QuASAR: A Question-Driven Structure-Aware Approach for Table-to-Text Generation

Weijie Liu, Yibin Zheng, Fang Kong*
School of Computer Science and Technology
Soochow University, China
weijieliu.cs@gmail.com, kongfang@suda.edu.cn

Abstract

Table-to-text generation aims to automatically produce natural language descriptions from structured or semi-structured tabular data. Unlike traditional text generation tasks, it requires models to accurately understand and represent table structures. Existing approaches typically process tables by linearizing them or converting them into graph structures. However, these methods either fail to adequately capture the table structure or rely on complex attention mechanisms, limiting their applicability. To tackle these challenges, we propose QuASAR, a question-driven self-supervised approach designed to enhance the model's structural perception and representation capabilities. Specifically, QuASAR formulates a set of structure-related queries for self-supervised training, explicitly guiding the model to capture both local and global table structures. Additionally, we introduce two auxiliary pre-training tasks: a word-to-sentence reconstruction task and a numerical summarization task, which further enhance the fluency and factuality of the generated text. Experimental results on the ToTTo and HiTab datasets demonstrate that our approach produces higher-quality text compared to existing methods. All of our source code and data are publicly available at https://github.com/weijieliu-cs/QuASAR.

1 Introduction

Table-to-text generation is the task of converting structured or semi-structured tables into coherent natural language descriptions. It has broad applications in areas such as sports reporting (Chen and Mooney, 2008), financial summaries (Liang et al., 2009), and medical reports (Nishino et al., 2020). Unlike traditional text generation, this task presents greater challenges due to the complex structure of tabular data (Liu et al., 2018). To generate high-quality descriptive text, models are expected to not only understand table content accurately but also model their structural characteristics effectively.

Early approaches simplify the table-to-text generation by framing it as a keyword-to-text generation problem (Uchimoto et al., 2002). These methods typically involve extracting key information, performing content planning, and then generating descriptions (Puduppully et al., 2019; Ma et al., 2019; Su et al., 2021). A recent work (Kale and Rastogi, 2020) linearizes tables into sequences of `(row, column, value)` triplets, leveraging pre-trained models like T5 (Raffel et al., 2020) to tackle table-to-text generation. However, neither of these methods effectively models the structural information inherent in tables. In response, some graph-based approaches have been proposed (Ke et al., 2021; Wang et al., 2022; Li et al., 2024), where nodes and edges are defined based on cell adjacency and row-column associations. Graph-based methods can more accurately capture the structural characteristics of tables. However, they often require modifications to the attention mechanisms in pre-trained models to better align with the graph structure. This adaptation process could incur high costs and potentially degrade the model's original generation capabilities.

To overcome these limitations, we propose a question-driven self-supervised method. It helps the model better perceive and represent table structures through structured querying. Specifically, we design a set of explicit queries (e.g., "*What is the header of cell A?*" and "*Which cells are in the same row as cell A?*") to help the model learn both local and global structural information. The answers to these queries serve as self-supervised signals, enhancing the model's ability to capture structural relationships between cells. Moreover, our method only modifies the last hidden layer of the encoder. This lightweight design significantly improves structural awareness while preserving the pre-trained model's original generation ability.

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26798–26812

Additionally, to further enhance the quality of the generated text, we introduce two auxiliary pre-training tasks. The first is a *word-to-sentence reconstruction* task that trains the model to expand sparse input into complete and fluent sentences. The second is a *numerical summarization* task designed to help the model summarize and aggregate numerical content effectively. To evaluate the effectiveness of our approach, we conducted extensive experiments on ToTTo (Parikh et al., 2020) and HiTab (Cheng et al., 2022) benchmarks using the T5 model (Raffel et al., 2020). The results show that QuASAR consistently achieves competitive performance on both datasets, demonstrating its effectiveness and generalizability.

Our contributions can be summarized as follows: (i) We propose a question-driven method based on table structure querying, which effectively improves the model's ability to perceive and represent structural information. (ii) We introduce two auxiliary pre-training tasks: a *word-to-sentence reconstruction* task for expanding sparse text into coherent descriptions, and a *numerical summarization* task for aggregating and abstracting numerical content. (iii) Extensive experiments on the ToTTo and HiTab benchmarks demonstrate that our approach significantly improves table-to-text generation quality, providing a simple yet effective solution for this task.

2 Related Work

Table-to-text generation aims to automatically produce natural language descriptions from structured or semi-structured table data. Some early studies (Puduppully et al., 2019; Ma et al., 2019; Su et al., 2021) treat this task as a two-stage process: first, extracting key information from the table and performing content planning; second, generating coherent textual descriptions. Ma et al. (2019) further explores how to construct pseudo-parallel data in low-resource scenarios by focusing on key facts and removing redundant text. However, their approach only adds noise to the core vocabulary without changing its order. In contrast, we randomize the word order, making the task closer to keyword-based sentence generation. More importantly, we argue that using keyword-to-text generation as a pretraining task is not just a workaround for limited parallel data. Instead, it reflects a principled choice: generating dense natural language from sparse input is intrinsic to the task.

Several other approaches have also been proposed for table-to-text generation. Ramamurthy et al. (2022) introduces the Natural Language Policy Optimization (NLPO) algorithm. It reduces the complexity of the action space in generation tasks, thereby improving training stability and learning efficiency. An et al. (2022) utilizes a contrastive framework that generates examples based on predictions. This approach enhances table-to-text generation by incorporating learned similarity during decoding. Liu et al. (2022) pretrains a model on a table-to-logical-form task, using logical forms as intermediaries to improve the faithfulness of logical reasoning in text generation. While these methods yield promising results, they all overlook table structure modeling and thus fail to fully leverage structural information in tables.

