Enhancing Temporal Understanding in LLMs for Semi-structured Tables

Irwin Deng[#], **Kushagra Dixit**[*], **Dan Roth**[#], **Vivek Gupta**[†]
[#]University of Pennsylvania, [*]University of Utah, [†]Arizona State University
`{ideng, danroth}@seas.upenn.edu, kushagra.dixit@utah.edu, vgupt140@asu.edu`

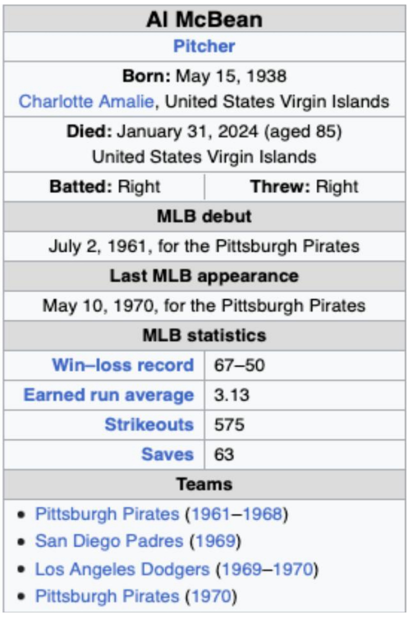
Abstract

Temporal reasoning over tabular data presents substantial challenges for large language models (LLMs), as evidenced by recent research. In this study, we conduct a comprehensive analysis of temporal datasets to pinpoint the specific limitations of LLMs. Our investigation leads to enhancements in TempTabQA, a benchmark specifically designed for tabular temporal question answering. We provide critical insights for enhancing LLM performance in temporal reasoning tasks with tabular data. Furthermore, we introduce a novel approach, C.L.E.A.R to strengthen LLM capabilities in this domain. Our findings demonstrate that our method improves evidence-based reasoning across various models. Additionally, our experimental results reveal that indirect supervision with auxiliary unstructured data (TRAM) substantially boosts model performance in these tasks. This work contributes to a deeper understanding of LLMs' temporal reasoning abilities over tabular data and promotes advancements in their application across diverse fields.

1 Introduction

Large Language Models (LLMs) have exhibited remarkable proficiency across various natural language processing tasks. However, recent investigations reveal a notable deficiency in their ability to reason effectively over *tabular* data, particularly when *temporal* relationships are involved (Chen, 2023; Sui et al., 2024). This discrepancy between model performance and human-level understanding underscores the pressing need for innovative approaches to enhance the capabilities of LLMs in this domain.

To identify the limitations of large language models (LLMs) in reasoning over structured and semi-structured tables, such as the one presented in Figure 1, we conducted a detailed analysis of the TempTabQA (Gupta et al., 2023) dataset. This investigation uncovered key discrepancies in model comprehension, prompting us to develop an enhanced evaluation set. Additionally, we explored how LLMs process temporal reasoning using standard prompts, which inspired the creation of a novel approach: C.L.E.A.R (Comprehend, Locate, Examine, Analyze, Resolve). This framework is designed to enhance temporal reasoning in LLMs.



Al McBean
Pitcher
Born: May 15, 1938
Charlotte Amalie, United States Virgin Islands
Died: January 31, 2024 (aged 85)
United States Virgin Islands
Batted: Right **Threw:** Right
MLB debut
July 2, 1961, for the Pittsburgh Pirates
Last MLB appearance
May 10, 1970, for the Pittsburgh Pirates
MLB statistics

Win–loss record	67–50
Earned run average	3.13
Strikeouts	575
Saves	63

Teams
- Pittsburgh Pirates (1961–1968)
- San Diego Padres (1969)
- Los Angeles Dodgers (1969–1970)
- Pittsburgh Pirates (1970)

Q1 : How many MLB wins did Al McBean have before turning 20? A: 0 (evidence: Born, MLB statistics)

Q2 : For how many years did Al McBean play for the Pittsburgh Pirates? A: 9 (evidence: Teams)

Q3 : What was the duration of Al McBean's Major League Baseball career? A: 10 (evidence: MLB debut, Last MLB appearance)

Figure 1: A semi-structured table of Baseball pitcher Al McBean with follow up question answers.

Although prompts help guide models, they don't inherently improve task understanding. Our research shows that even with C.L.E.A.R, models still miss crucial context, like relevant table rows, and often rely on memorization rather than evidence-based reasoning. This indicates a need for more robust solutions that lead to inherent im-

---

[*] Work done during internship at UPenn.

[†] corresponding author, Work done as a PostDoc at Upenn

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 4951–4970
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

provements in model parameters.

To tackle these challenges, we propose fine-tuning models on temporal data to enhance their reasoning capabilities. We utilized the TRAM dataset for indirectly supervised fine-tuning, incorporating Auxiliary Out of Domain (OOD) unstructured data. This cross-structured approach improves the model's generalization for structural temporal questions. Our findings indicate that integrating C.L.E.A.R with fine-tuning on TRAM—a straightforward temporal multiple-choice QA benchmark—produces optimal results, effectively addressing the limitations of LLMs in cross-format temporal reasoning. In summary, this paper advances the understanding and enhancement of LLMs' abilities to reason about temporal relationships in tabular data. Our key contributions are as follows:

- We conducted a in-depth analysis of the TempTabQA test set to assess the limitations of current approaches. This analysis also led to an improved test set, enhancing the evaluation of models in temporal tabular reasoning.

- We introduce an novel approach, C.L.E.A.R (Comprehend, Locate, Examine, Analyze, Resolve), designed to enhance temporal reasoning capabilities in LLMs for tabular data. Our approach grounds models in evidence, thus reducing memorization.

- By employing an indirect supervision technique, also known as auxiliary task training on unstructured data, we inherently enhance model performance. Fine-tuning models with the auxiliary TRAM dataset results in significant improvements in addressing temporal questions related to tabular data, thereby demonstrating cross-structure transferability.

These contributions enhance LLMs' temporal reasoning capabilities, paving the way for future research and applications in temporal reasoning with tabular data. Our code and enhanced evaluation set are available at `https://clear-prompting.github.io/`.

2 Motivation

Tables organize information chronologically, which is essential for analyzing timelines and facilitating temporal reasoning. Entity-centric tables, such as Wikipedia Infoboxes, blend structured and unstructured data, setting them apart from traditional SQL tables and knowledge graphs. This semi-structured data is heterogeneous and organized in key-value pairs without a defined hierarchy.

Despite their significance, recent research indicates that Large Language Models (LLMs) struggle with temporal reasoning in tabular data. Datasets such as TempTabQA and Table2vec (Zhang et al., 2019) have been developed to evaluate LLM performance, each presenting unique challenges for model development and evaluation. However, these datasets expose gaps in how models navigate the complexities of temporal reasoning. While symbolic reasoning has shown effectiveness for structured tables (Cheng et al., 2023; Ye et al., 2023; Wang et al., 2024), entity-centric heterogeneous semi-structured tables introduce further complexity. This complexity necessitates an understanding of both row-level key-value pairs and the implicit relationships that extend across the entire table, requiring advanced reasoning that combines domain knowledge and common sense (Gupta et al., 2020). Additionally, the temporal and tabular dimensions further complicate the reasoning process.

Moreover, while temporal reasoning has been examined in studies such as (Tan et al., 2023) and (Tan et al., 2024), these primarily focus on unstructured data. These studies do not adequately assess a model's ability to generalize from task-specific patterns to complex structural ones. To address this limitation, our work broadens logical reasoning across diverse data structures, demonstrating that temporal reasoning learned from unstructured data can effectively transfer to structured formats, thereby showcasing the adaptability of LLMs. Additionally, we utilize the TempTabQA dataset for our analysis, a benchmark specifically designed for temporal reasoning over semi-structured tabular data derived from Wikipedia Infoboxes. This dataset features over 2,900 temporal question-answer pairs, divided into "Head" (popular) and "Tail" (less common) sets, providing a balanced framework for evaluating model performance across both frequent and niche domains. Figure 1 illustrates examples of entity centric table based temporal questions in TempTabQA.

3 Where do LLMs Fail?

We evaluated the performance of GPT-3.5 using Chain of Thought (CoT) prompting on the TempTabQA test set. Of the 1,038 examples tested, 339 responses were incorrect, with errors split into two main types: data issues (159) and model limitations (180). The model limitations are categorized as follows:

1. Tabular Data Issues (75 examples): These errors were related to hallucinations, incomplete evidence extraction, missing evidence, or incorrect information extraction.

2. Temporal Calculation Errors (84 examples): These involved difficulties with calculations related to time, such as determining age, calculating the time between dates in different months, or assessing whether a value fell within a specified range.

3. Other Errors (31 examples): This category included errors stemming from common-sense reasoning, arithmetic, and other miscellaneous issues.

To address these issues, we used the error analysis to develop the C.L.E.A.R prompting method and refine the evaluation set. Model errors guided the design of each C.L.E.A.R step, while dataset errors were corrected. For instance, the "Locate" step resolves incomplete evidence extraction, and the "Examine" step simplifies temporal calculations into sub-questions. These details are outlined in Section 4.1.

Our observations reveal that even with Chain of Thought reasoning, models consistently generate incorrect responses and hallucinations, struggling with temporal calculations and common-sense reasoning. This highlights the need to improve model performance in this area. The 159 data issues fall into the following categories:

1. Tables Requiring External Knowledge to Answer (75 examples): These questions could not be answered correctly without additional information not present in the table, indicating a gap in the information provided in the context.

2. Wrong Human Annotation or Multiple Correct Answers (42 examples): These instances involved incorrect annotations by humans or questions that had multiple valid answers, but the annotations provided only one correct answer.

