COLA: Contextualized Commonsense Causal Reasoning from the Causal Inference Perspective

Zhaowei Wang[1], **Quyet V. Do**[1], **Hongming Zhang**[2], **Jiayao Zhang**[3], **Weiqi Wang**[1], **Tianqing Fang**[1], **Yangqiu Song**[1], **Ginny Y. Wong**[4], **& Simon See**[4]

[1]Department of Computer Science and Engineering, HKUST

[2]Tencent AI Lab, Bellevue, USA [3]University of Pennsylvania

[4]NVIDIA AI Technology Center (NVAITC), NVIDIA, Santa Clara, USA

{zwanggy,yqsong}@cse.ust.hk, {gwong,ssee}@nvidia.com

Abstract

Detecting commonsense causal relations (causation) between events has long been an essential yet challenging task. Given that events are complicated, an event may have different causes under various contexts. Thus, exploiting context plays an essential role in detecting causal relations. Meanwhile, previous works about commonsense causation only consider two events and ignore their context, simplifying the task formulation. This paper proposes a new task to detect commonsense causation between two events in an event sequence (i.e., context), called contextualized commonsense causal reasoning. We also design a zero-shot framework: COLA (**C**ontextualized C**o**mmonsense Causa**l**ity Re**a**soner) to solve the task from the causal inference perspective. This framework obtains rich incidental supervision from temporality and balances covariates from multiple timestamps to remove confounding effects. Our extensive experiments show that COLA[1] can detect commonsense causality more accurately than baselines.

1 Introduction

Commonsense Causal Reasoning (CCR) aims at identifying plausible causes and effects of events in natural language that are typically reasonable by an average person (Zhang et al., 2022c). To solve the task, existing efforts devoted by the community mainly rely on language models wholeheartedly with supervised learning approaches (Staliūnaitė et al., 2021; Sap et al., 2019; Tamborrino et al., 2020; He et al., 2020; Raffel et al., 2020). Those ingenious engineering works have brought significant progress in recent years. However, recent studies (Kavumba et al., 2019; Han and Wang, 2021) found that pure engineering designs are inadequate to seize commonsense causation, as language models tend to reach higher scores by exploiting superficial artifacts in data.

[1]The code and data are available at https://github.com/HKUST-KnowComp/COLA.

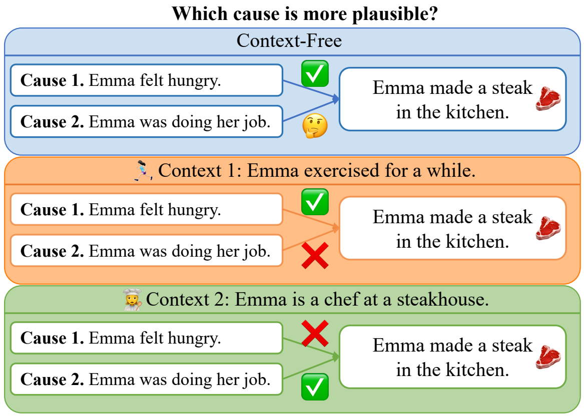

Figure 1: An illustration of leveraging context to conduct commonsense causal reasoning. Both causes could be plausible under different contexts, while only the frequent one (i.e., **Cause 1**) is plausible without context.

Recently, Zhang et al. (2022c) first studied grasping commonsense causation from the causal inference perspective, by drawing analogies between observational studies (Cochran and Chambers, 1965; Rosenbaum, 2002) and natural languages (Zhang and Zhang, 2021). The proposed framework ROCK achieves good potential for the zero-shot CCR task (e.g., COPA by Gordon et al. (2012)). However, Zhang et al. (2022c) only focuses on the commonsense causation between a pair of events without specifying context. Given that events are complex (Chen et al., 2021), an event may have different causes under different contexts (Mostafazadeh et al., 2020). Thus, it is necessary to utilize context when detecting commonsense causation, such as other events related to given ones. Missing a clear and specific context simplifies commonsense causal knowledge and hinders models from detecting commonsense causal relations more accurately. For example, as shown in Figure 1, the frequent cause of "Emma made a steak in the kitchen." is "Emma felt hungry." However, the cause also could

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers, pages 5253–5271
July 9-14, 2023 ©2023 Association for Computational Linguistics

be "Emma was doing her job" if "Emma is a chef at a steakhouse." Without the context of Emma's job, models cannot distinguish those two causes and may return to the frequent one.

To involve context when detecting commonsense causation, we propose a new task to detect causes between two events in an event sequence, called Contextualized Commonsense Causal Reasoning (Contextualized CCR). In this task, models are asked to detect commonsense causal relations between two given events enclosed in an event sequence. Other events in the event sequence can provide a clear and specific definition of the current context, helping models to capture commonsense causation more accurately. In fact, we find that contextualized CCR is a non-trivial task. Directly applying the framework ROCK (Zhang et al., 2022c) cannot achieve competitive performance on the contextualized CCR since it cannot integrate context information.

We propose the framework **COLA**, which incorporates contextual information from an event sequence, to solve the Contextualized CCR. Our framework adopts the potential-outcomes framework (Rubin, 1974; Rosenbaum, 2002; Rubin, 2005) to estimate the causal estimand Δ defined as a type of "*average treatment effect*" (ATE), which measures the change in the likelihood of E_j's occurrence when intervening E_i (denoted by $\neg\mathsf{E}_i$) as

$$\Delta = \mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_j) - \mathbb{P}(\neg\mathsf{E}_i \prec \mathsf{E}_j), \tag{1}$$

where $\mathbb{P}(\cdot)$ can be estimated with a pre-trained language model, such as a masked language model (Devlin et al., 2018). The magnitude of *average treatment effect* informs the strength of E_i's effect on E_j, and its sign indicates the direction of the effect. For instance, $\Delta \approx 1$ means E_j becomes more prone to occur due to the occurrences of E_i. In an ideal world (e.g., E_i and E_j on any study unit occur completely randomly), a plugging-in estimator in Equation (1) suffices for detecting commonsense causation. Nevertheless, spurious correlations introduced by pervasive confounding co-occurrences need to be eliminated for an unbiased estimation of the causal estimand. This can be done by *balancing* events that precede E_i, or *covariates*. To incorporate context, we design a mechanism to sample diversified covariates from multiple timestamps and use *temporal propensity* (Zhang et al., 2022c) for balancing.

We annotated commonsense causal relations between two events (~1.3k examples) within event sequences from ROCStories (Mostafazadeh et al., 2016) to benchmark our proposed contextualized CCR task. We conduct extensive experiments with multiple pre-trained language models, showing that **COLA** can detect cause-and-effect relations more accurately than competitive baselines by a large margin. Our experiments also show that temporality is essential in our framework but not sufficient to detect commonsense causation without covariates being appropriately balanced.

2 Background and Related Works

Understanding events and relations between them have long been a challenging NLP task (Chen et al., 2021). The community has dedicated many works to studying various event-centric tasks, including event relation reasoning (Ning et al., 2018; Zhou et al., 2021; Wang et al., 2020), event extraction (Huang et al., 2018; Lai et al., 2020; Zhang et al., 2022b; Lin et al., 2023), event-centric KG construction (Zhang et al., 2020b, 2022a), and many others (Chambers and Jurafsky, 2008; Chen et al., 2020; Jin et al., 2022; Wang et al., 2022b). Among them, there are a few lines of work that are most related to our work:

Commonsense Causal Reasoning Since our work is about Contextualized CCR, we first discuss related works about commonsense causal reasoning. Existing commonsense causal reasoning approaches are typically categorized under the general topic of commonsense reasoning (Rashkin et al., 2018; Sap et al., 2020). Most previous works depend on language models. Remarkable progress in CCR mainly comes from dataset augmentation, training procedure design, and external knowledge (Staliūnaitė et al., 2021; Sap et al., 2019; Shwartz et al., 2020; Tamborrino et al., 2020; Iter et al., 2020). Studies (Kavumba et al., 2019; Han and Wang, 2021) show that language models exploit superficial artifacts to achieve suspicious high performance.

