Retrieval over Large Language Model's Latent Causal Knowledge Graph for Deductive Reasoning

Anonymous ACL submission

Abstract

Deductive reasoning refers to the task of drawing conclusions based on a premise. While some deductive reasoning benchmarks exist, none focus on causal deductive reasoning and are from real-world applications. Therefore, this paper explores the causal deductive reasoning task conducted by Accident Investigators, who analyze accidents to determine probable causes. Recently, large language models (LLMs) used with prompt engineering techniques like retrieval-augmented generation (RAG) have demonstrated remarkable performance across various natural language processing benchmarks. However, adapting these techniques to handle scenarios with no knowledge bases and to different data structures, such as graphs, remains an ongoing challenge. In our study, we introduce a novel framework leveraging LLMs' decent ability to detect and infer causal relations to construct a causal Knowledge Graph (KG) which represents knowledge that the LLM recognizes. Additionally, we propose a RoBERTa-based Transformer Graph Neural Network (RoTG) specifically designed to select relevant nodes within this KG. Integrating RoTG-retrieved causal chains into prompts effectively enhances LLM performance, demonstrating usefulness of our approach in advancing LLMs' causal deductive reasoning capabilities.

1 Introduction

Large language models (LLMs) have shown impressive performance on some language tasks, however, their ability to plan and reason on complex tasks remains an ongoing challenge (Wei et al., 2022; Valmeekam et al., 2023). In Psychology, the standard test for deductive reasoning consists of giving people premises and asking them to draw conclusions (Evans, 2005; Rips, 1994; Johnson-Laird, 2010). In natural language processing (NLP), RuleTaker (Clark et al., 2020) and ProofWriter (Tafjord et al., 2021) are datasets that challenge models to assign *True* or *False* labels to statements about a probable implication. However, there are no NLP benchmarks on causal deductive reasoning, where the premise are facts about an outcome and the statement is about a probable cause. Furthermore, Huang and Chang (2023); Valmeekam et al. (2022) find that current benchmarks do not truly investigate the reasoning capabilities of LLMs, because the tasks are not meaningfully applied in the real-world.

Researchers have proposed prompt engineering techniques to improve few-shot and zero-shot task performance (Reynolds and McDonell, 2021), like using role-play (Kong et al., 2023; Wang et al., 2023), in-context learning (Xie et al., 2022; Min et al., 2022), and retrieval-augmented generation (RAG) (Lewis et al., 2020; Shao et al., 2023). Recent work has explored using LLMs to retrieve a task-relevant knowledge sub-graph to support reasoning (Li et al., 2024). However, extending these techniques to handle cases where no explicit knowledge base is available, or and how to best use knowledge graphs (KGs) in a RAG-based LLM system remains an open area for research.

This paper focuses on the causal deductive reasoning task performed by Accident Investigators. When an accident occurs, investigators conduct

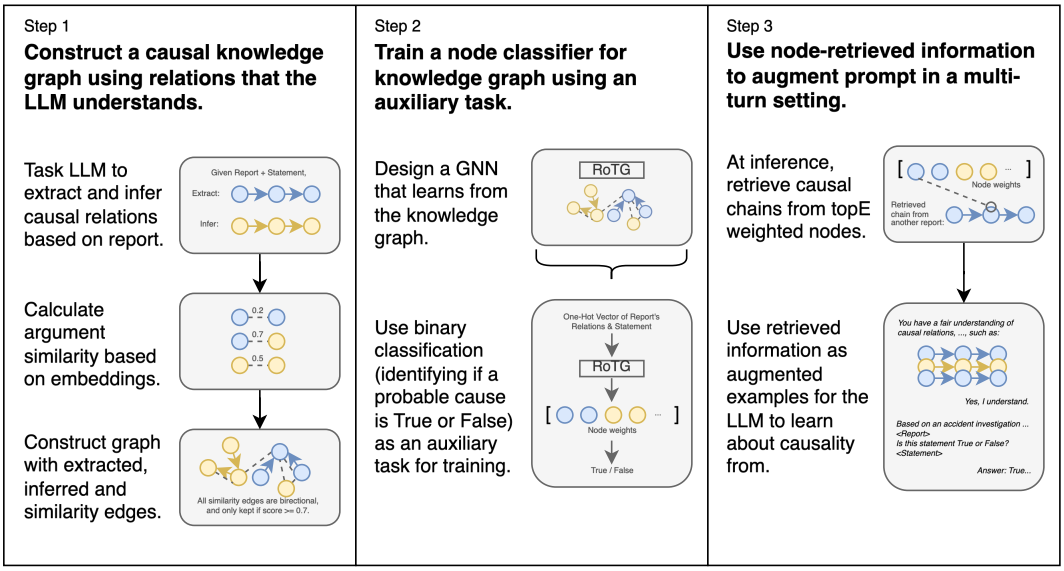

Figure 1: Overview of our proposed methodology. Detailed infographic is available in Appendix Figure 4.

thorough investigations, and come up with a probable cause for the accident. Our main contributions can be summarized as follows:

- We present a task (Section 2) and dataset (Section 3) comprising 631 reports with 11,422 statements. This dataset is curated from original reports written by humans and processed using rules and Claude 2.1. It will be made publicly available.
- We introduce a framework (Figure 1) employing LLMs such as Mistral-Instruct 7B to identify causal relations for constructing a causal KG. Additionally, we trained a RoBERTa-based Transformer Graph Neural Network (RoTG) to select relevant nodes, leveraging deductive reasoning labels as an auxiliary task. (Section 4)
- We observe that incorporating causal relations retrieved from the LLM-constructed KG improves the LLM's causal deductive reasoning performance. (Section 5)

2 Causal Deductive Reasoning

Given an input context C, the goal is to identify the likelihood of a statement s_i being a probable cause of accident a. This likelihood is represented by $y_i \in (0, 1)$, where $y_i = 1$ if s_i is a probable cause and $y_i = 0$ if not. The task is to determine $P(y_i|C)$ for each potential cause s_i within a report context C. Since we have multiple reports in our dataset, the objective extends to calculating $P(y_{it}|C_t)$, where t denotes the report ID. We define $G_t = F_{extract}(C_t)$ as the set of causal relations mentioned in context C_t. The function $F_{extract}(.)$ extracts causal relations from the context. The aggregated set of all extracted relations from the dataset is denoted as G, representing the repository of causal relations of our dataset. Each relation in G_t is represented by a cause and effect pair, denoted as (s_i, s_j).

If a causal chain $x_{it} = (s_i, s_{j1}), (s_{j1}, s_{j2}), ..., (j_k, k) \notin G_t$, then $y_i = 0$. However, if $x_{it} \in G_t$, the rank of y_{it} relative to other potential causes y_{jt} must be considered. Only the top z rank of most important causes can be the probable cause of an accident a. In the case where we only consider the top cause ($z = 1$) as the probable cause, then the probability of $P(y_{it})$ can be reformulated into:

$$\begin{aligned} P(y_{it} = 0) = P(y_{it}|x_{it} \notin G_t) \\ + P(y_i|x_{it} \in G_t, P(y_{jt} = 1) > P(y_{it} = 1)) \end{aligned} \tag{1}$$

$$\begin{aligned} P(y_{it} = 1) = P(y_i|x_{it} \in G_t, \\ P(y_{it} = 1) > P(y_{jt} = 1)) \end{aligned} \tag{2}$$

Since the task is a binary classification task, every example s_{it} is not aware of the other possible s_{jt} for the same report t. Therefore, s_{jt} are causes the model implicit tracks and has to rank against for the current task. Our causal deductive task can be re-framed into two sub-challenges: (1) extracting x_{it} and identifying $x_{it} \in G_t$, and (2) implicitly ranking $P(y_{it} = 1) > P(y_{jt} = 1)$ or not.

Hypothesis 1: Generalizing causal chain to out-of-context In the first challenge, extracting x_{it} and identifying $x_{it} \in G_t$, restricting the knowledge source to a report results in a high chance for there to be gaps in the causal chain. All else fixed, $P(y_{it}|x_{it} \notin G_t)$ will be overestimated (i.e., model predicts more 0s than 1s). If are willing to relax our criteria to check if $s_i \in C_t$ and $x_{it} \in G$, then we are allowing our model to generalize to its own knowledge base, to recognize more valid causal chains, and therefore, increase the probability of predicting $P(y_{it} = 1)$. When working with LLMs, therefore, we could inject causal relations outside of G_t but semantically part of x_{it} to improve prediction.

