THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

CAUSALITY-CENTRIC NARRATIVES REASONING

MU FEITENG

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Computing

Causality-centric Narratives Reasoning

Mu Feiteng

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
July 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Mu Feiteng

Abstract

Narratives are one of the foundational concepts of human society. It is an account of the development of human events, along with explanations of how and why these events happened. Narrative events serve as mirrors reflecting the intricate causality inherent in human activities, rendering them indispensable tools for comprehending the complexities of social dynamics.

Recently, artificial intelligence (AI) has spurred a new era of advancement. However, despite the crucial role narratives plays, a critical bottleneck confronting AI systems lies in enabling machines to comprehend narrative events and leverage them for commonsense narrative reasoning. Specifically, we identify at least three key research problems that must be addressed in this domain of commonsense reasoning within narratives. **Research Problem 1**: How to automatically obtain diverse and high-quality commonsense event knowledge to solve the knowledge bottleneck problem in commonsense reasoning? **Research Problem 2**: How to effectively utilize narrative knowledge for commonsense reasoning to mitigate low-quality issues like dullness and repetition in AI-generated narrative texts, while ensuring the content aligns with human commonsense? More importantly,how to teach AI systems to grasp the causal relationships within narrative events, enabling them to effectively address high-level counterfactual questions? **Research Problem 3**: Given the fact that narrative coherence evaluation is a notoriously difficult thing in the generation community, how can we devise robust quantitative methods to evaluate the coherence of AI-generated

narrative content, thereby furnishing valuable tools for the community?

To solve these challenges, we focus on developing comprehensive narrative reasoning systems from the following three aspects: automatically causality mining, causal-knowledge enhanced narrative reasoning, and hard-negatives mining for narrative coherence learning. Overall, in this thesis, we organize our research works into the following three parts.

In the first part (work 1 and work 2), we explore the research problem 1. Specifically, we explore the rule-based causality extraction method and possible de-biasing approach to harvest causal knowledge from text. In work 1, we manually create causal rules to extract cause-effect pairs from text. And we further construct the event causality network and demonstrate its use in the task of narrative effect generation. In addition, to mitigate the false-positive problem introduced by our rule-based system, in work 2, we explore possible de-biasing approach to obtain high-quality causal knowledge. We inaugurate counterfactual thinking for Event Causality Identification (ECI) to solve the context-keywords bias and event pairs bias problems in existing work. This allows us to obtain high-precision causal event pairs.

In the second part (work 3 and work 4), we explore the solution for research problem 2 with the aim of developing a causal knowledge enhanced reasoning system with stronger causal perception capabilities. In work 3, we delve into causality centric narrative reasoning and push forward the existing knowledge-aware narrative reasoning to a new frontier. We thoroughly leverage multi-level causal knowledge for narrative reasoning, employing a two-stage framework designed to fully exploit the unique characteristics of knowledge across various granularities. Experimental results have shown that our work is effective and can improve the quality of generated narrative effect text. In work 4, we are trying to endow AI systems with more advanced counterfactual reasoning capabilities. One major challenge of counterfactual narrative reasoning is to maintain the causality between the counterfactual condition and the generated counterfactual outcome. Previous works simply utilize supervised datasets

to train conditional generation models, but face the risk of exploiting artifacts of the dataset, instead of learning to robustly reason about counterfactuals. We propose a basic variational approach for counterfactual narrative reasoning. We further introduce a pre-trained classifier and external commonsense event causality to mitigate the model collapse problem in the variational approach, and hence improve the causality between the counterfactual condition and the generated counterfactual outcome. We assess the efficacy of our approach using real-world public benchmarks. Experimental results demonstrate its effectiveness.

In the third part (work 5), we target research problem 3 and propose novel hard-negatives mining strategies for self-supervised narrative coherence learning. Existing works mainly follow the contrastive learning paradigm. However, the negative samples in their methods can be easily distinguished, which makes their methods unsatisfactory. We devise two strategies for mining hard negatives, including (1) crisscrossing a narrative and its contrastive variants; and (2) event-level replacement. To obtain contrastive variants, we utilize the Brownian Bridge process to guarantee the quality of generated contrastive narratives. We assess our model across multiple tasks, confirming its effectiveness and demonstrating its applicability to various use cases.

To sum up, we conduct a comprehensive study on narrative reasoning. Through the use of our proposed methods to real-world datasets, we have illustrated the significant improvements that can be achieved in existing narrative reasoning models. We believe that our works will have a profound impact on the field of narrative reasoning. Although this thesis presents novel methods for this topic, it still has many open problems. We list some future research directions at the end of this thesis.

Publications Arising from the Thesis

1. *Mu, Feiteng*, and Wenjie Li. "Generating Contrastive Narratives Using the Brownian Bridge Process for Narrative Coherence Learning." Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics. (ACL 2024)

2. *Mu, Feiteng*, and Wenjie Li. "A Causal Approach for Counterfactual Reasoning in Narratives." Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics. (ACL 2024)

3. *Mu, Feiteng*, and Wenjie Li. "Enhancing Narrative Commonsense Reasoning With Multilevel Causal Knowledge." IEEE Transactions on Neural Networks and Learning Systems. 2024. (TNNLS)

4. *Mu, Feiteng*, and Wenjie Li. "Enhancing Text Generation via Multi-Level Knowledge Aware Reasoning." Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

5. *Mu, Feiteng*, and Wenjie Li. "Enhancing Event Causality Identification with Counterfactual Reasoning." Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). (ACL2023)

6. *Mu, Feiteng*, Wenjie Li, and Zhipeng Xie. "Effect Generation Based on Causal

Reasoning." Findings of the Association for Computational Linguistics: EMNLP 2021.

7. Wang, J., Li, W., Lin, P., & $\underline{Mu, F.}$(2021). Empathetic response generation through graph-based multi-hop reasoning on emotional causality. Knowledge-Based Systems, 233, 107547. (KBS)

Acknowledgments

Above all else, I would like to express my deepest gratitude to my family, (my mother, my brother, and my girlfriend) for their unwavering support and understanding throughout my doctoral journey. Their comforting presence and unconditional love sustained me during the challenging and uncertain times. Without the support of my family, completing this ultimate academic milestone would have been an infinitely more arduous task. Thank you for believing in me and being my pillars of strength.

I also wish to extend my heartfelt appreciation to Professor Wenjie Li (Maggie), my supervisor. I am deeply grateful for her invaluable feedback and unwavering support during my research endeavors. Under her guidance, I have benefited from a conducive research environment and have been encouraged to actively participate in academic discourse. I am truly privileged and honored to have had the opportunity to learn under her mentorship.

In addition, I am deeply appreciative of my dear friend, Dr. Muhui Jiang, whose invaluable advice and unwavering encouragement have been instrumental on my journey towards obtaining a PhD. His guidance and support have consistently served as a wellspring of motivation and inspiration for me.

Finally, I want to thank my classmates and friends, including Chenglong Hu, Xunpeng Huang, Heng Li, Shichao Sun, and Qin Wang, for their endless care and warmth.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Causality, also referred to as causation, indicates a special semantic relation between one process (the cause) with another process or state (the effect), where the cause is partially responsible for the effect, and the effect is partially dependent on the cause [43]. In the field of natural language processing (NLP), causality can be described using structural causal model [89, 91, 92], or be expressed with text [119, 24, 23, 103]. For example, the text "*Sara has a bad cold and she feels very uncomfortable.*" contains the causal relation "*have a cold* $\overset{cause}{\longrightarrow}$ *feel uncomfortable*".

Narratives are stories that describe a series of events that occur with causal logic. Narratives is one of the foundational concepts of human society [10, 135]. It is an account of the development of human events, along with explanations of how and why these events happened [36]. Narrative events serve as mirrors reflecting the intricate causality inherent in human activities, rendering them indispensable tools for comprehending the complexities of social dynamics. For instance, here is a narrative text: "*Sara felt famished, prompting her visit to McDonald's where she ordered a hamburger. This particular type of cuisine brings her immense joy.*". This text

distinctly illustrates the causal sequence of events: "*feeling hungry* $\overset{cause}{\longrightarrow}$ *eating food* $\overset{cause}{\longrightarrow}$ *feeling happy*".

With the emergence of deep learning, artificial intelligence (AI) has catalyzed a new era of advancement. The effectiveness of a large number of AI applications depends on a profound understanding of causal logic in narrative events [55]. Using the e-commerce scenario as an illustration, it's imperative for machines to understand that the "dating" event triggers a series of subsequent behaviors, such as "getting married", "traveling", and "having children", etc. This comprehension enables them to precisely anticipate users' future shopping patterns when they engage in "dating" activities, thereby facilitating precise product recommendations to the target user. In conclusion, there is an urgent need for research in narrative-based reasoning, as it will significantly drive the application and advancement of AI technology.

Narratives encapsulate complex contextual information, which is essential for understanding human experiences and behaviors. To fully engage with the world in a manner akin to human interaction, AI systems need to effectively comprehend and reason about narratives. However, despite the crucial role narratives plays, AI systems still have huge trouble in comprehending narrative events due to the great complexity of narratives. Humans can grasp narratives because they hold a vast reservoir of background knowledge in their minds. Machines devoid of narrative logic struggle to make analytical judgments that align with human expectations. This challenge becomes particularly pronounced in scenarios where AI systems are tasked with generating or interpreting narrative texts, such as storytelling [75, 143], dialogue generation [157, 39], causal explanation [27, 17], future prediction [103, 30], counterfactual reasoning [98, 11], and so on. In these tasks, AI models must not only generate informative, coherent, and engaging narratives but also ensure that the generated content aligns with human commonsense and accurately reflects the causal relationships inherent in the underlying events. Even more concerning is that in the majority of existing works, AI systems exhibit notably weak narrative generation

capabilities. They may even produce repetitive, hollow, and unengaging text.

To tackle these challenges, our focus lies in studying narrative reasoning from a causal perspective. This is because substantial evidence [128, 127, 125] suggests that people's comprehension of narratives is significantly shaped by the causal relationships within the narrative stories, highlighting causality as a crucial starting point for narrative comprehension and reasoning. The objective of this thesis is to improve the capability of narrative reasoning systems with causal strategies to break through the limitations of existing approaches, so that developing comprehensive narrative reasoning systems that can effectively understand, reason about, and generate narratives.

1.2 Research Problems and Motivations

Existing methods for narrative reasoning have trouble in generating high-quality and satisfactory content. More specifically, these methods either generate uninformative, repetitive, and unengaging content, or generate incoherent content that cannot reflect the causal relationships between narrative events. To handle these problems and improve the narrative reasoning ability of AI systems, we have summarized the following key research problems:

- **Research Problem 1**: How to automatically obtain diverse and high-quality commonsense event knowledge to solve the knowledge bottleneck problem in commonsense reasoning?
- **Research Problem 2**: How to effectively utilize narrative knowledge for commonsense reasoning to mitigate low-quality issues like dullness and repetition in AI-generated narrative texts, while ensuring the content aligns with human commonsense? More importantly,how to teach AI systems to grasp the causal relationships within narrative events, enabling them to effectively address high-level counterfactual questions?

- **Research Problem 3**: Given the fact that narrative coherence evaluation is a notoriously difficult thing in the generation community, how can we devise robust quantitative methods to evaluate the coherence of AI-generated narrative content, thereby furnishing valuable tools for the community?

The first research problem targets the causality extraction. Cause and effect are important components of narrative. Therefore, the premise of causal enhanced narrative reasoning is to solve the problem of the source of causality. Since the cost of manual annotation for obtaining causal knowledge is extremely high, the automated causal extraction method has extremely high value. To address this issue, we have designed a rule-based causal extraction method. We have manually designed a series of causal extraction rules that can extract large-scale causal pairs from unstructured text. To address this issue, we have designed a rule-based causal extraction method. We have manually designed a series of causal extraction rules that can extract large-scale causal pairs from unstructured text. Next, we designed a result generation task to verify the effectiveness of the extracted causal pairs. However, the extraction rules inevitably introduce noise. In order to reduce the impact of noise, we also explored de-noising methods to improve the quality of causal extraction.

The second research problem targets knowledge-enhanced narrative reasoning. Existing methods often produce low-quality content due to the difficulty in learning semantic interactions solely from training data [22] without a profound comprehension of the input context and background knowledge[157]. From this viewpoint, we focus on utilizing causal knowledge to enhance narrative reasoning systems because causality is a significant relationship for narrative understanding. Causal knowledge in text mainly occurs at the sentence-level and the event-level, and different levels of causality have different characteristics. For example, the sentence-level causalities generally have complex sentence structures, and it is difficult to locate the exact range of causes and effects from them. On the contrary, event-level causalities have simple structures and can be explicitly structuralized in knowledge bases. Though

having different forms, sentence-level, and event-level causalities are the embodiment of causality in different scenarios, and they complement each other. This requires us to design a comprehensive approach to fully utilize multi-level causal knowledge. Beyond the factual narrative reasoning, we also pay attention to the counterfactual reasoning ability of narrative reasoning systems. Narratives contains a large number of causal relationships, which puts a high demand on narrative reasoning systems for understanding event causality. In other words, a qualified narrative reasoning system should accurately capture the causal relationships between narrative events so that answering the counterfactual questions, i.e., anticipating the causal shifts in forthcoming events by applying a counterfactual condition to the original narrative event sequence. This issue naturally lends itself to being framed within a causal mechanism [94], which requires us to infer the posterior background knowledge that is compatible with the counterfactual scenario. With the causal theory based on variational Bayes [45], we are able to use the background compatible with the observed factual narratives to approximate the posterior distribution of the counterfactual scenario. Ultimately,we build the counterfactual narrative reasoning upon causal strategies.

The third research problem focuses on the challenge of effectively evaluating the coherence of AI-generated narrative content, which constitutes a significant challenge within the realm of narrative reasoning. Existing generative models follow the sequence-to-sequence [120] learning paradigm, but suffer from the issue of exposure bias [4, 154]. These models occasionally generate incoherent content, e.g., violating the causal consistency between input and output content. Therefore, tools like coherence evaluation models are crucial as they can be used to filter out incoherent generation. Previous works follow the contrastive learning paradigm, where negative samples are created by negative sampling. However, the negative samples in these methods are generally coarse-grained and superficial, making their methods unsatisfactory. To mine more qualified hard negatives, we conduct research on contrastive narratives generation. By comparing the observed narrative with its contrastive vari-

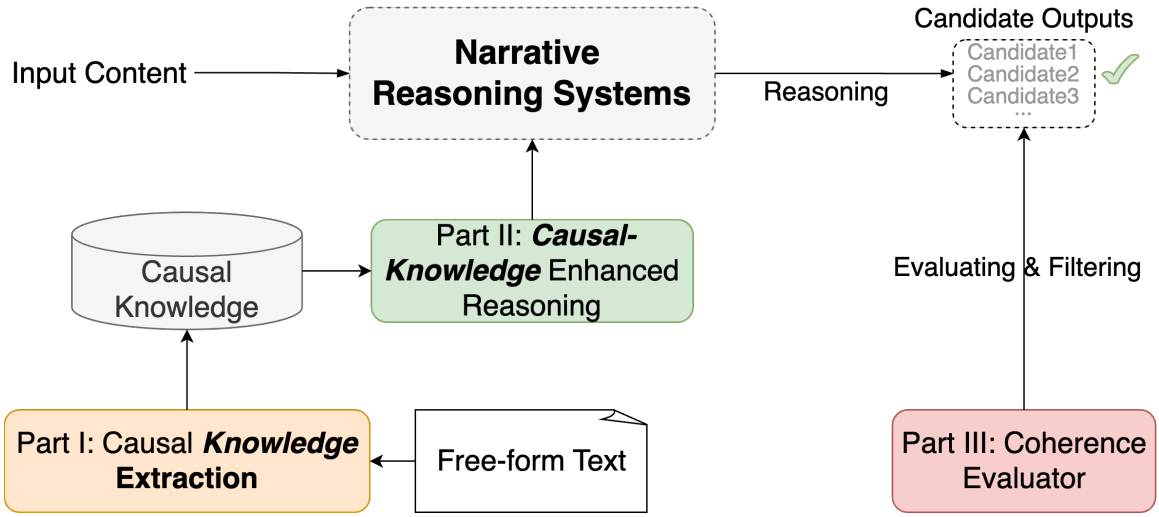


Figure 1.1: The overall framework of the thesis. We use our causality extraction approach to obtain causal knowledge, which are then used to improve narrative reasoning systems at the runtime stage. At the post-processing stage, we use our coherence evaluator to select the most coherent candidate output.

ants, the model can learn what makes the narrative coherent. That is, the model can learn what perturbations to the narrative would make it incoherent. This motivates us to synthesize hard negatives for self-supervised narrative coherence learning.

1.3 Research Framework

The key challenge of narrative reasoning is how to generate high-quality and satisfactory text, ensuring that the generated content conforms to commonsense and satisfies causal consistency between narratives. As shown in the literature review, some works have developed knowledge-enhanced approaches for narrative reasoning systems. However, these works primarily focus on utilizing low-level knowledge or implicit knowledge in pre-trained models, but neglect the significance of causal knowledge in narrative texts. In addition, existing works for counterfactual narrative reasoning mainly fine-tune pre-trained models for counterfactual narrative reasoning,

but face the risk of exploiting artifacts of the dataset, instead of learning to robustly reason about counterfactuals. Furthermore, for narrative coherence evaluation, previous methods are mainly based on the contrastive learning framework, where negative samples are obtained through negative sampling. However, negative samples are too easy to be distinguished, making their methods ineffective. As a result, the progress in this field is still in its initial stage, leaving ample room for potential research.

In this thesis, we focus on causality-centric narrative reasoning. Specifically, we study narrative reasoning from the following three aspects: the automatic causality extraction, the exploitation of causal knowledge for narrative reasoning, and coherence learning with contrastive narrative generation. The overview of these works is depicted in Figure 1.1. Part 1 (Work 1 and 2) mainly solve the problem of causal mining to provide knowledge ground for commonsense narrative reasoning. Part 2 (Work 3 and 4) utilizes causal knowledge to enhance the performance of narrative reasoning systems, including factual and counterfactual reasoning. In Part 3 (Work 5), from the perspective of hard negatives mining, we present a technique for creating contrasting pairs using Brownian bridges, enabling the generation of high-quality negative instances. A concise overview and the contributions of this research are outlined as follows.

Part 1: Automatically Causality Mining and De-biasing

Causality, as an important component of narrative, frequently appears in texts linked by casual connectives. To avoid the huge cost of manual annotation, we designed a rule-based extraction system to obtain causal pairs. Meanwhile, in order to reduce the noise introduced by rules, we also explored the de-noising approach of causal extraction, taking event causality identification (ECI) as an instantiation.

Work 1: To scale-up causality mining, we devise rule-based systems to automatically extract high-precision causal event pairs from free-form text. To demonstrate the

effectiveness of rule-based extraction, we convert the extracted relations into a causal event network. Finally, given an input cause sentence, a causal sub-graph is retrieved and is encoded with the graph attention mechanism, in order to support narrative effect generation.

Contribution: We have created extraction rules that can automatically obtain a large number of causal relationships, avoiding the huge cost of manual annotation. In addition, we devise the causal-graph based method for causal-centric narrative reasoning.

Work 2: In order to reduce extraction noise, we have conducted research on existing ECI works. Existing ECI methods focus on mining potential causal signals, including *causal context keywords* [64] and *causal event pairs* [163], to enhance ECI. However, due to the polysemy of language, causal signals are ambiguous. The occurrence of those signals does not always indicate that causality is established. As a result, they face the risk of amplifying the role of potential signals, resulting in context-keywords bias and event-pairs bias in inference. To solve this issue, we propose the control test that explicitly estimates the influence of context keywords and event pairs in training, so that we are able to eliminate the biases in inference.

Contribution: We consider the spurious correlation problem in ECI, which may make an ECI model overfit on ambiguous causal signals. To mitigate this problem, we propose a counterfactual reasoning mechanism for ECI. To the best of our knowledge, this is the first work that studies ECI from a counterfactual perspective.

Part 2: Utilizing Causal Knowledge for Factual and Counterfactual Narrative Reasoning

Researchers have developed extensive methods, such as graph attention [157] or multi-hop flow [39], to leverage word-level knowledge for improving the understanding of the input background. In fact, for narrative reasoning, causality is a more effective

semantic relationship because it can express richer semantics. In work 3, we have elevated causal-knowledge enhanced narrative reasoning to a new level by considering multi-level causalities. In work 4, We further consider counterfactual reasoning in narratives (CRN), which is a direct verification of the causal perception ability of narrative reasoning systems. Even though it is considered a crucial component of intelligent systems [89, 92], only a few resources have been devoted to CRN. To bridge this gap, we directly challenge counterfactual narrative reasoning to improve the causal ability of AI systems.

Work 3: Causality mainly occurs at the sentence-level and the event-level in text. Though having different forms, sentence-level and event-level causalities are the embodiment of causality in different scenarios, and they complement each other. To fully utilize the strengthen of multi-level causalities, we devise a two-stage approach for narrative reasoning. In the first stage, we use pre-trained models to memorize sentence-level causalities. In the second stage, we utilize event causalities. But, the sparsity of events remains an obstacle. Therefore, we innovatively break down events into multiple word components. The relations between word components capture the interplays between different events, and help mitigate the event sparsity. Based on the event-level causalities and the word-level relations, we construct the novel hierarchical knowledge graph (KG) and devised a KG-based reasoning method.

Contribution: We introduce a two-stage method. This method, designed in a generic framework, proves applicable to a range of narrative understanding tasks. Through the subdivision of events into multiple word components, we derive the hierarchical KG. This not only mitigates the challenge of event sparsity but also provides additional word-level information to enhance narrative reasoning.

Work 4: Generally, CRN relies on the ability to find causality in narratives. This issue naturally lends itself to being framed within a causal mechanism [91], which requires us to infer the background knowledge that is compatible with the counterfactual scenario. However, this is non-trivial as it involves estimating the posterior of

the background knowledge. Luckily, with the variational technique [45], we are able to use the background compatible with the observed narrative to approximate the posterior distribution. In fact, the variational process provides an approximation of the background, but it may face the problem of posterior collapse [106]. As a result, the generated counterfactual output may not be the precise effect of the counterfactual input. To mitigate this problem, we further propose two intuitive strategies, which introduce a pre-trained classifier and commonsense causality, to enhance the causality between the counterfactual input and output.

Contribution: We formulate CRN in a variational framework and introduce event causality and a pre-trained classifier to further improve the causality between the counterfactual input and output. Our method is a general approach that is applicable to multiple tasks. To the best of our knowledge, this is the first work that explores the CRN from a causal perspective. The experiment proves the effectiveness of our method. We also study the practicality of the generated counterfactual narratives via a data augmentation experiment.

Part 3: Hard Negatives Mining for Narrative Coherence Learning

A major challenge for narrative reasoning is to evaluate narrative coherence. Previous works mainly devise self-supervised tasks, in which negative samples are created by sampling-based strategies. The resulting negatives are less representative, and easily distinguishable. Narrative coherence learning urgently requires more hard negative samples. To mitigate this research gap, we propose to synthesize hard negatives with contrastive narratives, in work 5.

Work 5: The ideal of hard negative samples should be that are similar to a real narrative but actually less coherent. Starting from causality between the narrative prefix and suffix, we innovatively introduce contrastive narratives for synthesizing hard neg-

atives. Contrastive narratives are examples that are similar in content, but convey different semantics [69, 132]. By comparing the observed narrative with its contrastive variants, the model can learn what perturbations to the narrative would make it incoherent. Due to this property, we can crisscross a narrative and its contrastive variants to obtain hard negatives. To obtain contrastive narratives, we introduce the Brownian Bridge process to guarantee the quality of generated contrastive narratives. Then we create hard negatives for narrative coherence learning. The acquired model is exclusively trained via self-supervised contrastive learning and is adaptable to an extensive array of subsequent assignments.

Contribution: Based on the Brownian Bridge process, we generate high-quality contrastive narratives, which are used to synthesize hard negatives. We propose a new coherence evaluator, which is enhanced by diverse and high-quality hard negatives. We also conduct an in-depth analysis of our negative sample synthesis strategies.

1.4 Structure of Thesis

The thesis is organized as follows to give an overall picture.

- Chapter 1 first introduces the background of the research on narrative reasoning. This chapter also explains the three key problems, research overview, and contribution of this thesis.

- Chapter 2 reviews existing work in narrative reasoning. This chapter first briefly introduces recent advancements in the field of natural language process, including the RNN-based, transformer-based, and pre-trained foundation model-based technicals. Then, this chapter illustrates the related topics and existing methods in the field of narrative reasoning.

- Chapter 3 introduces our rule-based extraction system. We scale up causal

extraction with extraction rules and construct the event causalities network. Then we demonstrate its use in the task of narrative effect generation.

- Chapter 4 introduces our works about ECI. This chapter focuses on developing de-baising models to mitigate the noise in event causalities extraction. We introduce the novel counterfactual thinking into ECI to mitigate the biased inference problem in previous works.

- Chapter 5 introduces our works about making full use of multi-level causal knowledge for factual narrative reasoning. We present a two-stage framework, in which sentence-level causalities are utilized in the first stage, and event-level causalities are leveraged in the second stage. We also present a novel mechanism incorporating a hierarchical knowledge graph when mitigating the sparsity of events.

- In chapter 6, we introduce our work for counterfactual reasoning in narratives (CRN). We build CRN model on the variational Bayes theory, and propose two additional strategies to alleviate the posterior collapse problem in the variational process.

- Chapter 7 presents a novel narrative coherence learning method that introduces contrastive narratives for hard negatives mining. To obtain contrastive narratives, we use the Brownian Bridge process as the basis to ensure the quality of generated contrastive narratives.

- Chapter 8 summarizes the proposed approaches, our findings, contributions as well as suggestions for future work.

Chapter 2

Literature Review

This chapter first provides a brief overview of canonical neural models frequently employed in natural language processing. These include recurrent neural networks, the transformer model, and transformer based pre-trained language models. These models are necessary because some techniques are used in this thesis. Next, this chapter reviews related datasets and works for narrative commonsense reasoning.

2.1 Canonical Neural Modeling in Natural Language Process

Over the past decade, there has been a notable triumph in neural networks, establishing itself as the prevailing algorithm in various industries and finding widespread application across domains like computer vision, speech processing, natural language understanding, and more. In this segment, we provide a succinct overview of the latest innovations and prevalent neural network architectures utilized by researchers in NLP.

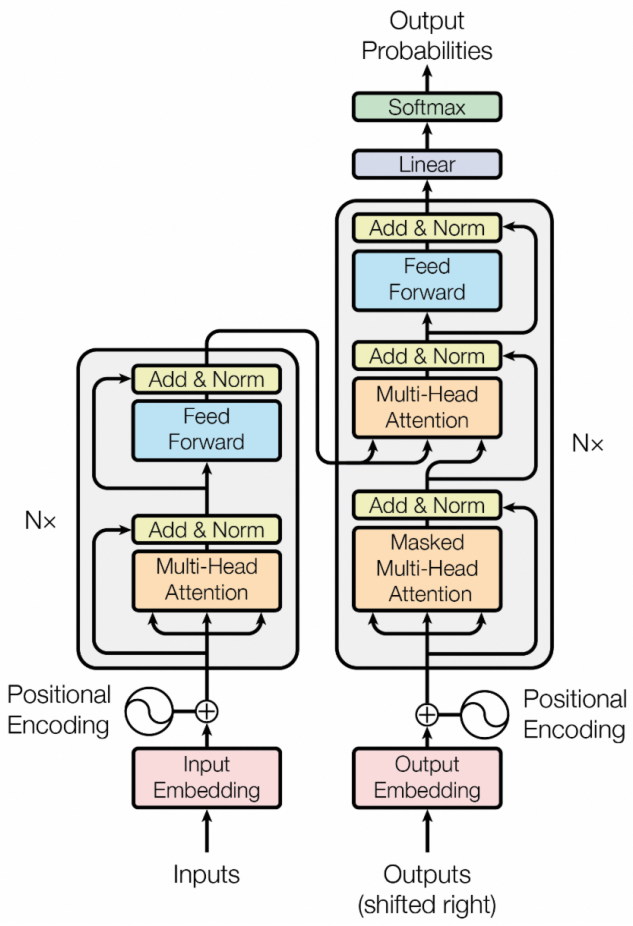


Figure 2.1: The architecture of transformer model [130].

2.1.1 Recurrent Neural Networks

Recurrent neural networks (RNNs) represent a prevalent computational framework employed for handling sequential data. Broadly speaking, RNNs amalgamate the input of the current temporal instance with the latent state from the antecedent temporal instance to calculate the current temporal instance's latent state and output. One significant advantage of RNNs is their linear memory consumption, enabling them to handle sequences of varying lengths. However, traditional RNNs encounter challenges such as gradient vanishing and exploding. To mitigate these issues, two RNN variants, namely Gated Recursive Unit (GRU) [13] and Long Short-Term Mem-

ory (LSTM) [32] networks, were proposed and have found widespread application in various domains.