Causal event detection (Mirza and Tonelli, 2014; Mirza et al., 2014) forms another line of work pertinent to CCR. The task aims to detect causal relations in documents, where various methods are proposed (Chang and Choi, 2005; Do et al., 2011; Ning et al., 2019). However, those works consider verbal (e.g., "attack") or nominal predicates (e.g., "explosion") as events, oversimplifying the relation detection task. In this paper, we study events expressed in free text, facing a more challenging setup but being closer to real applications.

Narrative-related Tasks Since Contextualized CCR is primarily about chains of events, our work is inspired by earlier research that deals with narratives and scripts (Chambers and Jurafsky, 2008; Granroth-Wilding and Clark, 2016; Mostafazadeh et al., 2016; Bhagavatula et al., 2019; Zhang et al., 2020a). In contrast, our work aims to identify causal relations in a chain of events.

Methodologies of Causal Inference Zhang and Zhang (2021) provided the first study to solve the CCR task from the causal inference perspective. The human population studies have scrutinized extensively the causal inference, which identifies causal relations from ubiquitous associations, including biomedical research, agriculture, epidemiology, and economics (Fisher, 1958; Imbens and Rubin, 2015; Giannarakis et al., 2022; Rosenbaum, 2002; Cochran and Chambers, 1965), where researchers usually use the potential-outcomes framework (Splawa-Neyman et al., 1990; Rubin, 1974; Holland, 1986), graphical and structural equation models (Robins, 1986; Pearl, 1995; Heckman, 2005), and Granger causality (Granger, 1969).

Recent studies have drawn causal inferences on textual data with the help of powerful pre-trained language models (Kang et al., 2017; Keith et al., 2020; Feder et al., 2022). Concurrently, causal inference can improve the robustness and fairness of NLP models (Feder et al., 2022) or boost performance on downstream tasks (Ghosal et al., 2021; Zheng et al., 2022; Alabdulkarim et al., 2021; Wang et al., 2022a).

3 Problem Formulation

Notation We use $\mathsf{sans-serif}$ fonts to represent an event, such as E_i in Figure 2, where the subscript i means the i-th event in a sequence. Simultaneously, uppercase serif letters denote *indicators* of whether the corresponding event occurs: $E_i = \mathbb{1}\{\mathsf{E}_i \text{ occurs}\}$, and a lowercase serif letter means the realizations of this indicator: $e_{i,j} = \mathbb{1}\{\mathsf{E}_i \text{ occurs to the } j\text{-th study unit}\}$. We introduce the point process to more clearly describe the order of **a pair of** events: $E(t)$ with $t \in \{0, 1\}$ (e.g., past versus present), so that $E_i(0)$ and $E_j(1)$ means that E_i happens before E_j. We also use $\mathsf{E}_i \prec \mathsf{E}_j$ to indicate that E_i occurs before E_j for simplicity. We write $\mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_j) = \mathbb{P}(E_i(0), E_j(1))$.

Task Description We articulate the Contextualized CCR as a tweaked form of the binary classification problem. Specifically, we provide models with an event sequence of n events: $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_n$. Models need to find the top-k events in this sequence that more plausibly have commonsense causal relations with the last event E_n, where k indicates the number of positive labels in the ground truth. Then, models need to predict the commonsense causal relation between each event pair (E_i, E_n) as a positive/negative one. The strength of causal relations between E_i and E_n can be expressed with *average treatment effect* as:

$$\Delta_i = \mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_n) - \mathbb{P}(\neg\mathsf{E}_i \prec \mathsf{E}_n). \qquad (2)$$

4 Theoretical Mechanism of COLA

As discussed in Section 1, we articulate the Contextualized CCR problem as the estimation of the causal estimand Δ, which we model as the change of temporal likelihood *with contexts controlled*. We adopt the potential-outcomes framework to design **COLA** to eliminate potential confounding effects due to co-occurrences of events when estimating the causal estimand from data. In this section, we first clarify all essential concepts in the theoretical mechanism of **COLA** one by one, including *study unit*, *intervention*, *covariate*, and *propensity*, by drawing analogies between the underlying causal mechanisms in natural languages with that in human population research. We then describe the implementation of each component in Section 5.

4.1 The Analogy and Study Unit

Motivated by Zhang et al. (2022c), we draw the analogy between human subjects and semantic meanings through the following process: assuming that every human subject kept a textbook recording each event (s)he has experienced, we can then treat each textbook (in natural language) as a study unit and infer the temporal relationships between events from it. In this analogy, we clearly understand the study unit in semantic meanings.

Then, we can formulate contextualized CCR with concepts from the potential-outcome framework. Given two events E_i and E_n from an event sequence $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_n$, where we assume that E_{ij} represents the event that the j-th study unit experienced at the timestamp i when E_i is supposed to occur. Then for each unit j, we can define the treatment assignment as $E_{ij} = \mathbb{1}\{\mathsf{E}_{ij} = \mathsf{E}_i\}$, realizations of covariates as $\boldsymbol{x}_j = (x_{jl})_{l=1}^N$ for

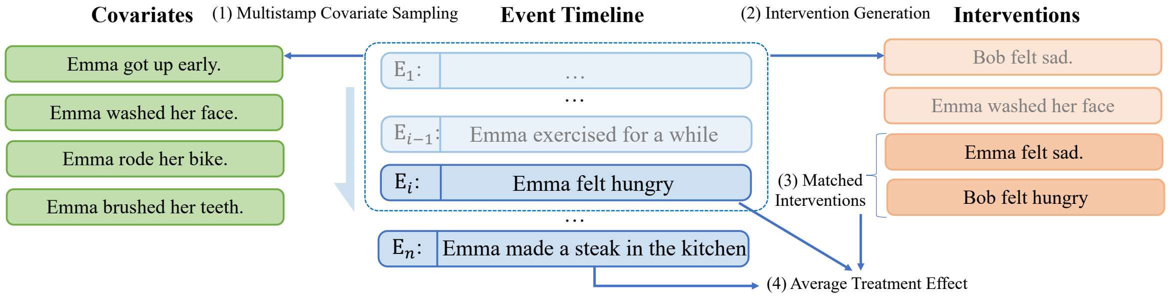

Figure 2: The overview of our proposed framework **COLA**. Given a pair of events $(\mathsf{E}_i, \mathsf{E}_n)$, the framework contains four steps to detect commonsense causal relations: (1) multistamp covariate sampling, (2) intervention generation (3) selecting a set of matched interventions, (4) computing *average treatment effect*.

$x_{jl} = \mathbb{1}\{\mathsf{X}_l \prec \mathsf{E}_{ij}\}$, and two potential-outcomes as

$$\begin{cases} r_{0,ij} = \mathbb{1}\{\mathsf{E}_{ij,E_i=0} \prec \mathsf{E}_n\}, \\ r_{1,ij} = \mathbb{1}\{\mathsf{E}_{ij,E_i=1} \prec \mathsf{E}_n\}. \end{cases} \tag{3}$$

When the j-th unit receives the treatment assignment E_{ij}, the hypothetical scenario is denoted by $\mathsf{E}_{ij,E_i=1-E_{ij}}$, which describes what if the assignment were flipped. Clearly, we can only observe either $r_{0,ij}$ or $r_{1,ij}$, but not both of them. We can rewrite the causal estimand Δ_i in Equation (2) exactly as an *average treatment effect* by averaging over the unit index:

$$\begin{aligned} \Delta_i &= \mathbb{E}_j[r_{1,ij} - r_{0,ij}] \\ &\equiv \mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_n) - \mathbb{P}(\neg\mathsf{E}_i \prec \mathsf{E}_n). \end{aligned} \tag{4}$$

The above formulation naturally embodies the temporal nature of covariates (Rubin, 2005), which, by definition, are pretreatments that precede treatments.

4.2 Intervention Beyond Negation

In human population studies, the generally accepted *stable unit treatment value assumption* (Rubin, 1980) ensures only one type of non-treatment (usually negation) for each study unit. As events are complicated, we would interpret intervention (manipulation) of semantic meanings in a broader sense. Take E_i "Emma felt hungry" from Figure 2 as an example. While "Emma didn't feel hungry" is the most direct intervention, it is nonetheless too restrictive: Emma may have felt happy; maybe Alice is the one who felt hungry, instead of Emma. Consequently, interventions in our framework are interpreted much broader as any event that could result in a plausible counterfactual of the outcome. We use $\mathcal{A}$ to represent all possible interventions of an event E_i.