Hypothesis 2: Ranking importance of cause within context If the LLM is exposed to too many relevant causal relations in the prompt, it would hallucinate and start to always view s_i as the most important probable cause (over other possible options in C_t). However, we do not know z. In some reports, there are a few probable causes. One approach is to explicitly expose the LLM to the available causes in the report, so that we re-ground the response, and in some way, a ranking based on context is encouraged.

3 Dataset & Task Creation

We wish to investigate the LLMs' ability to perform a real-world causal deductive reasoning task. Given an accident description (<CONTEXT>), the model must determine if a sentence about the probable cause of the accident (<STATEMENT>) is *True* or *False*. To facilitate our research, we leverage on reasoning-rich investigation reports from the National Transportation Safety Board (NTSB) [1]. NTSB publishes Accident Reports that provides details about an accident, analysis of the factual data, conclusions and the probable cause of the accident, and the related safety recommendations. There can be one or multiple probable cause(s). We downloaded reports published after Year 2000, across all reported categories (Aviation, Hazardous Materials, Highway, Marine, Pipeline and Railroad).

Report pre-processing Pre-processing was done to convert the PDF reports to JSON, and subsequently, we removed information like headers, page numbers, and table of contents. We identified the probable cause of the accident by searching for the title "Probable Cause". We discarded reports where this match was impossible. Any text before this section is defined as the <CONTEXT>. In our experiments, we constrained our coverage to 157 reports where the context length is $\leq 2,000$ words.

Extracting *True* statements Trailing descriptions in the probable cause were removed. [2] We used Anthropic's Claude 2.1[3] to convert the paragraphs into a list of probable causes. Prompt 1 in Appendix outlines the one-shot prompt template that we used. We manually annotated four examples to measure the extraction performance, of which we found ROUGEL score of 87.46 and BLEU4 score of 75.02. When evaluating by semantic match[4] with a threshold of ≥ 0.7 as a match, Claude 2.1 scored 100% for Recall, 72.92% for Precision, and 84.34% for F1. To summarize, the high scores for the evaluated sample provides us with the confidence to reliably use the extracted probable causes as *True* instances for our main causal deductive task.

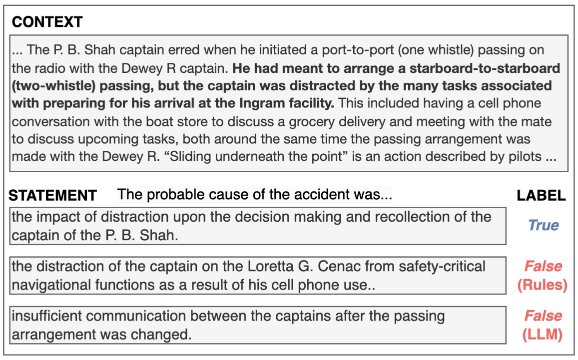

Figure 2: An example report from our dataset.

Generating *False* statements False examples were generated by two methods: (1) rule-based, and (2) LLM-based methods. For rule-based, each *True* statement was matched to three similar-but-not-too-similar statements are generated as negative examples. The degree of similarity between

[1] https://www.ntsb.gov/investigations/AccidentReports/Pages/Reports.aspx

[2] E.g. Descriptions unrelated to the cause (E.g. "*The National Transportation Safety Board determines that the*") were removed.

[3] We intentionally used an LLM different from Mistral when creating our dataset to avoid cases where the LLM recognizes its own phrasing or terms.

[4] We encoded each probable cause item into an embedding using the `princeton-nlp/sup-simcse-roberta-large` encoder (Gao et al., 2021) that was pre-trained on the Natural Language Inference task. Link to their repository: https://github.com/princeton-nlp/SimCSE.

Processing	#Docs	#Statement	*#True*	*#False*	*True %*
Total NTSB	631	11,422	1300	10,122	11.38%
$\leq$ 2000 words	157	2,523	243	2,280	9.63%
Success CRE	**133**	**1,677**	**155**	**1,522**	**9.24%**

Table 1: Data sizes at each filtering stage. The last row represents the working dataset for this paper after successful causal relation extraction (CRE). Our experiments are conducted using 10-folds CV, and the test data sizes per fold are provided in Appendix Table 6.

the *False* examples and the *True* statement was controlled to ensure that false examples are plausible but distinct from the true statement, with similarity scores ranging from 0.5 to 0.75. This approach aims to provide a challenging set of false examples for participants to evaluate. For LLM-based, we used Claude 2.1 (See Prompt 2 in the Appendix) to generate a list of 10 possible causes or contributing causes investigated within the context that are not stated as the final true probable cause.

Our task aims to provide a comprehensive evaluation of participants' ability to perform the challenging causal deductive reasoning task. Table 1[5] presents the statistics for our dataset. After keeping examples that we could extract causal relations described in the next section, our main dataset comprises of 133 reports and 1,677 statements. Of which, 155 are *True* while the remaining 1,522 are *False* probable cause statements. An example report is shown in Figure 2.

3.1 Evaluation Metrics

For each experiment, we report Macro F1, Micro F1 and the accuracy scores for each class label and label source. Since our dataset is small, we used a 10-fold cross validation (split by report ID) to train and generate predictions for the full dataset. Therefore, our evaluation metrics are first computed at the fold level, then averaged, where both the mean and standard deviations of each metric are reported. When making comparisons between two models, P-values are indicated by: $* < 0.15$, $** < 0.10$, $*** < 0.05$.

4 Causal KG RAG with LLM

We mentioned in Section 2 that we wish to help the LLM recognize generalized $(j_a, j_b) \in D$ by injecting relevant causal relations outside of G_t. However, we do not have a knowledge base for G. We also do not have any annotations for the intermediate causal chains that might be relevant given a probable cause i and accident a. To work around these problems, we constructed our knowledge base using the LLM itself. Afterwhich, we designed a novel graph-based retriever model, trained on the auxiliary binary classification task, to select relevant nodes.

4.1 Step 1. Mining LLM's Latent Causal KG

We wish to investigate properties regarding Equations 1 and 2. However, we do not have a knowledge base. Therefore, we separately tasked the LLM to mine the causal relations it recognizes and understands. Specifically, we mined two types of causal relations:

Extracted causal relations We tasked the LLM to extract all causal relations expressed within the <CONTEXT>. Prompt 3 in the Appendix outlines our zero-shot prompt, with only instructions about the desired output format.

Inferred causal relations We tasked the LLM to infer the chain of causal relations that could possibly link the cause stated within the <STATEMENT> to the accident stated within the <CONTEXT>. Prompt 4 in the Appendix outlines our zero-shot prompt, with only instructions about the desired output format. The causal chains from this step can be viewed as the LLM's hallucinated version of x_{it}.

Causal KG To maximize the size of our knowledge store, we constructed our heterogeneous causal knowledge based on a slightly larger dataset of 157 reports and 2,523 statements, which provided us with 4,128 extracted cause-effect pairs and 22,685 inferred cause-effect pairs. Reports with contexts longer than 2,000 words did not fit into our models' input context, so we did not explore the full dataset, although it would be an important future work to extend the size of the knowledge store further.

Our KG $G = (V, E)$ is a collection of nodes $V = \{(v_1, v_2, ..., v_n)\}$ and directed edges $E = \{(v_1, v_2), (v_2, v_3), ...\}$. The edges are directed, and comprises of three possible types: extracted, inferred, or similar. For extracted and inferred relations, a directed edge (v_x, v_y) represents the presence of causality between the two nodes, where v_x is the cause argument and v_y is the effect argument. To prevent a sparse graph, prior causal KG research employ various clustering (Tan et al., 2023) or generalization (Radinsky et al., 2012)

[5] We will release the full dataset of 11,422 statements to the community.

methods to group semantically similar arguments together. For us, we opted for a simple (and shown to be effective in Section 5.1) approach by adding bidirectional edges between two nodes v_x and v_y, weighted by the similarity score ss, for all node pairs with similarity score $ss > 0.7$. Overall, our final G is a collection of 16,675 nodes and 23,493 edges. The distribution of edge types are: 1,822 extracted, 11,399 inferred, and 10,272 similar.

4.2 Step 2. Node Selection over Causal KG

We re-frame our retrieval task as a node classification task: Given a causal KG, we wish to extract the most important and relevant nodes (arguments) to include in our downstream prompt. Since we have no labels as to what helps the LLM learn, we used the the binary classification task (to classify if a <STATEMENT> is *True* or *False*) as an auxiliary task to train our model. The model is encouraged to learn from the KG, and at inference, we discard the classification head and keep top-E nodes with highest node weights as pointers to obtain information for RAG.

