SERE: Structural Example Retrieval for Enhancing LLMs in Event Causality Identification

Zhifeng Hao[1,2], Zhongjie Chen[1], Junhao Lu[1], Shengyin Yu[1]
Guimin Hu[1], Keli Zhang[3], Ruichu Cai[1,4], Boyan Xu[1]*
[1]School of Computer Science, Guangdong University of Technology
[2]College of Mathematics and Computer, Shantou University
[3]Huawei Noah's Ark Lab [4]Peng Cheng Laboratory
haozhifeng@stu.edu.cn {bariancgg, lujunhao.code, yushengyin2022}@gmail.com
rice.hu.x@gmail.com zhangkeli1@huawei.com {cairuichu, hpakyim}@gmail.com

Abstract

Event Causality Identification (ECI) requires models to determine whether a given pair of events in a context exhibits a causal relationship. While Large Language Models (LLMs) have demonstrated strong performance across various NLP tasks, their effectiveness in ECI remains limited due to biases in causal reasoning, often leading to overprediction of causal relationships (causal hallucination). To mitigate these issues and enhance LLM performance in ECI, we propose **SERE**, a structural example retrieval framework that leverages LLMs' few-shot learning capabilities. **SERE** introduces an innovative retrieval mechanism based on three structural concepts: (i) **Conceptual Path Metric**, which measures the conceptual relationship between events using edit distance in ConceptNet; (ii) **Syntactic Metric**, which quantifies structural similarity through tree edit distance on syntactic trees; and (iii) **Causal Pattern Filtering**, which filters examples based on predefined causal structures using LLMs. By integrating these structural retrieval strategies, **SERE** selects more relevant examples to guide LLMs in causal reasoning, mitigating bias and improving accuracy in ECI tasks. Extensive experiments on multiple ECI datasets validate the effectiveness of **SERE**.

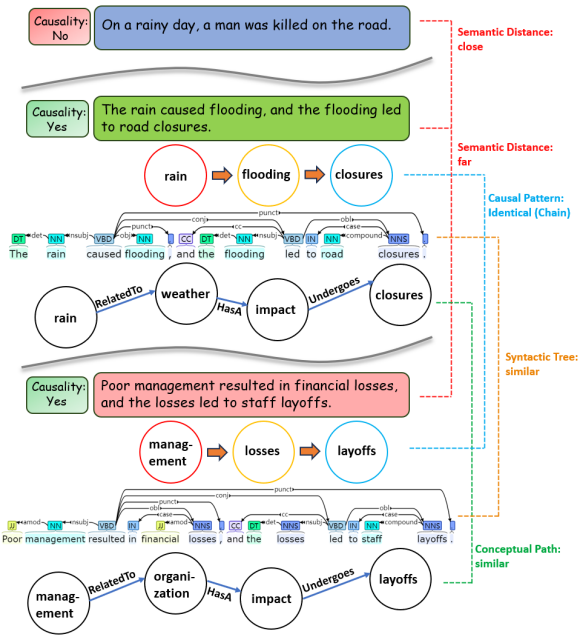


Figure 1: The middle instance is the sample to be inferred, while the two instances above and below are the corpus samples to be retrieved. The upper sample has a similar semantic similarity to the middle one but differs in causality. Selecting this sample may mislead the LLM. The lower sample has low semantic similarity to the sample to be inferred, but it is structurally similar and shares the same causality. Selecting this example helps guide the LLM to reason correctly.

1 Introduction

Event Causality Identification (ECI) requires models to determine whether a given pair of events within a context exhibits a causal relationship (Zuo et al., 2020; Dunietz et al., 2015). Traditionally, ECI methods have relied on fine-tuning pre-trained language models, primarily encoder-only architectures such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). However, Although fine-tuning enables the model to achieve higher performance on the ECI task, due to the limited size of available ECI datasets, fine-tuned models often struggle to generalize effectively. With the emergence of Large Language Models (LLMs) such as GPT (Radford, 2018), which leverage vast amounts of corpus data, it has become possible to apply these models to ECI without requiring additional training. This enables fast deployment of LLMs in ECI tasks. However, recent studies (Gao et al., 2023) have revealed that applying LLMs to ECI may introduce biases in causal reasoning, leading to a tendency to overpredict causal relationships—a phenomenon referred to as causal hallucination. This raises a fundamental challenge in adapting LLMs for accurate ECI.

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 42861–42886
July 2-7, 2026 ©2026 Association for Computational Linguistics

To mitigate bias and improve the model's ability to capture complex causal relationships, few-shot learning has emerged as a promising approach, where the effectiveness heavily depends on the example selection strategy (Liu et al., 2021; Dong et al., 2024). However, existing methods primarily rely on semantic retrieval, which presents notable limitations: In ECI, causal relationships between events are inherently structural rather than purely semantic. For instance, as illustrated in Figure 1, semantic retrieval methods may select examples based on shared words like "rain" and "road" appearing in both the first and second instances. However, these two instances exhibit opposite causality. Relying solely on semantic similarity can therefore lead to incorrect example selection, potentially causing LLMs to misidentify causal relationships.

To retrieve more structurally relevant examples and address the limitations of semantic retrieval methods, we introduce a structure-aware retrieval approach based on three key structural concepts: Conceptual Path, Syntactic Structure, and Causal Pattern.

- **Conceptual Path**: Captures the relationship between the source and target events, providing structural information through external knowledge.
- **Syntactic Structure**: Represents the syntactic properties of the context, commonly derived from dependency parsing and constituent parsing, offering insights into sentence structure.
- **Causal Pattern**: Proposed by Cai et al. (2025), it consists of predefined simple causal graphs, providing pattern-matching-based structural information.

By incorporating the aforementioned structural concepts, we can more accurately retrieve examples that share the same reasoning structure and causal pattern. As illustrated in Figure 1, while the second and third instances exhibit little semantic similarity, they share similar Conceptual Paths, syntactic structures, and identical Causal Patterns, enabling the correct retrieval of the third instance.

To better measure these structural concepts, we propose **SERE**, a **S**tructural **E**xample **R**etrieval framework designed to **E**nhance LLMs' performance on ECI tasks. Specifically, the **SERE** framework evaluates corpus samples using two key metrics: (i) Conceptual Path Metric: For each corpus sample and the target query, we extract the paths of their source event and target event from ConceptNet. The structural similarity is then measured using the *edit distance* between these paths, producing a path-based relevance score. (ii) Syntactic Metric: We extract the syntactic trees of both the corpus sample and the target query, computing the *tree edit distance* to quantify their structural alignment. The final score of each corpus sample is obtained by weighting and aggregating the scores from both metrics. Additionally, we introduce an LLM-based causal pattern extractor to identify and compare the causal patterns of both the corpus samples and the target query. Corpus samples that share the same causal pattern as the query and rank among the top-k highest scores are selected as the final examples. These retrieved examples are then incorporated into prompt design to guide the LLM in performing causal reasoning.

The main contributions of this paper are as follows:

- We propose **SERE**, a structural example retrieval framework that integrates three structural concepts—Conceptual Path, Syntactic Structure, and Causal Pattern—to enhance LLMs' ability to identify causal relationships in ECI tasks. To the best of our knowledge, we are the first to jointly integrate them for ECI task.
- We introduce two structural similarity metrics, **Conceptual Path Metric** and **Syntactic Metric**, along with a **Causal Pattern Filtering mechanism**, to effectively retrieve structurally aligned examples that guide LLM reasoning.
- We conduct extensive experiments on multiple ECI datasets, demonstrating the effectiveness, generalizability and robustness of **SERE**.

2 Preliminary Introduction for Structures

In this section, We introduce the three structural concepts used in this paper.

Conceptual Path: By connecting several concepts and the relationships between them, we can form a path between two events, known as a Conceptual Path. This path reveals the commonsense causal connections between events through a pre-constructed concept network, suitable for supplementing implicit connections between events miss-

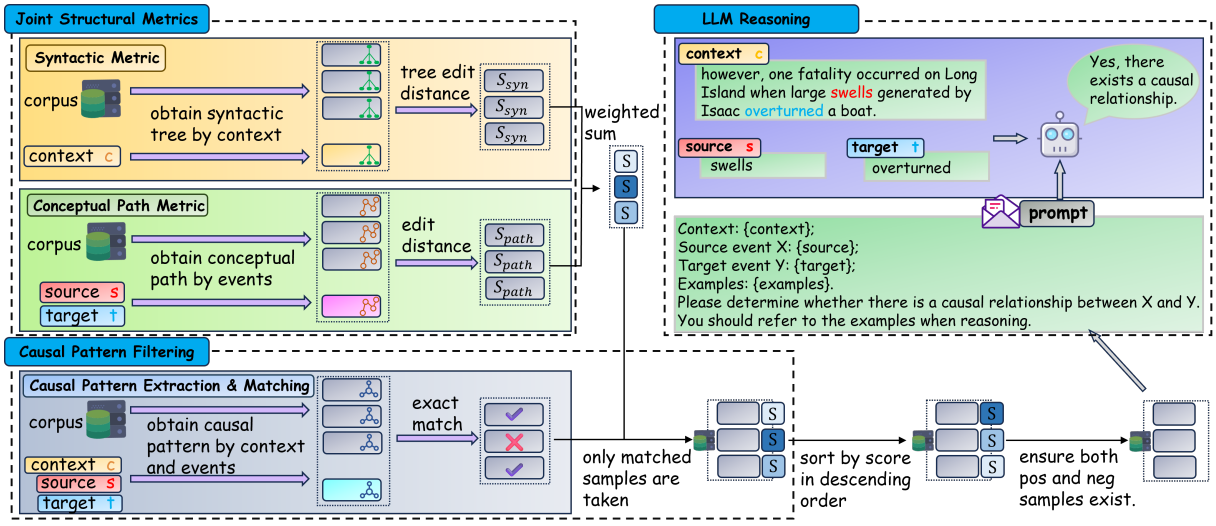


Figure 2: Overview of the SERE framework.

ing in the text. It provides supplementary information to the model from the perspective of external knowledge priors.

Syntactic Structure: This reveals the structural features of the context and provides causal clues at the syntactic level. It directly derives from the language structure of the text and can parse specific expressions, directionality, and complex syntactic patterns in the text, playing a role in causal reasoning from a syntactic prior perspective.