3. Ambiguous or Incomplete Questions (14 examples): These questions are either vague or lacked sufficient detail required for correct answer.

4. Other Issues (28 examples): This category included various problems such as questions relying on images within the HTML table or missing rows in the JSON table etc.

We addressed all data issues and created a refined evaluation set to better assess model performance. In refinement process, incorrect answers were corrected, and vague or ambiguous questions were reworded to be more specific, such as changing "when" to "in what year" or "on what date." This adjustment ensures clarity and fairness by preventing biases toward certain answer formats, which could impact prompting methods. Additionally, unanswerable questions—where the required information was missing from the provided table—were replaced with new ones of similar difficulty that require reasoning rather than favoring models with stronger memorization capabilities. Duplicate questions were removed and substituted with unique yet comparable ones to maintain dataset integrity.

To further enhance fairness, excessively verbose answers were refined by removing unnecessary information, ensuring responses remain focused and precise while reducing inconsistencies from noisy exact match checking. These modifications make the dataset solvable through reasoning rather than rote memorization, preventing any model from gaining an undue advantage. This new dataset removes noise, offering a more accurate benchmark. Importantly, all refinements were made before implementing the C.L.E.A.R. prompting method or fine-tuning techniques, ensuring that no systematic bias was introduced in favor of the proposed approach. Examples of this process are shown in Figures 8, 9, and 10. Throughout this paper, unless otherwise stated, we compare models based on accuracy on the new full test dataset, ensuring consistency and a clear assessment of improvements.

4 Methodology

Improving LLMs' temporal reasoning is crucial for time-based tasks, as current models often misinterpret temporal data. To address this, we developed C.L.E.A.R (Comprehend, Locate, Examine, Analyze, Resolve) to reduce errors and enhance accuracy. Additionally, we fine-tuned models with auxiliary temporal data to improve cross-generalization and performance on complex tasks.

4.1 C.L.E.A.R Prompting

To address the specific limitations of Large Language Models (LLMs) in temporal reasoning over semi-structured data, we propose the C.L.E.A.R (Comprehend, Locate, Examine, Analyze, Resolve) prompting method. As outlined in Section 3, our error analysis revealed several key challenges, including hallucinations, incomplete evidence extraction, and difficulties with temporal calculations. The C.L.E.A.R methodology was designed to directly target these issues, offering a structured approach to systematically answer complex temporal questions.

(a.) Comprehend: The Comprehend step ensures full understanding of the question by applying relevant domain knowledge, focusing on the correct interpretation of its temporal aspects and identifying essential information. For example, in "How many MLB wins did Al McBean have before turning 20?" (Fig 4), the model must calculate wins based on his MLB debut date before age 20.
This step reduces hallucinations and misinterpretations by grounding reasoning in a clear understanding of the question's scope and requirements, minimizing errors from task misinterpretation or irrelevant information extraction (Section 3).

(b.) Locate: The Locate step focuses on identifying and extracting the relevant rows from the table needed to answer the question. For the Al McBean example, this involves finding rows with his MLB debut date and win-loss record, ensuring all necessary evidence is gathered accurately.
This step tackles incomplete evidence extraction (Section 3) by ensuring the model isolates the correct data. By accurately extracting relevant rows, the model reduces errors from missing or incorrect information.

(c.) Examine: The Examine step breaks down the main question into smaller sub-questions to simplify complex temporal reasoning. For the Al McBean example, we might ask: (1) When did he turn 20? and (2) How many wins did he have by that date? This approach makes complex calculations more manageable.
This step addresses temporal calculation errors (Section 3), such as issues with date ranges and age calculations. Simplifying temporal reasoning into sub-questions reduces the likelihood of calculation mistakes.

(d.) Analyze: In this step, sub-questions are answered using the extracted evidence, applying logical reasoning to connect evidence to answers. For example, if Al McBean turned 20 on May 15, 1958, and debuted on July 2, 1961, he had 0 wins before turning 20.
This step ensures answers are logically consistent with the evidence, addressing issues like faulty deductions or temporal errors (Section 3), preventing misinterpretation of the data.

(e.) Resolve: In this final step, the model combines the sub-question answers to provide the final response. For example, in the Al McBean case, the final answer is 0 wins before turning 20. This step also includes a brief explanation to ensure the answer is clear and well-supported.
The Resolve step ensures the final answer is logically sound and free of errors, addressing potential inconsistencies from earlier stages (Section 3) and preventing error propagation.

Conclusion: The C.L.E.A.R prompting method systematically addresses common LLM errors in temporal reasoning through a structured, step-by-step approach that targets specific weaknesses (Section 3). By emphasizing thorough comprehension, precise evidence extraction, logical analysis, and well-supported answers, C.L.E.A.R enhances LLM performance on temporal tasks involving semi-structured tables. Unlike approaches prone to memorization, C.L.E.A.R fosters deeper reasoning, improving both robustness and effectiveness in temporal inference. Moreover, in contrast to iterative methods such as DATER, which require multiple inferences and prompts to generate a final result, C.L.E.A.R achieves this in a single inference (one prompt). This significantly reduces both inference time and computational costs, making it a more efficient alternative for temporal reasoning tasks.

4.2 Fine Tuning with Auxiliary Data

Fine-tuning improves models by adjusting their parameters with task-specific examples. We show that temporal reasoning benefits not only from fine-tuning on specific data but also by integrating auxiliary sources.

Auxiliary data, like unstructured temporal data, may not directly relate to the main task but provides relevant logic that enhances understanding. The TRAM dataset (Figure 10 in Appendix A) serves as such a source, improving temporal reasoning through exposure to diverse temporal questions.

This approach shows that using auxiliary data can effectively enhance temporal reasoning and boost model robustness across various tasks.

5 Experimental Setup

Models: In this paper, we experimented with several state-of-the-art large language models (LLMs), including GPT-3.5-Turbo, GPT-4O, Mistral-2-7B, LLaMA-2-7B-chat, and Gemini 1.5 Pro Flash. These models represent the forefront in both open-source and closed-model applications, showcasing advancements in natural language understanding and generation capabilities.

Prompts & Frameworks: Prompting models with detailed instructions enhances their understanding of tasks, leading to improved responses. These prompts may include demonstrations for the model's reference. We explore the following prompting techniques:

- Chain of Thought (CoT) (Wei et al., 2023): CoT guides models through step-by-step reasoning, promoting structured responses. We evaluated it in both zero-shot (Z.S) and few-shot (F.S) settings.

- Faithful Chain of Thought (F-CoT) (Lyu et al., 2023): F-CoT ensures consistency by maintaining fidelity to the initial prompt throughout response generation. It was tested in zero-shot and few-shot scenarios.

- Program of Thought (PoT) (Chen et al., 2023): PoT offers a predefined sequence of operations for structured, task-specific responses, evaluated in zero-shot and few-shot contexts.

- Plan-and-Solve (Wang et al., 2023): Plan-and-Solve Prompting first breaks a complex task into smaller, manageable subtasks before solving them sequentially, reducing missing-step errors in reasoning.

- BINDER (Cheng et al., 2023): BINDER combines LLMs with symbolic languages like SQL/Python to convert natural language queries into executable programs, improving interpretability and robustness.

- DATER (Ye et al., 2023): DATER breaks down complex questions and table evidence into simpler sub-questions using LLMs, enhancing reasoning and interpretability without extensive training data.

Auxiliary Data: We use several unstructured temporal reasoning datasets to evaluate language models' temporal reasoning abilities:

- DATE Understanding (Srivastava et al., 2023): This dataset tests a model's ability to reason about dates in various formats and contexts, including tasks like Date Format Conversion, Date Arithmetic, and Time Reasoning.

- Temporal Sequences (Srivastava et al., 2023): This dataset evaluates models' logical deduction skills, asking them to deduce the timing of events based on given sequences and durations.

- TRAM dataset (Wang and Zhao, 2024): This benchmark includes ten datasets focused on temporal aspects like order, arithmetic, frequency, and duration, providing a comprehensive evaluation of models' temporal reasoning. Examples are shown in Table 10.

6 Results and Analysis

In this section, we present the results of improving temporal reasoning in LLMs for tabular tasks. We evaluate two strategies: C.L.E.A.R (Comprehend, Locate, Examine, Analyze, Resolve), a method designed for tabular reasoning, and the integration of out-of-domain temporal data for fine-tuning.