Our retriever module uses a RoBERTa-based Transformer GNN (RoTG) framework. Since a traditional RoBERTa model (Liu et al., 2019)'s input token limit of 512 is too small for our reports, we designed a workaround that does not require the long <CONTEXT> sequences as inputs. Our model is trained only by the following inputs: (1) Encoded <STATEMENT> (r_i represents the [CLS] token vector with e features) and (2) A one-hot encoded vector (oh) assigned to each node if the span does appear in the extracted or inferred causal relations (1 if appear, 0 otherwise).

Node classification module Our initial node features were represented by Q_1, an attended representation of Q'_1. Q'_1 is a concatenation of the RoBERTa-encoded frozen embeddings for each node description s (R is a $n \times e$ matrix comprising of n nodes, an input that does not change over training) and the two one-hot vectors ($oh_{\text{extr}}, oh_{\text{inf}}$) indicating if the node was extracted or inferred based on the context and target statement or not. The attention mechanism then computes the attention weights between the node features Q'_1 and the target statement embedding r_i to generate the cross-attended node feature matrix Q. Since our graph is heterogeneous, we require message passing across edge features. Hence, we employed the Transformer (Vaswani et al., 2017) Graph Convolutional Network (TransformerGCN) (Shi et al., 2021), which helps to incorporate edge features into the multi-head attention for graph learning. The architecture of TransformerGCN is outlined in Appendix Section C.1.

$$r_i = \text{RoBERTa}(s_i) \tag{3}$$
$$R = \text{RoBERTa}(S) \tag{4}$$
$$Q'_1 = [R, oh_{\text{extr}}, oh_{\text{inf}}] \tag{5}$$
$$Q_1 = \text{Attention}(Q'_1, r_i, r_i) \tag{6}$$
$$ow_i = \text{TransformerGCN}(G_{(Q_1,E)}) \tag{7}$$

Auxiliary task training We multiplied the local graph weights ow_i onto the global node embeddings R, obtaining our node embeddings Q_2 that are now customized for our inputs. We proceeded with another round of message passing using TransformerGCN over our global graph, and obtained a vector representing the scores each node contributes (nw_i). We incorporated a skip-connection by concatenating nw_i with the original statement embedding r_i and applied dropout and layer normalization layers to get o_i. Subsequently, we ran o_i through multiple rounds of Linear layers, with LeakyReLU in between. In the last layer, we used a Linear layer with output dimension of 2 to obtain logits for our binary classification task.

$$ow'_i = \text{topKGating}(ow_i) \tag{8}$$
$$Q_2 = ow'_i R \tag{9}$$
$$nw_i = \text{TransformerGCN}(G_{(Q_2,E)}) \tag{10}$$
$$o_i = \text{LayerNorm}(\text{Dropout}([r_i, nw_i])) \tag{11}$$
$$o_i^{(l+1)} = W^{(l)} o_i^{(l)} + b^{(l)} \tag{12}$$

Each model was trained for 8 epochs, with an effective batch size of 8. Since our dataset is extremely unbalanced (∽9% *True* only), we also balanced class labels by oversampling *True* examples, such that the ratio is 1:2 for *True*:*False*, then included the post-oversampling class weights into the CrossEntropyLoss function. Model specifics are provided in Appendix Section A.

4.3 Step 3. Prompt Engineering with LLM

During inference, we selected the top-E nodes with the highest scores based on node weights, ow_i. Subsequently, we obtained the nodes' original reports' extracted or inferred causal chains, then kept all chains that contain the node span. We investigated 9 distinct prompt formats (see Prompts 5 to 13 in the Appendix), incorporating variations

	Macro F1	Micro F1	Accuracy		
			True	*False* (Rules)	*False* (LLM)
All	55.43 (6.09)	**83.96 (9.07)**	31.01 (31.19)	**67.44 (34.41)**	**99.45 (0.86)**
Similarity Only	**56.97 (6.05)**	82.75 (8.39)	34.70 (26.65)	66.77 (25.59)	98.14 (5.22)
Causality Only	56.90 (6.62)	81.48 (9.35)	**39.56 (30.79)**	60.62 (30.83)	97.92 (5.63)

Table 2: RoTG classification performance when trained over different edges types in G. Highest score per column is in bold. All scores are not statistically significant from the first row.

Relations Retrieved	Macro F1	Micro F1	Accuracy		
			True	*False* (Rules)	*False* (LLM)
None	70.36 (7.07)	90.30 (1.78)	46.53 (13.21)	92.23 (3.66)	95.69 (1.86)
Semantic	72.50 (6.37)	91.24 (1.40)	48.72 (11.04)	92.99 (2.48)	**96.54 (1.93)**
RoTG	**73.19 (7.01)**	**91.65 (1.42)****	**49.49 (13.47)**	**94.31 (3.49)**	96.37 (1.37)

Table 3: Mistral Instruct with *None*, Semantic, and RoTG (Ours) retrieval-augmented relations. Highest score per column is in bold. P-values against *None* scores indicated by: $*< 0.15$, $**< 0.10$, $***< 0.05$.

of retrieved, extracted, and inferred causal relations. Our best-performing prompt format (Prompt 10) consists of retrieved information that were presented as a multi-turn prompt: Initially, retrieved relations were introduced to the model. Next, we set the models' response to be "*Yes I understand.*". Finally, a description of the task followed in the subsequent reply. We found that including the retrieved information in the same responses as the task description led to poor performance.

All relations underwent post-processing to remove similar causal chains, defined by a Levenshtein ratio ≥ 0.8, with duplicates resolved by retaining only the first instance. Additionally, we limited each relation type to the first 10 rows of causal chains. Subsequent experiments revealed that such cleaning procedures enhanced the model's F1 scores. We categorized a model response as *False* if the word "False" appeared in any part of the response, and *True* otherwise. Due to the length of the reports, particularly when utilizing Mistral as our LLM, in-context learning was not feasible. Consequently, all experiments were conducted in a zero-shot manner.

5 Experimental Findings

This paper focuses the investigation on the Mistral-Instruct 7B LLM (Jiang et al., 2023). We used Mistral to extract and infer causal relations for our KG as described in Section 4.1, then trained RoTG over this KG as described in Section 4.2. Finally, we tested Mistral on the causal deductive reasoning task as described by Section 4.3.

5.1 Auxiliary Task Performance

Investigating RoTG's performance on the causal deductive task serves as a proxy of how helpful would the LLM's latent causal KG be for this task. From the first row of Table 2, we notice that RoTG achieves reasonable Macro F1 score of 55.43%. The model performs very well on identifying LLM-generated *False* statements, but struggle with semantically similar *False* statements. We wish to understand if our task can be performed without understanding causality in the first place. To investigate this, we destroyed all causal edges in G, and retrained the model on the task. Interestingly, we find that all scores decline from the initial baseline, but not by too much. This suggests that while causal edges are still important to the task, as long as some understanding of similarity between events in a KG exists, models can still perform the task. Conversely, we wish to understand the importance of our similarity edges. When we destroyed similar-type edges, we noticed a significant increase in the accuracy for the *True* prediction (along with the fall in accuracy for *False* prediction). Without similarity edges, the model focuses only on causal edges and in return, over-weighs the probability of a causal statement. To conclude this subsection, RoTG demonstrates that we can perform the causal deductive task reasonably well by only relying on extracted and inferred causal relations from LLM. This presents us with a lower bound of what the LLM can understand. In Appendix Section C.3, we investigated RoTG's performance across different K values. We found that a concave relationship across top-K and F1 scores, but the differences are not statistically significant when comparing $K = 4,096$ to $K = 8,192$ or more.

5.2 LLM's Deductive Reasoning Performance

In this section, we directly test the LLM on the causal deductive reasoning task. Table 3 presents the main findings while the full findings are available in Appendix Table 8. Our proposed RoTG method (73.19% Macro F1 and 91.65% Micro F1) outperforms the baseline (70.36% Macro F1 and 90.30% Micro F1) and also improved the LLM's accuracy for all class labels. The improvement for Micro F1 is statistically significant with P-value < 0.10. To provide an alternative baseline, we retrieved semantically similar causal relations for every causal relation extracted or inferred in a report. We encoded arguments (Cause span and Effect span) using `sentence-transformers/all-mpnet-base-v2` then did vector embedding search using FaissSearcher (Douze et al., 2024). Similar truncation and cleaning procedures were done as per RoTG. Mistral's performance also improves when we inject these semantic causal relations, however, the improvement is slightly smaller than ours and unlike ours, is not statistically significant.