Causal Pattern: It provides a structured representation of causal relationships by defining several causal structures that can be described by simple directed acyclic graphs (DAGs). This assists the model in causal reasoning from the perspective of structural pattern matching.

3 Methodology

3.1 Overview

The overview of the **SERE** framework is shown in Figure 2. This framework consists of three components: *Joint Structural Metrics*, *Causal Pattern Filtering*, and *LLM Reasoning*. In the *Joint Structural Metrics* section, we compute the edit distance based score of the event pairs in ConceptNet for both the sample to be inferred and the corpus sample, as well as the tree edit distance based score of the syntactic trees of their contexts. After obtaining the two scores, we weight and sum them to generate the structure-based score for the corpus sample. In the *Causal Pattern Filtering* section, we follow prior work and design an LLM Agent to generate the corresponding causal pattern for both the sample to be inferred and the corpus sample. We then select the top-k corpus samples that have the same causal pattern as the sample to be inferred and the highest structural scores as the final retrieved examples. In the *LLM Reasoning* phase, we use the retrieved examples to design an appropriate prompt to guide the LLM in reasoning, ultimately obtaining the inference results.

3.2 Notation Definition

Let the sample to be inferred be $x = (c, s, t)$ and y, where x represents the input, including c: the context string, s: the source event spans within c, and t: the target event spans within c; y: the ground truth causal label, which takes a value of "yes" or "no". Additionally, we define the corpus as $\mathcal{Q} = \{(c_i, s_i, t_i, y_i) \mid i \in \mathbb{Z}^+\}$, where each sample also contains context, source event, target event, and the ground truth.

3.3 Joint Structural Metrics

In this section, we introduce two structure-based metrics to calculate the score of each sample in the corpus.

3.3.1 Conceptual Path Metric

This metric is based on ConceptNet (Speer et al., 2018). ConceptNet is a freely available semantic knowledge graph, where the nodes represent various concepts from the real world and the edges represent the relationships between those concepts.

For a given sample $x = (c, s, t)$, we first match the concept nodes in ConceptNet corresponding to s and t. To improve the matching rate, we use an encoder to encode both the events and all the nodes in

ConceptNet. Then, we compute the cosine similarity between the event representation and the node representation, selecting the nodes with the highest similarity that exceeds a predefined threshold as the matching nodes for the events in ConceptNet, denoted as $node_s$ and $node_t$. Then, using the shortest path algorithm (denoted as SP), we find a path in ConceptNet between s and t:

$$path_{s,t} = \text{SP}(node_s, node_t).$$

It is important to note that this path is not strictly unidirectional, but rather any path where there is an edge between any two nodes along the path. Using the same method, for each corpus sample $q_i \in \mathcal{Q}$, we obtain $path_{s_i,t_i}$. Subsequently, we calculate the edit distance (ED) based similarity between the two paths, obtaining the Conceptual Path based score:

$$S^{path}_{x,q_i} = 1 - \frac{\text{ED}(path_{s,t}, path_{s_i,t_i})}{\max(|path_{s,t}|, |path_{s_i,t_i}|)},$$

where $|path_{s_i,t_i}|$ denotes the length of path between s_i and t_i.

3.3.2 Syntactic Metric

This metric is based on the syntactic structure of the text. In this paper, we design our method using the dependency syntax tree as the syntactic feature of the text. For the given sample x's text c, we first obtain its dependency syntax tree $tree_c$. Similarly, for each corpus sample $q_i \in \mathcal{Q}$, we obtain $tree_{c_i}$. In particular, for a text containing multiple sentences, we first obtain the syntax tree of each sentence separately, and then connect them to an artificial root node to construct the syntax tree of the entire text. After obtaining two trees, we calculate the tree edit distance (TED) based similarity between the two trees, obtaining the Syntax based score:

$$S^{syn}_{x,q_i} = \text{e}^{-0.05 \cdot \text{TED}(tree_c, tree_{c_i})}.$$

After calculating S^{path}_{x,q_i} and S^{syn}_{x,q_i}, we use their weighted sum as the joint structural score of the corpus sample q_i for the sample to be inferred x:

$$S_{x,q_i} = w_1 \cdot S^{path}_{x,q_i} + w_2 \cdot S^{syn}_{x,q_i},$$

where w_1 and w_2 are predefined weights. Using the weighted sum as the final score helps avoid missing samples that satisfy only specific structures.

Algorithm 1 Structure Based Example Retrieval

Input: $x = (c, s, t)$; $\mathcal{Q} = \{(c_i, s_i, t_i, y_i)\}$; k_{top}

$node_s \leftarrow$ match ConceptNet node for s
$node_t \leftarrow$ match ConceptNet node for t
$path_{s,t} \leftarrow \text{SP}(node_s, node_t)$

$tree_c \leftarrow$ syntactic parse tree for c
for $q_i = (c_i, s_i, t_i, y_i) \in \mathcal{Q}$ **do**
  $node_{s_i} \leftarrow$ match ConceptNet node for s_i
  $node_{t_i} \leftarrow$ match ConceptNet node for t_i
  $path_{s_i,t_i} \leftarrow \text{SP}(node_{s_i}, node_{t_i})$
  $S^{path}_{x,q_i} \leftarrow \text{norm\_sim}(\text{ED}(path_{s,t}, path_{s_i,t_i}))$
  $tree_{c_i} \leftarrow$ syntactic parse tree for c_i
  $S^{syn}_{x,q_i} \leftarrow \text{norm\_sim}(\text{TED}(tree_c, tree_{c_i}))$
  $S_{x,q_i} \leftarrow w_1 \cdot S^{path}_{x,q_i} + w_2 \cdot S^{syn}_{x,q_i}$
end for

$cp \leftarrow \text{PatternExtractor}(c, s, t)$
$\mathcal{E}' = \{\}$
for $q_i = (c_i, s_i, t_i, y_i) \in \mathcal{Q}$ **do**
  $cp_i \leftarrow \text{PatternExtractor}(c_i, s_i, t_i)$
  if $cp_i = cp$ **then**
    $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{q_i\}$
  end if
end for

$\mathcal{E}' \leftarrow$ sort $\mathcal{E}'$ by S_{x,q_i} in descending order

$k_{pos} \leftarrow \lfloor k_{top}/2 \rfloor$
$k_{neg} \leftarrow k_{top} - k_{pos}$
$\mathcal{E}'_{pos} \leftarrow$ first k_{pos} samples in $\{q_i \in \mathcal{E}' | y_i = \text{“yes”}\}$
$\mathcal{E}'_{neg} \leftarrow$ first k_{neg} samples in $\{q_i \in \mathcal{E}' | y_i = \text{“no”}\}$
$\mathcal{E} \leftarrow \mathcal{E}'_{pos} \cup \mathcal{E}'_{neg}$

Output: $\mathcal{E}$

3.4 Causal Pattern Filtering

In this section, we follow the prior work Dr.ECI (Cai et al., 2025) to use the causal pattern as the structural feature of the instance and filtering samples based on this. Unlike Dr.ECI, we further specify the definition of causal patterns, allowing the LLM to more accurately distinguish between different patterns from a structural perspective, thereby achieving more precise pattern extraction. Moreover, in this step, we only require the LLM to predict the coarse-grained pattern according to predefined rules, which is easier for the model than directly performing end-to-end causal reasoning. The pattern descriptions are shown in Table 1.

For a given sample $x = (c, s, t)$, we construct a prompt using its context and two events, allowing the LLM to generate the corresponding causal pattern, which is then extracted using regular expressions:

$$cp = \text{PatternExtractor}(c, s, t),$$

where PatternExtractor is an prompted LLM. Similarly, for each positive sample in the corpus, we obtain its causal pattern:

$$cp_i = \text{PatternExtractor}(c_i, s_i, t_i).$$

For negative samples, their Causal Pattern is directly set to "No".

Subsequently, we can begin filtering all samples in $\mathcal{Q}$ to obtain the examples needed for inference on x. First, we perform exact matching of the causal pattern to initially obtain

$$\mathcal{E}' = \{q_i | q_i \in \mathcal{Q} \wedge cp_i = cp\}.$$

Following this, we sort $\mathcal{E}'$ in descending order based on S_{x,q_i}. We select k_{top} examples with the highest scores from $\mathcal{E}'$. To reduce the "causal hallucination" phenomenon, we ensure that both positive and negative examples are selected. Finally, we get the resulting set of examples $\mathcal{E}$.

The algorithm for the entire example selection process is described in Algorithm 1. It is worth noting that the overall retrieval pipeline is efficient and practical rather than complex, as long as the corpus and knowledge base are pre-constructed.

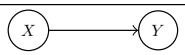
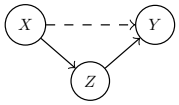
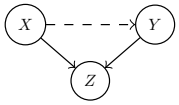
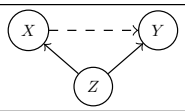
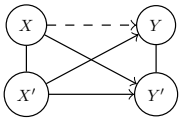
Causal Pattern	Causal Graph
Direct	
Chain	
Collider	
Fork	
Coreference	

Table 1: The Causal Patterns used in this paper. X and Y represent the source and target event, respectively; Z represents mediators; X' and Y' denote the coreferential events of the corresponding events. Solid lines indicate explicitly stated relationships in the context, while dashed lines represent inferred implicit relationships. A more detailed description of the patterns can be found in the appendix H.

3.5 LLM Reasoning

After obtaining $\mathcal{E}$, we can use it to guide the LLM for the final causal reasoning, which we treat as a few-shot ICL-based QA task. We design the prompt based on x and $\mathcal{E}$, and use an LLM: Reasoner for reasoning to obtain the final result:

$$y' = \text{Reasoner}(x, \mathcal{E}),$$

where y' is the final answer. Since this task is a binary classification task, y' is restricted to be either "Yes" or "No".

4 Experiment

4.1 Implementation Details

For the Conceptual Path Metric, we construct a localized ConceptNet on Neo4j (Neo4j, 2012), encode nodes with Contriever-msmarco (Izacard et al., 2021), and query the shortest paths between events using Cypher. For the Syntactic Metric, we build dependency trees from the text using spaCy (Honnibal and Montani, 2017). The shortest path algorithm and distance-based similarity algorithm are provided in the appendix I and J.

In the main experiments, we set the confidence threshold for node matching to 0.6, the weights of the conceptual path and syntactic metrics (w_1, w_2) to 0.5 each, the number of selected examples (k_{top}) to 2, and the LLM temperature to 0 to ensure consistent responses.