Prompt	**No FT**		**TRAM**		**TempTabQA**	
	Mean	**Std Dev**	**100**	**1000**	**100**	**1000**
			GPT-3.5 turbo			
Our prompt	**77.42%**	0.78%	**77.92%**	76.12%	**80.89%**	**82.04%**
F.S. CoT	73.67%	0.83%	75.53%	**76.19%**	77.81%	79.47%
F.S. F-CoT	70.96%	0.97%	71.13%	72.72%	77.85%	76.57%
F.S. PoT	70.63%	1.03%	70.58%	69.75%	79.37%	77.57%
Z.S. CoT	65.53%	0.75%	68.64%	69.61%	73.73%	72.72%
Z.S. F-CoT	64.96%	1.70%	67.32%	65.91%	71.27%	71.69%
Z.S. PoT	65.93%	0.78%	67.39%	68.05%	72.17%	72.76%
Plan-and-Solve	73.64%	0.95%	74.21%	77.95%	79.54%	77.95%
BINDER	72.43%	**0.52%**	73.49%	73.73%	78.05%	78.64%
DATER	73.36%	0.64%	73.76%	75.53%	79.23%	77.54%
			LLAMA 7B			
Our prompt	59.21%	1.38%	61.65%	59.85%	63.69%	63.17%
F.S. CoT	62.17%	0.72%	64.87%	63.66%	67.22%	**68.64%**
F.S. F-CoT	62.30%	1.67%	61.54%	64.24%	66.77%	67.32%
F.S. PoT	62.79%	1.46%	64.04%	66.08%	**67.43%**	65.94%
Z.S. CoT	60.71%	0.74%	62.48%	61.79%	63.07%	65.21%
Z.S. F-CoT	**63.47%**	0.63%	**66.98%**	**69.40%**	63.38%	64.31%
Z.S. PoT	62.71%	0.85%	63.69%	64.76%	65.07%	65.42%
Plan-and-Solve	61.18%	0.96%	64.80%	63.97%	64.66%	63.97%
BINDER	60.77%	**0.61%**	61.02%	64.14%	62.96%	64.11%
DATER	61.11%	0.67%	62.48%	63.62%	63.34%	65.59%
			MISTRAL-2 7B			
Our prompt	**70.15%**	1.05%	**73.52%**	**72.41%**	74.80%	73.87%
F.S. CoT	68.92%	**0.39%**	70.61%	71.13%	73.31%	**75.63%**
F.S. F-CoT	67.55%	1.39%	71.06%	70.02%	**74.84%**	75.25%
F.S. PoT	66.75%	1.43%	70.51%	69.99%	74.35%	73.59%
Z.S. CoT	64.70%	0.39%	66.22%	70.37%	71.62%	71.48%
Z.S. F-CoT	64.78%	0.63%	71.13%	69.51%	67.08%	71.10%
Z.S. PoT	64.99%	0.78%	66.63%	66.98%	71.69%	71.65%
Plan-and-Solve	67.92%	0.56%	70.99%	73.38%	72.27%	73.38%
BINDER	67.78%	0.84%	71.30%	71.17%	74.28%	73.10%
DATER	68.50%	0.63%	70.99%	72.17%	74.18%	73.66%

Table 1: Model's performance on various prompts, mean accuracy, and standard deviation (Std Dev) across 5 trials. Results with fine-tuning on auxiliary data (TRAM) and the TempTabQA dataset using 100 and 1000 examples are also shown. The best results are in bold.

Models	Original Prompt					Updated Prompt				
	CoT					CoT				
	Z.S	F.S	BINDER	DATER	C.L.E.A.R	Z.S	F.S	BINDER	DATER	C.L.E.A.R
	Missing Relevant Rows (Lower (↓) is better)									
GPT 4O	69.71%	70.89%	62.96%	65.73%	**62.82%**	**6.89%**	8.34%	22.08%	10.63%	**7.30%**
GPT 3.5	60.06%	56.59%	50.74%	50.22%	**49.46%**	11.60%	13.05%	27.66%	13.36%	**11.39%**
LLAMA 7B	64.42%	37.69%	33.26%	33.44%	**24.92%**	26.10%	16.13%	36.28%	13.67%	**11.18%**
MISTRAL-2	69.92%	50.85%	44.93%	44.79%	**39.84%**	16.44%	13.08%	30.46%	**10.31%**	10.76%
GEMINI 1.5	76.25%	68.05%	62.51%	62.72%	**57.29%**	9.07%	8.45%	23.88%	9.66%	**7.34%**
	Information Absence Detection (Higher (↑) is better									
GPT 4O	19.45%	22.71%	22.08%	21.77%	**25.16%**	89.89%	88.85%	89.69%	89.93%	**90.65%**
GPT 3.5	27.83%	29.73%	27.66%	29.32%	**31.71%**	79.75%	79.47%	79.51%	80.06%	**82.07%**
LLAMA 7B	25.37%	34.96%	36.28%	37.14%	**40.05%**	61.72%	61.96%	62.44%	**63.17%**	62.34%
MISTRAL-2	21.05%	29.42%	30.46%	31.15%	**32.71%**	75.42%	74.49%	75.04%	**77.81%**	**77.09%**
GEMINI 1.5	15.82%	22.33%	23.88%	21.46%	**25.23%**	89.48%	87.61%	91.21%	90.17%	**91.45%**

Table 2: This table compares C.L.E.A.R, zero-shot CoT, few-shot CoT, BINDER, and DATER frameworks for tasks involving Missing Relevant Rows and Information Absence Detection in evidence-based reasoning, considering both original and changed prompts. Experimental evaluations are conducted using GPT-4O, GPT-3.5 Turbo, LLAMA-7B-Chat, MISTRAL-7B, and Gemini 1.5 Pro Flash models.

6.1 C.L.E.A.R Prompting

We evaluated C.L.E.A.R prompting against various reasoning techniques, including CoT, F-CoT, PoT, Plan-and-Solve, and symbolic methods like BINDER and DATER, in both zero-shot and few-shot settings to determine the most effective approach. As shown in Table 1, C.L.E.A.R achieves 77.42% accuracy on GPT-3.5 Turbo, outperforming all textual reasoning methods, including Few-Shot CoT (73.67%). State-of-the-art approaches like Plan-and-Solve prompting fall short, achieving a mean accuracy of 73.64%. Similarly, symbolic methods such as FS PoT (70.63%), FS F-CoT (70.96%), DATER (72.43%), and BINDER (73.36%) underperform, highlighting the advantages of textual reasoning for semi-structured tables. Overall, C.L.E.A.R boosts mean accuracy by 3.75% over the next-best technique.

A similar pattern emerges with Mistral-2-7B, where C.L.E.A.R achieves a mean accuracy of 70.15%, surpassing textual reasoning methods like FS CoT (68.92%) and Plan-and-Solve (67.92%). Likewise, symbolic approaches such as FS PoT (66.75%), FS F-CoT (67.55%), DATER (68.50%), and BINDER (67.78%) perform below C.L.E.A.R. However, the performance gap is smaller for Mistral-2-7B, with a 1.23% improvement over the next-best method. In contrast, LLaMA-2 follows a different trend, where Zero-Shot F-CoT achieves the highest mean accuracy at 63.47%, while C.L.E.A.R reaches 59.21%. This discrepancy can likely be attributed to the smaller model size, as previous studies suggest that smaller models struggle to follow complex instructions effectively and show pre-training biases favoring simpler prompts. Additionally, we evaluated our method on GPT-4o and Gemini-1.5-Flash, alongside the Plan-and-Solve approach. The results can be found in Section A.3.

6.2 Efficacy of C.L.E.A.R

The results in Section 6.1 show that our method outperforms other prompting techniques. But is C.L.E.A.R *trustworthy?* Does it truly enhance the model's *evidence-based reasoning capabilities?* (Gupta et al., 2021). We compare C.L.E.A.R with Zero-Shot and Few-Shot Chain of Thought (CoT) methods across tasks that reveal model deficiencies. Task descriptions are as follows:

1. **Original Table:** The model uses the original table to answer the question, testing its ability to use the provided data effectively.
2. **Without Table:** The model answers the question without table access, testing if it relies on memorization or independent deduction. Ideal performance should decrease here.
3. **Altered Entity Name:** The table and question are provided with an altered entity name, testing the model's reasoning without relying on memorized data.
4. **Missing Relevant Rows:** The model is given the table with key rows removed, testing if it can use external knowledge to answer.
5. **Information Absence Detection:** The model identifies missing information when given a table with key rows deleted.

For the last two tasks i.e. Missing Relevant Rows and Information Absence Detection, we perform evaluations in two settings:

- **Original Prompt:** The model is tested with the original prompt to see if it can detect missing information or use external knowledge.
- **Updated Prompt:** The model is explicitly instructed that the information may or may not be present, to see if explicit instructions improve performance.

We aim to assess whether C.L.E.A.R prompting enhances evidence-based reasoning beyond just accuracy, evaluating its effectiveness in addressing specific reasoning challenges and improving model reliability across varied contexts.

Models	Task	Prompts				
	(expected)	Z.S CoT	F.S CoT	BINDER	DATER	C.L.E.A.R
GPT-4O	**O.T** (↑)	80.77%	82.42%	82.49%	82.61%	**84.59%**
	W.T (↓)	75.91%	73.31%	71.79%	69.78%	**69.44%**
	A.N.E (↑)	74.94%	76.36%	78.61%	79.06%	**82.62%**
GPT-3.5	**O.T** (↑)	65.53%	73.67%	72.43%	73.36%	**77.42%**
	W.T (↓)	62.27%	62.10%	58.74%	58.43%	**58.22%**
	A.N.E (↑)	61.09%	67.95%	67.91%	67.74%	**72.10%**
Llama	**O.T** (↑)	60.71%	**62.17%**	60.77%	61.11%	59.21%
	W.T (↓)	54.69%	42.85%	41.36%	35.51%	**31.81%**
	A.N.E (↑)	**66.87%**	60.78%	60.61%	58.95%	58.15%
MISTRAL-2	**O.T** (↑)	64.70%	68.92%	67.78%	68.50%	**70.15%**
	W.T (↓)	56.87%	52.58%	53.20%	50.99%	**49.36%**
	A.N.E (↑)	**74.87%**	63.55%	67.08%	66.53%	64.83%
GEMINI 1.5	**O.T** (↑)	74.48%	79.93%	81.65%	79.31%	**82.37%**
	W.T (↓)	70.85%	69.19%	67.43%	**64.69%**	68.64%
	A.N.E (↑)	79.02%	71.24%	**80.13%**	77.05%	79.75%

Table 3: Comparison among Z.S CoT, F.S CoT, BINDER, DATER, and C.L.E.A.R frameworks across Original Table(O.T), Without Table(W.T), and Altered Named Entity(A.N.E) tasks in evidence-based reasoning. We evaluate GPT-4O, GPT-3.5 Turbo, LLAMA-7B-Chat, MISTRAL-7B, and Gemini 1.5 Pro Flash models in our experiments.

