TAG–EQA: Text–And–Graph for Event Question Answering via Structured Prompting Strategies

Maithili Kadam **Francis Ferraro**
University of Maryland, Baltimore County
mkadam1@umbc.edu, ferraro@umbc.edu

Abstract

Large language models (LLMs) excel at general language tasks but often struggle with event-based questions—especially those requiring causal or temporal reasoning. We introduce **TAG-EQA** (Text-And-Graph for Event Question Answering), a prompting framework that injects causal event graphs into LLM inputs by converting structured relations into natural-language statements. TAG-EQA spans nine prompting configurations, combining three strategies (zero-shot, few-shot, chain-of-thought) with three input modalities (text-only, graph-only, text+graph), enabling a systematic analysis of when and how structured knowledge aids inference. On the TORQUESTRA benchmark, TAG-EQA improves accuracy by 5% on average over text-only baselines, with gains up to 12% in zero-shot settings and 18% when graph-augmented CoT prompting is effective. While performance varies by model and configuration, our findings show that causal graphs can enhance event reasoning in LLMs without fine-tuning, offering a flexible way to encode structure in prompt-based QA.[1]

1 Introduction

Consider the text in Figure 1: *"Organizers state the two days of music, dancing, and speeches is expected to draw two million people. But as supporters gathered... riot police deployed..."*. When asked, *"Did the protesters* GATHER *while the organizers* MADE A STATEMENT*?"* , answering correctly requires chaining events: $music \rightarrow draw\_crowd \rightarrow gather$, while recognizing that $riot\_police\_deployed \dashv organizers\_state$, where $\rightarrow$ denotes an "enables" relation and $\dashv$ denotes a "blocks" relation.

Such questions require structured event reasoning, where causal graphs make dependencies explicit by surfacing ENABLE and BLOCK relations that go beyond surface cues (Regan et al., 2023; Chambers and Jurafsky, 2008; Dunietz et al., 2020; Jain et al., 2023; Chi et al., 2024). Without structure, LLMs often rely on shallow lexical patterns and miss deeper event logic.

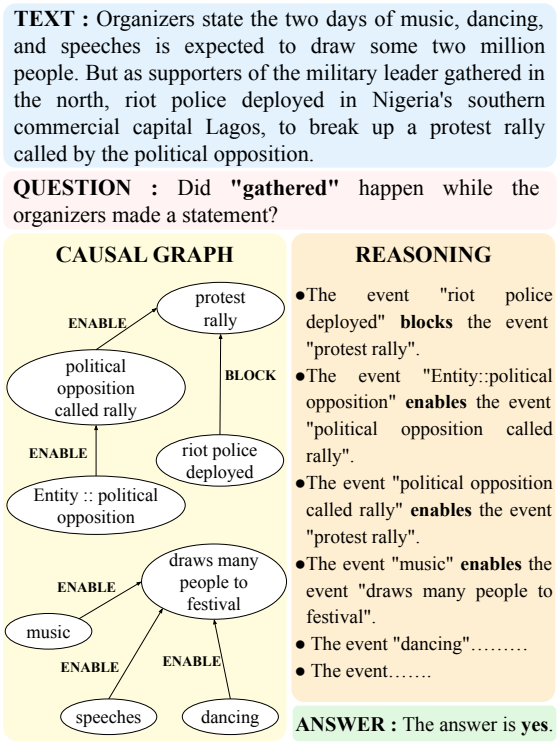


Figure 1: Illustrative example from the TORQUESTRA dataset. **Top:** Narrative passage and a binary event–based question. **Left:** Annotated causal graph showing ENABLE and BLOCK relations between events. **Right:** A step–by–step reasoning trace that follows the graph to support causal inference. Together, the graph and reasoning highlight how structured event relations enable models to answer questions that require indirect causal chaining.

We explore how structured causal knowledge can guide large language models in reasoning about events. Specifically, we introduce **TAG–EQA**–**T**ext–**A**nd–**G**raph for **E**vent **Q**uestion **A**nswering–a prompting framework that converts

[1]Code and data available at https://github.com/MaithiliKadam4/TAG-EQA

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 304–315
November 8-9, 2025

causal event graphs into natural language cues and embeds them directly into the prompt. Rather than fine–tuning the model, TAG–EQA steers its inference by aligning causal structure with prompt format, enabling models to reason more coherently about event dynamics. It spans nine prompting configurations, combining three strategies (zero–shot, few–shot, and chain–of–thought) with three input modalities (*text–only*, *graph–only*, and *text+graph*). While this space is broad, our analysis reveals that causal graphs are especially effective when paired with reasoning-oriented prompts such as chain–of–thought. See Section 3 for full details.

In our experiments on the TORQUESTRA dataset (Regan et al., 2023), TAG–EQA improves accuracy by approximately 5% over text-only baselines, with gains rising to 12% in zero-shot and 18% in chain-of-thought settings. To better understand where structure helps, we group questions into thirteen semantic categories—such as *causal*, *temporal*, and *hypothetical* reasoning—and find that graph-based prompts are particularly effective for *causal chains*, *temporal dependencies*, and *counterfactual what-if scenarios*, where structured event interactions are central to answering correctly. Because these experiments rely on gold human-annotated graphs, the reported numbers should be interpreted as an upper bound on the benefit of structured input; robustness to automatically induced or noisy graphs remains future work.

Our contributions are as follows:

- We introduce *TAG-EQA*, a prompting framework that incorporates causal event graphs into LLM inputs via natural-language serialization—without requiring model fine-tuning.
- We evaluate nine prompting configurations across three strategies and three input types, using T5-XXL, Qwen-32B, and GPT-3.5/4o.[2]
- We examine how causal graphs and reasoning traces interact, and when they improve model performance.
- We report accuracy trends across thirteen semantic question types to identify where structured and/or reasoning-based input helps the most.