Kale and Rastogi (2020) utilizes the pre-trained model T5 (Raffel et al., 2020) to tackle table-to-text generation by linearizing the table into a sequence of `(row, column, value)` triples. Andrejczuk et al. (2022) enhances table encoding by incorporating the row and column features into the cells. However, the former merely transforms the table format, while the latter lacks explicit modeling of structural features. Another typical approach (Ke et al., 2021; Wang et al., 2022; Li et al., 2024) represents the table as a graph and adjusts the attention mechanism in the pre-trained model accordingly. Although this method can effectively capture structural characteristics, it requires substantial modifications to the attention mechanism, which increase adaptation costs and may weaken the model's original generation ability. In contrast, our method only modifies the last hidden layer in the encoder, without altering the native attention mechanism. This minimizes the impact on the model's original generation capabilities.

Xing and Wan (2021) enhances the model's perception of table structures by predicting the adjacent cells (left, right, top, and bottom) of a given cell. Alonso et al. (2024) incorporates visual models and leverages two-dimensional image features. It trains the model to capture structural alignment by predicting cells in the same row and column as a given cell. These approaches aim to enhance structural awareness through auxiliary pretraining tasks while avoiding modifications to the original attention mechanism. However, the pretraining method employed by Xing and Wan (2021) primarily focuses on local structural information, making it difficult to capture the global structure of the

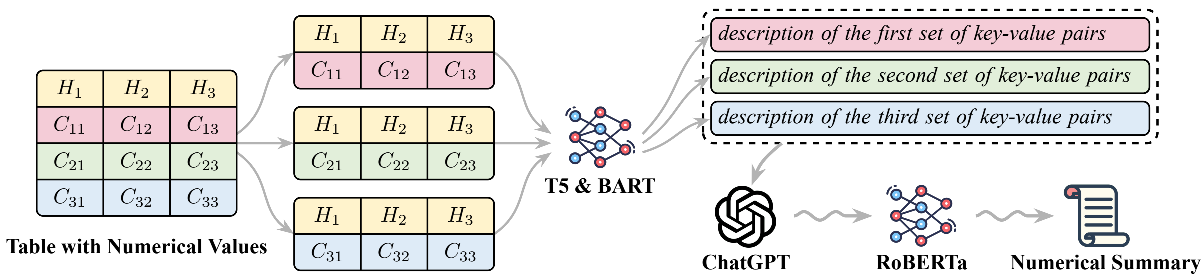

Figure 1: Overview of the dataset construction pipeline for numerical summarization. T5 and BART are used to generate descriptions of key-value pairs, and RoBERTa is employed to verify the correctness of the sentences.

table. Meanwhile, the pretraining tasks designed by Alonso et al. (2024) are relatively coarse, and converting tables into images introduces additional noise. Our approach shares similarities with these methods in that it also leverages row and column features to predict relevant cells. However, we further extend the pretraining tasks of Alonso et al. (2024) to textual models and refine the structural understanding process through a question-driven approach. Furthermore, we design 13 additional table structure-related questions to further enhance the model's comprehension of tabular structures.

3 Methodology

This section presents our method for table-to-text generation, which comprises three core components: (i) a word-to-sentence reconstruction pretraining task to improve fluent text generation from sparse input; (ii) a numerical summarization pretraining task to strengthen the model's ability to aggregate and summarize numerical information; and (iii) a table structure awareness mechanism trained via self-supervised structure-related queries. Additionally, we discuss the loss computation framework used to jointly optimize structural awareness and text generation.

3.1 Word-to-Sentence Reconstruction

Table-to-text generation is essentially a modeling process that transforms sparse textual input into dense, coherent sentences. Therefore, keyword-to-text generation can serve as an effective pretraining task. However, unlike standard keyword-to-text generation, the content in tables mainly consists of nouns, numerals, with only a small proportion of prepositions, verbs, and adjectives. This makes it more challenging to reconstruct complete sentences using standard keyword-to-text generation. To bridge this gap, we introduce a data construction method that simulates the lexical sparsity of table inputs while enabling large-scale training.

Specifically, for a given sentence, we first apply the Stanza[1] toolkit to perform part-of-speech tagging, retaining words that belong to table-relevant categories, such as nouns and numerals. To prevent the extracted text from becoming excessively sparse, we also retain some words from other parts of speech with a low probability, as detailed in Appendix A. To better align this pretraining task with the characteristics of table-to-text generation, we randomly shuffle the extracted core word sequence and inject a small number of noise tokens to increase the task complexity. In addition, we leverage ChatGPT to generate paraphrased versions of the original sentence, further enriching the diversity of training data. The model is then trained to recover the original sentence from this perturbed input, thereby learning to order words and construct coherent sentences from sparse lexical cues.

3.2 Numerical Summarization

In table-to-text generation, the model needs to possess the ability to expand sparse textual information into more coherent and detailed text. It also needs to be capable of precisely extracting key information from tables. This capability is particularly important, as tables often contain a large number of cells associated with the same header and exhibiting repetitive structural patterns. Moreover, the prevalence of numerical values in tables requires the model to understand, compare, and summarize quantitative information effectively.