2.1.2 Transformer Models

Recurrent architectures commonly distribute computation across the symbol positions of input and output sequences. By synchronizing positions with steps in computational time, they produce a succession of hidden states h_t, determined by the antecedent hidden state h_{t-1} and the input for position x_t. This intrinsic sequential characteristic obstructs parallelization within individual training instances, a factor of increasing significance with extended sequence lengths due to memory constraints constraining batch processing across instances. [130]. [130] introduced the Transformer, a paradigm that forsakes recurrence and relies solely on a self-attention mechanism to apprehend comprehensive interdependencies between input and output, devoid of sequence-aligned RNNs or convolutional layers. The architecture of transformers is shown in Figure 2.1. They posited the multi-head attention layer, primed with three vectors:the query, keys, and values.. The yield is calculated as a weighted summation of the values, wherein the weight to each value is derived through a concordance algorithm of the query with its corresponding key, as shown in Equation 2.1.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Furthermore, they observed advantages in applying h linear projections to the queries, keys, and values, enabling the model to collectively attend to diverse representation subspaces across various positions. This process culminated in the formation of the Multi-Head attention mechanism, as illustrated in Equation 2.2.

$$\begin{gathered}\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head1}_1, \text{head}_2, \cdots, \text{head}_h)W^O \\ \text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{gathered} \tag{2.2}$$

The original transformer model [130] takes an encoder-decoder framework, and there are three different types of multi-head attention layers in the transformer model:

- Within the encoder segment, each input token is scrutinized comprehensively, with even initial tokens considered akin to subsequent ones. Thus, the self-attention layers within the encoder adopt a bidirectional architecture, facilitating each position's ability to attend to all input positions.

- On the contrary, in the decoder part, tokens are inputted one by one, and the tokens inputted first cannot perceive the tokens that follow. To ensure this feature, the self-attention layers of the decoder part require an additional causal mask to ensure the autoregressive property, i.e., the current token can only perceive the previous tokens.

- Additionally, transformer models feature cross-attention layers. These layers entail setting the queries as the output of the preceding decoder layer, while keys and values are established as the encoder outputs. Such a configuration empowers each token within the decoder to attend comprehensively to all tokens within the encoder.

2.1.3 Pre-trained Language Models

The key bottleneck of deep learning systems is the supervised dataset, which requires human annotation. But this usually requires a huge cost. In order to avoid cost consumption, researchers have turned to exploring self-supervised and unsupervised training, where annotated information can be automatically constructed. Drawing from self-supervised learning paradigms applied to vast corpora, they engage in training models to acquire a universal language representation. These training objectives typically encompass:

- Language Modeling (LM): A quintessential endeavor in probabilistic density estimation, often delineated by auto-regressive LM or unidirectional LM.

- Masked Language Modeling (MLM): Encompassing a Cloze task where certain tokens are masked for prediction.
- Permuted Language Modeling: An undertaking involving language modeling on randomly permuted input sequences. Random permutations are sampled from the entirety of possible permutations. Subsequently, specific tokens within the permuted sequence are designated as targets, and the model is trained to forecast these targets, leveraging the remaining tokens and their natural positions.

Grounded in these pre-training endeavors, researchers have devised various pre-training models, attaining cutting-edge performance across multiple NLP tasks. Broadly, these models can be categorized into three main classes.

Pre-trained Models Based on Encoder-only Transformer

The Encoder-only Transformer model focuses on handling tasks that only input sequences without generating output sequences. Usually, this type of model is very useful when dealing with natural language understanding tasks, such as text classification, sentiment analysis, etc. Unlike traditional Transformer models, they do not include a decoder section, making them lighter and more suitable for tasks that only require encoding the input sequence.

BERT[15], proposed by Google in 2018, is the earliest encoder-only pre-trained transformers. It acquires a universal language representation via unsupervised pre-training on a vast corpus, subsequently amenable to fine-tuning across diverse downstream tasks. BERT, for instance, leverages Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) during pre-training. MLM involves the model predicting randomly masked segments within input sequences, fostering bidirectional language comprehension. NSP tasks the model with discerning the continuity between two sentences, augmenting its grasp of contextual interrelations.

Based on BERT, researchers have proposed many variants, the most famous of which is RoBERTa [66]. RoBERTa, an advancement over the BERT architecture, capitalizes on a larger dataset and extended pre-training duration while discarding the NSP task. These enhancements have propelled RoBERTa to outperform its predecessor across a spectrum of downstream tasks. DistilBERT [110] is a lightweight variant of BERT that achieves model compression through parameter compression and knowledge distillation. Although the model is smaller, it can still maintain high performance on many tasks and has faster inference speed.

In sum, encoder-only transformer architectures have ushered in substantial breakthroughs and innovations within the domain of NLP, furnishing robust tools and technical underpinnings for addressing diverse text-based challenges.

Pre-trained Models Based on Encoder-Decoder Transformer

This kind of pre-trained model includes both encoder and decoder parts and is suitable for conditional language generation, such as text completion and machine translation. So far, researchers have proposed many powerful encoder decoder pre-trained models.

BART [50], proposed by Facebook in 2019, combines the characteristics of bidirectional and auto-regressive in its design, aiming to be suitable for multiple language processing tasks such as text generation, text rewriting, and text summarization. The pre-training objectives of BART include two main tasks: MLM and text reconstruction. Similar to BERT, BART uses MLM tasks during the pre-training phase. In addition, in order to learn and generate continuous text sequences, BART also adopts a text reconstruction task. In this task, the model will be asked to remove a randomly selected small portion from a sentence and attempt to reconstruct the original sentence. This task helps the model learn the ability to generate and rewrite sequences. Through these two pre-training tasks, BART can learn universal language representations and achieve excellent performance on various downstream tasks.

T5 [104] is a generation model proposed by Google in 2020. It converts all of NLP problems into text-to-text conversion tasks and adopts a consistent input-output representation to simplify model design and training processes. Their work demonstrated that scaling up the pre-trained model as well as the training corpus can effectively improve data diversity and the model's memorization ability, thereby continuously improving the model performance of various downstream tasks.

PEGASUS [150], proposed by Google in 2020, focuses on generative tasks such as text summarization, article rewriting, etc. PEGASUS adopts the strategy of reverse Autoregressive to improve generation speed, and employs continuous text reconstructor tasks in both pre-training and fine-tuning stages.

Pre-trained Models Based on Decoder-only Transformer

Decoder-only pre-trained model focuses on processing tasks that only output sequences and do not accept input sequences. Unlike traditional transformers, it only includes a decoder section and is suitable for generative tasks such as dialogue.

The earliest decoder only pre trained model was GPT [101], proposed by OpenAI in 2018. It adopts a standard auto-regressive LM objective. With the large-scale pre-training on BookCorpus data, it presents good performance on various tasks, such as question answering, and text completion. Later, OpenAI develops more powerful GPT-family models, including GPT2 [102] and GPT3 [6]. The pre-training data of GPT2 includes large-scale web data collected from various sources on the Internet, including news articles, web content, e-books, Wikipedia, etc. These data span various themes and fields, making the model more widely applicable to different language comprehension tasks. Many experiments show that GPT2 has learned a large amount of language knowledge and is able to generate a universal text representation, making it perform well on vast NLP tasks.

The parameter count of GPT3 can reach 175B, making it one of the largest and

most powerful language models to date, with high flexibility and wide application capabilities. Compared to GPT2, GPT3 uses a wider and more diverse range of training data, which enables the model to better understand and generate various types of text. However, GPT3 has not yet been open-source, and its training corpus and details have not been disclosed. In order to break the monopoly of OpenAI, researchers have invested a lot of effort to replicate GPT3, and hence have developed many excellent open-source models as a result, such as OPT [152], BLOOM [48], LLAMA [124], GLM [148], etc. Although these pre-trained large models have good zero-shot performance, there is a risk of generating harmful information when directly using them for downstream tasks. To avoid this issue, [86] proposes reinforcement learning from human feedback (RLHF). This enable large models to generate more appropriate outputs with less toxicity information [86].

2.2 Narrative Commonsense Reasoning

Generally, narrative commonsense reasoning is a relatively large problem that is related to multiple sub-problems, such as commonsense causal reasoning, narrative understanding and generation, event causalities identification from narrative text, counterfactual story generation, narrative coherence learning, contrastive narratives generation, etc. In this segment, we furnish an elaborate overview of the literature pertinent to narrative commonsense reasoning.

2.2.1 Commonsense Causal Reasoning

Commonsense causal reasoning refers to the ability of people to infer causal relationships based on daily experience and commonsense. This ability is crucial for understanding the world, making decisions, and solving problems. Existing commonsense causal reasoning methods generally collect causal word pairs to construct causal

estimators. [108] leverages personal stories as a source of causality and uses PMI between word pairs to identify the causal pairs with high correlation. [116, 140] create causal embedding based on causal word pairs which are extracted using hand-crafted causal rules. Then these methods use the constructed causal estimators to reason causality. [59] proposes a guided beam-search technique to generate causes and effects based on the pre-constructed large-scale causal graph. However, these works model causality in terms of word pairs and consequently they are of great limitation in causal reasoning. Recently researchers begin to paid more attention to event-level causal reasoning. [155] cluster the observed similar events together and connect these clusters in event causality networks, based on which the future events are predicted. [111] introduce ATOMIC, an compendium of everyday commonsense causal reasoning, structured around 877k textual descriptions of inferential knowledge. ATOMIC centers on inferential knowledge categorized as typed if-then relationships. By training on ATOMIC, [37] proposes CoMeT, a generative neural model that can generate cause or effect according to the input event. In chapter 3, we target event-centric effect generation. By selecting the guided effect events, we further realize the predicted event skeletons into the full sentences to fill in the missing information in the skeletons.

2.2.2 Event Causality Identification from Narrative Text

Event causality Identification (ECI) is a fundamental task in NLP because causality between events can be used in many applications. Early works [155, 140, 59] typically employed rule-based methods to identify causal relationships between events. Each rule follows the template <Pattern, Constraint, Priority>, where Pattern is a regular expression containing the selected keywords, Constraint is a syntactic constraint on sentences that can apply the pattern, and Priority is the priority of the rule when matching multiple rules. These rules have been carefully designed by humans, demonstrating high extraction precision, but face the problem of insufficient cover-

age. Later, researchers turn to model-based recognition methods. They usually train models based on supervised datasets. The models learn potential causal signals, i.e., causal patterns and causal event pairs, from data, thereby gaining the generalization ability. Based on the basic supervised learning framework, researchers have proposed different incremental strategies. [64] advocates a mention masking generalization mechanism for acquiring event-agnostic yet context-specific causal patterns. [162] devises a self-supervised framework to glean context-specific causal patterns from external causal statements. From the perspective of data augmentation, [164, 163] use possible causal event pairs to find potentially useful data from the external corpus. [20, 93] propose graph-based methods for document-level ECI. Different from these works, we are the first to notice the biased inference problem in supervised ECI, and we have proposed a corresponding de-bias method to improve the precision of ECI.

2.2.3 Narrative Understanding and Generation

Given the input text, narrative understanding and generation require models to produce fluent and coherent narrative output text under predefined conditions. Due to the high demand for commonsense knowledge in narrative generation, researchers have invested a lot of energy in knowledge-enhanced narrative generation. In terms of knowledge utilization, some researchers have adopted a continuous training method based on pre-trained models to inject knowledge into pre-trained models. Another group of researchers has adopted explicit knowledge graphs based reasoning methods.

Injecting Domain Knowledge into Pre-trained Models

The recent development of PLMs, such as BERT[15] and BART [50], is seeing newfound success in the NLP field. These pre-trained models demonstrate strong knowledge extraction and memory capabilities. Therefore, researchers attempt to use pre-trained models as carriers to carry commonsense knowledge. They designed different

continuous training tasks to inject common sense knowledge into pre-trained models. For example, [33] injects external knowledge into language models for cause-effect relation classification. [28] continually trains PLMs by predicting missing events in temporal event sequences to focus on narrative event reasoning. Meanwhile, [63] uses BART to reconstruct a temporally-disorganized event sequence to focus on narrative event reasoning. [158] extracts eventuality knowledge by discourse connectives, then uses the knowledge to train PLMs for event correlation reasoning. [57] injects causal sentences into PLMs for commonsense causal reasoning tasks. In chapter 5, we extend the work of [57] by injecting sentence-level causalities into PLMs. In addition, we also utilize event-level causalities for narrative generation.

Knowledge Graph Grounded Narrative Generation

Due to the lack of knowledge in neural networks, researchers focus on providing external structural knowledge as background for narrative reasoning. Earlier works focus on grounding reasoning on concepts or entities knowledge graphs. For example, [60] leverages structural commonsense knowledge graphs to conduct interpretable reasoning for answering commonsense questions. [39] introduces explicit knowledge from ConceptNet for narrative story generation. [123] enhances contextual word representations using neighboring entities in knowledge graphs. These works prove that external knowledge helps to enhance the performance of narrative generation systems. However, these methods are all based on word or entity-level knowledge graphs, facing the problem of semantic distortion when solving multi-word expressions, since word or entity-level knowledge has low-level semantics and cannot express multi-word text units. As the basic semantic unit of natural language, an event carries richer information than a single word, hence self-contained event knowledge might help the narrative reasoning and generation. Therefore, recent works focus on utilizing structural event knowledge. For example, [88] encodes structured event knowledge with a transformer-based model for narrative commonsense reasoning. [62] integrates event

sequence knowledge for story writing. Although these methods have made progress, they overlook the sparsity of events. Different from these works, we propose to split a coarse-grained event into fine-grained word components to obtain the hierarchical knowledge graph, making it possible to mitigate the event sparsity problem.

2.2.4 Counterfactual Story Generation

Counterfactual reasoning in narratives (CRN) refers to the process of predicting potential outcomes that could have arisen from alternative events, diverging from what actually occurred [98, 2]. Existing works for counterfactual story generation mainly include unsupervised methods or supervised fine-tuning.

The earliest several unsupervised methods was proposed by [98], In their original paper, [98] proposed to first train the story generation models on story generation datasets, then adapted the models on counterfactual story generation task in a zero-shot manner. However, the performance of zero-shot evaluation is unsatisfactory. [99] proposed DELOREAN, an unguided extrapolation technique adept at flexibly assimilating antecedent and subsequent contexts utilizing solely off-the-shelf, dexterous, left-to-right linguistic models, devoid of any guiding oversight. This approach ingests the narrative premise and counterfactual condition as input sans any form of dataset-specific training. Subsequently, [12] introduced EDUCAT, an editing-centered unsupervised technique tailored for counterfactual narrative rewriting. This method encompasses a discernment mechanism for identifying target positions and a transformative maneuver. [12] regards the problem as a controllable text generation task, and adopted Metropolis-Hastings sampling for iteratively edit the original story ending to expected counterfactual story ending. At each step, this method determines whether to modify, including insert, delete, and replace actions, the token at the current position. The decision is made by the probability of predefined evaluation functions. After continuous iteration, the original ending will be modified to a coun-

terfactual ending that is coherent to the counterfactual condition. Similar to [12], [100] also treat the problem from a perspective of controllable text generation. They advocate an energy-based constrained decoding methodology, drawing on insights from [49] and incorporating Langevin dynamics as [134]. This approach harmonizes constrained generation by delineating constraints via an energy function. However, the decoding speed is very slow, which limits the efficiency of their method.

Overall, unsupervised methods for counterfactual story generation have poor performance, so some researchers have proposed several supervised learning methods. [98] was an early proposer of supervised training methods, which simply finetuned pretrained models using annotated data. Following, [12, 53] proposed two-stage approaches. Typically, during the initial phase, each token within the original story conclusion undergoes scrutiny to ascertain whether modification is warranted. Subsequently, in the subsequent stage, the earmarked terms are adjusted to harmonize with the narrative logic prescribed by the counterfactual circumstance. However, the two-stage methods are dataset-specific, it is difficult to migrate this dataset-specific framework to other datasets [2]. In addition, supervised methods face the risk of exploiting artifacts of the dataset [98], making these methods sub-optimal.

Counterfactual reasoning targets to explore the causal relationships in the data [90, 144]. Recently, there has been a strong interest in equip the current text generation with counterfactual reasoning ability [91]. These works involve fields such as dialogue generation [85], machine translation [65], style transferring [34], etc. Yet there have been few works that apply causal perspective to counterfactual reasoning in narratives. In chapter 6, we use the idea of counterfactual inference to alleviate the spurious correlation issue in ECI. In addition, we propose additional strategies to improve the causality between the counterfactual condition and the generated counterfactual outcome. To the best of our knowledge, this is the first work which reviews counterfactual narrative reasoning from a causal view. We believe that our work can bring new insights to this field.

2.2.5 Narrative Coherence Learning

Narrative coherence learning aims to evaluate the coherence between the (input, output) narrative text pairs, which is a key challenge in the field of narrative reasoning. Narrative coherence evaluation requires high generalization ability of neural models, so supervised learning methods are not suitable for this task due to the fact that supervised methods have difficulty in adapting to out-of-domain data. In order to obtain models with good generalization, researchers mainly focus on self-supervised contrastive learning methods. These methods generally devise self-supervised tasks, in which positive samples are from large-scale real narratives [145, 75], and negative samples are created by sampling-based strategies. For example, [138] presents three self-supervised learning tasks aimed at transferring the narrative-level knowledge from ROCStories into the backbone model, comprising vanilla BERT and the Multi-Choice Head architecture. [158] randomly masks an event in the event sequences to create negative samples. [7] also adopts event-masking strategy to create negative samples. In addition, they propose the event-shuffling strategy which randomly shuffle a ordered event sequence into the disordered sequence, then the disordered sequences are treated as negative samples. [47] incorporates randomly sampled sequences and model-completed [102, 6] sequences as negative samples. However, these strategies are generally coarse-grained and superficial. The resulting negatives still face problems of low quality, such as being irrelevant or repetitive [47], making them less representative, and easily distinguishable. To mine more-qualified negative examples, researchers are devoted to developing methods for mining hard negatives and negative samples, which are more difficult for neural models to distinguish, therefore benefiting narrative coherence learning. For example, [40] retrieves hard negatives from the corpus with a momentum encoder. [149, 41] proposes the Mixup strategy that mixes different negatives in latent space to create hard negatives. [151] mixes multiple positive samples to produce hard negatives. These works motivate us to develop more sophisticated method for mining hard negatives. In chapter 7, we propose

to crisscross an observed narrative with its contrastive counterparts for synthesizing hard negatives. Since the contrastive narratives are similar to the original ones, we can obtain qualified negatives, which are similar to the real narrative but actually less coherent.

2.2.6 Contrastive Narratives Generation

Contrastive examples are data points that are close in the hidden space, i.e., share similar embedding representations, but the model produces different predictive likelihoods [69]. More specifically in the field of narrative reasoning, contrastive narratives are examples that are similar in content, but convey different semantics. For example, the counterfactual variants [98] of an observed narrative story can be seen as a kind of contrastive narrative. Due to the similarity between observational narratives and their contrasting variants, we can use them to synthesize high-quality negative samples. To obtain contrastive examples, researchers [69, 132, 1] have proposed different approaches. For example, [69] selects unlabeled data points from the data pool, whose predicted likelihoods differs the most from their neighbors in the training set. In the field of language, there has recently been a trend [137, 16, 109] towards producing counterfactual explanations. These counterfactual explanations are similar in content but present different labels, and therefore can be generally regarded as contrastive examples. These methods generally produce textual outputs conditioned on pre-defined control codes derived from semantic representations, allowing for flexible perturbation strategies. However, these methods generate contrastive samples based on the (example, label) paired data. Due to the emphasis on the relationships between (prefix, suffix) pairs of narratives, these methods for obtaining contrastive examples are no longer applicable. To solve these problem, we innovately adopt the Brownian bridge process [133] for contrastive narratives generation because the Brownian bridge allows for the smooth modeling of gradual changes between two narrative states. Based on the simple constraint, we are able to generate coherent contrastive

narratives, which are used to synthetic hard negatives.

Part I

Automatically Causality Mining and De-biasing

Narrative causality is one of the core concepts of human society. The performance of many artificial intelligence applications depends on a deep understanding of logical knowledge. However, existing neural network models typically use annotated corpora for training with maximum likelihood estimation and mechanically memorize frequent patterns in the corpus, thus lacking a profound understanding of narrative logic and making reasonable analysis and judgments. To solve this problem, it is necessary to use external causal knowledge to assist the model. Due to the widespread occurrence of causality in narrative texts and its importance as the most important logical relationship in human society, we take causality as the starting point and explore narrative reasoning methods which are enhanced causal knowledge.

To obtain causal knowledge, we first investigate rule based system to extract causlity automatically. We group event-level causalities into an event causality network and demonstrate its use in the task of narrative effect generation (Chapter 3). Then, we investigate the task of event causality identification (Chapter 4) to mitigate the noose problem in causality extraction..

Chapter 3

Narrative Effect Generation Based on Causal Reasoning

3.1 Introduction

In this chapter, we propose rule-based causality extraction system, and demonstrate its use in the task of cause-to-effect narrative generation.

Causal reasoning is the process of observing an action and reasoning future scenarios that may be potentially caused by it [103]. Its importance is reflected in effective narrative reasoning, which places high demands on the understanding of causal relationships. Earlier causal reasoning methods [108, 67] collect causally related word pairs (e.g., *earthquake→tsunami*) to build the statistical models of causality, and then predict effects words for given cause words. Recently, [140] uses causal embedding to predict possible effect words of the input causes. [59] proposed the lexically-constrained beam-search to generate possible effects given provided word guidance. However, all these methods tend to reason causalities at word-level.

Causalities between word pairs are not always self-contained (i.e., intelligible) when

they are extracted without the context [30]). For example, "*quarrel→break*" is not self-contained since this is not intelligible without the context: "*They always quarrel→They break up*". Given the action "*They always quarrel*", word-level causal reasoning methods will give the effect of "*break*" conditioned "*quarrel*". In other words, word-level causal reasoning may give inconsistent predictions about causality. Considering this deficiency, a better way is to use causal events to enhance causal reasoning [103, 155]. An event is a tuple containing a subject, a verb, a direct object, and some additional disambiguation token(s) [141]. As the semantic unit of natural language, an event carries richer information and describe a more specific scenario than a single word, hence causal information between event pairs is self-contained, which can maintain the causal consistency between the input and the inferred result. For example, the agent can predict "*They break up*" according to the observed event "*They quarrel*". However, an observed causal event is very likely to appear only once, which brings about huge sparsity to causalities and great difficulty to the event-level causal reasoning. To solve this problem, we design lexicon based abstraction rules to structuralize observed causal events into a hierarchical event causality network where similar events are clustered together. This allows us to mitigate the sparsity of events. As such, we are able to predict the most reasonable effect event based on the event causality network. The predicted effect event contains the skeleton information, with the detailed context information neglected in the event extraction process. So we further use the predicted effect event as a template to generate an effect sentence in order to fill in the missing information.

In this chapter, we propose a novel event-level reasoning method and demonstrate its use in the task of narrative effect generation. First, given a cause sentence, a causal-related event graph is extracted from the constructed event causality network and is encoded with a graph attention mechanism in order to support better reasoning of the potential effect. The most probable effect event is then retrieved from the causal event graph and is used as template for the effect generation. Next, for the input

cause sentence, we develop a effect generator to generate the corresponding effect sentence. We use the retrieved effect event as a template for the generation process. The problem now is that all tokens of the retrieved event should be contained in the generated sentence. Otherwise, the part of causal information carried by the retrieved event will be lost, leading to the incompleteness of the expected causal relation. Our generator realizes the retrieved event into a sentence in a way analog to sentence completion. Hence, our generator guarantees that the generated sentence retains all tokens of the retrieved effect event.

To sum up, this paper makes the following contributions.

- we devise an effect generation method which is based on causal event reasoning (EGCER) to generate effect sentences for given input cause sentences.
- We present a method to construct an event causality network, from which we can obtain a causal subgraph to facilitate effect event retrieval and event-template-based effect sentence realization.
- Empirical results on the widely used English Wikipedia and COPA corpora show that our model achieves the best performance compared among various well-designed baselines.

3.2 Event Causality Network Construction

In this paper, we use causal events to bridge the causalities between input sequences and generated sequences. Hence, we must first collect sufficient cause-effect sentence pairs so that from each sentence pair a cause-effect event pair can be eventified. In this section, we explain how we construct the event causality network, including the steps of causality mining and event structuralization.

3.2.1 Causality Mining and Event Eventification

In Chapter 4, we develop the supervised method for identifying event causalities. However, due to the small size of the datasets, our supervised method face difficulties in scaling up event causalities extraction. To avoid this problem, we instread adopt the rule-based extraction system. Following [67], we make use of a few high-precision causal rules to extract cause-effect sentence pairs. Each rule follows the template of (*Pattern*, *Constraint*, *Priority*), where *Pattern* is a regular expression containing a selected connector, *Constraint* is a syntactic constraint on the sentence to which the pattern can be applied, and *Priority* is the priority of a rule when there are more than one rule matched. To ensure there is a causal event which can be eventified from the matched causal sentence, only causal connective patterns are adopted, as shown in Table 3.1. Then we extract causal event pairs from causal sentence pairs based on dependency analysis. Specifically, we adopt the commonly used 4-tuple event representation (s, v, o, m) [94] where v denotes the verb, s denotes the head noun of the subject, o denotes the head noun of the direct object or the adjective, and m denotes the head noun of the prepositional or indirect object. Any of these components excluding the verb can be $\emptyset$, denoting the absence of the corresponding component. We stipulate that the number of the valid components of an event must be no less than two.

Adopted Causal Connectives		
C, as a consequence E	E, because C	C, so that E
C, as a result E	C, therefore E	C, hence E
C, consequently E	Because C, E	C, thus E

Table 3.1: C denotes the cause, and E denotes the effect.

3.2.2 Events Structuralization

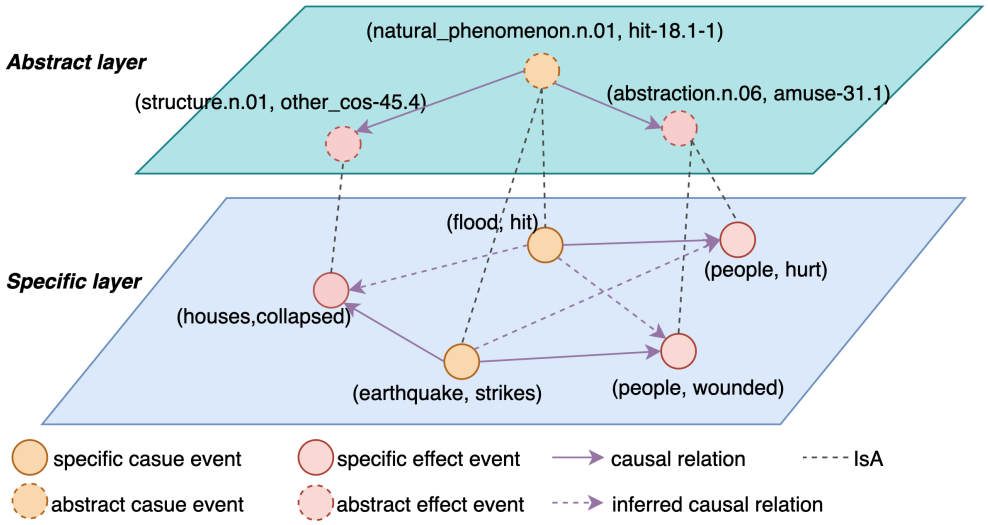


Figure 3.1: Our hierarchical event causality network.

We structuralize the extracted causal event pairs with a hierarchical event causality network, as illustrated in Figure 3.1, in which the specific causal events are generalized to their abstract representations, and similar events in the specific layer are clustered together in the abstract layer. Specifically, the verb in each event is generalized to its class in VerbNet [115]. The other components are generalized by the WordNet [72] synset two levels up in the inherited hypernym hierarchy. If a noun is a named entity, it is replaced by its NER category. An event is represented in the abstract layer by the **f**requent **t**rigram **t**uple (FTT) of its abstract representation, where the FTT of an event refers to the most frequent one among the abstract tuples of (s, v, o), (s, v, m) and (v, o, m). The events that have the same FTT are merged into the same abstract class. The edges in the abstract layer are generated corresponding to those in the specific layer. As shown in Figure 3.1, since there is a causal relation from *(earthquake, strikes)* to *(houses, collapsed)*, an edge from *(natural_phenomenon.n.01, hit-18.1-1)* to *(structure.n.01, other_cos-45.4)* is created.

In addition, we explicitly use the similarity-based inferring rule to extend causalities

from the abstract layer to the specific layer. For example, if a causal relation holds from *(earthquake, strikes)* to *(houses, collapsed)*, and both *(earthquake, strikes)* and *(flood, hit)* belong to the same abstract class, then it is most likely to conclude that there may be a causal relation from *(flood, hit)* to *(houses, collapsed)*. Such a manipulation significantly reduces the sparsity of causalities at the specific-layer, and hence supports better reasoning about the effect events.

3.3 Effect Generation

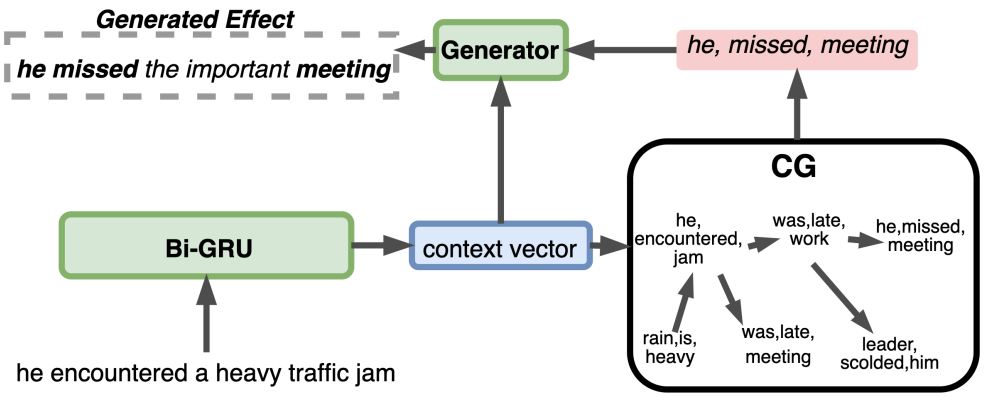


Figure 3.2: The overview of EGCER.