2 Related Work

Prior work on event modeling, causal reasoning, and prompt engineering has independently advanced narrative QA. We synthesize these strands by embedding structured causal graphs into prompt formats to guide event-centric inference in LLMs.

2.1 Event Modeling

Narrative understanding has long relied on modeling event relations such as causality, enablement, and sequence. Early work induced event chains using verb–argument frames (Chambers and Jurafsky, 2008), while later approaches inferred causal links from raw text without explicit structure (Dunietz et al., 2020). TORQUESTRA (Regan et al., 2023) builds on this by aligning QA pairs with human-annotated causal graphs, enabling evaluation of structured reasoning in context.

We build on these efforts by treating enable and block relations as first-class prompt components. Each edge is serialized into a natural language sentence, allowing LLMs to ground their reasoning in structured temporal and causal dependencies.

2.2 Cause-Effect Graphs and Causal Reasoning

Causal reasoning from text remains a significant challenge for large language models (LLMs), which often conflate correlation with causation (Yamin et al., 2024). Early methods extracted causal links using pattern-based heuristics (Radinsky et al., 2012), while later approaches employed pretrained language models to infer implicit dependencies from raw text (Dunietz et al., 2020). More recent work has shown that explicitly incorporating cause–effect graphs can improve question answering on narrative and commonsense tasks (Roy et al., 2024; Bethany et al., 2024). However, most prior efforts emphasize direct or temporal links, leaving finer-grained structures underutilized.

However, enabling (A enables B) and blocking (C blocks D) relations remain underexplored despite their value in modeling conditional constraints and counterfactuals. We address this by formalizing them into natural-language prompts that explicitly guide LLM reasoning.

2.3 Prompt Engineering and Chain-of-Thought Reasoning

Prompt engineering enables pretrained language models to perform new tasks without parameter updates, leveraging Zero- and Few- shot in-context learning (Petroni et al., 2019; Brown et al., 2020). Chain-of-thought (CoT) prompting extends this approach by encouraging step-by-step reasoning

[2]GPT-3.5 is used for non-reasoning prompts (Zero and Few), while GPT-4o is used for reasoning (CoT) due to its stronger multi-step inference ability.

through natural language traces (Wei et al., 2022). Enhancements such as self-consistency decoding and automatic CoT generation, aim to improve reliability and reduce dependence on handcrafted examples (Wang et al., 2023; Zhang et al., 2023). Although CoT prompting has shown strong results in arithmetic and symbolic tasks (Wei et al., 2022; Kojima et al., 2022), its use in structured, event-based inference remains limited. We explore this intersection by aligning CoT prompts with causal graphs—letting models reason over explicitly structured event dynamics across prompt formats.

Track Name	Strategy	Modality	Avg. Prompt Length	Reason Length
Zero–Text	Zero	Text	95.2	–
Zero–Graphs	Zero	Graphs	80.6	–
Zero–TAG	Zero	TAG	138.0	–
Few–Text	Few	Text	121.7	–
Few–Graphs	Few	Graphs	178.0	–
Few–TAG	Few	TAG	242.8	–
CoT–Text	CoT	Text	229.2	30.7
CoT–Graphs	CoT	Graphs	287.6	30.7
CoT–TAG	CoT	TAG	336.8	30.7

Table 1: **Prompt lengths for each TAG–EQA track:** Prompt lengths (tokens) across the three strategies (Zero, Few, CoT) and input modalities (Text, Graphs, TAG). CoT prompts include explicit reasoning traces.

3 Method

TAG-EQA investigates whether structured causal knowledge and explicit reasoning can improve event-based question answering (QA) when delivered through prompt design. We vary two orthogonal factors: (1) the *prompting strategy*—Zero, Few, or CoT, and (2) the *input modality*—Text, Graphs, or TAG (text and graph combined). This results in nine prompting configurations, each combining a reasoning style with one or more input sources. We evaluate these configurations across three instruction-tuned LLMs (T5-XXL, Qwen-32B, and GPT-3.5/4o) to understand how prompt structure and content influence QA accuracy.
Figure 2 provides a visual overview of our prompting pipeline. Prompts are constructed by combining a narrative passage, a natural-language representation of a causal graph (if present), and optionally, demonstration QA examples or intermediate reasoning traces. See Section 3.3 for full details.

3.1 Task Formulation

Each instance consists of a short passage P, a yes/no question Q about events in P, and optionally a causal event graph G–either an *instance* or *schema* graph–encoding directed ENABLES/BLOCKS dependencies. An ENABLES edge ($A \rightarrow B$) indicates that event A provides a prerequisite or supportive condition for event B to occur, while a BLOCKS edge ($C \dashv D$) denotes that event C prevents, interrupts, or otherwise inhibits event D. The model must output "yes" or "no." In CoT prompts, it must first produce a natural-language reasoning trace, then the final answer.

3.2 Dataset: TORQUESTRA

We use the TORQUESTRA dataset (Regan et al., 2023) to construct prompts for event-based QA grounded in causal and temporal structure. Each instance provides a short narrative passage, a yes/no question, and one or more directed causal graphs with ENABLES/BLOCKS edges. We generate prompts for all nine configurations by combining QA pairs with the corresponding passage and/or a verbalized version of the graph (i.e., each edge serialized into a natural-language sentence such as "Event A enables Event B"), formatted according to the selected prompting strategy (Zero, Few, or CoT) and input modality (Text, Graphs, or TAG).