5.2.1 Which types of causal relations help?

In Hypothesis 1 of Section 2, we hypothesized that injecting causal relations outside of G_t but semantically part of x_{it} would improve prediction, or at least increase the likelihood of predicting *True*. Apart from exposing the model to semantic or RoTG relations, which both increased accuracy of *True* (46.53% (Row 1) compared to 48.72% (Row 5) and 49.49% (Row 7) in Table 4), we could also inject the inferred causal relations in the prompt. As expected, the accuracy for *True* in the baseline model increases to 55.99% (Row 3).

However, consistent with Hypothesis 2 of Section 2, accuracy for *False* falls significantly. This fall is slightly mitigated if we inject the extracted causal relations alongside the inferred causal relations (Row 4), supporting our grounding hypothesis. With either semantic or RoTG retrieved relations, injecting extracted relations have a negligible effect, suggesting when relations out of G_t are shown, hallucination is less of an issue, and grounding is unnecessary.

Overall, we find that we need to expose the LLM to relevant causal relations outside of the report's relations G_t to increase accuracy of *True* predictions (Hypothesis 1). However, if the inferred relations are included (relations partially in G_t, partially not), LLMs might take the provided causal chains to be the truth, and so grounding becomes helpful (Hypothesis 2). The best balance between the two would be to incorporate retrieved relations (relations $\notin G_t$), so that the model can better focus on learning about causality instead of being confused by the truthfulness of the given chain.

5.2.2 Does the number and quality of RoTG relations matter?

We described our post-processing steps for causal relations in Section 4.3. In Table 5, we investigate if we do not truncate to first 10 causal relations (No truncate), and if we do not post-process at all (No cleaning). In general, we did not find lower statistically significantly different scores. For the RoTG relations only prompt, the LLM performed best with truncation and de-duplication. For the RoTG and extracted relations prompt, the LLM performed best if we do not clean the RoTG relations. This again suggests that ensuring more retrieved relations outside of C_t, as opposed to re-exposing the model to relations from C_t, are more helpful.

5.2.3 Investigating the generation probability

We investigated the generation probabilities of the model by tracking the logits of the "True" and "False" token at the first utterance of the "True" / "False" token. We comparing the model with and without our RoTG relations, and notice that for the 1446 examples where both models correctly predicted *False*, our RoTG model returned an average *False* probability of 3.39%, while the baseline model had a probability of 2.07%. Meanwhile, for the 69 examples where both models correctly predicted *True*, our RoTG model returned an average *True* probability of 47.02%, while the baseline model had a probability of 35.60%. There are two interesting findings from here: (1) Apart from returning a higher F1, incorporating RoTG-relations helps the model become more confident in its predictions for the overlapping correct examples. (2) On average, we found that it takes the model a much higher probability to generate the *True* token than it takes for it to generate the *False* token. When models generate *True*, the next most likely word is almost always *False*. Meanwhile, for *False* predictions, the probabilities are small and more spread across all possible tokens in the models' dictionary. More investigation is needed to explain

S/N	Relations			Macro F1	Micro F1	Accuracy		
	Extract	Infer	Retrieved			*True*	*False* (Rules)	*False* (LLM)
1			*None*	70.36 (7.07)	90.30 (1.78)	46.53 (13.21)	92.23 (3.66)	95.69 (1.86)
2	✓		*None*	72.42 (7.19)	90.59 (2.52)	52.62 (13.79)	91.73 (4.22)	95.60 (2.06)
3		✓	*None*	63.97 (4.87)***	83.15 (2.85)***	**55.99 (11.38)***	78.56 (4.79)***	89.03 (4.35)***
4	✓	✓	*None*	63.66 (5.31)***	84.10 (2.53)***	50.36 (12.18)	80.12 (4.66)***	90.65 (3.38)***
5			Semantic	72.50 (6.37)	91.24 (1.40)	48.72 (11.04)	92.99 (2.48)	96.54 (1.93)
6	✓		Semantic	70.97 (4.69)	90.67 (2.11)	45.54 (7.10)	91.70 (4.21)	96.91 (1.89)
7	✓	✓	Semantic	64.48 (6.02)***	86.83 (2.27)***	41.81 (12.63)	86.19 (4.56)***	93.59 (2.44)***
8			RoTG	**73.19 (7.01)**	**91.65 (1.42)**	49.49 (13.47)	**94.31 (3.49)**	96.37 (1.37)
9	✓		RoTG	71.15 (6.40)	91.09 (2.14)	44.07 (10.02)	93.43 (3.89)	**97.02 (1.63)**
10	✓	✓	RoTG	64.21 (7.89)***	87.28 (3.23)***	37.98 (13.90)**	87.21 (4.02)***	94.46 (2.79)**

Table 4: Mistral Instruct with various relations included into prompt. Highest score per column is in bold. P-values against scores from the first row per line-separated section is indicated by: $* < 0.15$, $** < 0.10$, $*** < 0.05$.

Retrieved Processing	Relations Extracted	Macro F1	Micro F1	Accuracy		
				True	*False* (Rules)	*False*
		73.19 (7.01)	**91.65 (1.42)**	**49.49 (13.47)**	**94.31 (3.49)**	96.37 (1.37)
No truncate		72.92 (6.43)	91.60 (1.11)	48.87 (12.59)	93.75 (3.24)	96.66 (1.04)
No cleaning		71.93 (5.57)	91.19 (1.37)	46.53 (8.61)	94.01 (3.72)	96.38 (1.03)
	✓	71.15 (6.40)	91.09 (2.14)	44.07 (10.02)	93.43 (3.89)	97.02 (1.63)
No truncate	✓	70.96 (6.69)	90.95 (2.07)	44.50 (11.16)	93.43 (3.89)	96.73 (1.70)
No cleaning	✓	71.52 (5.94)	91.12 (2.16)	45.04 (9.33)	93.28 (4.17)	**97.13 (1.38)**

Table 5: Mistral Instruct with RoTG retrieval-augmented relations post-processed using three strategies: (1) With truncation (first 10) and de-duplication, (2) Without truncation but with de-duplication, (3) Without truncation and without de-duplication. Highest score per column is in bold.

why this is the case.

6 Related Work

Our dataset and task is most relevant to the deductive reasoning NLP literature, like efforts by RuleTaker (Clark et al., 2020) and ProofWriter (Tafjord et al., 2021). Different from them, our dataset is a real-world deductive reasoning task about accident investigations, and dive deep into the causal aspect. Huang and Chang (2023); Valmeekam et al. (2022) stated that current reasoning benchmarks are not meaningfully applied in the real-world. Thus, we hope that our dataset and work allievates this gap in the literature.

Our methodology is relevant to literature on RAG for LLMs (Gao et al., 2024). However, due to the nature of causal relations in our task, we focus on retrieval techniques over a graph. Thus, we were also inspired by prior research on retrieval on KGs (Liu et al., 2018; Reinanda et al., 2020) and on node classification (Shi et al., 2021; Xiao et al., 2022). Since encoding graph structured data for LLMs is also an ongoing research (Fatemi et al., 2023; Perozzi et al., 2024), more investigations on how to best present the causal chains in the prompts are needed. Different from previous works, we investigate how to leverage on knowledge already present in the dataset (extract) and within the LLMs (infer) to improve performance, instead of relying on external databases that many RAG methodologies focus on.

7 Conclusion

Our study addresses the challenging task of causal deductive reasoning, particularly within the context of real-world Accident Investigation reports. Firstly, we introduced a framework that constructs a causal KG based on what LLMs' can extract and infer. Secondly, we proposed RoTG, trained to select relevant nodes, utilizing deductive reasoning labels as an auxiliary task. Our experiments demonstrate that incorporating RoTG relations into the prompt enhances the performance of LLMs (from 70.36% (90.30%) to 73.19% (91.65%) Macro (Micro) F1), highlighting the effectiveness of integrating graph-based retrieved relations in improving LLMs' causal deductive reasoning abilities. Lastly, our dataset will be released and will be a valuable resource for researchers. Overall, our study advances the understanding and application of deductive reasoning tasks in NLP, specifically in the domain of KG-based RAG for LLMs.

8 Limitations & Ethics Statement

Our investigations are confined to findings derived from Mistral-Instruct, as such, the generalizability of our results to other LLMs may be limited. Future research should aim to explore a broader range of LLM architectures to gain a more comprehensive understanding of the phenomena under investigation. All datasets are attributed to the National Transportation Safety Board (NTSB), "Courtesy: National Transportation Safety Board."