3.3.1 Task Description

Given a cause sentence $X = \{x_1 x_2 \cdots x_m\}$, the goal of effect generation is to generate a proper sentence $Y = y_1 y_2 \cdots y_n$ that conforms to causality with X. To maintain consistence of causality between X and Y, we decompose this task into two steps. Given a sentence X, and a causal subgraph $CG = \{e_1, e_2, \cdots, e_{N_{CG}}\}$, which consists of a set of events $\{e_j = (s_j, v_j, o_j, m_j)\}$ $(j = 1, \cdots, N_{CG})$ as nodes, the task in the first step is to retrieve an effect event e_Y from CG according to X. In the second step, e_Y is used as the template to generate the Y, which is required to retain all

tokens of e_Y. Essentially, the model estimates the probability of:

$$P(Y|X, CG) = P(Y|e_Y, X)P(e_Y|X, CG) = P(e_Y|X, CG)\prod_{t=1}^{n} P(y_t|y_{<t}, e_Y, X). \quad (3.1)$$

The overview of the proposed EGCER is illustrated in Figure 3.2. EGCER takes a cause sentence X and causal subgraph CG as the input to retrieve an event e_Y, which has a causal relation with X. A graph-based reasoning mechanism is used to enhance the event retrieval. Then EGCER realizes e_Y to the effect sentence Y by using e_Y as the template that ensures Y contains all the tokens of e_Y.

3.3.2 Causality Aware Effect Event Retriever

The causality-aware effect event retriever consists of a cause sequence encoder that encodes X and a casual graph encoder that helps derive e_Y.

Cause Sequence Encoder

We implement the sequence encoder using a bidirectional GRU model [13]. It reads the sequence $X = \{x_1 x_2 \cdots x_m\}$ from both directions and computes hidden states for each token:

$$\overrightarrow{\mathbf{h}_{x_i}} = \overrightarrow{\mathbf{GRU}}(\overrightarrow{\mathbf{h}_{x_{i-1}}}, \mathbf{e}(x_i)), \overleftarrow{\mathbf{h}_{x_i}} = \overleftarrow{\mathbf{GRU}}(\overleftarrow{\mathbf{h}_{x_{i+1}}}, \mathbf{e}(x_i)), \quad (3.2)$$

where $\mathbf{e}(x_i) \in \mathcal{R}^d$ is the embedding of the word x_i, d is the size of embeddings. The final hidden representation of the $i-$th token is $\mathbf{h}_{x_i} = [\overrightarrow{\mathbf{h}_{x_i}}; \overleftarrow{\mathbf{h}_{x_i}}]$, where $[\cdot;\cdot]$ denotes a concatenation operation. The context vector of X is $\mathbf{H}_X = \{\mathbf{h}_{x_1}, \cdots, \mathbf{h}_{x_m}\}$.

Causal Graph Encoder

The event causality network embodies thousands of nodes. Most of the nodes are not relevant to a particular cause sentence. It is thus unrealistic to directly encode

the whole event network with graph neural networks (GNNs) [46, 131]. Instead, we merely retrieve a subgraph from the event network according to the cause sentence. Specifically, we match the FTT of the cause sentence at abstract-layer. Once the FTT is matched, its L-hop neighbors together with itself is preserved. The abstract subgraph is the abstract representation of causalities, which is not enough to predict real-world scenarios. So, we transfer the abstract subgraph to the specific layer to obtain the specific causal subgraph CG. The weight of an edge in CG is derived by the following rules:

- If the edge between the event pair (e_i, e_j) is extracted from the dataset, the weight w_{ij} of this edge is $w_{ij} = 1$.

- If the edge of (e_i, e_j) is inferred based on the similarity between (e_i, e_k) and the causal relation between (e_k, e_j), we have $w_{ij} = sim(e_i, e_k)$, where $sim(e_i, e_k)$, calculated by the path-similarity measure in WordNet, is the similarity score between e_i and e_k.

The causality graph is the key component of our reasoning framework. Given a causal subgraph CG, a GNN module with the graph attention mechanism is used to model the causal interactions among multi-hop neighbor nodes in order to reason the most reasonable effect event with regard to X. Specifically, the causality GNN module works as follows.

Learning Initial Event Representations The initial representation of an event in CG is learned by composing word embedding of its verb and arguments. Given an event $e_i = (s_i, v_i, o_i, m_i)$ and the word embedding of its verb and arguments $\{\mathbf{e}_{s_i}, \mathbf{e}_{v_i}, \mathbf{e}_{o_i}, \mathbf{e}_{m_i}\}$, the initial representation of e_i is represented by: $\mathbf{h}_{e_i} = [\mathbf{e}_{s_i}; \mathbf{e}_{v_i}; \mathbf{e}_{o_i}; \mathbf{e}_{m_i}]$. Absent event arguments are represented by zero vectors.

Updating Event Representations using a GNN The initial representation of an event takes no account of causal relationships between events. However, the neighborhood information in the CG represents the causality tendencies, which are especially useful for reasoning the most reasonable effect event e_Y. We use a simple GNN to capture the neighborhood information. The intuition of applying the GNN to CG is (1) to contextually refine event vectors and (2) to capture multi-hop causal relationships for generation. Specifically, on the first layer of the GNN, the hidden vector of the event $e_i \in CG (i = 1, \cdots, N_{CG})$ is initialized by its initial representation ($\mathbf{h}^0_{e_i} = \mathbf{h}_{e_i}$). The l-th layer's vectors of e_i and its neighboring nodes are then pooled to obtain the vector of e_i on the $(l+1)$-th layer with a non-linear activation σ:

$$
\begin{aligned}
\mathbf{z}_i^{(l)} &= \mathbf{W}^{(l)} \mathbf{h}_i^{(l)}, \\
\gamma_{ij}^{(l)} &= \text{LeakyReLU}(w_{ij}(\mathbf{z}_i^{(l)} \cdot \mathbf{z}_j^{(l)})) \\
\alpha_{ij}^{(l)} &= \frac{exp(\gamma_{ij}^{(l)})}{\sum_{k=1}^{N_{CG}} exp(\gamma_{ik}^{(l)})} \\
\mathbf{h}_i^{(l)} &= \sigma(\sum_{j=1}^{N_{CG}} \alpha_{ij}^{(l)} \mathbf{z}_j^{(l)}),
\end{aligned}
\tag{3.3}
$$

where σ is defined to be ReLU, $\mathbf{W}^{(l)}$ is a parameter, $\cdot$ denotes the inner product of the two vectors, w_{ij} is the weight of the edge (e_i, e_j), and α_{ij}^{l} can be deemed as the l-th layer's causal score between the graph nodes i and j. The final hidden vector $\mathbf{h}_i^{(L)}$ $(i = 1, \cdots, N_{CG})$ of events are used to select the guided effect event e_Y.

Select Guided Event from Causal Graph Given the hidden state $\mathbf{h}_i^{(L)}$ of the event e_i $(i \in [1, \cdots, N_{CG}])$ and the hidden state $\mathbf{H}_X = \{\mathbf{h}_{x_1}, \cdots, \mathbf{h}_{x_m}\}$ of the cause sentence X, the causal score between each candidate event $\mathbf{h}_i^{(L)}$ and X is calculated by $cs_i = \mathbf{h}_i^{(L)} \cdot h_X$, where $\mathbf{h}_X = \frac{1}{m} \sum_{k=1}^{m} \mathbf{h}_{x_k}$ is the mean-pooling representation of X. In the training phase, the ground-truth effect event e_Y is used as the guided event. In the test phase we choose the guided event by $e_Y = \max_i cs_i$, which has the maximum

causal score in relation to the sequence X. The retrieved event will be used as a template to generate the effect sentence.

3.3.3 Event Template based Effect Generator

The effect event e_Y retrieved by the event retriever is the expected causality prediction of the cause sentence. We should guarantee that all tokens of e_Y will be included in the final output sequence to maintain the completeness of the expected causal relation. Inspired by [76, 70], we propose to use retrieved effect events as templates and expand these templates into more complete sentences. Specifically, given the effect event template $e_Y = (s, v, o, m)$, the resulting sentence would be $[\_s][\_v][\_o][\_m]$, where blanks indicate where words should be added in order to make a sentence richer in content. In accordance with 4-tuple event representation, the generator consists of 4 decoders. Taking the verb component as an example, the verb decoder is responsible for filling in the blank before the event token v. At each decoding time-step, the attention mechanism is adopted to attend to the context vector of X when generating a new word. Specifically, for the verb decoder, the hidden state $\mathbf{s}_t^v$ at time-step t is

$$\begin{aligned}
\mathbf{s}_t^v &= \mathbf{GRU}^v(\mathbf{s}_{t-1}^v, [\mathbf{e}(y_{t-1}^v); \mathbf{c}_t^v]) \\
\mathbf{c}_t^v &= \sum_{i=1}^{m} \alpha_{ti}^v \mathbf{h}_{x_i} \\
\alpha_{ti}^v &= \frac{\exp(\beta_{ti}^v)}{\sum_{j=1}^{m} \exp(\beta_{tj}^v)} \\
\beta_{ti}^v &= \mathbf{v}_\alpha^\top \sigma(\mathbf{W}_\alpha [\mathbf{s}_{t-1}^v; \mathbf{h}_{x_i}]),
\end{aligned} \tag{3.4}$$

where y_{t-1}^v is the ground-truth word at time-step $t-1$ for the verb decoder, $\mathbf{c}_t^v$ is the attended context vector of the sequence X, $\mathbf{v}_\alpha$ and $\mathbf{W}_\alpha$ are shared parameters among 4 decoders, σ is an activation function (tanh by default). The probability $p(y_t^v)$ of

generating a gold token y_t^v for the verb decoder at time step t is formulated as:

$$\begin{aligned} p(y_t^v|y_{<t}) &= \frac{\exp(\mathbf{s}(y_t^v|s_t^v, c_t^v))}{\sum_i \exp(\mathbf{s}(y_i^v|s_t^v, c_t^v))} \\ \mathbf{s}(y_t^v|s_t^v, c_t^v) &= \mathbf{w}_n^\top \tanh(\mathbf{W}_o[\mathbf{s}_t^v; \mathbf{c}_t^v]) \end{aligned} \tag{3.5}$$

where $\mathbf{W}_o, \mathbf{w}_n$ are the parameters shared among 4 decoders. The decoders of the other event components work in the same way. To ascertain the model's confidence for each effect generation, we aggregate the loss incurred after the generation of each token, normalizing it by the sentence length. Subsequently, the generated segments are concatenated to yield the ultimate effect sentence.

3.3.4 Training Objective

The objective of the event retriever is to minimize the sum of the negative log-likelihood (NLL) losses of all samples:

$$\begin{aligned} J_R(\theta) &= -\log p(e_Y|X, CG) \\ &= -\log \frac{\exp(cs_*)}{\sum_{j=1}^{N_{CG}} \exp(cs_j)}, \end{aligned} \tag{3.6}$$

where θ denotes the model parameters, cs_* denotes the causal score of the ground-truth effect event e_Y with regarding the cause sentence X. For the generator, the objective is to maximize the estimated probability of the ground-truth effect sequence. We use the NLL of it as the loss function:

$$J_G(\theta) = P(Y|e_Y, X) = \sum_t -\log p(y_t|y_{<t}). \tag{3.7}$$

Statistics	Training	Validation	Test
Count	79K	9.6K	9.8k
AvgSentLen	7.91	8.03	8.04
AveEventLen	2.86	3.01	3.01

Table 3.2: The statistics of the English Wikipedia. AvgSentLen and AveEventLen mean the average sentence length and event length.

3.4 Experiments

3.4.1 Datasets

English Wikipedia[1]: We extract cause-effect sentence pairs from the English Wikipedia corpus, split all pairs into training/validation/test, and tune hyper-parameters on the validation data. The training data is used to construct the event causality network. We retrieve 2-hop causal subgraphs according to input cause sentences. The percentage of the test samples whose gold effect events exist in the retrieved causal subgraphs is 70.8%. The statistics is presented in Table 3.2.

COPA Benchmark: The *Choice of Plausible Alternatives* (COPA) [108] dataset consists of 1,000 multiple-choice questions (500 for validation and 500 for testing) requiring causal reasoning in order to answer correctly. Each question is composed of a premise and two alternatives, and the task is to select a more plausible alternative as a cause (or an effect) of the premise. We only use the most plausible alternative and its premise to collect cause-effect sentence pairs, on which we can perform and evaluate effect generation. We use the COPA causes to retrieve causal subgraphs from our event causality network. Finally, 186 COPA pairs with their corresponding causal subgraphs are obtained, leading to the average sentence length of 4.77 and the

[1]https://dumps.wikimedia.org/enwiki/20201020/enwiki-20201020-pages-articles.xml.bz2

average event length of 2.83. The percentage of the samples whose gold effect events exist in causal subgraphs is 11.2%. Because there is no released training data for the COPA task, we train all models on Enwiki and evaluate them on COPA.

3.4.2 Implementation Details

Our retriever consists of a 2-layer bidirectional GRU for encoding input sequences and a 2-layer GNN for updating event representations. The retriever and the generator have their own separate parameters, and their hidden sizes are set to 512. The word embedding size is 300. We use the Adam optimizer with the mini-batch size of 96. The learning rate is 0.001.

3.4.3 Baselines

We compare our method with state-of-the-art text generation methods, including GPT2 [102], BART[50], CopyNet[160] and CausalBERT[59]. We concat cause-effect sentence pairs and finetune GPT2-base in a language model setting. BART-base is finetuned with the encoder-decoder setting. Both GPT2 and BART are implemented by *transformers*[2]. CopyNet employs the copy mechanism which either copies tokens from the retrieved event or generates words from the vocabulary. CausalBERT employs the lexically-constrained beam-search to generate possible effects for provided word guidance. ConceptNet[118] is used to retrieve causal relevant constraints for CausalBERT.

3.4.4 Evaluation Metrics

For automatic evaluation, we use metrics including BLEU-4 [87], Distinct-n [54] to evaluate the generated effect sentences. Abstraction-Matching (AbsMat) evaluates

[2]https://huggingface.co/

the percentage of the generated effect sequences that have the same abstraction as the corresponding gold effect sequences.

For the manual evaluation, we examine whether the generated sequence is a plausible effect of the input, which is denoted as *plausibility* (Plau.). Specifically, 100 samples are randomly selected from the Wikipedia test set and COPA, respectively, and distribute them to the two graduate students from the NLP field. Each student is asked to give a score from {0, 0.5, 1} for the (input, generation) pair, given the following guidelines. Assign 0 to the pair if the generation can never be considered as a possible effect of the input, assign 0.5 to the pair if the generation is a possible effect of the input but has certain grammatical errors and assign 1 to the pair if the generation is a possible effect of the input and there is no grammatical error. We average scores over the two annotators. The *cohen's kappa* scores on Wikipedia and COPA are 0.65 and 0.63, respectively.

3.4.5 Result and Analysis

Result: The automatic evaluation result is shown in Table 3.3, where EGCER achieves the best results. BART performs better than GPT2 due to the adopted encoder-decoder architecture. Based on the event skeletons provided by the effect event predictor, CopyNet and EGCER are aware of the topic which should be generated, and hence perform better than BART and GPT2. CopyNet performs worse than EGCER because CopyNet cannot cover all tokens of the retrieved event, as a result, the causal information in the generated sequence is incomplete. CausalBert performs worse than EGCER because it is based on the word-level causal analysis, which can also be found in Section 3.4.7. Given the effect event, EGCER sees a more complete skeleton, hence generate a more reasonable effect sentence.

The result of the manual evaluation is also shown in Table 3.3. As for EGCER, we find that it may sometimes generate negation expressions or grammatical errors, as

English Wikipedia				
Model	BLEU-4	Distinct-1/2	AbsMat	Plau.
GPT2	0.69	5.57/16.82	0.3	0.08
BART	1.28	8.23/24.83	1.7	0.11
CausalBERT	0.74	5.33/22.23	8.5	0.12
CopyNet	2.85	10.63/39.82	16.4	0.17
EGCER(ours)	4.90	13.99/43.58	26.4	0.27
COPA				
Model	BLEU-4	Distinct-1/2	AbsMat	Plau.
GPT2	1.35	22.61/44.25	0.2	0.02
BART	1.22	22.37/43.71	0.5	0.04
CausalBERT	0.92	22.39/52.56	3.7	0.06
CopyNet	1.18	32.74/75.17	2.6	0.04
EGCER(ours)	1.74	48.08/83.97	5.3	0.07

Table 3.3: Automatic and manual evaluation results.

a result, the generated sequence is not a plausible effect even if the retrieved event is plausible. The proportion of the generated sequences in this case is about 21%. We speculate that the errors in data preprocessing and the insufficiently powerful generator are the possible reasons. In the future, we will further improve generators in order to generate more high-quality effect sentences. It can also be found that EGCER performs far worse on COPA than on Enwiki, this is because a great gap exists between these two datasets. However, EGCER is still superior to any other model, which demonstrates event-level causal reasoning contributes to the effect generation.

3.4.6 Ablation Study

Models	BLEU-4	Distinct-1/2	AbsMat	Plau
Full model	4.90	13.99/43.58	26.4	0.27
w/o weights	4.37	14.10/42.86	23.3	0.24
w/o 2nd layer	3.89	13.15/41.56	20.6	0.21
w/o GNN	2.89	13.00/42.02	18.3	0.19

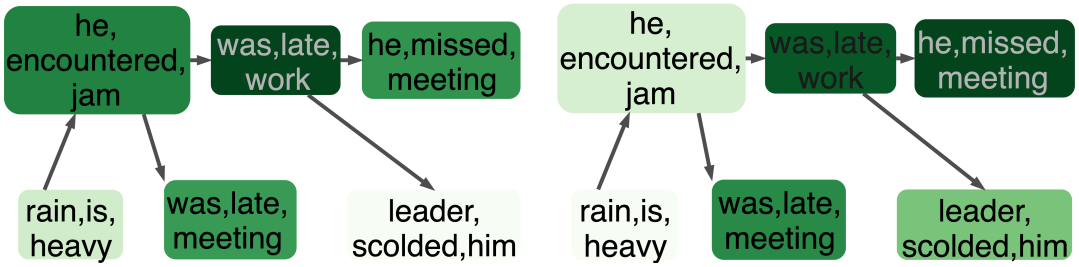
Table 3.4: Ablation study on the Enwiki testset.

To understand the importance of the key components of our approach, we perform an ablation study by training multiple ablated versions of our model, including the one without *weights* of edges in the retrieved causal subgraph, the one without the *2nd-layer* of GNN, and the one without *GNN*. The results are provided in Table 3.4. When the GNN module is gradually ablated, the performance of the model gradually degrades. This demonstrates that all modules of our multi-layer GNN effectively contribute to effect sentence generation.

3.4.7 Visualization

Input cause	he encountered a heavy traffic jam.
GPT2	the lighthouse was closed over three weeks.
BART	he was delayed for over an hour.
CopyNet	he missed missed the meeting.
CausalBert	causing him to miss bus.
EGCER	he missed the important meeting.

Table 3.5: A case with generations of different models.



(a) The causal scores calculated using the event vectors on the first layer of GNN.

(b) The causal scores calculated using the event vectors on the second layer of GNN.

Figure 3.3: The darker blue indicates the higher causal score.

Table 3.5 gives the generations of the different models for the two examples that have not been seen in the training data. CausalBERT generates "missing bus" given "missing" as guidance. However, from the input we can see that this person may be in a car, therefore the generated sequence is not an effect. That is CausalBERT, which is based on the word-level analysis, generates causal inconsistent sequence. In contrast, our method successfully predicts the expected effect event "*(he,missed,meeting)*", and generates the correct effect sentence.

We extract a part of CG according to the input cause, and visualize the causal scores cs using event vectors on the first and second layers of GNN respectively, as shown in Figure 3.3. In Figure 3.3a, the "*(was, late, work)*" receives the highest score, followed by "*(he, encountered, jam)*" and "*(was, late, meeting)*" in one-hop reasoning. And, the "*(leader, scolded, him)*" receives the lowest score. Noted that "*(he, encountered, jam)*" is actually not an effect event. However, in Figure 3.3b, the "*(he, missed, meeting)*" receives the highest score, followed by "*(was, late, work)*", "*(was, late, meeting)*" and "*(leader, scolded, him)*" in two-hop reasoning. The "*(he, encountered, jam)*" and "*(rain, is, heavy)*" receive lower scores. This makes sense because they are not effect events at all. This shows that the multi-layer GNN can well capture multi-hop causal relationships and thus are able to select the plausible effect events.

3.5 Discussion

We use the rule-based approach to extract causal events. Although the extraction quantity can be increased by expanding the corpus size, rule-based methods still face the problem of insufficient coverage, and a large number of causal relationships are still overlooked. We use event templates to guide the generation of result sentences. Although it can ensure that all event components are retained in the generated sentences, this approach brings additional drawbacks such as sentence rigidity and lack of diversity. But this problem can be solved through additional rewriting, we leave this in the future.

3.6 Chapter Summary

We present an event-level causal reasoning based effect generation method to generate the plausible effect sentences for the input cause sentences. Experiments show that our method performs better than competitors in capturing the causal semantics that should be generated. In the future, we would like to develop more effective approaches to enhance the effect event reasoning, and more powerful generators to generate the effect sentences with higher quality.

Chapter 4

Enhancing Event Causality Identification with Counterfactual Reasoning

4.1 Introduction

Rule-based extraction may introduce false-positives. To improve the precision of causality extraction, we make a preliminary exploration in the task of event causal identification (ECI).

Formally, ECI aims to identify causal relations between event pairs. For example, given the sentence "The *earthquake* generated a *tsunami*.", an ECI system should identify that a causal relation holds between the two mentioned events, i.e., earthquake $\xrightarrow{cause}$ tsunami. A good ECI system is able to discover a large number of causal relations from text and hence supports lots of intelligence applications, such as commonsense causal reasoning [67], narrative story generation [75], and many others.

Existing methods focus on mining potential causal signals, including *causal context*

keywords [64, 162] and *causal event pairs* [164, 163, 8], to enhance ECI. For example, [64] masks mentioned events from an ECI sentence to mine event-agnostic causal context patterns, e.g., "*generate*". And [164, 163] utilize external knowledge to mine causal event pairs, e.g., *"(earthquake, tsunami)"*. By mining potential causal signals, these methods improve the coverage of unseen events and causal relations, which is the reason for their success. However, they face the risk of amplifying the role of potential signals, resulting in biased inference.

Sentence	Label
A **6.1-magnitude earthquake** which hit the Indonesian province of Aceh on Tuesday killed at least one person, injured dozens and destroyed buildings, sparking panic in a region devastated by the quake-triggered **tsunami** of 2004.	0

Table 4.1: The example comes from the development set of EventStroyLine [9].

Due to the polysemy of language, causal signals are ambiguous. The occurrence of those signals does not always indicate that causality is established. That is, ambiguous *context keywords* and *event pairs* may lead to the **context-keywords bias** and the **event-pairs bias** in ECI. Specifically, in most cases, *"(earthquake, tsunami)"* in the training set occurs as a causal event pair, but in the sentence which is from the development set, as shown in Table 4.1, this event pair is not causal. Similarly, ambiguous keywords, such as "generate", do not always indicate causality [139, 140]. Relying heavily on those ambiguous signals may make an ECI model learn the spurious correlation [89] between ambiguous signals and labels. In other words, existing methods may overfit those ambiguous causal signals in training, and tends to predict a causal relation once the ambiguous signals appear when inference.

Considering this problem, we aim to mitigate these spurious correlations to develop a robust ECI model. Since the spurious correlations are caused by ambiguous causal signals, we question whether it is possible to directly learn the influence of those

ambiguous signals in training, so that we can mitigate those biases in inference. Motivated by this idea and existing dataset-debiasing works [83, 122, 96], we introduce *factual* and *counterfactual* reasoning for ECI. The *factual* reasoning takes the entire samples as input, which captures the combined features between context keywords and the event pairs, with the side-effect of learning features of biases. The *counterfactual* reasoning considers the two situations where only context keywords or event pairs are available. Intuitively, in counterfactual reasoning, a model can only make predictions based on context keywords or event pairs, so that the biases can be identified. In inference, we use counterfactual reasoning to estimate context-keywords bias and event-pairs bias, then subtract the biases from the factual predictions. To achieve this goal, we must locate the exact position of context keywords in a sentence[1]. But this is difficult because it requires extensive manual annotation. To avoid this, we adopt a model-based strategy. Considering the powerful feature extraction ability of pre-trained language models (PLMs), if we feed an event-removed sentence into PLMs, PLMs should be able to pay the most attention to the important context keywords. Based on this assumption, we split a sentence into two exclusive parts: an event-masked context and an event pair. They are fed into the counterfactual reasoning module to learn the context-keywords bias and event-pairs bias.

To summarize, the key contributions of this work are as follows.

- We consider the spurious correlation problem in ECI, which may make an ECI model overfit on ambiguous causal signals. To mitigate this problem, we propose a counterfactual reasoning mechanism for ECI. To the best of our knowledge, this is the first work that studies ECI from a counterfactual perspective.

- We conduct extensive experiments on two benchmarks. The result shows that our method is effective and achieves the new state-of-the-art results. Ablation study demonstrates the effectiveness of main components in our method.

[1]The positions of event pairs are already annotated.

4.2 Counterfactual ECI

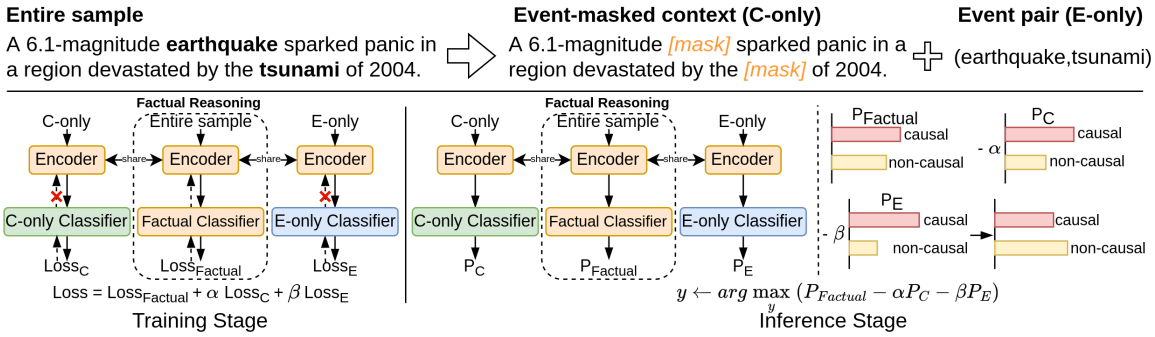


Figure 4.1: In the upper part, we split a sample into an event pair and an event-masked context. In the bottom part, we show the training and inference process of our method.

Previous ECI methods may overfit the ambiguous context keywords and event pairs, making biased inferences. We use counterfactual reasoning to eliminate this issue. Our method is depicted in Figure 4.1, which consists of a factual reasoning module and a counterfactual reasoning module.

4.2.1 Factual Reasoning Module

Factual reasoning learns the influence of entire ECI samples, following the traditional ECI paradigm. Here we present two classical methods.

Fine-tuning PLMs For ECI

We first fine-tune PLMs as a basic backbone. Given a sentence with a mentioned event pair (denoted as e_1 and e_2), we use PLMs, e.g., BERT [15], to encode the sentence and the event pair. Then the embeddings of [CLS], e_1 and e_2[2] are concatenated and

[2]An event is annotated as a text span, so the average-pooling operation is applied to obtain the event embedding.

applied with a non-linear transformation to obtain the hidden representation of the factual reasoning:

$$\mathbf{h}_{\text{ECI}} = tanh(\mathbf{W}_f^\top([\mathbf{h}_{[CLS]}; \mathbf{h}_{e_1}; \mathbf{h}_{e_2}])), \tag{4.1}$$

where $\mathbf{W}_f^\top \in \mathcal{R}^{3d \times d}$, $\mathbf{h}_{\text{ECI}} \in \mathcal{R}^d$, d is the hidden size of BERT. $\mathbf{h}_{\text{ECI}}$ is then projected with a linear layer $\mathbf{W}_p^\top \in \mathcal{R}^{d \times 2}$ to make a binary classification:

$$P_{\text{ECI}} = softmax(\mathbf{W}_p^\top \mathbf{h}_{\text{ECI}}). \tag{4.2}$$

Knowledge-Enhanced ECI

Existing works prove that knowledge is helpful for ECI. So we develop a knowledge-enhanced backbone. Following [64], we leverage external knowledge to further improve ECI. We use ConceptNet [117] as knowledge base. In ConceptNet, knowledge is structured as graph, where each node corresponds a concept, and each edge corresponds to a semantic relation. For e_1 and e_2, we search their related knowledge, i.e., matching an event with the tokens of concepts in ConceptNet. Events and concepts are lemmatized with the Spacy [3] toolkit to improve the rate of matching. We only consider 12 semantic relations that are potentially useful for ECI: *CapableOf, Causes, CausesDesire, UsedFor, HasSubevent, HasPrerequisite, Entails, ReceivesAction, UsedFor, CreatedBy, MadeOf, and Desires*. For each relation, we retrieve at most two knowledge relations according to the weights of relations.