All prompts are derived from the human-refined subset ($\text{TORQUESTRA}_{\text{human}}$), which provides gold-standard causal graphs. Figure 1 illustrates a typical example: the passage, graph, and question are used to build the prompt, although the figure content is for exposition only and not used verbatim.

Our filtered Full split contains 477,549 QA instances, balanced across strategies and input types. To support ablations and cost-sensitive models, we also define a Small subset of 1,024 instances, stratified by question category and prompting configuration. Unless otherwise noted, results are reported on the Full set, with Small results shown separately for GPT-based models.

Prompt length varies considerably by configuration. For example, Zero–Text prompts average 95.2 tokens, while CoT–TAG prompts reach 336.8 tokens on average, with reasoning traces contributing 30.7 tokens. These differences affect both model performance and context length constraints.

As shown in Table 1, 26.5% of answers are "yes" and 73.5% are "no." Each causal graph omits approximately 5.3 events on average, requiring inference over missing links—a key motivation for evaluating the utility of structured prompts.

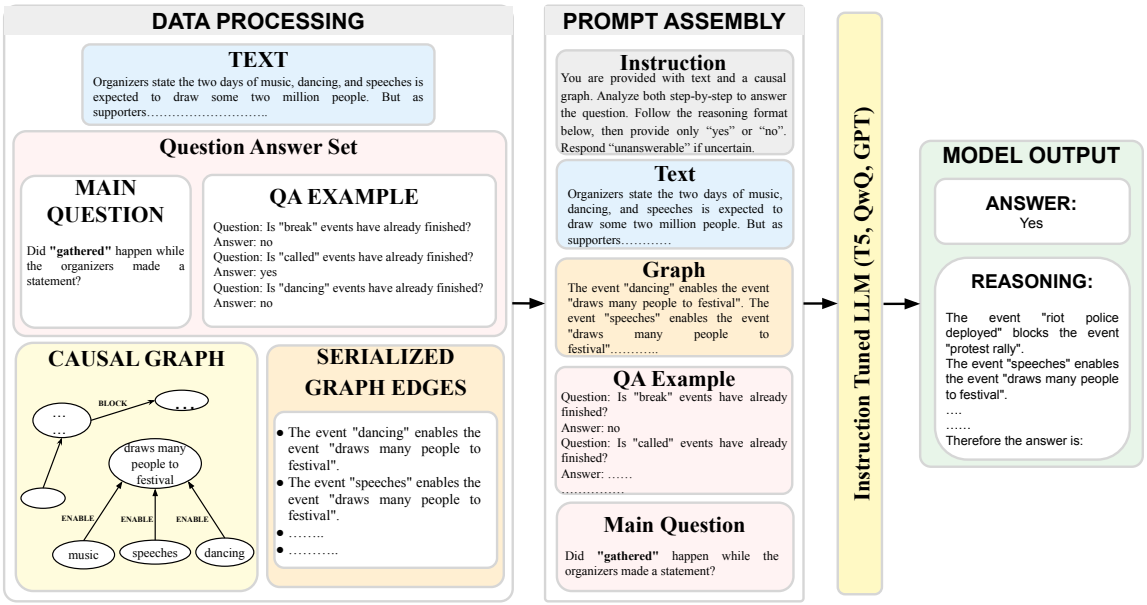


Figure 2: **Overview of our QA prompting pipeline for TAG + CoT configuration.** From left to right: a narrative passage and associated causal graph are processed into a structured input. The causal graph is serialized into natural-language edges (yellow), and the original passage text is retained (blue). Prompt assembly combines task instructions, the text, the graph, in-context QA examples, and the main question into a single input to the instruction-tuned LLM (T5, QwQ, or GPT). The model produces both a yes/no answer and a step-by-step reasoning trace grounded in the causal structure (green).

3.3 Prompt-Track Configurations

TAG-EQA combines three prompting strategies with three input modalities, yielding a 3×3 grid of nine prompt configurations (e.g., Zero–Text, Few–Graphs, CoT–TAG) evaluated in Section 5. Strategies differ in how much supervision or explicit reasoning they include; modalities differ in whether the model receives natural language text, a structured graph, or both.

Each strategy is paired with one input modality:

Text : the narrative passage only,

Graphs : a serialized causal graph representing event dependencies,

TAG : both the passage and graph, concatenated.

Zero-shot prompting (Zero) In the Zero track, the model receives task instructions, the input modality (Text, Graphs, or TAG), and the target yes/no question—without demonstrations or reasoning traces. This setting tests whether an LLM can reason directly from the input without prior examples. For instance, using only the Text portion of Figure 1, the model must decide whether "gathered" occurred while the organizers made a statement.

Few-shot prompting (Few) Few prompts add three in-context demonstrations that match the target configuration. Text-only prompts show how to answer using narrative context; graph-based prompts illustrate how causal structure maps to a yes/no label. TAG prompts include both modalities. This setting provides the model with worked examples aligned to the input type.

Chain-of-thought prompting (CoT) CoT prompts build on Few by requesting an explicit reasoning trace. Demonstrations include step-by-step rationales showing how answers are derived from temporal or causal chains. When the graph is present, traces may reference edges (e.g., `BLOCKS`) or event dependencies. This strategy encourages multi-step inference grounded in structured input. See Appendix A.1 for formatting templates across all nine configurations.

3.4 Causal Graph Integration

Each causal graph G is verbalized into natural language using one sentence per edge—e.g., *"Event A* ENABLES *Event B."* or *"Event C* BLOCKS *Event D."* Sentences are ordered topologically to preserve causal flow and reduce reference distance. Events are described using surface forms from the original passage to ensure clarity and self-containment.

