Can LLMs Generate Tabular Summaries of Science Papers? Rethinking the Evaluation Protocol

Anonymous ACL submission

Abstract

Literature review tables are essential for summarizing and comparing collections of scientific papers. We explore the task of generating tables that best fulfill a user's informational needs given a collection of scientific papers. Building on recent work (Newman et al., 2024), we extend prior approaches to address real-world complexities through a combination of LLM-based methods and human annotations. Our contributions focus on three key challenges encountered in real-world use: (i) User prompts are often under-specified; (ii) Retrieved candidate papers frequently contain irrelevant content; and (iii) Task evaluation should move beyond shallow text similarity techniques and instead assess the utility of inferred tables for information-seeking tasks (e.g., comparing papers). To support reproducible evaluation, we introduce ARXIV2TABLE, a more realistic and challenging benchmark for this task, along with a novel approach to improve literature review table generation in real-world scenarios. Our extensive experiments on this benchmark show that both open-weight and proprietary LLMs struggle with the task, highlighting its difficulty and the need for further advancements.

1 Introduction

Literature review tables play a crucial role in scientific research by organizing and summarizing large amounts of information from selected papers into a concise and comparable format (Russell et al., 1993). At the core of these tables are the *schema* and *values* that define their structure, where *schema* refers to the categories or aspects used to summarize different papers and *values* correspond to the specific information extracted from each paper. A well-defined *schema* allows each work to be represented as a row of *values*, enabling structured and transparent comparisons across different studies.

With recent advancements in large language models (LLMs; OpenAI, 2025b; DeepSeek-AI et al., 2025), several studies (Newman et al., 2024; Dagdelen et al., 2024; Sun et al., 2024) have explored generating literature review tables by prompting LLMs with a set of pre-selected papers and the table's caption. While these efforts represent meaningful progress, we argue that the existing task definition and evaluation protocols are somewhat unrealistic, thus hindering the practical applicability of generation methods.

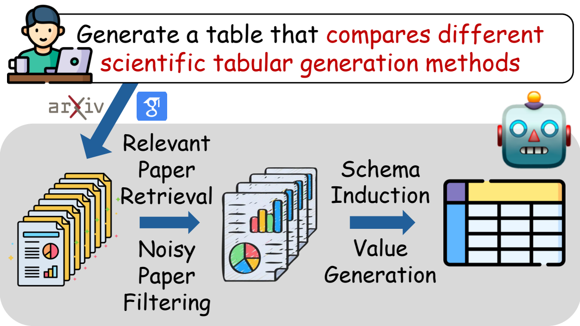

Figure 1: Overview of our proposed task: Given a user's demand, the LLM first selects the relevant papers that match the request and then generates the schema and values for the desired table.

First, existing pipelines assume that all provided papers are relevant and should be included in the table. However, in real-world scenarios, distractor papers—those that are irrelevant or contain limited useful information—are common (OpenAI, 2025a). Models should be able to identify and filter out such papers before table construction. Additionally, current pipelines use the ground-truth table's descriptive caption as the objective for generation. These captions often lack sufficient context, making it difficult for LLMs to infer an appropriate schema, or they may inadvertently reveal the schema and values, leading to biased evaluations.

In this paper, we introduce our task, as illustrated in Figure 1, which improves upon previous task definitions through two key adaptations. First, our pilot study shows that LLMs struggle to retrieve relevant papers from large corpora. To benchmark this, we introduce distractor papers by selecting them based on semantic similarity to papers in the ground-truth table. LLMs must first determine which papers should be included before generating the table. Second, we replace table captions with abstract user demands that describe the goal of curating the table, making the task more aligned with real-world scenarios. We build upon the ARXIVDIGESTABLES (Newman et al., 2024) dataset and construct a sibling benchmark through human annotation to verify the selected distractors, comprising 1,957 tables and 7,158 papers.

Meanwhile, current evaluation methods rely on static semantic embeddings to estimate schema overlap between generated and ground-truth tables and require human annotations to assess the quality of unseen schemas and values. However, semantic embeddings struggle to capture nuanced, context-specific variations due to their reliance on pre-trained representations, while human annotation is costly and time-consuming. Moreover, the most effective table generation approaches define schemas primarily based on paper abstracts. This method risks missing important aspects present in the full text, leading to loosely defined schemas with inconsistent granularity.

To address these issues, we propose an annotation-free evaluation framework that instructs an LLM to synthesize QA pairs based on the ground-truth table and assess the generated table by answering these questions. These QA pairs evaluate table content overlap across three dimensions: schema-level, single-cell, and pairwise-cell comparisons. Additionally, we introduce a novel table generation method that batches input papers, iteratively refining paper selection and schema definition by revisiting each paper multiple times. Extensive experiments using five LLMs demonstrate that they struggle with both selecting relevant papers and generating high-quality tables, while our method significantly improves performance on both fronts. Expert validation further confirms the reliability of our QA-synthetic evaluations.

In summary, our contributions are threefold: (1) We introduce an improved task definition for literature review tabular generation, benchmarking it in a more realistic scenario by incorporating distractor papers and replacing table captions with abstract user demands; (2) We propose an annotation-free evaluation framework that leverages LLM-generated QA pairs to assess schema-level, single-cell, and pairwise-cell content overlap, addressing the limitations of static semantic embeddings and human evaluation; and (3) We develop a novel iterative batch-based table generation method that processes input papers in batches, refining schema definition and paper selection iteratively.

To the best of our knowledge, we are the first to introduce a task that simulates real-world use cases of scientific tabular generation by incorporating user demands and distractor papers, providing a more robust assessment of LLMs in this domain.

2 Related Works

Scientific literature tabular generation Prior works primarily attempt to generate scientific tables through two stages: schema induction and value extraction. For schema induction, early methods like entity-based table generation (Zhang and Balog, 2018) focused on structured input, while recent work has explored schema induction from user queries (Wang et al., 2024) and comparative aspect extraction (Hashimoto et al., 2017). For value extraction, various approaches such as document-grounded question-answering (Kwiatkowski et al., 2019; Dasigi et al., 2021; Lee et al., 2023), aspect-based summarization (Ahuja et al., 2022), and document summarization (DeYoung et al., 2021; Lu et al., 2020) have been proposed to extract relevant information. Beyond these methods, several datasets have been introduced to support scientific table-related tasks, such as TableBank (Li et al., 2020), SciGen (Moosavi et al., 2021), and SciTabQA (Lu et al., 2023). Recently, Newman et al. (2024) proposed streamlining schema and value generation with LLMs sequentially and curated a large-scale benchmark for evaluation. However, all these methods assume a clean and fully relevant set of papers and rely on predefined captions or abstract-based schemas, which risk missing key details. In contrast, we argue for an evaluation approach where candidate papers include tangentially relevant or distracting papers, aligning more closely with real-world literature review workflows.