We directly use the official APIs to access `gpt-4o-mini-2024-07-18` (OpenAI, 2024) and `gemini-1.5-pro` (Team et al., 2024) in the main experiments. The prompts used in this paper are provided in the appendi K.

4.2 Datasets and Metrics

We conduct experiments on three causality-annotated datasets: **EventStoryLine (ESC)** (Caselli and Vossen, 2017), which provides event, temporal, and causality annotations (22 topics, 258 docs, 5,334 events, 1,770 causal pairs); **Causal-TimeBank (CTB)** (Mirza et al., 2014), sourced from TempEval-3 (UzZaman et al., 2013) (183 docs, 6,813 events, 318 causal pairs); and **MAVEN-ERE** (Wang et al., 2022), a large-scale Wikipedia-based dataset (90 topics, 4,480 docs, 103,193 events, 57,992 causal pairs). Data preprocessing follows Gao et al. (2023). For evaluation, we adopt Precision (**P%**), Recall (**R%**), and F1 (**F1%**) as metrics.

Method	ESC			CTB			MAVEN-ERE		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
PaLM2:									
Dr.ECI†	29.0	75.4	41.9	-	-	13.0	22.0	80.0	34.5
GPT-4o-mini:									
Base	28.4	82.9	42.3	5.4	80.5	10.1	23.1	91.5	36.9
CoT	29.4	80.6	43.1	6.3	79.6	11.6	25.4	87.9	39.4
Dr.ECI	31.0	**90.2**	46.1	8.2	**91.2**	15.1	26.0	**93.0**	40.7
SERE (ours)	**48.3**	51.5	**49.9**	**13.8**	36.3	**20.0**	**34.5**	54.6	**42.3**
Gemini-1.5-pro:									
Base	24.0	80.8	37.0	4.7	**95.6**	9.0	21.2	**95.1**	34.6
CoT	25.1	**85.4**	38.7	4.7	88.5	8.9	23.3	89.2	37.0
Dr.ECI	27.6	82.0	41.3	7.2	88.5	13.3	23.4	91.7	37.3
SERE (ours)	**36.5**	59.3	**45.2**	**9.9**	69.9	**17.4**	**29.9**	60.2	**39.9**

Table 2: Main experiment results. The best scores are marked in bold. †: results from (Cai et al., 2025).

4.3 Baselines

In the main experiment, we primarily compare methods based on pre-trained LLMs:

- **Base**: The LLM is prompted with a simple task description without additional guidance.
- **CoT** (Wei et al., 2023): The LLM is prompted to generate a step-by-step reasoning process before producing the final answer, enhancing its reasoning capability.
- **Dr.ECI** (Cai et al., 2025): The task is decomposed into multiple steps: the model first identifies mediators in the context that connect the target and source, and then follows predefined causal patterns to guide causal reasoning. For a fair comparison, we report both the original results obtained with PaLM2 (Anil et al., 2023) and our reproduced scores. However, since PaLM2 is not publicly accessible, we did not conduct experiments using PaLM2.

Since **SERE** relies on LLM APIs without fine-tuning, the main experiments only include non-fine-tuning baselines, while additional comparisons with fine-tuned models are provided in the analysis section.

To further demonstrate the generalizability of our method, we also conduct experiments under the fine-tuning setting, as detailed in the appendix B.

4.4 Main Results

We report the Precision, Recall, and F1 scores of different methods on the ESC, CTB, and MAVEN-ERE datasets, as detailed in Table 2.

As shown in the table, our method achieves the highest F1 scores across all three datasets, demonstrating its generalization capability. The Base method for both models performs the worst on the CTB dataset, whereas our method achieves nearly double the score improvement and an approximately 20% increase on the ESC dataset. Compared to the Base method, the CoT method shows slight improvements in most cases, but the gains are limited. CoT may help reduce the model's reliance on surface-level causal keywords, thereby improving precision, but step-by-step reasoning alone is insufficient to address causal bias. The low Precision and high Recall of the Base and CoT methods indicate the presence of "Causal Hallucination", where the model without specific optimization tends to misclassifies non-causal relationships as causal. In contrast, **SERE** effectively suppresses such misclassifications, reducing the hallucination.

Dr.ECI generally exhibits higher Recall but limited improvements in Precision. In contrast, **SERE** enhances final performance by improving precision, indicating that our method is more conservative. Due to the use of three retrieval methods, **SERE** applies stricter criteria when selecting examples, which may lead to missing some causal relationships. This could explain why **SERE** achieves higher Precision but lower Recall. Considering that current LLM-based approaches struggle to balance precision and recall, our method achieves a trade-off between the two. While this means that some true causal relations may be missed, it significantly

Method	ESC-intra			ESC-inter			CTB		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
GenECI	59.5	57.1	58.8	-	-	-	60.1	53.3	56.5
DPJL	65.3	70.8	67.9	-	-	-	63.6	66.7	64.6
KEPT	50.0	68.8	57.9	-	-	-	48.2	60.0	53.5
CPATT	79.4	81.3	**80.4**	74.9	60.1	66.7	77.5	73.2	75.2
SERE (ours)	75.5	81.6	78.4	66.7	87.2	**75.6**	84.3	89.4	**86.8**

Table 3: CPATT settings results. The results of GenECI (Man et al., 2022), DPJL (Shen et al., 2022), KEPT (Liu et al., 2023) and CPATT (Zhang et al., 2023) are all taken from the paper of CPATT. ESC-intra and ESC-inter refer to subsets of the ESC dataset where the two events appear in the same sentence or in different sentences, respectively. CTB does not distinguish between intra and inter cases. The best scores are marked in bold.

reduces the risk of the model generating spurious causal links, making our approach more suitable for risk-sensitive scenarios. Overall, **SERE** achieves stronger performance across different models and datasets.

5 Analysis Experiments

5.1 CPATT Settings

To demonstrate the generality of **SERE**, we further evaluate its performance in the fine-tuning scenario, following the experimental settings of CPATT (Zhang et al., 2023), a prior state-of-the-art method. CPATT fine-tunes an LLM with a different preprocessing strategy from Gao et al. (2023): for long contexts, only sentences containing the target events are retained, while negative samples are constructed by randomly pairing non-causal events with their corresponding sentences. We directly perform inference on the test split.

As shown in Table 3, **SERE** performs consistently well. On ESC-intra, it achieves an F1 of 78.4%, ranking second only to CPATT, while on ESC-inter it attains the best F1 of 75.6%. Although fine-tuned CPATT performs better in intra-sentence settings, its precision and recall drops in cross-sentence contexts. In contrast, the non-fine-tuned SERE remains stable across settings, showing stronger robustness and generalization. On CTB, **SERE** also achieves marginally the highest precision, recall, and F1, further confirming its cross-domain robustness.

Note that CPATT's preprocessing is primarily designed for training, where random negative sampling helps reduce causal hallucination; therefore, these results are not included in the main experiments. Nevertheless, they clearly demonstrate the strong generalization ability of **SERE**.

5.2 Ablation Study

Component	ESC	CTB	MAVEN-ERE
SERE	49.9	20.0	42.3
w/ Conceptual Path	46.7	18.9	39.2
w/ Syntactic	44.5	18.9	38.7
w/ Causal Pattern	47.1	17.3	38.3

Table 4: Performance when retrieval is guided by a single structural signal. For "w/ Causal Pattern", we randomly select k samples with the same pattern. The evaluation metric is F1(%).

Component	ESC	CTB	MAVEN-ERE
SERE	49.9	20.0	42.3
w/o Conceptual Path	46.6	18.8	40.8
w/o Syntactic	48.1	18.9	40.5
w/o Causal Pattern	47.5	18.2	39.4

Table 5: Performance when one structural signal is removed from **SERE**. For "w/o Causal Pattern", the Causal Pattern Filtering step is skipped, and retrieval relies solely on the Joint Structural Metrics. The evaluation metric is F1 (%).

We conduct ablation studies to examine the role of three structural signals in retrieval. As shown in Table 4, using any single structural signal alone leads to a noticeable performance drop compared to the full **SERE** model, although it still outperforms the Base method. This suggests that each signal captures useful but incomplete structural information.

Further analysis in Table 5 shows that removing any one component from **SERE** also results in consistent degradation across all datasets. This indicates that the three signals provide complementary, non-redundant contributions to the retrieval process. While combining any two signals already

yields clear improvements over the Base method, it remains insufficient to match the full model.

Overall, the best performance is achieved when all three structural signals are jointly incorporated. This demonstrates that effective retrieval benefits from simultaneously enforcing structural type consistency (via Causal Pattern) and structural similarity (via Conceptual path and syntactic metrics), leading to more accurate and robust example selection.

5.3 Alternative Retrieval Methods Analysis

Retrieval Method	ESC	CTB	MAVEN-ERE
Base	42.3	10.1	36.9
Random	46.6	13.9	39.1
Contriever-msmarco	46.2	18.8	37.9
BM25	46.5	17.1	33.4

Table 6: Experiments of three retrieval methods. The evaluation metric is F1 (%).

We evaluate three retrieval strategies. **Random** samples k instances uniformly at random. **Contriever-msmarco** encodes text into dense vectors and retrieves examples based on cosine similarity. **BM25** ranks candidates using term frequency–inverse document frequency (TF–IDF). In both cases, the top-k samples are selected according to their relevance scores. The results are shown in Table 6. Compared with **SERE**, all three reported methods (Random, Contriever-msmarco, and BM25) lead to performance drops. Although adding examples generally helps—yielding higher scores than the Base method on ESC and CTB—BM25 performs worse than Base on MAVEN-ERE, indicating that traditional scoring-based retrieval may be unsuitable for ECI.

By jointly examining Tables 4 and 5, we observe that using a single structural signal or a pair of signals generally performs better than traditional retrieval, underscoring the need to integrate structural cues.

5.4 Effect of varying the number of retrieved examples

We study the impact of varying the number of retrieved examples on the performance of **SERE**. In the main experiment, we set $k_{top} = 2$. As shown in Table 7, increasing k_{top} to 4 yields a slight improvement on CTB, but leads to minor performance drops on ESC and MAVEN-ERE. When k_{top} is further increased to 6, performance consistently degrades across all three datasets.

top-k	ESC	CTB	MAVEN-ERE
top-2	49.9	20.0	42.3
top-4	49.0	22.7	39.8
top-6	47.1	19.7	38.9

Table 7: The results of varying numbers of retrieved examples. The evaluation metric is F1 (%).