4.3 Balancing Covariates and Comparable Study Units

We have discussed that the plugging-in estimator in Equation (1) suffers from biases due to potential confounders. One mitigation strategy is properly balancing the covariates (Rubin, 2005), namely events that occur before E_i, which ensures that covariates of untreated study units are comparable to those of treated ones. Consider the vaccine trial as an example; one needs to ensure that the health conditions (covariates) in the control group (not administered vaccines) are comparable to the treated group (administered vaccines). As such, we rewrite the causal estimand Δ in Equation (2) as expectations conditional on the covariates $\boldsymbol{x}$ among comparable study units:

$$\mathbb{E}_{\boldsymbol{x}}\left[\mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_n|\boldsymbol{x}) - \mathbb{P}(\neg\mathsf{E}_i \prec \mathsf{E}_n|\boldsymbol{x})\right], \tag{5}$$

provided that the treatment assignment is strongly ignorable with respect to potential outcomes (i.e., $r_{1,ij}$ and $r_{0,ij}$) according to the strong ignorability assumption.

The strong ignorability assumption is essential in causal inference. Simply, it means that given a set of covariates, the treatment assignment among study units can be viewed as "random" (or "ignorable") with respect to potential outcomes (see, e.g., Rosenbaum (2002); Rubin (2005) for textbook treatments and Zhang et al. (2022c) for more discussions on this assumption in CCR).

4.4 Matching Temporal Propensities

Directly estimating Equation (5) may face the issue of data sparsity since we may sample multiple covariates, and combinations of their values grow exponentially. There are various methods to mitigate this issue in balancing covariates, including assignment modeling, outcome modeling, and doublyrobust estimations (Rubin, 2005), among which we base our method on propensity score matching. It is a simple and effective method that is widely used in observational studies (Rosenbaum and Rubin, 1983), which matches the propensity scores of study units to balance covariates. The propensity score is defined as $p(\boldsymbol{x}) = \mathbb{P}(E_i(1) = 1|\boldsymbol{x}(0))$, which represents the probability of E_i taking place conditioning on covariates $\boldsymbol{x}$. Since it is unclear how to pack an unordered set of covariates (events) into a sequential input, Zhang et al. (2022c) proposed to use a relaxed notion of temporal propensity vector, defined as the vector of probabilities of E_i happening conditional on each covariate $x \in \boldsymbol{x}$:

$$q(\boldsymbol{x}) = q(\boldsymbol{x}; \mathsf{E}_i) = \left(\mathbb{P}(E_i(1) = 1|x(0))\right)_{x \in \boldsymbol{x}}. \tag{6}$$

Hence, we can rewrite the conditional expectation in Equation (5) in the form of matching temporal propensity vectors for some fixed threshold ϵ, given below:

$$\begin{cases} \hat{\Delta}_i = f(\mathsf{E}_i, \mathsf{E}_n) - \frac{1}{|\mathcal{A}'|} \sum_{\mathsf{A} \in \mathcal{A}'} f(\mathsf{A}, \mathsf{E}_n), \\ \mathcal{A}' := \left\{ \mathsf{A} \in \mathcal{A} : \frac{1}{|\mathcal{X}|} \|q(\boldsymbol{x}; \mathsf{A}) - q(\boldsymbol{x}; \mathsf{E}_i)\|_2 \leq \epsilon \right\}, \end{cases} \tag{7}$$

where $f(\mathsf{E}_i, \mathsf{E}_n)$ is an estimate for $\mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_n)$ produced by a language model.

5 The COLA Framework

After establishing the theoretical mechanism for our framework **COLA**, we describe the implementation of each component of **COLA** in this section. Generally, since events are in free-text form in our task, pre-trained language models play a central role in our framework. Given that LMs are pre-trained on an enormous amount of textual data (Gao et al., 2020; Raffel et al., 2020), it is sensible to suppose that those LMs would emulate the responses of an average reasonable person.

Specifically, our framework **COLA** takes two events E_i and E_n from a sequence $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_n$ as input. As shown in Figure 2, our framework **COLA** contains four steps: (1) a multistamp covariate sampler samples a set $\mathcal{X}$ of covariates. (2) an intervention generator generates a set $\mathcal{A}$ of interventions. (3) A score estimator builds temporal propensity vectors and selects a matched subset $\mathcal{A}'$ out of $\mathcal{A}$, by estimating the temporality with a temporal predictor. (4) Eventually, the same score estimator computes $\hat{\Delta}_i$ according to Equation (7).

Multistamp Covariate Sampler Our multistamp covariate sampler is based on GPT-J (6b) (Wang and Komatsuzaki, 2021). For an input event E, we add "Before that," at the end to build "E Before that," as the prompt template. For the i-th event in a sequence, we first sample a covariate set $\mathcal{X}_i$, which contains events before E_i. To diversify covariates, we also sample events before $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_{i-1}$ separately, forming $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{i-1}$. Those events are also before E_i and can serve as covariates due to the transitivity of the temporal relation[2]. We evenly merge covariates before each timestamp to construct the final covariate set:

$$\mathcal{X} = \cup_{l=1}^{i} \mathcal{X}_l. \tag{8}$$

Union vs. Intersection While the aforementioned method takes the union of events sampled according to the left-side context of E_i, another intuitive approach is to take the intersection of events sampled according to multiple timestamps in the right-side context. In this way, we can collect covariates that happen before all of $\mathsf{E}_i, \mathsf{E}_{i+1}, \ldots, \mathsf{E}_n$, that is, $\mathcal{X} = \cap_{l=i}^{n} \mathcal{X}_l$. We discuss the experimental results of these two methods in Section 7.3 and found taking ***union*** works better, which will be the default in the remaining sections.

Intervention Generator This component generates a set $\mathcal{A}$ of interventions as discussed in Section 4.2. There are a variety of related works about generating interventions (counterfactuals) of an event (Gardner et al., 2020; Qin et al., 2019; Ribeiro et al., 2020) and we choose `PolyJuice` (Wu et al., 2021) in our framework owing to its non-task-specific training objective. `PolyJuice` generates interventions by masking some phrases individually and filling in masks with a fine-tuned GPT2. Then, we apply the semantic role labeling (SRL) tool provided by AllenNLP (Gardner et al., 2018) to extract the verb `V` and two arguments `ARG0` and `ARG1` as phrases to be manipulated (see Appendix A.2 for more details).

Temporal Predictor We prompt a masked language model to estimate the temporal relation scores between two given events E_i and E_n. The prompt template "E_i `<MASK>` E_n" predicts scores $f_b(\mathsf{E}_i, \mathsf{E}_n)$ and $f_a(\mathsf{E}_i, \mathsf{E}_n)$ for the output tokens `before` and `after`. Similarly, we can obtain a reversed estimation by inputting "E_n `<MASK>` E_i." Final temporal score f averages scores from

[2] In a sequence of temporally ordered events if $\mathsf{A} \prec \mathsf{B}$ and $\mathsf{B} \prec \mathsf{C}$, then $\mathsf{A} \prec \mathsf{C}$.

both directions: $f(\mathsf{E}_i, \mathsf{E}_n) = \frac{1}{2}(f_b(\mathsf{E}_i, \mathsf{E}_n) + f_a(\mathsf{E}_n, \mathsf{E}_i))$

Our temporal predictor needs to be fine-tuned on a temporal relation corpus. Directly applying a pre-trained LM encounter the problem of low coverage, where the tokens `before` and `after` cannot be found in the top-k prompted tokens (even $k = 30$). Thus, we fine-tuned masked language models to predict the masked connectives in a prompt learning setting. Intuitively, temporal relations exist between each pair of adjacent events in a chronologically ordered event sequence. Assuming an event sequence contains two adjacent events $\mathsf{E}_i, \mathsf{E}_{i+1}$, we then can create an example E_i `before` E_{i+1} and an symmetric example E_{i+1} `after` E_i. We also construct negative samples by replacing E_i or E_{i+1} with a randomly sampled event from other sequences. Those negative examples can teach models when no temporal relation exists. While we mask `before` or `after` in a positive example for models to predict, a special token `[none]` should be prompted for negative examples. Then, the cross-entropy loss is used to optimize the temporal predictor. We call this fine-tuning process ***temporal fine-tuning***. More details about the temporal relation dataset and fine-tuning process are shown in Appendix A.1.