Apart from the presence or absence of the passage,

all other aspects of the prompt remain fixed: task instructions, in-context examples (in Few), and reasoning traces (in CoT) follow a shared scaffold across modalities. This design isolates the effect of graph structure while controlling for phrasing, format, and token budget.
Examples of full Text, Graphs and TAG prompts for each strategy track appear in Appendix A.1.

3.5 Model Families and Setup

We evaluate three instruction-tuned large language model (LLM) families across the full 3×3 TAG-EQA prompt matrix:

- **T5-XXL** (Google): 11B encoder–decoder model, pretrained with UL2 and fine-tuned on diverse instructions.
- **Qwen-32B (QwQ)** (Alibaba): 32B multilingual decoder trained with chat and instruction tuning.
- **GPT-3.5-Turbo** and **GPT-4o** (OpenAI): proprietary decoder-only models; GPT-3.5 is used for Zero and Few, while GPT-4o is reserved for CoT evaluation on a smaller subset due to cost.

All models use greedy decoding (temperature = 0). Inputs are truncated to model-specific context limits (T5: 1k, Qwen: 2k, GPT: 16k), with graph content prioritized over passage if needed. CoT answers are extracted via regex targeting "Therefore, the final answer is: <yes/no>".
T5 and Qwen are evaluated on both Full and Small subsets; GPT-3.5 runs Zero/Few on Full, and GPT-4o runs CoT on Small due to API constraints.

4 Evaluation

We evaluate **TAG-EQA** using binary classification accuracy: the percentage of questions answered correctly as "yes" or "no." Each model is tested across all nine configurations—three prompting strategies (Zero, Few, CoT) × three input modalities (Text, Graphs, TAG).
For CoT prompts, we extract the final answer using a regex targeting phrases like "Therefore, the final answer is: yes." If absent, we fall back to the first standalone yes/no token[3]. This ensures consistent evaluation across models with variable output formats.
We report results on both the full TORQUESTRA test set (Full, 477K examples) and a 1,024-instance Small subset used for low-resource and cost-sensitive runs (GPT-4o).

To analyze how structure and reasoning affect performance across reasoning types, we group questions into thirteen semantically grounded clusters derived from TORQUESTRA annotations. These extend the original eight-category taxonomy to include finer-grained types such as *positive*, *negative*, *existential*, and *counterfactual*. Accuracy is reported per cluster and per configuration.

See Appendix A.5 for full cluster definitions and results.

5 Results

We evaluate how prompting strategy and input modality affect event-based QA performance across three instruction-tuned LLMs: T5-XXL, Qwen-32B (QwQ), and GPT models (GPT-3.5 and GPT-4o). Each model is tested under nine prompting configurations (Zero/Few/CoT × Text/Graphs/TAG). T5 and Qwen are evaluated on both the full TORQUESTRA test set (Full) and a 1,024-example subset (Small). GPT-3.5 and GPT-4o are evaluated only on the Small subset: GPT-3.5 for Zero and Few (non-reasoning), and GPT-4o for CoT (reasoning), due to API cost and throughput constraints.

Across models, Few-shot prompting consistently outperforms Zero-shot in Text-only settings. CoT prompting yields mixed results: QwQ achieves the highest overall accuracy (74.8%) with TAG-CoT, while T5 performs best with Few-Text. For T5, accuracy drops when CoT is combined with structured input, suggesting difficulty integrating reasoning traces and graph content.

Graphs inputs significantly enhance zero-shot and CoT performance for QwQ, sometimes outperforming TAG inputs. However, modality fusion does not always help: TAG configurations often underperform compared to single-modality prompts, particularly for T5. GPT results remain relatively flat across input types, with Zero-Text (58.7%) performing best for GPT-3.5, and modest gains from CoT in GPT-4o.

These findings highlight the importance of model-aware prompt design: performance gains depend not just on adding structure or reasoning, but on whether a given model can effectively integrate them.

[3] Regex: `[Tt]herefore,.*answer is: (yes|no)`

Model	Dataset	Zero	Few	CoT
T5	Full	54.08	58.49	55.21
	Small	52.64	59.47	55.96
QwQ	Full	66.78	70.21	65.77
	Small	68.03	78.32	73.70
GPT	Full	-	-	-
	Small	58.65	52.73	72.28

Table 2: **Prompt-Type Accuracy (%) Comparison on Text–Only Input.** Each model is evaluated on the Full and Small TORQUESTRA subsets. Few–shot prompting consistently outperforms Zero–shot on both scales. CoT shows limited gains on Full, but outperforms Few on Small for QwQ and GPT-4o. GPT results are based on Small only due to cost constraints.

5.1 Does reasoning (CoT) improve performance over Zero or Few-shot using just text?

We begin by comparing Zero, Few, and CoT prompting under Text–only inputs. As shown in Table 2, Few consistently outperforms Zero across models and data sizes. For example, T5 improves from 54.1% to 58.5% on Full, and QwQ improves from 66.8% to 70.2%.

CoT prompting shows mixed effects in the absence of graph input. On the Full set, it underperforms Few for both T5 and QwQ. However, on the Small subset, CoT provides noticeable gains: QwQ improves from 70.2% to 73.7%, and GPT-4o achieves 72.3%, outperforming GPT-3.5's Few score of 52.7%.

These results suggest that chain-of-thought reasoning can help in low-data settings or with models tuned for step-by-step reasoning, such as GPT-4o. Still, Few remains the most reliable strategy when using plain text alone—especially on larger test sets. GPT results are limited to the Small subset: Zero and Few use GPT-3.5, while CoT uses GPT-4o.