To enhance the model's ability to summarize numerical content, we attempted to fine-tune it using existing text summarization datasets, such as Multi-News (Fabbri et al., 2019), XSum (Narayan et al., 2018), Newsroom (Grusky et al., 2018), and CNN/DailyMail (Nallapati et al., 2016). However, these datasets contain sparse and scattered sum-

[1] https://stanfordnlp.github.io/stanza

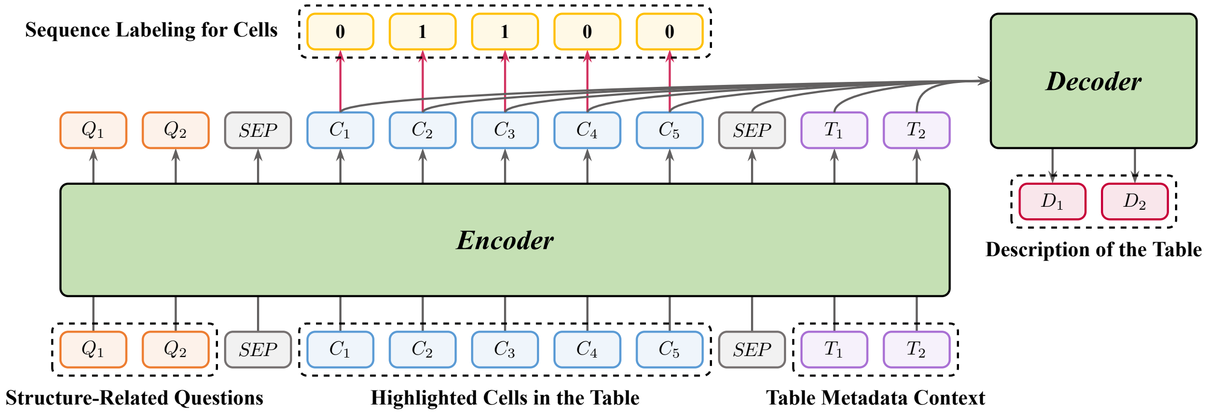

Figure 2: Overview of the proposed model for table structure awareness and text generation. "Table Metadata Context" refers to the contextual information of the table, consisting of the table's page title, section title and text.

maries for numerical content, which makes it difficult to effectively improve the model's ability to generalize numerical information. Therefore, we decided to construct a custom text summarization dataset specifically focused on numerical content.

In table-based question answering task (Pasupat and Liang, 2015), there are many high-quality datasets whose tables are typically rich in numerical information. Thus, we propose using existing text generation models to directly generate detailed descriptions of these numerically dense tables, guiding the table-to-text generation model in capturing key numerical information. However, this approach relies on the model's ability to accurately perceive the table's structural information, which current models do not handle well. To bypass the dependency on the model's structural perception, we propose a simplified solution.

Specifically, we decompose each table into a set of independent rows. We then pair each cell in a row with its corresponding row header, transforming the row into a set of key-value pairs. These key-value pairs are then processed by pretrained generation models, such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), to produce detailed natural language descriptions. Finally, we concatenate the descriptions of all key-value pairs into a single coherent text and use ChatGPT's powerful text generation capabilities to perform numerical summarization. To ensure the correctness of the generated summaries, we further split the summaries into individual sentences and verify each one using a textual entailment model, RoBERTa (Liu et al., 2019). An overview of this pipeline is illustrated in Figure 1. The model is then trained to generate a numerical summary from the detailed table description, thereby learning to extract and summarize key numerical insights from tabular data.

3.3 Table Structure Awareness

Existing methods typically model table structure either by inserting special tokens between input cells (Kale and Rastogi, 2020) or by appending row and column features to each cell (Andrejczuk et al., 2022). However, the special tokens used in the former approach are often semantically shallow and dispersed, while the features introduced in the latter lack task-specific training. As a result, these methods fail to effectively guide the model in focusing on key structural information, making it difficult to capture cell relationships.

3.3.1 Structure-Aware Question Design

Our method is inspired by the table-based question answering (QA) task (Pasupat and Liang, 2015). In table QA, posing questions about the table content, such as "*Which department has the highest sales?*" or "*What is the sales trend from 2020 to 2025?*", can effectively guide the model to focus on key information in the table. This enhances the model's understanding of the table content. Similarly, posing questions related to the table structure can also guide the model to better capture and perceive structural information. Therefore, based on human understanding of table structure, we designed a set of 20 structure-related questions that cover various structural relationships within a table. These questions are categorized into five types:

(1) **Row-column relationships:** "Which cells are in the same row / column as cell C_{ij}?"

(2) **Header relationships:** "Which cells serve as the row / column header of cell C_{ij}?"

(3) **Spatial relationships:** "Which cells are located directly to the left / right / above / below cell C_{ij}?"

(4) **Proximity relationships:** "Which cells are adjacent to cell C_{ij}, positioned to its left / right / above / below?"

(5) **Relative positioning:** "In terms of row / column direction, which cell, C_{ij} or C_{mn}, is positioned earlier / later?"

In the above questions, C_{ij} and C_{mn} denote arbitrary cells in the table positioned at (i, j) and (m, n). A complete list of the 20 structure-related questions is provided in Appendix C.

3.3.2 Structural Representation Learning

To equip the model with structural perception, we extend the original table input by incorporating a structure-related query Q. At the encoder side, the model need to perform sequence labeling over each input cell, conditioned on the given query. For example, given the query "*Which non-header cells are in the same row as cell C_{ij}?*", if cell C_{mn} is in the same row as cell C_{ij} and is not a header cell, it is labeled as "relevant"; otherwise, it is labeled as "irrelevant." In this way, we can guide the model to perceive the table's structural information, enabling it to establish relationships between cells based on their structural dependencies. In practice, for the same table and highlighted cells, multiple structure-related questions can be posed. Our method randomly samples one question template from a pool of candidate templates. An overview of the model architecture is provided in Figure 2.