Score Estimator With the temporal predictor, we can estimate $\mathbb{P}(\mathsf{X} \prec \mathsf{A})$ for all covariates $\mathsf{X} \in \mathcal{X}$ and interventions $\mathsf{A} \in \mathcal{A}$: $\mathbb{P}(X(0), A(1)) = \mathbb{P}(\mathsf{X} \prec \mathsf{A}) = f(\mathsf{X}, \mathsf{A})$. We also need to estimate $\mathbb{P}(X(0))$ to compute conditional probabilities $\mathbb{P}(A(1) = 1|X(0))$ in temporal propensity vectors $q(\boldsymbol{x}; \mathsf{A})$. As all covariates X are events preceding E_i sampled by GPT-J, there is an implicit conditioning on E_i. Thus, we can approximately get $\mathbb{P}(X(0)) \approx f(\mathsf{X}, \mathsf{E}_i)$ (see Appendix B for more details). Then, temporal propensity vectors are computed as

$$q(\boldsymbol{x}; \mathsf{A}) = \left(\frac{\mathbb{P}(X(0), A(1))}{\mathbb{P}(X(0))} \right)_{\mathsf{X} \in \mathcal{X}}. \quad (9)$$

Finally, the score estimator computes $\hat{\Delta}_i$ in Equation (7). We also test normalization methods in Appendix E and observe that some of those normalization methods can benefit our framework.

6 Experiment

We conduct extensive experiments and compare **COLA** with a wide selection of baselines.

6.1 Dataset

Since our work is the first attempt to study Contextualized CCR, we carried out human annotation on Amazon Mechanical Turk. We randomly sampled event sequences from ROCStories (Mostafazadeh et al., 2016), where each sequence contains five chronologically ordered events. Workers are asked to annotate whether an event causes the last event in a sequence. There are two qualification tests to choose workers to maintain rigorous quality control. See more details in Appendix D.

Eventually, we collected a dataset containing 1,360 event pairs, called *Choice of Plausible Event in Sequence* (**COPES**). We equally divide them into a validation set and a testing set.

6.2 Evaluation Metric

We calculate accuracy, F1-score, and Macro F1-score between predicted labels and ground truth labels, to automatically evaluate all models on our dataset. Notice that our task definition provides the number of positive events in a sequence, so that recall, precision, and F1-score are the same.

6.3 Baseline Methods

We compare our framework to three baselines:

CLM Perplexity An intuitive solution to the contextualized CCR task would be computing perplexity scores for each pair of events with a causal language model (CLM). An event pair (E_i, E_j) within a sequence $E_1, E_2, \ldots, E_n$ (n is sequence length) is converted into full-text input with the prompt template: "If $E_1, E_2, \ldots, E_n$, E_j because E_i". The causal language models we tested are GPT2, GPT2-medium/large/XL (Radford et al., 2019), and GPT-J (Wang and Komatsuzaki, 2021).

Cloze Prompt Score This baseline proposed by Tamborrino et al. (2020) concatenates two events (E_i, E_j) into full-text input. Then, it masks and tries to recover each token with a masked language model. It averages log-likelihood over every token as the final score of two events. The prompt used is the same as **CLM Perplexity**. Multiple masked language models are tested: BERT-base/large (Devlin et al., 2018), RoBERTa-base/large (Liu et al., 2019), DeBERTa-base/large (He et al., 2020).

ROCK This baseline is a causal inference framework (Zhang et al., 2022c) that draws analogies between human subjects and natural language. We

Models	Validation Acc	F1	Ma-F1	Δ_{Acc}	Testing Acc	F1	Ma-F1	Δ_{Acc}
Random	59.47	42.35	55.55	-	58.94	41.10	54.79	-
CLM Perplexity (GPT2)	61.76	45.61	58.06	-	61.47	44.73	57.58	-
CLM Perplexity (GPT2-medium)	60.29	43.51	56.45	-	61.76	45.15	57.90	-
CLM Perplexity (GPT2-large)	62.94	47.28	59.35	-	62.65	46.41	58.87	-
CLM Perplexity (GPT2-XL)	62.65	46.86	59.03	-	62.35	45.99	58.55	-
CLM Perplexity (GPT-J 6b)	63.82	48.54	60.32	-	62.06	45.57	58.22	-
ClozePromptScore (BERT-base)	64.41	49.37	60.97	-	63.53	47.68	59.84	-
ClozePromptScore (BERT-large)	66.47	52.30	63.23	-	62.06	45.57	58.22	-
ClozePromptScore (RoBERTa-base)	59.71	42.68	55.81	-	59.71	42.19	55.63	-
ClozePromptScore (RoBERTa-large)	60.59	43.93	56.77	-	59.12	41.35	54.99	-
ClozePromptScore (DeBERTa-base)	56.76	38.49	52.58	-	58.53	40.51	54.34	-
ClozePromptScore (DeBERTa-large)	56.47	38.08	52.26	-	57.06	38.40	52.72	-
ROCK (BERT-base)	66.18	51.88	62.90	-	65.29	50.21	61.79	-
ROCK (BERT-large)	65.59	51.05	62.26	-	66.47	51.90	63.08	-
ROCK (RoBERTa-base)	61.76	45.61	58.06	-	61.18	44.30	57.25	-
ROCK (RoBERTa-large)	62.94	47.28	59.35	-	65.59	50.63	62.11	-
ROCK (DeBERTa-base)	62.65	46.86	59.03	-	60.59	43.46	56.61	-
ROCK (DeBERTa-large)	64.41	49.37	60.97	-	64.12	48.52	60.49	-
COLA (BERT-base)	67.65	53.97	64.52	↑ 1.47	68.82	55.27	65.67	↑ 3.53
COLA (BERT-large)	70.29	57.74	67.42	↑ 4.70	**70.29**	**57.38**	**67.29**	↑ 3.82
COLA (RoBERTa-base)	69.71	56.90	66.77	↑ **7.95**	66.76	52.32	63.41	↑ 5.58
COLA (RoBERTa-large)	70.59	58.16	67.74	↑ 7.65	70.00	56.96	66.97	↑ 4.41
COLA (DeBERTa-base)	69.71	56.90	66.77	↑ 7.06	**70.29**	**57.38**	**67.29**	↑ **9.70**
COLA (DeBERTa-large)	**71.18**	**59.00**	**68.39**	↑ 6.77	69.41	56.12	66.32	↑ 5.29

Table 1: Performance of all frameworks on the validation and testing set of the **COPES** dataset. **COLA** is our model. We abbreviate Accuracy, F1-score, Macro F1-score to Acc, F1, Ma-F1, respectively. We test **COLA** with the temporal predictor based on different models: BERT-base/large, RoBERTa-base/large, and DeBERTa-base/large. Compared with ROCK, improvements of our frameworks are shown under Δ_{Acc} for each language model, respectively.

test different language models for the temporal predictor: BERT-base/large, RoBERTa-base/large, and DeBERTa-base/large.

7 Main Evaluation

We provide results in Table 1 for baselines and **COLA** with the temporal predictor based on different language models. In general, **COLA** can detect commonsense causal relations more accurately, outperforming all baseline models by a large margin. Our framework **COLA** based on DeBERTa-large and DeBERTa-base (also BERT-large) achieves the best performance on the validation and testing set, respectively. Also, changing language models of the temporal predictor in **COLA** only involves a small fluctuation in performance, showing that our framework is robust to underlying language models.

Another observation is that CLM Perplexity and ClozePromptScore can achieve performance higher than the random result. This manifests that pre-trained language models with commonly used pre-training objectives can capture commonsense causal relations to some extent.

7.1 Ablation Study

In this section, we conduct four ablation experiments to demonstrate that our causal inference-motivated framework can mitigate spurious correlations between events and boost performance.