5.2 Are Graphs helpful when used alone or combined with Text?

We evaluate the effect of input modality—Text, Graphs, and TAG—under both Zero and Few prompting.

As shown in Table 3a, Graphs–only inputs consistently outperform Text–only across models. For instance, QwQ improves from 66.8% (Text) to 78.8% (Graphs), and T5 gains from 54.1% to 58.0%. Combining Text and Graphs in a TAG prompt further improves performance for QwQ (74.5%) but reduces accuracy for T5 (52.6%). On the Small subset, GPT shows limited variation across modalities – ranging from 56.8% to 58.8% – indicating relative insensitivity to structured input in zero-shot settings. Overall, these results suggest that causal graphs substantially aid zero-shot inference, but modality fusion (Text+Graph) can introduce conflicts depending on the model.

Table 3b shows that Few–shot prompting generally boosts absolute performance compared to Zero. For example, QwQ achieves its highest score (79.4%) with TAG, confirming that demonstrations and graph input are complementary. GPT gains from Graph input (62.9%) compared to Text-only (52.7%), while T5 shows limited or negative gains from structure, dropping from 59.5% (Text) to 50.8% (TAG). These results suggest that few-shot demonstrations amplify the utility of structured graphs for models like QwQ, and GPT, but highlight integration challenges for T5.

5.3 When reasoning is explicitly used, does adding a Graphs help or hurt?

We now examine the effect of input modality under CoT prompting. As shown in Table 4, Graphs–only inputs improve performance for models capable of leveraging structured representations. QwQ achieves its highest accuracy (74.8%) with TAG, while also showing strong performance with Graphs–only input (72.7%).

T5 shows modest gains from Graphs input: on Full, accuracy rises from 55.2% (Text) to 56.9% (Graphs), but drops to 50.4% with TAG, suggesting that reasoning traces may conflict with multimodal inputs for models not tuned for integration. This trend persists on the Small subset.

GPT, evaluated only on Small, shows a slight drop in performance with TAG (70.6%) compared to Text–only input (72.3%), while Graphs–only input yields comparable performance (71.1%). This suggests that GPT–4o does not consistently benefit from structured input when combined with reasoning traces in zero-shot settings.

Overall, these results suggest that graph-augmented reasoning is most effective when the model can exploit structure natively–QwQ benefits most—while other models struggle to integrate multiple information sources effectively under CoT prompting.

Model	Dataset	Text	Graphs	TAG
T5	Full	54.08	57.96	52.58
	Small	52.64	58.89	52.50
QwQ	Full	66.78	78.77	74.48
	Small	68.03	68.09	67.77
GPT	Full	-	-	-
	Small	58.65	56.84	58.79

(a) Zero–shot prompting.

Model	Dataset	Text	Graphs	TAG
T5	Full	58.49	57.54	51.87
	Small	59.47	57.32	50.76
QwQ	Full	70.21	70.48	79.37
	Small	78.32	70.51	78.10
GPT	Full	-	-	-
	Small	52.73	62.99	59.28

(b) Few–shot prompting.

Table 3: **Input Modality Accuracy (%) Comparison. (a)** Zero-shot results: Graphs–only inputs outperform Text–only for most models, with QwQ showing the largest gains. **(b)** Few-shot results: Demonstrations improve overall accuracy, and combining graphs with examples (TAG) is especially effective for QwQ, and GPT, while T5 struggles with multimodal integration.

Model	Dataset	Text	Graphs	TAG
T5	Full	55.21	56.85	50.35
	Small	55.96	56.74	49.56
QwQ	Full	65.77	72.68	74.75
	Small	73.70	71.55	72.05
GPT	Full	-	-	-
	Small	72.28	71.07	70.61

Table 4: **Input Modality Accuracy (%) Comparison in CoT Prompting.** Each model is evaluated using CoT prompting on the TORQUESTRA dataset. T5 and QwQ show modest to strong gains with Graphs inputs. QwQ performs best with combined inputs (TAG), while GPT-4o shows minimal benefit from multimodal prompts. GPT is evaluated only on the Small subset due to API constraints.

Model	Dataset	Best Config	Accuracy%
T5	Full	Few + Text	58.49
	Small	Few + Text	59.47
QwQ	Full	Few + TAG	79.37
	Small	Few + Text	78.32
GPT	Full	-	-
	Small	Zero + Text	72.28

Table 5: **Best-Prompting Configuration per Model.** Top-performing strategy and input modality for each model on the Full and Small TORQUESTRA subsets. GPT results are based on the Small set only due to API cost constraints.

5.4 Which prompting strategy works best for each model?

To better understand model-specific behavior, we report each model's highest-scoring configuration across all nine prompt types (Zero/Few/CoT × Text/Graphs/TAG) for both the Full and Small TORQUESTRA subsets (Table 5). Each entry reflects the optimal combination of prompting strategy and input modality at a given data scale.

QwQ achieves the highest overall accuracy (79.4%) on Full with Few+TAG, showing strong ability to integrate demonstrations and graph input. On Small, it performs best with Few+Text, indicating that graph augmentation is less beneficial under data constraints.

T5 reaches its top accuracy with Few+Text on both subsets (58.5% and 59.5%), showing a clear preference for demonstrations alone. Performance declines when graph input or reasoning traces are included, consistent with earlier observations.

GPT, evaluated only on Small, performs best with Zero+Text (72.3%), suggesting that neither examples nor reasoning traces help much in this setup.