Furthermore, to better leverage the two-dimensional structure of the table and avoid redundant input (Alonso et al., 2024), we do not add special separators between cells. Similar to Andrejczuk et al. (2022), we enhance the model's structural perception by adding row and column features to each cell. However, unlike their method, we also introduce three additional features for each cell: `segment` (cell category), `row_span` (row span), and `col_span` (column span). Details of these feature representations can be found in Appendix B.

3.4 Loss Computation

To optimize the model for both structural awareness and text generation, we employ a dual-task learning approach, incorporating *sequence labeling loss* for structural perception and *text generation loss* for natural language generation.

Sequence Labeling Loss To enable the model to predict structural relationships between table cells accurately, we frame this as a binary classification task, optimized using binary cross-entropy loss:

$$L_i = y_i \log p_i + (1 - y_i) \log(1 - p_i)$$
$$\mathcal{L}_{\text{seq}} = -\frac{1}{N} \sum_{i=1}^{N} L_i \tag{1}$$

where N is the total number of input cells, $y_i \in \{0, 1\}$ is the ground truth label (1 if the cell belongs to the same structural group, otherwise 0), and p_i is the predicted probability that cell i belongs to the same structural group.

Text Generation Loss Since the ultimate goal is to generate fluent table descriptions, we adopt the standard sequence-to-sequence cross-entropy loss, defined as:

$$\mathcal{L}_{\text{gen}} = -\frac{1}{T} \sum_{t=1}^{T} \log p_\theta(w_t \mid w_{<t}) \tag{2}$$

where T is the length of the target text, w_t is the t-th token in the target sequence, and $p_\theta(w_t \mid w_{<t})$ is the probability of generating token w_t given the previously generated tokens.

Total Loss To jointly optimize for structural understanding and text generation, we combine both objectives into a unified loss function:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{seq}} \mathcal{L}_{\text{seq}} + \lambda_{\text{gen}} \mathcal{L}_{\text{gen}} \tag{3}$$

where λ_{seq} and λ_{gen} are hyperparameters that control the relative importance of structural perception and text generation.

4 Experiment

4.1 Datasets

In this work, we use the ToTTo (Parikh et al., 2020) and HiTab (Cheng et al., 2022) datasets, which place greater demands on the model's ability to understand table structure.

ToTTo: An open-domain table-to-text generation dataset with over 120,000 examples from Wikipedia tables. The task is to generate a single-sentence description based on a table and highlighted cells. The dataset includes a variety of topics and covers challenging linguistic phenomena, such as reasoning and numerical inference. It is split into a training set (120,761 examples), a development set (7,700 examples), and a test set (7,700 examples). To evaluate generalization, overlapping tables are removed between the training and test sets, and the development and test sets are divided into overlapping and non-overlapping subsets.

HiTab: A dataset consisting of hierarchical tables, designed for table-based question answering (QA) and table-to-text generation (NLG). It contains tables from statistical reports and Wikipedia, annotated with entity and quantity alignments. The task involves generating a description based on the table, highlighted cells, and symbolic operators. HiTab presents challenges due to its complex structure, requiring advanced reasoning and numerical inference. The dataset includes 3,597 tables, divided into training, development, and test sets.

4.2 Baselines

We present baseline results of the following representative methods:

T5-based (Kale and Rastogi, 2020): Employs pre-trained T5 model by linearizing the table into sequences of `(row, column, value)` triplets and adding special tokens to model table structure.

LATTICE (Wang et al., 2022): Uses an equivariant learning framework with graph-based self-attention to capture relationships within the table while ignoring irrelevant interactions.

UniD2T (Li et al., 2024): Converts structured data into a graph format to enable graph-to-text generation, and enhances the T5 model with novel position and attention matrices.

PixT3 (Alonso et al., 2024): Utilizes a visual-language model to treat tables as images. The model is pre-trained to predict all cells that are in the same row and column as a given cell.

4.3 Experimental Settings

In the *word-to-sentence reconstruction* phase, we aimed to ensure that the topic distribution of the constructed data aligns with the ToTTo and HiTab datasets. To achieve this, we applied simple regular expressions to filter 1.7 million sentences from the Wikipedia dataset[2] provided by Wikimedia. To prevent data leakage, we excluded sentences with URLs that matched those in the ToTTo dataset. In the *numerical summarization* phase, we collected 400,000 tables from the OTTQA (Chen et al., 2021), TabFact (Chen et al., 2020), and WikiSQL (Zhong et al., 2017) datasets, resulting in 4.5 million key-value pair sequences after splitting. After cleaning, we obtained 920,000 numerical summarization data points. Additionally, to enable T5 and BART to generate fluent text from key-value pair sequences, we fine-tuned the models using data from E2E (Novikova et al., 2017), WikiBio (Lebret et al., 2016), and 8,000 annotated examples from ChatGPT. The prompt used for generating the numerical summaries via the ChatGPT API can be found in Appendix D. In the *structural perception* phase, we used question templates from categories 1, 2, 3, and 5, as explained in Section 4.6.

All our experiments were conducted using the T5-base pretrained model. We set the loss weights for the sequence labeling and text generation tasks, λ_{seq} and λ_{gen}, to 1, respectively. The learning rate was set to 2e-4, and we trained for 30 epochs using the AdamW optimizer, with a linear learning rate scheduler and a warmup ratio of 0.15. Pretraining tasks 1 and 2 were trained simultaneously with a batch size of 20. We utilized 8 NVIDIA 4090 GPUs for pretraining, which took approximately 23 hours. For fine-tuning on the ToTTo dataset, we again used 8 NVIDIA 4090 GPUs, with a batch size of 24, and the training time was around 0.8 hours. The model input sequence length was limited to 300 tokens. During the text generation phase, we set the number of beams to 5, and the maximum output length to 300 tokens.

