA approaches. Testing these models on HiTab provides an assessment of their generalization capabilities when handling a distinct dataset.
- **Baseline comparison:** The inference results from the models fine-tuned on WTQ serve as a baseline to compare against the performance of Tapex fine-tuned directly on HiTab, highlighting the importance of fine-tuning on the target dataset.
- **Ease of deployment:** Pre-trained models fine-tuned on WTQ are readily accessible via libraries like Hugging Face, saving resources and time while facilitating research under constrained conditions.
- **Data similarity:** Both WTQ and HiTab include tables from diverse domains such as finance, sports, and economics, requiring the models to retrieve, reason over tabular data, and solve quantitative tasks. This ensures that the models can grasp the semantic relationships between data fields.

Using TAPAS and TAPEX trained on WTQ for HiTab evaluates the generalization capabilities, fine-tuning effectiveness, and practical usability of current QA models when processing distinct datasets.

4.3 Encoding

The encoding process was conducted to ensure that the input data complies with the requirements of the TAPEX model. As TAPEX requires the number of input tokens not to exceed **1024**, the data was processed and encoded following the key steps below:

- **Tokenization for TAPEX**:
  - TAPEX enforces a maximum input token limit of **1024**, thereby only accepting tables with a number of rows less than or equal to **1024**.

  - This filtering resulted in **8649** valid rows.

- **Encoding tables and questions**:
  - Tables and questions were tokenized and encoded into tensors, such as ***input_ids*** and ***attention_mask***, using the tokenizer.
  - Each table and question was encoded into the following tensor shapes:
    * **input_ids**: torch.Size([1, 1024]).
    * **attention_mask**: torch.Size([1, 1024]).
  - This process ensures that the input data adheres to the requirements of the TAPEX model.

- **Label encoding and padding**:
  - Labels were tokenized and encoded into tensors using the tokenizer.
  - Padding was applied to ensure uniform tensor sizes. For example:
    * **Sample 8648**: label shape: torch.Size([3]).
    * **Sample 8649**: label shape: torch.Size([5]).
  - The final size of the label tensors after padding is: **torch.Size([8649, 53])**.

- **Diversity of operations in the dataset after encoding**:
  - **(none)**: 6254 (71.96%).
  - **(pair-argmax)**: 533 (6.13%).
  - **(div)**: 354 (4.07%).
  - **(opposite)**: 290 (3.34%).
  - **(argmax)**: 246 (2.83%).
  - **(sum)**: 189 (2.17%).
  - **(pair-argmin)**: 180 (2.07%).
  - **(diff)**: 161 (1.85%).
  - **(argmin)**: 81 (0.93%).
  - **(max)**: 59 (0.68%).
  - **(topk-argmax)**: 52 (0.60%).
  - **(greater_than)**: 47 (0.54%).
  - **(min)**: 34 (0.39%).
  - and other minor categories.

To execute the encoding and training process effectively, experiments were conducted on the Google Colab platform with the following key configurations:

- **Platform**: Google Colab.
- **Hardware**:
  - GPU: NVIDIA Tesla T4.
  - RAM: 15 GB.
  - Storage: 50 GB (Google Drive).
- **Software**:
  - Python: Version 3.10.
  - Libraries used: TensorFlow, PyTorch, NLTK, scikit-learn, NumPy, Pandas, Transformers (Hugging Face), Regular Expressions (re).

5 Results and Analysis

5.1 Model Configuration

5.1.1 Fine-tuned on WTQ:

TAPEX utilizes the tokenizer and model *microsoft/tapex-large-finetuned-wtq*.
TAPAS employs the tokenizer and model *google/tapas-large-finetuned-wtq*.

Predictions:

- **TAPEX** returns a **numerical value, word, or phrase** from the table.
- **TAPAS** returns a **numerical value, word, or phrase** along with its coordinates from the table.

5.1.2 Self Fine-tuned TAPEX:

Training Configuration:

- **Optimizer:** AdamW with a learning rate of 5×10^{-5}.
- **Batch size:** 8.
- **Number of epochs:** 3.

Training Loop:

- Prepare batches of data from encoded tables and questions.
- Perform a forward pass to compute the loss between predictions and ground-truth labels.
- Conduct a backward pass to update model weights using the AdamW optimizer.

Average Loss Calculation: Compute the average loss during training to monitor performance.

Evaluation on Development Set (Dev): After each epoch, compute the average loss on the development set to assess generalization capabilities and identify potential overfitting.

5.2 Evaluation Metrics

To evaluate the performance of the fine-tuned TAPEX model, we utilized **Cross-Entropy Loss**: This metric was employed during training to optimize the model's parameters. The loss value is calculated based on the Cross-Entropy between the ground-truth answer and the predicted answer. This ensures that the model learns to minimize prediction errors.

For assessing the correctness of predictions across all Table QA models, we used **Accuracy**: Accuracy is calculated as the proportion of correct answers where the model's predictions match the ground-truth labels. An answer is considered correct if the predicted sequence matches the ground-truth sequence exactly, disregarding case sensitivity. The formula for accuracy is as follows:

$$\text{Accuracy} = \frac{\text{Correct}}{\text{Total Samples}} \times 100$$

The accuracy metric provides insight into how effectively the model generates correct answers to questions based on tabular data.

5.3 Results

Method	Accuracy (%)
Tapex (fine-tuned WTQ)	29.04
Tapas (fine-tuned WTQ)	24.50
Tapex (fine-tuned HiTab)	45.07

Table 2: Prediction Results on the Test Dataset (1298 rows).