Overall, effective prompting varies by model and scale: structure and reasoning help only when the model can integrate them meaningfully.

5.5 Do certain question types benefit more from reasoning or graphs?

We evaluate model accuracy across thirteen question types derived from TORQUESTRA annotations, extending the original eight clusters (see Appendix A.6 for details). Figure 3 shows accuracy under TAG input—combined text and graph—across Zero-shot, Few-shot, and Graph-CoT prompting. **QwQ and GPT perform best in causal and temporal categories** such as *causal*, *past*, *positive*, and *temporal_conflict*, particularly with CoT prompting. Structured input and reasoning traces appear to help these models handle abstract event relationships.

QwQ and GPT perform best on structured categories—such as *causal*, *past*, *positive*, and *temporal_conflict*—especially when using CoT prompting. Structured input and step-by-step reasoning

appear to help these models capture abstract event relationships.

T5 performs best with Few-shot prompting, but its performance drops with Graph-CoT on speculative or underspecified types like *possible*, *present*, and *unknown*, suggesting difficulty integrating structure and reasoning.

Appendix Figures 4 and 5 show that Text prompts gain from Few-shot examples but struggle with relational types, while Graphs prompts provide stronger performance for QwQ in categories like *causal*, *past*, and *event*.

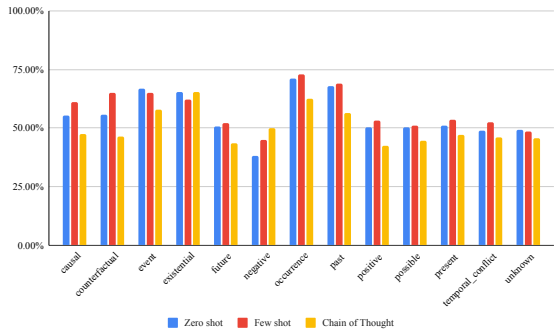
Overall, structured prompting benefits causal and temporal reasoning, with QwQ and GPT showing the strongest gains from graph-augmented CoT. Ambiguous or speculative questions remain difficult across models.

6 Conclusion

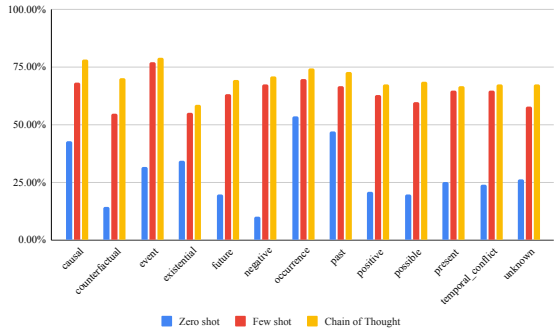
We introduced TAG-EQA, a systematic framework for evaluating event-based question answering (QA) in large language models (LLMs) using structured causal graphs and reasoning-driven prompting. Our experiments covered nine prompting configurations—three strategies (Zero, Few, CoT) crossed with three input modalities (Text, Graphs, TAG)—evaluated on three instruction-tuned LLMs: T5, Qwen (QwQ), and GPT models. Causal graphs consistently improved accuracy on event-centric questions, particularly for relational categories such as *causal*, *past*, and *temporal_conflict*. QwQ achieved the highest overall performance when combining structure and reasoning (TAG+CoT), while T5 performed best with Few+Text and showed limited gains from structured input. GPT models, evaluated only on a smaller subset, showed moderate benefits from CoT prompting but little sensitivity to input modality. Ambiguous or underspecified categories—such as *possible* and *unknown*—remained challenging across models and prompting styles. These findings highlight both the strengths and limitations of using structured causal input to guide reasoning in LLMs.

7 Limitations and Future Work

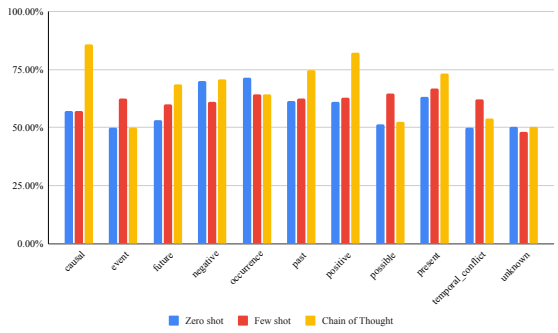
Our evaluation relies on expert-annotated causal graphs from TORQUESTRA, which provide clean structure but do not reflect the sparsity or noise of automatically induced graphs. The reported numbers should therefore be interpreted as an *upper bound* on the benefit of structured input. Prompt construction is also manually designed—including example selection and reasoning trace format—which may limit generalization to new domains without automation. Performance further varies across models: QwQ benefits most



(a) T5 under TAG: *Few—Text* is strongest overall; *Graph—CoT* tends to underperform on speculative or underspecified types (*possible*, *present*, *unknown*).



(b) QwQ under TAG: *Graph—CoT* generally leads on structured types (*causal*, *past*, *temporal_conflict*); *Few—Text* remains competitive elsewhere.



(c) GPT under TAG: *Graph—CoT* improves relational categories (e.g., *causal*, *temporal_conflict*); strategy gaps narrow on underspecified types (*possible*, *unknown*).

Figure 3: **Cluster-wise accuracy under the TAG configuration.** Bars denote *Zero—Text* (blue), *Few—Text* (red), and *CoT* with TAG (yellow) across thirteen question types. Subfigures (a–c) report T5, QwQ, and GPT respectively. Text–only and Graph–only cluster results appear in Appendix Figures 4 and 5.

from graph-augmented CoT prompting, whereas T5 and GPT show more modest or inconsistent gains. Due to API cost constraints, GPT models are evaluated only on the Small subset—a full-scale CoT run with GPT-4o would exceed $950.[4] Lastly, our binary QA task simplifies causal reasoning and does not capture the complexity of multi-hop inference or generative outputs.