We attribute this trend to the limited context capacity of LLMs. Incorporating more demonstrations increases the prompt length and may introduce structurally similar yet less relevant examples, which can dilute informative signals. Notably, across all examined values, **SERE** consistently outperforms Base, CoT, and Dr.ECI, indicating that the method remains robust within a reasonable range of k_{top}.

5.5 Causal Pattern Extraction Analysis

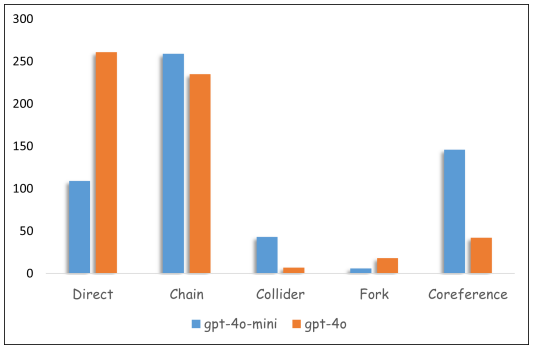


Figure 3: The Causal Pattern extraction results for the positive samples in the ESC dataset by two models. Due to cost constraints, we conduct experiments only on a subset of ESC.

We analyze the Causal Pattern extraction results of GPT-4o-mini and GPT-4o on the positive sample set of the ESC dataset, as shown in Figure 3.

Both models identify a large number of Direct and Chain patterns, while Collider and Fork patterns are rarely recognized. This suggests that the ESC dataset is likely dominated by the first two patterns, which aligns with intuition: most texts, especially the short texts in the dataset, contain relatively simple causal relationships or are explicitly marked by causal trigger words. The low occurrence of the latter two patterns may also indicate that the models struggle to recognize more complex causal structures.

GPT-4o perceives Direct and Chain patterns in similar proportions, whereas GPT-4o-mini identifies significantly fewer Direct patterns compared to Chain. This suggests that the mini one may have a stronger preference for parsing multi-hop reasoning while being less inclined to recognize direct causal relationships. Additionally, some causal relationships that the mini model considers to require intermediate steps are classified as Direct by GPT-4o. This may be due to GPT-4o's stronger reasoning ability, allowing it to infer causal relationships directly based on its internal knowledge.

6 Related Work

6.1 Structure-based Methods in ECI

Traditional structure-based methods mainly focus on lexical and syntactic patterns (Riaz and Girju, 2013; Hashimoto et al., 2014), aiming to capture surface-level cues that signal causal relations. Rule-based approaches typically rely on handcrafted linguistic patterns and predefined causal connectors (*e.g.*, "`causes`", "`leads to`"), which offer interpretability but require substantial manual effort and domain expertise.

To alleviate these issues, recent studies have increasingly turned to pre-trained models, such as SemSIn (Hu et al., 2023) for implicit causal detection and Dr.ECI (Cai et al., 2025) for predefined causal patterns. In addition, some works incorporate external knowledge bases such as ConceptNet (Cao et al., 2021; Huang et al., 2024; Liu et al., 2023; Su et al., 2025) to facilitate the discovery of latent relations between events. However, most existing research relies on training models, which increases implementation complexity and limits the flexibility of these approaches.

6.2 Generative LLM in ECI

The rise of generative LLMs has reshaped the ECI research paradigm. Many methods that directly invoke LLMs have achieved strong performance in the ECI domain (Sun et al., 2024; Wang et al., 2024; Guan et al., 2025; Zeng et al., 2025), demonstrating the potential of large language models to capture complex causal relations without extensive task-specific training.

However, despite their causal reasoning ability, the "causal hallucination" phenomenon highlights a clear gap compared with fine-tuned models (Gao et al., 2023), particularly in tasks requiring precise and reliable inference.

Many prompt-based fine-tuning methods for LLMs have also demonstrated significant effectiveness on the ECI task (Peng et al., 2023; Hu et al., 2025; Xiang et al., 2025). By incorporating task-specific prompts or instructions during training, these approaches can better align the model with causal reasoning objectives. Nevertheless, they still rely heavily on the availability of high-quality annotated data, and their performance can degrade when applied to low-resource or cross-domain settings.

6.3 In-context Learning in LLM

The in-context learning (ICL) ability of LLMs allows them to perform tasks with only a few examples, as widely demonstrated in prior work (Brown et al., 2020; Touvron et al., 2023; Bertsch et al., 2025; Li et al., 2024). However, ICL performance is highly sensitive to factors such as the number, order, distribution, and quality of examples (Agarwal et al., 2024; Kumar and Talukdar, 2021; Min et al., 2022; Zhao et al., 2021; Arora et al., 2022; Voronov et al., 2024), making the selection and processing of samples crucial (Peng et al., 2024). Traditional approaches often rely only on general statistical patterns or sentence semantics and do not exploit task-specific causal priors, which limits their suitability for ECI. Therefore, we propose a structure-based method.

7 Conclusion

In this paper, we propose **SERE**, a structural example retrieval framework designed to enhance LLM performance in ECI tasks. **SERE** introduces three structural factors: conceptual path, syntactic structure, and causal patterns. The core idea of **SERE** is to integrate these structural concepts and design a retrieval method that incorporates two structural similarity metrics—Conceptual Path Metric and Syntactic Metric—along with a Causal Pattern Filtering mechanism into few-shot learning, thereby improving LLMs' causal reasoning ability in ECI. Experimental results demonstrate the effectiveness of **SERE** across multiple ECI datasets. We believe that **SERE** and the underlying structural concepts provide valuable insights for future research in ECI, particularly in refining example selection strategies and improving LLM adaptability in causal reasoning tasks.

Limitations

This work has two main limitations: (1) Due to the high cost of API calls and deploying, we did not evaluate the performance of other mainstream large models or more open-source models. (2) For the three types of structural information mentioned in this paper, we only explore their application in example retrieval, and further development remains for future research.

Acknowledgments

This research was supported in part by National Science and Technology Major Project (2021ZD0111502), Natural Science Foundation of China (U24A20233, 62406078, 62476163), the Guangdong Basic and Applied Basic Research Foundation (2023B1515120020), CCF-DiDi GAIA Collaborative Research Funds (CCF-DiDi GAIA 202521), and Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)(GML-KF-24-23).

References

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-shot in-context learning. *Preprint*, arXiv:2404.11018.

Rohan Anil, Andrew M. Dai, Orhan Firat, and 1 others. 2023. Palm 2 technical report. *Preprint*, arXiv:2305.10403.

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models. *Preprint*, arXiv:2210.02441.

Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. 2025. In-context learning with long-context models: An in-depth exploration. *Preprint*, arXiv:2405.00200.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Ruichu Cai, Shengyin Yu, Jiahao Zhang, Wei Chen, Boyan Xu, and Keli Zhang. 2025. Dr.ECI: Infusing large language models with causal knowledge for decomposed reasoning in event causality identification. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9346–9375, Abu Dhabi, UAE. Association for Computational Linguistics.

Pengfei Cao, Xinyu Zuo, Yubo Chen, Kang Liu, Jun Zhao, Yuguang Chen, and Weihua Peng. 2021. Knowledge-enriched event causality identification via latent structure induction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4862–4872, Online. Association for Computational Linguistics.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Jesse Dunietz, Lori Levin, and Jaime Carbonell. 2015. Annotating causal language using corpus lexicography of constructions. In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 188–196, Denver, Colorado, USA. Association for Computational Linguistics.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is chatgpt a good causal reasoner? a comprehensive evaluation. *Preprint*, arXiv:2305.07375.

Yong Guan, Hao Peng, Lei Hou, and Juanzi Li. 2025. MMD-ERE: Multi-agent multi-sided debate for event relation extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6889–6896, Abu Dhabi, UAE. Association for Computational Linguistics.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 2014. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 987–997, Baltimore, Maryland. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2023. Semantic structure enhanced event causality identification. *Preprint*, arXiv:2305.12792.

Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2025. Large language model-based event relation extraction with rationales. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7484–7496, Abu Dhabi, UAE. Association for Computational Linguistics.

Peixin Huang, Xiang Zhao, Minghao Hu, Zhen Tan, and Weidong Xiao. 2024. Distill, fuse, pre-train: Towards effective event causality identification with commonsense-aware pre-trained model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5029–5040, Torino, Italia. ELRA and ICCL.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

Sawan Kumar and Partha Talukdar. 2021. Reordering examples helps during priming-based few-shot learning. *Preprint*, arXiv:2106.01751.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *Preprint*, arXiv:2101.06804.

Jintao Liu, Zequn Zhang, Zhi Guo, Li Jin, Xiaoyu Li, Kaiwen Wei, and Xian Sun. 2023. Kept: Knowledge enhanced prompt tuning for event causality identification. *Knowledge-Based Systems*, 259:110064.

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. *Preprint*, arXiv:1907.11692.

Hieu Man, Minh Nguyen, and Thien Nguyen. 2022. Event causality identification via generation of important context words. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 323–330, Seattle, Washington. Association for Computational Linguistics.

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? *Preprint*, arXiv:2202.12837.

Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. 2014. Annotating causality in the TempEval-3 corpus. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, Gothenburg, Sweden. Association for Computational Linguistics.

Neo4j. 2012. Neo4j - the world's leading graph database.

OpenAI. 2024. Chatgpt. `https://www.openai.com/chatgpt`. Accessed: 2024-07-13.

Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2023. When does in-context learning fall short and why? a study on specification-heavy tasks. *Preprint*, arXiv:2311.08993.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. *Preprint*, arXiv:2401.12087.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Mehwish Riaz and Roxana Girju. 2013. Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *Proceedings of the SIGDIAL 2013 Conference*, pages 21–30, Metz, France. Association for Computational Linguistics.

Shirong Shen, Heng Zhou, Tongtong Wu, and Guilin Qi. 2022. Event causality identification via derivative prompt joint learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2288–2299, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge. *Preprint*, arXiv:1612.03975.

Ya Su, Hu Zhang, Guangjun Zhang, Yujie Wang, Yue Fan, Ru Li, and Yuanlong Wang. 2025. Enhancing event causality identification with LLM knowledge and concept-level event relations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7403–7414, Abu Dhabi, UAE. Association for Computational Linguistics.

Yidan Sun, Qin Chao, and Boyang Li. 2024. Event causality is key to computational story understanding. *Preprint*, arXiv:2311.09648.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and 1 others. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. *Preprint*, arXiv:2401.06766.