Results and Analysis. As shown in Table 3, C.L.E.A.R consistently outperforms other methods on most tasks. GPT 4O achieves the highest accuracy at 85.08%, a 2.39% improvement over few-shot CoT. Our method even outperform both BINDER and DATER in almost all the tasks.

(a) Without Table. When evaluated without table access, C.L.E.A.R shows a performance decline across all models using original labels, indicating a reduced reliance on memorization in favor of contextual reasoning. Specifically, GPT-4o and GPT-3.5 Turbo exhibit a 4% drop, while LLaMA experiences an 11.04% decrease compared to Z.S COT. This suggests that C.L.E.A.R encourages models to base their responses on the provided context rather than drawing from parametric knowledge. The significance of this is further explored in Section A.4.

(b) Altered Table Entity. C.L.E.A.R excels in this task, except with LLaMA-2 and Mistral-2. GPT 4O achieves the highest accuracy at 82.62%, a 6.26% improvement over few-shot CoT. LLaMA-2 7B and Mistral-2 7B underperform, likely due to their smaller size, which limits their ability to follow instructions. GPT 3.5 turbo and Gemini 1.5 pro flash see gains of 4.15% and 8.51%, respectively.

(c) Missing Relevant Row. As shown in Table 2, C.L.E.A.R consistently outperforms other methods with the original prompt, despite a larger performance drop compared to zero-shot and few-shot CoT. With the modified prompt, it remains the best, except for GPT-4, where zero-shot slightly surpasses it. Prompting causes a ~1-5% drop across all models as the focus shifts from memorization to context-driven responses. While symbolic frameworks like BINDER and DATER also decline, C.L.E.A.R still leads overall.

(d) Information Absence Detection. As shown in Table 2, our method surpasses zero-shot and few-shot CoT, BINDER, and DATER with the original prompt across all models. LLaMA achieves the highest accuracy (40.05%), followed by GPT-3.5 (31.71%). With the updated prompt, C.L.E.A.R performs better for GPT models and GEMINI, while DATER slightly outperforms for LLaMA-2 and Mistral.

We conducted an ablation study on GPT-3.5 Turbo to evaluate the individual contributions of each component within the C.L.E.A.R methodology. Table 4 illustrates the synergistic effects of these components, highlighting that the removal of any step decreases model accuracy. Notably, the 'Examine & Analyze' step is identified as the most crucial; its omission results in the largest accuracy drop of 2.72% compared to the complete C.L.E.A.R prompt. This finding underscores the 'Examine & Analyze' step as the most vital component of the C.L.E.A.R framework. Further analysis shows that eliminating steps in pairs exacerbates the accuracy decline, particularly when 'Examine & Analyze' is paired with 'Comprehend' and 'Locate', leading to decreases of 3.69% and 4.16%, respectively. Removing all three—'Comprehend', 'Locate', and 'Examine & Analyze'—results in the most substantial drop of 4.18% in accuracy.

6.3 Auxiliary Data Fine Tuning

C.L.E.A.R enhances a model's ability to process context, handle temporal questions, and improve evidence-based reasoning. However, intrinsic improvement requires fine-tuning. We recommend

Prompt	No FT	TRAM		TempTabQA	
		100	1000	100	1000
C.L.E.A.R	**77.99%**	**77.92%**	76.12%	**80.89%**	**82.04%**
remove Comprehend	76.55%	77.52%	76.45%	80.52%	81.79%
remove Locate	76.25%	75.99%	**77.04%**	80.07%	81.89%
remove Examine & Analyze	75.27%	74.85%	76.02%	79.19%	79.70%
remove Comprehend + Locate	76.80%	76.03%	76.76%	80.57%	81.49%
remove Comprehend + Examine & Analyze	74.30%	74.86%	75.13%	77.66%	78.93%
remove Locate + Examine & Analyze	73.83%	74.09%	75.29%	77.94%	79.81%
remove Comprehend + Locate + Examine & Analyze	73.81%	75.27%	74.29%	77.96%	79.06%

Table 4: Ablation study of different components of C.L.E.A.R without Fine-Tuning and with Fine-tuning on TRAM (Auxiliary) and TempTabQA train sets with model GPT-3.5-Turbo (Single Run).

fine-tuning with auxiliary data to boost reasoning across formats and overall capabilities.

Why TRAM dataset? We evaluated the impact of fine-tuning GPT-3.5 Turbo using the auxiliary datasets discussed in Section 5, along with the TempTabQA dataset. For this evaluation, we fine-tuned with 100 examples from each dataset.

Train Dataset	Exact Match
No fine-tuning	73.49%
DATE	74.28%
Temporal Sequences	74.42%
TRAM	75.53%
TempTabQA	77.81%

Table 5: This table showcases the performance of GPT-3.5 Turbo following fine-tuning on auxiliary datasets within a zero-shot chain-of-thought (CoT) setting."

Analysis. Table 5 shows that the TRAM dataset delivers the highest performance boost among auxiliary datasets, with a 2.04% improvement over the base model without fine-tuning. In contrast, DATE and Temporal Sequences datasets provide gains of less than 1.01%. TRAM's strength lies in its diversity, covering a wide range of temporal tasks like Ordering, Frequency, Duration, and Temporal NLI, helping models handle various temporal scenarios more effectively. See Table 11 in Appendix A for more details.

The significant gains from the TRAM dataset emphasize the value of diverse auxiliary data for fine-tuning. By spanning various temporal reasoning tasks, TRAM helps models better understand nuanced temporal relationships, boosting performance across challenges. This suggests that future fine-tuning should leverage similarly diverse datasets for maximum improvement.

Fine-tuning on TRAM. We fine-tuned models on subsets of the TRAM dataset and TempTabQA with 100 and 1000 evenly sampled examples, as shown in Table 1.

Analysis. With 100 examples, GPT-3.5-Turbo achieves 77.92% accuracy with C.L.E.A.R, outperforming textual reasoning methods like FS CoT (75.53%) and Plan-and-Solve (74.21%), as well as symbolic approaches such as FS PoT (70.58%), FS F-CoT (71.13%), Binder (73.49%), and Dater (73.76%). When trained on 1000 examples, C.L.E.A.R maintains strong performance at 76.12%, falling behind the best-performing method, FS CoT (76.19%), by just 0.07%.

Fine-tuning with auxiliary data significantly improves performance, bringing it closer to task-specific fine-tuning levels. This approach not only strengthens temporal reasoning but also enhances generalization across different tasks, making it more versatile than task-specific datasets. Moreover, it scales effectively for larger datasets beyond traditional fine-tuning constraints.[1] The observed accuracy gains suggest that auxiliary unstructured datasets like TRAM can help overcome data scarcity challenges while significantly enhancing model robustness and adaptability for semi-structured temporal tasks. Notably, fine-tuning on TRAM leads to substantial improvements in temporal reasoning, outperforming models trained solely on TempTabQA and demonstrating strong cross-format transferability from unstructured to semi-structured data.

7 Discussion and Related Work

Tabular Reasoning. Previous research has explored large language models' (LLMs) applications with semi-structured tabular data (Chen et al., 2020a; Gupta et al., 2020; Zhang and Balog, 2019), but temporal reasoning in this domain remains underexplored. While studies such as (Zhang et al., 2020b; Zhang and Balog, 2020; Pasupat and Liang, 2015; Krishnamurthy et al., 2017; Abbas et al., 2016; Sun et al., 2016; Chen et al., 2020b; Lin et al., 2020; Zayats et al., 2021; Oguz et al., 2020; Chen et al., 2021b; Iyyer et al., 2017) focus on table-to-text generation, various datasets and models like Table2vec (Zhang et al., 2019), TAPAS (Herzig

---

[1] Fine-tuned on 1000 examples due to computational limit.

et al., 2020), TaBERT (Yin et al., 2020), TabStruc (Zhang et al., 2020a), TABBIE (Iida et al., 2021), TabGCN (Pramanick and Bhattacharya, 2021), and RCI (Glass et al., 2021) aim to improve tabular data understanding. However, despite these advancements, temporal aspects of tables remain overlooked. Unlike studies that target predefined schema tables, our approach addresses tables without set hierarchies, characterized by complex and interrelated data, introducing a novel perspective.

Structural Temporal Reasoning. Traditional research has mainly addressed temporal reasoning in unstructured and knowledge-based systems, including temporal links in knowledge graphs (Saxena et al., 2021; Jia et al., 2018a), open-domain settings (Zhang and Choi, 2021), cloze-form question answering (Dhingra et al., 2022), and event-centric datasets (Ning et al., 2018; Wen et al., 2021; Chen et al., 2021a). These models often struggle with grounding outputs in specific contexts due to a reliance on parametric knowledge. Our experiments using Chain-of-Thought and Program-of-Thought prompting reveal these memorization issues, which C.L.E.A.R. overcomes by consistently grounding answers in the context and breaking down complex tasks into manageable steps. This makes C.L.E.A.R. particularly effective for semi-structured tables, surpassing the limitations of generic methods.

Symbolic Reasoning. Symbolic approaches that generate SQL queries (Cheng et al., 2023; Ye et al., 2023; Wang et al., 2024) have shown promise for structured data reasoning, but techniques like Chain-of-Table struggle with semi-structured, entity-centric tables. These tables introduce complexities that require understanding both row-level key-value pairs and implicit relationships across the table, necessitating advanced reasoning that combines domain knowledge and common sense, especially when dealing with temporal and tabular aspects. Our method surpasses other symbolic approaches, such as BINDER and DATER, by effectively managing these complexities. Moreover, while models like CRONKBQA (Saxena et al., 2021) and TEQUILA (Jia et al., 2018b) focus on temporal reasoning, we extend these capabilities to structured data, as demonstrated by our results on TempTabQA (Gupta et al., 2023) and TRAM (Wang and Zhao, 2024).