Beyond these constraints, our study is limited to three instruction-tuned LLM families (T5, QwQ, GPT); other architectures may respond differently to structured prompts. We also restrict evaluation to TORQUESTRA, leaving extensions to broader narrative QA datasets (e.g., NarrativeQA, MCTest) for future work. Finally, while we report average prompt lengths, a systematic study of context budget and scaling effects remains open.

Future directions include automated graph construction, robustness to noisy or incomplete graphs, and adaptive graph selection to filter only edges relevant to a query. Extending TAG-EQA with dynamic reasoning traces, instruction tuning for graph-structured CoT prompting, and applications to generative or interactive tasks—such as story simulation, causal forecasting, or decision support—offers promising next steps for leveraging structured knowledge in real-world applications.

Acknowledgments

We thank the reviewers for their detailed comments and suggestions. Some experiments were conducted on the UMBC HPCF, supported by the National Science Foundation under Grant No. CNS-1920079. This material is also based on research that is in part supported by the Army Research Laboratory, Grant No. W911NF2120076, and by DARPA for the SciFy program under agreement number HR00112520301. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA or the U.S. Government.

References

Mazal Bethany, Emet Bethany, Brandon Wherry, Cho-Yu Chiang, Nishant Vishwamitra, Anthony Rios, and Peyman Najafirad. 2024. Enhancing event reasoning in large language models through instruction fine-tuning with semantic causal graphs. *arXiv preprint arXiv:2409.00209*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797. Association for Computational Linguistics.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. 2024. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems*, 37:96640–96670.

Jesse Dunietz, Sam Thomson, Chris Dyer, and Noah A. Smith. 2020. An interpretable, lexicalized model for implicit event causality. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1713.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2463–2473.

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning to predict from textual data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 613–622.

Michael Regan, Jena D. Hwang, Keisuke Sakaguchi, and James Pustejovsky. 2023. Causal schema induction for knowledge discovery. *arXiv preprint arXiv:2303.15381*.

Kaushik Roy, Alessandro Oltramari, Yuxin Zi, Chathurangi Shyalika, Vignesh Narayanan, and Amit Sheth. 2024. Causal event graph-guided language-based spatiotemporal question answering. In *Proceedings of the AAAI Symposium Series*, volume 3, pages 227–233.

[4] See Appendix A.3 for detailed cost calculations.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Khurram Yamin, Shantanu Gupta, Gaurav R Ghosal, Zachary C Lipton, and Bryan Wilder. 2024. Failure modes of llms for causal reasoning on narratives. *arXiv preprint arXiv:2410.23884*.

Yao Zhang, Cheng Deng, Wayne Xin Zhao, Bing Qin, and Ting Liu. 2023. Automatic chain-of-thought prompting in large language models. *arXiv preprint arXiv:2302.00923*.

A Appendix

This appendix provides additional implementation details, example prompts, and full evaluation results referenced in the main paper. We include: (1) prompt format illustrations, (2) input component breakdowns, (3) per-model prompting results, (4) API cost estimates, and (5) expanded cluster definitions and analysis.

A.1 Prompt Format Examples

We show two full prompt examples in the CoT setting: one using only a causal graph (Graphs) and one using both the passage and graph (TAG). These correspond to the instance in Figure 1.

A.1.1 Graphs – CoT Prompt

```
### Instruction ###
You are provided with a causal graph and
examples showing how to answer. Use only
the graph and answer “yes” or “no” only.

### Graph ###
The event "riot police deployed" blocks
the event "protest rally".
The event "political opposition" enables
the event "political opposition called
rally".
The event "political opposition called
rally" enables the event "protest
rally".
The event "music" enables the event
"draws many people to festival".
The event "dancing" enables the event
"draws many people to festival".
The event "speeches" enables the event
"draws many people to festival".

### Examples ###
Question: Did "protest rally" happen
after "riot police deployed"?
Answer: no
Question: Did "music" cause "draws many
people"?
Answer: yes

### Question ###
Did "gathered" happen while the
organizers made a statement?
### Answer ###
```

A.1.2 TAG – CoT Prompt

```
### Instruction ###
You are provided with text, a causal
graph, and examples showing how to
answer. Integrate both and answer “yes”
or “no” only.

### Text ###
Organizers state the two days of music,
dancing, and speeches is expected to
draw some two million people. But as
supporters gathered in the north, riot
police deployed in Lagos to break up a
protest rally called by the political
opposition.

### Graph ###
[Same graph as above]

### Examples ###
[Same examples as above]

### Question ###
Did "gathered" happen while the
organizers made a statement?
### Answer ###
```

Modality	Input	Zero	Few	CoT
Text	Text	✓	✓	✓
	Graph			
	Examples		✓	✓
Graphs	Text			
	Graph	✓	✓	✓
	Examples		✓	✓
TAG	Text	✓	✓	✓
	Graph	✓	✓	✓
	Examples		✓	✓

Table 6: Components used in each prompt configuration.

A.2 Prompt Component Matrix

Table 6 summarizes which components appear in each of the nine prompting configurations used in TAG-EQA.

A.3 API Cost Estimate

We compute cost estimates for GPT-3.5 and GPT-4o using OpenAI’s May 2025 pricing. Table 7 shows that a full CoT evaluation with GPT-4o would exceed $950, so we restrict GPT results to the Small subset.