Based on the results in **Table 2**, the TAPAS and TAPEX models exhibit significant differences in performance on the HiTab dataset. Specifically, it was observed that Tapex (fine-tuned WTQ) achieved an accuracy of **29.04%**, which is higher than TAPAS (fine-tuned WTQ) at **24.50%**. This suggests that Tapex, with its specialized design for tabular data, can better leverage information from tables compared to TAPAS. Notably, when Tapex was fine-tuned directly on HiTab, its performance increased significantly to **45.07%**, markedly higher than both models fine-tuned on WTQ. This result highlights the importance of fine-tuning on target datasets to enhance the adaptability and performance of models, particularly when dealing with distinct datasets.

These findings emphasize that self fine-tuning TAPEX not only leads to substantial performance improvements but also demonstrates that models optimized for the specific characteristics of the target dataset can significantly outperform general-purpose models.

5.4 Error Analysis

Question	Predicted Answer	True Answer
list any cups that queen of the south f.c play for?	league cup, league cup	challenge cup, scottish cup, league cup
what is the decline of the rate of persons charged in criminal incidents in general has declined between 2004 and 2014?	-19.7	19.7
how many percent of adults under correctional supervision in the provinces and territories in 2013/2014 were in custody?	87	0.186393

Table 3: Error Analysis Fine-tuned TAPEX

Detailed Error Analysis

1. Question: ***list any cups that queen of the south f.c play for?***

Error:

- The answer only listed *league cup* twice, omitting *challenge cup* and *scottish cup*.

Cause:

- The training data lacks examples of questions requiring the enumeration of multiple items.

2. Question: ***what is the decline of the rate of persons charged in criminal incidents in general has declined between 2004 and 2014?***

Error:

- The predicted value is *-19.7*, which has the wrong sign compared to the true value *19.7*.

Cause:

- A semantic processing error where the model failed to infer that the answer required the opposite value.

3. Question: ***how many percent of adults under correctional supervision in the provinces and territories in 2013/2014 were in custody?***

Error:

- The model returned a value of *87* instead of the true value *0.186393*.

Cause:

- This is an example where the model struggled to handle complex arithmetic operations.

Proposed Solutions:

- Augment the training data with similar types of questions.
- Implement post-processing mechanisms to eliminate duplicate values in the results.
- Improve handling of arithmetic operations.

6 Conclusion and Future Work

The results of this study demonstrate that the TAPEX model, when fine-tuned on the HiTab dataset, achieved remarkable performance in the task of tabular data processing, with an accuracy of up to **45.07%**. Compared to TAPEX (fine-tuned on WTQ) and TAPAS (fine-tuned on WTQ), the self fine-tuned TAPEX model exhibited superior performance, suggesting that fine-tuning on target datasets can play a crucial role in optimizing the effectiveness of table-querying systems. However, these findings require further validation through in-depth studies and experiments on diverse datasets to better assess the model's generalization capabilities. Additionally, the study highlights that the architecture of TAPEX may be more suitable for tasks requiring quantitative reasoning. Nevertheless, this observation warrants further evaluation and comparison with other models across different contexts.

Despite the promising results, the error analysis reveals several critical issues that need to be resolved:

- **Incomplete and potentially redundant enumeration in answers:** The model occasionally fails to enumerate all the required information or returns duplicate values. This impacts the accuracy and comprehensiveness of the answers, particularly for questions that demand a complete list of items.
- **Misinterpretation of semantics and context:** In some cases, the model misunderstands the requirement to retrieve a value that is the opposite or complementary to the original value in the table.
- **Limited capability in handling complex computations:** Questions that require precise calculations or data conversions remain a significant challenge for the current model. This limitation highlights gaps in the training data or deficiencies in the mathematical reasoning capabilities of TAPEX.

Although the aforementioned errors are noteworthy, they do not diminish the value of TAPEX in addressing tabular data problems. On the contrary, these errors serve as valuable insights, helping to identify specific areas for improvement to enhance the model's performance.

In the near future, efforts to improve TAPEX will focus on the following key directions:

- **Enhancing the training dataset:** Expanding the dataset with more diverse and comprehensive examples, particularly those involving enumeration, trend analysis, or numerical calculations. This will enable the model to learn a broader range of scenarios and improve prediction accuracy.
- **Refining result processing:** Incorporating post-processing steps to eliminate common issues, such as duplicate outputs or misinterpretation of the question requirements, ensuring answers are both complete and accurate.
- **Strengthening computational and semantic reasoning capabilities:** Improving the model's ability to handle numerical data and semantic understanding, enabling it to respond more effectively to questions involving numerical reasoning and trend analysis.

With these directions in place, TAPEX is expected to become more robust and better suited to address complex tabular data challenges and answer intricate queries.

References

[1] Jonathan Herzig, Peter Nowak, Thomas Muller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*

(ACL), pages 4320–4333. Association for Computational Linguistics, 2020.

[2] Pengcheng Yin and Graham Neubig. Tabert: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8413–8426. Association for Computational Linguistics, 2020.

[3] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, Jinfeng Ma, Irene Li, Bo Pang, Tianbao Zhou, et al. 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 (EMNLP)*, pages 3911–3921. Association for Computational Linguistics, 2018.

[4] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1470–1480. Association for Computational Linguistics, 2015.

[5] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1213. Association for Computational Linguistics, 2016.

[6] D. Parikh, N. Tandon, A. Sharma, and S. Jain. Controlled natural language generation for hierarchical tables. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 256–265. Association for Computational Linguistics, 2020.

[7] Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 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 (ACL)*, pages 1473–1484. Association for Computational Linguistics, 2022.

[8] Qian Liu, Yutai Mao, Xiang Zhou, Baichuan Peng, and Qingkai Guo. Tapex: Table pre-trained model for text-to-sql generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5806–5817. Association for Computational Linguistics, 2021.