TraceBack: Multi-Agent Decomposition for Fine-Grained Table Attribution

Anonymous TACL submission

Abstract

Question answering (QA) over structured tables requires not only accurate answers but also transparency about which cells support them. Existing table QA systems rarely provide fine-grained attribution, so even correct answers often lack verifiable grounding, limiting trust in high-stakes settings. We address this with TRACEBACK, a modular multi-agent framework for scalable, cell-level attribution in single-table QA. TRACEBACK prunes tables to relevant rows and columns, decomposes questions into semantically coherent sub-questions, and aligns each answer span with its supporting cells, capturing both explicit and implicit evidence used in intermediate reasoning steps. To enable systematic evaluation, we release CITEBENCH, a benchmark with phrase-to-cell annotations drawn from ToTTo, FetaQA, and AITQA. We further propose FAIRSCORE, a reference-less metric that compares atomic facts derived from predicted cells and answers to estimate attribution precision and recall without human cell labels. Experiments show that TRACEBACK substantially outperforms strong baselines across datasets and granularities, while FAIRSCORE closely tracks human judgments and preserves relative method rankings, supporting interpretable and scalable evaluation of table-based QA.

1 Introduction

Question answering (QA) is a core task in NLP with applications ranging from customer support to scientific analysis. Large language models (LLMs) have dramatically improved QA quality (Brown et al., 2020; Achiam et al., 2023), but they frequently hallucinate or produce factually incorrect statements (Ji et al., 2023), undermining reliability and user trust (Xu et al., 2025; Snyder et al., 2024). Even when answers are correct, the underlying reasoning may be spurious or opaque, motivating methods that make both answers and their evidence more transparent.

Source	Cost	Efficiency	Scalability
Solar Power	30–50	15–20	4
Wind Power	20–40	30–45	5
Hydropower	40–70	70–90	3
Geothermal	50–80	90+	2

Question: Among renewable sources costing $\leq$ 50/MWh and scalability $\geq$ 3, which is most efficient, and what is its efficiency?

Reasoning: Step 1: Cost Filter $\rightarrow$ Keep Solar, Wind. Step 2: Scalability Filter $\rightarrow$ Keep Solar, Wind. Step 3: Efficiency Selection $\rightarrow$ Choose Wind (30–45%).

S1: Cost $\leq$ 50 | S2: Scalability $\geq$ 3 | S3: Max Efficiency

Answer: Wind Power, 30–45% efficiency.

Figure 1: Example of fine-grained table attribution: intermediate filters (cost, scalability) and final selection (efficiency) each correspond to specific cells.

Attribution has therefore become central to evidence-based reasoning in LLMs, linking answers to verifiable sources (Gao et al., 2023a; Li et al., 2023). Most prior work focuses on unstructured text and relatively coarse citation; structured data such as tables introduce additional challenges due to schema constraints, hierarchical organization, and multi-step filtering (Figure 1).

For tabular QA, accurately identifying which cells support the answer is crucial for trust and interpretability. However, cell-level attribution remains largely unexplored. To date, the only prior work explicitly targeting table attribution,

MATSA (Mathur et al., 2024), operates at row-column granularity, lacks phrase-level alignment, and focuses solely on final answers. It does not expose intermediate evidence, such as start and end time cells used to compute a duration, even though these are essential steps in the reasoning process.

This raises a central question: *How can we design an interpretable, scalable framework that traces both answers and intermediate reasoning steps back to precise, cell-level evidence in structured tables?*

To address this, we introduce CITEBENCH, a benchmark for fine-grained attribution in table QA comprising 1,500 manually annotated examples from ToTTo (Parikh et al., 2020), FetaQA (Nan et al., 2022), and AITQA (Katsis et al., 2022). Each example includes cell-level attributions for both final answer spans and intermediate reasoning, enabling supervision for multi-hop and implicit inference.

Building on this resource, we propose TRACEBACK, a modular multi-agent LLM-based framework for post-hoc cell-level attribution over single tables. TRACEBACK sequentially (i) identifies relevant columns, (ii) filters rows via schema-aware conditions, (iii) decomposes questions into sub-questions aligned with intermediate reasoning steps, and (iv) grounds each sub-answer in specific cells, which are then aligned with answer spans (Figure 1). On the 1,500 annotated examples, TRACEBACK yields substantial gains over strong baselines in both fine-grained attribution and coverage of intermediate evidence.

Manual cell-level annotation is, however, expensive and subjective, making large-scale evaluation difficult. We therefore introduce FAIRSCORE, a reference-less metric that compares atomic facts extracted from predicted cells with those in the answer. By estimating attribution precision and recall via fact alignment, FAIRSCORE enables scalable, consistent, and interpretable evaluation across datasets, including unlabeled settings.

Our main contributions are:

1. We present CITEBENCH, a benchmark of manually annotated QA examples from ToTTo, FetaQA, and AITQA with cell-level attributions covering both final answers and intermediate reasoning steps.

2. We propose TRACEBACK, a multi-agent LLM-based framework for cell-level attribution that combines schema-aware pruning, question decomposition, and fine-grained cell grounding.

3. We develop FAIRSCORE, a reference-less evaluation metric based on atomic fact alignment that estimates attribution precision and recall without human cell labels.

4. We conduct extensive experiments and ablations, showing that TRACEBACK consistently outperforms strong baselines while FAIRSCORE tracks human judgments and preserves relative method rankings.

2 Related Work

Attribution in QA is central to improving the reliability, interpretability, and trustworthiness of large language models (LLMs), particularly in structured and multi-source settings. Early work, such as FEVER (Thorne et al., 2018), framed fact verification as attributing claims to supporting evidence, while (Evans et al., 2021) emphasized its role in truthful AI. Fake news detection has also been cast as an attribution task (Hanselowski et al., 2018). In citation evaluation, (Gao et al., 2023b) introduced ALCE, a benchmark assessing citation quality alongside answer fluency and correctness. Its reliance on MAUVE (Pillutla et al., 2021) for fluency scoring can yield unstable results due to sensitivity to output length and style, and correctness remains difficult to automate for multi-step answers.