Haoyu Wang, Fengze Liu, Jiayao Zhang, Dan Roth, and Kyle Richardson. 2024. Event causality identification with synthetic control. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1725–1737, Miami, Florida, USA. Association for Computational Linguistics.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Wei Xiang, Chuanhong Zhan, Qing Zhang, and Bang Wang. 2025. Evaluating instructively generated statement by large language models for directional event causality identification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 779–785, Vienna, Austria. Association for Computational Linguistics.

Zefan Zeng, Xingchen Hu, Qing Cheng, Weiping Ding, Wentao Li, and Zhong Liu. 2025. Zero-shot event causality identification via multi-source evidence fuzzy aggregation with large language models. *Preprint*, arXiv:2506.05675.

Hang Zhang, Wenjun Ke, Jianwei Zhang, Zhizhao Luo, Hewen Ma, Zhen Luan, and Peng Wang. 2023. Prompt-based event relation identification with constrained prefix attention mechanism. *Knowl. Based Syst.*, 281:111072.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *Preprint*, arXiv:2102.09690.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Xinyu Zuo, Yubo Chen, Kang Liu, and Jun Zhao. 2020. KnowDis: Knowledge enhanced data augmentation for event causality detection via distant supervision. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1544–1550, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Broader Societal Implications and Potential Risks

Potential societal risks primarily relate to biases in external knowledge sources and the consequences of incorrect causal inference in high-stakes contexts. External knowledge bases such as ConceptNet may encode cultural, gender, or sociodemographic biases and exhibit uneven domain coverage. If not carefully managed, these biases could influence retrieval behavior or subtly distort causal interpretation. In addition, misidentifying causal relations in domains such as healthcare, law, or public policy may lead to misleading conclusions or inappropriate downstream decisions.

SERE mitigates these risks by not treating any single source—whether ConceptNet relations or LLM outputs—as authoritative causal evidence. Instead, these components serve as complementary

structural cues within a retrieval-based framework alongside syntactic structure and causal pattern abstractions. This multi-signal design helps prevent any single biased or spurious source from dominating the final decision, and grounding predictions in multiple retrieved examples further reduces the impact of isolated errors.

Even so, responsible use of event causal inference systems requires continued attention to fairness, robustness, and domain safety. Future work may incorporate bias detection, domain-aware filtering of external knowledge, and human oversight, particularly in safety-critical applications.

B Performance of Finetuned LLM

To further demonstrate the generalizability of the **SERE** method, we fine-tune Qwen2.5-3B-Inst (Qwen et al., 2025) using the supervised instruction-based LoRA (Hu et al., 2021) approach with the LlamaFactory library (Zheng et al., 2024). Each training instance has the same input as in the API-based setup. We then compare it with CPATT, which is also fine-tuned (using BART-base (Lewis et al., 2019) with 139M parameters). The results are shown in Table 8.

As the results show, **SERE** that uses fine-tuned Qwen outperforms both its base counterpart and the CPATT method in terms of F1 score, achieving a new state-of-the-art level. In essence, fine-tuning improves both precision and recall, thereby alleviating the issue of "causal hallucination" while ensuring maximum coverage of positive instances.

It should be noted that our **SERE** method is not originally designed for the fine-tuning setting. Therefore, we do not consider this experiment part of the main experiments. Nevertheless, based on these scores, we believe this experiment demonstrates the generalizability and robustness of the **SERE** method in the fine-tuning setting.

C Effects of Individual Structural Components

To further support our design and clarify the effect of each structure, we conduct additional ablation studies. We directly incorporate Conceptual Path, Syntactic Tree, and Causal Pattern into the Base prompt, without introducing examples. As shown in table 9, all three structures independently improve the model's performance compared to the Base prompt. Conceptual Path significantly boosts Recall, especially on ESC and CTB. This aligns with intuition—LLMs often tend to predict "causal" by default, and the path connecting events makes the model more confident, albeit at the risk of false positives. Syntactic Tree also improves both Precision and Recall slightly, possibly because it helps the model better understand grammatical dependencies relevant to causality. Causal Pattern enhances Precision but reduces Recall. We attribute this to the stricter constraints imposed by predefined patterns, which encourage the model to avoid overpredicting causality—thereby mitigating hallucination at the cost of coverage.

These results further validate that each structure contributes differently and meaningfully to the model's reasoning behavior. While our current work applies these signals in a retrieval scenario, we believe their independent effects suggest deeper potential.

D Applying Dr.ECI's Causal Pattern in SERE

We conduct experiments using the original Causal Pattern defined in the Dr.ECI paper (Cai et al., 2025) on **SERE**, and the results are shown in Table 10. As can be seen, using the original causal pattern leads to a performance drop for **SERE** across all three datasets. This may be due to the inclusion of some rules unrelated to structure in the extraction process of the causal pattern, such as determining direct/indirect pattern based on the presence of the trigger words, and allowing the LLM to determine the pattern based on common sense. This causes the model to inaccurately grasp the intrinsic structure of the sample, leading to incorrect pattern extraction and ultimately misleading the final causal reasoning.

E Inference Cost Analysis

To provide a clearer picture of **SERE**'s efficiency, we conduct an additional runtime and token usage analysis using 300 random samples from the ESC dataset. We measure inference latency and token cost using GPT-4o-mini for the three methods: CoT, Dr.ECI, and **SERE**. The results are shown in table 11.

SERE is slower overall primarily because it performs structural example retrieval over a corpus of 3,000 instances. However, only about 8 seconds of **SERE**'s runtime come from LLM inference. The remaining time is dominated by structure-matching computations—especially tree-edit-distance evalu-

	ESC-intra			ESC-inter			ESC			CTB		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
CPATT	79.4	81.3	80.4	74.9	60.1	66.7	76.5	66.2	71.0	77.5	73.2	75.2
Qwen2.5-3B-Inst:												
Base	76.1	84.3	79.9	77.3	66.7	71.6	78.5	69.4	73.7	95.0	86.4	90.5
SERE	**77.4**	**90.1**	**83.3**	**78.8**	**72.8**	**75.7**	**79.3**	**73.7**	**76.4**	**91.3**	**95.5**	**93.3**

Table 8: Comparison between finetuned **SERE** and CPATT. Base indicates that the fine-tuning instructions do not include retrieved examples; **SERE** indicates that the instructions include examples retrieved by the **SERE** method. The results of CPATT are taken from the original paper (Zhang et al., 2023).

Method	ESC			CTB			MAVEN-ERE		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Base	28.4	82.9	42.3	5.4	80.5	10.1	23.1	91.5	36.9
w/ Conceptual Path	28.8	93.3	44.0	5.7	93.8	10.7	24.0	92.6	38.1
w/ Syntactic Tree	29.3	91.1	44.3	6.0	93.8	11.3	25.5	91.1	39.9
w/ Causal Pattern	31.6	75.5	44.5	6.4	72.6	11.8	26.4	77.8	39.5

Table 9: Effects of individual structural components (Conceptual Path, Syntactic Tree and Causal Pattern) when added to the Base prompt.

Method	ESC	CTB	MAVEN-ERE
SERE	49.9	20.0	42.3
w/ Dr.ECI	47.0	16.1	41.4

Table 10: The results of applying Dr.ECI's Causal Patterns to **SERE**. The evaluation metric is F1 (%).

Method	Time (s)	Input Tokens	Output Tokens
CoT	4.02	137	285
Dr.ECI	14.26	2017	872
SERE	21.85	1843	621

Table 11: The average time and the token usage of different methods during inference.

ation for syntactic structures—which are performed on CPU.

Although structural retrieval introduces additional overhead, **SERE** achieves substantially better task performance. Given that the retrieval step can naturally benefit from standard acceleration strategies such as parallel computation, cached structural features in future deployments, we believe this cost–effectiveness trade-off is acceptable for many settings.

In terms of token consumption, **SERE** consumes fewer tokens than Dr.ECI, since it uses fewer LLM calls and retrieves a controlled number of demonstrations. This makes **SERE** comparatively more practical in environments where token cost is a bottleneck.

F Computational Cost and Baseline Comparison

Method	ESC	CTB	MAVEN-ERE
Stage 1	43.7	11.3	39.3
Stage 2 (w/ examples)	12.5	1.6	11.7
Stage 2 (w/o examples)	31.6	7.3	29.9
SERE	**49.9**	**20.0**	**42.3**

Table 12: Comparison of **SERE** with two-stage LLM refinement baselines.

We quantify the number of LLM calls and assess whether comparable performance can be achieved using simpler inference pipelines. In **SERE**, LLM usage mainly occurs in two stages: (1) a one-time corpus preprocessing step, where each instance is processed once to extract Causal Patterns, and (2) the inference stage, where each test instance requires two calls—one for extracting its Causal Pattern and one for generating the final prediction conditioned on retrieved examples.

To investigate whether similar performance can be obtained with fewer calls, we construct a two-stage refinement baseline. In Stage 1, the LLM predicts causal relations using a CoT prompt. In Stage 2, it is queried again to reassess the initial reasoning and output the final label, optionally with randomly selected examples. This setup limits inference to two LLM calls per instance, acknowledg-

ing that corpus-level pattern extraction in **SERE** is performed once offline and does not add to per-sample cost.

The results in Table 12 show that invoking the LLM twice without carefully curated evidence leads to a substantial performance drop. The degradation becomes even more pronounced when randomly selected examples are introduced, indicating that the model cannot reliably self-correct using unrelated or unstructured contextual inputs. These results demonstrate that **SERE**'s improvements do not simply stem from increasing the number of LLM calls or exposing the model to arbitrary examples.

G Case Study

We select four cases for illustration.

In Case 1, it can be seen that the CoT method exhibits "causal hallucination", whereas the **SERE** method correctly infers that there is no causal relationship in the given example.

Case 2 demonstrates how the PatternExtractor in the **SERE** method accurately identifies a indirect causal relationship. Compared to Dr.ECI's multi-agent approach, we accomplish this step using only a single LLM. By following our manually crafted rules, the LLM accurately extracts the causal pattern.

Case 3 shows a representative positive example, including retrieved supporting examples, pattern induction, and final inference.

In Case 4, the model incorrectly predicts the patterns of the examples (the two examples should have been recognized as Direct; however, as we discuss in Appendix H, GPT-4o-mini is less likely to predict Direct). Nevertheless, thanks to the assistance of the other two signals, the system is still able to identify the correct positive examples, which in turn guides the LLM to correctly complete the final reasoning.