Given (e_1, e_2), we retrieve the related knowledge tuples for e_1 and e_2 respectively, namely $K_{e_i} = \{\tau_{e_i}^1, \tau_{e_i}^2, \cdots, \tau_{e_i}^{N_i}\}$, where $i = 1, 2$ denotes the event index, $\tau = (h, t)$ denotes a knowledge tuple (head, tail), N_1 and N_2 is the number of knowledge tuples. We obtain the knowledge-enhanced features of e_1 and e_2 by average-pooling on the embeddings of corresponding knowledge tuples:

$$\mathbf{h}_{e_i}^K = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{W}_k^\top [\mathbf{h}_{e_i}^j; \mathbf{t}_{e_i}^j], \tag{4.3}$$

[3] https://spacy.io/

where $i = 1, 2$, $\mathbf{h}$ and $\mathbf{t}$ denote the embeddings of a tuple (h, t), $\mathbf{W}_k \in \mathcal{R}^{2d \times d}$ is trainable. Then the knowledge-enhanced event representations $\mathbf{h}_{e_1}^K$ and $\mathbf{h}_{e_2}^K$ are concatenated with $\mathbf{h}_{\text{ECI}}$ (Equation 4.1), and input into a MLP to make a binary classification:

$$P_{\text{ECI}}^K = softmax(\text{MLP}([\mathbf{h}_{\text{ECI}}; \mathbf{h}_{e_1}^K; \mathbf{h}_{e_2}^K])). \tag{4.4}$$

Finally, the cross-entropy loss is applied to P_{ECI} and P_{ECI}^K to train the two backbones. Factual reasoning learns combined features between the context and the event pair, but biases may be entangled into the combined features. Next, we propose counterfactual reasoning to capture the entangled biases.

4.2.2 Counterfactual Reasoning Module

To estimate the context-keywords bias and the event-pairs bias in training, we split a sentence into two exclusive parts: an event-masked context and an event pair. For each part, we use counterfactual reasoning to estimate the corresponding bias.

Estimating Context-Keywords Bias

We consider the counterfactual situation where only the event-masked context is available. We input the context into PLMs, and let PLMs automatically attend to the important context keywords. The [CLS] token embedding $\overline{\mathbf{h}_{[CLS]}}$ is used as the representation of the event-masked context. Note that $\overline{\mathbf{h}_{[CLS]}}$ is different from $\mathbf{h}_{[CLS]}$ (Equation 4.1) because the event pair is removed in the current situation. We obtain the hidden state of the current situation by:

$$\overline{\mathbf{h}_C} = tanh(\mathbf{W}_f^\top([\overline{\mathbf{h}_{[CLS]}}; \mathbf{\Phi}_E; \mathbf{\Phi}_E])), \tag{4.5}$$

where $\mathbf{W}_f$ is the shared parameter (Equation 4.1), $\mathbf{\Phi}_E \in \mathcal{R}^d$ is a learnable constant, and represents the void input events. The insight of this setting is that if we have no

information about the event pair, we would like to make inferences by random guess. Then $\overline{\mathbf{h}_C}$ is projected to make binary classification:

$$P_C = softmax(\mathbf{W}_C^\top \overline{\mathbf{h}_C}), \tag{4.6}$$

where $\mathbf{W}_C$ is trainable, P_{C} estimates the influence of the context-keywords bias.

Estimating Event-Pairs Bias

Next, we consider the counterfactual situation where only the event pair (e_1, e_2) is available. Through PLMs, we get the event embeddings of $\overline{\mathbf{h}_{e_1}}$ and $\overline{\mathbf{h}_{e_2}}$. Note that $\overline{\mathbf{h}_{e_1}}$ and $\overline{\mathbf{h}_{e_2}}$ is different from $\mathbf{h}_{e_1}$ and $\mathbf{h}_{e_2}$ (Equation 4.1) because the context is invisible in the current situation. We obtain the hidden state of the current situation by:

$$\overline{\mathbf{h}_E} = tanh(\mathbf{W}_f^\top([\mathbf{\Phi}_C; \overline{\mathbf{h}_{e_1}}; \overline{\mathbf{h}_{e_2}}])), \tag{4.7}$$

where $\mathbf{\Phi}_C$ is a learnable constant, and represents the void input context. Then $\overline{\mathbf{h}_E}$ is projected with a linear layer to make binary classification:

$$P_E = softmax(\mathbf{W}_E^\top \overline{\mathbf{h}_E}), \tag{4.8}$$

where $\mathbf{W}_E$ is trainable, P_E estimates the influence of the event-pairs bias.

4.2.3 Training and De-biased Inference

We jointly train the factual and counterfactual reasoning modules, the final loss is:

$$Loss = Loss_{Factual} + \alpha Loss_C + \beta Loss_E. \tag{4.9}$$

$Loss_{Factual}$ is over P_{ECI} or P_{ECI}^K. $Loss_C$ is over P_C and $Loss_E$ is over P_E. α and β are two trade-off coefficients that balance the two types of biases. Note that we share the encoding process (Equation 4.1) between factual and counterfactual modules, but we do not backpropagate $Loss_C$ and $Loss_E$ to the encoder, as shown in Figure 4.1. This

is because we require the counterfactual reasoning module to make predictions only based on the event-masked context or the event pair, and has no information about the missing part.

After training, the counterfactual reasoning module will learn the bias-estimation mechanism. Therefore, we can make de-biased inference by:

$$y \leftarrow \text{argmax}_y(P_{Factual} - \alpha P_C - \beta P_E), \tag{4.10}$$

where $P_{Factual}$ can be P_{ECI} or P^K_{ECI}.

4.3 Experiment

4.3.1 Datasets

Datasets include EventStoryLine [9] and Causal-TimeBank [73]. These two benchmarks have been widely used by previous methods as standard datasets for ECI. EventStoryLine contains 22 topics, and 1770 of 7805 event pairs are causally related. Causal-TimeBank contains 184 documents, and 318 of 7608 event pairs are causally related. We conduct the 5-fold and 10-fold cross-validation on EventStoryLine and Causal-TimeBank respectively. The last two topics of EventStoryLine are used as the development set for two tasks. All of this is the same as previous works for fairness. Evaluation metrics are Precision (P), Recall (R) and F1-score (F1).

4.3.2 Baselines

We compare our method with following baselines:

- KMMG [64], which proposes a mention masking generalization method and also utilizes the external knowledge.

- KnowDis [164], a data-augmentation method that utilizes the distantly labeled training data.
- LearnDA [163], a data-augmentation method with iteratively generating new examples and classifying event causality in a dual learning framework.
- LSIN [8], a latent-structure induction network to leverage the external knowledge;.
- CauSeRL [162], a self-supervised framework to learn context-specific causal patterns from external causal corpora.

4.3.3 Experimental Settings

When implementing our factual reasoning models, we adopt BERT(base), which is same as previous methods. We denote our two factual backbones as BERT and BERT$_K$. All parameters are searched according to the F1 on the Dev set.

Due to the data imbalance problem, we adopt a over-sampling strategy for training. The early-stop is used due to the small scale of datasets. We use the Adam optimizer and linearly decrease learning rate to zero with no warmup. We use PyTorch toolkit to conduct all experiments on the Arch Linux with RTX3090 GPU. All the hyperparameter for two tasks are searched according to the F1 score on the development set. For reproduction, we set the random seed to 42 for all experiments. The searched parameters for two datasets are shown in Table 4.2.

4.3.4 Overall Result and Ablation Study

The overall result is shown in Table 4.3. We have the following observations.

- BERT$_K$ has a similar result with compared baselines, and performs better than BERT. This coincides with previous works that knowledge is helpful for ECI.

Parameters	ESL	CTB
Batch Size	32	32
Learning Rate	5e-5	5e-5
Drop-rate	0.3	0.2
α	0.15	0.25
β	0.35	0.25

Table 4.2: The used hyperparameters for two datasets.

- Our CF-ECI method achieves consistent improvement when deployed on BERT or BERT$_K$. This shows the effectiveness of our method.
- Compared with the previous methods, our method has a higher precision score. This is because we make a de-biased inference, which is able to reduce the false-positive predictions, hence improve the precision.
- Utilizing knowledge may reduce the precision score, because irrelevant knowledge may be introduced. This coincides with LSIN [162].

Ablation Study We conduct ablation study to investigate the influence of context-keywords de-biasing (§ 4.2.2) and event-pairs de-biasing (§ 4.2.2). We develop two ablated variants: (1) "w/o EPB" denotes that we ablate the event-pairs de-biasing module; (2) "w/o CKB" denotes that we ablate the context-keywords de-biasing module. The result is shown in Table 4.3. We have following observations.

- No matter what backbone (BERT or BERT$_K$) is used, after ablating "EPB" or "CKB", the ablated variant has a performance drop. This indicates that ambiguous context-keywords and event-pairs have adversely influence of ECI. By making de-biased inference, our CF-ECI achieves the best performance.
- In addition, we observe that the context-keywords bias is more severe than the

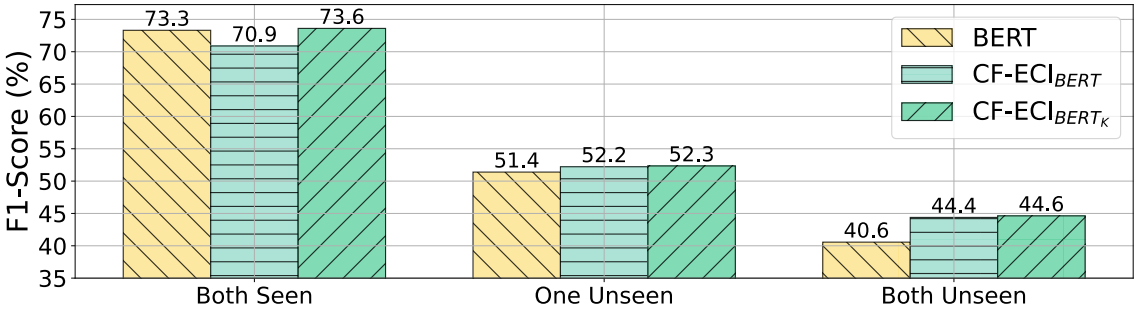


Figure 4.2: F1 scores (%) of identifying unseen events.

event-pairs bias, which indicates that the trained models tend to use superficially keywords for inference. The possible reason is that this strategy inevitably leverages ambiguous keywords that are potential biases, though it can capture some causal keywords as good evidence.

4.3.5 Further Discussion

Bias Analysis

Previous works [121, 96] point out that the unfairness of a trained model can be measured by the imbalance of the predictions produced by the model. Following [96], we use the metric *imbalance divergence* (ID) to evaluate whether a predicted distribution P is unfair: $ID(P, U) = JS(P||U)$, where $JS(\cdot)$ denotes the JS divergence of P and the uniform distribution U. To evaluate the unfairness of a trained model M, we calculate its ID over all dev or test samples: $ID(M) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} JS(P(x), U)$, where $P(x)$ can be the output distribution of a factual (§ 4.2.1) or counterfactual (§ 4.2.2) model. As shown in Table 4.4, when deployed on different backbones, our method can obviously and consistently reduce the ID metric. This indicates that our method is helpful to eliminate two kinds of biases.

Models	ESL			CTB		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
KMMG	41.9	62.5	50.1	36.6	55.6	44.1
KnowDis	39.7	66.5	49.7	42.3	60.5	49.8
LearnDA	42.2	**69.8**	52.6	41.9	68.0	51.9
CauSeRL	41.9	69.0	52.1	43.6	**68.1**	53.2
LSIN	47.9	58.1	52.5	51.5	56.2	52.9
This Paper						
BERT	45.8	57.4	50.9	49.8	50.3	50.1
BERT_K	43.2	65.8	52.2	48.3	54.5	51.2
$\text{CF-ECI}_{\text{BERT}}$	**48.7**	59.0	53.4*	**54.1**	53.0	53.5*
$\text{CF-ECI}_{\text{BERT}_K}$	47.1	66.4	**55.1***	50.5	59.9	**54.8**
Ablation Experiment						
$\text{CF-ECI}_{\text{BERT}}$						
: w/o EPB	47.7	57.6	52.2	51.7	53.6	52.6
: w/o CKB	48.0	56.7	52.0	51.1	52.5	51.8
$\text{CF-ECI}_{\text{BERT}_K}$						
: w/o EPB	46.8	63.8	54.0	50.8	56.4	53.4
: w/o CKB	47.0	62.6	53.7	50.2	56.3	53.1

Table 4.3: The overall and ablation-study result. Scores with **bold** denotes the best results. *: the significant test is conducted using paired t-test between our method and the used backbones, with the level of $p = 0.05$. "CKB" denotes the context-keywords de-biasing. "EPB" denotes the event-pairs de-biasing.

Identifying Unseen Events

We explore the ability of our method to identify unseen events. We first randomly select 1/3 of ESL documents as the training set, then divide the remaining documents

Methods	ESL		CTB	
	Dev	Test	Dev	Test
BERT	17.75	16.71	20.47	21.02
CF-ECI$_{\text{BERT}}$	02.40	02.09	02.71	02.64
BERT$_K$	17.08	15.70	20.46	21.04
CF-ECI$_{\text{BERT}_K}$	02.44	02.25	02.81	02.77

Table 4.4: The model unfairness result (lower is better) on the dev-set and test-set of ESL and CTB.

Figure 4.3: The heatmaps of the predictions by BERT and CF-ECI$_{BERT}$ respectively. Text with the dotted line denotes the annotated events.

into (1) "Both Seen", where two events of a sample appear in training data; (2) "One Unseen", where only one event of a sample exists in training data; (3) "Both Unseen", where both events are unobserved during training. From Figure 4.2, we have following observations. (1) CF-ECI has a significant improvement on the "Both Unseen" set, compared with BERT. (2) CF-ECI$_{BERT_K}$ performs better than CF-ECI$_{BERT}$ on the "Both Seen" set.

Visualization

We depict the heatmaps of predictions by BERT and CF-ECI$_{BERT}$ respectively, in Figure 4.3. BERT pays the most attention to the words: "*eqrthquake, spark, quake, tsunami*", and gives a causal prediction with the 97.9% probability. In contrast, CF-ECI$_{BERT}$ dispersedly attends to words and does not find enough causal evidence, hence it gives a non-causal prediction.

4.4 Discussion

First, we only access limited computation resources and perform continual pre-training from BERT [15], which is not general enough for every event-related reasoning task. Second, counterfactual reasoning makes our approach conservative in identifying causal relationships, so our method has a higher precision. However, some potential causal relationships will be discarded. How to achieve a good trade-off between precision and coverage is a problem. In addition, the way we utilize knowledge is relatively simple, and it is very likely that we have not made full use of knowledge. Designing more complex knowledge-enhanced methods may lead to better results.

4.5 Chapter Summary

We discuss the issue of context-keywords and event-pairs biases in ECI. To mitigate this problem, we propose the counterfactual reasoning which explicitly estimates the influence of the biases, so that we can make a de-biased inference. Experimental results demonstrate the significant superiority of our method. The robustness and explainability of our method are also verified by further studies.

Part II

Causality Enhanced Factual and Counterfactual Reasoning in Narratives

Chapter 5

Enhancing Narrative Commonsense Reasoning With Multilevel Causal Knowledge.

5.1 Introduction

In Chapter 3 and 4, we explore the method of causality mining and de-baising, so, in this chapter, we investigate the causality enhanced factual reasoning in narratives.

In recent years, narrative reasoning has attracted much attention. It provokes a variety of intelligent systems, including commonsense causal reasoning [108, 27, 67], abductive reasoning [5], narrative story generation [75], and so on. Extensive evidence [128, 127, 125] shows that the way in which people comprehend narratives is heavily influenced by the causal relations among narrative stories, which implies that causal relations are an essential component of narrative text. However, neural models usually lack causal background knowledge [31, 57], and have a very limited ability for narrative reasoning. In order to make up for this deficiency, researchers focus on providing causal knowledge to neural models to enhance their narrative reasoning ability.

Because causality mainly occurs at the sentence-level and the event-level in text, existing works can be divided into two groups. (1) The sentence-level causalities generally have complex sentence structures, and it is difficult to locate the exact range of causes and effects from them. To solve this problem, the first group of works uses sentence-level causalities to design training tasks and continues to train pre-trained language models (PLMs) [57, 156]. In this way, these works utilize the powerful feature-extracting ability of PLMs, and *implicitly* inject causal features into PLMs. As the carrier of sentence-level causalities, causal-enhanced PLMs can be easily transferred to downstream narrative reasoning tasks. (2) The other group of works focuses on utilizing event-level causalities. Different from sentence-level causalities, event-level causalities have simple structures and can be *explicitly* structuralized in knowledge bases. Therefore, these works [58, 88, 82] typically exploit graph-based neural networks (GNNs) [131, 46] to learn structural information from causal event graphs for narrative reasoning. Both two groups of works have made some achievements in narrative reasoning, however, they still face the following deficiencies:

- In the first group of methods, although PLMs have very large-scale parameters, it is difficult for PLMs to remember all sentence-level causal knowledge. When transferred to downstream narrative reasoning tasks, causal-enhanced PLMs may forget some background causal knowledge.

- The second group of works usually uses GNNs to encode causal event graphs. However, causal event graphs are generally extracted by rule-based methods [103, 155] or human annotations [111], therefore the scale of event-level causalities is limited. In addition, an event generally contains a sequence of words, which makes events too unique from each other. These two facts make it difficult to find a sufficient number of relations between event pairs. In other words, event knowledge is very sparse. Event sparsity brings difficulty to GNNs when learning useful event representations and capturing meaningful causal semantics.

Though having different forms, sentence-level and event-level causalities are the embodiment of causality in different scenarios, and they complement each other. On the one hand, if we provide event-level causalities, i.e., causal event graphs, as explicit knowledge ground when using causal-enhanced PLMs, it is promising to mitigate the forgetting problem. On the other hand, if we combine the two levels of causalities, we can reduce the number of unseen relationships, thus improving the coverage of neural models to causal relations. That is, it is reasonable and necessary to use both of them for narrative reasoning. However, previous works study either sentence-level or event-level causalities. This motivates us to effectively organize the two levels of causalities. In addition, we make a step towards reducing the sparsity in event causalities. We notice that an event usually contains several word components. The component of an event can interact with the component of another event, although there may be no relationship between the two events. This motivates us to divide an event into several word components, so that the word-word relations between event components can be retrieved. Since word-word relations capture the interplays between event components, it is possible to alleviate the event sparsity problem.

In this chapter, we make full use of multi-level causalities and present a two-stage narrative reasoning method. *In the first stage*, we devise post-training tasks to inject sentence-level causalities into PLMs. We design different training tasks for narrative understanding and generating scenarios. For narrative understanding, we use causal sentence pairs as positive examples, and negatively sample non-causal sentence pairs as negative examples. Then we train PLMs, e.g., BERT [15] and RoBERTa [66], to rank positives upon negatives. For narrative generation, we input cause (or effect) sentences into PLMs, e.g., BART [50], and train PLMs to generate the corresponding effect (or cause) sentences. After post-training, we obtain causal-enhanced PLMs, which carry sentence-level causalities. The causal-enhanced PLMs are used as the backbones in the next stage. *In the second stage*, we exploit event-level causalities, and ground narrative reasoning on structural knowledge graph. Specifically, we break

an event into word components, so that we are able to construct the hierarchical two-level knowledge graph (KG), which consists of an event-level graph and a word-level graph. The event-level graph contains event-level causalities, which are typical narrative processes related to the given context. The word-level graph contains relations between event components as well as word-level commonsense knowledge. The word-level graph captures the interactions between event components, making it possible to mitigate event sparsity. Based on the hierarchical KG, we devise a novel KG-based reasoning process, which leverages useful knowledge from the KG for narrative reasoning. In summary, we make the following contributions:

- We introduce a two-stage method known as Multi-Level Causal-Knowledge for Narrative Reasoning (MCNR). This method, designed in a generic framework, proves applicable to a range of narrative understanding and generation tasks.
- Through the subdivision of events into multiple word components, we derive the hierarchical knowledge graph. This not only mitigates the challenge of event sparsity but also provides additional word-level information to enhance narrative reasoning.
- Our method undergoes validation on narrative understanding and generation tasks. Experimental results prove its superiority over compared baselines. Detailed ablation studies further confirm the effectiveness of our approach.

5.2 Method

The overall framework of our method is shown in Figure 5.1. Our method consists of two stages. In the first stage (Section 5.2.1), we design post-training tasks to inject sentence-level causalities into PLMs. At last, we obtain causal-enhanced PLMs, which carries sentence-level causalities and can be adapted to downstream tasks. In

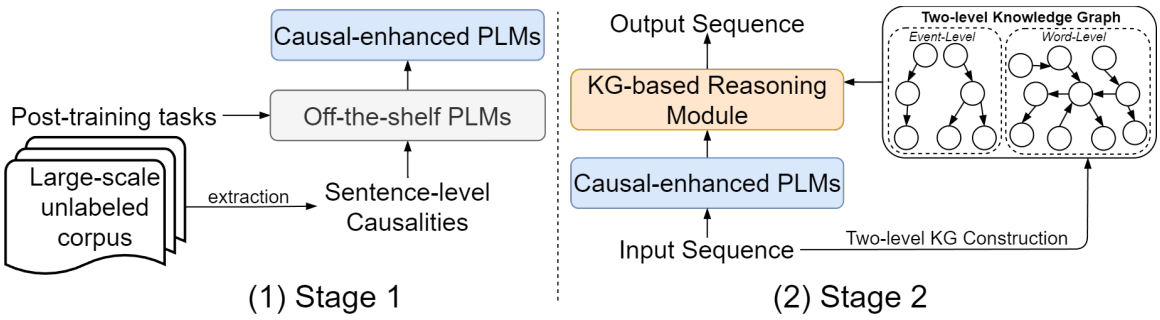


Figure 5.1: The overall framework of our method. In the first stage, we extract sentence-level causalities which are injected into PLMs via post-training tasks. Finally, we obtain causal-enhanced PLMs. In the second stage, we use causal-enhanced PLMs as backbone, and utilize structural knowledge for narrative reasoning. The construction process of two-level KG is in Section 5.2.2.

the second stage (Section 5.2.2), by using causal-enhanced PLMs as the backbone, we additionally exploit our two-level KG for narrative reasoning.

5.2.1 Sentence-level Causalities Enhanced Post-training

To obtain sentence-level causalities, we pre-define causal extraction rules to extract sentence-level causalities from the large-scale unlabeled corpus. Then we devise post-training tasks to inject sentence-level causalities into PLMs.

Sentence-level Causalities Extraction

We leverage BookCorpus [161] as our data source. There are several widely used data corpora, such as WIKIPEDIA and web-crawl [25], but these corpora are generally very noisy. Differently, BookCorpus mainly contains books, is a relatively clean corpus, and is widely used for academic research. In particular, BookCorpus contains 11K books in various subgenres (e.g., historical) and is likely to contain rich causal knowledge. To extract sentence-level causalities from BookCorpus, we collect some causal discourse

markers from PDTB [95]. These markers are: *Thus*, *Therefore*, *So*, *because*, *Thereby*, *Hence*, *As a result*, *Consequently*. For each causal marker, we match text like "Arg_1 *Marker* Arg_2" to extract causal sentence pairs, where "Arg_1" and "Arg_2" denote the matched sentence pair. There may be some poor-quality causal sentences, so we design several heuristic rules to filter them out. For example, we discard sentences that contain no more than 10 words, or contain less than two content words, or contain special symbols. Considering the directionality of causality, we balance the number of cause-to-effect and effect-to-cause sentence pairs to avoid the data-imbalance problem. Finally, we obtain about 180K causal sentence pairs.

To inject the extracted sentence-level causalities into PLMs, we design different post-training tasks for narrative understanding and narrative generation scenarios. Next, we introduce our post-training tasks.

Post-training Tasks

We devise post-training tasks to further train PLMs based on sentence-level causalities, so that PLMs can serve as the carrier of sentence-level causalities. Narrative reasoning generally involves two different scenarios: narrative understanding and narrative generation, therefore we separately devise post-training tasks for each scenario. For narrative understanding, we design the task of causal/non-causal sentence pairs ranking (Section 5.2.1). For narrative generation, we design the task of causal text generation (Section 5.2.1).

Causal/Non-causal Sentence Pairs Ranking for Narrative Understanding This task is devised for the narrative understanding scenario, in which a model usually takes an input premise, and is required to select the most reasonable hypothesis from several alternatives. To make our method applicable to this setting, we devise a contrastive ranking task that requires PLMs to rank causal sentence pairs above

non-causal sentence pairs. Specifically, we regard causal sentence pairs as positive examples. Given a cause-to-effect sentence pair (c, e), where c denotes the cause and e denotes the effect, we fix the cause c, and randomly sample several sentences from the effect set to generate the negative examples (c, e')[1]. We also fix the effect e and randomly sample several sentences from the cause set to get (c', e). For simplicity, we denote the positive sample (c, e) as x, and denote the negative samples (c, e') and (c', e) as $\overline{x}^i$, where $i = \{1, \cdots, N\}$ and N is the amount of negative samples. By default, for a (c, e)[2], we sample one (c, e') and one (c', e), so $N = 2$. Next, we use *bi-directional* PLMs, e.g. BERT [15] and RoBERTa [66], to obtain sentence-level representations of training examples:

$$\mathbf{h}_x = \text{PLMs}(x) \quad \text{and} \quad \mathbf{h}_{\overline{x}^i} = \text{PLMs}(\overline{x}^i). \tag{5.1}$$

Next, sentence-level representations are passed into a linear layer to derive the causal scores (cs) carried by training examples:

$$cs = \text{Linear}(\mathbf{h}_x; \theta^{(cs)}) \quad \text{and} \quad \overline{cs}^i = \text{Linear}(\mathbf{h}_{\overline{x}^i}; \theta^{(cs)}), \tag{5.2}$$

where $\theta^{(cs)}$ denotes the parameters of the used linear layer. Lastly, to distinguish the true causal pair from the corrupted sentence pairs, we use the contrastive ranking object to distinguish the positive example from the negatives:

$$\begin{aligned} \mathcal{L}^{rank} &= -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log P(x | x, \{\overline{x}^i\}_{i=1}^N) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log \frac{\exp(cs)}{\exp(cs) + \sum_{i=1}^N \overline{cs}^i}, \end{aligned} \tag{5.3}$$

where $\mathcal{D}$ denotes the set of causal sentence pairs. After convergence, we obtain causal-BERT or causal-RoBERTa, which carries abundant sentence-level causalities, and are applicable to narrative understanding tasks. Next, we introduce the post-training task for narrative understanding.

[1]Causal markers are removed when training.

[2]Note that for effect-to-cause relations (e, c), we use the same process to obtain negative samples (e, c') and (e', c).

Causal Text Generation for Narrative Generation This task is devised for the narrative generation. In this setting, a model usually takes a text as input and is required to generate a semantic-related output text. To inject sentence-level causal knowledge into *generative* PLMs, e.g. T5 [104] and BART [50], we devise a causal text generation task. Specifically, for a cause tuple "(Arg_1, *Marker*, Arg_2)", we merge Arg_1 and *Marker* as input $X = [Arg_1; Marker] = \{x_1, x_2, \cdots, x_m\}$ which has m tokens, and regard Arg_2 as the gold output $Y = \{y_1, y_2, \cdots, y_n\}$ which has n tokens. We use *Marker* as the prompt to indicate the direction of causality when generation, which preserves flexibility when adapting to downstream tasks. This is because there are different semantic relationships between input and output in different downstream tasks. For example, in the story generation task [75], the output is usually the effect of the input. But in the abductive reasoning [5] task, the output should be the cause of the partial input. We firstly use the generative PLM to encode the input X:

$$\mathbf{H}_X = \text{PLM-Encoder(X)}, \tag{5.4}$$

where $\mathbf{H}_X \in \mathcal{R}^{m \times d}$ is the embeddings of X, d is the hidden size. Then, we use the decoder of the PLM to obtain the token distribution of the target sequence at the time-step t:

$$\begin{aligned} \mathbf{h}_{y_t} &= \text{PLM-Decoder}(\mathbf{Y}_{<t}, \mathbf{H}_X), \\ P(y_t|Y_{<t}, X) &= \text{softmax}_V(\mathbf{W}_v \mathbf{h}_{y_t} + \mathbf{b}). \end{aligned} \tag{5.5}$$

The training goal is to maximize the likelihood of generating the gold Y for the input X, and we adopt the auto-regressive language model loss as the loss function:

$$\mathcal{L}^g = -\frac{1}{|\mathcal{D}|} \sum_{(X,Y) \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \log P(y_i^{\text{gold}} | Y_{<i}^{\text{gold}}, X). \tag{5.6}$$

The causal-enhanced PLMs, e.g., causal-T5 or causal-BART, are obtained after convergence, which are applicable to narrative generation tasks.