References

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as Soft Reasoners over Language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 3882–3890. https://doi.org/10.24963/IJCAI.2020/537

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. (2024). arXiv:2401.08281 [cs.LG]

JSBT Evans. 2005. Deductive reasoning. *The Cambridge handbook of thinking and reasoning* (2005), 169–184.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2023. Talk like a Graph: Encoding Graphs for Large Language Models. arXiv:2310.04560 [cs.LG]

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL]

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards Reasoning in Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1049–1065. https://doi.org/10.18653/v1/2023.findings-acl.67

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR* abs/2310.06825 (2023). https://doi.org/10.48550/ARXIV.2310.06825 arXiv:2310.06825

Phil Johnson-Laird. 2010. Deductive reasoning. *Wiley Interdisciplinary Reviews: Cognitive Science* 1, 1 (2010), 8–17.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou. 2023. Better Zero-Shot Reasoning with Role-Play Prompting. *CoRR* abs/2308.07702 (2023). https://doi.org/10.48550/ARXIV.2308.07702 arXiv:2308.07702

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Yihao Li, Ru Zhang, Jianyi Liu, and Gongshen Liu. 2024. An Enhanced Prompt-Based LLM Reasoning Scheme via Knowledge Graph-Integrated Collaboration. arXiv:2402.04978 [cs.CL]

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 2395–2405. https://doi.org/10.18653/V1/P18-1223

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations:

What Makes In-Context Learning Work?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 11048–11064. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.759

Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. 2024. Let Your Graph Do the Talking: Encoding Structured Data for LLMs. arXiv:2402.05862 [cs.LG]

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning causality for news events prediction. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, Alain Mille, Fabien Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab (Eds.). ACM, 909–918. https://doi.org/10.1145/2187836.2187958

Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2020. Knowledge Graphs: An Information Retrieval Perspective. *Found. Trends Inf. Retr.* 14, 4 (2020), 289–444. https://doi.org/10.1561/1500000063

Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama Japan, May 8-13, 2021, Extended Abstracts*, Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, and Takeo Igarashi (Eds.). ACM, 314:1–314:7. https://doi.org/10.1145/3411763.3451760

Lance J Rips. 1994. *The psychology of proof: Deductive reasoning in human thinking*. Mit Press.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 9248–9274. https://aclanthology.org/2023.findings-emnlp.620

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 1548–1554. https://doi.org/10.24963/IJCAI.2021/214

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 3621–3634. https://doi.org/10.18653/V1/2021.FINDINGS-ACL.317

Fiona Anting Tan, Debdeep Paul, Sahim Yamaura, Miura Koji, and See-Kiong Ng. 2023. Constructing and Interpreting Causal Knowledge Graphs from News. *CoRR* abs/2305.09359 (2023). https://doi.org/10.48550/ARXIV.2305.09359 arXiv:2305.09359

Karthik Valmeekam, Alberto Olmo Hernandez, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change). *CoRR* abs/2206.10498 (2022). https://doi.org/10.48550/ARXIV.2206.10498 arXiv:2206.10498

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change. arXiv:2206.10498 [cs.CL]

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhu Chen, Jie Fu, and Junran Peng. 2023. RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models. *CoRR* abs/2310.00746 (2023). https://doi.org/10.48550/ARXIV.2310.00746 arXiv:2310.00746

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.* 2022 (2022). https://openreview.net/forum?id=yzkSU5zdwD

Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. 2022. Graph neural networks in node classification: survey and evaluation. *Mach. Vis. Appl.* 33, 1 (2022), 4. https://doi.org/10.1007/S00138-021-01251-0

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=RdJVFCHjUMI

A Experimental Details

Claude 2.1 inference

- Model = `anthropic.claude-v2:1`
- Max tokens to sample = 1000 for extracting causes as a list, 1800 for generating *False* statements
- Temperature = 0.5

RoTG training

- Encoder = `roberta-base`
- Local graph node dim = 770
- Global graph node dim = 768
- Num layers in GNN = 4
- Top-K = 4096
- Dropout = 0.1
- Post-GNN to Auxiliary Clf Layers:
  - Linear1 Out Dim = 128
  - Linear2 Out Dim = 64
  - Linear3 Out Dim = 2
- CrossEntropyLoss with class weights, reduction='mean'
- Top-E = 3

Mistral-Instruct inference

- Model = `Mistral-7B-Instruct-v0.1`
- Max new tokens = 1500
- Temperature = 0.5

Fold#	#Statements	*#True*	*#False*
1	159	10	149
2	169	15	154
3	191	14	177
4	179	15	164
5	185	18	167
6	169	11	158
7	151	16	135
8	138	16	122
9	168	26	142
10	168	14	154

Table 6: Count of examples per fold by class labels.

B Dataset & Task Creation

B.1 Prompts

Prompt 1: Prompt for extracting probable causes into a list.

```
##### INSTRUCTIONS #####

Please help to extract the key Causes
into point forms based on a paragraph
bounded by [START_CONTEXT] and
[END_CONTEXT].
Do not add any explanations, or leading
or trailing descriptions. Add as many
bullet points as needed to exhaustively
extract all stated Causes.

##### EXAMPLE #####

[START_CONTEXT]
The probable cause of the employee
fatality at the Dyno Nobel facility was
a result of the conductor being
impacted by the moving railcars during
a shoving movement while located in an
area with insufficient walking space
available for the employee to perform
trackside duties.
[END_CONTEXT]

Expected Output:
[START_CAUSES]
- Conductor impacted by the moving
railcars during a shoving movement
- Accident was located in area with
insufficient walking space available
for the employee to perform trackside
duties
```

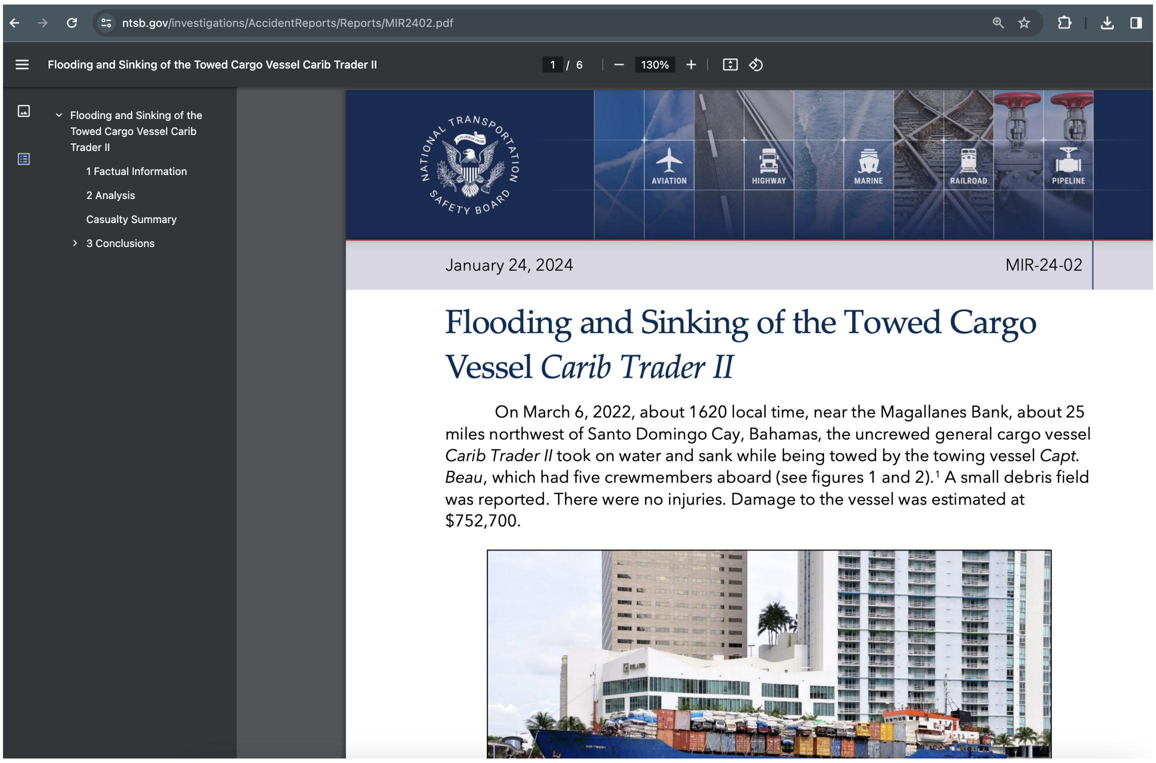

Figure 3: First page of an NTSB report in PDF.

```
[END_CAUSES]

##### TASK #####
```

Prompt 2: Prompt for generating negative causal examples.

```
Based on the following accident
investigation bounded by <CONTEXT>
delimiters, the true probable cause(s)
are provided within <CAUSES> delimiters.
Given these information, provide a list
of 10 possible causes or contributing
causes investigated within the context
that is not stated as a final true
probable cause.
Your output should only contain a list
of 10 enumerated statements/sentences
with no explanation.

<CAUSES>
{causes}
</CAUSES>

<CONTEXT>
{context}
</CONTEXT>
```

C Mining Causal Knowledge in LLMs

Figure 4 provides a detailed outline of our proposed methodology, corresponding to the descriptions in Section 4.