H Causal Pattern

The detailed introduction to the Causal Pattern is in Table 13.

The causal patterns used in this paper are adapted from Dr.ECI (Cai et al., 2025), but in order to emphasize "structure", we do not use *Explicit Words*, *Implicit Words*, and *Causal Order* patterns. We use *Direct* to represent explicit causal relationships, while the rest are classified as implicit causal relationships. Additionally, we require the LLM not to use common sense knowledge for pattern identification, as its rich pre-training data and powerful reasoning abilities might lead it to assume that many implicit causal relationships can be judged by common sense knowledge, thereby failing to extract a more accurate structure.

I Implementation of the Shortest Path algorithm

We directly perform Shortest Path matching on the Neo4j graph database using the Cypher query language (see Code 1).

J Similarity Algorithm

J.1 Path Serialization

We serialize a multi-hop path in the knowledge graph into a natural-language-like sequence by concatenating quoted node identifiers and direction-sensitive relation templates. For each relation r, a pair of textual templates $\langle t_{\rightarrow}, t_{\leftarrow} \rangle$ is provided to describe the forward and inverse directions (see Table 14). During serialization, we traverse the path sequentially; for each hop, the template is selected according to the direction of the edge in the path. For example, a path $a \xrightarrow{\text{HasA}} b \xleftarrow{\text{Causes}} c$ may be serialized as

```
"a" has a "b", and "b" is caused by "c".
```

This procedure produces a uniform sequence suitable for edit distance computation.

J.2 Edit Distance based Path Similarity

After serializing each multi-hop path into a textual sequence, we measure the similarity between two paths based on the Levenshtein edit distance. Given two serialized paths t_1 and t_2, let $d(t_1, t_2)$ denote their edit distance. We normalize this distance by the length of the longer sequence to obtain a similarity score in the range $[0, 1]$:

$$\text{sim}(t_1, t_2) = 1 - \frac{d(t_1, t_2)}{\max(|t_1|, |t_2|)},$$

where $|t_i|$ denotes the length of sequence t_i. A higher value of $\text{sim}(t_1, t_2)$ indicates greater structural resemblance between the two paths.

J.3 Tree Edit Distance Based Text Similarity

Let T_1 and T_2 denote the dependency trees of two text (for a single sentence, we directly parse its dependency tree; for a multi-sentence document,

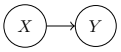
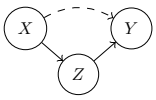
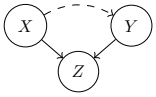
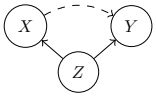
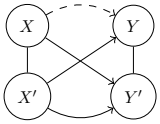
Causal Pattern	Causal Graph	Description	Example
Direct		The text explicitly states a causal relationship between X and Y.	...to **shut down[X]** resulted in workers **losing[Y]** their jobs.
Chain		There exist mediators (Z) that causally relate to both X and Y, with X acting on Z and Z acting on Y.	**Deforestation[X]** in ... soil **erosion[Z]**, which led to a **decline[Y]** in agricultural productivity.
Collider		There exist mediators (Z) that causally relates to both X and Y, with X acting on Z and Y acting on Z.	The **subsidy[X]** boosted electric vehicle ... rising **demand[Y]** led ... a **strain[Z]** on battery ....
Fork		There exist mediators (Z) that causally relate to both X and Y, with Z acting on X and Z acting on Y.	...economic **slowdown[Z]** led to a **decline[X]** in consumer spending and a **rise[Y]** in unemployment rates.
Coreference		There are different expressions of the same meaning for X and Y in the text, and these expressions are causally related.	...the heavy rain, ...water level **rose[X]**, causing **inundation[Y]** in ...areas; the **increased[X']** rainfall then triggered a **flood[Y']** disaster.

Table 13: All Causal Patterns used in this paper.

we first obtain the dependency tree of each sentence and then connect them to an artificial root node to construct the document-level dependency tree). We define a distance function $d(T_1, T_2)$ using tree edit distance, where the cost of updating, inserting, or deleting a node is determined by the dependency relation label.

Specifically, for nodes with labels l_1 and l_2, the update cost is

$$\text{update}(l_1, l_2) = \begin{cases} 0, & l_1 = l_2, \\ \text{cost}(l_1) + \text{cost}(l_2), & l_1 \neq l_2, \end{cases}$$

where the cost of each dependency label is weighted according to its structural importance (see Table 15). Core grammatical relations (*e.g.*, "`nsubj`", "`dobj`", "`ROOT`") receive higher weights, while modifiers receive lower weights. This weighted design ensures that changes to structurally central components contribute more to the edit distance than peripheral modifications.

Additionally, insertion and deletion costs are defined symmetrically:

$$\text{update}(\emptyset, l) = \text{update}(l, \emptyset) = \text{cost}(l),$$

where $\emptyset$ indicates missing label.

The normalized similarity score between two trees is then computed as

$$\text{sim}(T_1, T_2) = \text{e}^{-0.05 \cdot d(T_1, T_2)}.$$

$\text{sim}(T1, T2)$ lies in the interval $(0, 1]$. We model similarity as an exponential function of the distance to ensure a controlled decay rate. A higher value of $\text{sim}(T_1, T_2)$ indicates greater structural resemblance.

K Prompts

We present the prompts used in our paper, including the prompt for the Base method (Prompt 1); the prompt for the CoT method (Prompt 2); the prompt for extracting causal patterns from positive samples in the training set (Prompt 3); the prompt for extracting causal patterns from inference samples (Prompt 4); and the prompt used for the final inference in the SERE method (Code 2).

The difference between prompt 3 and prompt 4 is that the former one allows the LLM to know that the events in the sample must have a causal relationship, thus forcing it to infer the causal pattern and preventing it from answering "No". In these two prompts, we provide step-by-step reasoning instructions and definitions of different patterns. Additionally, we manually craft examples for each pattern to help the model extract them more accurately.

Relation	Forward Template	Inverse Template
HasContext	has context in	is the context of
MotivatedByGoal	is motivated by the goal of	motivates the goal of
FormOf	is a form of	has the form of
SimilarTo	is similar to	is similar to
HasA	has a	is owned by
dbpedia	is associated with the DBpedia concept of	has association from the DBpedia concept of
HasProperty	has the property of	is a property of
Causes	causes	is caused by
NotDesires	does not desire	is not desired by
HasPrerequisite	requires as a prerequisite	is a prerequisite for
PartOf	is part of	has as a part
Antonym	is an antonym of	is an antonym of
HasLastSubevent	has the last subevent of	is the last subevent of
MadeOf	is made of	is the material of
HasFirstSubevent	has the first subevent of	is the first subevent of
ReceivesAction	receives the action of	is the action performed on
RelatedTo	is related to	is related to
HasSubevent	has the subevent of	is a subevent of
DistinctFrom	is distinct from	is distinct from
InstanceOf	is an instance of	has an instance of
DerivedFrom	is derived from	is the origin of
UsedFor	is used for	uses
MannerOf	is a manner of	has as a manner
Desires	desires	is desired by
IsA	is a	has as a type
AtLocation	is located at	is the location of
CapableOf	is capable of	enables
EtymologicallyRelatedTo	is etymologically related to	is etymologically related to
Synonym	is a synonym of	is a synonym of
CreatedBy	is created by	creates
CausesDesire	causes the desire for	is desired because of
Entails	entails	is entailed by
DefinedAs	is defined as	defines
NotHasProperty	does not have the property of	is not a property of

Table 14: Mapping from predefined relations in Neo4j to natural language, including both forward and inverse forms.

Case 1: Handling Hallucination

Input

Williams was **[event2: booked into]** county jail and later released on **[event1: bail]**.

CoT

To determine if there is a causal relationship between Event X (bail) and Event Y (booked into), I will analyze the text provided.

1. The text states that "Williams was booked into county jail and later released on bail."
2. This indicates that the booking into jail (Event Y) occurred first, followed by the release on bail (Event X).
3. The sequence of events suggests that being booked into jail is a prerequisite for being released on bail. Therefore, the booking into jail leads to the possibility of being released on bail.
4. This implies a causal relationship where Event Y (booked into) leads to Event X (bail).

Based on this reasoning, I conclude that there is a causal relationship where Y -> X.

Final answer in JSON format:
{"Answer": "Yes"}

SERE

To determine whether there is a causal relationship between Event X (bail) and Event Y (booked into), I will follow a step-by-step reasoning process:

1. **Identify the Events:**
- Event X: "bail" refers to the process of being released from custody after paying a certain amount of money.
- Event Y: "booked into" refers to the act of being formally admitted into a jail or prison.

2. **Analyze the Context:**
- The text states that Williams was "booked into county jail" and later "released on bail." This indicates a sequence of events where the booking into jail occurs first, followed by the release on bail.

3. **Determine the Nature of the Relationship:**
- The act of being "booked into" jail is a prerequisite for the possibility of being released on bail. In other words, one cannot be released on bail without first being booked into jail.
- However, the act of "bail" itself does not cause the booking into jail; rather, it is a consequence of being booked.

4. **Causal Direction:**
- The causal direction is important here. The booking into jail (Event Y) is what allows for the possibility of bail (Event X). Therefore, while bail is a result of the booking, it does not cause the booking.

5. **Conclusion:**
- Since Event Y (booked into) must occur before Event X (bail) can happen, and Event X does not influence or cause Event Y, there is no causal relationship between the two events.

Based on this reasoning, the final answer is:

{"Answer": "No"}

Case 2: Extracting an Indirect Causal Chain

Input

Woman **[event2: Convicted]** of Double Murder In 2010 Kraft Foods **[event1: Rampage]** In NE Philadelphia.

Pattern Extractor

1. The two events are:
- EVENT X: Rampage
- EVENT Y: Convicted

2. Analyzing the direct causal relationship:
- The text does not explicitly state that the "Rampage" directly caused the "Convicted" event. Instead, it indicates that a woman was convicted of double murder in connection with the rampage. Therefore, there is no direct causal relationship between the two events.

3. Since there is no direct relationship, we need to analyze for indirect causal patterns.

4. Considering mediators:
- The text mentions "double murder" as a significant event related to both the rampage and the conviction. The rampage likely led to the double murder, and the double murder is the basis for the conviction. Thus, "double murder" can be seen as a mediator (Z) between the two events.