Actually, in the first stage, we use PLMs to extract causal feature in unstructured sentence-level causalities, so that the causal knowledge can be saved in PLMs and

can be transferred to real-world applications. And it is straightforward to transfer our causal-enhanced PLMs to downstream tasks. Either supervised fine-tuning or zero-shot inference can be used to achieve the transfer process.

Next, we introduce our second stage, which combines the causal-enhanced PLMs and external structural knowledge to further improve narrative reasoning.

5.2.2 Combining Event Causalities for Narrative Reasoning

Though causal-enhanced PLMs carry abundant causal knowledge, they may face the forgetting problem when adapting to downstream tasks. To solve this problem, it is intuitive to combine event causalities, i.e., grounding reasoning on causal event graphs. The advantage of this solution is that diverse knowledge provides more comprehensive background for reasoning. In addition, explicit knowledge graphs allow us to explain the prediction of a model by tracking the knowledge used in the reasoning process. However, the sparsity of events impedes a model to learn useful event representations. To alleviate this issue, we construct the hierarchical two-level KG by dividing an event into several word components. Next, we introduce how to construct the two-level KG, and how to apply the two-level KG to downstream reasoning tasks.

Constructing Two-Level KG

Given an input context, we first retrieve the context-related event-level causalities. We use COMeT [37] as the knowledge base of event causalities. COMeT is a BART-based [50] model, which is fine-tuned on the manually annotated if-then relationship dataset ATOMIC [111]. Nine different types of inferential relations about events can be produced using COMeT. Details about these relations can be seen in [37]. In this article, we use the following relations to generate event causalities: *xWant*, *oWant*, *xEffect*, *oEffect*, *HasSubevent*, *xIntent*, *xNeed*, *Causes*.

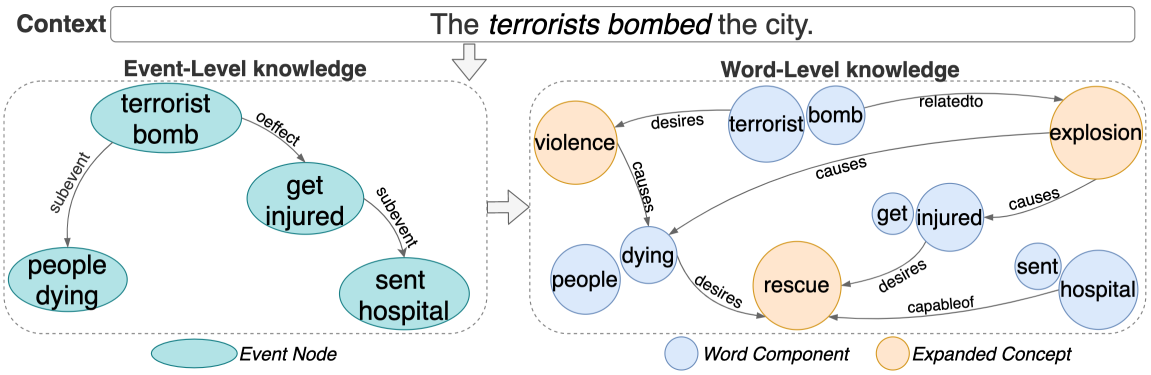


Figure 5.2: An illustration of building the two-level KG for the input context.

Given the input context, we use the Spacy dependency parser to extract context events (called central events). By feeding central events into COMeT, we obtain one-hop events, which are then fed into COMeT to produce two-hop events. Finally, a large number of causal event chains are produced. There may be some low-quality chains, so we design several heuristic rules to filter them. For instance, a chain will be filtered if any event in the chain consists of fewer than two words. There are still many event chains, we randomly keep no more than 80 chains for each input. Next, for each input, we convert the kept event chains to a causal event graph. However, due to the sparsity of events, there are a very small number of edges in the graphs. Specifically, the average in-degree of nodes is less than 1.5. The sparsity of events brings difficulties to learning event representations. So, we divide each event into a sequence of words. For each word, we retrieve three connected words from ConceptNet [118] according to the weights of connections. Next, we discover the relations between each pair of words. This helps to alleviate the issue of event sparsity by allowing previously unconnected events to interact through relations between event components. A detailed example of constructing a two-level knowledge graph is depicted in Figure 5.2.

Generally, a two-level KG G contains two types of nodes: event nodes $\mathcal{V}_e$ and word nodes $\mathcal{V}_w$. An event $e_i \in \mathcal{V}_e$ is a sequence of words $e_i = \{w_{i1}, \cdots, w_{ik}\}$, where $w_{i1}, \cdots, w_{ik} \in \mathcal{V}_w$. Additionally, G has two kinds of edges. An event-event edge

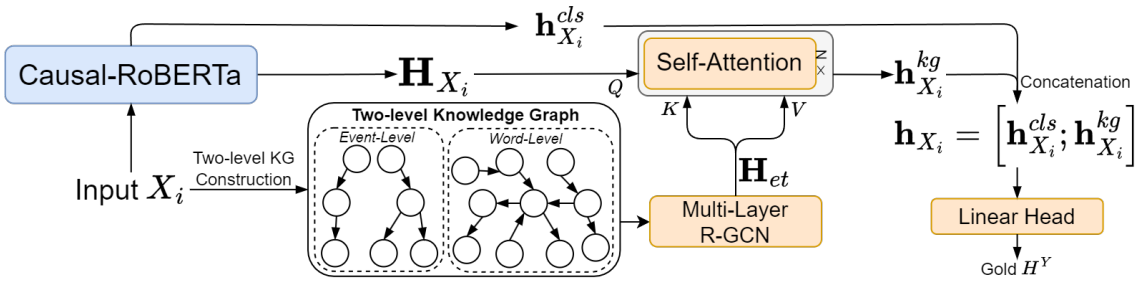


Figure 5.3: Our two-level KG-based reasoning method for narrative understand. By combining causal-RoBERTa with the two-level KG, we make full use of multi-level knowledge for narrative understanding.

(h_e, r_e, t_e) denotes that an event-level relation r_e exists between the head event h_e and the tail event t_e. A word-word edge (h_w, r_w, t_w) denotes that a word-word relation r_w exists between the head word h_w and the tail word t_w, where h_w and t_w may exist in different events.

Next, we combine causal-enhanced PLMs and our two-level KG to improve narrative reasoning. Because narrative understanding and narrative generation have different input-output formats, we design different network structures for the two scenarios. We first introduce our method for narrative understanding (5.2.2), then introduce our method for narrative generation (5.2.2).

KG-Enhanced Narrative Understanding

Task Definition In this setting, a model usually takes an input text P as the premise, and is asked to choose the most reasonable hypothesis from several alternatives $H = \{H_i\}, (i = 1, \cdots, I)$, where I is the number of alternatives. To enhance this reasoning process, according to P, we extract a relevant two-level KG G as knowledge ground. In the training stage, we aim to maximize the following probability:

$$P(H^Y|P, G), \tag{5.7}$$

where H^Y is the gold hypothesis. In inference, we select the hypothesis that has the highest probability. Our method is shown in Figure 5.3, which mainly consists of three modules: (1) the context encoding module which encodes input texts into hidden states, (2) the knowledge encoding module consists of stacked graph attention networks, (3) the knowledge-integrated reasoning module consists of transformer blocks that jointly attend to the input and the encoded knowledge.

Context Encoding Module We concatenate the premise P and each hypothesis H_i as a sequence of tokens:

$$X_i = [\text{CLS}]P[\text{SEP}]H_i[\text{SEP}], \tag{5.8}$$

where [CLS] and [SEP] are special tokens of PLMs. [CLS] denotes the start token of a sequence. [SEP] is the separator token between different sentences. We compute the contextualized representations of X_i with our causal-enhanced PLMs, e.g., causal-RoBERTa, and obtain the sequence-level vector (the vector of [CLS]) and the token-level vectors:

$$\{\mathbf{H}_{X_i}, \mathbf{h}^{cls}_{X_i}\} = \text{Causal-RoBERTa}(X_i), \tag{5.9}$$

where $\mathbf{H}_{X_i} \in \mathcal{R}^{M \times d}$ is the token vectors, $\mathbf{h}^{cls}_{X_i} \in \mathcal{R}^d$ is the sequence-level vector, d is the hidden size, M is the number of tokens in X_i. We can also choose other encoders. For example, we can replace our causal-RoBERTa with off-the-shelf RoBERTa to validate the effectiveness of sentence-level causalities, as shown in the ablation study.

Knowledge Encoding Module Relational Graph Convolutional Network (RGCN) [114] is used to encode a two-level KG so that we are able to use structural graph information to improve the representations of events and words. The RGCN contains stacked L layers. Initially, we obtain word embeddings via the used causal-RoBERTa, and randomly initialize embeddings of word-level relations. Then, for a word $t_w \in \mathcal{V}_w$, we gather the neighbors $\mathcal{N}(t_w)$ of t_w, involving in the connected (head word, word-

level relation) pairs, to renew its embedding at the $l+1$-th layer:

$$\mathbf{h}_{t_w}^{t+1} = \sigma(\frac{1}{|\mathcal{N}(t_w)|} \sum_{\substack{(h_w, r_w) \in \\ \mathcal{N}(t_w)}} \mathbf{W}_a^l(\mathbf{h}_{h_w}^l - \mathbf{h}_{r_w}^l) + \mathbf{W}_s^l \mathbf{h}_{t_w}^l), \tag{5.10}$$

where σ is the ReLU function, the parameters $\mathbf{W}_s^l$ and $\mathbf{W}_a^l$ are particular to the l-th layer. Another linear transition is used to renew the embeddings of word-level relations at the l-th layer: $\mathbf{h}_{r_w}^{t+1} = \mathbf{W}_r^l \mathbf{h}_{r_w}^l$. Finally, words embeddings $h_{h_w}^L$, $h_{t_w}^L$ at the L-th layer are obtained. We then use the average-pooling operation to get event embeddings:

$$\mathbf{h}_{e_i} = \text{average-pooling}(\mathbf{h}_{w_{i1}}^L, \cdots, \mathbf{h}_{w_{ik}}^L), \tag{5.11}$$

where $\{w_{i1}, \cdots, w_{ik}\}$ are word components of the event e_i. This process actually integrates word-level knowledge into event embeddings, which makes it possible to mitigate the event sparsity problem. For a event-level triple (h_e, r_e, t_e), we concatenate the embeddings of (h_e, r_e, t_e) to obtain embeddings of the event triple:

$$\mathbf{h}_{(h_e, r_e, t_e)} = \mathbf{W}_e^\top [\mathbf{h}_{h_e}; \mathbf{h}_{r_e}; \mathbf{h}_{t_e}], \tag{5.12}$$

where $\mathbf{W}_e \in \mathcal{R}^{3d \times d}$ is a trainable parameter, the event relation embedding $\mathbf{h}_{r_e}$ is initialized by randomly. The hidden representations of all event-level triples can be denoted as:

$$\mathbf{H}_{et} = \{\mathbf{h}_{(h_e, r_e, t_e)}^i\}_{i=1}^{Z_{et}}, \tag{5.13}$$

where $\mathbf{h}_{(h_e, r_e, t_e)}^i$ is the embedding of i-th event triple, Z_{et} is the number of event-level triples, and $\mathbf{H}_{et} \in \mathcal{R}^{Z_{et} \times d}$ will be used to enhance the later reasoning process.

Knowledge-Integrated Reasoning Module Event triples contain the possible causes or effects of the input, which are useful for narrative understanding. That is, by attending to the knowledge embeddings, the model should be able to refine the context embeddings to make a more reasonable reasoning process. Our reasoner is a multi-layer Transformer, where each layer is a multi-head self-attention [130] module. The reasoner continually executes self-attention over the context and knowledge

embeddings, and thus can iteratively refine the context embeddings. There are three inputs of the multi-head self-attention: a query Q (context embeddings), key K, and value V (both embeddings of knowledge triples). It relies on scaled dot-product attention to obtain knowledge-enhanced representations for X_i:

$$\mathbf{H}_{X_i}^{kg} = \mathit{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d}})V, \tag{5.14}$$

where $Q = \mathbf{H}_{X_i}, K = V = \mathbf{H}_{et}$, $\mathbf{H}_{X_i}^{kg} \in \mathcal{R}^{M \times d}$. We finally apply the average-pooling operation on the output of the reasoner to obtain knowledge enhanced feature for X_i:

$$\mathbf{h}_{X_i}^{kg} = \text{average-pooling}(\mathbf{H}_{X_i}^{kg}), \tag{5.15}$$

where $\mathbf{h}_{X_i}^{kg} \in \mathcal{R}^d$. The final hidden representation of X_i is obtained by concatenating $\mathbf{h}_{X_i}^{cls}$ and $\mathbf{h}_{X_i}^{kg}$:

$$\mathbf{h}_{X_i} = \text{concatenate}(\mathbf{h}_{X_i}^{cls}, \mathbf{h}_{X_i}^{kg}). \tag{5.16}$$

We next project the hidden representations of all inputs, i.e., $\{\mathbf{h}_{X_i}\}_{i=1}^{I}$, into logit(s) of size I. In the training process, we maximize the likelihood of the gold hypothesis. In the inference process, we select the hypothesis with the max logit, i.e. $y = \max_i(s_i)_{i=1}^{I}$, as the model prediction.

The above is our second stage for narrative understanding. Next, we introduce our second stage for narrative generation.

KG-Enhanced Narrative Generation

Task Definition In this setting, the input X is a text sequence that may consist of several sentences, and we aim to generate another text sequence Y. We extract a relevant two-level KG G in accordance with X to facilitate the reasoning process. The task of narrative generation is then divided into two steps. In the first step, under the condition X, we make a content-planning based on G, i.e., selecting reasonable event sketches E_k. E_k is used as guidance of the later generation process. In the second

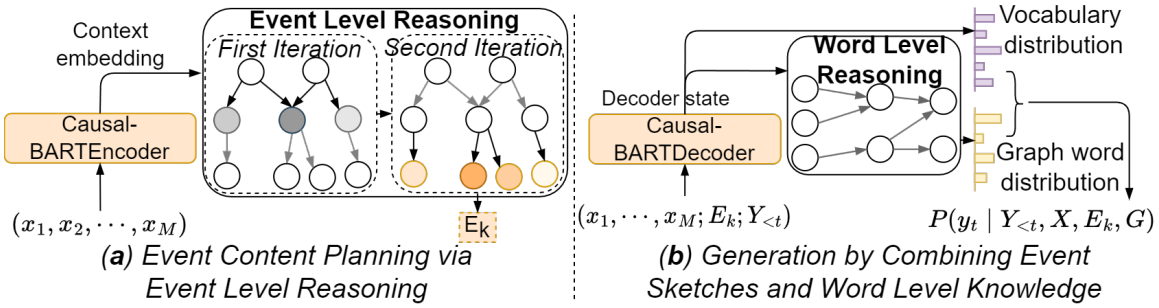


Figure 5.4: Our two-level KG-based reasoning method for narrative generation. In (a), we iteratively calculate the causal scores of one-hop and two-hop events. Color intensity reveals the strength of the scores. In each iteration, we use black arrows to present the used edges, whereas use grey arrows to indicate unused edges. In (b), we combine word-level knowledge to generate text. The process is similar to (a).

step, based on X and E_k, we make full use of word-level knowledge to generate text. Our objective is to maximize the following probability:

$$P(Y|X,G) \propto P(E_k|X,G) \cdot P(Y|X,G,E_k). \tag{5.17}$$

The overall process is shown in Figure 5.4, where we first select guided events (5.2.2), and then combine event guidance and word-level knowledge for generation (5.2.2).

Selecting Event Guidance via Event-level Reasoning Given the input $X = (x_1, x_2, \cdots, x_M)$, we first use our Causal-BART to encode it into a hidden state $\mathbf{h}_X$:

$$\begin{aligned} \{\mathbf{h}_{x_m}\}_{m=1}^{M} &= \text{Causal-BARTEncoder}(X) \\ \mathbf{h}_X &= \text{max-pooling}_m(\{\mathbf{h}_{x_m}\}_{m=1}^{M}), \end{aligned} \tag{5.18}$$

where $\mathbf{h}_{x_m}$ is the embedding of x_m, $\mathbf{h}_X \in \mathcal{R}^d$, d is the hidden size. Then, RGCN is used to encode G to learn the structure-enhanced representations of events and words, as shown in Equation 5.10.

Next, we select guided events by conducting event-level reasoning. Since we are not concerned with how to implement reasoning on the knowledge graph, we directly

take the existing technique: multi-hop reasoning in [39]. As drawn in Figure 5.4 (a), the causal scores of one-hop and two-hop events with regard to X are iteratively calculated, so that we can select reasonable events according to the causal scores of events. In each iteration, the causal scores of same-hop events are concurrently calculated. For an event $t_e \in \mathcal{V}_e$, we gather the neighbors $\mathcal{N}_{t_e}$ of t_e, involving in the connected (head event, event-level relation) pairs, to calculate the causal score $s(t_e)$ between t_e and X:

$$s(t_e) = \frac{1}{|\mathcal{N}_{t_e}|} \sum_{(h_e, r_e) \in \mathcal{N}_{t_e}} (\gamma \cdot s(h_e) + R(h_e, r_e, t_e)), \tag{5.19}$$

where γ (0.5 by default) determines the strength of the causal scores flow from the preceding hop. At first, we assign zero-hop events (central events) with a score of 1, while giving a score of 0 to all other events. $R(\cdot)$ denotes the score of the event tuple (h_e, r_e, t_e) with respect to X in the current iteration, which is obtained via:

$$\begin{aligned} R(h_e, r_e, t_e) &= \text{sigmoid}(\tanh(\mathbf{h}_X^\top \mathbf{W}_{cs}) \cdot \mathbf{h}_{(h_e, r_e, t_e)}) \\ \mathbf{h}_{(h_e, r_e, t_e)} &= \mathbf{W}_e^\top [\mathbf{h}_{h_e}; \mathbf{h}_{r_e}; \mathbf{h}_{t_e}], \end{aligned} \tag{5.20}$$

where $\mathbf{W}_{cs} \in \mathcal{R}^{d \times 3d}$ is a parameter. Finally, we obtain the $s(\cdot)$ of all events, the top-ranked events are chosen as guidance: $E_k = \text{top-k}_i(s(e_i))$, where k denotes the number of selected events. The influence of k is investigated in our experiment.

Text Generation via Combining Event Guidance and Word-Level Knowledge E_k are used as guidance for generating text. To fully utilize knowledge, word-level relations are also considered. Particularly, X and E_k are first concatenated and encoded to get the guided context representations:

$$\mathbf{H}_C = \text{Causal-BARTEncoder}([X; E_k]), \tag{5.21}$$

where $\mathbf{H}_C \in \mathcal{R}^{c \times d}$, c is the number of tokens in $[X; E_k]$.

Then, we use the decoder of Causal-BART to obtain the hidden state of the target

sequence $\mathbf{h}_{y_t}$ at the time-step t:

$$\mathbf{h}_{y_t} = \text{Causal-BARTDecoder}(\mathbf{Y}_{<t}, \mathbf{H}_C). \tag{5.22}$$

In time-step t, the token distribution over the vocabulary V is:

$$P(y_t|Y_{<t}, X, E_k) = \text{softmax}_V(\mathbf{W}_v\mathbf{h}_{y_t} + \mathbf{b}). \tag{5.23}$$

To utilize word-level knowledge, we compute the relevance scores of all words according to the current decoder state $\mathbf{h}_{y_t}$, as shown in 5.4 (b). Specifically, for a word t_w, we gather the neighbors $\mathcal{N}_{t_w}$ of t_w, involving in the connected (head, relation) pairs, to calculate the relevance score $s(t_w)$ between t_w and the current decoder state $\mathbf{h}_{y_t}$:

$$s(t_w) = \frac{1}{|\mathcal{N}_{t_w}|} \sum_{\substack{(h_w, r_w) \in \\ \mathcal{N}(t_w)}} (\gamma \cdot s(h_w) + R(h_w, r_w, t_w)), \tag{5.24}$$

where $R(\cdot)$ is the score of the word-level relation (h_w, r_w, t_w) in accordance to $\mathbf{h}_{y_t}$, which is calculated via:

$$\begin{aligned} R(h_w, r_w, t_w) &= \text{sigmoid}(\mathbf{h}_{y_t}^\top \mathbf{h}_{(h_w, r_w, t_w)}) \\ \mathbf{h}_{(h_w, r_w, t_w)} &= \mathbf{W}_{wv}^\top [\mathbf{h}_{h_w}^L; \mathbf{h}_{r_w}^L; \mathbf{h}_{t_w}^L]. \end{aligned} \tag{5.25}$$

At the beginning, we assign zero-hop words (existed in X) with a score of 1, while giving a score of 0 to all other words. Finally, we obtain the relevance scores of all words, the token distribution over all words in $\mathcal{V}_w$ is:

$$P(y_t^w|Y_{<t}, X, E_k, G) = \text{softmax}_{\mathcal{V}_w}(s(t_w)), \tag{5.26}$$

where $y_t^w \in \mathcal{V}_w$. We combine the distribution over the vocabulary and the distribution over $\mathcal{V}_w$ with a soft gate, to get the final token distribution:

$$\begin{aligned} P(y_t|Y_{<t}, X, E_k, G) = (1 - g_t) \cdot P(y_t|Y_{<t}, X, E_k) \\ + g_t \cdot P(y_t^w|Y_{<t}, X, E_k, G), \end{aligned} \tag{5.27}$$

where $g_t = \text{sigmoid}(\mathbf{W}_\text{g}\mathbf{h}_{y_t})$ is the soft gate, which controls whether to copy words from $\mathcal{V}_w$ when generation.

Model Training We need supervised labels to train the event selection module and the soft gate. But it is impractical to manually annotate labels. Instead, heuristic strategy, i.e., word overlap, is used to obtain supervision. Specifically, we label a word as positive if it exists in the gold sequence. And we label an event as positive if 70% of its component words exist in the gold sequence. The intuition is that an event is good guidance if there is a large word overlap between the event and the gold sequence. We use event labels to train our event selection module (5.2.2), and use word labels to train the soft gate (5.2.2) in decoding.

We adopt the cross-entropy loss of choosing positive events to train the event selection module:

$$J_P = \frac{1}{|\mathcal{V}_e|} \sum_i -l_i \cdot logp(e_i) - (1 - l_i) \cdot log(1 - p(e_i)), \tag{5.28}$$

where l_i is the event label, $p(e_i) = \text{sigmoid}(s(e_i))$ represents the probability that the event e_i gains. We adopt the NLL loss of producing the gold sequence to train the decoder:

$$J_{NLL} = \frac{1}{|Y^{gold}|} \sum_{t=1}^{N} -\log P(y_t^{\text{gold}} | Y_{<t}^{\text{gold}}, X, E_k, G). \tag{5.29}$$

We additionally add the gate loss J_g to supervise the training of the gate g_t, the loss takes the form of binary cross-entropy:

$$J_g = \frac{1}{|Y^{\text{gold}}|} \sum_{t=1}^{N} -(1 - l_{y_t^{\text{gold}}}) \cdot \log(1 - g_t) - l_{y_t^{\text{gold}}} \cdot \log g_t, \tag{5.30}$$

where $l_{y_t^{\text{gold}}}$ is the label for the gold token y_t^{gold}. If y_t^{gold} exists in the two-level KG, i.e., $y_t^{\text{gold}} \in \mathcal{V}_w$, $l_{y_t^{\text{gold}}} = 1$. Otherwise, $l_{y_t^{\text{gold}}} = 0$.

The total loss is $J = J_{NLL} + \alpha J_g + J_P$, where α is set to 0.5 by default. We jointly train the event selection module and the text generator.

Datasets	Train	Validation	Test
COPA	500	N/A	500
BCOPA	500	N/A	500
αNLG	50481	7252	14313
SEG	78530	9816	9816

Table 5.1: The statistics of the used datasets.

5.3 Experiments

5.3.1 Datasets

We evaluate our method on two multi-choice datasets and two text-generation datasets. These datasets are independent of BookCorpus. They are:

- *Choice of Plausible Alternatives* (**COPA**) is a commonsense reasoning dataset in the domain of causality. In this dataset, a system sees a premise sentence and is required to select the reasonable cause or effect of the premise from two hypotheses. There are 1000 manually annotated examples in total, in which 500 for training and 500 for test. *Balanced COPA* (**B-COPA**) extends COPA by adding a mirrored example for each COPA training case, resulting in additional 500 training cases. For each COPA training case, B-COPA retains the two hypotheses, but creates a new premise that matches the wrong hypothesis. B-COPA aims to mitigate the superficial cues in COPA that may be utilized by PLMs like BERT, to improve the robustness of the dataset.

- *Story Ending Generation* (**SEG**) aims to produce a reasonable story ending for an input four-sentence context. The stories originate from ROCStories [75], which contains a wide range of commonsense relationships in human daily life. Consequently, they serve as a useful resource for narrative reasoning. Same as [143], we

randomly divide these stories into the ratio of 8:1:1 for training/validation/test.

- *Abductive Natural Language Genration* (α**NLG**) is a abductive commonsense reasoning dataset. In this dataset, a system sees two observations: O_1 and O_2, and is asked to produce an explanation to make O_1 and O_2 be the cause and consequence of the explanation, respectively. The official data partition [5] is adopted for training/validation/test.

The statistics of the used datasets are shown in Table 5.1. For COPA and B-COPA, we use the given premise to retrieve the two-level knowledge graph. For αNLG and SEG, we use the two observations and the four-sentence story context to retrieve the two-level knowledge graph, respectively. The statistics of the retrieved two-level knowledge graphs are shown in Table 5.2. *#AvgEventNode* and *#AvgEventRel* denote the average number of event-level nodes and relations, respectively. *#AvgWordNode* and *#AvgWordRel* denote the average number of word-level nodes and relations, respectively.

Graph Statistics	COPA	B-COPA	SEG	αNLG
#AvgEventNode	36	24	79	73
#AvgEventRel	43	29	82	78
#AvgWordNode	126	84	211	204
#AvgWordRel	253	202	741	639

Table 5.2: Statistics of retrieved multi-level knowledge graphs.

5.3.2 Baselines and Experimental Setting

Baselines

For COPA and B-COPA, we consider the two recent PLMs: **BERT** [15] and **RoBERTa** [66]. We evaluate BERT and RoBERTa, including both the base and large versions, on the COPA and B-COPA. For αNLG and SEG, we consider the following baselines:

- **GPT2-FT** which is a fine-tuned GPT2 by [39].
- Pre-trained models T5[104] and BART[50]. We fine-tune them two and denote them as **T5-FT** and **BART-FT**.
- **GPT2-OMCS** [39] is a knowledge-enhanced GPT-2 which first post-trained on Open Mind Commonsense (OMCS) [3], then fine-tuned on αNLG and SEG.
- **GRF** [39] is a GPT2 based model which grounds reasoning on commonsense knowledge graphs. We additionally use BART as the backbone to reproduce GRF to evaluate its performance under different PLMs. We denote this model as **GRF-BART**.

Experimental Setting

In the first stage, we search hyper-parameters according to the performance on the training set of COPA and BCOPA, and the validation set of SEG and αNLG. In the second stage, we perform 10-fold cross-validation on the training set of COPA and B-COPA to find the best parameters, which is the same as [56]. And we use BLEU-2 on the validation set of SEG and αNLG to select the best parameters. The some of parameters, including learning-rate, batch-size, and num-epoch, are shown in Table 5.3. In addition, in KG-enhanced multi-choice inference, we select a 2 layers

[3]http://openmind.media.mit.edu

Setting		Learning Rate	Batch Size	Num Epoch
Stage1	Multi-Choice	5e-6	4	2
	Text Generation	5e-6	16	5
Stage2	Multi-Choice	5e-6	4	10
	Text Generation	1e-5	16	10

Table 5.3: The some of searched parameters in our experiment.

transformer, where each layer has 4 heads. We adopt a 2-layer RGCN module in the second stage, and adopt the Adam optimizer with a linearly-decreased learning rate in our experiment. In the text generation setting, the beam search strategy with a beam size of 3 is adopted by our approach and all baselines we create.

5.3.3 Results on Multi-Choice Tasks

We evaluate our method under two settings: (1) the zero-shot setting that we directly evaluate our causal-enhanced PLMs on COPA and B-COPA testset; (2) the fine-tuning setting that we fine-tunes our method on the training set and then evaluate on the test set. Table 5.4 presents the result under the fine-tuning setting. We have following observations.

- As shown in Line #6, the causal-RoBERTa-large achieves the 90.3%/90.0% accuracy by using COPA and B-COPA training set, respectively, obtains 0.6%/0.8% increase compared with RoBERTa-large (Line #4). This indicates that sentence-level causalities are helpful for commonsense causal reasoning.

- After combining sentence-level causalities and two-level knowledge graph, our $\text{MCNR}_{\text{RoBERTa-large}}$ (Line #7) achieves a 91.2%/91.5% accuracy, a further 0.9%/1.5% improvement on the basis of causal-RoBERTa-large (Line # 6). This

demonstrates the effectiveness of grounding reasoning on the explicit two-level knowledge graph.

We additionally perform the ablation study to verify the modules in the two-level KG-based reasoning process.

- As shown in Line #8, we ablate sentence-level causalities. In other words, we use the off-the-shelf RoBERTa-large as the basis and utilize two-level KGs for reasoning. We find that this variant has a better performance than causal-RoBERTa-large (Line #6). This shows that grounding reasoning on explicit knowledge graphs is better than injecting sentence-level causalities into PLMs. The possible reasons lies in two aspects: (1) in the fine-tuning process, the model may gradually forget the sentence-level causal knowledge which is related to the input context, and (2) the two-level KG may provide background causal knowledge which is more related to the input.