For structured data, (Mathur et al., 2024) proposed MATSA, a multi-agent framework for table attribution with an 8.5K QA-pair benchmark from ToTTo (Parikh et al., 2020), FetaQA (Nan et al., 2022), and AITQA (Katsis et al., 2022). While effective on short contexts, MATSA is limited to row-column attribution and omits cell-level or long-context QA. Multi-source attribution typically follows two paradigms: joint response-and-citation generation (Menick et al., 2022; Thoppilan et al., 2022; Glaese et al., 2022), often restricted to single-source settings, and post-hoc attribution (Gao et al., 2023b; Yue et al., 2023), which struggles on human-annotated multi-source datasets. (Patel et al., 2024) addressed this with POLITICITE, a benchmark featuring multi-paragraph questions and human-labeled multi-source attributions.

Complementary to attribution methods, Inseq (Sarti et al., 2023) provides a toolkit for token- and sequence-level attribution in neural text generation, enabling analysis of model behavior via gradient- and attention-based explanations. While Inseq is designed for textual inputs and does not natively support structured tables or reasoning-aligned cell-level grounding, we repurpose its attribution primitives as part of our framework to operate over table representations.

To improve traceability, (Khalifa et al., 2024) proposed source-aware pretraining and instruction tuning by linking document IDs to citations, though adaptability to evolving knowledge remains limited. Attribution research has also expanded to multimodal QA, underscoring the need for explainable grounding across text, tables, and visual data.

Existing methods lack fine-grained attribution in structured data that captures both final answers and intermediate reasoning , critical for transparent multi-hop inference. Current benchmarks omit reasoning-aligned cell-level annotations, and evaluation remains costly, subjective, or unsuitable for scale. We address these gaps by introducing a benchmark with manual cell-level attributions for both final and intermediate steps, a modular multi-agent framework for post-hoc cell-level attribution, and a scalable reference-free metric for attribution quality.

3 CITEBENCH Benchmark

3.1 Limitations of the TabCite Benchmark

TabCite (Mathur et al., 2024) is a closely related benchmark for table-based citation. While it enables coarse-grained evaluation, several aspects of its design limit its suitability for *fine-grained* phrase-to-cell attribution.

Absence of phrase-cell alignment. In TabCite, attribution labels are defined at the level of the full answer, not individual answer phrases. As a result, the benchmark cannot verify whether each semantic unit in the answer is grounded in the correct cell.

For example, consider the answer *"Wind Power, with an efficiency between 30% and 45%"* with attributed cells $\{[1,0],[1,2]\}$. Without phrase-level alignment, the evaluation cannot distinguish the correct mapping *"Wind Power"* $\rightarrow [1,0]$, *"30%-45%"* $\rightarrow [1,2]$ from the incorrect mapping where these associations are reversed. Both are treated as equally valid, even though only the first is semantically correct. This prevents TabCite from assessing fine-grained grounding.

Noise in ground-truth attributions. TabCite is constructed from ToTTo, FetaQA, and AITQA. Re-examining these sources reveals substantial label noise. In a manual inspection of 500 randomly sampled examples per dataset, we observe error rates of 21.7% for *ToTTo* and 55.3% for *FetaQA*. Moreover, *AITQA* does not provide gold attribution labels, and its label construction procedure is undocumented.

Table 1 shows a representative FETAQA:
Question: In which films did Pooja Ramachandran play the role of Cathy?
Answer: Pooja Ramachandran starred as Cathy in *Kadhalil Sodhapuvadhu Yeppadi* and its Telugu version *Love Failure*.

While the relevant films and roles are correctly highlighted, the ground-truth attribution also marks several unnecessary cells, including other entries from the same year and an unrelated film (*Pizza*).

Year	Film	Role	Language
2002	Yathrakarude Sradakku	–	Malayalam
2012	Kadhalil Sodhapuvadhu Yeppadi	Cathy	Tamil
2012	Love Failure	Cathy	Telugu
2012	Nanban	Jeeva's Wife	Tamil
2012	Pizza	Smitha	Tamil
2013	Swamy Ra Ra	Bhanu	Telugu

Table 1: Example from FETAQA with ground-truth attribution highlighting (green = relevant, red = irrelevant).

$$\{[2,0],[2,1],[2,2],[3,0],[3,1],[3,2],[3,3],[5,0],[5,1]\}$$

These ground truth labels systematically over-include cells that are loosely related (e.g., same year, similar structure) rather than strictly necessary for the answer. This can encourage models to rely on partial or column-level matches instead of consistent row-level evidence.

Overall, the lack of phrase-level alignment and the presence of noisy or undocumented attributions limit TabCite's usefulness for evaluating fine-grained table grounding. These limitations motivate CITEBENCH, which provides explicit phrase-to-cell alignment and carefully curated, transparent ground-truth annotations.

3.2 CITEBENCH Construction

To address the aforementioned limitations of TabCite Mathur et al. (2024), we construct a new benchmark by reformulating and manually annotating examples from three public datasets: ToTTo (Parikh et al., 2020), FetaQA (Nan et al., 2022), and AITQA (Katsis et al., 2022).

ToTTo is an open-domain, Wikipedia-based table-to-text dataset. Because it does not contain natural-language questions, we follow the approach of MATSA Mathur et al. (2024) by treating the reference descriptions as answers and synthetically generating corresponding questions using DeepSeek-V3 (DeepSeek-AI and et al., 2025). ToTTo includes a wide range of table structures, such as merged cells, variable row counts, and nested headers, and thus captures much of the structural complexity found in real-world tables.

FetaQA consists of free-form, multi-hop question-answer pairs over Wikipedia tables. Many questions require aggregating evidence from multiple cells across different rows. Although the tables typically lack deeply nested headers, the dataset presents substantial reasoning challenges due to its long-form answers and diverse table content.

AITQA is a domain-specific question-answering dataset derived from SEC 10-K annual reports in the airline industry. Its tables contain complex column hierarchies, specialized financial terminology, and heterogeneous numerical formats, making the dataset representative of enterprise and scientific document settings.

Human Annotation. From each dataset, we manually annotate 500 randomly sampled examples (all 513 examples from AITQA), yielding a gold set with precise, phrase-aligned cell-level attributions (Table 2). For each example, every answer phrase is explicitly linked to the table cell(s) that support it, enabling fine-grained and interpretable evaluation. Inter-annotator agreement, measured using Cohen's κ on 250 randomly sampled instances across all three datasets, is 0.72, indicating substantial agreement.