Table induction for general domains Other than the scientific domain, table induction is also widely studied as text-to-table generation. Prior works attempt this as a sequence-to-sequence task (Li et al., 2023; Wu et al., 2022) or as a question-answering problem (Sundar et al., 2024; Tang et al., 2023). Similar to these works, our framework is capable of better handling both structured and distractive input for real-world literature review and knowledge synthesis.

3 Task Definition

We first define a pipeline consisting of three subtasks that extend prior definitions and better capture the real-world usage of literature review tabular generation. For all the following tasks, we are given a user demand prompt p, which specifies the intended purpose of creating the table. **(T1) Candidate Paper Retrieval:** We begin with a given *universe* of papers (e.g., the content of Google Scholar or arXiv) from which relevant papers need to be identified. Given a large collection, the goal is to use a search engine (IR) to retrieve a subset of *candidate* papers $C := \{d_i\}_{i=1}^{M}$ of size M, which may include distractor papers—i.e., papers that resemble the user demand prompt but do not fully satisfy the requirement. **(T2) Paper Selection:** Given C, the second subtask is to select the *relevant* subset of size m ($m < M$): $R := \{d_i\}_{i=1}^{m} \subseteq C$, which best aligns with the user demand p. T2 differs from T1 in scale. Due to the large scale of T1, IR engines must optimize for recall, ensuring that as many relevant papers as possible are retrieved. However, T2 operates at a smaller scale, where precision is the priority, as it focuses on filtering out distractors and selecting only the most relevant papers. **(T3) Table Induction:** Given the selected papers R, the objective is to generate a table with m rows and N columns, where $N \geq 2$ (i.e., no single-column tables). Each row $r_i \in \{r_1, r_2, \ldots, r_m\}$ corresponds to a unique input document $d_i \in R$, and each column $c_j \in \{c_1, c_2, \ldots, c_N\}$ represents a unique aspect of the documents. We refer to these N columns as the *schema* of the table and the $N \times m$ cells as the *values* of the table. The value of each cell is derived from its respective document according to the aspect defined by the corresponding column.

4 ARXIV2TABLE Construction

We then construct ARXIV2TABLE based on the ARXIVDIGESTABLES dataset which consists of literature tables (extracted from computer science papers) and their corresponding captions. We filter out tables that are structurally incomplete or lack full text for all referenced papers. As a result, we are left with 1,957 tables (with captions) which have rows referring to 7,158 papers. Our construction involves three pillars: user demand inference (§4.1), a simulated paper retrieval (§4.2) and evaluation through utilization (§4.3).

4.1 Constructing User Demand Prompts

The first step is to collect user demands p that explicitly describe the desired table (can be understood without the table content) and do not reveal the table's schema or specific values.

Table captions are not appropriate prompts While the input dataset contains one caption per table, collected from arXiv papers, these captions are meant to complement tables rather than fully describe them. As a result, they are generally concise. For example, a table caption might read: *"Performance comparison of different approaches,"* which is too vague to understand without seeing the table. Consequently, using table captions as prompts may not yield a well-defined task. A more contextually self-contained rewritten user demand might instead be: *"Draft a table that compares different knowledge editing methods, focusing on their performance on QA datasets."*

Our prompt construction To address this issue, we propose rewriting the captions of literature review tables into abstract yet descriptive user intentions using LLMs. We guide GPT-4o with a prompt (see §A) that first explains the task to the LLM, specifying that the user demand should be sufficiently contextualized to clearly state the table's purpose while avoiding the inclusion or direct description of column names or specific values. GPT-4o is then expected to infer the user demand for the given table and its caption. For simplicity, we collect only one user demand per table. More examples are provided in Appendix D.

Table captions vs. constructed user demand prompts To verify that our collected user demands align with our objective, we visualize: (1) the distribution of the number of tokens in the original and modified user demands, and (2) the ratio of captions and user demands of different lengths that have token overlap with the schema or values. From Figure 2, we observe that our modified user demands are generally longer than the original captions, providing a more detailed description of the table's goal. Furthermore, as shown in Table 1, user demands exhibit a significantly lower overlap ratio with the schema and table values, resulting in fewer overlapping tokens. This ensures a fairer subsequent evaluation.

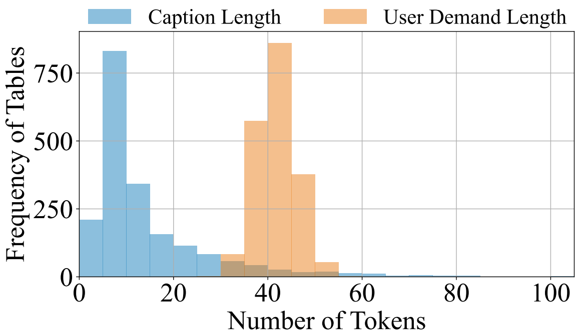

Figure 2: Distribution of the number of tokens between original captions and our modified user demands.

Prompt	Content	#Table ↓	#Tokens ↓
Caption	Schema	101 (5.2%)	1.2
	Value	46 (2.4%)	1.3
User Demand	Schema	14 (0.7%)	1.0
	Value	8 (0.4%)	1.0

Table 1: Overlap statistics between prompts (the original caption or our constructed user demand) and table content (schema or values). **#Table:** Number (and %) of tables with at least one token from table content overlapping with the prompt. **#Tokens:** Average count of overlapping tokens between table content and prompt.

4.2 Paper Retrieval Simulation

The unreliability of paper retrieval Next, we approach the first subtask, candidate paper retrieval, by conducting a pilot study to assess whether LMs can reliably retrieve relevant papers from a large corpus. For each table, we employ a SentenceBERT (Reimers and Gurevych, 2019) encoder as a retrieval engine, selecting papers from the entire corpus based on the highest similarity between the table's user demand and each paper's title and abstract. We vary the number of retrieved papers between 2 and 100 and plot the precision and recall of retrieval against the ground-truth papers in the original table (Figure 3).