- On the basis of the variant in Line #8, we further remove word-level knowledge. That is, we only utilize event causalities. As shown in Line #9, this variant leads to a 0.2%/0.4% performance drop. This demonstrates that word-level knowledge can be complementary to event causalities. The possible reason is that word-level knowledge helps to mitigate the sparsity of events, as well as provides additional knowledge for narrative understanding.

Table 5.5 presents the result of our causal-enhanced PLMs under the zeroshot setting. Our causal-RoBERTa-large obtains 78.4% accuracy on the COPA testset, which performs better than some fine-tuned PLMs[56]. This further demonstrates that sentence-level causalities are helpful to narrative commonsense reasoning.

Line	Methods	COPA	B-COPA
	BERT-base [57]	74.5	76.3
	BERT-large [112]	75.0	N/A
	BERT-large [44]	76.5	74.5
	BERT-large [57]	77.8	80.0
	RoBERTa-base [57]	80.5	81.3
	RoBERTa-large [44]	87.7	89.0
	RoBERTa-large [57]	90.3	90.2
#1	BERT-base (Ours)	74.6±0.7	75.0±1.1
#2	BERT-large (Ours)	77.0±1.1	78.7±1.0
#3	RoBERTa-base (Ours)	80.2±0.6	79.4±0.8
#4	RoBERTa-large (Ours)	89.7±0.5	89.2±1.1
#5	Causal-BERT-large (Ours)	78.1±0.7	79.6±0.9
#6	Causal-RoBERTa-large (Ours)	90.3±1.0	90.0±0.6
#7	$\text{MCNR}_{\text{RoBERTa-large}}$ (Ours)	**91.2±0.6**	**91.5±0.5**
	Ablation Study		
#8	w/o 1 (Ours)	90.9±0.8	91.0±0.6
#9	w/o 1,2 (Ours)	90.7±0.5	90.6±0.8

Each result is reported as the mean of five models trained with random seeds, with the standard deviation.

1: sentence-level causalities. 2: word-level knowledge.

Table 5.4: Accuracy (%) on COPA and B-COPA testset under the fine-tuning setting. Scores with **bold** denote the best results.

Methods	COPA (train)	B-COPA (train)	COPA (test)
BigramPMI [26]	N/A	N/A	63.4
PMI [26]	N/A	N/A	65.4
CausalNet+PMI [67]	N/A	N/A	70.2
Multiword+PMI [113]	N/A	N/A	71.4
This article			
Causal-BERT-base	59.0	62.1	67.8
Causal-BERT-large	64.4	65.6	70.8
Causal-RoBERTa-base	62.6	65.2	70.8
Causal-RoBERTa-large	80.4	78.8	78.4

Table 5.5: Accuracy on COPA and B-COPA under zero-shot setting.

5.3.4 Results on Text Generation Tasks

Evaluation Metrics

The textual-overlap based metrics, including BLEU-n [87], METEOR [3] and ROUGE-L [61], are used to automatically evaluate the similarity between a generated text and a collection of references offered by the datasets. These precise string-matching metrics might not effectively identify paraphrases and capture crucial semantic ordering alterations [98]. So we additionally use model-based metrics, e.g. BertScore [153], to evaluate the quality of the generated text. We also report Distinct [54] score to measure the diversity of generated sequences.

Zero-shot Evaluation

We first present the zero-shot evaluation results of causal-T5 and causal-BART, by comparing them with off-the-shelf T5 and BART. As shown in Table 5.6, whether

Datasets	αNLG		SEG	
	BLEU-2/4	ROUGE-L	BLEU-1/2	ROUGE-L
T5	6.41/1.31	13.32	9.70/3.62	13.32
Causal-T5	7.59/1.23	16.94	17.17/5.15	17.09
BART	6.70/1.40	15.33	10.92/4.12	13.28
Causal-BART	10.62/1.77	17.01	19.59/6.10	20.70

Table 5.6: Zero-shot evaluation results on the testsets of SEG and αNLG.

based on T5 or BART, the causal-enhanced model has achieved consistent improvement, which shows the effectiveness of sentence-level causal knowledge.

Our method vs. Previous methods

The results on the testsets of αNLG and SEG are shown in Table 5.7. We have the following observations.

- BART-FT outperforms GPT2-FT, and T5-F, so we replicate GRF [39] based on BART. In addition, we observe that the result of GRF-BART is superior to the original one [39]; BART is the reason for this improvement.

- Causal-T5 and causal-BART, perform better than T5-FT and BART-FT, respectively, which shows that sentence-level causalities are useful for commonsense text generation.

- The result of GPT2-MOCS is not significantly better than that of GPT2-FT. On the contrary, GRF-BART performs better than BART-FT by a large margin. The possible reason is that explicit knowledge graphs benefit commonsense reasoning tasks that emphasize reasoning and explanation. This suggests that

αNLG					
Models	BLEU-4	METEOR	ROUGE-L	Distinct-3	BERTScore
GPT2-FT†	9.80	25.82	32.90	N/A	N/A
GPT2-OMCS†	9.62	25.83	32.88	N/A	N/A
GRF†	11.62	27.76	34.62	N/A	N/A
T5-FT	12.62±0.07	28.97±0.17	35.54±0.10	16.36±0.06	55.70±0.05
BART-FT	12.99±0.07	29.77±0.23	34.25±0.18	16.35±0.27	55.45±0.09
GRF-BART	14.82±0.06	31.70±0.10	36.04±0.14	16.41±0.21	56.20±0.05
Causal-T5	12.91±0.06	29.39±0.07	35.84±0.10	16.31±0.09	55.76±0.04
Causal-BART	13.15±0.02	30.21±0.06	34.59±0.09	16.88±0.21	55.54±0.04
$\text{MCNR}_{\text{BART}}$	**16.06±0.01**	**33.14±0.09**	**37.23±0.09**	**27.54±0.31**	**56.61±0.02**
SEG					
Models	BLEU-2	METEOR	ROUGE-L	Distinct-3	BERTScore
GPT2-FT†	10.20	N/A	N/A	N/A	N/A
GPT2-OMCS†	10.40	N/A	N/A	N/A	N/A
GRF†	11.00	N/A	N/A	N/A	N/A
T5-FT	9.40±0.05	17.52±0.07	25.32±0.02	42.95±0.48	48.34±0.03
BART-FT	10.35±0.00	18.78±0.02	26.25±0.04	47.09±0.28	49.04±0.03
GRF-BART	11.23±0.02	19.66±0.03	26.95±0.02	52.42±0.27	49.76±0.02
Causal-T5	9.55±0.02	17.69±0.02	25.40±0.05	43.41±0.15	48.43±0.01
Causal-BART	10.57±0.03	19.04±0.04	26.46±0.02	48.68±0.09	49.15±0.02
$\text{MCNR}_{\text{BART}}$	**13.03±0.02**	**21.81±0.02**	**28.98±0.02**	**55.18±0.32**	**51.05±0.02**

Table 5.7: The results of automatic evaluation on the testsets of αNLG and SEG. Each result is reported as the mean of five models trained with random seeds, with the standard deviation. Values with † denote the values are borrowed from [39]. Scores with **bold** denote the best results.

grounding reasoning on explicit KG is more efficient than implicitly injecting knowledge into PLMs.

- Our $\text{MCNR}_{\text{BART}}$ shows a substantial improvement over GRF-BART. This is due to the fact that GRF-BART only uses word-level knowledge, whereas our method uses sentence-level and event-level causalities. This shows that our framework is effective for narrative text generation.
- We find that our method has a better performance on the Distinct score. The possible reason is that multi-level knowledge provides a more relevant background to the input context, which prevents a model from generating generic text, and hence improves the text diversity.

The overall result demonstrates the superiority of our framework over compared baselines.

Ablation Study

We further perform the ablation study to explore the impact of different components of our method on narrative text generation. We divide our full method into the following components: (1) "SC" denotes the **s**entence-level **c**ausalities enhanced post-training (5.2.1); (2) "TKE" denotes the **t**wo-level **k**nowledge graph **e**ncoding module; (3) "EPS" denotes the **e**vent **p**rompts **s**election module (5.2.2); (4) "GWK" denotes the module which **g**enerates text with **w**ord-level **k**nowledge (5.2.2). We also validate whether the heuristically obtained event labels are helpful for selecting reasonable event guidance by ablating J_p (Equation 5.28).

Starting from our full model, we gradually ablate different modules to explore their impact. The results are shown in Table 5.8. The following are our observations.

- "w/o 1": ablating sentence-level causalities leads to performance drop, e.g. the BLEU-4 drops about 0.24 on the αNLG test set. This shows that injecting

αNLG					
Models	BLEU-4	METEOR	ROUGE-L	Distinct-3	BERTScore
$\text{MCNR}_{\text{BART}}$	**16.06±0.01**	**33.14±0.09**	**37.23±0.09**	**27.54±0.31**	**56.61±0.02**
w/o 1	15.82±0.05	32.81±0.18	36.87±0.16	27.54±0.36	56.53±0.06
w/o 1,4	15.46±0.09	32.48±0.26	36.67±0.14	25.81±0.70	56.49±0.08
w/o 1,4,2	15.27±0.02	32.43±0.19	36.57±0.16	22.04±0.81	56.38±0.09
w/o 1,2,3	14.78±0.07	31.85±0.15	36.08±0.05	16.45±0.37	56.23±0.02
w/o J_p	15.24±0.08	32.32±0.26	36.48±0.22	21.18±3.59	56.34±0.10
SEG					
Models	BLEU-2	METEOR	ROUGE-L	Distinct-3	BERTScore
$\text{MCNR}_{\text{BART}}$	**13.03±0.02**	**21.81±0.02**	**28.98±0.02**	**55.18±0.32**	**51.05±0.02**
w/o 1	12.86±0.04	21.42±0.04	28.73±0.05	53.53±0.80	50.86±0.05
w/o 1,4	12.54±0.04	21.13±0.07	28.60±0.04	50.66±0.23	50.75±0.04
w/o 1,4,2	12.41±0.04	20.98±0.03	28.47±0.02	50.56±0.11	50.65±0.02
w/o 1,2,3	10.98±0.08	19.32±0.15	26.67±0.20	49.48±1.37	49.57±0.08
w/o J_p	11.52±0.09	19.83±0.14	27.50±0.16	46.29±0.34	49.94±0.14

Table 5.8: Each result is reported as the mean of five models trained with random seeds, with the standard deviation. 1: SC. 2: TKE. 3: EPS. 4: GWK.

sentence-level causalities into BART helps these two tasks. In other words, αNLG and SEG tasks need causal knowledge.

- "w/o 1,4": we do not utilize word-level knowledge in the decoding process. Compared with the result in "w/o 1", this variant leads to a performance drop. This indicates that word-level knowledge can provide additional evidence in the generation process.

- "w/o 1,4,2": Based on the variant in "w/o 1,4", we keep removing the "TKE" module and obtain the event embeddings through the embedding layer of BART.

As shown in "w/o 1,4", the result of this variant decreases. The possible reason is that utilizing the "TKE" module is helpful for selecting more reasonable event guidance. This shows that the "TKE" module can integrate word-level relationships into event embeddings, thus reducing the sparsity of events.

- "w/o 1,2,3": we ablate event-level causalities, and only utilize word-level knowledge for text generation. In this case, this variant has degraded to GRF-BART. The slight result gap between this variant and GRF-BART may be caused by the difference in data pre-processing, e.g. the different number of word-level relations. After removing event-level knowledge, the result decreases significantly. This shows that event-level knowledge is essential for text generation.

- "w/o J_p" leads to a large drop. This is consistent with humans that event labels are helpful for selecting more reasonable guided events.

- By comparing the results between line "w/o 1,4,2" and line "w/o 1,2,3", we observe that removing event-level causalities leads to a larger result drop than removing word-level knowledge. This suggests that event-level causalities are more beneficial than word-level knowledge. This is reasonable because event-level knowledge contains richer information than word-level knowledge. And guided events can provide the skeleton information for narrative text generation.

- The result of combining event causalities and word-level knowledge is better than using only one kind of them. This shows that word-level knowledge complements event causalities. The reasons are two-fold: (1) word-level knowledge captures the interaction between event components, which is helpful for choosing reasonable event guidance; (2) two-level knowledge provides more related background to input context, which helps to generate high-quality text.

The ablation study result demonstrates that each component contributes to narrative text generation.

Datasets	Models	Informativeness		Reasonability	
		W(%)	L(%)	W(%)	L(%)
αNLG	vs. BART-FT	19.33	10.33	26.33	12.33
	vs. GRF-BART	19.33	12.67	24.00	10.67
	vs. w/o 1,2,3	22.00	14.67	27.67	17.67
	vs. w/o 1.4	15.33	7.67	18.00	9.33
SEG	vs. BART-FT	18.57	8.67	25.00	10.33
	vs. GRF-BART	17.67	11.00	25.67	15.33
	vs. w/o 1,2,3	17.00	9.00	28.67	11.67
	vs. w/o 1,4	17.67	9.33	20.33	11.67

Table 5.9: Manual evaluation results on two datasets. Scores indicate the percentage of Win (W) and Lose (L).

Manual Evaluation

We perform the manual evaluation to manually judge the performance of our method under narrative text generation. *Informativeness* and *reasonability* are evaluation criteria. The informativeness of the generated text denotes whether it contains non-genetic information that is related to the input sequence. The reasonability of the generated text denotes whether it is causal and temporal related to the input context. BART-FT, GRF-BART, "w/o 1,2,3" and "w/o 1,4" are respectively compared with our $\text{MCNR}_{\text{BART}}$. 100 test cases as well as the text generated by each model are randomly sampled from SEG and αNLG, respectively. Given the input context in each test case, three annotators are asked to make a judgment among "win", "lose" and "tie" between a pair of sequences generated by $\text{MCNR}_{\text{BART}}$ and a compared baseline. To ensure a fair judgment of employed metrics, the annotators are restricted to students who have research experience in text generation. The manual evaluation

result is presented in Table 5.9. $\text{MCNR}_{\text{BART}}$ is superior to all compared models. Fleiss's kappa coefficient is computed to assess the inter-rater agreement. For SEG, the coefficient of informativeness and reasonability is 0.419 and 0.501. For αNLG, the two values are 0.486 and 0.462, respectively. This indicates that the inter-rater agreement shows a moderate ($0.4 \leq \kappa < 0.6$) agreement. We notice that in the pairwise comparisons, annotators gave a tie result for the most of compared pairs. After checking the generated text of compared models, we find that these models generate text with good quality. This may be because these models are all based on BART, which has a powerful ability in text generation. Our method further enhances the quality of generated text on the basis of BART, and exhibits superior performance compared to others, underscoring the effectiveness of our approach.

5.3.5 Additional Analyses

Deeper Investigation of Two-Level KG

we conduct a preliminary study on the SEG testset to explore how word-level knowledge complements event causalities, The compared models include: (1) "w/o 1,2,4" which only utilizes event-level causalities, (2) "w/o 1,2,3" which only utilizes word-level knowledge and (3) "w/o 1" which utilizes the two-level knowledge. On the basis of whether the two-level KG of a test case has at least a positive event or a positive word[4], we separate the test samples as well as the texts generated by selected models into four classes. The amount of test cases in each class is (A) 434, (B) 1417, (C) 1500, and (D) 6467. We compute BLEU-2 of texts generated by different models in each class. Figure 5.5 presents the result, from which we find:

- In A and B, "w/o 1,2,4" may succeed in choosing a supporting event for text

[4]Positively-labeled events or words are denoted as supporting events or words for the sake of simplicity.

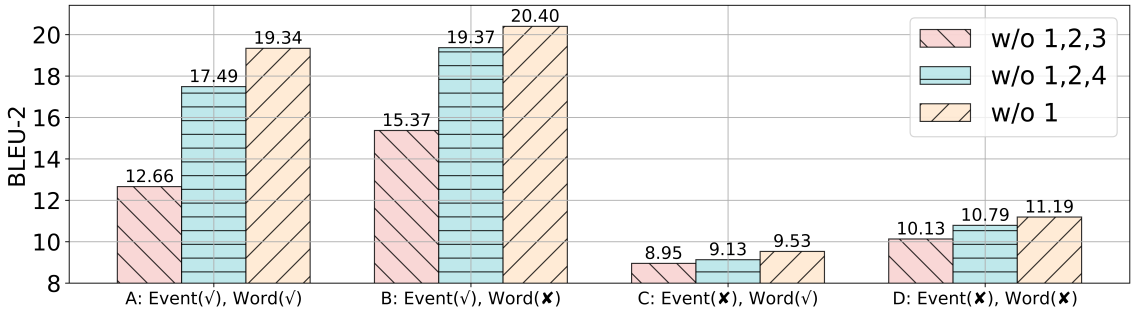


Figure 5.5: In A, each two-level KG has at least one supporting word and one supporting event. In B, each two-level KG has at least one supporting event, but no supporting words.

generation, therefore "w/o 1,2,4" has a far better performance than "w/o 1,2,3".

- The result difference between "w/o 1,2,3" and "w/o 1,2,4" in class C is the most minor. Note that "w/o 1,2,3" may choose supporting words but "w/o 1,2,4" can never choose any supporting event. Hence, the resulting gap between the two models is narrowed. In other words, even if a two-level KG lacks supporting events, it may still have supporting words that are helpful for text generation. This illustrates how word-level knowledge complements event causalities.

- The best performance is achieved when combining two-level knowledge. That is, being aware of more supporting knowledge, our method is able to generate high-quality text.

Influence of the number of selected events

We explore the influence of the number of guided events E_k on the SEG testset. Figure 5.6 presents the result. The majority of metrics experience an initial increase, reaching their peak at $k = 3$, followed by a subsequent decline. One contributing factor is that as k increases, the likelihood of selecting a supporting event also rises. On another note, an increased number of chosen events enhances the probability

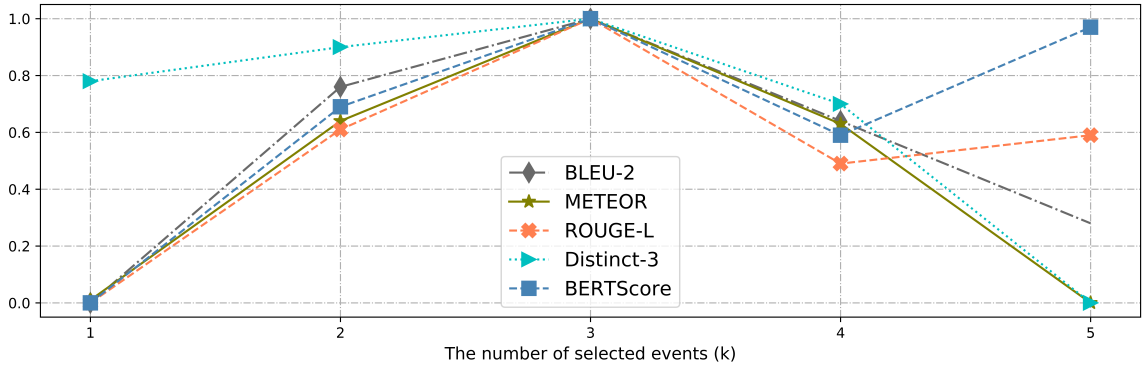


Figure 5.6: The influence of the number of selected guided events. The results of different metrics are normalized to 0-1.

of selecting irrelevant events. We set k to 3 by default since the two aspects are well-balanced in this case.

Performance under the low-resource scenario

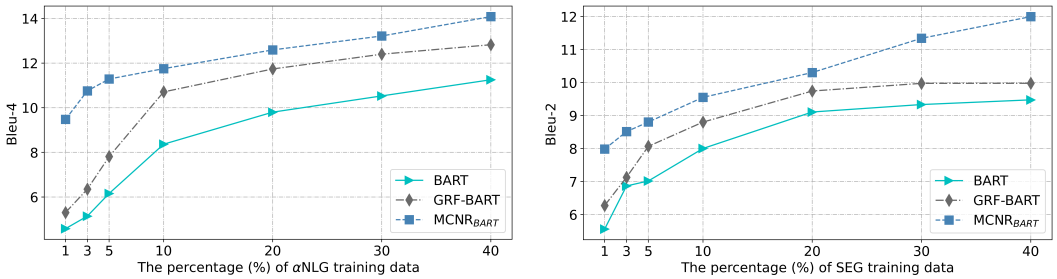


Figure 5.7: Performance under the low-resource scenario.

We systematically reduce the amount of training data and subsequently evaluate our model on the test sets of SEG and αNLG. The results, depicted in Figure 5.7, highlight the outcomes. Our method consistently demonstrates improvements, even when trained with minimal data (1%). In contrast, both BART and GRF-BART experience more pronounced performance declines. This emphasizes the robustness of our approach, attributed to the integration of multi-level knowledge.

Case Study

#1	O_1	It was a bright, warm day.
	O_2	Joe regret going outside.
	BART-FT	Joe went outside to play.
	GRF-BART	Joe went outside and it started to rain.
	$\text{MCNR}_{\text{BART}}$	Joe got sunburned.
#2	Story Context	Lisa had a job interview in three days . She was nervous and unprepared. Lisa decided to study for the interview. She practiced for nine hours every day.
	BART-FT	Lisa got the job.
	GRF-BART	Lisa got the job.
	$\text{MCNR}_{\text{BART}}$	Lisa did well at the interview and was hired.
#3	Premise	The man urgently leaped out of bed.
	Hypothesis_1	He wanted to shut off the alarm clock. (✓)
	Hypothesis_2	He wanted to iron his pants before work.
	Ask-for	Cause
	Knowledge	leaped out of bed $\overset{\text{Causes}}{\longleftarrow}$ alarm clock went off alarm clock went off $\overset{\text{xWant}}{\longrightarrow}$ to turn off

#1 is from the αNLG testset. #2 is from the SEG testset.
#3 is from the COPA testset.

Table 5.10: Case study.

Table 5.10 presents some cases as well as predictions of our method and compared baselines. Compared with BART-FT and GRF-BART, we observe from #1 and #2 that our method can produce more informative and reasonable text. In #3,

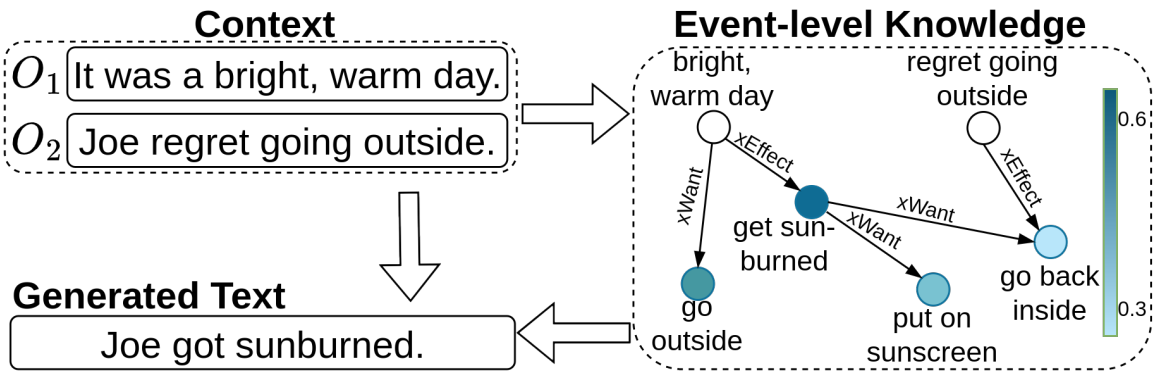


Figure 5.8: The event-level subgraph of case #1. The darker color indicates the higher relevance score (Equation 5.19).

our framework gives the correct prediction, but RoBERTa-large fails. We visualize some related event-level causalities, which supports our method to give the correct prediction. In Figure 5.8, we present a sub-graph of the event causalities related to case #1. The event "get sunburned" gains the largest causal score, and is selected as guidance for generating "Joe got sunburned.". These findings indicate that our method has good explainability.

Error Analysis

We conduct an error analysis to investigate the limitations of our method. As shown in Table 5.11, all models, including ours, generate unreasonable text. For instance, in case #1, the context involves implicit negation, yet all models struggle to discern this nuance, leading to incorrect endings. In case #2, O_2 suggests that Fred does not succeed in losing weight, but all models fail to understand this implication. These instances highlight the models' inability to handle these implicit cases, emphasizing the ongoing research value in the domain of narrative reasoning.

#1	Original Story	Todd wanted to buy a mouse for PC gaming. He found that most options were expensive and gimmicky. There were a few reasonable-looking and high-quality options. But these were outside of his budget.
	BART-FT	Todd decided to buy a mouse for his gaming console.
	GRF-BART	Todd bought the mouse and was happy with his purchase.
	$\text{MCNR}_{\text{BART}}$	Todd bought the mouse and enjoyed his PC gaming experience.
#2	O_1	Fred made a bet with Sam over who could lose more weight in a month.
	O_2	Fred cancelled the bet at the end of the month.
	BART-FT	Fred lost the bet and Sam lost.
	GRF-BART	Fred lost a lot of weight.
	$\text{MCNR}_{\text{BART}}$	Fred lost a lot of weight.

#1 is from the SEG testset. #2 is from the αNLG testset.

Table 5.11: Error Analysis.

5.4 Discussion

Our research provides a comprehensive method that utilizes multi-level knowledge, which has important influence for narrative reasoning. The novel hierarchical KG

provides an intuitive and structured way for communities to understand complex relationships. This directly promotes the research progress of KG-based narrative reasoning. By combining hierarchical knowledge graphs with knowledge-enhanced PLMs, we make full use of diverse knowledge and achieve significant improvements in narrative reasoning. Researchers can use this technology to accurately understand and solve problems, and promote research progress in various fields.

Our method is a general approach that can be applied to diverse real-world applications, such as question-answering, dialogue systems, story generation, causal prediction, and so on. Our method also has enlightening implications for retrieval augmented generation (RAG). In RAG, it is often necessary to use queries of different granularities to retrieve diverse evidence. How to effectively utilize evidence of different granularities has become a problem. Our method has effectively organized and utilized multi-level knowledge. This provides a case for the RAG system.

5.5 Chapter Summary

We present a two-stage framework that utilizes sentence-level and event-level causalities for narrative commonsense reasoning. In the first stage, we utilize sentence-level causalities to enhance PLMs. As the carrier of sentence-level causalities, causal-enhanced PLMs can be easily transferred to downstream tasks. In the second stage, we propose the hierarchical two-level knowledge graph to mitigate the event sparsity problem. Then we ground narrative reasoning on the hierarchical knowledge graph. Numerous experiments illustrate the effectiveness of our framework. We also notice that temporal relation, a significant discourse relation, plays an important role in our daily life. It should be useful for understanding narratives. Future research can consider integrating our method with temporal relationships to further expand the dimensions of knowledge. This development will enable communities to understand and utilize information more comprehensively and diversely, providing stronger support

for narrative commonsense reasoning.

Chapter 6

A Causal Approach for Counterfactual Reasoning in Narratives

Beyond factual reasoning, another important problem in narrative reasoning is counterfactual reasoning, since it is a direct verification of the causal perception ability of narrative reasoning systems.

6.1 Introduction

Counterfactual reasoning in narratives (CRN) refers to the prediction of alternative events and their potential outcomes, diverging from what actually occurred [98, 2]. Specifically, given the observed narrative $S = (c, x, y)$, where c, x, and y denote the context, condition, and outcome, respectively, CRN considers how y' would be if keeping the context c unchanged while perturbing x to a similar but different x'. Figure 6.1 presents a case of CRN.

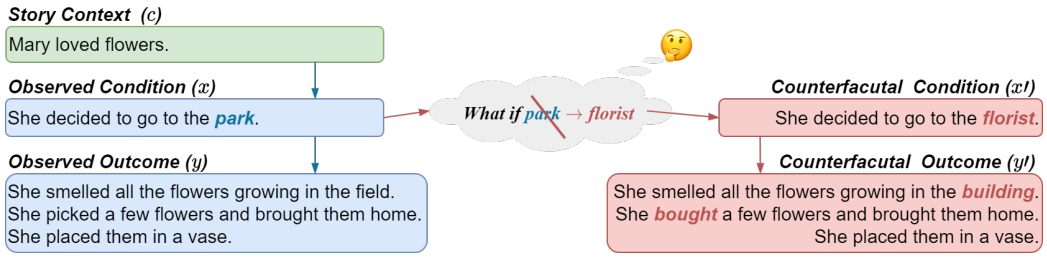


Figure 6.1: An example of counterfactual reasoning in narratives. The example comes from TimeTravel [98]. The colored text in the counterfactual outcome denotes the modified parts.

Even though it is considered a crucial component of intelligent systems [90, 92], only a few resources have been devoted to CRN. Some of the works [29, 12, 53] design dataset-specific heuristic methods, but they are actually abusing unique patterns, i.e., the feature of minimum editing, in the dataset, which limits the generality of their methods. Other works [98, 159] take advantage of the progress of pre-trained language models (PLMs), and fine-tune PLMs for CRN, i.e., learning the conditional distribution $p(\mathbf{y}'|\mathbf{c}, \mathbf{x}', \mathbf{S})$. Despite the success of simulating real examples, the conditional distribution is notorious for being susceptible to exploiting artifacts of the dataset, instead of learning to robustly reason about counterfactuals [98]. For example, the models often directly copy the original y or learn to paraphrase y without acknowledging the counterfactual condition [98, 29]. As a result, the predicted counterfactual outcome y' usually conflicts with the counterfactual condition x'.