Temporal Propensity Matching The first three ablation experiments prove the effectiveness of temporal propensity matching. Here, we separately remove three modules in our framework: (i) We drop the *multistamp covariate sampler* and sample covariates only based on the last timestamp (⋄ w/o Multi Step). This experiment verifies the benefit of utilizing context to detect commonsense causality. (ii) We remove all interventions (⋄ w/o Inter) and use temporal precedence as causation: $\hat{\Delta}_i = \mathbb{P}(\mathsf{E}_i \prec \mathsf{E}_n)$, equivalent to $\epsilon = 0$ in Equation (7). (iii) Covariates are removed (⋄ w/o Cov) so that interventions are not adjusted. This unadjusted score keeps all sampled interventions, equivalent to $\epsilon = 1$ in Equation (7).

From the results in Table 2, we observe that balanced estimand $\hat{\Delta}_i$ achieve better performances, showing that multiple timestamp sampling, treat-

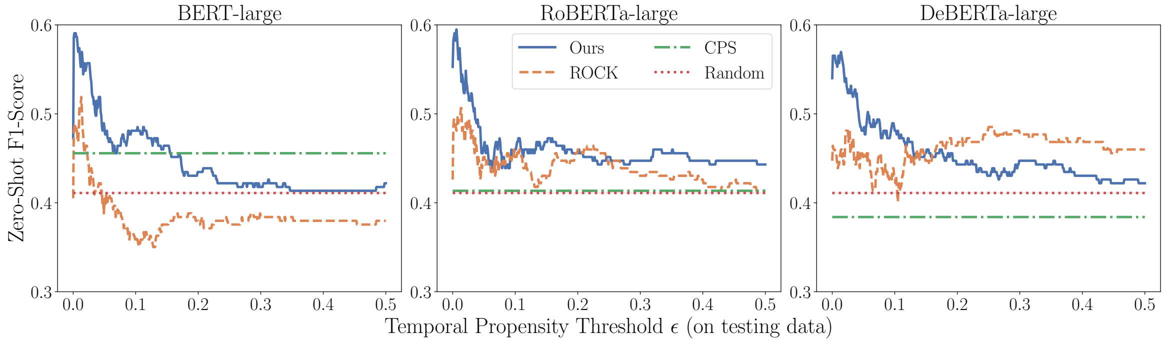

Figure 3: F1-score of **COLA** ("Ours") with ϵ ranging from 0 to 0.5. We also compare our framework with baselines ClozePromptScore ("CPS") and ROCK. Balancing covariates in our framework can significantly outperform unbalanced variants ($\epsilon \approx 0.5$).

Models	**Acc**	**F1**	**Ma-F1**	Δ_{Acc}
Ours(BERT-large)	**70.29**	**57.38**	**67.29**	-
◇ w/o Multi Step	70.00	56.96	66.97	↓ 0.29
◇ w/o Inter	60.29	43.04	56.28	↓ 10.00
◇ w/o Cov	57.65	39.24	53.37	↓ 12.64
◇ w/o Temp	64.41	48.95	60.82	↓ 5.88
Ours(RoBERTa-large)	**70.00**	**56.96**	**66.97**	-
◇ w/o Multi Step	67.94	54.01	64.70	↓ 2.06
◇ w/o Inter	64.71	49.37	61.14	↓ 5.29
◇ w/o Cov	58.53	40.51	54.34	↓ 11.47
◇ w/o Temp	65.00	49.79	61.46	↓ 5.00
Ours(DeBERTa-large)	**69.41**	**56.12**	**66.32**	-
◇ w/o Multi Step	68.24	54.43	65.03	↓ 1.17
◇ w/o Inter	67.65	53.59	64.38	↓ 1.76
◇ w/o Cov	55.88	36.71	51.42	↓ 13.53
◇ w/o Temp	64.41	48.95	60.82	↓ 5.00

Table 2: Ablation study on **COLA** ("Ours"). The column Δ_{Acc} indicates accuracy drops. The first three experiments shows the effectiveness of temporal propensity matching and the last one is about temporal predictor.

Models	**Acc**	**F1**	**Ma-F1**	Δ_{Acc}
Uni (BERT-large)	70.29	57.38	67.29	-
Int (BERT-large)	68.82	54.89	65.54	↓ 1.47
Uni (RoBERTa-large)	70.00	56.96	66.97	-
Int (RoBERTa-large)	68.82	54.89	65.54	↓ 1.18
Uni (DeBERTa-large)	69.41	56.12	66.32	-
Int (DeBERTa-large)	68.82	54.89	65.54	↓ 0.59

Table 3: Comparison of the different covariate sampling methods. **Uni** and **Int** stand for "Union" and "Intersection," respectively.

ment effect, and balancing covariates play essential roles in detecting commonsense causation accurately. Removing any of the three modules will result in sheer drops in all metrics. These experiments imply that temporal relation is vital in Contextualized CCR, but it is still insufficient due to spurious correlations. Thus, we need to measure *average treatment effect* with balancing covariates.

Temporal Predictor We also ablate the temporal predictor (◇ w/o Temp) to verify the effectiveness of ***temporal fine-tuning*** (in Section 5). Here, we use pre-trained language models and increase k to 30 to mitigate the problem of low coverage.

As shown in Table 2, a directly pre-trained language model without fine-tuning cannot perform well. We conclude that pre-trained language models do not have sufficient "temporal awareness" and ***temporal fine-tuning*** is necessary for our framework.

7.2 Rules-of-thumb for Choosing ϵ:

The hyperparameter ϵ controls the number of interventions when balancing covariates in Equation (7). In Figure 3, we can observe that a recommended range for ϵ is $\epsilon \in [0, 0.1]$. We also list the optimal ϵ in Appendix A.3. From Table 6, ϵ should be fairly small within [0.001, 0.015]. Though the best solution relies on how to implement the components in **COLA** and data distribution, our analysis can provide a good starting point. We also study the effect of changing another important hyperparameter: covariate set size $N = |\mathcal{X}|$, in Appendix C.

7.3 Union and Intersection

Our framework **COLA** samples covariates from multiple timestamps and take union on them to get the final covariate set. We also introduce another method to sample covariates preceding all timestamps after and including E_i ("intersection") in Section 5. Here, we conduct experiments to discuss the differences between these two methods. For the "intersection" method, we also use

Models	**Acc**	**F1**	**Ma-F1**
ChatGPT	75.83	60.27	71.45
ChatGPT w/o k	73.33	62.79	71.01
COLA (DeBERTa-large)	**80.00**	**65.71**	**75.80**

Table 4: Comparison with ChatGPT. Our model outperforms ChatGPT with much fewer parameters.

GPT-J to sample covariates with "There are temporally ordered events $[E_i, E_{i+1}, \ldots, E_n]$. Before all events," being the prompt template.

As shown in Table 3, the "union" method gets better performance since it can diversify covariate sets. It samples covariates conditioned on each event before E_i separately. Meantime, each covariate of the "intersection" method is only conditioned on the same context $E_i, E_{i+1}, \ldots, E_n$. We compute the self-BLEU (Zhu et al., 2018) to evaluate the diversity of the generated covariates. The self-BLEU of the "intersection" method is 66.40% while that of the "union" method is 41.34%, quantitatively showing that our method can diversify the covariate set.

7.4 Comparison with ChatGPT

Large language models have shown strong performance on extensive NLP tasks (Radford et al., 2019; Ouyang et al., 2022). Thus, we compare our framework with ChatGPT[3], the latest large language model trained using Reinforcement Learning from Human Feedback (RLHF). To adopt ChatGPT to our task, we design the prompt template: "Given a story as a chain of events $E_1, E_2, \ldots, E_n$, which k event(s) in the first $n-1$ events more plausibly cause the last event?" where k is the number of positive events in the ground truth. We randomly sample 120 examples within 30 event sequences from our dataset **COPES** and manually read predicted labels from ChatGPT's answers. We also test ChatGPT without providing the number k, i.e., removing k from the prompt.

The experimental result in the selected 120 examples is shown in Table 4. From the table, we can find that providing ChatGPT with the number k does not lead to too much change in all metrics. (More discussion about this in Appendix F) Also, our framework **COLA** achieves better zero-shot performance than ChatGPT with much fewer parameters.