4.4 Main Results

Table 1 presents our results on the ToTTo test set. We used the ToTTo leaderboard's standard evaluation metrics: BLEU (Papineni et al., 2002) for fluency, PARENT (Dhingra et al., 2019) for faithfulness to the table content, and BLEURT (Sellam et al., 2020) for both fluency and overall adequacy. The development and test sets are divided into two subsets: the "Overlap Subset," where table headers are present in the training set, and the "Non-Overlap Subset," where they are absent. The "Overall" scores reflect the aggregated performance across these two subsets. Test set results are obtained via submissions to the ToTTo leaderboard, as the test set is not publicly available.

Compared to the original T5-base, our method improves overall BLEU and PARENT scores by 1.5 and 1.8 points, respectively, demonstrating its effectiveness. Furthermore, our approach performs comparably to the larger T5-3B model in terms of PARENT and BLEURT scores. Notably, apart from our method, both LATTICE and UniD2T also outperform the original T5-base, further highlight-

[2]https://huggingface.co/datasets/wikimedia/wikipedia

Model	Overall			Overlap Subset			Non-Overlap Subset		
	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT
LATTICE	48.4	58.1	0.222	56.1	62.4	0.345	40.4	53.9	0.099
UniD2T	48.6*	58.0*	0.233*	56.5*	62.2*	**0.352***	40.6*	53.8*	0.114*
PixT3	45.4	55.5	—	53.2	60.4	—	37.5	50.6	—
T5-base	47.4	56.4	0.221	55.5	61.1	0.344	39.1	51.7	0.098
T5-3B	**49.5**	**58.4**	0.230	**57.5**	**62.6**	0.351	**41.4**	**54.2**	0.108
QuASAR	48.9	58.2	**0.234**	56.5	62.3	**0.352**	41.1	54.0	**0.116**

Table 1: Results on the ToTTo test set. Best scores are in bold. LATTICE uses the T5-base model, and the results for UniD2T are those we reproduced using T5-base. PixT3 does not report the BLEURT metric in their paper.

ing the importance of structural perception in table-to-text generation. In comparison, although PixT3 incorporates table structure modeling, its performance remains below that of T5-base. PixT3 treats tables as images to leverage their two-dimensional visual features; however, this may introduce additional noise, suggesting that visual information is not essential for this task. Additionally, our method either matches or outperforms both LATTICE and UniD2T across all evaluation metrics.

To further validate the effectiveness of our approach, we conducted additional experiments on the HiTab dataset, with results presented in Table 2. The experimental results for the first three methods are from Cheng et al. (2022). As shown, all models scored relatively low on the BLEU and PARENT metrics. This can be attributed to the small size of the HiTab dataset, its complex table structures, and discrepancies in numerical precision between the table content and the generated text. Despite these limitations, our method still achieved the best performance, further demonstrating its advantages in table-to-text generation.

Model	BLEU	PARENT
BERT-to-BERT (Rothe et al., 2020)	11.4	16.7
BART-base (Lewis et al., 2020)	17.9	28.0
T5-large (Raffel et al., 2020)	19.5	35.7
LATTICE (Wang et al., 2022)	16.3	22.7
QuASAR (T5-base)	**23.7**	**40.8**

Table 2: Results on the HiTab test set.

4.5 Human Evaluation

Given the limitations of automatic metrics such as BLEU for tasks beyond translation (Reiter, 2018), we additionally conducted a human evaluation to assess our method's ability to reduce hallucinations and improve factual accuracy. Concretely, we recruited five annotators with solid backgrounds in NLP to perform the evaluation. We carefully selected 100 samples from the HiTab test set and the ToTTo development set, focusing on cases with complex table structures, a large number of highlighted cells, and non-trivial numerical summarization. We compared the outputs of T5-base, T5-3B, LATTICE, UniD2T, and our QuASAR model along four human evaluation dimensions: Fluency, Factual Consistency, Numerical Accuracy, and Information Coverage. Each sample was rated on a 1–5 Likert scale. The averaged scores (mean ± standard deviation) are reported in Table 3:

	Fluency	Consistency	Accuracy	Coverage
T5-base	3.85±0.23	2.74±0.28	2.94±0.23	3.24±0.40
T5-3B	**4.24±0.17**	3.19±0.18	3.42±0.39	**4.09±0.26**
LATTICE	3.95±0.43	3.60±0.34	3.02±0.14	3.75±0.14
UniD2T	4.03±0.35	3.58±0.22	3.16±0.28	3.83±0.25
QuASAR	3.98±0.19	**3.86±0.28**	**3.47±0.15**	3.91±0.23

Table 3: Human evaluation results on four dimensions.

Our method outperforms all baselines in Factual Consistency and Numerical Accuracy, while maintaining competitive performance on the other two metrics. T5-3B achieves the highest Fluency and Coverage scores. However, its Factual Consistency remains close to that of T5-base, indicating a tendency to hallucinate table structure. This problem is alleviated by our structure-aware approach.

4.6 Ablation Study

To assess the contribution of each component in our method, we performed two ablation studies on the ToTTo dataset: one on structure-related question categories and the other on the core components of table-to-text generation.

Structure-related Question Categories We first validated the effectiveness of the five categories of structure-related questions. These categories consist of 20 questions in total, as listed in Appendix

Model	Overall			Overlap Subset			Non-Overlap Subset		
	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT
QuASAR	**49.2**	**58.5**	**0.246**	**57.5**	**63.2**	**0.367**	**40.9**	**53.9**	**0.126**
w/o reconstruct	48.6	58.0	0.231	56.8	62.2	0.338	40.4	53.8	0.123
w/o summary	48.8	58.2	0.224	57.1	62.7	0.335	40.6	53.7	0.113
w/o structure	47.9	57.4	0.222	55.9	61.8	0.340	40.2	53.2	0.107

Table 4: Ablation study of the core components of table-to-text generation on the ToTTo development set.