The remaining ToTTo and FetaQA examples retain their original (and potentially noisy) attributions and form a silver set, which provides broader coverage for large-scale evaluation. Due to its smaller size, AITQA does not include a silver subset. Together, the gold and silver sets support both rigorous fine-grained evaluation and scalable benchmarking of attribution performance.

Dataset	Total Examples	Gold Set (Human-Annotated)	Silver Set (Original)
ToTTo	7,700	500	7,200
FetaQA	3,004	500	2,504
AITQA	513	513	–

Table 2: Dataset statistics for the CITEBENCH.

4 TRACEBACK Agentic Framework

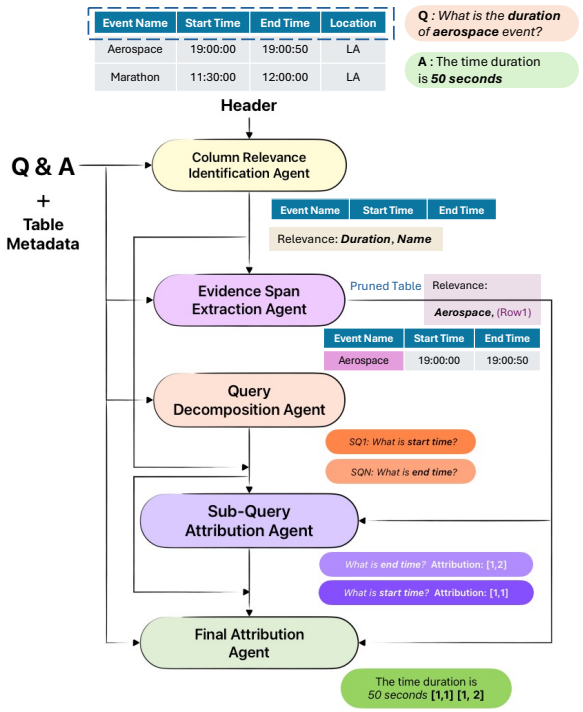

Figure 2: Architecture diagram of the TRACEBACK.

4.1 Task Formulation

We consider *cell-level structured single-table attribution*, where the goal is to identify the minimal yet sufficient set of table cells that support a model-generated answer.

Let T be a table with R rows and C columns, and let each cell be indexed as $T[i, j]$ for $i \in [1, R]$ and $j \in [1, C]$. Given a natural-language question q and its answer a, we seek an attribution set $\mathcal{A}_T \subseteq \{T[i, j]\}$ such that $f(T, q, a) \to \mathcal{A}_T$.

The set $\mathcal{A}_T$ must (i) be sufficient and non-contradictory for deriving a from T, and (ii) ground all semantic content in a. It may include both *explicit cells*, whose values appear verbatim

in a, and *implicit cells*, which are required for intermediate reasoning but not directly mentioned. As illustrated in Figure 1, correctly answering the efficiency question for *Wind Power* requires cost and scalability cells in addition to the final efficiency cell.

4.2 TRACEBACK Framework

To solve this task, we propose TRACEBACK, an LLM-based multi-agent framework in which each agent handles a distinct subtask in the attribution pipeline. The framework is designed for structured and semi-structured single tables and scales to arbitrary table size and schema. Figure 2 gives an overview.

1. Column Relevance Identification. This agent selects the subset of columns needed to derive the answer. Relevant columns may be *explicit*, whose values appear in the answer, or *implicit*, required for intermediate computations.

Given table metadata and schema, together with (q, a), the agent uses few-shot prompting to predict the full set of relevant columns (Figure 2). For example, for the question *"What is the duration of the event?"* with answer *"50 seconds"*, it must identify *Start_Time* and *End_Time* even though no column is labeled "duration." Similarly, in Figure 1, answering the constrained efficiency query requires *Cost*, *Scalability*, and *Efficiency*, even though only the efficiency value appears in the final answer.

2. Evidence Span Extractor. Given the selected columns, this agent identifies the subset of rows required to derive the answer. Following the SQL-based row extraction strategy of Abhyankar et al. (2025), it uses few-shot prompting to generate filtering conditions over the relevant columns and metadata.

In Figure 2, for the question *"What is the duration of the aerospace event?"*, the agent can generate a filter such as `WHERE Event_Name = 'Aerospace'`, retaining only the row for the Aerospace event. In Figure 1, more complex multi-step filtering is required: rows are filtered by *Cost* $\leq$ 50/MWh, then by *Scalability* $\geq$ 3, and finally ranked by *Efficiency*. This pruning reduces computational overhead while preserving the complete reasoning chain.

3. Query Decomposition. This agent decomposes the original question into semantically coherent sub-questions, each corresponding to an intermediate reasoning step needed to reconstruct the answer. The goal is to surface implicit dependencies that holistic attribution often misses.

Given table metadata, (q, a), and the pruned table, the agent generates sub-questions whose answers collectively entail the original response (Figure 2). For example, *"What is the duration of the event Marathon?"* may be decomposed into *"What is the start time of the event?"* and *"What is the end time of the event?"*, explicitly exposing the role of *Start_Time* and *End_Time*. To ensure faithfulness, each generated fact is checked by a pretrained NLI model (RoBERTa; Zhuang et al. 2021), following Mathur et al. (2024).

In Figure 1, the comparative efficiency question is similarly decomposed into sub-questions for cost filtering, scalability filtering, and efficiency selection, revealing how multiple columns contribute to the final answer.

4. Sub-Query Attribution. This agent performs fine-grained attribution by grounding each sub-question to its supporting cells. Given the sub-questions and the pruned table, it outputs cell coordinates (i, j) corresponding to the evidence used to answer each sub-query.

In Figure 2, sub-questions about start and end time are mapped to the *Start_Time* and *End_Time* cells in the relevant row; these cells are essential for computing the duration, even though they are absent from the answer text. In Figure 1, sub-questions are similarly grounded to the *Cost*, *Scalability*, and *Efficiency* cells that jointly determine the answer. Operating at the cell level allows the agent to capture such implicit evidence, which row- or column-level attribution would miss.

5. Final Attribution. The final agent consolidates all intermediate attributions into a single phrase-to-cell mapping. It aligns the cell coordinates produced by the Sub-Query Attribution agent with spans in the original answer, yielding a complete, interpretable grounding of each answer component in verifiable tabular evidence and enabling faithful evaluation of table-based QA systems.