5. Analyzing the patterns:
- ****Direct**:** Not applicable as established.
- ****Coreference of X**:** Not applicable; there is no similar event to "Rampage" that has a causal relationship with "Convicted."
- ****Coreference of Y**:** Not applicable; there is no similar event to "Convicted" that has a causal relationship with "Rampage."
- ****Collider**:** Not applicable; while "double murder" is a mediator, it does not satisfy the collider condition since it does not act on both X and Y.
- ****Fork**:** Not applicable; "double murder" does not act on both X and Y.
- ****Chain**:** This pattern is satisfied because the rampage (X) leads to the double murder (Z), and the double murder (Z) leads to the conviction (Y). Thus, we have a causal chain: Rampage -> Double Murder -> Convicted.

Final answer in JSON format:
{"pattern": "Chain"}

Case 3: Representative Positive Example

Input

[event1: Fire] [event2: destroys] historic Mat Maid dairy building in Palmer.

Input Events ConceptNet Path

"fire" is an antonym of "employ", and "employ" is related to "destroy", and "destroy" has the form of "destroys".

Top Supporting Examples

Example 1
Text: Wladimir Klitschko **[event2:won]** his 16th straight **[event1:fight]** today , successfully defending the WBA , WBO , and IBF heavyweight titles against Tony Thompson , stopping the 40 - year - old American in the sixth round of a fight that was never really close .
ConceptNet Path: "fight" is the origin of "prizefight", and "prizefight" is derived from "prize", and "prize" receives the action of "won". (score = 0.426)
Tree similarity score: 0.012
Pattern: Direct

Example 2
Text: The woman who allegedly shot three co - workers — killing two– at the Kraft Plant Thursday night has been **[event2:charged]** with **[event1:murder]** , attempted murder , aggravated assault , burglary , and other related charges , says Philadelphia District Attorney Seth Williams .
ConceptNet Path: "murder" has context in "legal", and "legal" is the context of "uncharged", and "uncharged" is derived from "charged". (score = 0.398)
Tree similarity score: 0.009
Pattern: Direct

Generated Pattern at Inference

1. The two events in natural language are:
   - EVENT X: A fire.
   - EVENT Y: Destroys a historic dairy building.

2. The ConceptNet Path indicates that "fire" is related to "destroy" through the antonym of "employ." The Dependency Tree shows that "Fire" is the subject of the verb "destroys," which directly connects the two events.

3. Analyzing for a direct causal relationship: The text states that a fire destroys a building. This indicates a direct causal relationship where the fire (X) directly causes the destruction (Y). Therefore, the causal pattern is "Direct."

4. Since we have already established a direct causal relationship, we do not need to analyze for indirect causal patterns.

Final answer in JSON format:
{"pattern": "Direct"}

Case 4: Error Analysis Example

Input

After a quick survey of the scene , Owen said , McNutt's main **[event2:concern]** was keeping the **[event1:fire]** from spreading to the heating oil stored next door at the Crowley fuel distribution center .

Input Events ConceptNet Path

"fire" is related to "danger", and "danger" is related to "concern".

Top Supporting Examples

Example 1
Text: Reports on Tuesday morning indicated that a fiber optic cable in the northern part of Egypt had been **[event1:damaged]** , resulting in the **[event2:loss]** of all telecommunications in Egypt's second largest city of Alexandria and parts of the Nile Delta region .
ConceptNet Path: "damaged" is related to "waste", and "waste" is related to "loss". (score = 0.706)
Tree similarity score: 0.004
Pattern: Chain

Example 2
Text: Of the bombs that **[event1:landed]** inside the camp , one **[event2:detonated]** about a quarter mile from the NP compound .
ConceptNet Path: "landed" is related to "property", and "property" is related to "mine", and "mine" is used for "detonate", and "detonate" has the form of "detonated". (score = 0.367)
Tree similarity score: 0.039
Pattern: Chain

Generated Pattern at Inference

1. The two events are:
   - EVENT X: fire
   - EVENT Y: concern

2. The ConceptNet Path indicates that "fire" is related to "danger," and "danger" is related to "concern." This suggests a conceptual link between the two events. The Dependency Tree shows that the concern is explicitly stated as McNutt's main concern regarding the fire, indicating a direct relationship.

3. Analyzing for a direct causal relationship, the text does not explicitly state that the fire causes concern in a direct manner. Instead, it indicates that the concern is about preventing the fire from spreading. Therefore, the causal relationship is not direct.

4. Since there is no direct causal relationship, we need to analyze for indirect causal patterns.

5. The mediators in this context can be identified as "danger" (which is implied by the fire) and "spreading" (the concern is about preventing the fire from spreading). The fire (X) creates a situation of danger, which leads to concern (Y). Thus, we have:
   - X (fire) -> Z (danger)
   - Z (danger) -> Y (concern)

6. Analyzing the patterns:
   - **Direct**: Not satisfied, as there is no explicit direct causal relationship.
   - **Coreference of X**: Not satisfied, as there is no similar event to "fire" that has a causal relationship with "concern."
   - **Coreference of Y**: Not satisfied, as there is no similar event to "concern" that has a causal relationship with "fire."
   - **Collider**: Not satisfied, as there are no mediators that both events relate to in a way that satisfies the collider structure.
   - **Fork**: Not satisfied, as there are no mediators that act on both events in a way that satisfies the fork structure.
   - **Chain**: Satisfied, as we have a mediator (danger) that connects fire to concern through the relationship of X -> Z -> Y.

Final answer in JSON format:
{"pattern": "Chain"}

```
MATCH p = shortestPath((a)-[r*0..{max_path_len}]-(b))
WHERE a.id = $start_id AND b.id = $end_id
RETURN p, [rel IN relationships(p) | {type: type(rel), start: startNode(rel).id, end: endNode(rel).id}] AS r
```

Code 1: Cypher query for Shortest Path Algorithm

Tree Node	Description	Weight
acl	Noun clause modifier (structural, affects nominal structures)	2
acomp	Adjectival complement (supplementary description, does not affect core structure)	1
advcl	Adverbial clause modifier (enhances overall sentence modification)	3
advmod	Adverbial modifier (modifies verbs, moderate weight)	2
agent	Agent (critical, affects subject-verb structure)	5
appos	Appositive modifier (enhances sentence expression, affects structure)	3
attr	Attribute modifier (supplementary explanation, affects description)	1
cc	Coordinating conjunction (connects sentence structures, moderate weight)	2
ccomp	Complement clause (structural, complement clause has some impact)	3
compound	Compound modifier (enhances sentence structure expression, moderate weight)	3
conj	Coordinating word (moderate weight, used in coordinate structures)	2
csubj	Subject of a clause (structural, affects core sentence grammar)	5
csubjpass	Subject of a passive clause (structural, subject role within the clause)	5
det	Determiner (modifies nouns but does not change core structure)	1
dobj	Direct object (core component of the sentence, affects structure)	4
neg	Negation modifier (negates sentence meaning, has some impact on structure)	2
nounmod	Noun modifier (modifies nouns, affects sentence structure)	2
npmod	Noun phrase as adverbial modifier (contributes to grammatical structure)	2
nsubj	Noun subject (one of the main components of the sentence, core element)	5
nsubjpass	Passive noun subject (core element, affects structure)	5
nummod	Numeral modifier (modifies numerals, minimal structural impact)	1
oprd	Object predicate (affects grammatical structure, part of the core)	3
parataxis	Parataxis (affects structure connection, moderate weight)	2
pcomp	Prepositional phrase complement (affects structure, but not the main clause core)	3
pobj	Prepositional object (important component of sentence structure)	4
poss	Possessive modifier (minimal impact, part of modification)	1
preconj	Preceding conjunction (connects sentences, affects structure)	1
predet	Preceding determiner (modifies nouns, minimal structural impact)	1
prep	Prepositional modifier (structural connection within the sentence, significant impact)	3
prt	Particle (minimal structural impact)	1
quantmod	Quantifier modifier (modifies quantifiers, minimal structural impact)	1
relcl	Relative clause modifier (has a significant impact on sentence structure)	3
ROOT	Sentence root node (core structure, most important)	5
xcomp	Open complement clause (affects sentence structure)	3

Table 15: The dependency syntax tree nodes used in this paper and their corresponding weights. For nodes not listed in the table, we set their weight to 0.

Prompt 1: Base Method Prompt

Given a text and two events (Event X and Event Y), please determine whether there is a causal relationship between Event X and Event Y (i.e., whether X -> Y or Y -> X).
- Instructions:
1. The final judgement you give must be either 'Yes' or 'No', and nothing else.
2. You need to organize the final answer in JSON format: {"Answer": "Your answer, the answer must be either 'Yes' or 'No', and nothing else."}.
Text: {input_text};
Event X: {source};
Event Y: {target}.
Your answer in JSON format {"Answer": "Your answer, the answer must be either 'Yes' or 'No', and nothing else."}:

Prompt 2: CoT Method Prompt

Given a text and two events (Event X and Event Y), please determine whether there is a causal relationship between Event X and Event Y.
Text: {input_text};
Event X: {source};
Event Y: {target}.
Give step-by-step reasoning step and then give your answer in JSON format {"Answer": "Your answer, the answer must be either 'Yes' or 'No', and nothing else."}:

Prompt 3: Causal Pattern Extraction Prompt for Positive Sample

TEXT: {text}, **EVENT X:** {source}, **EVENT Y:** {target}.
QUESTION: There is a causal relationship between EVENT X and EVENT Y. Please follow the following instructions to explain why there exists a causal relationship between X and Y based on the given text.
NOTE: You should put the final answer in JSON format: {"pattern": THE CAUSAL PATTERN YOU FIND. DO NOT ANSWER "None" or "No".}
- Instructions:
1. Analyze and determine whether X and Y have direct causal relationship, and meet the causal pattern rule "Direct". If so, answer causal pattern as "Direct"; If not, continue to analyze.
2. Determine which indirect causal pattern given below the given input and events satisfy. Note: If X and Y have the indirect causal relationship, they must satisfy to one of the following patterns.
3. Consider whether there are mediators between events X and Y: write down other events (or entities) that relates to X, and other events (or entities) that relates to Y, and determine whether there is any intersection between the events (or entities) that relate to both events. Note: Mediators can be given explicitly from the input text. If not given, you can also use common sense to think about whether there are implicit mediators.
4. Finally, analyze all the following patterns ONE-BY-ONE to determine whether the given text and events satisfy. DO NOT answer "None" or "No".
- Pattern Rules:
Direct: If the text explicitly states a causal relationship between X and Y without involving any mediating event (Z), then the causal connection is "Direct". This means that X directly influences Y, or Y directly influences X, with no intermediary mentioned.
Coreference of X: In the text, if an event with the same or similar meaning as the X can be found, and this similar event has a causal relationship with Y, then there is a causal relationship between X and Y.
Coreference of Y: In the text, if an event with the same or similar meaning as the Y can be found, and this similar event has a causal relationship with X, then there is a causal relationship between X and Y.
Collider: In the text, If there are one or multiple mediators (Z) that both events X and Y have causal relationship to: Then consider specific rule: First, it satisfies that it is related to X and Y respectively, then it satisfies that X acts on Z and Y acts on Z, i.e. X -> Z and Y -> Z. Therefore, it can be concluded that there is a causal relationship between X and Y.
Fork: In text, if there are one or multiple mediators (Z) that both events X and Y have causal relationship to: Then consider specific rule: First, it satisfies that it is related to X and Y respectively, then it satisfies that Z acts on X and Z acts on Y, i.e. Z -> X and Z -> Y. Therefore, it can be concluded that there is a causal relationship between X and Y.
Chain: In the text, if there are at least one or multiple mediators (Z) that both events X and Y have causal relationship to: Then consider specific rules: First, a mediator satisfies that it is related to X and Y respectively, then it satisfies that X acts on Z and Z acts on Y, i.e. they form a causal chain structure: X -> Z -> Y (or inversely, Y -> Z -> X). Then, it can be concluded that there is a causal relationship between X and Y.
NOTE: If one pattern rule is met, you DON'T need to analyze the remaining rules.
EXAMPLE 1:
TEXT: The factory's decision to shut down immediately resulted in hundreds of workers losing their jobs.
X: shut down; *Y*: losing
PATTERN: Direct (The text clearly indicates a direct causal link between "factory's decision to shut down" and "workers losing their jobs.")
EXAMPLE 2:
TEXT: The company was accused of negligence in maintaining its pipelines, which were found to be leaking crude oil into the river. The oil spill caused significant harm to the local ecosystem.
X: negligence in maintaining its pipelines; *Y*: Harm
PATTERN: Coreference ("The company was accused of negligence in maintaining its pipelines" and "pipelines leaking crude oil into the river" describe the same event in different ways. "Pipelines leaking crude oil" caused "harm to the ecosystem".)
EXAMPLE 3:
TEXT: A major tech company introduced aggressive hiring policies, while a spike in tech startups also attracted talent to the industry. The resulting competition for skilled workers drove up average salaries in the tech sector.
X: aggressive hiring policies; *Y*: spike in tech startups; *Z*: Competition for skilled workers.
PATTERN: Collider ("Aggressive hiring policies" and "spike in tech startups" both increase competition for skilled workers (X -> Z, Y -> Z), which in turn drives up salaries, indirectly linking X and Y.)
EXAMPLE 4:
TEXT: A global economic slowdown led to a decline in consumer spending and a rise in unemployment rates, as businesses struggled to stay profitable.
X: decline in consumer spending; *Y*: rise in unemployment rates
PATTERN: Fork ("Global economic slowdown" causes both "a decline in consumer spending" (Z -> X) and "a rise in unemployment rates" (Z -> Y). This forms a Fork structure linking X and Y via Z.)
EXAMPLE 5:
TEXT: Heavy deforestation in the region caused soil erosion, which eventually led to a decline in agricultural productivity.
X: deforestation; *Y*: decline; *Z*: soil erosion
PATTERN: Chain ("Heavy deforestation" leads to "soil erosion" (X -> Z), and "soil erosion" causes "a decline in agricultural productivity" (Z -> Y). This forms a causal chain)
Your answer:

Prompt 4: Causal Pattern Extraction Prompt for Inference Sample

TEXT: {text}, **EVENT X:** {source}, **EVENT Y:** {target}.
QUESTION: Determine whether there is a causal relationship between EVENT X and EVENT Y.
NOTE: You should put the final answer in JSON format: {"pattern": THE CAUSAL PATTERN YOU FIND. IF NO PATTERN RULES ARE MET, GIVE "No".}
- Instructions:

1. Analyze and determine whether X and Y have direct causal relationship, and meet the causal pattern rule "Direct". If so, answer causal pattern as "Direct"; If not, continue to analyze.

2. Determine which indirect causal pattern given below the given input and events satisfy. Note: If X and Y have the indirect causal relationship, they must satisfy to one of the following patterns.

3. Consider whether there are mediators between events X and Y: write down other events (or entities) that relates to X, and other events (or entities) that relates to Y, and determine whether there is any intersection between the events (or entities) that relate to both events. Note: Mediators can be given explicitly from the input text. If not given, you can also use common sense to think about whether there are implicit mediators.

4. Finally, analyze all the following patterns ONE-BY-ONE to determine whether the given text and events satisfy. If no pattern rules are met, give "No"

- Pattern Rules:

Direct: If the text explicitly states a causal relationship between X and Y without involving any mediating event (Z), then the causal connection is "Direct". This means that X directly influences Y, or Y directly influences X, with no intermediary mentioned.

Coreference of X: In the text, if an event with the same or similar meaning as the X can be found, and this similar event has a causal relationship with Y, then there is a causal relationship between X and Y.

Coreference of Y: In the text, if an event with the same or similar meaning as the Y can be found, and this similar event has a causal relationship with X, then there is a causal relationship between X and Y.

Collider: In the text, If there are one or multiple mediators (Z) that both events X and Y have causal relationship to: Then consider specific rule: First, it satisfies that it is related to X and Y respectively, then it satisfies that X acts on Z and Y acts on Z, i.e. X -> Z and Y -> Z. Therefore, it can be concluded that there is a causal relationship between X and Y.

Fork: In text, if there are one or multiple mediators (Z) that both events X and Y have causal relationship to: Then consider specific rule: First, it satisfies that it is related to X and Y respectively, then it satisfies that Z acts on X and Z acts on Y, i.e. Z -> X and Z -> Y. Therefore, it can be concluded that there is a causal relationship between X and Y.

Chain: In the text, if there are at least one or multiple mediators (Z) that both events X and Y have causal relationship to: Then consider specific rules: First, a mediator satisfies that it is related to X and Y respectively, then it satisfies that X acts on Z and Z acts on Y, i.e. they form a causal chain structure: X -> Z -> Y (or inversely, Y -> Z -> X). Then, it can be concluded that there is a causal relationship between X and Y.

NOTE: If one pattern rule is met, you DON'T need to analyze the remaining rules.

EXAMPLE 1:
TEXT: The factory's decision to shut down immediately resulted in hundreds of workers losing their jobs.
X: shut down; *Y*: losing
PATTERN: Direct (The text clearly indicates a direct causal link between "factory's decision to shut down" and "workers losing their jobs.")

EXAMPLE 2:
TEXT: The company was accused of negligence in maintaining its pipelines, which were found to be leaking crude oil into the river. The oil spill caused significant harm to the local ecosystem.
X: negligence in maintaining its pipelines; *Y*: Harm
PATTERN: Coreference ("The company was accused of negligence in maintaining its pipelines" and "pipelines leaking crude oil into the river" describe the same event in different ways. "Pipelines leaking crude oil" caused "harm to the ecosystem".)

EXAMPLE 3:
TEXT: A major tech company introduced aggressive hiring policies, while a spike in tech startups also attracted talent to the industry. The resulting competition for skilled workers drove up average salaries in the tech sector.
X: aggressive hiring policies; *Y*: spike in tech startups; *Z*: Competition for skilled workers.
PATTERN: Collider ("Aggressive hiring policies" and "spike in tech startups" both increase competition for skilled workers (X -> Z, Y -> Z), which in turn drives up salaries, indirectly linking X and Y.)

EXAMPLE 4:
TEXT: A global economic slowdown led to a decline in consumer spending and a rise in unemployment rates, as businesses struggled to stay profitable.
X: decline in consumer spending; *Y*: rise in unemployment rates
PATTERN: Fork ("Global economic slowdown" causes both "a decline in consumer spending" (Z -> X) and "a rise in unemployment rates" (Z -> Y). This forms a Fork structure linking X and Y via Z.)

EXAMPLE 5:
TEXT: Heavy deforestation in the region caused soil erosion, which eventually led to a decline in agricultural productivity.
X: deforestation; *Y*: decline; *Z*: soil erosion
PATTERN: Chain ("Heavy deforestation" leads to "soil erosion" (X -> Z), and "soil erosion" causes "a decline in agricultural productivity" (Z -> Y). This forms a causal chain)

Your answer:

```
def inference_by_examples_prompt(text: str, source: str, target: str, examples: list[dict[str, Any]]) -> str:
    instruction_prompt = '''Given a text, two events (Event X and Event Y). Based on the related examples, you
        need to determine whether there is a causal relationship between the given events X and Y. Please follow
        the instructions below and refer to the provided examples when answering.
###
Instructions:
You should refer to the examples but not be entirely influenced by them. Whether the events in the examples have
    a causal relationship DOES NOT affect whether the given events in the provided text have a causal
    relationship.
You should give step-by-step reasoning path before giving the final answer.
'''

    example_prompt = '''***Example {idx}***
Text: {text};
Event X: {source};
Event Y: {target};
Answer: {{"Answer": "{answer}"}}
'''

    examples_prompt = ''
    for idx, e in enumerate(examples, start=1):
        examples_prompt += example_prompt.format(idx=idx,
                                                 text=e['input_text'],
                                                 source=e['source'],
                                                 target=e['target'],
                                                 answer='Yes' if e['ground'] == 1 else 'No') + '\n'

    prompt = '''{instruction_prompt}
###
Here are some examples.
{examples_prompt}
###
Text: {text};
Event X: {source};
Event Y: {target};
Give step-by-step reasoning path, and then organize the final answer in JSON format: {{"Answer": "Your answer,
    the answer must be either 'Yes' or 'No', and nothing else."}}
Your response:
'''

    return prompt.format(instruction_prompt=instruction_prompt.format(text=text, source=source, target=target),
                         examples_prompt=examples_prompt,
                         text=text,
                         source=source,
                         target=target)
```

Code 2: SERE final inference prompt