Model	BLEU	PARENT	BLEURT
All Categories	49.1	58.4	0.237
w/o Category 1	48.8	57.9	0.235
w/o Category 2	48.6	57.8	0.224
w/o Category 3	48.9	58.1	0.229
w/o Category 4	**49.2**	**58.5**	**0.246**
w/o Category 5	49.0	58.4	0.235
w/ Category 5*	48.7	57.9	0.214

Table 5: Ablation study of structure-related question categories on the ToTTo development set. The category with an asterisk (*) is the extended question category.

C. The detailed experimental results are shown in Table 5. Removing the templates of category 1 and category 2 questions led to a significant drop in generation performance. This result is consistent with our intuitive understanding of table structure. Category 1 question templates guide the model to identify relationships among cells sharing the same structural or hierarchical level, whereas category 2 templates help the model capture the modifying relationship between header and non-header cells.

Interestingly, removing the category 4 question templates resulted in a slight improvement in the model's generation performance. This may be because category 4 questions mainly focused on adjacency relationships between cells. Such local structural information contributes less to modeling the overall table structure. Moreover, removing this template increased the proportion of other question categories, which may have indirectly enhanced the model's ability to capture more global structural patterns. Removing the category 3 and category 5 question templates slightly reduced the model's generation performance.

Additionally, we further expanded the category 5 question templates to allow cells C_{ij} and C_{mn} not to be restricted to the same row or column, aiming to model the sequential relationships between all cells. However, this led to a notable performance drop, likely because the expanded templates allowed the association of unrelated cells, disrupting the model's attention to table structure.

Core Components of Table-to-Text Generation We further assessed the contribution of three core components: *word-to-sentence reconstruction*, *numerical summarization*, and *table structure awareness*, with the detailed results shown in Table 4. Removing the *table structure awareness* component led to a notable drop in BLEU and PARENT scores, which fell to 47.9 and 57.4, respectively. This underscores the essential role of structural information in table-to-text generation. Removing the *word-to-sentence reconstruction* component caused a modest 0.6-point drop in BLEU and a 0.5-point drop in PARENT. While the performance change is not drastic, it suggests that this component helps the model enrich sparse input into more coherent text. In contrast, removing the *numerical summarization* component resulted in an even smaller drop (BLEU -0.4, PARENT -0.3), indicating a more limited contribution. One possible reason is that T5-base, having been pretrained and fine-tuned on large-scale corpora, already possesses strong text generation capabilities, which limits the observable gains from our two pretraining tasks. Whether these tasks would bring larger gains for models trained from scratch or under low-resource settings remains to be further explored.

4.7 Case Study

To better illustrate our method's ability in structural understanding and numerical summarization, we present a representative example in Appendix E. The table in this example exhibits complex structural features, including merged cells and subtle cell alignments. Additionally, the example requires the model to perform a certain degree of numerical summarization. Specifically, the model is expected to infer that Veronica's Wish led to Nisha Kalema winning her third Best Actress Award.

As shown by the model outputs below the table, the T5-based method misinterprets "Association" as another work by Nisha Kalema. It fails to recognize that "Association" is modifying the "Uganda

Film Festival Awards" structural information. Although the UniD2T and LATTICE methods produce outputs that approximate the reference, they both fail to capture the numerically inferred information, specifically omitting the key detail that she won for the third time. In contrast, our method successfully captures and generates this crucial detail, demonstrating its strength in both structural understanding and numerical summarization.

5 Conclusion

In this work, we presented QuASAR, a question-driven self-supervised framework for table-to-text generation that explicitly models both local and global table structures. Our approach introduces structure-related queries and two auxiliary pretraining tasks: *word-to-sentence reconstruction* and *numerical summarization*, which together enhance the model's structural awareness and text generation quality. While our method is designed for general table-to-text generation, current experiments are limited to datasets such as ToTTo and HiTab, where the input is a small set of highlighted cells and the output consists of brief descriptions. Its effectiveness on broader table summarization tasks, where the input is a large table and the output is a more detailed description, remains to be verified. Future work could therefore focus on extending our approach to large-table summarization scenarios. Another direction is to enhance the model's robustness when handling noisy or irregular tables, which is essential for real-world deployment.

6 Limitations

While our question-driven self-supervised framework enhances structural perception, it fundamentally depends on a manually crafted set of 20 structural queries. Although diverse, these question templates may not fully capture complex table structures, such as deeply nested or multi-layered layouts, and poorly designed templates may limit the model's ability to generalize structural patterns.

Moreover, while our two pretraining tasks (*word-to-sentence reconstruction* and *numerical summarization*) are beneficial for table-to-text generation, the improvements they bring are limited when applied to already strong pretrained models such as T5. In such scenarios, where pretrained models and ample computational resources are available, these tasks may provide only marginal utility.

Additionally, when using the ChatGPT API to construct numerical summarization training data, we generated 920,000 data points to help improve the model's numerical summarization capabilities. However, this large-scale data construction may be costly and time-consuming, which could limit its feasibility for certain applications.

7 Ethical Considerations

The datasets used in this work, including ToTTo, HiTab, Wikipedia, and OTTQA, are publicly available and comply with relevant usage licenses and privacy regulations. During data construction and model training, we place strong emphasis on ensuring the diversity and representativeness of the data. This helps minimize the potential biases that could lead to discriminatory or unfair content in the generated text. Although our system is capable of automatically generating text descriptions from tabular data, human review remains essential to ensure the accuracy and reliability of the output. Given the potential misuse of automated table-to-text generation, ensuring its legal and ethical use is essential as the technology evolves.