5 Experimental Setup

Baselines. We evaluate our approach CITEBENCH and compare against the following baselines. Unless otherwise noted, all

methods use `GPT-4o` (OpenAI et al., 2024) as the LLM. Finally, all prompt of our method are provided in Appendix A

Few-shot ICL (Gao et al., 2023c) prompts the model with a small number of annotated examples and directly asks it to produce cell-level attributions for each question-answer pair.

Post-hoc Retrieval (Gao et al., 2023a) linearizes each table row into text and uses a dense retriever (Sentence-BERT (Reimers and Gurevych, 2019)) to rank rows by cosine similarity to the concatenated question and answer; the top-k rows (k=10) are then passed to the LLM to generate cell-level attributions.

GENERATEPROGRAMS (Wan et al., 2025) converts each row into atomic fact sentences, assigns each cell a unique identifier, and applies the original GENERATEPROGRAMS pipeline unchanged to produce executable programs and their induced attributions.

INSEQ-based Attribution (Sarti et al., 2023) repurposes token- and sequence-level attribution by linearizing tables into structured text and treating cell values as attribution units; gradient- and attention-based scores are aggregated at the cell level. For this baseline we use `Qwen3-30B-A3B` (Qwen et al., 2025a), as gradient-based attribution requires direct access to model parameters.

To our knowledge, no existing method natively supports reasoning-aligned cell-level attribution over structured tables. The closest framework, MATSA (Mathur et al., 2024), operates at row-column granularity and is not publicly available, precluding direct comparison.

Evaluation Metrics. We evaluate attribution quality at three levels of granularity: **row**, **column**, and **cell** using precision and recall computed from the overlap between predicted and gold-standard attribution sets.

Let R', C', and S' denote the sets of predicted rows, columns, and cells, respectively, and let $\hat{R}$, $\hat{C}$, and $\hat{S}$ denote the corresponding gold-standard sets. Precision measures the proportion of predicted elements that are correct, while recall measures the proportion of gold elements that are successfully retrieved.

Evaluating across multiple granularity enables analysis of attribution performance at both coarse and fine levels: row- and column-level metrics capture broader localization accuracy, while cell-level metrics assess fine-grained grounding fidelity.

Granularity	Precision	Recall
Row	$\frac{\lvert R' \cap \hat{R}\rvert}{\lvert R'\rvert}$	$\frac{\lvert R' \cap \hat{R}\rvert}{\lvert \hat{R}\rvert}$
Column	$\frac{\lvert C' \cap \hat{C}\rvert}{\lvert C'\rvert}$	$\frac{\lvert C' \cap \hat{C}\rvert}{\lvert \hat{C}\rvert}$
Cell	$\frac{\lvert S' \cap \hat{S}\rvert}{\lvert S'\rvert}$	$\frac{\lvert S' \cap \hat{S}\rvert}{\lvert \hat{S}\rvert}$

Table 3: Precision and Recall at different granularity.

5.1 Main Results

Table 4 reports precision and recall for row-, column-, and cell-level attribution on ToTTo, FetaQA, and AITQA. We highlight the main trends here and refer to the table for complete figures.

Row-Level Attribution. TRACEBACK achieves the best row-level performance on all three datasets. On ToTTo, it improves precision from a best baseline of 50.00 (GENERATIONPROGRAMS) to 71.19 and recall from 74.50 (INSEQ) to 80.38. On FetaQA, TRACEBACK attains 94.30 precision and 93.36 recall, a gain of roughly 19 precision points over the strongest baseline (75.00) and +8.7 recall over the best baseline (84.70). On AITQA, it reaches 96.65 precision versus 66.82 for the best baseline and matches the high-recall regime (97.12 vs. 97.60 for INSEQ), showing that schema-aware filtering avoids the over-retrieval that hurts baseline precision.

Column-Level Attribution. Column attribution is strong for several methods, but TRACEBACK is consistently competitive or best. On ToTTo, it nearly matches the highest baseline precision (91.50 vs. 92.70) while maintaining solid recall (77.64). On FetaQA, it achieves both the highest precision (96.39) and near-best recall (83.07 vs. 84.77 for SBERT+`GPT-4o`). On AITQA, TRACEBACK attains 54.09 precision and 98.09 recall, improving precision over the best baseline (49.86) while staying in the same high-recall regime as SBERT+`GPT-4o` and INSEQ.

Cell-Level Attribution. Cell-level attribution is the hardest setting, and here TRACEBACK shows the largest margins. On ToTTo, it raises precision from 42.70 and recall from 53.80 (both from INSEQ) to 74.20 and 67.05. On FetaQA, it im-

Method	ToTTo		FetaQA		AITQA	
	Precision	Recall	Precision	Recall	Precision	Recall
Row-Level Attribution						
SBERT + GPT-4o	43.38	39.09	57.02	55.40	66.82	68.10
GENERATIONPROGRAMS	50.00	31.28	75.00	40.06	59.68	71.90
Fewshot + CoT	17.30	12.80	34.40	30.10	04.30	04.30
INSEQ	37.50	74.50	56.40	84.70	31.20	**97.60**
TRACEBACK - Lite	77.00	62.60	83.00	79.20	91.30	92.90
TRACEBACK	**71.19**	**80.38**	**94.30**	**93.36**	**96.65**	97.12
Column-Level Attribution						
SBERT + GPT-4o	90.51	**85.91**	94.67	**84.77**	46.68	97.14
GENERATIONPROGRAMS	71.81	24.71	78.42	20.00	49.86	83.81
Fewshot + CoT	**92.70**	76.40	95.80	67.30	47.50	94.30
INSEQ	73.10	74.10	82.60	65.50	34.70	**99.98**
TRACEBACK - Lite	88.60	48.90	94.70	54.80	**79.20**	85.30
TRACEBACK	91.50	77.64	**96.39**	83.07	54.09	98.09
Cell-Level Attribution						
SBERT + GPT-4o	39.78	36.97	52.08	46.16	31.96	66.67
GENERATIONPROGRAMS	29.35	13.61	50.78	15.74	30.32	67.14
Fewshot + CoT	14.50	10.10	27.40	17.80	02.20	04.30
INSEQ	42.70	53.80	56.50	44.20	19.20	**97.10**
TRACEBACK - Lite	73.80	39.60	75.40	42.30	**73.70**	80.60
TRACEBACK	**74.20**	**67.05**	**89.81**	**78.84**	52.37	95.22

Table 4: **Attribution performance across granularities and datasets.** Precision and recall for row-, column-, and cell-level attribution on ToTTo, FetaQA, and AITQA. Blocks correspond to different granularities. For each dataset-granularity pair, **bold** best method; underline marks second best

proves precision from 56.50 to 89.81 and recall from 46.16 to 78.84, a gain of more than 30 points on both metrics. On AITQA, TRACEBACK boosts precision from 31.96 (SBERT+GPT-4o) to 52.37 while retaining very high recall (95.22 vs. 97.10 for INSEQ). These gains indicate that query decomposition and sub-query attribution are particularly effective at isolating the exact supporting cells, including implicit evidence.