Other Finetuning Approaches. Fine-tuning techniques to improve performance on semi-structured data (Yu et al., 2018; Eisenschlos et al., 2020; Neeraja et al., 2021) often risk overfitting, where models learn task-specific patterns instead of generalizable logic. To address this, our approach leverages the TRAM dataset, fine-tuning on auxiliary datasets to emphasize core logical principles over task-specific structures, demonstrating adaptability across different data formats. In related areas like financial document QA, the EEDP model (Srivastava et al., 2024) excels in task decomposition, but our method goes further in evidence-based reasoning, enabling more effective temporal reasoning with tabular data.

In conclusion, although recent studies have explored temporal reasoning in unstructured data (Tan et al., 2023, 2024), our research tackles the additional complexities of semi-structured data, particularly the intricate relationships between rows and columns. We demonstrate that temporal reasoning can effectively transfer across different formats, enhancing our understanding of temporal reasoning in semi-structured contexts. Unlike synthetic data methodologies (Tan et al., 2024), our approach applies logical reasoning across diverse data types, offering a more comprehensive perspective on temporal reasoning for LLMs and paving the way for future research in this area.

8 Conclusion and Future Work

Our results show that C.L.E.A.R (Comprehend, Locate, Examine, Analyze, Resolve) enhance model performance, particularly in tabular data and temporal reasoning tasks, by grounding LLMs in evidence rather than pre-trained knowledge. Fine-tuning with auxiliary unstructured simple temporal data further enhances temporal understanding, boosting reasoning capabilities.

For the future, we propose: **(a.) Synthetic Data Generation:** Creating synthetic temporal data from tables to boost model performance through diverse exposure. **(b.) Neuro-symbolic Learning:** Combining neural networks with symbolic reasoning for better temporal understanding. **(c.) Expanding C.L.E.A.R Applications:** Applying C.L.E.A.R across different tasks and domains to demonstrate its versatility. **(d.) Integration with Existing Models:** Seamlessly incorporating C.L.E.A.R and auxiliary data into current models to maximize benefits without changing architecture.

Limitations

The experiments in this paper have been conducted exclusively on the English language. This study can be extended to a multilingual setting to evaluate the approach's effectiveness across different languages. Additionally, the temporal datasets used in our study are limited to simple, entity-centric tables. Since structured data can exist in more complex forms, such as hierarchical tables, further research is necessary to assess the impact of our methods on these more complex structures.

Moreover, our computational limitations restricted us to fine-tuning models on only 1000 samples of auxiliary data. To fully understand the potential improvements from fine-tuning on auxiliary data, it is essential to explore the effects of fine-tuning on larger datasets. Future work should focus on overcoming these limitations to provide a comprehensive evaluation of our approach.

Ethics Statement

We confirm that our work adheres to the highest ethical standards in research and publication. We will publicly release our code and enhanced evaluation set to enable the research community to validate and build upon our findings. We are committed to the responsible and fair use of computational linguistics methodologies. The claims in our paper accurately reflect the experimental results. While using black-box large language models introduces some stochasticity, we mitigate this by maintaining a fixed temperature. We utilize an AI assistive tools for writing while ensuring absence of bias. We provide comprehensive details on annotations, dataset splits, models used, and prompting methods tried, ensuring the reproducibility of our work.

Acknowledgements

We thank Jennifer Sheffield at UPenn for her support throughout this project. This research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-20-1-0080. This work was partially funded by ONR Contract N00014-23-1-2364. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Faheem Abbas, M. K. Malik, M. Rashid, and Rizwan Zafar. 2016. Wikiqa — a question answering system on wikipedia using freebase, dbpedia and infobox. *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, pages 185–193.

Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021a. Event-centric natural language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks.

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

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages

281–296, Online. Association for Computational Linguistics.

Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avi Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1212–1224, Online. Association for Computational Linguistics.

Vivek Gupta, Riyaz A. Bhat, Atreya Ghosal, Manish Srivastava, Maneesh Singh, and Vivek Srikumar. 2021. Is my model using the right evidence? systematic probes for examining evidence-based tabular reasoning. *CoRR*, abs/2108.00578.

Vivek Gupta, Pranshu Kandoi, Mahek Vora, Shuo Zhang, Yujie He, Ridho Reinanda, and Vivek Srikumar. 2023. TempTabQA: Temporal question answering for semi-structured tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2431–2453, Singapore. Association for Computational Linguistics.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. Tabbie: Pretrained representations of tabular data.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1057–1062, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018b. TEQUILA: Temporal Question Answering over Knowledge Bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 1807–1810, New York, NY, USA. ACM.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526, Copenhagen, Denmark. Association for Computational Linguistics.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, Online. Association for Computational Linguistics.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating external knowledge to enhance tabular reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2799–2809, Online. Association for Computational Linguistics.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77, Brussels, Belgium. Association for Computational Linguistics.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unified open-domain question answering with structured and unstructured knowledge. *arXiv preprint arXiv:2012.14610*.

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

Aniket Pramanick and Indrajit Bhattacharya. 2021. Joint learning of representations for web-tables, entities and types using graph convolutional network.

In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1197–1206, Online. Association for Computational Linguistics.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676, Online. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, and Abubakar Abid. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Pragya Srivastava, Manuj Malik, Vivek Gupta, Tanuja Ganu, and Dan Roth. 2024. Evaluating llms' mathematical reasoning in financial document question answering.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Huan Sun, Hao Ma, Xiaodong He, Scott Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table cell search for question answering. In *Proceedings of the companion publication of the 25th international conference on World Wide Web*. ACM - Association for Computing Machinery.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2024. Towards robust temporal reasoning of large language models via a multi-hop qa dataset and pseudo-instruction tuning.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Yuqing Wang and Yun Zhao. 2024. Tram: Benchmarking temporal reasoning for large language models.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Haoyang Wen, Yanru Qu, Heng Ji, Qiang Ning, Jiawei Han, Avi Sil, Hanghang Tong, and Dan Roth. 2021. Event time extraction and propagation via graph attention networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 62–73, Online. Association for Computational Linguistics.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning.

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Vicky Zayats, Kristina Toutanova, and Mari Ostendorf. 2021. Representations for question answering from documents with tables and text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2895–2906, Online. Association for Computational Linguistics.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020a. Table fact verification with structure-aware transformer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1629, Online. Association for Computational Linguistics.

Li Zhang, Shuo Zhang, and Krisztian Balog. 2019. Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19. ACM.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shuo Zhang and Krisztian Balog. 2019. Auto-completion for data cells in relational tables. In *Proceedings of the 28th ACM International Conference* *on Information and Knowledge Management*, CIKM '19, page 761–770, New York, NY, USA. Association for Computing Machinery.

Shuo Zhang and Krisztian Balog. 2020. Web table extraction, retrieval, and augmentation: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(2):13:1–13:35.

Shuo Zhang, Zhuyun Dai, Krisztian Balog, and Jamie Callan. 2020b. Summarizing and exploring tabular data in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 1537–1540, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Additional Results on PALM.

We conducted experiments using the PaLM-2 model, where C.L.E.A.R outperformed all other prompting techniques without fine-tuning (Table 6), showing approximately a 3% improvement over FS COT.

In both the "Original Table" and "Without Table" tasks, C.L.E.A.R demonstrated the best performance (Table 7), highlighting its ability to mitigate memorization issues even in the PaLM-2 model. However, for the "Altered Named Entity" task, Z.S COT performed the best.

For the "Missing Relevant Info" and "Information Absence Detection" tasks, C.L.E.A.R achieved the highest accuracy with both the original and updated prompts (Table 8). We were unable to compare C.L.E.A.R's performance with BINDER and DATER on PaLM-2, as the model was deprecated at the time of revision.

Prompt	**No FT**	**TRAM**		**TempTabQA**	
		100	**1000**	**100**	**1000**
			PALM 2		
C.L.E.A.R	**80.06%**	**81.97%**	**83.28%**	**85.01%**	**85.01%**
F.S. CoT	77.40%	81.14%	82.24%	80.17%	81.59%
F.S. F-CoT	75.63%	75.35%	78.82%	82.45%	82.21%
F.S. PoT	76.01%	75.98%	76.25%	84.56%	80.27%
Z.S. CoT	70.13%	73.97%	74.42%	78.89%	76.74%
Z.S. F-CoT	69.75%	74.04%	74.90%	75.22%	75.94%
Z.S. PoT	71.24%	74.32%	76.29%	79.09%	79.13%

Table 6: PaLM-2 model's performance on various prompts without fine-tuning, with fine-tuning on auxiliary data (TRAM), and with fine-tuning on the TempTabQA dataset using 100 and 1000 examples. The best result is highlighted in bold.

A.2 Additonal Details

Table 6 presents sample questions from the TRAM dataset, showcasing the complexity and diversity of the queries. Figure 2 illustrates an example of a prompt used in our method, emphasizing the input structure. Table 7 provides an overview of the tasks in the new evaluation set, categorizing different objectives. Figure 3 outlines the step-by-step process of the C.L.E.A.R. instruction method. Figures 4 and 5 present Example 1 from the TempTabQA head set, with Figure 4 showing the input and Figure 5 displaying the response generated using C.L.E.A.R. prompting with GPT-3.5 turbo. Similarly, Figures 6 and 7 depict Example 2 from the TempTabQA tail set, where Figure 6 illustrates the input and Figure 7 provides the corresponding response.