Acknowledgements

We thank the anonymous reviewers and the meta-reviewer for their valuable feedback on our paper. We are also grateful to Ankur P. Parikh for his support with evaluating our method on ToTTo dataset and providing helpful feedback. This work was supported by the Project 62276178 under the National Natural Science Foundation of China, the Key Project 23KJA520012 under the Natural Science Foundation of Jiangsu Higher Education Institutions, the project 22YJCZH091 of Humanities and Social Science Fund of Ministry of Education and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Iñigo Alonso, Eneko Agirre, and Mirella Lapata. 2024. PixT3: Pixel-based table-to-text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6721–6736, Bangkok, Thailand. Association for Computational Linguistics.

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 2197–2210. Curran Associates, Inc.

Ewa Andrejczuk, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, and Yasemin Altun. 2022. Table-to-text generation and pre-training with TabT5. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6758–6766, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 128–135, New York, NY, USA. Association for Computing Machinery.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. Open question answering over tables and text. In *International Conference on Learning Representations*.

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

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

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

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

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. 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*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

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

Shujie Li, Liang Li, Ruiying Geng, Min Yang, Binhua Li, Guanghu Yuan, Wanwei He, Shao Yuan, Can Ma, Fei Huang, and Yongbin Li. 2024. Unifying structured data as graph for data-to-text pre-training. *Transactions of the Association for Computational Linguistics*, 12:210–228.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore. Association for Computational Linguistics.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022. PLOG: Table-to-logic pretraining for logical table-to-text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5531–5546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2047–2057, Florence, Italy. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Toru Nishino, Ryota Ozaki, Yohei Momoki, Tomoki Taniguchi, Ryuji Kano, Norihisa Nakano, Yuki Tagawa, Motoki Taniguchi, Tomoko Ohkuma, and Keigo Nakamura. 2020. Reinforcement learning with imbalanced dataset for data-to-text medical report generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2223–2236, Online. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

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

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2002. Text generation from keywords. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics.

Xinyu Xing and Xiaojun Wan. 2021. Structure-aware pre-training for table-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2273–2278, Online. Association for Computational Linguistics.

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

A Linguistic Tags for Word-to-Sentence Reconstruction

Tag	Description	Probability
ADJ	Adjective	0.20
ADP	Adposition	0.15
ADV	Adverb	0.20
AUX	Auxiliary verb	0.15
CCONJ	Coordinating conjunction	0.15
DET	Determiner	0.15
INTJ	Interjection	0.15
NOUN	Noun	1.00
NUM	Numeral	1.00
PART	Particle	0.15
PRON	Pronoun	0.50
PROPN	Proper noun	1.00
PUNCT	Punctuation	0.15
SCONJ	Subordinating conjunction	0.25
SYM	Symbol	1.00
VERB	Verb	0.25
X	Other	1.00

Table 6: Linguistic tags for word-to-sentence reconstruction and their retention probability distribution.

B Structural Embeddings for Table Input

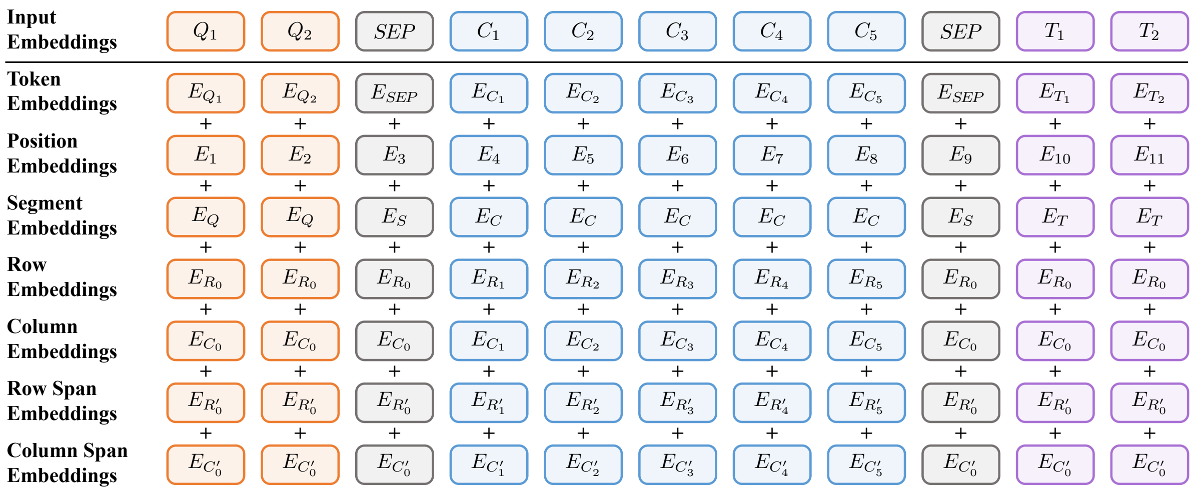

Figure 3: Embedding representations of model input tokens, with structural features (row, column, span) added. The orange tokens correspond to structure-related queries, blue tokens represent highlighted table cells and their associated headers, and purple tokens denote the table context.