C.1 TransformerGCN architecture

We introduced the overall structure of our RoTG model in Section 4.2. This section outlines the detailed model architecture for TransformerGCN (Shi et al., 2021).

Our initial node features are represented by Q, an attended representation of Q'. Q' is a concatenation of the RoBERTa-encoded embeddings for each node description s and the two one-hot vectors $(oh_{\text{extr}}, oh_{\text{inf}})$ indicating if the node is extracted or inferred to the target statement s_i or not. The attention mechanism then computes the attention weights between the node features Q' and the target statement embedding r_i to generate the cross-attended node feature matrix Q.

$$r_i = \text{RoBERTa}(s_i) \tag{13}$$

$$R = \text{RoBERTa}(S) \tag{14}$$

$$Q' = [R, oh_{\text{extr}}, oh_{\text{inf}}] \tag{15}$$

$$Q = \text{Attention}(Q', r_i, r_i) \tag{16}$$

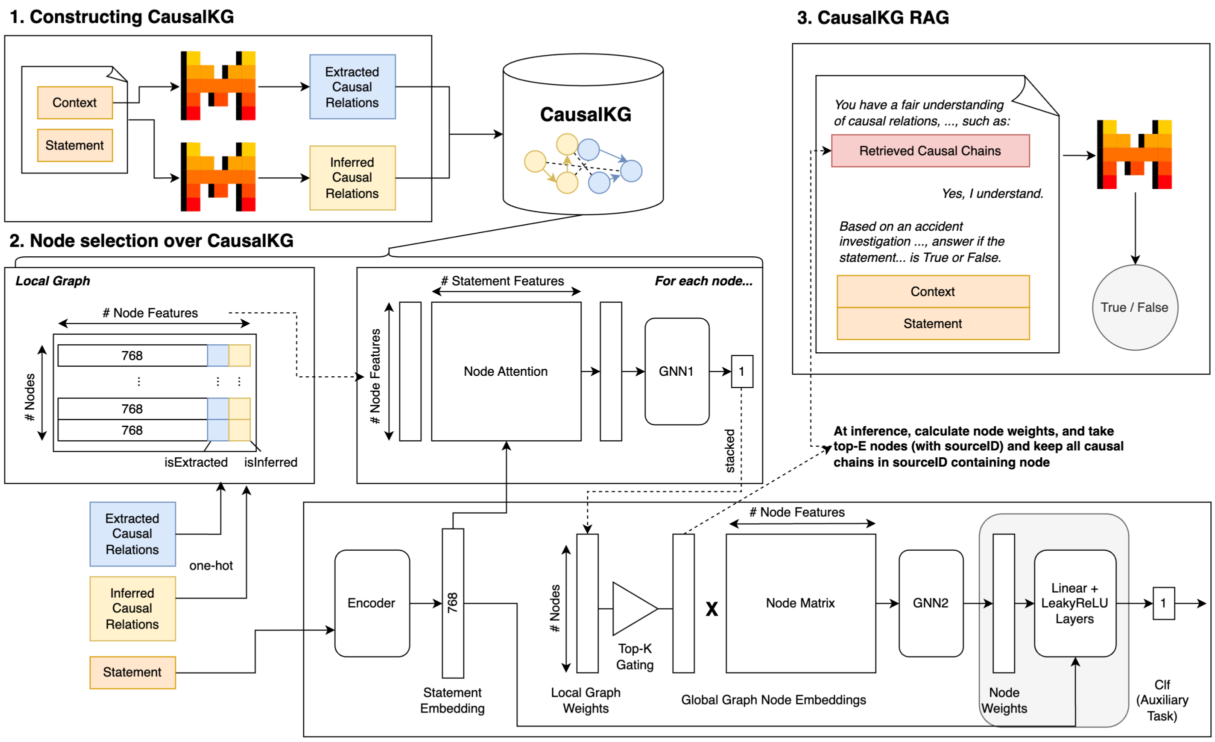

Figure 4: Detailed outline of our proposed methodology.

K Value	Macro F1	Micro F1	Accuracy		
			True	*False* (Rules)	*False* (LLM)
2048	54.12 (6.55)	79.99 (9.80)*	**34.46 (29.03)**	57.20 (35.58)*	97.78 (4.44)
4096	55.43 (6.09)	83.96 (9.07)	31.01 (31.19)	67.44 (34.41)	99.45 (0.86)
8192	**56.06 (6.53)**	**86.17 (6.09)**	24.10 (20.63)	**77.03 (21.26)**	**99.82 (0.38)**
All ∽16K	53.98 (5.79)	83.75 (10.40)	28.27 (32.49)	68.04 (37.25)	99.65 (0.84)

Table 7: Mean (Std) F1 and Accuracy across different K values for Top-K Gating. Highest score per column is in bold. P-values against K=8192 scores indicated by: $* < 0.15$.

Our graph G is equivalently represented by the adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$. The diagonal degree matrix is denoted by $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, where $d_i = \sum_j a_{ij}$ is the degree of node i. A normalized adjacency matrix is defined as $D^{-1}A$ or $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$.

A typical GCN transforms and propagates node features across the graph by several layers to build the approximation of the mapping of input to output. In other words, the feature propagation scheme of GCN in layer l is:

$$H^{(l+1)} = \sigma\left(D^{-1}AH^{(l)}W^{(l)}\right) \tag{17}$$

$$Y = f_{\text{out}}(H^{(L)}) \tag{18}$$

where σ is an activation function, $W^{(l)}$ is the trainable weight in the l-th layer, and $H^{(l)}$ is the l-th layer representations of nodes. $H^{(0)}$ is equal to node input features Q. Finally, an f_{out} output linear layer is applied on the final representation to make predictions Y for each node.

However, since our graph is heterogenous, we require message passing across edge features too. Therefore, TGCN helps by incorporating edge features into the multi-head attention for graph learning. Given node features $H^{(l)} = \{h_1^{(l)}, h_2^{(l)}, \ldots, h_n^{(l)}\}$, multi-head attention for each edge j to i is computed as follows:

$$q_{c,i}^{(l)} = W_{c,q}^{(l)} h_i^{(l)} + b_{c,q}^{(l)} \tag{19}$$

$$k_{c,j}^{(l)} = W_{c,k}^{(l)} h_j^{(l)} + b_{c,k}^{(l)} \tag{20}$$

$$e_{c,ij} = W_{c,e} e_{ij} + b_{c,e} \tag{21}$$

$$\alpha_{c,ij}^{(l)} = \frac{\exp(q_{c,i}^{(l)} \cdot k_{c,j}^{(l)} + e_{c,ij})}{\sum_{u \in N(i)} \exp(q_{c,i}^{(l)} \cdot k_{c,u}^{(l)} + e_{c,iu})} \tag{22}$$

where $h_{q,k}^{(l)} = \exp\left(\frac{q_{c,i}^{(l)} \cdot k_{c,j}^{(l)}}{\sqrt{d}}\right)$ is the exponential scale dot-product function and d is the hidden size of each head. For the c-th head attention, we transform the source feature $h_i^{(l)}$ and distant feature $h_j^{(l)}$ into query vector $q_{c,i}^{(l)} \in \mathbb{R}^d$ and key vector $k_{c,j}^{(l)} \in \mathbb{R}^d$ respectively using different trainable parameters $W_{c,q}^{(l)}$, $W_{c,k}^{(l)}$, $b_{c,q}^{(l)}$, $b_{c,k}^{(l)}$. The provided edge features e_{ij} are encoded and added into the key vector as additional information for each layer.