Models	**Task (expected)**	**Prompts**		
		Z.S CoT	**F.S CoT**	**C.L.E.A.R**
Palm 2	**Original Table (↑)**	69.75%	77.29%	**79.58%**
	Without Table (↓)	66.63%	63.24%	**60.26%**
	Altered Name Entity (↑)	**82.10%**	71.34%	74.97%

Table 7: Performance of the PaLM-2 model across different tasks using various prompting techniques.

Models	**Original Prompt**			**Updated Prompt**		
	CoT			**CoT**		
	Z.S	**F.S**	**C.L.E.A.R**	**Z.S**	**F.S**	**C.L.E.A.R**
	Missing Relevant Rows (Lower (↓) is better)					
PALM 2	78.40%	61.68%	**55.90%**	10.90%	8.27%	**7.58%**
	Information Absence Detection (Higher (↑) is better					
PALM 2	19.21%	23.68%	**29.11%**	86.43%	86.78%	**88.44%**

Table 8: Performance of the PaLM-2 model in tasks involving Missing Relevant Rows and Information Absence Detection using original and updated prompts.

A.3 Comparison with Plan-and-Solve

We compared the performance of C.L.E.A.R with the Plan-and-Solve method on GPT-4o and Gemini-1.5-Flash models.

Prompt	**No FT**	
	GPT-4o	**Gemini-1.5-flash**
Plan-and-Solve	82.32%	82.03%
C.L.E.A.R	84.59%	82.37%

Table 9: Performance comparison of Plan-and-Solve and C.L.E.A.R under No FT (Fine-Tuning) on models GPT4o and Gemini-1.5-flash.

As shown in Table 9, C.L.E.A.R outperforms Plan-and-Solve on both models. For GPT-4o, C.L.E.A.R achieved an accuracy of 84.59%, compared to Plan-and-Solve's 82.32%. Similarly, for Gemini-1.5-Flash, C.L.E.A.R demonstrated superior performance, achieving 82.37% accuracy versus Plan-and-Solve's 82.03%.

A.4 Prioritizing Grounded Reasoning Over Memorization

The C.L.E.A.R prompting method ensures that LLMs engage in evidence-based reasoning rather than relying solely on memorized factual knowledge. While factual recall is valuable, it does not guarantee accuracy, especially in dynamic domains where information evolves. Since LLMs cannot autonomously update their internal knowledge. The parametric knowledge of LLMs can often be outdated or inaccurate. C.L.E.A.R emphasizes reasoning grounded in provided evidence, reducing reliance on outdated or unverifiable memorization.

A key challenge with memorization is that it may lead to incorrect conclusions when faced with counterfactual or updated information. For example, imagine a counterfactual scenario where the evidence table has been updated to reflect *"McBean's hypothetical career with the Yankees, showing a different length of service than his time with the Pittsburgh Pirates"*. In this case, the question would be: *"For how many years did Al McBean play for the Pittsburgh Pirates?"* The model must now reason based on the updated table, grounding its answer in the new, counterfactual data. This demonstrates that the model cannot simply recall memorized facts about McBean's actual career with the Pirates; instead, it must rely on the provided evidence to generate the correct answer.

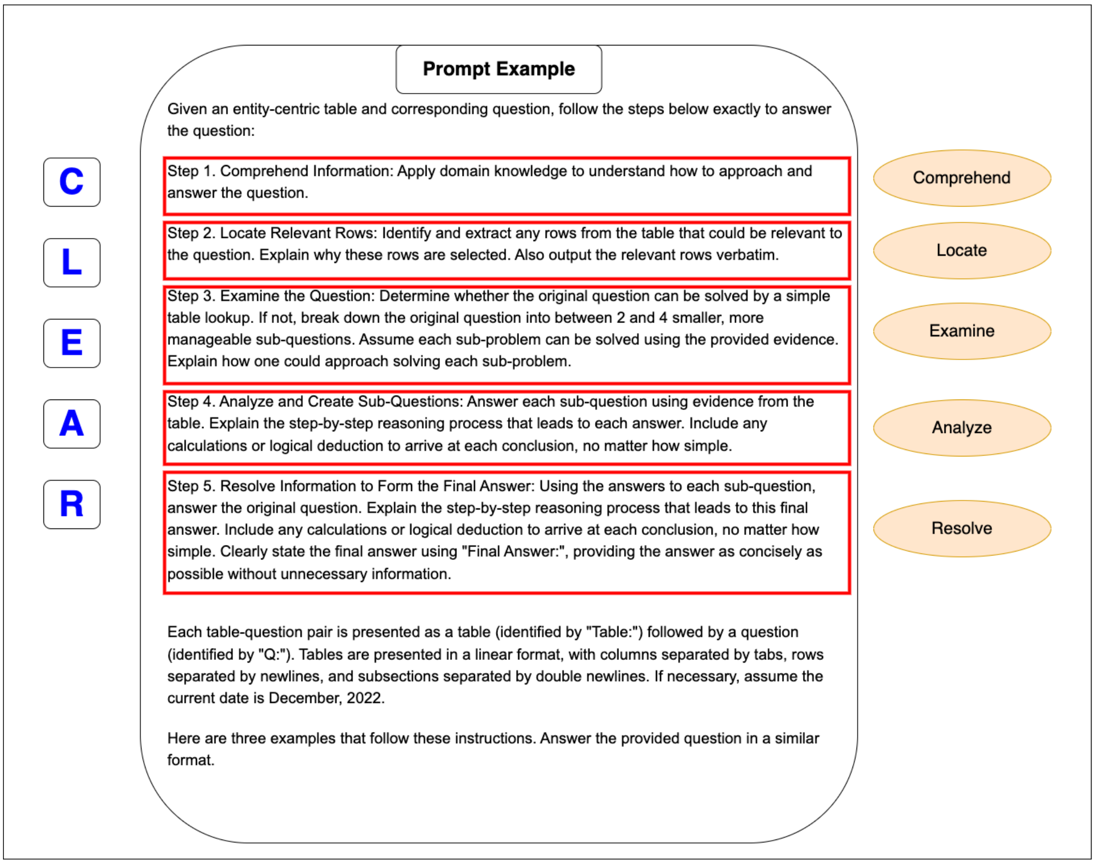
Moreover, when working with entity-centric tables, different entities can share the same name, making context vital for disambiguation. For instance, ambiguous entity references—such as *"How many championships has Adam Jones won?"* require contextual disambiguation, here an LLM might get confuse *"Adam Jones"* with multiple individuals, such as an *American football player* or *a famous baseball player*. C.L.E.A.R enforces by guiding the model to prioritize information within the provided context i.e. given evidence, rather than defaulting to pre-trained knowledge.

Beyond improving factual accuracy, C.L.E.A.R enhances the model's ability to recognize when a question is unanswerable due to missing evidence. Instead of inferring an unsupported answer, the model is encouraged to respond with *"not answerable due to missing evidence,"* reducing the risk of hallucinations. This is validated through the Information Absence Detection test, which shows that C.L.E.A.R effectively curbs incorrect completions when crucial data is missing.

Ultimately, C.L.E.A.R does not diminish factual recall but reinforces structured reasoning in the presence of explicit evidence. By ensuring that responses are contextually aligned, accurate, and verifiable, this approach enhances LLM robustness across various reasoning tasks. Grounded reasoning is particularly crucial in real-world applications where factual correctness depends not just on stored knowledge but on the ability to interpret and synthesize relevant information dynamically. This approach aligns with Retrieval-Augmented Generation (RAG) techniques, which enhance factual accuracy by incorporating the given context.

Ordering (Facts)	**Q**: Arrange the following events in chronological order: (1) Brusilov Offensive by Russia. (2) Kamehameha I of the Island of Hawaii defeats the Oahuans at the Battle of Nu'uanu. (3) The Kuomintang, the Chinese nationalist party, is founded. (4) Emperor Claudius dies and is succeeded by his grand nephew Nero. (5) St. Norbert and 29 companions make their solemn vows marking the beginning of the Premonstratensian Order. **A**. (1), (2), (4), (5), (3) ✗ **B**. (4), (5), (2), (3), (1) ✓ **C**. (3), (1), (2), (4), (5) ✗
Frequency (Commonsense)	**Q**: It is also a love story, between Ace and Tobio, a trans woman. How often do they break up? **A**. Once ✓ **B**. Always ✗ **C**. Once per week ✗
Duration (Analogy Inference)	**Q**: While Yoga Session gave attendees time to plant an entire garden, Jazz Concert was enough to water a few plants, and Board Game Night was merely smelling a flower. Which event was the longest? **A**. Jazz Concert ✗ **B**. Board Game Night ✗ **C**. Yoga Session ✓
Typical Time (Comparison)	**Q**: Which event typically happens earlier: morning yoga or farmer starting their day? **A**. Morning yoga ✗ **B**. Farmer starting their day ✓ **C**.Around the same time✗
Ambiguity Resolution (Long-term Shift)	**Q**: The dynasty which fell in 1830 had risen to power roughly 90 years earlier. When was its establishment? **A**. 1742 ✗ **B**. 1745 ✗ **C** 1740 ✓
Arithmetic (Week Identification)	**Q**: In which week of year 2007 does the date 10-12-2007 occur? **A**.Week41 ✓ **B**.Week28 ✗ **C**.Week5 ✗ **D**.Week10 ✗
Temporal Relation	**Q**: It added that the Ministry of Economic Affairs and Finance was assigned to draw up practical procedure for the ceding, while the Ministry of Welfare and Social Security would be responsible for identifying the beneficiaries in two months. What is the relationship between the event 'added' and the event 'ceding'? **A**. IS_INCLUDED ✗ **B**. SIMULTANEOUS ✗ **C**. AFTER ✓
Temporal NLI	**Q**: Premise: Two guys playing football on a campus green. Hypothesis: They are practicing before the big game tomorrow **A**. Entailment ✗ **B**. Neutral ✓ **C**. Contradiction✗
Temporal Causality (Effect)	**Q**: The seasons changed from summer to autumn. What's the more plausible RESULT? **A**. People evacuated their homes. ✗ **B**. Leaves fell from the trees. ✓
Temporal Storytelling	**Q**: There is a huge clock in my living room. I turned the clock back one hour for daylight savings. My wife also turned the clock back one hour for daylight savings. Our 2 kids each turned the clock back one hour for daylight savings. Which of the two endings is the most plausible correct ending to the story? **A**. Then we wondered why it got so dark so early. ✓ **B**. The kids were not happy ✗

Table 10: Examples Questions in TRAM Dataset.