C Categories and Templates of Structure-Related Questions

Category	Question Type	Question Template
Category 1	Row and Column Relationships	Which cells are in the same row as C_{ij} in the table?
		Which cells are in the same column as C_{ij} in the table?
Category 2	Header Relationships	Which cells serve as the headers for C_{ij} in the table?
		Which cells serve as the row headers for C_{ij} in the table?
		Which cells serve as the column headers for C_{ij} in the table?
		What are the non-header cells located in the same row as C_{ij}?
		What are the non-header cells located in the same column as C_{ij}?
Category 3	Spatial Positioning	What are the cells located to the left of cell C_{ij}?
		What are the cells located to the right of cell C_{ij}?
		What are the cells located above cell C_{ij}?
		What are the cells located below cell C_{ij}?
Category 4	Proximity Relationships	What are the cells that are immediately adjacent to cell C_{ij}?
		What is the neighboring cell located to the left of cell C_{ij}?
		What is the neighboring cell located to the right of cell C_{ij}?
		What is the neighboring cell located above cell C_{ij}?
		What is the neighboring cell located below cell C_{ij}?
Category 5	Relative Positioning	Between cell C_{ij} and cell C_{mn}, which one comes earlier in the row?
		Between cell C_{ij} and cell C_{mn}, which one comes later in the row?
		Between cell C_{ij} and cell C_{mn}, which one comes earlier in the column?
		Between cell C_{ij} and cell C_{mn}, which one comes later in the column?

Table 7: The questions in the table are designed to inquire about structural information of tables. In practical use, we employ multiple paraphrased versions of each question template to enhance the model's understanding of table structures. Here, C_{ij} and C_{mn} represent the text content randomly selected from highlighted cells in the table. These cells are either from the same row or the same column.

D Numerical Summarization Prompt for Table Data

Name	Completed	Architect	Location	Year of listing
1 Booth Street	circa 1850s	Unknown	Booth Street	1974
8 Lower Park Road	circa 1875	Alfred Waterhouse	Lower Park Road, Rusholme	1974
15-17 King Street	circa 1920-30	Maxwell and Tuke	King Street	1994
29 Swan Street	circa 1865s	Unknown	Swan Street, Ancoats	1989
42-44 Sackville Street	circa 1873	Pennington and Brigden	Sackville Street	1974
50 Newton Street	circa 1900	Clegg and Knowles	Newton Street, Piccadilly	1988

Numerical Summary Prompt

I will provide a detailed description of a table. Based on the given information, generate a concise and insightful summary that synthesizes and reasons with numerical data. Strictly limit your response to 80 words or fewer—any response exceeding this limit will be rejected. Focus on identifying numerical insights such as ranges, extremes, totals, averages, and trends or relationships. Use brief, precise language to maximize informativeness. Here is an example:

Input:
1 Booth Street, completed in the 1850s, is located on Booth Street and was listed in 1974. The architect is unknown. 8 Lower Park Road, completed circa 1875, is located in Rusholme, designed by Alfred Waterhouse, and listed in 1974. 15-17 King Street, completed between 1920 and 1930, was designed by Maxwell and Tuke and listed in 1994. 29 Swan Street, completed in the 1860s, is located on Swan Street, Ancoats, and was listed in 1989. The architect is unknown. 42-44 Sackville Street, completed in 1873, was designed by Pennington and Brigden, located on Sackville Street, and listed in 1974. 50 Newton Street, completed circa 1900, was designed by Clegg and Knowles, located on Newton Street, Piccadilly, and listed in 1988.

Output:
Between the 1850s and 1930s, six buildings were completed across Manchester, with architects identified for four. Maxwell and Tuke's design for 15-17 King Street was listed in 1994, while Alfred Waterhouse's work and Pennington and Brigden's building were both listed in 1974. The earliest building, 1 Booth Street, remains standing but its architect is unknown.

Figure 4: A prompt designed to generate concise and insightful numerical summaries from table data. The input consists of alternating orange and blue text. The first orange text describes the first key-value pair sequence. The second blue text describes the second key-value pair sequence, and so on.

E Structure and Numerical Analysis

Table Title: Nisha Kalema
Section Title: Awards & Nominations
Table Description: None

<table>
<tr><th>Year</th><th>Nominated work</th><th>Association</th><th>Category</th><th>Result</th></tr>
<tr><td>2015</td><td>The Tailor</td><td rowspan="5">Uganda Film Festival Awards</td><td rowspan="3">Best Actress</td><td>Won</td></tr>
<tr><td>2016</td><td>Freedom</td><td>Won</td></tr>
<tr><td rowspan="3">2018</td><td rowspan="4">Veronica's Wish</td><td>Won</td></tr>
<tr><td>Best Script (Screen Play)</td><td>Won</td></tr>
<tr><td>Best Feature Film</td><td>Won</td></tr>
<tr><td>2019</td><td>Mashariki African Film Festival</td><td>Best East African Feature Film</td><td>Nominated</td></tr>
</table>

Gold Answer: Nisha Kalema received her **third** Best Actress Award for the film Veronica's Wish at the 2018 Uganda Film Festival Awards.
T5-based: Nisha Kalema won the Best Actress award at the 2018 Uganda Film Festival Awards for **Association** and Veronica's Wish.
UniD2T: Nisha Kalema got Best Actress at the 2018 Uganda Film Festival Awards for Veronica's Wish.
LATTICE: Nisha Kalema won the Best Actress award at the Uganda Film Festival Awards for her role in Veronica's Wish (2018).
QuASAR: Nisha Kalema won the Best Actress award at the 2018 Uganda Film Festival Awards for Veronica's Wish, making it the **third** time she won Best Actress.

Figure 5: An example from the ToTTo dev set illustrating table-to-text generation. The input for each model consists of the highlighted cells and their corresponding headers, along with the table's contextual information. In the generated descriptions, blue word represents key information, while red word indicates incorrect information.