[3]https://openai.com/blog/chatgpt/

8 Conclusion

In this paper, we design a new task to consider the context when detecting commonsense causal relations. We also crowd-sourced a dataset with strict quality control to benchmark the task. Our **COLA** framework is motivated by the principles of causal inference and attempts to leverage context information. The extensive evaluation demonstrates the effectiveness of **COLA** for detecting commonsense causal relations.

9 Acknowledgement

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong and the National Key R&D Program of China (2019YFE0198200) with special thanks to HKMAAC and CUSBLT. This paper was also supported by the Tencent Rhino-bird Focused Research Program. We also thank the support from NVIDIA AI Technology Center (NVAITC) and the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

Limitations

Our **COLA** framework is mainly based on pre-trained language models, such as GPT-J and `PolyJuice`. We only utilize the simplest prompt to sample covariates and interventions. Further efforts such as prompt engineering are expected to boost the performance of **COLA** further. Also, pre-trained language models can be biased by implicit events and reporting biases in their training data. Such biases lead the framework to omit some critical covariates and interventions, hindering our framework from achieving more accurate detection of commonsense causal relations. How to account for this problem of language model remains to be studied. Last but not least, the temporal propensity used in our framework is only an approximation of the exact propensity (Zhang et al., 2022c) since it is unclear how to pack an unordered set of covariates into a sequential input for language models. Further studies are needed to design the exact propensity and improve performance.

Ethics Statement

This work presents **COPES**, a free and open dataset for the research community to study the contextualized CCR problem. Examples in **COPES** are collected from ROCStories (Mostafazadeh et al., 2016), a free and open dataset about anonymized chains of events. Each chain logically follows everyday topics and does not involve privacy problems about any specific entities (e.g., a person or company). We carried out human annotation on Amazon Mechanical Turk. Annotators are fairly paid 1.2 USD for each HIT, which fulfills the minimum wage requirement.

References

Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. 2021. Automatic story generation: Challenges and attempts. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 72–83.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

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

Du-Seong Chang and Key-Sun Choi. 2005. Causal relation extraction using cue phrase and lexical pair probabilities. In *International Conference on Natural Language Processing*, pages 61–70. Springer.

Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021. Event-centric natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 6–14.

Muhao Chen, Hongming Zhang, Haoyu Wang, and Dan Roth. 2020. What are you trying to do? semantic typing of event processes. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 531–542.

William G Cochran and S Paul Chambers. 1965. The planning of observational studies of human populations. *Journal of the Royal Statistical Society. Series A (General)*, 128(2):234–266.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Quang Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303.

Amir Feder, Katherine A Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E Roberts, et al. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158.

Ronald A Fisher. 1958. Cancer and smoking. *Nature*, 182(4635):596–596.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *ACL 2018*, page 1.

Deepanway Ghosal, Pengfei Hong, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2021. Cider: Commonsense inference for dialogue explanation and reasoning. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 301–313.

Georgios Giannarakis, Vasileios Sitokonstantinou, Roxanne Suzette Lorilla, and Charalampos Kontoes. 2022. Towards assessing agricultural land suitability with causal machine learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1442–1452.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398.

Clive WJ Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Mingyue Han and Yinglin Wang. 2021. Doing good or doing right? exploring the weakness of commonsense causal reasoning models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 151–157.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

James J Heckman. 2005. Rejoinder: response to sobel. *Sociological Methodology*, 35(1):135–150.

Paul W Holland. 1986. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170.

Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. 2020. Pretraining with contrastive sentence objectives improves discourse performance of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4859–4870.

Zijian Jin, Xingyu Zhang, Mo Yu, and Lifu Huang. 2022. Probing script knowledge from pre-trained models. *arXiv preprint arXiv:2204.10176*.

Dongyeop Kang, Varun Gangal, Ang Lu, Zheng Chen, and Eduard Hovy. 2017. Detecting and explaining causes from text for a time series event. *arXiv preprint arXiv:1707.08852*.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. 2019. When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42.

Katherine A Keith, David Jensen, and Brendan O'Connor. 2020. Text and causal inference: A review of using text to remove confounding from causal estimates. *arXiv preprint arXiv:2005.00649*.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45.

Zizheng Lin, Hongming Zhang, and Yangqiu Song. 2023. Global constraints with prompting for zero-shot event argument classification. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2482–2493.

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. *arXiv preprint arXiv:1907.11692*.

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.

Paramita Mirza and Sara Tonelli. 2014. An analysis of causality between events and its relation to temporal information. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2097–2106.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2019. Joint reasoning for temporal and causal relations. *arXiv preprint arXiv:1906.04941*.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

James Robins. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512.

Paul R Rosenbaum. 2002. Observational studies. In *Observational Studies*, pages 1–17. Springer.

Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Donald B Rubin. 1980. Bias reduction using mahalanobis-metric matching. *Biometrics*, pages 293–298.

Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. *arXiv preprint arXiv:2004.05483*.

Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. 1990. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pages 465–472.

Ieva Staliūnaitė, Philip John Gorinski, and Ignacio Iacobacci. 2021. Improving commonsense causal reasoning by adversarial training and data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13834–13842.

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training is (almost) all you need: An application to commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3878–3887.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 696–706.

Yuxin Wang, Jieru Lin, Zhiwei Yu, Wei Hu, and Börje F Karlsson. 2022a. Open-world story generation with structured knowledge enhancement: A comprehensive survey. *arXiv preprint arXiv:2212.04634*.

Zhaowei Wang, Hongming Zhang, Tianqing Fang, Yangqiu Song, Ginny Wong, and Simon See. 2022b. SubeventWriter: Iterative sub-event sequence generation with coherence controller. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1590–1604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. 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 6707–6723.

Bo Zhang and Jiayao Zhang. 2021. Some reflections on drawing causal inference using textual data: Parallels between human subjects and organized texts. In *First Conference on Causal Learning and Reasoning*.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020a. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550.

Hongming Zhang, Xin Liu, Haojie Pan, Haowen Ke, Jiefu Ou, Tianqing Fang, and Yangqiu Song. 2022a. Aser: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities. *Artificial Intelligence*, page 103740.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020b. Aser: A large-scale eventuality knowledge graph. In *Proceedings of the web conference 2020*, pages 201–211.

Hongming Zhang, Wenlin Yao, and Dong Yu. 2022b. Efficient zero-shot event extraction with context-definition alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7169–7179, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiayao Zhang, Hongming Zhang, Weijie Su, and Dan Roth. 2022c. Rock: Causal inference principles for reasoning about commonsense causality. In *International Conference on Machine Learning*, pages 26750–26771. PMLR.

Junhao Zheng, Zhanxian Liang, Haibin Chen, and Qianli Ma. 2022. Distilling causal effect from miscellaneous other-class for continual named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Models	Valid Acc	Test Acc
Ours(BERT-base)	87.10	85.81
Ours(BERT-large)	88.93	87.38
Ours(RoBERTa-base)	89.76	88.97
Ours(RoBERTa-large)	91.91	91.10
Ours(DeBERTa-base)	90.07	89.06
Ours(DeBERTa-large)	93.10	92.40

Table 5: The accuracy of each temporal predictor on the validation and testing splits of the temporal relation dataset.

A Implementation Details

In this appendix, we introduce the implementation details of every component in our framework **COLA**. We conduct all experiments on 8 NVIDIA A100 GPUs. Since our **COLA** and baselines are zero-shot, we test each model once and report the results.

Parameter Number We list parameter numbers of all pre-trained langauge models here. BERT-base/large, RoBERTa-base/large, and DeBERTa-base/large have 110M, 340M, 125M, 355M, 100M, and 350M parameters, respectively. GPT, GPT-medium/large/XL, and GPT-6b contain 117M, 345M, 774M, 1.5B, and 6B parameters, respectively.

A.1 Temporal Predictor

We fine-tune temporal predictors based on BERT-base/large, RoBERTa-base/large, and DeBERTa-base/large for ten epochs. The learning rate is 1e-5, and the batch size is 256. We sampled 800k examples (event pairs) from ROCStories (Mostafazadeh et al., 2016) to build a temporal relation dataset. The ROCStories dataset contains about 100k temporally ordered event sequences. The proportion of training, validation, and testing splits of our sampled event pairs are 98:1:1. We show the accuracy of each fine-tuned temporal predictor on the validation and testing splits in Table 5.