We observe consistently low precision and recall across different retrieval sizes, highlighting the challenge of retrieving relevant papers from a noisy corpus. This demonstrates that the first subtask is non-trivial and may introduce noise into subtask T2. However, various information retrieval engines, such as Google Scholar and Semantic Scholar, can replace LMs in this subtask. Thus, we decide to simulate T1 by manually adding noisy distractor papers into C to construct R, ensuring a noisy input for T2. This allows us to focus on evaluating LLMs' capabilities in the T2 and T3 subtasks.

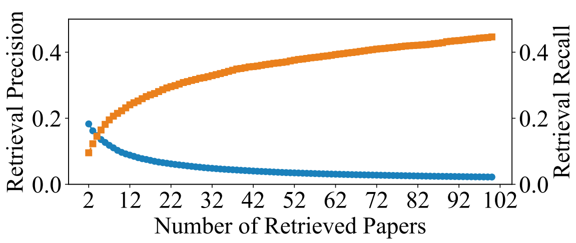

Figure 3: Precision and recall curves for different numbers of retrieved papers.

Similarity-based paper retrieval Moving forward, we associate distractor paper candidates with each table to simulate a potentially noisy document pool before constructing the table. Ideally, distractor candidates should be semantically related to the table but exhibit key differences that fail to meet the user demand. To select such candidates, we adopt a retrieve-then-annotate approach. First, we use a SentenceBERT encoder F to obtain embeddings for (1) the user demand $F(p)$ and (2) all papers in the corpus $\{F(d_i) \mid d_i \in C\}$. Each paper's embedding is computed by encoding the concatenation of its title and abstract. We then rank all papers $d_i \notin R$ based on the average of two cosine similarities: (1) the similarity between the candidate and the user demand, and (2) the average similarity between the candidate and each referenced paper:

$$s(d_i) = \cos(F(d_i), F(p)) + \frac{1}{m}\sum_{j=1}^{m}\cos(F(d_i), F(d_{u_j})).$$

Higher values of $s(d_i)$ indicate stronger semantic relevance, and we select the top 10 ranked papers for each table as its distractor candidates.

Candidates verification via human annotation After selecting these candidates, we conduct human annotations to verify whether they should indeed be excluded from the table. Given that annotating these tables requires expert knowledge in computer science, we recruit seven postgraduate students with research experience in the field as annotators. To ensure they are well-prepared for the task, the annotators undergo rigorous training, including pilot annotation exams. Their task is to make a binary decision on whether a given distractor paper—based on its title, abstract, user demand, the ground-truth table, and the titles and abstracts of all referenced papers—should be included in the table. Each table contains annotations for 10 papers, with each distractor paper initially assigned to two randomly selected annotators. If both annotators agree on the label, it is finalized. Otherwise, two additional annotators review the paper until a consensus is reached. In the first round, the inter-annotator agreement (IAA) is 94% based on pairwise agreement, and the Fleiss' Kappa (Fleiss, 1971) score is 0.73, indicating a substantial level of agreement (Landis and Koch, 1977). Finally, for each table, we randomly select a number of distractor papers between $[m, 10]$ and merge them with R to form C.

4.3 Evaluation via LLM-based Utilization

After constructing the benchmark, we propose evaluating the quality of generated tables from a utilization perspective to address the challenge of aligning schemas and values despite potential differences in phrasing. This is achieved by synthesizing QA pairs based on the ground-truth table and using the generated table to answer them, or vice versa. The flexibility of this QA synthesis allows us to evaluate multiple dimensions of the table while ensuring a structured and scalable assessment. An overview with running examples is shown in Figure 4.

Dimensions of evaluating a table with QAs We introduce three key aspects for evaluating a table in terms of its usability: (1) **Schema**: whether a specific column is included in the generated schema, (2) **Unary Value**: whether a particular cell from the ground-truth table appears in the generated table, and (3) **Pairwise Value**: whether relationships between two cells remain consistent in the generated table.

Recall evaluation We guide GPT-4o in generating these binary QA pairs based on the ground-truth table. For the first two aspects, we generate QA pairs for all columns and cells, whereas for the third aspect, we randomly sample 10 pairs of cells per table and synthesize them into QA pairs. We then prompt GPT-4o to answer these questions based on the generated table, providing yes/no responses. If the answer cannot be found, the model is instructed to respond with "no," and vice versa for "yes." The ratio of "yes" answers indicates how well the generated table preserves the schema, individual values, and pairwise relationships. This represents the **recall** of the ground-truth table, measuring how much original information is retained in the generated table.

Precision evaluation To additionally evaluate **precision**, we reverse the process: instead of generating QA pairs from the ground-truth table, we generate them from the generated table and ask another LLM to answer them using the ground-truth table. The precision score reflects how much of the generated table's content is actually supported by the original data. By computing the ratio of "yes" answers, we quantify the accuracy of the generated table in reflecting genuine ground-truth information, as well as any additional useful information not present in the ground-truth table.

5 Tabular Generation Methodologies

We explore a range of methods to evaluate on our proposed task, starting from several baselines inspired by prior work (§5.1) and then our proposed approach (§5.2).

5.1 Baseline Methods

We first introduce three methods for generating literature review tables to evaluate their performance on our task and use them as baselines for our proposed method. For easy reference, these methods are termed numerically.

First, **Method 1** generates the table in a one-step process. It takes all available papers R and the user demand p as input, and the model is asked to select all relevant papers and output a table with a well-defined schema and filled values in a single round of conversation. However, this method struggles with extremely long prompts that exceed the LLMs' context window when generating large tables.

To address this issue, **Method 2** processes papers individually. For each document, the model decides whether it should be included based on the user demand. If included, the model generates a table for that document. After processing all documents, the final table is created by merging the schemas of all individual tables using exact string matching and copying the corresponding values. While this approach reduces the input prompt length, it results in highly sparse tables due to inconsistent schema across papers and the potential omission of relevant information when individual papers lack sufficient context to define comprehensive table aspects.

To overcome both issues, **Method 3** (Newman et al., 2024) introduces a two-stage process. In the first stage, the model selects papers relevant to the user demand based on their titles and abstracts, then generates a corresponding schema. In the second stage, the model loops through the selected papers and fills in the respective rows based on the full text of each document. A minor drawback of this method is that the schema is generated solely from titles and abstracts, which may overlook details present only in the full text.