Prompt Example

Given an entity-centric table and corresponding question, follow the steps below exactly to answer the question:

Step 1. Comprehend Information: Apply domain knowledge to understand how to approach and answer the question.

Step 2. Locate Relevant Rows: Identify and extract any rows from the table that could be relevant to the question. Explain why these rows are selected. Also output the relevant rows verbatim.

Step 3. Examine the Question: Determine whether the original question can be solved by a simple table lookup. If not, break down the original question into between 2 and 4 smaller, more manageable sub-questions. Assume each sub-problem can be solved using the provided evidence. Explain how one could approach solving each sub-problem.

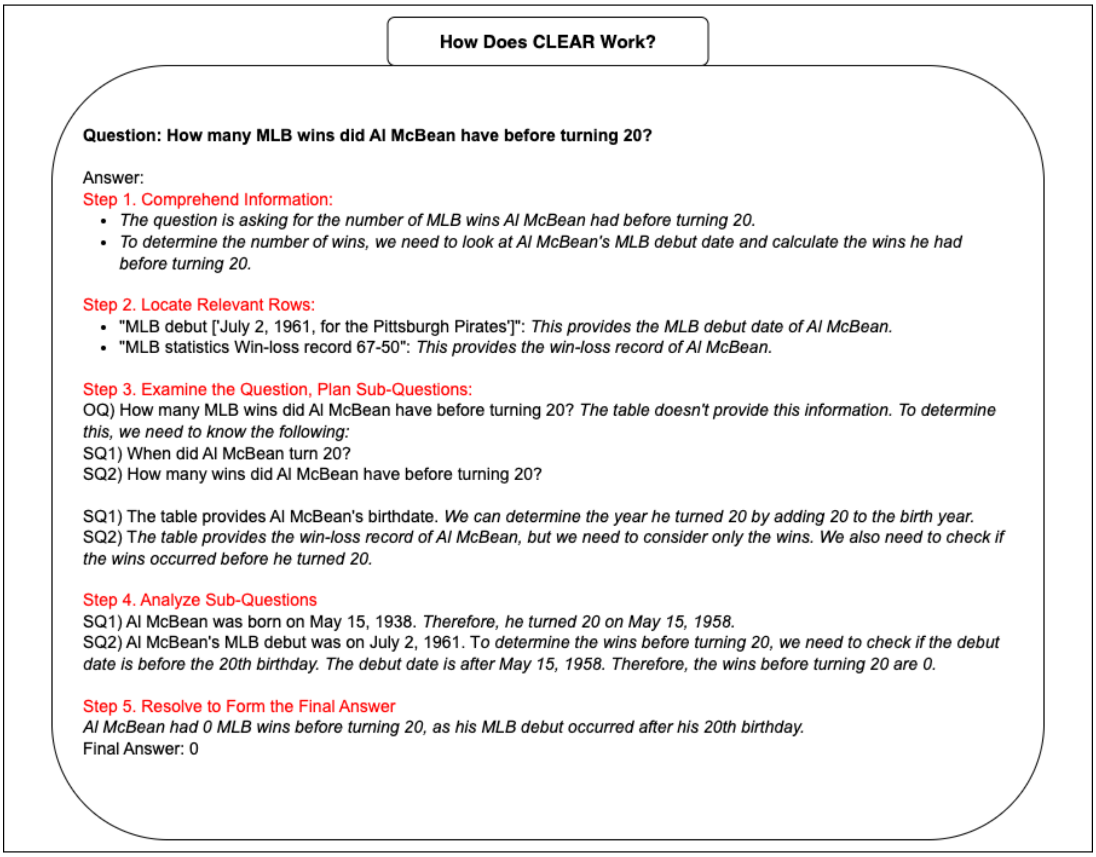
Step 4. Analyze and Create Sub-Questions: Answer each sub-question using evidence from the table. Explain the step-by-step reasoning process that leads to each answer. Include any calculations or logical deduction to arrive at each conclusion, no matter how simple.

Step 5. Resolve Information to Form the Final Answer: Using the answers to each sub-question, answer the original question. Explain the step-by-step reasoning process that leads to this final answer. Include any calculations or logical deduction to arrive at each conclusion, no matter how simple. Clearly state the final answer using "Final Answer:", providing the answer as concisely as possible without unnecessary information.

Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.

Here are three examples that follow these instructions. Answer the provided question in a similar format.

Figure 2: Prompt Example



How Does CLEAR Work?

Question: How many MLB wins did Al McBean have before turning 20?

Answer:
Step 1. Comprehend Information:

- *The question is asking for the number of MLB wins Al McBean had before turning 20.*
- *To determine the number of wins, we need to look at Al McBean's MLB debut date and calculate the wins he had before turning 20.*

Step 2. Locate Relevant Rows:

- "MLB debut ['July 2, 1961, for the Pittsburgh Pirates']": *This provides the MLB debut date of Al McBean.*
- "MLB statistics Win-loss record 67-50": *This provides the win-loss record of Al McBean.*

Step 3. Examine the Question, Plan Sub-Questions:
OQ) How many MLB wins did Al McBean have before turning 20? *The table doesn't provide this information. To determine this, we need to know the following:*
SQ1) When did Al McBean turn 20?
SQ2) How many wins did Al McBean have before turning 20?

SQ1) The table provides Al McBean's birthdate. *We can determine the year he turned 20 by adding 20 to the birth year.*
SQ2) T*he table provides the win-loss record of Al McBean, but we need to consider only the wins. We also need to check if the wins occurred before he turned 20.*

Step 4. Analyze Sub-Questions
SQ1) Al McBean was born on May 15, 1938. *Therefore, he turned 20 on May 15, 1958.*
SQ2) Al McBean's MLB debut was on July 2, 1961. T*o determine the wins before turning 20, we need to check if the debut date is before the 20th birthday. The debut date is after May 15, 1958. Therefore, the wins before turning 20 are 0.*

Step 5. Resolve to Form the Final Answer
Al McBean had 0 MLB wins before turning 20, as his MLB debut occurred after his 20th birthday.
Final Answer: 0

Figure 3: The figure illustrates the step-by-step process of C.L.E.A.R instruction. The reference table is provided in Figure 1

Task	Data Size	# Problem Types	Metrics	Answer Type	Text Sources
Foundational Temporal Understanding Tasks					
Ordering	29,462	Commonsense, Facts	Acc.	3-Way MC	MCTACO[1], Wikipedia, Misc.
Frequency	4,658	Commonsense, Reading Comprehension, Application, Computation, Comparison, Facts	Acc.	3-Way MC	MCTACO[1], SQuAD[2], Misc.
Duration	7,232	Commonsense, Reading Comprehension, Analogy Inference, Computation, Direct Comparison, Multi-step Comparison, Facts	Acc.	3-Way MC	Same
Typical Time	13,018	Commonsense, Comparison, Facts, Reading Comprehension	Acc.	3-Way MC	Same
Temporal Interpretation and Computation Tasks					
Amb. Res.	3,649	Interpretation, Calendar shift, Long-term shift, Mid-term shift, Short-term shift	Acc.	3-Way MC	Misc.
Arithmetic	15,629	Application, Date Computation, 12-hour Adjustment, 24-hour Adjustment, Month Shift, Week Identification, Year Shift, Time Computation, Time Zone Conversion	Acc.	4-Way MC	Same
Advanced Temporal and Conceptual Understanding Tasks					
Relation	102,462	-	Acc./F1	3-Way MC	TempEval-3[3]
Temporal NLI	282,144	-	Acc./F1	3-Way MC	MNLI[4], SNLI[5]
Causality	1,200	Cause, Effect	Acc.	2-Way MC	COPA[6], Misc.
Storytelling	67,214	-	Acc.	2-Way MC	ROC[7], SCT[8]

Table 11: Overview of tasks in TRAM.

Input :

Given an entity-centric table and corresponding question, follow the steps below exactly to answer the question:
Step 1. Comprehend Information: Apply domain knowledge to understand how to approach and answer the question.
Step 2. Locate Relevant Rows: Identify and extract any rows from the table that could be relevant to the question. Explain why these rows are selected. Also output the relevant rows verbatim.
Step 3. Examine the Question: Determine whether the original question can be solved by a simple table lookup. If not, break down the original question into between 2 and 4 smaller, more manageable sub-questions. Assume each sub-problem can be solved using the provided evidence. Explain how one could approach solving each sub-problem.
Step 4. Analyze Sub-Questions: Answer each sub-question using evidence from the table. Explain the step-by-step reasoning process that leads to each answer. Include any calculations or logical deduction to arrive at each conclusion, no matter how simple.
Step 5. Resolve to Form the Final Answer: Using the answers to each sub-question, answer the original question. Explain the step-by-step reasoning process that leads to this final answer. Include any calculations or logical deduction to arrive at each conclusion, no matter how simple. Clearly state the final answer using "Final Answer:", providing the answer as concisely as possible without unnecessary information.

Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.

========================

Table:
Title Bette Davis
Died October 6, 1989 | (1989-10-06) | (aged 81) |
Neuilly-sur-Seine, France
Resting place Forest Lawn Memorial Park
Occupation Actress
Years active 1929-1989
Political party Democratic
Spouse(s) Harmon Oscar Nelson | (| married | 1932; | divorced | 1938) | Arthur Farnsworth | (| married | 1940; died 1943) | William Grant Sherry | (| married | 1945; | divorced | 1950) | Gary Merrill | (| married | 1950; | divorced | 1960)
Children 3, including B. D. Hyman
Bette Davis ['BETTEDavis (cropped).jpg | Publicity photo, 1935']

Question: How many years after Davis's career began did she take the photo?

Figure 4: Example 1 from TempTabQA head set with C.L.E.A.R prompting on GPT 3.5 turbo : Input

Response :

Step 1. Comprehend Information Domain Knowledge:
- The question asks "how many years", so the answer should be a length of time in years
- To determine the time between two years, take the absolute value of the difference between the two years.

Step 2. Locate Relevant Rows
- "Title Bette Davis": This indicates that the information in the table is about Bette Davis. - "Years active 1929-1989": This provides the start and end year of Bette Davis's career. - "Bette Davis ['BETTEDavis (cropped).jpg | Publicity photo, 1935']": This provides the year the photo was taken.

Step 3. Examine the Question OQ) How many years after Davis's career began did she take the photo? The table doesn't provide this information. To determine the time between these two events, we need to know the following:

- SQ1) When did Davis's career begin?
- SQ2) When did Davis take the photo? : Davis took the photo in 1935

SQ1) When did Davis's career begin?
The table gives the range years spanning Davis's career. We can determine the begin year of her career by looking at the start year of this range.
SQ2) When did Davis take the photo?
The table provides the year the photo was taken.

Step 4. Answer Sub-Questions SQ1) Davis's career lasted from 1929 to 1989. Therefore, her career began in 1929. SQ2) The photo was taken in 1935.

Step 5. Synthesize to Form the Final Answer OQ) How many years after Davis's career began did she take the photo?
Davis's career began in 1929
(SQ1), and she took the photo in 1935
(SQ2). To get the number of years between these two events, take the absolute value of the difference between the two years.
The absolute value of (1929-1935) is 6.
Therefore, the photo was taken 6 years after Davis's career began.

Final Answer: 6

Figure 5: Example 1 from TempTabQA head set with C.L.E.A.R prompting on GPT 3.5 turbo : Response

Input :

Given an entity-centric table and corresponding question, follow the steps below exactly to answer the question:
Step 1. Comprehend Information: Apply domain knowledge to understand how to approach and answer the question.
Step 2. Locate Relevant Rows: Identify and extract any rows from the table that could be relevant to the question. Explain why these rows are selected. Also output the relevant rows verbatim.
Step 3. Examine the Question: Determine whether the original question can be solved by a simple table lookup. If not, break down the original question into between 2 and 4 smaller, more manageable sub-questions. Assume each sub-problem can be solved using the provided evidence. Explain how one could approach solving each sub-problem.
Step 4. Analyze Sub-Questions: Answer each sub-question using evidence from the table. Explain the step-by-step reasoning process that leads to each answer. Include any calculations or logical deduction to arrive at each conclusion, no matter how simple.
Step 5. Resolve to Form the Final Answer: Using the answers to each sub-question, answer the original question. Explain the step-by-step reasoning process that leads to this final answer. Include any calculations or logical deduction to arrive at each conclusion, no matter how simple. Clearly state the final answer using "Final Answer:", providing the answer as concisely as possible without unnecessary information.

Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.

=======================
Table:
Title Dwight Bernard
Dwight Bernard ['Dwight Bernard.jpg']
Pitcher
Born | (| 1952-05-31 |) | May 31, 1952 | (age 70) | Mount Vernon, Illinois
Batted | Right | Threw | Right
MLB debut
MLB debut ['June 29, 1978, for the New York Mets']
Last MLB appearance
Last MLB appearance ['October 2, 1982, for the Milwaukee Brewers']
MLB statistics
Win-loss record 4-8
Earned run average 4.14
Strikeouts 92
Teams
Teams ['New York Mets (1978-1979) | Milwaukee Brewers (1981-1982)']

Question: In what year did Dwight Bernard start playing with the last team he was associated with?

Figure 6: Example 2 from TempTabQA tail set with C.L.E.A.R prompting on GPT 3.5 turbo : Input

Response :

Step 1. Comprehend Information
Domain Knowledge:
- The question asks "In what year", so the answer should be a year.
- To determine the last team someone was associated with, identify the one that occurred most recently.
- Given an event spanning a range of years, the start is the first number in that range.

Step 2. Locate Relevant Rows
- "Title   Dwight Bernard": This indicates that the information in the table is about Dwight Bernard.
- "Teams   ['New York Mets (1978-1979) | Milwaukee Brewers (1981-1982)']": This provides a list of all teams Dwight Bernard played with and the years he played with each.

Step 3. Examine the Question
OQ) In what year did Dwight Bernard start playing with the last team he was associated with? The table doesn't provide this information. To determine this, we need to know the following:

- SQ1) What was the last team Dwight Bernard was associated with?
- SQ2) When did Dwight Bernard start playing for this team?

SQ1) What was the last team Dwight Bernard was associated with?
The table doesn't provide this information. However, the table provides a list of all teams Bernard was associated with and the years he was associated with each team. We can determine the last team Dwight Bernard was associated with by comparing the last year he was associated with each of his teams.
SQ2) When did Dwight Bernard start playing for this team?
The table provides the range of years Bernard was associated with each of his teams. We can determine the year he started playing for that team by looking at the start year in the range.

Step 4. Analyze Sub-Questions
SQ1) The only teams Dwight Bernard has played with are the New York Mets and Milwaukee Brewers. Bernard played for the Mets from 1978 to 1979 and the Brewers from 1981 to 1982. He was associated with the Brewers in 1982, which is more recent than the Mets in 1979; therefore, the last team he was associated with is the Brewers.
SQ2) He played with the Brewers from 1981 to 1982, so he started playing for the Brewers in 1981. **Step 5.** Resolve to Form the Final Answer
OQ) In what year did Dwight Bernard start playing with the last team he was associated with?
The last team Dwight Bernard was associated with was the Brewers (SQ1), which he started playing for in 1981 (SQ2). Therefore, Dwight Bernard started playing with the last team he was associated with in 1981.

Final Answer: 1981

Figure 7: Example 2 from TempTabQA tail set with C.L.E.A.R prompting on GPT 3.5 turbo : Response

Incorrect Human Annotation (head set)

Table:
Title   Chandrayaan-2
Mission type   Lunar orbiter, lander, rover
Operator   Indian Space Research Organisation (ISRO)
COSPAR ID   2019-042A | Edit this at Wikidata
SATCAT | number   44441
Website   www | .isro | .gov | .in | /chandrayaan2-home-0
Mission duration   Orbiter: 7.5 years (planned); | 2 years, 10 months (elapsed) | Vikram | lander: â‰¤ 14 days (planned); | 0 days (landing failure) | Pragyan | rover: â‰¤ 14 days (planned); | 0 days (landing failure)

Question: Rounding up, what was the difference in planned and elapsed time of the Orbiter Mission duration?

Original Annotated Answer: 4 years

Correct Answer: The planned time is 7 years and 6 months, and the elapsed time is 2 years and 10 months. The difference is 4 years and 8 months, which rounds up to 5 years.

Figure 8: Example 1: Error Analysis (Incorrect Human Annotation)

Requires External Knowledge (tail set)

Table:
Title Black Friday
Type Commercial
Significance Popular shopping day
Observances Shopping
Date Day after U.S. Thanksgiving
2021 date November 26 | (| 2021-11-26 |)
2022 date November 25 | (| 2022-11-25 |)
2023 date November 24 | (| 2023-11-24 |)
2024 date November 29 | (| 2024-11-29 |)
Frequency Annual
Related to Thanksgiving, Small Business Saturday, Cyber Monday, Giving Tuesday, Christmas, Buy Nothing Day

Question: What would the day of November 27th 2021 be referred to as?

Original Annotated Answer: Small Business Saturday.

Rewritten Question: What day of the week was November 27, 2021?

Reasoning for Rephrased Question: Unlike the original question (which requires nontrivial external knowledge that "Small Business Saturday" is the day after Black Friday), the revised question can be answered with common sense (i.e. Saturday is the day after Friday)

Correct Answer: Saturday

Figure 9: Example 2: Error Analysis (Requires External Knowledge)

Ambiguous Question (head set)

Table:
Title B-24 Skua
Role Carrier-based dive
bomber/fighter | Type of aircraft
Manufacturer Blackburn Aircraft
First flight 9 February 1937
Introduction November 1938
Retired 1941 (withdrawn from front line) | March 1945 (withdrawn from other duties)
Primary user Fleet Air Arm
Number built 192
Variants Blackburn Roc
B-24 Skua ['15 Blackburn Skua, Bristol Perseus (15650612347).jpg', 'Blackburn Skua in flight']

Question: How many years did B-24 Skua was in active service as bomber/fighter?

Original Annotated Answer: 3 years

Rewritten Question: How many years were B-24 Skuas on the front line?

Reasoning for Rephrased Question: *"In active service"* is ambiguous, so the end date could be either of the retirement dates (1941 or 1945). The rewritten question clarifies this ambiguity.

Correct Answer: 3 years

Figure 10: Example 3: Error Analysis (Ambiguous Question)