A.2 Intervention and Covariate

For each event pair, we sample 50 covariates using GPT-J with a maximum length of generated events of 15 and a temperature of 0.9. We also sample 50 interventions using `PolyJuice` with a maximum length of generated events of 40 and a temperature of 1.0.

`PolyJuice` provided various control codes to manipulate events in various manners. The full list of control codes is `negation`, `lexical`, `resemantic`, `quantifier`, `insert`, `restructure`, `shuffle`, and `delete`. We only use `resemantic`, `negation`, `lexical`, `quantifier`, `insert`, `delete` to generate interventions since `restructure` and `shuffle` do not generate counterfactual events.

Models	Best ϵ
Ours(BERT-large)	0.006
Ours(RoBERTa-large)	0.001
Ours(DeBERTa-large)	0.014

Table 6: The best ϵ for our framework **COLA** with the temporal predictor based on different language models.

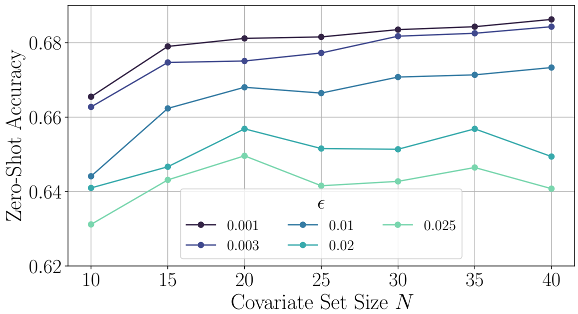

Figure 4: Accuracy of **COLA** ("Ours") with different covariate set size N. Increasing covariate set size improves performance with a reasonable ϵ, such as 0.001, 0.003, and 0.01.

A.3 Temporal Propensity Threshold ϵ

In this section, we list the best temporal propensity threshold ϵ for our framework **COLA** with the temporal predictor based on different language models in Table 6.

B Probability Approximation

Our approximation of the probability of a covariate $\mathbb{P}(X(0))$ in Section 5 is a consequence of the efficiency trade-off. Notice that even though a covariate X is an event preceding E_i sampled by GPT-J, it can occur before other events. Strictly, we must enumerate all possible events E, compute $\mathbb{P}(\mathsf{X} \prec \mathsf{E})$, and marginalize out E. Here, we assume $\mathbb{P}(E(1)|X(0)) \approx 1$ to compute $\mathbb{P}(X(0))$ efficiently. We approximately have:

$$\begin{aligned}\mathbb{P}(\mathsf{X} \prec \mathsf{E}) &= \mathbb{P}(X(0), E(1)) \\ &= \mathbb{P}(X(0))\mathbb{P}(E(1)|X(0)) \\ &\approx \mathbb{P}(X(0))\end{aligned} \tag{10}$$

C Number of Covariates

We show that balancing covariates are essential in our framework **COLA**. Meanwhile, the number of covariates also has an impact on performance. Here we test different covariate set size $N = |\mathcal{X}|$ spanning from 10 to 40, shown in Figure 4 with various ϵ values.

We observe that gradually increasing the covariate set size N can enhance the performance of our framework when the ϵ is within [0.001, 0.01] as we recommend in Section 7.2. On the other hand, adding more covariates may introduce more noise if we test with a larger ϵ. For example, when ϵ equals 0.02 and 0.025, increasing covariates only brings about fluctuations in accuracy.

D Dataset Annotation

Since our work is the first attempt to study Contextualized CCR, we carried out human annotation on Amazon Mechanical Turk to construct a new dataset. In this appendix, we discuss the annotation process of the dataset **COPES** in our paper. We randomly sampled event sequences from ROCStories (Mostafazadeh et al., 2016), where each sequence contains five chronologically ordered events. Workers are provided with event sequences and are asked to annotate whether an event causes the last event in a sequence[4]. Each human intelligence task (HIT) includes ten potential (*cause*, *effect*) event pairs (with corresponding event sequences), and each pair is labeled by seven workers. We take the majority vote among seven votes as the final result for each pair.

We conduct two qualification tests to choose workers to maintain rigorous quality control. First, we invited annotators who meet the following conditions to take our qualification examinations: 1) an approval rate of at least 95% and 2) at least a thousand approved HITs. In the second round, a qualification question set including both effortless and tricky examples is collected by experts, namely the authors of this paper, who have a clear understanding of Contextualized CCR. The experts annotate 160 event pairs sampled from ROCStories. An annotator needs to answer a HIT involving ten questions from the qualification set, and the answers are compared with the experts' answers. An annotator should correctly answer 8 out of 10 questions to pass the second round test. While 307 annotators participated in the second round qualification test, only 29 (9.45%) were selected as our main round annotators.

Eventually, we collected the dataset containing

[4]There may be multiple causes of an event.

#Causal Relation	0	1	2	3	Total
#Seq	12	192	124	12	340
Proportion (%)	3.5	56.5	36.5	3.5	100

Table 7: The breakdown of numbers of causal relations in all event sequences in the **COPES** dataset. Most event sequences (56.6%) own only one causal relation.

human-annotated labels for 1,360 pairs from 340 event sequences ($1360 = 340 \times 4$), called *Choice of Plausible Event in Sequence* (**COPES**). The IAA score is 61.13% calculated using pairwise agreement proportion, and Fleiss's κ (Fleiss, 1971) is 0.52. We equally divide them into a validation set and a testing set (e.g., each contains 680 examples). We also provide a screenshot of our annotation platform in Figure 5.

D.1 Dataset Statistics

Here we provide the breakdown of numbers of causal relations k in all 340 event sequences in Table 7. Also, 476 event pairs are positive (with causal relation), and 884 are negative (without causal relation).

E Normalization

In this section, we enumerate some methods to normalize estimand $\hat{\Delta}$ and temporal propensity vectors in Equation (7).

Direct Matching (D): Instead of condition probability in Equation (9), we can directly use the vectors of temporal relation scores $(f(\mathsf{A}, \mathsf{X}))_{\mathsf{X} \in \mathcal{X}}$ as propensity vectors.

Score Simplification (S): We fine-tune temporal predictors to prompt `before`, `after`, and `[none]`, which might be difficult for smaller Masked LMs, like BERT-base. When constructing propensity vectors, we simplify the task to consider only `before`, `after` by normalizing scores:

$$f(\mathsf{X}, \mathsf{A}) = \frac{f_b(\mathsf{X},\mathsf{A}) + f_a(\mathsf{A},\mathsf{X})}{f_b(\mathsf{X},\mathsf{A}) + f_a(\mathsf{X},\mathsf{A}) + f_b(\mathsf{A},\mathsf{X}) + f_a(\mathsf{A},\mathsf{X})}$$

Propensity Covariate Normalization (Q): We also try to normalize temporal relation scores on the covariate set $\mathcal{X}$ before building propensity vectors: $\mathbb{P}(X) = f(\mathsf{X}, \mathsf{E}_i) / \sum_{\mathsf{X}' \in \mathcal{X}} f(\mathsf{X}', \mathsf{E}_i)$ and $\mathbb{P}(X, A) = f(\mathsf{X}, \mathsf{A}) / \sum_{\mathsf{X}' \in \mathcal{X}} f(\mathsf{X}', \mathsf{A})$.

Co-occurrence Normalization (C): The fine-tuned temporal predictor may sometimes still faces

Models	Acc	F1	Ma-F1
Ours(BERT-base)	68.82	55.27	65.67
⋄ w/o S	↓ 0.59	↓ 0.84	↓ 0.65
⋄ w/o Q	↓ 3.24	↓ 4.64	↓ 3.56
⋄ w/o E	↓ 5.29	↓ 7.59	↓ 5.83
Ours(BERT-large)	70.29	57.38	67.29
⋄ w/o Q	↓ 0.59	↓ 0.84	↓ 0.65
⋄ w/o E	↓ 7.06	↓ 10.13	↓ 7.77

Table 8: Ablation study of normalization methods on BERT-base and BERT-large.