After obtaining the graph multi-head attention, message passing and aggregation from the distant j to the source i is computed by:

$$v_{c,j}^{(l)} = W_{c,v}^{(l)} h_j^{(l)} + b_{c,v}^{(l)} \tag{23}$$

$$\hat{h}_i^{(l+1)} = \sum_{j \in N(i)} \alpha_{c,ij}^{(l)} (v_{c,j}^{(l)} + e_{c,ij}) \tag{24}$$

where k is the concatenation operation for C head attention. This multi-head attention matrix replaces the original normalized adjacency matrix in Equation 17 as the transition matrix for message passing.

Finally, we apply a linear transformation to the last layer of node features $h_i^{(l)}$, obtaining a representation of local node weights (ow_i), trained to represent how important this node is to the downstream task.

$$ow_i = W_{c,v}^{(l)} h_i^{(l)} + b_{c,v}^{(l)} \tag{25}$$

C.2 Prompts

Prompt 3: Prompt for extracting causal relations

```
Extract all the causal events in this
report:
{context}

Format the extracted Cause and Effect
events into a list, like:
1. Engineer’s inattentiveness to signal
indications --> Engineer failed to
operate train in accordance with signal
indications and speed restriction -->
Train collided with another train
2. Lack of positive train control
system --> Train A not stopped before
red signal --> Train A passed red
signal --> Collision between Train A
and Train B
...
where "-->" represents "causes", so
"Cause Event --> Effect Event".

Answer:
```

Prompt 4: Prompt for inferring causal relations

```
Based on your knowledge, suggest the
series of Cause and Effect events that
explain how the cause within the
STATEMENT could have led to the
accident in the CONTEXT.

<STATEMENT>
{statement}
</STATEMENT>
<CONTEXT>
{context}
</CONTEXT>

Format the suggested Cause and Effect
events into a list, like:
- Engineer’s inattentiveness to signal
indications --> Engineer failed to
operate train in accordance with signal
indications and speed restriction -->
Train collided with another train
(Accident)
where "-->" represents "causes", so
"Cause Event --> Effect Event".

Answer:
```

Prompt 5: Prompt V1 for causal deductive reasoning task.

```
Based on an accident investigation
bounded by <CONTEXT> delimiters, answer
if the statement within <STATEMENT>
delimiters about the probable cause(s)
of the accident is True or False. Your
answer must be based on the
investigation facts and details within
<CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT>

Answer:
```

Prompt 6: Prompt V2 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where ”-->”
represents ”causes”.
[/INST] Yes, I understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 7: Prompt V3 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where ”-->”
represents ”causes”.
[/INST] Yes, I understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

<RELATIONS>
Relations extracted from <CONTEXT>:
{extracted}
</RELATIONS>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 8: Prompt V4 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where ”-->”
represents ”causes”.
[/INST] Yes, I understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

<RELATIONS>
Possible relations linking probable
cause in <STATEMENT> to accident:
{inferred}
</RELATIONS>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 9: Prompt V5 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where ”-->”
represents ”causes”.
[/INST] Yes, I understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.
```

```
<CONTEXT>
{context}
</CONTEXT>

<RELATIONS>
Relations extracted from <CONTEXT>:
{extracted}

Possible relations linking probable
cause in <STATEMENT> to accident:
{inferred}
</RELATIONS>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 10: Prompt V6 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where "-->"
represents "causes", such as:
{retrieved} [/INST] Yes, I
understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 11: Prompt V7 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where "-->"
represents "causes", such as:
{retrieved} [/INST] Yes, I
understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

<RELATIONS>
Relations extracted from <CONTEXT>:
{extracted}
</RELATIONS>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 12: Prompt V8 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where "-->"
represents "causes", such as:
{retrieved} [/INST] Yes, I
understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

<RELATIONS>
Relations extracted from <CONTEXT>:
{extracted}
```

```
Possible relations linking probable
cause in <STATEMENT> to accident:
{inferred}
</RELATIONS>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

Prompt 13: Prompt V9 for causal deductive reasoning task.

```
<s>[INST] You have a fair understanding
of causal relations, where "-->"
represents "causes", such as:
<RELATIONS>
Historical relations:
{retrieved}

Relations extracted from <CONTEXT>:
{extracted}

Possible relations linking probable
cause in <STATEMENT> to accident:
{inferred}
</RELATIONS> [/INST] Yes, I
understand.</s>
[INST] Based on an accident
investigation bounded by <CONTEXT>
delimiters, answer if the statement
within <STATEMENT> delimiters about the
probable cause(s) of the accident is
True or False. Your answer must be
based on the investigation facts and
details within <CONTEXT>.

<CONTEXT>
{context}
</CONTEXT>

Is this statement True or False?
<STATEMENT>
{statement}
</STATEMENT> [/INST]

Answer:
```

C.3 RoTG Findings

Our RoTG model includes a gating framework to focus on top-K nodes. Table 7 presents scores from RoTG across different K values. In terms of Macro and Micro F1, K=8192 returns the best performance. We notice a slight concave pattern of F1 against K values, suggesting an optimal amount of gating is needed. However, the findings did not show statistically significant differences across K=4096 to when all nodes were allowed to be differentiated against.

C.4 LLM Findings

Findings from all experiments with Mistral-Instruct are available in Table 8. The first column indicates the corresponding Prompt number used, while the next four columns indicate the additional information included in the prompt, or if any different processing method was used.

C.5 Qualitative Examples

Table 9 shows the output response from Mistral-Instruct across the three main prompt versions, corresponding to Table 3. The last two columns details the retrieved relations that were included in the prompt.

Prompt #	Relations			Other Tweaks	Macro F1	Micro F1	Accuracy		
	Extract	Infer	Retrieved				*True*	*False* (Rules)	*False* (LLM)
5			*None*		70.36 (7.07)	90.30 (1.78)	46.53 (13.21)	92.23 (3.66)	95.69 (1.86)
6			*None*	Role-play	71.04 (5.99)	89.64 (0.87)	53.82 (12.26)	91.12 (2.76)	94.00 (1.63)***
7	✓		*None*		72.42 (7.19)	90.59 (2.52)	52.62 (13.79)	91.73 (4.22)	95.60 (2.06)
8		✓	*None*		63.97 (4.87)***	83.15 (2.85)***	**55.99 (11.38)***	78.56 (4.79)***	89.03 (4.35)***
9	✓	✓	*None*		63.66 (5.31)***	84.10 (2.53)***	50.36 (12.18)	80.12 (4.66)***	90.65 (3.38)***
10			Semantic		72.50 (6.37)	91.24 (1.40)	48.72 (11.04)	92.99 (2.48)	96.54 (1.93)
11	✓		Semantic		70.97 (4.69)	90.67 (2.11)	45.54 (7.10)	91.70 (4.21)	96.91 (1.89)
12	✓	✓	Semantic		64.48 (6.02)**	86.83 (2.27)***	41.81 (12.63)	86.19 (4.56)***	93.59 (2.44)***
10			RoTG		**73.19 (7.01)**	**91.65 (1.42)****	49.49 (13.47)	**94.31 (3.49)**	96.37 (1.37)
11	✓		RoTG		71.15 (6.40)	91.09 (2.14)	44.07 (10.02)	93.43 (3.89)	97.02 (1.63)*
12	✓	✓	RoTG		64.21 (7.89)**	87.28 (3.23)***	37.98 (13.90)	87.21 (4.02)***	94.46 (2.79)
10			RoTG	No cleaning	71.93 (5.57)	91.19 (1.37)	46.53 (8.61)	94.01 (3.72)	96.38 (1.03)
10			RoTG	No truncate	72.92 (6.43)	91.60 (1.11)	48.87 (12.59)	93.75 (3.24)	96.66 (1.04)
11	✓		RoTG	No cleaning	71.52 (5.94)	91.12 (2.16)	45.04 (9.33)	93.28 (4.17)	**97.13 (1.38)****
11	✓		RoTG	No truncate	70.96 (6.69)	90.95 (2.07)	44.50 (11.16)	93.43 (3.89)	96.73 (1.70)
13	✓	✓	RoTG	Rels before Task	64.55 (6.48)**	86.58 (1.80)***	42.85 (11.72)	87.09 (2.66)***	92.74 (2.00)***

Table 8: Mistral Instruct with various relations included into prompt. Highest score per column is in bold. P-values against scores from the *None* scores in the first row is indicated by: $*< 0.15$, $**< 0.10$, $***< 0.05$.