Model	Prompt Type	Total Tokens	Cost (USD)
GPT-3.5	Zero-shot	9.7M in / 53k out	$4.95
GPT-3.5	Few-shot	12.6M in / 53k out	$6.36
GPT-4o	CoT	21.1M in / 212M out	$957.16

Table 7: Estimated cost to run GPT models on Full dataset.

A.4 Per-Model Prompting Results

We report accuracy for each model across all 3×3 prompting configurations. These tables complement Section 5.4 and clarify which strategies and modalities are most effective for different architectures.

T5. Performs best with Few+Text, but degrades when structure or reasoning is added.

Prompt Type	Text	Graphs	TAG
Zero	54.1	58.0	52.6
Few	58.5	57.5	51.9
CoT	55.2	56.9	50.4

Table 8: T5 accuracy across all strategies and modalities.

QwQ. Excels with TAG+Few and TAG+CoT. Gains are consistent across most settings.

Prompt Type	Text	Graphs	TAG
Zero	66.8	66.8	74.5
Few	70.2	70.5	79.4
CoT	65.8	72.7	74.8

Table 9: QwQ accuracy across all strategies and modalities.

GPT. Best performance under CoT (GPT-4o). GPT-3.5 shows smaller gains and flat modality sensitivity.

Prompt Type	Text	Graphs	TAG
Zero	58.7	56.8	58.8
Few	52.7	63.0	59.3
CoT	72.3	71.1	70.6

Table 10: GPT accuracy across all strategies and modalities.
Note: All GPT results are reported on the Small subset due to API cost constraints.

A.5 Expanded Cluster Definitions

We extend TORQUESTRA's original eight cluster categories into thirteen to better capture event-centric reasoning. Table 11 aligns our expanded taxonomy with the original groups.

A.6 Cluster-Based Accuracy Analysis

We present accuracy trends by question category across all prompting configurations.

Expanded Category	Original Cluster
causal	causal
counterfactual	causal (extended)
event	event
existential	event (subtype)
future	future
negative	event (negative polarity)
occurrence	event / temporal
past	past
positive	event (positive polarity)
possible	possible
present	present
temporal_conflict	temporal_conflict
unknown	unknown

Table 11: Expanded category mapping.

T5: Best with Few+Text on most clusters. Accuracy drops with Graphs+CoT.
QwQ: Excels with TAG+CoT. Leads in most structured and relational categories.
GPT (3.5/4o): CoT (GPT-4o) performs best across categories like *causal* and *past*; GPT-3.5 (Zero/Few) is stable but less sensitive to modality.

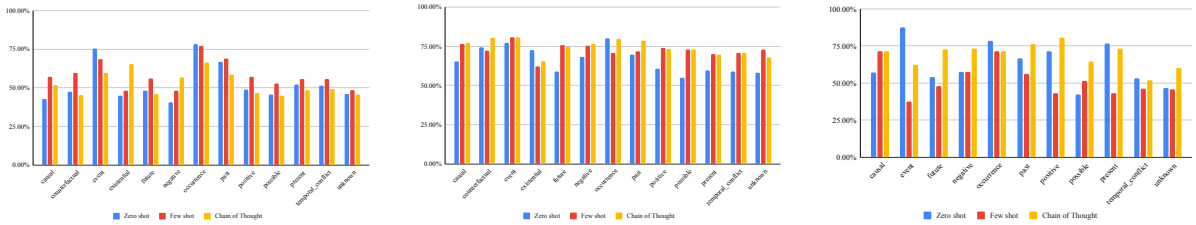


Figure 4: **Cluster-wise Accuracy by Model and Prompting Strategy.** Accuracy across thirteen question categories for each model (T5, QwQ, GPT) under three prompting strategies: Zero–Text (blue), Few–Text (red), and CoT with TAG input (yellow). QwQ and GPT benefit most from graph-augmented CoT prompting on structured categories such as *causal*, *past*, and *temporal_conflict*. T5 performs best with Few–shot but struggles to integrate structure and reasoning. All models show weaker performance on underspecified or speculative categories like *possible* and *unknown*.

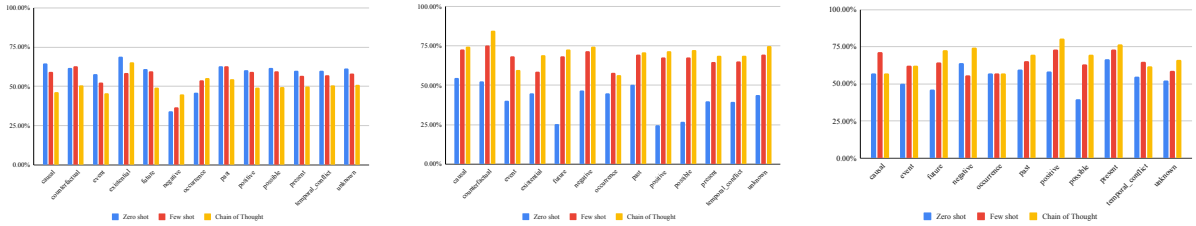


Figure 5: **Cluster-wise Accuracy by Model and Prompting Strategy.** Accuracy across thirteen question categories for each model (T5, QwQ, GPT) under three prompting strategies: Zero–Text (blue), Few–Text (red), and CoT with TAG input (yellow). QwQ and GPT benefit most from graph-augmented CoT prompting on structured categories such as *causal*, *past*, and *temporal_conflict*. T5 performs best with Few–shot but struggles to integrate structure and reasoning. All models show weaker performance on underspecified or speculative categories like *possible* and *unknown*.