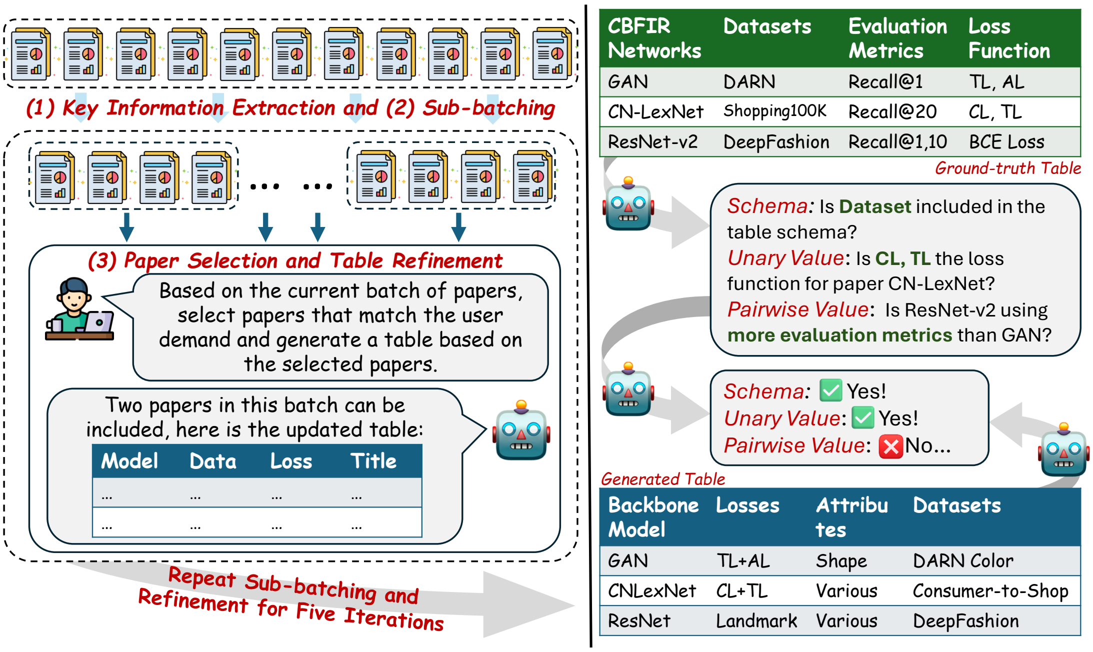

Figure 4: Overview of our proposed iterative batch-based tabular generation method (left) and LLM-based QA-synthesis evaluation protocol (right) with running examples.

5.2 Iterative Batch-based Tabular Generation

Then, we introduce our proposed method for generating literature review tables, as illustrated in Figure 4. Our approach consists of three steps: (A) key information extraction, (B) paper batching, and (C) paper selection and schema refinement, where the latter two steps can be iterated multiple times.

(A) Key Information Extraction Processing multiple papers simultaneously using their full text often results in excessively long prompts that exceed the LLMs' context window. To address this, we first shorten each paper by instructing the LLM to extract key information from the full text that is relevant to the user's requirements. Notably, we do not rely solely on the abstract, as important details often appear in the full text but are omitted from the abstract. For each paper, we provide the LLM with its title, abstract, and full text, along with the user's request, and ask it to generate a concise paragraph that preserves all potentially relevant details. These summary paragraphs serve as condensed representations of the papers for subsequent processing.

(B) Paper Batching Next, we divide all key information paragraphs into smaller batches. Processing too many papers at once negatively affects the model's performance (as demonstrated by the comparison of Method 1 in Table 2), whereas batching facilitates more efficient comparisons within each batch. For simplicity, we set a batch size of 4 and randomly partition R into $\left\lceil \frac{|R|}{4} \right\rceil$ batches.

(C) Paper Selection and Schema Refinement We initialize an empty schema and table, then sequentially process each batch with the LLM by providing it with the user's request and summaries of batched papers. The LLM is instructed to (1) decide whether each paper should be included or removed based on its key information and (2) refine the schema based on the current batch of papers. Schema refinement involves adding or removing specific columns or modifying existing values to align with different formats. For new papers that are not deemed suitable for inclusion yet are not in the current table, we also prompt the LLM to insert a new row according to the refined schema. This ensures that the table remains dynamically structured, continuously adapting to new information while maintaining consistency across batches.

Afterward, we iterate steps B and C for k iterations. Here k is a hyper-parameter and we set $k = 5$ in our experiments. The rationale is that multiple iterations allow the schema and table contents to progressively improve, ensuring better alignment with user demands. In each iteration, the batches are newly randomized so that each paper is compared with different subsets, enabling more robust decision-making and reducing bias from specific batch compositions. This iterative refinement also mitigates errors from earlier batches by revisiting and adjusting prior decisions based on newly processed information. After completing all iterations, we individually prompt the LLM to revisit the full

Backbone Model	Method	Paper	Schema			Unary Value			Pairwise Value			Avg
		Recall	P	R	F1	P	R	F1	P	R	F1	
LLAMA-3.3 (70B)	Method 1	52.8	31.3	37.7	34.2	29.6	40.4	34.2	28.4	31.8	30.0	32.8
	Method 2	65.4	26.7	**69.3**	38.5	17.0	56.8	26.2	11.2	22.5	15.0	26.6
	Method 3	61.9	36.4	40.5	38.3	32.8	44.5	37.8	29.5	30.2	29.8	35.3
	Ours	69.3	41.9	55.4	47.7	43.1	62.6	51.1	36.4	46.9	41.0	46.6
Mistral-Large (123B)	Method 1	54.7	33.1	34.5	33.8	31.6	30.4	31.0	15.5	24.7	19.0	27.9
	Method 2	66.8	27.4	65.0	38.5	22.7	47.4	30.7	17.8	30.7	22.6	30.6
	Method 3	67.9	39.9	41.6	40.7	34.7	46.3	39.7	29.9	35.1	32.3	37.6
	Ours	71.3	45.4	56.7	50.4	43.3	61.5	50.8	42.0	49.2	45.3	48.8
DeepSeek-V3 (685B)	Method 1	57.5	38.7	41.7	40.1	32.5	43.8	37.3	28.7	31.8	30.1	35.8
	Method 2	69.8	34.9	69.0	46.4	27.1	55.5	36.4	25.7	32.7	28.8	37.2
	Method 3	70.9	39.4	44.2	41.7	36.6	49.2	42.0	33.3	36.5	34.8	39.5
	Ours	74.3	39.6	56.9	46.7	47.7	65.2	55.1	40.4	49.8	44.6	48.8
GPT-4o-mini	Method 1	55.9	32.0	35.7	33.7	28.9	39.3	33.3	25.0	31.0	27.7	31.6
	Method 2	68.2	31.5	67.7	43.0	27.7	50.8	35.9	21.6	28.3	24.5	34.5
	Method 3	69.3	40.3	45.9	42.9	38.3	47.5	42.4	35.0	37.8	36.3	40.5
	Ours	72.6	46.5	59.7	52.3	**49.0**	**66.7**	**56.5**	43.5	51.9	47.3	52.0
GPT-4o	Method 1	58.5	35.8	43.2	39.2	36.9	41.8	39.2	29.0	34.7	31.6	36.7
	Method 2	70.2	34.2	68.0	45.5	27.9	56.0	37.2	19.4	33.6	24.6	35.8
	Method 3	71.3	45.0	47.9	46.4	38.7	49.8	43.6	36.9	40.0	38.4	42.8
	Ours	**74.6**	**51.5**	59.4	**55.2**	46.1	**66.7**	54.5	**45.9**	**55.7**	**50.3**	**53.3**