Baseline Behavior and Ablations. Few-shot CoT generally underperforms, especially at the cell level, confirming that unguided prompting lacks the structured reasoning needed for fine-grained grounding. SBERT+GPT-4o and INSEQ often achieve strong recall (e.g., row-level recall of 97.60 on AITQA), but their precision remains low due to aggressive retrieval and limited schema awareness. GENERATIONPROGRAMS trades precision for recall without closing this gap. TRACEBACK-Lite (uses `Qwen2.5-7B-Instruct` (Qwen et al., 2025b) as LLM), which omits some components of our full framework, already outperforms all baselines in most settings, while the full TRACEBACK consistently delivers the best precision-recall balance across datasets and granularities.

5.2 Variants and Ablations

Table 5 reports cell-level precision and recall for variants of TRACEBACK and ablations of its main components.

Variants. Both pipeline variants underperform the full system. Running *query decomposition before table pruning* yields lower precision on all datasets (e.g., 63.00 vs. 74.20 on ToTTo), suggesting that pruning benefits from operating on the original table rather than on decomposed sub-questions. Processing *one subquery at a time* further reduces precision (61.32 on ToTTo and 69.67 on FetaQA), indicating that joint reasoning over sub-questions is important for maintaining consistent evidence selection. The full TRACEBACK achieves the best trade-off, with 74.20/67.05 on ToTTo and 89.81/78.84 on FetaQA.

Ablations. Removing *table pruning* slightly improves precision on AITQA (86.33 vs. 52.37) but reduces precision on ToTTo and FetaQA and lowers recall overall, confirming that pruning is generally beneficial for accuracy and efficiency. The largest degradation occurs when removing *query decomposition*: precision drops to 56.00 on ToTTo and 65.40 on FetaQA, with recall also reduced. This shows that decomposition is crucial for surfacing intermediate evidence and aligning attributions with multi-step reasoning.

Overall, these results indicate that both table pruning and query decomposition are necessary: pruning controls noise, while joint reasoning over decomposed sub-questions enables accurate cell-level grounding.

6 FAIRSCORE Metric

To reduce the cost of manually annotating cell-level attributions, we introduce FAIRSCORE, a *reference-less* evaluation pipeline that estimates attribution quality by comparing *atomic facts* derived from predicted cells with atomic facts extracted from the answer (Figure 3). The pipeline has three stages.

1. **Cell-to-fact conversion.** Each cell predicted as relevant is converted into a short, declarative *atomic fact* by combining its value with the corresponding column header and, when needed, a row key. For example, the cells (*Kadhalil Sodhapuvadhu Yeppadi*, *Cathy*) under columns (*Film*, *Role*) yield facts such as *"The role of Cathy appears in the film Kadhalil Sodhapuvadhu Yeppadi."* This step turns structured evidence into natural-language statements that can be compared to answer-derived facts.

2. **Answer-to-fact conversion.** The reference answer is decomposed into minimal, self-contained atomic facts using an LLM. For instance, the answer *"Pooja Ramachandran starred as Cathy in Kadhalil Sodhapuvadhu Yeppadi and its Telugu version Love Failure."* can be split into *"Pooja Ramachandran starred as Cathy in Kadhalil Sodhapuvadhu Yeppadi."* and *"Pooja Ramachandran starred as Cathy in the Telugu version Love Failure."* This yields fact set against which table-derived facts can be aligned.

3. **Fact alignment and comparison.** A scoring LLM compares the two fact sets and decides, for each pair, whether the cell-derived fact correctly supports an answer fact. Unmatched answer facts correspond to *recall errors* (missing attributions), whereas unmatched cell-derived facts correspond to *precision errors* (over-attribution). In this way, we can evaluate attribution quality without access to gold cell labels.

FAIRSCORE Precision and Recall. Let a be the number of answer-derived facts that are supported by at least one predicted cell-derived fact, and b the number of unsupported answer facts. Let c be the number of predicted cell-derived facts that correctly match an answer fact, and d the number of predicted facts that do not match any answer fact. We define

$$\text{Recall} = \frac{a}{a+b}, \qquad \text{Precision} = \frac{c}{c+d}.$$

Recall (answer→cell view) measures how completely the predicted attributions cover the answer content, while precision (cell→answer view) measures how many predicted evidential cells are actually needed for the answer. Together, these quantities provide an interpretable, scalable proxy for cell-level attribution quality.

6.1 FAIRSCORE Systematic Analysis

We next assess the reliability of FAIRSCORE as a surrogate for human evaluation (Figure 4). Our analysis considers two complementary aspects: (i) how closely FAIRSCORE precision and recall track true cell-level precision and recall when gold attributions are available, and (ii) how consistently FAIRSCORE scores distinguish correct from incorrect attributions when applied both to model predictions and to gold cell sets themselves.

1. Human agreement check. We first apply FAIRSCORE to the 1,500 gold examples, pairing human cell-level attributions with their reference answers. Since these labels are manually annotated, we expect scores close to 1.0. In this setting, FAIRSCORE attains 83.20 precision and 88.34 recall, indicating strong alignment with human judgments. The remaining gap is largely due to variability in LLM-generated atomic facts and occasional errors in the entailment-based alignment.