Models	Acc	F1	Ma-F1
Ours(RoBERTa-base)	66.76	52.32	63.41
⋄ w/o C	↓ 1.47	↓ 2.11	↓ 1.62
Ours(RoBERTa-large)	70.00	56.96	66.97
⋄ w/o S	↓ 0.59	↓ 0.84	↓ 0.65
⋄ w/o E	↓ 2.06	↓ 2.95	↓ 2.27

Table 9: Ablation study of normalization methods on RoBERTa-base and RoBERTa-large.

the problem of low coverage, causing estimates $f(\mathsf{E}_i, \mathsf{E}_n)$ and $f(\mathsf{A}, \mathsf{E}_n)$ inaccurate in Equation (7). We set them to $(f(\mathsf{E}_i, \mathsf{E}_n) + f(\mathsf{E}_n, \mathsf{E}_i))/2$ and $(f(\mathsf{A}, \mathsf{E}_n) + f(\mathsf{E}_n, \mathsf{A}))/2$, respectively.

Estimand Normalization (E): In this method, estimates $f(\mathsf{E}_i, \mathsf{E}_n)$ and $f(\mathsf{A}, \mathsf{E}_n)$ in estimand $\hat{\Delta}$ in Equation (7) are normalized by $f(\mathsf{E}_i, \mathsf{E}_n) + f(\mathsf{E}_n, \mathsf{E}_i)$ and $f(\mathsf{A}, \mathsf{E}_n) + f(\mathsf{E}_n, \mathsf{A})$, respectively.

We also conduct comprehensive analyses about removing these normalizations. The results are shown in Tables 8 to 10. We present drops in every metric when removing each normalization method. Some normalization methods cannot improve performance, and we delete those rows from tables. We observe that some of these normalization methods, such as **Estimand Normalization (E)**, can benefit our framework on Contextualized CCR.

F ChatGPT

In this section, we show and discuss the result of ChatGPT on `COPES`. Given the insignificant differences in performance between the prompts with and without the number k in Section 7.4, we focus on the answers of the prompt with k. We show three examples, respectively corresponding to three cases:

1. ChatGPT's answer is correct (with a good explanation)

Models	Acc	F1	Ma-F1
Ours(DeBERTa-base)	70.29	57.38	67.29
⋄ w/o Q	↓ 0.59	↓ 0.84	↓ 0.65
Ours(DeBERTa-large)	69.41	56.12	66.32
⋄ w/o E	↓ 1.18	↓ 1.69	↓ 1.30

Table 10: Ablation study of normalization methods on DeBERTa-base and DeBERTa-large.

2. ChatGPT's answer is different from the ground truth answer, but the correctness is debatable with the explanation from ChatGPT

3. ChatGPT's answer is incorrect.

Example 1. Which event mainly/directly causes the last event?

E_1: Charlie went to food truck event at the beach.

E_2: He said they had a ton of trucks there.

E_3: He tried food from a few trucks.

E_4: He said the food was great.

E_5: I am going to try to go next year.

Gold answer: E_4

ChatGPT answer (short): E_4

ChatGPT answer (original): The event "He said the food was great." directly causes the last event "I am going to try to go next year." as it expresses the protagonist's positive experience and desire to attend again in the future.

Example 2. Which event mainly/directly causes the last event?

E_1: Ericka enjoyed going to restaurants, but she liked to save money too.

E_2: She found a coupon for her favorite restaurant.

E_3: She went there for lunch and ordered a burger.

E_4: When the bill came, she was nervous about the coupon.

E_5: The restaurant accepted the coupon and Ericka saved a lot of money!

Gold answer: $\mathsf{E}_2, \mathsf{E}_3$

ChatGPT answer (short): $\mathsf{E}_2, \mathsf{E}_4$

ChatGPT answer (original): The second and fourth events mainly/directly cause the last event. The second event ("She found a coupon for her favorite restaurant.") sets up the possibility of her saving money, and the fourth event ("When the bill came, she was nervous about the coupon.") establishes that the coupon is being used in this specific instance. Together, these two events lead to the final event ("The restaurant accepted the coupon and ericka saved a lot of money!") where she is able to save money.

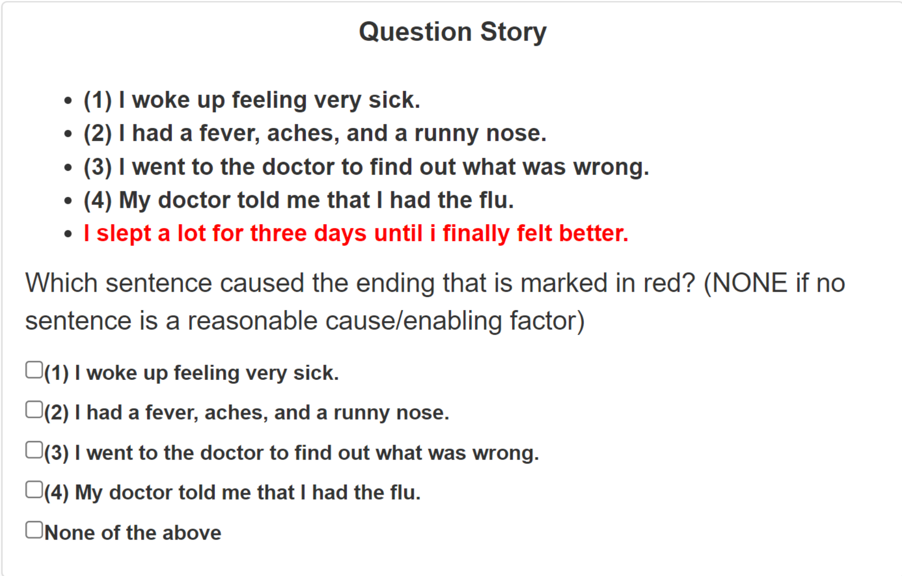

Figure 5: A screenshot of our annotation platform.

Example 3. Which event mainly/directly causes the last event?

E_1: We decided to get married on the beach.
E_2: Our family and friends met us there.
E_3: It was a perfectly sunny summer day.
E_4: Best of all, no one had to wear shoes!.
E_5: We ended our day with a dip in the cool ocean.

Gold answer: E_1
ChatGPT answer (short): E_4
ChatGPT answer (original): The fourth event "Best of all, no one had to wear shoes!" mainly/directly causes the last event "We ended our day with a dip in the cool ocean." as it describes the condition that allowed the last event to happen: people were barefoot.

In most cases, ChatGPT clearly explains its choices. For example, in the **Example 1**, it points out the positive **emotion** *"... food was great."* as the underlying fact that causes the last event *"... going to try to go next year."* (i.e., **decision**). Similarly in the **Example 2**, even though ChatGPT's answer is different from the ground truth answer, ChatGPT mentions two key points *"... The second event ... sets up the possibility of ..."* (i.e., **potential**) and *".. the forth event ... establishes that the coupon is being used ..."* (i.e., **action**) that together causes the last event (i.e., **effect**). The reasoning paradigm implicit in its explanation, e.g **emotion** $\rightarrow$ **decision** and **potential + action** $\rightarrow$ **effect** in the abovementioned two examples, implies the inherent reasoning capability of ChatGPT. Nonetheless, there are cases where ChatGPT fails to give a persuasive answer. However, as a foundation model instead of a task model, ChatGPT's performance is fair enough.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In the Section Limitations, we discuss the limitations of our work.

☑ A2. Did you discuss any potential risks of your work?
In the Section Ethics Statement, we discuss the ethical considerations, including privacy problems.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We annotate new datasets based on a previous dataset (Section 6, Appendix B, and Appendix D) and a new model (Section 4 and Section 5).

☑ B1. Did you cite the creators of artifacts you used?
We cite the previous dataset a few times in Section 6, Appendix B, Appendix D.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The artifacts we use is open sourced without a license.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. No intended use is stated for the existing artifact(s) in our paper.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The artifact we use is already anonymized.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The Section Ethics Statement.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 6, Appendix B, and Appendix D.

C ☑ Did you run computational experiments?

Sections 6 and 7.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 6 and Appendix B.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix B.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 6 and Appendix B.

D ☑ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**
Section 6 and Appendix D.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix D

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 6, Section Ethics Statement, and Appendix D.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 6 and Appendix D.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. For anonymity.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. For anonymity.