Table 2: Tabular evaluation results (%) of five LLMs on the ARXIV2TABLE. The best performances within each backbone are underlined and the best among all backbones are **bold-faced**. Avg refers to averaging three F1 scores.

text of the selected papers to verify the values, thereby completing the tabular generation process.

6 Experiments and Analyses

6.1 Experiment Setup

To demonstrate the generalizability of our method and evaluations, we conduct experiments using two proprietary and three open-source LLMs as backbone model representatives: GPT-4o (OpenAI, 2024b), GPT-4o-mini (OpenAI, 2024a), DeepSeek-V3 (685B; DeepSeek-AI et al., 2024), LLAMA-3.3 (70B; Dubey et al., 2024), and Mistral-Large (123B; Mistral-AI, 2024). We apply all baseline methods and our proposed method to each model and use our evaluation framework to assess the quality of the generated tables based on our benchmark, focusing on four aspects: paper selection (**Paper**), schema content overlap (**Schema**), single-cell value overlap (**Unary Value**), and comparisons across cells (**Pairwise Value**). For paper selection, we use **recall** as the metric to measure the number of ground-truth papers successfully selected. For the latter three tasks, we report precision (P), recall (R), and F1 scores (F1), as explained in §4.3.

6.2 Main Evaluation Results

We report the main evaluation results in Table 2 and summarize our key findings as follows:

(1) All methods and models struggle to distinguish relevant papers from distractors. For example, even with their best-performing methods, LLAMA-3.3 and GPT-4o achieve only 65.4% and 71.3% recall on average, respectively. This indicates that a significant number of distractor papers are still being included in the generated tables. Additionally, we observe that processing papers individually or using only abstracts for inclusion decisions yields better performance than concatenating full texts. This suggests that excessively long prompts may weaken LLMs' ability to make accurate inclusion decisions for each paper.

(2) Aligning generated schemas with the ground-truth table remains challenging. Among the baselines, the second method consistently achieves higher recall (e.g., 69.3% with LLAMA-3.3), primarily because it generates a larger number of columns, leading to more overlaps with the ground-truth schema. However, other methods exhibit significantly lower recall, indicating that LLMs still struggle to generate meaningful columns that align well with the ground-truth structure.

(3) While unary values are well preserved, pairwise comparisons suffer substantial losses. Most methods, especially our proposed approach, extract unary values with relatively high F1 scores. However, extracting and maintaining pairwise relationships remains challenging. For instance, using LLAMA-3.3, our method achieves a unary F1 score of 51.1 but drops to 41.0 for pairwise values. This trend is consistent across different models, suggesting that while individual entries are correctly identified, capturing the relationships between them remains difficult. The significant gap highlights the challenge of preserving complex relational comparisons within the generated tables.

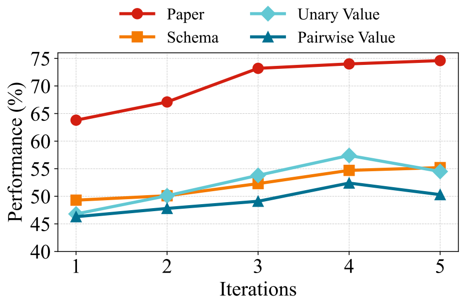

Figure 5: Ablation study on the number of iterations for our iterative batch-based table generation method.

(4) Our proposed method improves performance across all aspects and models. Across all backbone models and evaluation criteria, our method consistently outperforms the baselines. For example, it achieves the highest recall and F1 scores for both unary and pairwise metrics, regardless of model size. This demonstrates that our approach not only enhances overall performance but also provides a more robust solution for handling distractor paper selection and precise table generation.
(5) Larger models lead to better performance. For the three open-source LLMs, we observe a clear trend that increasing the model size improves performance across all aspects when using the same method. For instance, with our approach, scaling from 70B to 123B parameters leads to consistent improvements in most aspects and metrics, reinforcing the importance of stronger generative capabilities in addressing this task.

6.3 Ablation Study on Iteration Number

We further study the impact of the number of iterations, k, in our proposed method to illustrate the importance of refining the schema and table contents over multiple iterations using different batches of papers. As described in §5.2, we perform one round of paper selection and schema refinement five times to achieve optimal performance. In this section, we analyze this process by studying the model's performance across previous rounds. We select GPT-4o as the backbone model and visualize changes in the recall of paper selection and the F1 scores for schema, unary value, and pairwise comparison overlap by applying the same evaluation protocol to the generated tables after completing iterations ranging from 1 (the first cycle) to 5.

The results are plotted in Figure 5. We observe that during the first four iterations, performance steadily improves across all aspects, demonstrating the effectiveness of iteratively refining paper selection and table schema through multiple iterations and comparisons between different subsets of papers. At the fifth iteration, however, the improvement slows down, and in some cases, performance even decreases. One possible reason is that the table starts overfitting by including additional values that do not appear in the ground-truth table, reducing precision and leading to lower F1 scores. Considering the overall performance, $k = 5$ is supported as the optimal number of iterations.

Table	Schema	Unary Value	Pairwise Value
Source	99.5%	100%	98.5%
Target	98.5%	99.5%	97.0%

Table 3: Expert acceptance rate for the synthesized QA pairs sampled from our evaluations.