Method	ToTTo		FetaQA		AITQA	
	Precision	Recall	Precision	Recall	Precision	Recall
Variants of TRACEBACK						
Query Decomposition before Table Pruning	63.00	60.15	71.23	75.10	85.38	92.89
Passing One Subquery at a Time	61.32	60.10	69.67	73.33	**85.95**	92.00
TRACEBACK	**74.20**	**67.05**	**89.81**	**78.84**	52.37	**95.22**
Ablation Study on TRACEBACK						
TRACEBACK	**74.20**	**67.05**	**89.81**	**78.84**	52.37	**95.22**
- w/o Table Pruning	73.14	60.10	72.89	75.78	**86.33**	93.10
- w/o Query Decomposition	56.00	56.30	65.40	68.32	47.22	91.80

Table 5: **Variants and ablations of** TRACEBACK **for cell-level attribution.** Cell-level precision and recall on ToTTo, FetaQA, and AITQA for (top) pipeline variants that modify how sub-questions are processed and (bottom) ablations that remove specific modules. **Bold** indicate the best score.

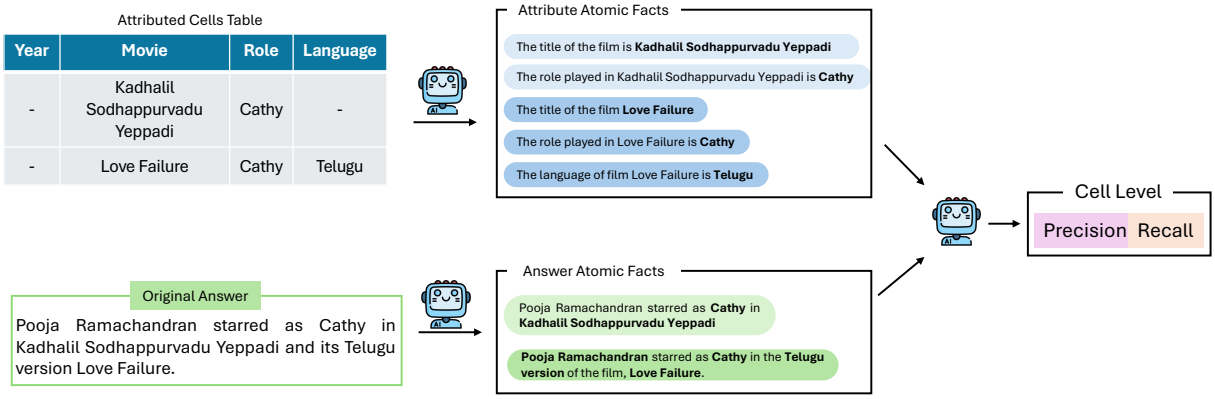

Figure 3: Illustration of Reference-less Evaluation Metric FAIRSCORE

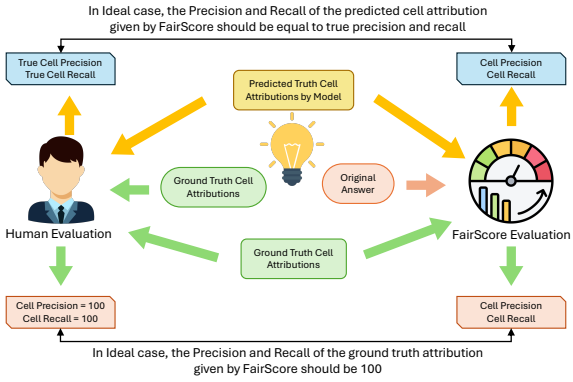

Figure 4: Systematic analysis of the metric - FAIRSCORE

2. Alignment with attribution evaluation. We next assess how well FAIRSCORE approximates true attribution quality when gold labels are used only for analysis. Table 6 compares FAIRSCORE-predicted precision and recall with human-annotated cell-level scores for TRACEBACK on the same 1,500 examples.

Analysis across datasets, FAIRSCORE systematically underestimates absolute scores but tracks relative trends. On ToTTo and FetaQA, precision is underestimated by about 16 points and recall by 6-11 points. AITQA exhibits a larger recall gap (-38.38), reflecting the greater linguistic and structural variability of financial tables, while precision remains within 4.3 points. Despite these discrepancies, FAIRSCORE preserves the correct ranking and relative differences, supporting its use as a scalable proxy for human cell-level evaluation.

Dataset	Pred P	Actual P	ΔP	Pred R	Actual R	ΔR
ToTTo	58.81	74.52	-15.71	56.36	67.31	-10.95
FetaQA	73.45	89.81	-16.36	72.40	78.84	-06.44
AITQA	48.10	52.38	-04.28	57.27	95.65	-38.38

Table 6: FAIRSCORE-predicted vs. human-annotated cell-level precision and recall for TRACEBACK's attributions. Δ columns show prediction error.

Taken together, these results indicate that FAIRSCORE closely mirrors human judgments of attribution quality. By converting predicted cells

and answers into atomic facts and aligning them via entailment-based matching, it captures both explicit and implicit evidence while penalizing omissions and hallucinations, enabling robust and scalable evaluation without additional gold labels.

6.2 Evaluation with FAIRSCORE

Method	ToTTo		FetaQA		AITQA	
	P	R	P	R	P	R
Fewshot + CoT	30.81	13.25	15.67	17.73	11.73	6.69
SBERT + GPT-4o	20.51	16.84	20.05	21.87	4.51	5.47
INSEQ	16.85	18.95	15.48	17.94	10.99	16.53
GP	14.96	11.32	16.04	14.13	7.62	3.77
TRACEBACK-Lite	53.87	40.20	51.88	45.12	**63.44**	**55.13**
TRACEBACK	**56.89**	**48.39**	**63.73**	**64.15**	42.20	49.93

Table 7: Cell-level precision and recall predicted by FAIRSCORE for different attribution methods on the gold sets of ToTTo, FetaQA, and AITQA. Note: here P refers to Precison and R refers to recall, Method GP refers to GENERATIONPROGRAMS.

6.3 Evaluation with FAIRSCORE

We report FAIRSCORE scores in terms of precision (P) and recall (R).