Generally, CRN relies on the ability to find causality in narratives [12], i.e., y' should express a clear causal relation to x' to make it clear how the perturbation makes the observed outcome change. This issue naturally lends itself to formulation using causal mechanisms [91], which requires us to infer the background knowledge that is compatible with (c, x', y'). However, this is non-trivial as it involves estimating the posterior of the background knowledge. Luckily, with the variational technique [45], we are able to use the background compatible with the observed S to approximate the

posterior distribution. In fact, the variational process provides an approximation of the background of (c, x', y'), but it may face the problem of model collapse [106]. As a result, the generated y' may not be the precise effect of x', and the resulting model may be sub-optimal. To mitigate this problem, we further propose two intuitive strategies, which introduce a pre-trained classifier and commonsense causality, to enhance the causality between (c, x') and the generated y'.

In this work, we propose a causal approach for CRN. We utilize the variational process to approximate the *implicit* background of counterfactual scenarios. In addition, we devise two strategies to alleviate the model collapse problem in this variational process. First, inspired by research on natural language inference [42, 19], we want to ensure that the generated y' entails its true condition x'. In other words, the model should correctly learn the influence of the condition on the outcome. Therefore, we introduce a pre-trained classifier that estimates the likelihood of a text y entails an input (c, x). We use the Gumbel-softmax technique [38, 34] to enable gradient back-propagation. Second, we exploit COMeT [37] to retrieve diverse event causality tailored for (c, x'), which allows for deducing plausible event sequences and provides an *explicit* background for the unobserved counterfactual outcome y'.

To summarize, we formulate CRN in a variational framework and introduce event causality and a pre-trained classifier to further improve the causality between x' and the generated y'. Our method is a general approach that is applicable to multiple tasks. The experiment proves the effectiveness of our method. We also study the practicality of the generated counterfactual narratives via a data augmentation experiment.

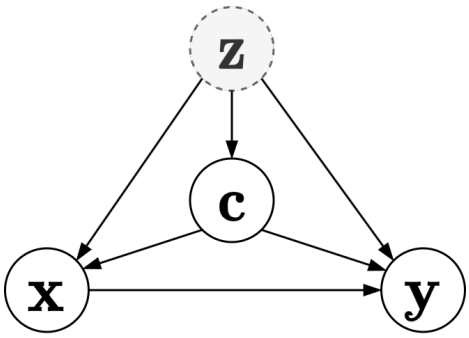


Figure 6.2: The proposed structural causal model. The dashed circle indicates that the variable is latent, while the solid circle indicates that the variable is observed.

6.2 Methods

6.2.1 Problem Setting with Causal Mechanism

Given a narrative $S = (c, x, y)$, we perturb x into a counterfactual condition x' and want to predict the new outcome y'. To solve this problem, we need to speculate on the background knowledge compatible with (c, x', y'), which allows us to predict the precise effect of x'. This problem is naturally suitable to be expressed with a causal mechanism. Figure 6.2 shows the structural causal model (SCM) [90] that describes the generation process of narratives. Here the latent variable $\mathbf{z}$ denotes the unobserved background knowledge. The SCM thus defines a joint distribution:

$$p(\mathbf{y}, \mathbf{x}, \mathbf{c}, \mathbf{z}) = p(\mathbf{y}|\mathbf{x}, \mathbf{c}, \mathbf{z})p(\mathbf{x}, \mathbf{c}|\mathbf{z})p(\mathbf{z}), \tag{6.1}$$

where $p(\mathbf{z})$ is a standard Gaussian distribution following common practices. Similarly, conditioned on the observed S, the joint distribution of the counterfactual scenario is defined as:

$$p(\mathbf{y}', \mathbf{x}', \mathbf{c}, \mathbf{z}|\mathbf{S}) = p(\mathbf{y}'|\mathbf{x}', \mathbf{c}, \mathbf{z}, \mathbf{S})p(\mathbf{x}', \mathbf{c}|\mathbf{z}, \mathbf{S})p(\mathbf{z}|\mathbf{S}), \tag{6.2}$$

where $p(\mathbf{y}'|\mathbf{x}', \mathbf{c}, \mathbf{z}, \mathbf{S})$ is the decoder model requiring us to infer $\mathbf{z}$ from all (S, c, x', y') data. However, the inference of $\mathbf{z}$ involves estimating the posterior distribution of the knowledge, i.e., $p(\mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{y}', \mathbf{S})$. We next introduce our basic variational process to

approximate the distribution.

6.2.2 The Basic Variational Objective

Variational Inference

Our basic objective follows the common VAE approach [45]. By introducing the approximate network $q(\mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{y}', \mathbf{S})$, a lower bound of the model's marginal log-likelihood (that marginalizes out $\mathbf{z}$) is:

$$\begin{aligned} \log p(\mathbf{y}'|\mathbf{c}, \mathbf{x}', \mathbf{S}) &= \log \int_{\mathbf{z}} p(\mathbf{y}', \mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{S}) \\ &\geq \mathbf{ELBO} = \mathbf{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{c},\mathbf{x}',\mathbf{y}',\mathbf{S})} \log \frac{p(\mathbf{y}', \mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{S})}{q(\mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{y}', \mathbf{S})}. \end{aligned} \tag{6.3}$$

For simplicity, we denote $q(\mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{y}', \mathbf{S})$ as $q(\mathbf{z}|\cdot)$. Then, according to Equation 6.2, we have:

$$\begin{aligned} \mathbf{ELBO} &= \mathbf{E}_{\mathbf{z}\sim q(\mathbf{z}|\cdot)}[\log \frac{p(\mathbf{y}', \mathbf{z}, \mathbf{c}, \mathbf{x}'|\mathbf{S})}{q(\mathbf{z}|\cdot)} - \log p(\mathbf{c}, \mathbf{x}'|\mathbf{S})] \\ &\approx \mathbf{E}_{\mathbf{z}\sim q(\mathbf{z}|\cdot)}[\log p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S}) + \log p(\mathbf{c}, \mathbf{x}'|\mathbf{z}, \mathbf{S})] \\ &\quad - \mathrm{KL}[q(\mathbf{z}|\cdot)||p(\mathbf{z}|\mathbf{S})], \end{aligned} \tag{6.4}$$

where $p(\mathbf{c}, \mathbf{x}'|\mathbf{S})$ is a constant for the given dataset and independent of the parameterized model. Hence, given the labeled set $\mathcal{D}$ which contains all (S, c, x', y') examples, our basic objective is:

$$\begin{aligned} \mathcal{L}_{\mathrm{VAE}} = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} &\mathbf{E}_{\mathbf{z}\sim q(\mathbf{z}|\cdot)}[\log p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S}) \\ &+ \lambda_x \log p(\mathbf{c}, \mathbf{x}'|\mathbf{z}, \mathbf{S}) + \lambda_k \mathrm{KL}[q(\mathbf{z}|\cdot)||p(\mathbf{z}|\mathbf{S})]. \end{aligned} \tag{6.5}$$

λ_x and λ_k are hyper-parameters. We use the cyclic schedule [51] to anneal λ_k from 0 to 1 to avoid excessive regularization of the KL term.

$p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$ **vs.** $p(\mathbf{y}'|\mathbf{c}, \mathbf{x}', \mathbf{S})$

Current generative models follow the auto-regressive paradigm, but suffer from exposure bias. Note that (c, x, y) and (c, x', y') have similar content. When inference, given the input (S, c, x'), $p(\mathbf{y}'|\mathbf{c}, \mathbf{x}', \mathbf{S})$ have no information about the gold y', so it may paraphrase y. Differently, we encode y' into $q(\mathbf{z}|\cdot)$, and use the KL term to bridge the gap between $q(\mathbf{z}|\cdot)$ and $p(\mathbf{z}|\mathbf{S})$. When inference, we sample $\mathbf{z} \sim p(\mathbf{z}|\mathbf{S})$ and feed it into $p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$. This can somewhat alleviate the issue of exposure bias and mitigate the problem of paraphrasing y.

Model Implementation

We use PLMs, e.g., BART [50], as backbone to implement $p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$. We first encode the input part (S, c, x') into the context vectors $\mathbf{H}_C = \text{BARTEncoder}(S, c, x')$, where $\mathbf{H}_C \in \mathcal{R}^{l \times d}$, l is the total length of $[S; c, x']$, d is the hidden size. To fuse $\mathbf{z} \sim q(\mathbf{z}|\cdot)$ into PLMs, as suggested in [51], we concatenate $\mathbf{z}$ with $\mathbf{H}_C$, and pass it into the decoder for autoregressive learning. The hidden state of t-th time step of the target sequence $\mathbf{h}_{y_t}$ is computed by:

$$\mathbf{h}_{y_t} = \text{BARTDecoder}(Y_{<t}, [\mathbf{H}_C; \mathbf{z}]). \tag{6.6}$$

The word distribution of t-th time-step over the standard vocabulary V is:

$$P(y_t|Y_{<t}) = \text{softmax}_V(\mathbf{W}_v \mathbf{h}_{y_t} + b). \tag{6.7}$$

To implement $q(\mathbf{z}|\cdot)$ and $p(\mathbf{z}|\mathbf{S})$, we approximate them to Gaussian distributions. We use the pre-trained BARTEncoder to initialize different text encoders, which are used to encode S and (c, x', y', S). Following several linear layers, we obtain the mean and log-variance of two distributions, which are used to calculate the KL loss. To implement $P(\mathbf{c}, \mathbf{x}'|\mathbf{z}, \mathbf{S})$, we adopt the in-batch contrastive learning. For the positive example (c, x', z, S), we collect different $\bar{x'}$ from the mini-batch and regard $(c, \bar{x'}, z, S)$

as negative examples. Then the representations of examples are projected into scalar values for binary classification.

Training with the above base objective alone can lead to model collapse, i.e., the KL term tends to be zero. As a result, the decoder will ignore the information from $q(\mathbf{z}|\cdot)$, and the generated text is not the precise result of x'. We next introduce our two strategies, which introduce the pre-trained classifier and external event causality to improve the causality between (c, x') and the generated y'.

6.2.3 Introducing the Pre-trained Classifier

Intuitively, we expect that the generated y' truly entails its condition x'. To achieve this goal, we pre-train a classifier $f([c, x, y])$ that estimates the likelihood of the input (c, x) entailed by the output y. Motivated by [12], we use the training set of the used datasets to obtain positive and negative examples. For example, given the example (c, x, y, x', y'), (c, x') should entail by y' but contradict with y, and (c, x) should entail by y but contradict with y'. That is, (c, x, y) and (c, x', y') are positive, and (c, x', y) and (c, x, y') are negative. We initialize $f(\cdot)$ with BARTEncoder to keep the embedding space the same as the generator.

Then, we train the generator so that its predicted outcome entails the corresponding condition with a high likelihood measured by the classifier:

$$\mathcal{L}_{\text{Cla}} = -\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}} \mathbf{E}_{\mathbf{z}\sim q(\mathbf{z}|\cdot),\tilde{y}\sim p(\mathbf{y}|\mathbf{z},\mathbf{c},\mathbf{x},\mathbf{S}')}[\log f([c, x, \tilde{y}]) + \log(1 - f([c, x', \tilde{y}]))], \quad (6.8)$$

where $S' = (c, x', y')$ and $p(\mathbf{y}|\mathbf{z}, \mathbf{c}, \mathbf{x}, \mathbf{S}')$ is the mirror of $p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$. Here, we consider $p(\mathbf{y}|\mathbf{z}, \mathbf{c}, \mathbf{x}, \mathbf{S}')$ rather than $p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$ because it has been optimized in Equation 6.4. In fact, we use the classifier to restrict the generated $\tilde{y}$ entails its true condition x but contradicts with x'. As in [35, 34], we use Gumbel-softmax technique to enable gradient backpropagation for the discrete text.

6.2.4 Utilizing External Event Causalities

The variational process provides an *implicit* background for unobserved counterfactual outcomes, this further motivates us to utilize external event causalities which allows for introducing diverse event commonsense and providing an *explicit* background for generating counterfactual outcomes.

Retrieving Event Causality

We use COMeT [37] as the event knowledge base. We first feed the zero-hop events (c, x') into COMeT to generate one-hop events with corresponding relations. The one-hop events are then fed into COMeT to generate two-hop events. The details are same as in Chapter 5. We next organize the retrieved knowledge into an event graph $G = (V, E)$ where V denotes the node set and E denotes the edge set. Each node $e \in V$ is an event which is a word sequence. Each edge in E is a tuple (e_h, r, e_t) containing a head event e_h, a relation r, and a tail event e_t. Then, we perform reasoning on G to select guided events, which are the possible effects of (c, x'). We use the selected events as guidance for generating y'.

Selecting Guided Events

Motivated by [39, 77], we perform multi-hop reasoning on G to select important event nodes. We iteratively compute the relevance scores of multi-hop events with respect to (c, x'), as shown in Figure 6.3(a). In each iteration, we parallelly calculate the scores of events in the same hop. For the tail event e_t, the score $s(e_t)$ is calculated by polymerizing information from its neighbors $\mathcal{N}_{e_t}$ including pairs of (e_h, r):

$$s(e_t) = \frac{1}{|\mathcal{N}_{e_t}|} \sum_{(e_h, r) \in \mathcal{N}_{e_t}} (s(e_h) + R(e_h, r, e_t)). \tag{6.9}$$

At the beginning, zero-hop events, i.e., (c, x'), are assigned a score of 1, e.g., $s(c) = s(x') = 1$, while other events are assigned a score of 0. $R(\cdot)$ is the relevance of the

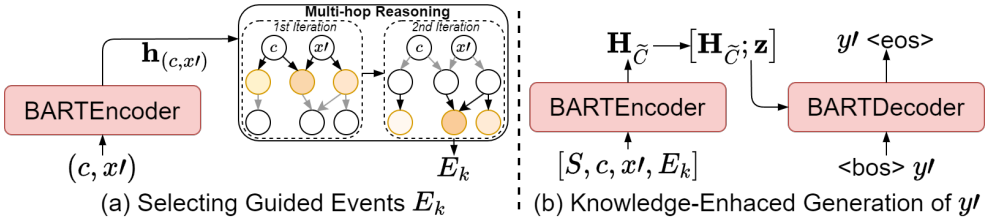


Figure 6.3: (a) The scores of one-hop and two-hop events are parallelly calculated in each iteration. Color intensity indicates the score difference. In each iteration, the black arrows denote used edges, while the grey arrows denote unused edges. (b) We concatenate E_k with (S, c, x') for the auto-regressive decoding.

edge (e_h, r, e_t) with respect to the (c, x'), which is calculated by:

$$R(e_h, r, e_t) = \sigma(\mathbf{h}_{(c,x')}^T \mathbf{W}_k \cdot [\mathbf{h}_{e_h}; \mathbf{h}_r; \mathbf{h}_{e_t}]), \tag{6.10}$$

where $\mathbf{W}_k \in \mathcal{R}^{d \times 3d}$, $[\cdot;\cdot]$ denotes the concatenation, $\mathbf{h}_{(c,x')} \in \mathcal{R}^d$ is the embedding of (c, x'), $\mathbf{h}_{e_h}, \mathbf{h}_r, \mathbf{h}_{e_t}$ are the embeddings of e_h, r, e_t.

We select the top-k events according to their scores: $E_k = \text{topk}_i(s(e_i))$. k is set to 4 after searching on the dev set. To fuse the guided E_k into the generation, we concatenate (S, c, x') with E_k, and pass them to BARTEncoder to obtain knowledge-enhanced context vectors $\mathbf{H}_{\tilde{C}} = \text{BARTEncoder}(S, c, x', E_k)$. Then, we concatenate $\mathbf{H}_{\tilde{C}}$ with $\mathbf{z} \sim q(\mathbf{z}|\cdot)$, and feed it into the decoder for autoregressive learning, i.e., $y' \sim p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S}, \mathbf{E}_k)$, as shown in Figure 6.3(b).

6.2.5 Training and Inference

Similar to [77], we add an additional object to guide the event selection. We maximize the probability of selecting positive events by:

$$\mathcal{L}_E = \frac{\sum_{\mathcal{D}}}{|\mathcal{D}|} \sum_i -l_i \log p(e_i) - (1 - l_i) \log(1 - p(e_i)), \tag{6.11}$$

where $p(e_i) = \sigma(s(e_i))$ is the probability that the event e_i is selected. l_i is the label of e_i which is subject to the overlap between e_i and the gold y'. The details are in

Appendix **??**. The final object is:

$$\mathcal{L} = \mathcal{L}_{VAE} + \alpha\mathcal{L}_{Cla} + \beta\mathcal{L}_{E}, \tag{6.12}$$

where α and β are hyper-parameters.

When inference, given input (S, c, x'), we first sample $\mathbf{z} \sim p(\mathbf{z}|\mathbf{S})$, then we select guided event E_k according to (c, x'), at last we generate the counterfactual outcome $y' \sim p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S}, \mathbf{E}_k)$.

6.3 Experiment

Datasets	Train	Dev	Test
TimeTravel	28363	1871	1871
PossibleStories	3404	458	671

Table 6.1: Statistics of the datasets used in this work.

6.3.1 Datasets

We evaluate our method on two datasets.

- **TimeTravel** [98] is a dataset designed for counterfactual story rewriting. It builds upon the ROCStories [75] corpus, comprising numerous five-sentence stories $S = s_{1:5}$. In this setup, s_1 serves as the context c, s_2 as the condition x, and $s_{3:5}$ establish the outcome y. TimeTravel involves the rewriting of the initial condition x by humans into a counterfactual condition x', followed by annotators making minimal edits to the original ending y to generate the counterfactual outcome y'. One of the primary challenges in TimeTravel is balancing the generation of natural stories with minimal modifications to the original y.

- **PossibleStories** [2], also built on the ROCStories corpus, considers the problem that possible consequences for the same context may vary depending on the situation we refer to. It is originally a multiple-choice dataset, where each example consists of the original context c, the original ending y, the counterfactual question x', and candidate options including the counterfactual ending y'. To adapt it to text generation, we set the original condition x as a simple text "*what's the most likely story ending?*", then we generate y' according to (c, x, y, x'). The statistics of two datasets are in Table 6.1.

6.3.2 Baselines

We produce the following kinds of baselines:

- **Prompting large chat models**, e.g., ChatGLM2(6B) [148], Llama2Chat(7B) [124], ChatGPT [84]. We use one-shot prompting for experiments, the used prompts are in Table 6.2 and 6.3.

- **Supervised fine-tuning**. We fine-tune several pre-trained language models, including GPT2(base) [102], T5(base) [105], BART(base) [50], and Llama2(7B) [124]. We use QLoRA [14] to adapt Llama2(7B) on a single RTX 3090 GPU.

For TimeTravel, we additionally compare ours with some task-specific methods:

- DELOREAN [99] and EDUCAT [12] regard the task as a controllable generation problem, and unsupervised edit the original y to the counterfactual y'.

- CLICK [52], a two-stage method, first detects which words in the original ending need to be modified, and then implements the modification.

Tasks	Prompt
TimeTravel	Each story contains 5 sentences, where the first two sentences are the story premise, and the last 3 sentences are the story ending. I will apply subtle a perturbation to the second sentence, making the first two sentences a counterfactual story premise. Due to the slight perturbation, the counterfactual premise is very similar to the original premise, with only some words being different. According to the original story and the counterfactual story premise, you are required to predict the counterfactual story ending. Note that the counterfactual story ending should be similar to the original story ending, as well as being coherent with the counterfactual story premise. Here is one example: ### <Original 5-sentences story> 1. Bella wanted to cook some spaghetti and meatballs. 2. She discovered she had no pasta noodles. 3. She found a recipe online that used spaghetti squash instead. 4. Bella luckily had a spaghetti squash on hand. 5. She was surprised to find the spaghetti and meatballs delicious! <Counterfactual story premise> 1. Bella wanted to cook some spaghetti and meatballs. 2. She realized she didn't have the time to make it properly so she changed made an omelette instead. <Counterfactual story ending > 3. She found a recipe online that used egg whites instead. 4. Bell luckily had many eggs on hand. \\ 5. She was surprised to find the egg white omelette delicious! ### Now, given the following example, please write the counterfactual story ending. There should be only three sentences at the counterfactual story ending. <Original 5-sentences story> {original_story} <Counterfactual story premise> {counterfactual_premise} <Counterfactual story ending>

Table 6.2: The prompts used for the TimeTravel dataset.

Implementation Details

We use the train set of the two datasets to train the classifier. We use the AdamW optimizer and set lr to 5e-6. We select checkpoint according to F1 on the dev set. The best checkpoint achieves the F1 scores of 66.1 and 70.1 in the test set of two datasets. When training the counterfactual generator, we use the AdamW optimizer and set lr to 5e-5. We linearly decrease lr to zero with a 10% warmup ratio. We

Tasks	Prompt
PossibleStories	You will observe a story that consists of a context and an ending. Then given the counterfactual question, please generate a new story ending that is compatible with the question. Here is an example: ###<Observed story context> Fred and James both claimed they were the best basketball player. One day they decided to find out who was better. James loved to brag, but Fred was focused on the game. Eventually Fred beat James by 1 point. <Observed story ending> James learned that day to focus on the game, not on bragging. <Counterfactual question> What is most likely to happen if Fred has a lot of empathy for others? <Counterfactual story ending> Fred felt bad that he won, so the next game he eased up and let James win. ### Now, given the following example, please write the counterfactual story ending. You can only generate one sentence, do not add additional content. <Observed story context> {original_context} <Observed story ending> {original_ending} <Counterfactual question> {cf_context} <Counterfactual story ending>

Table 6.3: The prompts used for the PossibleStories dataset.

Datasets	bs	lr	α	β	λ_x
TimeTravel	8	5e-5	1.0	0.5	1.0
PossibleStories	8	5e-5	0.5	0.5	0.5

Table 6.4: The searched hyper-parameters.

search for the best hyper-parameters according to ENTScore on the dev set of each dataset. The searched parameters are in Table 6.4. When inference, we adopt the

multinomial sampling strategy to generate y', and we repeat for 5 times to calculate the average performance.

6.3.3 Automatic Evaluation

Metrics For Timetravel, we follow the previous works and use BLEU [87], BertScore [153], ENTScore [12], and HMean = $\frac{2 \cdot \text{BLEU} \cdot \text{ENTScore}}{\text{BLEU+ENTScore}}$ [12] as metrics. BLEU and BertScore evaluate the similarity between the generated y' and the ground truth. ENTScore evaluates the coherence between (c, x') and the generated y'. For PossibleStories, we use BLEU, BertScore, and ENTScore as metrics.

Our Method vs. Baselines

The automatic evaluation result is shown in Table 7.3 and 6.6. We can see BART generally performs better than GPT2 and T5, therefore we use BART as the backbone. In addition, we observe that:

- In Table 7.3, unsupervised editing-based methods have poor performances, indicating that this kind of unsupervised approach is unable to produce qualified counterfactual stories.
- Compared with BART, our method achieves an obvious improvement, especially in the ENTScore metric, e.g., obtaining a 4.2/4.0 gain on two datasets. In addition, our method outperforms Llama2(7B), which indicates that our method is effective in improving the causality between (c, x') and the generated y'.
- Due to the extremely large-scale pre-training, Chat models, e.g., ChatGLM2, Llama2Chat, and ChatGPT, have a strong ability to generate coherent stories. However, chat models get a low BLEU and BertScore, indicating that they tend to less consider what has happened.

TimeTravel				
Methods	BLEU	BertS.	ENTS.	HMean
Prompting Chat Models				
ChatGLM2(6B)	16.5	60.0	66.2	26.4
Llama2Chat(7B)	16.9	58.8	77.8	27.8
ChatGPT	36.4	69.8	**82.6**	50.6
Unsupervised Editing-based Methods				
DELOREAN	23.9	59.9	51.4	32.6
EDUCAT	44.1	74.1	32.3	37.3
Supervised Fine-tuning				
CLICK	46.7	73.2	36.7	41.1
GPT2	63.5(0.2)	77.8(0.3)	43.5(1.0)	51.6(0.7)
T5	71.2(0.3)	**80.1(0.1)**	42.7(0.8)	53.3(0.6)
BART	66.5(0.3)	79.4(0.2)	52.0(1.0)	58.3(0.6)
Llama2(7B)	**70.3(0.4)**	79.9(0.2)	54.1(0.7)	60.9(0.5)
Ours	67.0(0.1)	79.5(0.1)	56.2(0.4)	**61.1(0.2)**
Ablation Experiment				
w/o Clas	67.5(0.2)	79.8(0.1)	54.6(0.6)	60.4(0.4)
w/o Event	65.6(0.4)	79.0(0.1)	55.2(0.5)	60.0(0.4)
w/ VAE	65.9(0.3)	79.2(0.1)	54.1(0.6)	59.4(0.4)

Table 6.5: The automatic and ablation-study result on TimeTravel. We report the mean(std) under 5 random experiments. Scores with **bold** denote the best results.

- On TimeTravel, the ENTScore result of our method is not as good as the results of chat models, but our method achieves the best trade-off between BLEU and ENTScore. On PossibleStories, our method approximates ChatGPT

PossibleStories			
Methods	BLEU	BertScore	ENTScore
Prompting Chat Models			
ChatGLM2(6B)	1.9	48.4	38.8
Llama2Chat(7B)	3.0	49.9	43.8
ChatGPT	5.0	53.5	**48.5**
PLMs-based Finetuning			
GPT2	6.0(0.7)	49.4(0.3)	37.3(0.4)
T5	5.7(0.3)	49.2(0.3)	35.8(0.7)
BART	13.2(0.5)	53.8(0.2)	42.9(1.0)
Llama2(7B)	**16.3(1.1)**	54.4(0.6)	45.1(0.9)
Ours	16.1(0.2)	**56.2(0.1)**	46.9(1.0)
Ablation Experiment			
w/o Clas	15.7(0.4)	55.6(0.3)	45.6(0.7)
w/o Event	15.5(0.4)	55.8(0.2)	46.0(0.5)
w/ VAE	15.5(0.5)	55.9(0.3)	45.0(0.4)

Table 6.6: The automatic and ablation-study result on PossibleStories. We report the mean(std) under 5 random experiments. Scores with **bold** denote the best results.

and surpasses ChatGLM2 by a large margin. This indicates that the small-model-based sophisticated method is expected to be comparable to LLM-based prompting, indicating that it still has research value in the era of LLMs.

Ablation Study

Settings To investigate the effectiveness of different components, we devise the following ablated variants to compare with our full model. (1) "w/o Event" means

we do not use event causality. (2) "w/o Cla" means we remove the pre-trained classifier. (3) "w/ VAE" means we ablate both event causality and the pre-trained classifier. In this case, this variant degenerates into the basic VAE module.

Result The ablation study result is shown in Table 7.3 and 6.6. We have the following observations.

- Compared with BART, "w/ VAE" achieves a 2.1/2.1 gain in ENTScore on both datasets. This demonstrates the effectiveness of the variational process. The possible reason is that the variational process learns an approximation of the latent $\mathbf{z}$, which provides an implicit background for generating counterfactual outcomes.

- Compared with "w/ VAE", "w/o Clas" and "w/o Event" achieve higher ENTScore on both datasets, indicating the two strategies contribute to improving the causality between x' and the generated y'. This makes sense because (1) external event causality provides a causal background for generating y' and (2) the classifier will punish the unqualified generation. The best result is achieved when combining two strategies, indicating that two strategies complement each other.

- "w/o Clas" performs better than "w/o Event" on both datasets. This shows that the classifier is more important than event knowledge. The possible reason lies in two aspects. (1) Though retrieved event causality may contain useful information for generation, unrelated and noisy knowledge may also be retrieved. (2) The classifier directly measures the causality between the condition and the generated outcome, and penalizes incoherent generation.

Methods	TimeTravel				PossibleStories			
	MinEdits		Coherence		Similarity		Coherence	
	W	L	W	L	W	L	W	L
vs. w/o Clas	14.7	23.7	28.0	10.3	10.0	9.0	16.3	7.0
vs. w/o Event	17.0	25.3	37.3	10.0	13.3	7.0	24.3	6.7
vs. Llama2Chat(7B)	61.0	11.0	24.3	37.7	32.3	12.7	41.7	20.7
vs. ChatGPT	52.3	15.7	16.7	47.0	21.7	13.0	23.7	27.3

Table 6.7: The manual evaluation result. MinEdits denotes Minimal-Edits.

6.3.4 Manual Evaluation

Setting For TimeTravel, we follow previous works and use *Minimal-Edits* and *Coherence* as manual evaluation metrics. *Coherence* denotes the logical consistency between the counterfactual context (c, x') and generated y'. *Minimal-Edits* denotes the extent of minimal revision between the original y and the generated y'. For PossibleStories, we use Similarity and Coherence as metrics, where *Similarity* evaluates the similarity between the generated y' and the ground truth. We carry out pairwise comparisons between our method with some baselines, including Llama2Chat, ChatGPT, and two ablated models "w/o Event" and "w/o Clas". We randomly sample 100 cases from the two test sets for each pair of models, respectively. Three annotators are recruited to make a preference among Win, Tie, and Lose given the input and two outputs generated by our model and a baseline respectively. The annotators are research students from the field of commonsense text generation to make sure they have a fair judgment of used metrics.