6.4 Expert Validation on Synthesized QAs

Lastly, we verify the reliability of synthesizing QA pairs with LLMs for evaluating tabular data. To achieve this, we conduct expert annotations by inviting the authors to manually inspect a random sample of 200 QA pairs covering schema, unary value, and pairwise value comparison aspects. They are asked to annotate (1) whether the generated QA pair is firmly grounded in the source table and (2) whether the LLM correctly answers it based on the target table. The expert acceptance rates are reported in Table 3. We observe that our LLM-synthesized QA pairs are highly reliable, with most acceptance rates above 98% for both source and target tables across schema, unary, and pairwise value comparisons. These results support our evaluation protocol, demonstrating that LLMs can effectively automate the assessment of semantically diverse tabular data.

7 Conclusions

In this work, we introduce an improved literature review table generation task that incorporates distractor papers and replaces table captions with abstract user demands to better align with real-world scenarios, and curated an associated benchmark. Additionally, we propose an annotation-free evaluation framework using LLM-synthesized QA pairs and a novel method to enhance table generation. Our experiments show that current LLMs and existing methods struggle with our task, while our approach significantly improves performance. We envision that our work paves the way for more automated and scalable literature review table generation, ultimately facilitating the efficient synthesis of scientific knowledge in large-scale applications.

Limitations

A minor limitation is that our work uses ArxivDIGESTables as the source of literature review tables for subsequent data reconstruction. However, Newman et al. (2024) have included their pipeline for scalably extracting literature review tables from scientific papers, thus resolving the data reliance gap. Another limitation of our work is its reliance on GPT-4o, a proprietary LLM, for benchmark curation and subsequent evaluation, which may introduce several issues. First, it raises concerns about data contamination (Deng et al., 2024; Dong et al., 2024), as the model may generate user demands (during benchmark curation) and synthesis evaluation questions (when evaluating a generated table against the ground truth) that are similar to its training data, potentially leading to inflated performance in table generation. A data provenance check (Longpre et al., 2024) can be further implemented to address this issue. Second, the benchmark and evaluation process may inherit the internal knowledge or semantic distribution biases of GPT-4o, which could skew the evaluation of other LLMs and reduce the generalizability of our findings. Lastly, a minor issue is scalability, as curating larger datasets using a proprietary model can be resource-intensive and may limit accessibility when extending our framework to other literature or domains. Future work can explore the use of open-source LLMs to replicate the entire process for convenient adaptation to other tabular datasets.

Ethics Statement

The ArxivDIGESTables (Newman et al., 2024) dataset used in our work is shared under the Open Data Commons License, which grants us access to it and allows us to improve and redistribute it for research purposes. Regarding language models, we access all open-source LMs via the Hugging Face Hub (Wolf et al., 2020) and proprietary GPT models through their official API[1]. The number of these models, if available, is marked in Table 2. All associated licenses for these models permit user access for research purposes, and we commit to following all terms of use.

When prompting GPT-4o to generate user demands and synthetic QA questions, we explicitly state in the prompt that the LLM should not generate any content that contains personal privacy violations, promotes violence, racial discrimination, hate speech, sexual, or self-harm contents. We also manually inspect a random sample of 100 data entries generated by GPT-4o for offensive content, and none are detected. Therefore, we believe that our dataset is safe and will not yield any negative or harmful impact.

Our human annotations are conducted by recruiting five graduate-level students who have sufficient experience in data collection for training large language models. They are proficient in English, primarily from Asia, and are paid above the minimum wage in their local jurisdictions. They receive thorough training on the task and are reminded to have a clear understanding of the task instructions before proceeding to annotation. The high level of inter-agreement also confirms the quality of our annotation. The expert annotators have agreed to participate as their contribution to the paper without receiving any compensation.

[1] https://platform.openai.com/

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6494–6506. Association for Computational Linguistics.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4599–4610. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui

Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 8706–8719. Association for Computational Linguistics.

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. Ms\^2: Multi-document summarization of medical studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7494–7513. Association for Computational Linguistics.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 12039–12050. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Hayato Hashimoto, Kazutoshi Shinoda, Hikaru Yokono, and Akiko Aizawa. 2017. Automatic generation of review matrices as multi-document summarization of scientific papers. In *Proceedings of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017) co-located with the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017), Tokyo, Japan, August 11, 2017*, volume 1888 of *CEUR Workshop Proceedings*, pages 69–82. CEUR-WS.org.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-In Lee, and Moontae Lee. 2023. QASA: advanced question answering on scientific articles. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19036–19052. PMLR.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2020. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 1918–1925. European Language Resources Association.

Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su. 2023. A sequence-to-sequence&set model for text-to-table generation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5358–5370. Association for Computational Linguistics.

Shayne Longpre, Robert Mahari, Naana Obeng-Marnu, William Brannon, Tobin South, Katy Ilonka Gero, Alex Pentland, and Jad Kabbara. 2024. Position: Data authenticity, consent, & provenance for AI are all broken: what will it take to fix them? In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Xinyuan Lu, Liangming Pan, Qian Liu, Preslav Nakov, and Min-Yen Kan. 2023. SCITAB: A challenging benchmark for compositional reasoning and claim verification on scientific tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7787–7813. Association for Computational Linguistics.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8068–8074. Association for Computational Linguistics.

Mistral-AI. 2024. Large enough. *Mistral AI Blog*.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Scigen: a dataset for reasoning-aware text generation from scientific tables. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Benjamin Newman, Yoonjoo Lee, Aakanksha Naik, Pao Siangliulue, Raymond Fok, Juho Kim, Daniel S. Weld, Joseph Chee Chang, and Kyle Lo. 2024. Arxivdigestables: Synthesizing scientific literature into tables using language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 9612–9631. Association for Computational Linguistics.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. *OpenAI*.

OpenAI. 2024b. Hello gpt-4o. *OpenAI*.

OpenAI. 2025a. Introducing deep research. *OpenAI Blog*.

OpenAI. 2025b. Openai o3-mini. *OpenAI Blog*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Daniel M. Russell, Mark Stefik, Peter Pirolli, and Stuart K. Card. 1993. The cost structure of sensemaking. In *Human-Computer Interaction, INTERACT '93, IFIP TC13 International Conference on Human-Computer Interaction, 24-29 April 1993, Amsterdam,* *The Netherlands, jointly organised with ACM Conference on Human Aspects in Computing Systems CHI'93*, pages 269–276. ACM.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Scieval: A multi-level large language model evaluation benchmark for scientific research. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19053–19061. AAAI Press.