Methods evaluated with FAIRSCORE on gold sets. Table 7 shows FAIRSCORE scores for TRACEBACK, TRACEBACK-Lite, and all baselines on the 1,500 gold examples. On **ToTTo**, TRACEBACK reaches P/R of 56.89/48.39, improving over the strongest baseline (Few-shot ICL: 30.81 P; INSEQ: 18.95 R) by more than 25 points in P and nearly 30 in R. On **FetaQA**, it achieves 63.73/64.15, exceeding the best baseline (SBERT+GPT-4o: 20.05/21.87) by over 40 points on both metrics. On **AITQA**, TRACEBACK-Lite attains the highest scores (63.44/55.13), while the full TRACEBACK still substantially outperforms all non-TRACEBACK variants (42.20/49.93). Overall, FAIRSCORE clearly separates strong from weak attribution methods and preserves the performance ordering observed with gold-label evaluation.

TRACEBACK **evaluated with FAIRSCORE on silver sets.** We also apply FAIRSCORE to silver sets instances without human annotations from ToTTo (7,200 examples) and FetaQA (2,504 examples). On ToTTo, TRACEBACK attains 55.10/76.12; on FetaQA, it reaches 69.30/68.93. These scores are consistent with the corresponding gold-set estimates (56.89/48.39 for ToTTo and 63.73/64.15 for FetaQA) and preserve relative trends across datasets. Taken together, these results indicate that FAIRSCORE is a reliable reference-less proxy for human evaluation, enabling large-scale benchmarking of cell-level attribution when gold labels are scarce.

7 Conclusion and Future Work

We presented TRACEBACK, a modular multi-agent framework for fine-grained cell-level attribution in table QA, and CITEBENCH, a 1,500-example benchmark with phrase-to-cell annotations drawn from ToTTo, FetaQA, and AITQA. We also introduced FAIRSCORE, a reference-less metric based on atomic fact alignment that enables scalable evaluation without gold cell labels. Experiments show that TRACEBACK consistently outperforms strong baselines across datasets and granularities, and that FAIRSCORE closely tracks human judgments while preserving relative performance trends.

Future work includes extending TRACEBACK to more complex reasoning (e.g., aggregation and multi-hop over multiple tables) and improving FAIRSCORE 's robustness so it can be applied more broadly across domains and modalities.

Limitations

TRACEBACK has several limitations. First, it is restricted to single-table settings and does not yet support multi-table joins, hierarchical schemas, or semi-/multimodal tables, limiting applicability to more complex QA scenarios. Second, its core components (e.g., query decomposition and cell attribution) rely heavily on LLMs, making performance sensitive to prompt design, model changes, and domain shift. Third, while FAIRSCORE is scalable, it depends on the quality of fact extraction and entailment-based alignment and tends to underestimate absolute precision, especially for compositional or ambiguous answers. Fourth, our comparisons are constrained by the lack of publicly available implementations for some close baselines (e.g., MATSA). Finally, CITEBENCH currently covers only English Wikipedia-style tables, so generalization to multilingual or domain-specific and more heterogeneous settings remains untested.

References

Nikhil Abhyankar, Vivek Gupta, Dan Roth, and Chandan K. Reddy. 2025. H-STAR: LLM-driven hybrid SQL-text adaptive reasoning on tables. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8841–8863, Albuquerque, New Mexico. Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

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

DeepSeek-AI and Aixin Liua et al. 2025. Deepseek-v3 technical report.

Owain Evans, Owen Cotton-Barratt, Lukas Finnveden, Adam Bales, Avital Balwit, Peter Wills, Luca Righetti, and William Saunders. 2021. Truthful AI: Developing and governing AI that does not lie.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023c. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12).

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan,

Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2022. AIT-QA: Question answering dataset over complex tables in the airline industry. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 305–314, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Muhammad Khalifa, David Wadden, Emma Strubell, Honglak Lee, Lu Wang, Iz Beltagy, and Hao Peng. 2024. Source-Aware Training Enables Knowledge Attribution in Language Models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023. A Survey of Large Language Models Attribution.

Puneet Mathur, Alexa Siu, Nedim Lipka, and Tong Sun. 2024. MATSA: Multi-agent table structure attribution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 250–258, Miami, Florida, USA. Association for Computational Linguistics.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, and Aditya Ramesh et al. 2024. Gpt-4o system card.

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.

Nilay Patel, Shivashankar Subramanian, Siddhant Garg, Pratyay Banerjee, and Amita Misra. 2024. Towards improved multi-source attribution for long-form answer generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3906–3919, Mexico City, Mexico. Association for Computational Linguistics.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*.

Qwen, : An Yang, and Anfeng Li et al. 2025a. Qwen3 technical report.

Qwen, : An Yang, and Baosong Yang et al. 2025b. Qwen2.5 technical report.

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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, and Oskar van der Wal. 2023. Inseq: An interpretability toolkit for sequence generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 421–435, Toronto, Canada. Association for Computational Linguistics.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2024. On Early Detection

of Hallucinations in Factual Question Answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 2721–2732, New York, NY, USA. Association for Computing Machinery.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

David Wan, Eran Hirsch, Elias Stengel-Eskin, Ido Dagan, and Mohit Bansal. 2025. GenerationPrograms: Fine-grained Attribution with Executable Programs. In *Second Conference on Language Modeling*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2025. Hallucination is Inevitable: An Innate Limitation of Large Language Models.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635, Singapore. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Appendix

A Prompt Templates

Prompt A: Column Relevance Identification Agent

```
You will be given table title, column names, question and answer. You need to find all the
    ↪ columns in column names, which are relevant to question and answer.

<Example-1> :

Input :

<Table Title>: "Kings Time Duration Info"

<Column Names>: ["King Name", "Start of the Reign", "End of the Reign", "Duration", "
    ↪ Kingdom"]

Question : "How long did king Rajat held the throne and when it ended?"
Answer : "King Rajat held the throne for 10 years and it ended at 1777." :


Final Answer :
<Relevant Columns>: ["Duration","End of the Reign"]

</Example-1>

<Instructions>
1. Follow the given format for the final answer.
2. Don't miss any relevant columns.

Now, give output for the following inputs.

Input :

Table Title: {table title}

<Column Names>: {columns}

Question: {question}

Answer: {answer}

Output :
```

B Ethics Statement

We affirm that this work adheres to established ethical standards in NLP research and publication. All datasets used in our study ToTTo, FetaQA, and AITQA are publicly available and used in compliance with their respective terms. Our benchmark, CITEBENCH, is constructed through manual annotation with care to ensure fairness, factual accuracy, and privacy. We disclose all model configurations, prompting strategies, and evaluation protocols to support transparency and reproducibility. While FAIRSCORE is designed to eliminate the need for human labels, it depends on LLM judgments, which may reflect inherent biases; we mitigate this by using temperature-controlled generation and standardized prompts. AI assistance was used in experiments and drafting to improve clarity, structure, and analysis quality, with human oversight throughout.