Statement	Label	Base	Semantic	RoTG	Semantic_Rels	RoTG_Rels
The probable cause of the accident was insufficient communication between the captains after the passing arrangement was changed...	False (True statement: "The probable cause of the accident was the impact of distraction upon the decision-making and recollection of the captain of the P. B. Shah.")	Answer: The statement is True. Explanation: The accident occurred due to a miscommunication between the captains of the two vessels regarding the passing arrangement. The P. B. Shah captain initially proposed a starboard-to-starboard passing arrangement, but the Dewey R captain misunderstood and believed it was a port-to-port passing arrangement. The Dewey R captain then steered his vessel towards the port side of the river in an attempt to conduct a two-whistle pass, which resulted in a collision with the P. B. Shah tow. The miscommunication and confusion about the passing arrangement led to the accident.	Answer: False. The probable cause of the accident was the P. B. Shah captain's error in initiating a port-to-port (one whistle) passing on the radio with the Dewey R captain, who had meant to arrange a starboard-to-starboard (two-whistle) passing. The P. B. Shah captain was distracted by the many tasksassociated with preparing for his arrival at the Ingram facility, including having a cell phone conversation with the boat store and meeting with the mate to discuss upcoming tasks. The Dewey R captain made a mistake by not following the inland navigation rules and should have assessed the risk of collision presented by the two alternative meeting arrangements, chosen the option that did not include passing in front of the P. B. Shah tow, and then used his privilege as the downbound vessel to overrule the P. B. Shah captain's change to a two-whistle meeting arrangement.	Answer: False. Explanation: The probable cause of the accident was not insufficient communication between the captains after the passing arrangement was changed. The investigation found that the P. B. Shah captain misunderstood the initial passing agreement and believed he was proposing a starboard-to-starboard passage, while the Dewey R captain believed he was proposing a port-to-port passage. The miscommunication was due to the P. B. Shah captain being distracted by tasksassociated with preparing for his arrival at the Ingram facility, and the Dewey R captain doubting his understanding of the original agreement. The investigation also found that the Dewey R captain made a mistake by not following the inland navigation rules and should have assessed the risk of collision presented by the two alternative meeting arrangements and chosen the option that did not include passing in front of the P. B. Shah tow.	- BART's simple approval process allowed access along the right-of-way without protection from moving trains –> Safety issues and concerns identified during the NTSB accident investigation - Workers were not able to properly protect themselves from moving trains –> Accident occurred. - - NTSB accident investigation –> Major findings identified safety issues and concerns - Identification of safety issues and concerns –> New safety regulations and requirements - - BART assistant chief transportation officer distributed a memorandum immediately prohibiting simple approvals –> Access to the right-of-way must provide protection from moving trains - Workers were not aware of safety protocols and procedures –> Workers were not able to properly protect themselves from moving trains - Trains exceeding speed limits were not uncommon, leading to increased risk of accidents –> Accident occurred - - Major findings from investigations into worker fatalities –> Identification of safety issues and concerns - BART's simple approval process allowed access along the right-of-way that did not provide workers with protection from moving trains –> BART issued General Order 175 (GO 175) to govern roadway worker protection for rail transit workers in California, prohibiting the type of access that was allowed under BART's simple approval process. - - BART assistant chief transportation officer distributed a memorandum immediately prohibiting simple approvals –> Access to the right-of-way must provide the work crew with protection from moving trains	- Defective control system in unoccupied cars –> Errant control signals sent to power systems - Errant control signals sent to power systems –> Unoccupied cars moved and stopped - Based on evidence and statements regarding the float driver's medical history, sleep opportunity, sleep quality, sleep schedule, and time awake, it is unlikely that he was fatigued at the time of the collision –> Based on evidence and statements regarding the float driver's medical history, sleep opportunity, sleep quality, sleep schedule, and time awake, it is unlikely that he was fatigued at the time of the collision. - Lack of fatigue –> Based on evidence and statements regarding the float driver's medical history, sleep opportunity, sleep quality, sleep schedule, and time awake, it is unlikely that he was fatigued at the time of the collision. - There is no evidence that the float driver was experiencing stress or had a health, hearing, or visual condition that affected his ability to perceive the grade crossing warnings, perceive the train, or safely operate his vehicle –> Based on evidence and statements regarding the float driver's medical history, sleep opportunity, sleep quality, sleep schedule, and time awake, it is unlikely that he was fatigued at the time of the collision. - Based on evidence and statements regarding the float driver's medical history, sleep opportunity, sleep quality, sleep schedule, and time awake, it is unlikely that he was fatigued at the time of the collision. –> The grade crossing warning system provided 20 seconds of warning as required by federal regulations. - The float driver was not distracted by the use of in-vehicle electronic devices –> The float driver was not distracted by electronic devices, which could have contributed to the accident.
The probable cause of the accident was current conditions contributing to intermittent submersion of wastage holes..	False (True statement: "The probable cause of the accident was the company's lack of effective hull inspection and maintenance..")	Answer: True. The investigation found that the accident was caused by the intermittent submersion of wastage holes on the starboard stern quarter of the Miss Roslyn due to the captain pushing against the tow at a 900̆0b0 angle to the bank for 2.5 to 3 hours. The current likely lowered and submerged the holes, allowing continuous water ingress to the starboard steering void for about 3 hours. Once the hull flooded, the vessel sank lower, increasing the rate of flooding through the holes to the starboard steering void, resulting in the observed starboard list. The port flanking void eventually flooded, causing the vessel's loss of stability and capsizing.	Answer: The statement is True. The investigation found that the accident was caused by the intermittent submersion of wastage holes on the starboard stern quarter of the Miss Roslyn due to the captain pushing against the tow at a 90̆0b0 angle to the bank for 2.5 to 3 hours. The current likely lowered and submerged the holes, allowing continuous water ingress to the starboard steering void for about 3 hours. Once the hull flooded, the vessel sank lower, increasing the rate of flooding through the holes to the starboard steering void, resulting in the observed starboard list. The port flanking void eventually flooded, causing the vessel to lose stability and capsize.	Answer: False. The probable cause of the accident was not the current conditions contributing to intermittent submersion of wastage holes. The investigation found that the two wastage holes and a fractured weld at the main deck to the starboard steering void were not watertight, and the captain was assisting in making up a tow, which required him to push against the tow at a 90̆0b0 angle to the bank for 2.5 to 3 hours. This action likely lowered and intermittently submerged the wastage and fracture holes on the starboard stern quarter, allowing continuous water ingress to the starboard steering void for about 3 hours. The aft steering void was partially flooded, which likely caused the observed starboard list. Once the hull flooded, the vessel sank lower, increasing the rate of flooding through the holes to the starboard steering void, thus increasing the starboard heel and submerging the starboard bulwark and stern deck. The port flanking void flooded, resulting in the vessel's loss of stability and capsizing. The severe wastage found throughout the vessel in the post-accident survey indicates that the Miss Roslyn was poorly maintained. The company did not have an effective maintenance program on board the vessel, so holes in and fractures to the hull and deck went undetected and ultimately led to flooding while the vessel was conducting normal operations.	- Pedestrian's decision to run across the multilane roadway in front of the oncoming car –> Driver's decision to make a left turn from the left-turn lane onto eastbound Leesburg Pike - Driver failed to see pedestrian –> Driver applied brakes and attempted to steer left, colliding with pedestrian (Accident)	- Defective control system in unoccupied cars –> Errant control signals sent to power systems - Errant control signals sent to power systems –> Unoccupied cars moved and stopped - Conductor was pinched between the knuckle of the stationary car and the drawbar carrier of the free-rolling car –> Fatal accident - By-passed couplers on the 17th and 18th cars –> Fatal accident - Train movement before going between cars to perform work on cars –> Fatal accident - Violation of these rules escalated the discipline policy by one step –> Fatal accident - Death of the pedestrian –> Fatal accident - Fatal accident –> Accident Number: HWY16SH023, Accident Type: Fatal pedestrian collision with car, Location: 9th Street and P Street NW, Washington, DC, Date and Time: August 18, 2016, about 2:20 a.m. eastern daylight time, Vehicle: 2000 Mercedes-Benz CLK 320, Driver: 31-year-old female, Pedestrian: 44-year-old male, Fatalities: 1 - Coding error in the software upgrade –> Acceleration and deceleration of the train - Acceleration and deceleration of the train –> Injury of passengers

Table 9: Qualitative examples from Mistral-Instruct across the three different prompts (Base (No additional relations in prompt), Semantic, and RoTG causal relations in prompt).