Anirudh Sundar, Christopher Richardson, and Larry Heck. 2024. gtbls: Generating tables from text by conditional question answering. *CoRR*, abs/2403.14457.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2023. Struc-bench: Are large language models really good at generating complex structured data? *CoRR*, abs/2309.08963.

Xingbo Wang, Samantha L. Huey, Rui Sheng, Saurabh Mehta, and Fei Wang. 2024. Scidasynth: Interactive structured knowledge extraction and synthesis from scientific literature with large language model. *CoRR*, abs/2404.13765.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Xueqing Wu, Jiacheng Zhang, and Hang Li. 2022. Text-to-table: A new way of information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2518–2533. Association for Computational Linguistics.

Shuo Zhang and Krisztian Balog. 2018. On-the-fly table generation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 595–604. ACM.

Appendices

A Implementation Details

In this section, we provide additional implementation details about our benchmark curation and evaluation pipeline, including the prompt we used and the models we accessed.

A.1 Prompts Used

We first introduce the prompt used to construct the ARXIV2TABLE benchmark, as explained in Section 4. The main step involves prompting LLM is to collect user demands that describe the purpose of creating the table while remaining contextually self-contained and not revealing the actual schema or values of the table. We use the following prompt to instruct GPT-4o in generating these user demands.

```
Given a literature review table, along with its caption, you are tasked with writing a user demand or intention for the creator of this table. The user demand should be written as though you are instructing an AI system to generate the table. Avoid directly mentioning column names in the table itself, but instead, focus on explaining why the table is needed and what information it should contain. You may include a description of the table's structure, whether it requires detailed or summarized columns. Additionally, infer the user's intentions from the titles of the papers the table will include. Limit each user demand to 1-2 sentences. Examples of good user demands are: I need a table that outlines how each study conceptualizes the problem, categorizes the task, describes the data analyzed, and summarizes the main findings. The table should have detailed columns for each of these aspects. Generate a detailed table comparing the theoretical background, research methodology, and key results of these papers. You can use several columns to capture these aspects for each paper. I want to create a table that summarizes the datasets used to evaluate different GNN models, focusing on the common features and characteristics found across the papers listed below. The table should have concise columns to
```

highlight these dataset attributes. Now, write a user demand for the table below. The caption of the table is "<CAPTION>". The table looks like this:
<TABLE>
The following papers are included in the table:
<PAPER-1> . . . <PAPER-N>
Write the user demand for this table. Do not include the column names in the user demand. Write a concise and clear user demand covering the function, topic, and structure of the table with one or two sentences. The user demand is:

Then, for synthesizing QA pairs from a table, we use the following prompt to guide GPT-4o in generating some QA pairs with answers:

You will evaluate the quality of a generated table by comparing it against a ground-truth table. The goal is to assess whether the generated table correctly retains the schema, individual values, and pairwise relationships. This is achieved by generating targeted QA pairs based on the ground-truth table and answering them using the generated table. Step 1: QA Pair Generation Based on the Ground-Truth Table Generate binary (Yes/No) QA pairs focusing on three aspects: Schema QA Pairs: Check whether a specific column from the ground-truth table appears in the generated table schema. Example: Is Dataset included in the table schema? Unary Value QA Pairs: Check whether a specific cell value from the ground-truth table is present in the generated table. Example: Is CL, TL the loss function for paper CN-LexNet? Pairwise Value QA Pairs: Check whether a relationship between two values remains consistent in the generated table. Example: Is ResNet-v2 using more evaluation metrics than GAN? For Schema and Unary Value, generate a QA pair for every column and every cell, respectively. For Pairwise Value, randomly sample 10 pairs per table and construct the corresponding QA pairs. Step 2: Answering QA Pairs Using the Generated Table After generating the QA pairs, answer them using the generated table. Provide only "yes" or "no" responses: If the information is present in the generated table, respond with "yes." If the information is missing or different, respond with "no." Your task is to generate the QA pairs based on the ground-truth table and then answer them based on the generated table. Now, begin by generating the QA pairs.

The distribution of number of papers per table in ARXIV2TABLE is shown in Figure 6.

A.2 Evaluation Implementations

We access all open-source LLMs via the Hugging Face library (Wolf et al., 2020). The models used are `meta-llama/Llama-3.3-70B-Instruct`, `mistralai/Mistral-Large-Instruct-2411`, and `deepseek-ai/DeepSeek-V3`.

For GPT models, we access them via the official OpenAI Batch API[2]. The models used are `gpt-4o-mini-2024-07-18` and `gpt-4o-2024-08-06`.

Note that the DeepSeek model family has a context window limit of 64K tokens, whereas the others have a limit of 128K tokens. The generation temperature is set to 0.5 for all experiments. All experiments are repeated twice and the average performance is reported.

B Method Efficiency Evaluations

In addition to the empirical evaluation of the generated tables in Table 2, we also compare the efficiency of different methods based on their generation success rate and the average number of tokens used per table. The generation success rate refers to the average proportion of tables successfully generated within the context window limit of each backbone model. The statistics are reported in Table 4. Our observations indicate that while all baseline methods encounter issues with the context window limit, our schema induction method effectively mitigates this problem. Furthermore, our method achieves comparable token usage while delivering superior performance, highlighting its advantage.

C Annotation Details

To ensure the high quality of our human annotations, we implement strict quality control measures.

[2] https://platform.openai.com/docs/guides/batch

Method	GSR	#Tokens
Method 1	48.19%	128K
Method 2	98.23%	167K
Method 3	99.71%	110K
Ours	100.0%	118K

Table 4: Comparison of the efficiency of different methods. GSR stands for generation success rate.

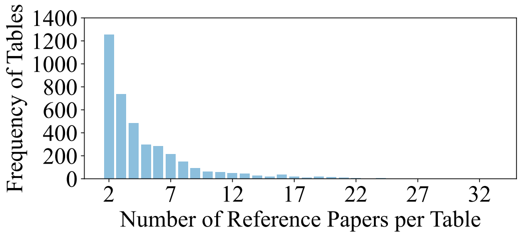

Figure 6: Distribution of number of papers in each table.

First, we select only postgraduate students with research experience in computer science to ensure they are familiar with relevant topics. All selected annotators undergo qualification rounds, and we invite only those who demonstrate satisfactory performance to serve as our main annotators.