Prompt B: Evidence Span Extractor Agent

```
Given a table schema and other table meta-data, question based on table and answer for the question, write a MySQL query
to retrieve all the relevant rows of the table, such that the answer to the given question can be generated from the
retrieved rows. Your focus should be to not eliminate any relevant rows from the table. Think step-by-step.

Important MySQL formatting rules:
- Use backticks (`...`) around table names and column names (identifiers), especially if they contain spaces or punctuation.
- Put ONLY the SQL inside the <SQL> ... </SQL> tags (do not add a leading colon after <SQL >).

<Example-1> :

Input :
Table-Schema :
<Column Names>: ["King Name", "Start of the Reign", "End of the Reign", "Duration"]
<Table Title>: "Kings\_Time\_Duration\_Info"
<Table-Rows> : [["Bhopal","Devanshu","1901","1998","98 Years"],["Chennai","Poojah","1999","2100","101"],["Ujjain","Rajat","2111","2211","100"]]

Question : "How long did king Poojah held the throne and who was his successor?"
Answer : "King Poojah held the throne for 101 years and King Rajat was his successor."

Output :
<step-by-step reasoning>
1. In the question, the duration and the successor is asked for King Poojah.
2. In the answer, duration of King Poojah is mentioned along with the sucessor to King Poojah - King Rajat.
3. The SQL query will be such that it retrieves the row of King Poojah and all the rows that has the "Start of the Reign" >= 2100 ("End of the Reign" for King Poojah).
Final Answer :
<SQL> CREATE TABLE Pooja\_King AS SELECT * FROM `Kings\_Time\_Duration\_Info` WHERE `King Name` = 'Poojah' OR CAST(`Start of the Reign` AS SIGNED) >= 2100; </SQL>
</Example-1>

<Instructions>
1. Follow the given format for the final answer. The final SQL query should be between <SQL > </SQL> tags.
2. Only write CREATE MySQL statements.
3. Give a SQL statement for MySQL.
4. The query giving the filtered table should cover all parts of the question and answer.

Now, give output for the following input.

Input :

Table-Schema :

<Table Title>: {title}

<Column Names>: {relevant columns}

Table: {table with relevant columns}

Question: {question}

Anser: {answer}

Output :
```

Prompt C: Query Decomposition Agent

```
Given the table information, a question based on the table, and its corresponding answer,
    ↪ break down the original question into smaller sub-questions such that each sub-
    ↪ question can be directly answered using specific parts of the table. The combined
    ↪ answers to these sub-questions should comprehensively cover all the information in
    ↪ the original answer. Ensure that each sub-question focuses on a specific aspect or
    ↪ data point present in the table.

Input Format:

Table-Info: Includes the table schema (column names), table title, and section title if
    ↪ available.
Question: A natural language question based on the table.
Answer: The corresponding answer derived from the table.

Output Format:

A list of sub-questions that together lead to the original answer.
Each sub-question should correspond to specific columns or rows of the table.

<Example-1>
Input :
Table-Schema :
<Column Names>: ["King Name", "Start of the Reign", "End of the Reign", "Duration"]
<Table Title>: "Kings Time Duration Info"

Question : "How long did king Rajat held the throne and who was his successor"
Answer : "King Rajat held the throne for 10 years and King Aayush was his successor."

Output :
Sub-Questions:
1. What was the duration of King Rajat's reign?
2. What was the End of the Reign year of King Rajat?
3. Which king started his reign at the end of King Rajat's reign?

</Example-1>

<Instructions>
1. Just give the sub-questions in the given format. Don't return anything else.


Now, give output for the following input.

Input :

Table-Schema :
<Column Names>: {column}
Table Title: {title}
Question: {question}
Answer: {answer}
Output :
```

Prompt D: Sub-Query Attribution Agent

```
You are given a pruned table (only the relevant columns and the relevant rows), the
    ↪ original Answer, plus a list of sub-questions.

Your task: for EACH sub-question, return the minimal set of table cells (row, column)
    ↪ needed to answer that sub-question, considering the context of the original Answer.
Include cells used for filtering/comparisons/calculations even if they are not explicitly
    ↪ mentioned in the final answer text.

Indexing rules:
- Use 0-based indexing for BOTH rows and columns.
- Row indices refer to <Table Rows> (the pruned rows, header excluded).
- Column indices refer to <Relevant-Columns> (the ordered list provided).

Output format:
- Return one line per sub-question, in the same order as given:
  <Cells>: [(row\_index, col\_index), (row\_index, col\_index), ...]
- If a sub-question has no evidence in the table, output:
  <Cells>: []
- Do not output anything other than these <Cells>: ... lines.

Input:
<Relevant-Columns>: {columns}
<Table Rows>: {pruned_rows}
Answer: {answer}
<Sub-Questions>:
{sub-questions}
```

Prompt E: Final Attribution Agent

```
You are given:
- The original question and its final (ground-truth) answer.
- A pruned table (<Relevant-Columns> and <Table Rows>, header excluded).
- A list of sub-questions.
- A set of candidate evidence cells gathered from the sub-questions (optional).

Your task: output the final set of evidence cells (row, column) that justify the final
    ↪ answer.
Important:
- Include implicit evidence cells needed for the reasoning chain (e.g., filter conditions,
    ↪ intermediate values), even if they are not verbatim in the answer.
- Exclude irrelevant cells.

Indexing rules:
- Use 0-based indexing for BOTH rows and columns.
- Row indices refer to <Table Rows> (the pruned rows, header excluded).
- Column indices refer to <Relevant-Columns> (the ordered list provided).

Output format:
- Output a single line:
  <Final Cells>: [(row\_index, col\_index), (row\_index, col\_index), ...]
- If no evidence is found, output:
  <Final Cells>: []
- Do not output anything else.

Input:
<Relevant-Columns>: {cols}
<Table Rows>: {rows}
Question: {question}
Answer: {answer}
<Sub-Questions>:
{sub-questions}
<Candidate Cells>: {cell_indexs}

Output:
```