For each task, we provide workers with comprehensive task explanations in layman's terms to enhance their understanding. Additionally, we offer detailed definitions and multiple examples for each choice to help annotators make informed decisions. Each entry requires the worker to provide a binary vote on whether the paper should be excluded or not. Our annotation interface is shown in Figure 7.

To ensure comprehension, we require annotators to confirm that they have thoroughly read the instructions by ticking a checkbox before starting the annotation task. We also manually monitor the performance of annotators throughout the annotation process and provide feedback based on common errors. Spammers or underperforming workers are disqualified. As described in Section 4.2, the inter-annotator agreement supports the quality of our collected annotations.

D Case Studies

Table 5 presents randomly sampled examples of original table captions alongside their improved user demands, demonstrating how refining vague captions enhances specificity and ensures more structured table generation. The findings highlight that well-defined user demands help capture key aspects of table construction, leading to more informative and targeted tabular representations.

Table 6 illustrates schema, unary value, and pairwise value questions designed to assess the quality of generated tables, ensuring alignment with ground-truth information. The results reveal that this QA-based evaluation effectively quantifies schema retention, individual value accuracy, and consistency in relationships, providing a structured approach for benchmarking table generation models.

Original Table Caption	User Demand
Comparison of Trajectory and Path Planing Approach	Generate a table that compares different trajectory and path planning approaches, focusing on their collision avoidance techniques, benefits, limitations, and applicable scenarios. The table should include detailed columns to capture these aspects for each method mentioned in the relevant papers.
Publications with deep-learning focused sampling methods. We cluster the papers based on the space the sample through and how the samples are evaluated. Some approaches further consider an optional refinement stage.	Create a table that categorizes publications focused on deep-learning-based sampling methods for grasp detection, organizing them by the space in which samples are generated, the evaluation criteria used, and whether a refinement stage is included. The table should provide a comprehensive yet concise overview of the methodological variations and enhancements across different papers.
Categorization of textual explanation methods.	Create a table that categorizes the methods used for providing textual explanations in visual question answering systems, focusing on the types of texts generated and the reasoning processes employed. The table should use succinct columns to differentiate between these methodological aspects for each paper.
Metadata of the three benchmarks that we focus on. XSumSota is a combined benchmark of cite:1400aac and cite:d420ef8 for summaries generated by the state-of-the-art summarization models.	Create a table that details the metadata for three summarization benchmarks, focusing on the composition of annotators, the dataset sizes for validation and testing, and the distribution of positive and negative evaluations. The table should provide a comprehensive comparison across these aspects for each benchmark.
Review of open access ground-based forest datasets	Create a table that reviews various open-access forest datasets, focusing on the publication and data recording years, types of data collected, and their applicability to specific forestry-related tasks. The table should offer a concise summary of each dataset's attributes, including the number of classification categories and geographical location.
Comparison of existing consistency-type models.	Create a table that compares different models focusing on their purpose, the trajectory they follow, the main objects they equate, and their methodological approach. The table should provide detailed insights into how each model addresses consistency issues, drawing from specified papers.

Table 5: Randomly sampled examples of the original captions and their corresponding improved user demands. Most captions are relatively short and may be vague without the full table's content.

Schema	Unary Value	Pairwise Value
Is Dataset included in the table schema?	Is CL, TL the loss function for paper CNLexNet?	Is ResNet-v2 using more evaluation metrics than GAN?
Is Model Architecture included in the table schema?	Is GPT-4o the model used for multimodal understanding?	Does GPT-4o have a larger parameter size than LLaMA-2?
Is Training Dataset included in the table schema?	Is ImageNet the dataset used for training ResNet?	Is ResNet trained on more samples than EfficientNet?
Is Performance Metric included in the table schema?	Is BLEU-4 the evaluation metric for MT-BERT?	Does BERT outperform LSTM on BLEU-4 score?
Is Activation Function included in the table schema?	Is ReLU the activation function used in Transformer?	Is GELU smoother than ReLU in function continuity?
Is Optimization Algorithm included in the table schema?	Is Adam the optimizer used for training BERT?	Does Adam converge faster than SGD for BERT training?
Is Pretraining Task included in the table schema?	Is Masked Language Modeling the pretraining task for BERT?	Does BERT use a more complex pretraining strategy than GPT?
Is Hyperparameter included in the table schema?	Is the learning rate set to 0.001 for training ViT?	Does ViT use a higher learning rate than ResNet?
Is Hardware Accelerator included in the table schema?	Is TPU used for training T5?	Do TPUs provide faster training than GPUs for T5?

Table 6: Randomly sampled examples of schema, unary value, and pairwise value questions used to evaluate the quality of generated tables. Each row contains three related questions derived from the same table.

Annotation Task

User Demand

"I need a table that summarizes the key characteristics of various benchmark datasets used in temporal knowledge graph reasoning, including the number of entities, relations, timestamps, and triplets for training, validation, and testing. The table should present this information in a concise manner to facilitate comparison across the studies represented."

Papers in the Current Table

# Entities	# Relations	# Timestamps	# Train Triplets	# Val. Triplets	# Test Triplets
500	20	366	2,735,685	341,961	341,961
15,403	34	198	110,441	13,815	13,800
125,726	203	1,700	323,635	5,000	5,000

Current Literature Review Table

Paper Arxiv Link	Title	Corpus ID
https://arxiv.org/pdf/2104.08419	TIE: A Framework for Embedding-based Incremental Temporal Knowledge Graph Completion	233295959
https://arxiv.org/pdf/1809.03202	Learning Sequence Encoders for Temporal Knowledge Graph Completion	52183483
https://arxiv.org/pdf/2112.05785	TempoQR: Temporal Question Reasoning over Knowledge Graphs	245124416

Paper to Be Decided

Title: Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks

Abstract: We present Wiki-CS, a novel dataset derived from Wikipedia for benchmarking Graph Neural Networks. The dataset consists of nodes corresponding to Computer Science articles, with edges based on hyperlinks and 10 classes representing different branches of the field. We use the dataset to evaluate semi-supervised node classification and single-relation link prediction models. Our experiments show that these methods perform well on a new domain, with structural properties different from earlier benchmarks. The dataset is publicly available, along with the implementation of the data pipeline and the benchmark experiments, at this https URL .

Link: https://arxiv.org/pdf/2007.02901

Decision

Based on the user demand and the existing literature review table, should this paper be included?

○ **Include** ○ **Exclude**

Submit

Figure 7: The annotation interface we used for collecting the gold labels for distractor papers.