Result The result is shown in Table 6.7. Compared with the two ablated variants, our full method shows an increase in Coherence, but a decrease in Minimal-Edits. This is because both of the two strategies prevent copying the original ending y.

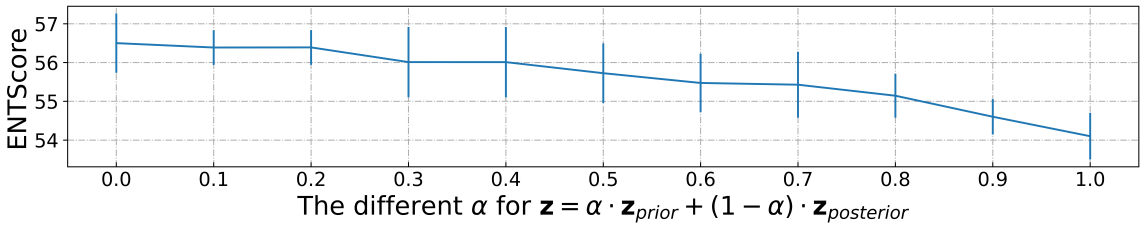


Figure 6.4: Linearly interpolating $\mathbf{z}_{prior}$ and $\mathbf{z}_{posterior}$ for the VAE decoding, i.e., $y' \sim p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$.

On TimeTravel, our full model performs better in Minimal-Edits, but not as well in Coherence as the chat models. This is consistent with automatic evaluation. We calculate Fleiss's kappa reliability as the inter-rater agreement. For TimeTravel, the agreement of Minimal-Edits and Coherence is 0.43 and 0.56. For PossibleStories, the agreement of Similarity and Coherence is 0.50 and 0.52.

6.3.5 Further Discussion

Analyzing the VAE Module by Manipulating z

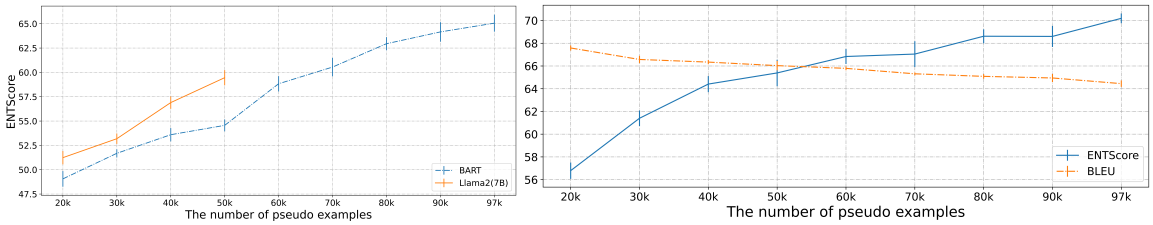
Since the strength of our VAE module lies in its ability to approximate the posterior distribution, we are interested in whether the encoded $\mathbf{z}$ benefits counterfactual narrative generation. We conduct a pilot study on TimeTravel. We first replace $\mathbf{z} \sim p(\mathbf{z}|\mathbf{S})$ with a random noise $\mathbf{z}_{noise} \sim \mathcal{N}(0, 1)$, and feed $\mathbf{z}_{noise}$ into the VAE decoder $p(\mathbf{y}'|\mathbf{z}, \mathbf{c}, \mathbf{x}', \mathbf{S})$ for generation. We get a 43.8 ENTScore, which is significantly worse than the result of BART. This is reasonable because the random noise disrupts the model structure and brings about a significant negative impact. Next, we make a linearly interpolation $\bar{\mathbf{z}} = \alpha \cdot \mathbf{z}_{prior} + (1 - \alpha) \cdot \mathbf{z}_{posterior}$, where $\mathbf{z}_{prior} \sim p(\mathbf{z}|\mathbf{S})$ and $\mathbf{z}_{posterior} \sim q(\mathbf{z}|\mathbf{c}, \mathbf{x}', \mathbf{y}', \mathbf{S})$, and feed $\bar{\mathbf{z}}$ for generation. The result is shown in Figure 6.4. The larger the proportion of $\mathbf{z}_{posterior}$, that is, the more posterior information about the gold y', the better the result is achieved. When using $\mathbf{z}_{posterior}$ for gener-

ation, we get a 56.5 ENTScore, but the value is still not satisfactory enough. The possible reason is that too much information is lost during the process of encoding the sequence into a vector $\mathbf{z}$, making it hard for the model to reconstruct the gold y'. According to above results, we speculate that the more effective solution to this problem is to introduce more diverse and more large-scale data. Therefore, we conduct the following data augmentation experiment.

Generating Counterfactual Stories for Data Augmentation

[98] provides an additional data partition that only has counterfactual conditions x' but no counterfactual outcomes. This partition contains about 97k examples. We use this partition to study the practicality of the generated counterfactual stories via a data augmentation experiment. Specifically, we use our ablated variant "w/o Event" as the generator since its performance is not significantly worse than our full model, and there is no need for external knowledge, making it easy to use. For each (S, c, x'), we use "w/o Event" to generate 60 candidates and keep the one with the highest ENTScore as the pseudo counterfactual outcome, denoted as $\widetilde{y'}$. Finally, we obtain the pseudo set $\mathcal{D}_P = \{(S, c, x', \widetilde{y'})\}$. We test this set for both generation and classification tasks.

Testing for the Generation Task First, We only use $\mathcal{D}_P$ to directly fine-tune BART and Llama2, i.e., learning $p(\mathbf{y}'|\mathbf{c}, \mathbf{x}', \mathbf{S})$. The result is shown in Figure 6.5(a). When training with about 32k pseudo examples, BART achieves a 52.0 ENTScore, which is obtained using the labeled set $\mathcal{D}$. When using more pseudo examples for training, the result continuously improves. We have a similar observation from the result of Llama2(7B). Because finetuning Llama2(7B) is time-consuming, e.g., it takes about 1.5 hours to train an epoch with 30k samples, we use a maximum of 50k samples for fine-tuning. Next, we mix the labeled set $\mathcal{D}$ and a different number of pseudo examples to fine-tune BART. The result is shown in Figure 6.5(b). When mixing



(a) (b)

Figure 6.5: (a): Fine-tuning BART and Llama2(7B) with a different number of pseudo examples. (b): Fine-tuning BART via mixing the labeled set $\mathcal{D}$ and a different number of pseudo examples.

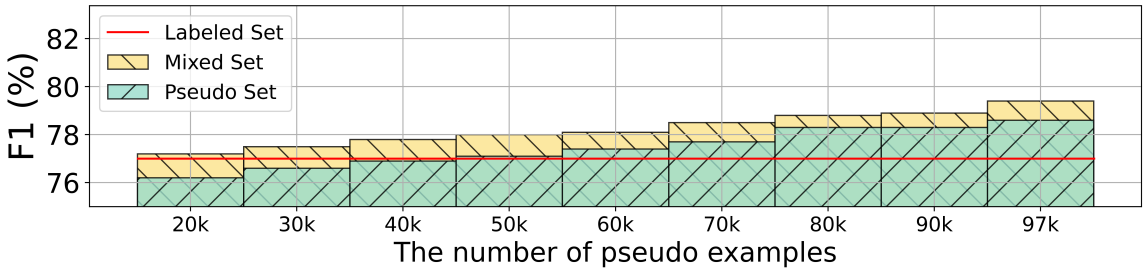


Figure 6.6: Fine-tuning RoBERTa-large with different types of training examples.

$\mathcal{D}$ and all pseudo examples, BART obtained a 70.1 ENTScore, which is better than ChatGLM2 and closer to Llama2Chat. However, as the number of pseudo examples increases, BLEU continues to decline, but overall the decline is acceptable.

Testing for the Classification Task Motivated by [12], we construct a binary classification task to test the quality of pseudo examples, which is the same as training the classifier. We explore three types of training examples to train RoBERTa-large [66] and then validate on the test set of TimeTravel: (1) the labeled set $\mathcal{D}$, (2) the pseudo set $\mathcal{D}_P$ in which we randomly sample a different number of pseudo examples, and (3) the mixed set in which we mix the label set $\mathcal{D}$ and a different number of pseudo examples. The F1 result on the test set of TimeTravel is shown in Figure 6.6. When training with more pseudo examples, F1 achieves a stable improvement. The

F1 under the mixed set is better than that under the labeled set, indicating that the pseudo set is an effective supplement to the labeled set.

Overall, these results demonstrate the practicality of the generated pseudo examples, which further proves the effectiveness of our method.

Case Study and Error Analysis

Table 6.8 presents a case study. The counterfactual ending generated by ChatGPT is coherent, but it differs significantly from the original ending. Our model generates a coherent counterfactual ending with minimal-edits. However, we find that the issue of paraphrasing y still exists in our method, as shown in Table 6.9. However, this issue is less likely to occur in large chat models. We speculate that there are two reasons: (1) the problem of exposure bias cannot be completely eliminated; (2) The used model is small, and the scale and diversity of data are insufficient.

6.4 Disscusion

We construct our method based on small-scale datasets and the small model, e.g., BART, therefore our method cannot outperform large language models. In the experiment, we find that fine-tuned Llama2 performs better than fine-tuned BART. This indicates that constructing our method upon larger pre-trained models may have a better performance. In addition, the generated counterfactual stories are beneficial for counterfactual narrative reasoning. This foreshadows the future direction, that is, we can transform x into different x' through different perturbations, thus generating diverse counterfactual stories for data augmentation. Different from predicting counterfactual outcomes, it is easy to perturb x into x', and there have been a lot of related research works. We leave this in the future work.

Original Story	A man dug a well on his farm. Instead of water, though, he struck oil. Jubilant, he placed some calls. A large oil company arrived the next day. They bought the land from the man for a million dollars.
Counterfactual Context	A man dug a well on his farm. Instead of water, though, he struck fossils.
w/o Clas	Jubilant, he placed some calls. A large fossils company arrived the next day. They bought the land from the man for a million dollars.
w/o Event	Jubilant, he placed some calls. A large oil company arrived the next day. They bought the land from the man for a million dollars.
Llama2Chat	Excited, he placed some calls. A paleontology company arrived the next day. They bought the land from the man for a large sum of money!
ChatGPT	Jubilant, he contacted a local museum to share his discovery. A team of paleontologists arrived the next day. They offered to buy the fossils from the man for a substantial amount of money.
Ours	Jubilant, he placed some calls. A large fossil company arrived the next day. They bought the fossils from the man for a million dollars.

Table 6.8: A case study with the generated texts by different models. The case is from the test set of TimeTravel.

Original Story	Megan loved her sock monkey. She took it to her grandad's house when she visited him. Megan got home and realized she had left her monkey. I had to meet grandad halfway to his house and pick up her monkey.
Original ending	Megan was so happy and she was then able to go to bed.
Counterfactual question:	Why was it so important to get the sock monkey back before bedtime?
w/o Clas	The monkey needed to be taken back before Megan got to play with it.
w/o Event	Megan was so happy and she was then able to go to bed.
Llama2Chat	Megan needs to play with her sock monkey before going to bed.
ChatGPT	Megan couldn't sleep without her sock monkey by her side.
Ours	Megan was so happy and she was then able to go to bed.

Table 6.9: A case for error analysis. The case is from the test set of PossibleStories.

6.5 Chapter Summary

In this work, we formulate counterfactual reasoning in narratives in a VAE framework. In addition, we introduce a pre-train classifier and external event causality to further

improve the causality between the counterfactual condition and the generated counterfactual outcome. The experiment proves the effectiveness of our method. We also conduct a data augmentation experiment to verify the practicality of our method.

Part III

Hard Negatives Mining for Narrative Coherence Learning

Chapter 7

Generating Contrastive Narratives Using the Brownian Bridge Process for Narrative Coherence Learning

7.1 Introduction

Narrative reasoning [10, 135] is an account of the development of events, along with explanations of how and why these events happened [36], which has provoked a variety of applcations, including commonsense causal reasoning [108, 27, 67], abductive reasoning [5], and so on.

A major challenge for narrative reasoning is to evaluate narrative coherence [75]. Existing methods mainly focus on devising self-supervised tasks, in which positive samples are from large-scale real narratives [75, 145], and negative samples are created by sampling-based strategies. For example, [138, 63, 126] create negative samples by shuffling or masking real narratives. [47] incorporates randomly sampled sequences and model-completed [102, 6] sequences as negative samples. However, these strategies are generally coarse-grained and superficial. The resulting negatives still face

problems of low quality, such as being irrelevant or repetitive [47], making them less representative, and easily distinguishable.

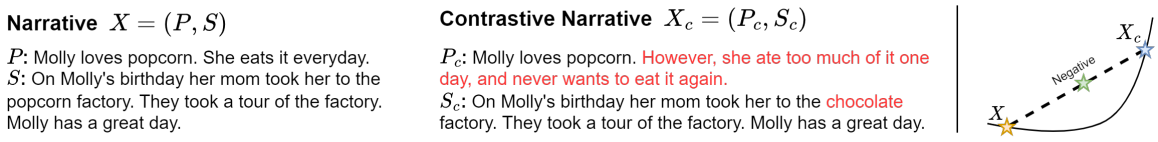


Figure 7.1: We define that an example consists of a prefix (P) and a suffix (S). **Left**: An ideal contrastive narrative X_c, which is similar with X but conveys different semantics. Text with red color denotes the difference. **Right**: The solid line denotes the data manifold. The dashed line represents the methods for synthesizing negative samples, such as Mixup [149] or crisscrossing. As X_c approaches X, the corresponding negative sample should be more "hard".

Hard negatives are critical in the contrastive learning framework [136, 74, 142]. The ideal of hard negative samples should be that are similar to a real narrative but actually less coherent. To mine such negatives, a possible approach is to introduce contrastive narratives. Contrastive narratives are examples that are similar in content, but convey different semantics [69, 132]. Due to this property, we can crisscross[1] a narrative and its contrastive variants to obtain negative samples, as shown in Figure 7.1. The resulting negatives should be similar to the real narratives but less coherent, making them good candidates for hard negatives. However, existing works for collecting contrastive narratives rely heavily on manual annotation, which is costly and not scalable. To solve this problem, exploiting automated methods has great value, but is difficult since it requires preserving subtle differences while providing a clear delineation between the observed narrative and the generated ones.

Actually, the generation of contrastive narratives involves exploring the latent space surrounding a given narrative, enabling the creation of similar narratives with dis-

[1]For example, according to $X = (P, S)$ and $X_c = (P_c, S_c)$, we can exchange their prefixes and suffixes to obtain the negatives (P, S_c) and (P_c, S). We define this strategy as "crisscrossing", and use this definition in the rest of our paper.

tinct characteristics. Assuming that the evolution tendency of an observed narrative can be represented as a continuous trajectory in latent space, which can be modeled by Brownian motion [107, 133]. Consequently, we can sample the latent trajectories which exhibit proximity to the observed trajectory, and then decode the sampled trajectories into explicit narratives. But the problem is that the decoded narratives may differ significantly in content from the observed narrative, which may not meet the requirements for contrastive narratives. To simplify the problem, we further suggest that contrastive narratives keep the same endpoint as the observed narrative, which directly models the fact that a narrative event can evolve to the same end through different paths [98]. Based on this constraint, we are able to sample different trajectories from the Brownian Bridge [68, 133] region that is centered around the observed narrative. The sampled trajectories are decoded as narratives with the same start and end as the observed narrative, while also having similar but different intermediate event chains. Then we crisscross the observed narrative and the generated ones to synthesize negative samples. In fact, in our crisscrossing strategy, the start and end points of resulting negatives remain the same as the positive ones. That is, the start and end of positive narratives will never be perturbed. This further motivates us to design an event-level perturbation to obtain negatives, as more diverse negatives definitely benefit contrastive learning.

In this paper, we devise two strategies to create hard negatives for narrative coherence learning. The first strategy crisscrosses a narrative with its contrastive variants, and the second strategy performs an event-level replacement. To obtain contrastive narratives, we sample different latent trajectories from the Brownian Bridge region, then fix the start and end points of the narrative, and generate diverse contrastive narratives. The generated contrastive examples are used to create hard negatives.

Our contributions can be summarized as follows.

- Based on the Brownian Bridge process, we generate high-quality contrastive

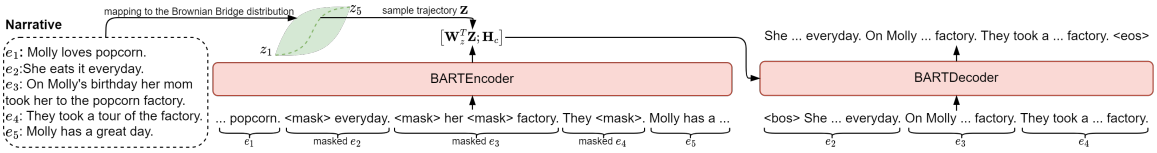


Figure 7.2: The training phrase of contrastive narratives generation. Given z_1 and z_5, $\mathbf{Z}$ is sampled according to Equation 7.1. The masked e_2, e_3, e_4 are used as the prompt for decoding.

narratives, which are used to synthesize hard negatives.

- We propose a new *coh*erence *eval*uator (*CohEval*), which is enhanced by diverse and high-quality hard negatives. Our model is trained exclusively via self-supervised contrastive learning and is applicable across a diverse spectrum of downstream tasks within the AI domain.

- We evaluate our model on multi-choice tasks and one narrative generation task. We additionally perform a comprehensive examination of our strategies for synthesizing negative samples. The empirical findings validate the efficacy of our approach.

7.2 Methods

7.2.1 Data Preparation

Following the previous method [7], we use RocStories [75] as data corpus, since it contains abundant event commonsense knowledge, making it a good resource for narrative reasoning. Due to the limitation of computational resources, we randomly select about 20k samples from RocStories, and denote them as the positive sample set $\mathcal{D}^+$. Each sample in $\mathcal{D}^+$ is a narrative $X = \{e_1, \cdots, e_5\}$, in which each e_i $(i = 1, \cdots, 5)$ is an event. Following previous works, we lay narrative coherence learning

in contrastive learning, in which the negative samples are needed for training.

We devise two strategies for mining hard negatives: (1) crisscrossing a narrative and its contrastive variants; (2) event-level replacement. Next, we introduce how to obtain contrastive narratives.

7.2.2 Generating Contrastive Narratives via the Brownian Bridge Process

Given a narrative, the contrastive variants should be similar to it and express distinctive characteristics. We regard this problem as exploring the latent space surrounding the given narrative, and propose to model this problem by the Brownian Bridge process [133]. The transition distribution of a Brownian Bridge process from a start point z_0 at $t = 0$ to an endpoint z_T at $t = T$ is:

$$p(z_t|z_0, z_T) \sim \mathcal{N}((1 - \frac{t}{T})z_0 + \frac{t}{T}z_T, \frac{t(T-t)}{T}). \tag{7.1}$$

It acts like a noisy linear interpolation between the start and end points of the trajectory, which can maintain a smooth transition of event evolution given the start and end points.

Following [133], we pre-train an encoder with the Brownian Bridge loss, so that we can encode an event e to the latent code z. The event encoder is a nonlinear mapping from raw input space to latent space, $f_\theta : \mathcal{X} \to \mathcal{Z}$. Consider a set of triplet observations, (x_1, x_2, x_3), the goal is to ensure that $f_\theta(x_1), f_\theta(x_2), f_\theta(x_3)$ follow the Brownian bridge transition density in Equation 7.1. Following [133], we ensure this using a contrastive objective. Formally, given a narrative event sequences, $S = \{e_0, \cdots, e_4\}$, we draw batches consisting of randomly sampled positive triplets e_0, e_t, e_T where $0 < t < T$: $\mathcal{B} = \{(e_0, e_t, e_T)\}$. Note that we use indices $0, t, T$ to denote the start, middle, and end points of a Brownian bridge, but these do not correspond to strictly sampling

the first, middle, and last events of a narrative story. The encoder is optimized by:

$$\begin{aligned} \mathcal{L}_f &= -\log \frac{\exp(d(e_0, e_t, e_T; f_\theta))}{\sum_{(e_0, e_{t'}, e_T) \in \mathcal{B}} \exp(d(e_0, e_{t'}, e_T; f_\theta))} \\ d(e_0, e_t, e_T; f_\theta) &= -\frac{1}{2\sigma^2} ||f_\theta(e_t) - \mu||_2^2, \end{aligned} \tag{7.2}$$

where μ and σ^2 are the mean and variance in Equation 7.1. As suggested by [133], we freeze the BART and add a non-linear layer that converts the BART output to a latent vector. The size of the latent space is set to 64 by default.

Then, by fixing (z_1, z_5), we sample z_t according to Equation 7.1 to obtain the latent trajectories $\mathbf{Z} = \{z_1, z_2, z_3, z_4, z_5\}$. To generate contrastive narratives, we encode (e_1, e_5) with BART [50] to obtain the context embeddings:

$$\mathbf{H}_c = \text{BARTEncoder}([e_1, e_5]), \tag{7.3}$$

where $[;]$ denotes the concatenation, $\mathbf{H}_c \in \mathcal{R}^{l \times d}$, l is the length of $[e_1; e_5]$. Next, given $\mathbf{H}_c$ and latent codes $\mathbf{Z}$, we generate middle events $y = (e_2, e_3, e_4)$. Specifically, let y_t denotes the t-th tokens in y. At the timestep t, the decoder must predict y_t using $\mathbf{H}_c$, all tokens in the past $y_{<t}$, as well as the event latent codes $\mathbf{Z}$:

$$\begin{aligned} \mathbf{h}_{y_t} &= \text{BARTDecoder}(y_{<t}, \mathbf{H}_c, \mathbf{W}_z^T \mathbf{Z}) \\ P(y_t | Y_{<t}) &= \text{softmax}_V(\mathbf{W}_v \mathbf{h}_{y_t} + b). \end{aligned} \tag{7.4}$$

where V denotes the standard vocabulary, $\mathbf{W}_z$ denotes a linear layer that maps the dimension of z to be identical to $\mathbf{H}_c$. This can be seen as decoding a latent trajectory $\{z_1, z_2, z_3, z_4, z_5\}$ into narrative events given the start event e_1 and end event e_5.

However, in our preliminary trials, we found that the generated narratives are coherent but less similar to the original one, which brings difficulties to the construction of hard negatives. The possible reason is that the encoding process, i.e., encoding e to z, lost too much information, making it difficult for the model to reconstruct y. To solve this problem, we randomly mask the y with the ratio of ρ (0.85 by default), and use the masked sequence as the prompt for the decoding phrase, which encourages

the decoder to generate more similar events to y. Actually, these can be seen as two types of constraints, where $\mathbf{Z}$ requires that y and the generated text show similar trajectories in latent space, and the masked prompt requires that y and the generated text are similar in vocabulary. The whole training process is shown in Figure 7.2.

When training, we use RocStories excluding $\mathcal{D}^+$ as training data. We have also tried other pre-trained models, such as GPT2 [102] and T5 [104], and BART empirically performs best, as shown in Table 7.8. Therefore, we choose BART as the backbone. After training, for each $X \in \mathcal{D}^+$, we fix its start and end events, then sample different intermediate events. For each X, we first generate 200 candidates, then use several criteria to filter low-quality candidates. Specifically, for each positive narrative, we generate 200 candidates. In practice, we observe that the generator may produce incoherent or duplicate candidates. Therefore, we set several rules to filter low-quality items. We first use our event-level replacement strategy to train the base evaluator M_{ER}. We use M_{ER} to filter items whose coherence scores are smaller than a threshold (empirically set to 0). Next, for each candidate, we calculate its text similarity with the remaining candidates. We gradually discard the candidates with the highest similarity until there are 100 remaining. When training Coheval, we select N top-ranked candidates according to their coherence scores for synthesizing negative samples. We finally retain N (60 by default) most-qualified contrastive examples.

7.2.3 Synthesizing Negative Examples

We devise two strategies to create negative examples. The first strategy crisscrosses a narrative with its contrastive variants, and the second strategy performs an event-level replacement.

Crisscrossing a Narrative and its Contrastive Variants

Note that each X contains five events. For simplicity, we define the first two events as the prefix (P), and the last three events as the suffix (S), so that we denote $X = (P, S)$ and the contrastive variant $X_c = (P_c, S_c)$. Then we are able to synthesize the negative example $X^- = (P, S_c)$. The basic intuition is: S_c is coherent with P_c, so it should be less coherent with P. This is because X and X_c are different paths with the same start and end points. Meanwhile, $X^- = (P, S_c)$ is similar to $X = (P, S)$, making it qualified as a hard negative[2]. With loss of generality, we denote the obtained negative samples as $\mathcal{C}_X = \{X_i^-\}_{i=1}^{2N}$.

For each training epoch, we randomly sample K (15 by default) negatives samples $\{X_k^-\}_{k=1}^{K}$ from $\mathcal{C}_X$ for each X, and feed them as well as X into a pre-trained language model (PLM) [15, 66], e.g. RoBERTa, to obtain sequence-level representations:

$$\begin{aligned} \mathbf{h}^+ &= \text{RoBERTa}(X), \\ \mathbf{h}_k^- &= \text{RoBERTa}(X_k^-), \end{aligned} \tag{7.5}$$

where $k = \{1, \cdots, K\}$, $\mathbf{h}^+$ and $\mathbf{h}_k^- \in \mathcal{R}^d$, d is the hidden size of RoBERTa. We have also tried BERT [15] as the backbone, as shown in Figure 7.3.

Next, the sequence-level representations are passed into a linear layer $\mathbf{W}_c \in \mathcal{R}^d$ to derive coherence scores of all samples:

$$\begin{aligned} s^+ &= \mathbf{W}_c^T \mathbf{h}^+, \\ s_k^- &= \mathbf{W}_c^T \mathbf{h}_k^-. \end{aligned} \tag{7.6}$$

Lastly, we use the contrastive classifying objective to distinguish the positive examples from the corresponding negative examples:

$$\mathcal{L}_1 = -\frac{1}{|\mathcal{D}^+|} \sum_{\mathcal{D}^+} \log \frac{\exp(s^+)}{\exp(s^+) + \sum_{k=1}^{K} \exp(s_k^-)}. \tag{7.7}$$

[2]Similarly, we can obtain the negative example $X^- = (P_c, S)$ by defining the first three events as the prefix.

It should be noted that the difference between $X^- = (P, S_c)$ and $X = (P, S)$ lies in the third and fourth events, i.e., e_3 and e_4. Due to the masked prompt, some tokens in (e_3, e_4) of X^- are similar to those of X, making X^- qualified. However, in the crisscrossing strategy, e_1 and e_5 will never be perturbed. This further motivates us to perform an event-level perturbation to X to create more diverse negative samples.

Event-level Replacement

Due to the fact that events are the basic semantic unit of neural language, for a narrative, if we replace a component event with another similar but different event, the resulting example should be less coherent and similar to the original narrative.

Specifically, based on $\mathcal{D}^+$, we build an event pool, which consists of about 100k different events. We pre-compute the cosine similarity among all event pairs using SimCSE [21], and cache the top 20 most similar events Q^e for each query event e. Then, given a positive example X, we randomly select a position i and replace i-th event e_i with a randomly sampled event $\bar{e}$ from Q^e to create a negative example $\bar{X} = \{\cdots, e_{i-1}, \bar{e}, e_{i+1}, \cdots\}$. Likewise, for each training epoch, we create K negatives samples $\{\bar{X}_k\}_{k=1}^K$. After obtaining hidden states of negatives: $\bar{\mathbf{h}}_k = \text{RoBERTa}(\bar{X}_k)$, we derive coherence scores of all samples and use the contrastive loss to rank the positive sample above the negatives:

$$\begin{aligned} s^+ &= \mathbf{W}_c^T \mathbf{h}^+, \bar{s}_k = \mathbf{W}_c^T \bar{\mathbf{h}}_k, \\ \mathcal{L}_2 &= -\frac{1}{|\mathcal{D}^+|} \sum_{\mathcal{D}^+} \log \frac{\exp(s^+)}{\exp(s^+) + \sum_{k=1}^K \exp(\bar{s}_k)}. \end{aligned} \tag{7.8}$$

7.2.4 Training and Knowledge Transferring

When training, the final loss is

$$\mathcal{L} = \gamma \mathcal{L}_1 + (1 - \gamma)\mathcal{L}_2, \tag{7.9}$$

where γ is set to 0.5. It should be noted that another way is to merge two types of negatives and directly perform contrastive learning. However, this requires more GPU memory, which exceeds our condition. Therefore, we calculate the two losses separately and then average them.

Our *CohEval* can be easily transferred to many downstream applications. For example, for the multi-choice task with a input C and option candidates $O = \{o_1, \cdots, o_n\}$, we can use *CohEval* to select most reasonable o by:

$$o \leftarrow \arg\max_i \mathit{CohEval}([C, o_i]). \tag{7.10}$$

Motivated by existing plug-and-play text generation methods [71, 11], we also evaluate our *CohEval* in narrative text generation, with *CohEval* as coherence guidance. Details can be seen in the experiment.

7.3 Experiment

	COPA	e-Care	αNLI	Cloze	Swag	HS.	TimeT.
#numAns	2	2	2	2	4	4	N/A
#numVal	500	2132	1532	1871	20006	10041	1871
#numTest	500	N/A	3059	1871	N/A	N/A	1871

Table 7.1: The statistics of the used datasets. #numVal and #numTest denotes the number of samples in the val and test set. #numAns denotes the size of the answer set of multi-choice datasets. HS. and TimeT. denotes HellaSwag and TimeTravel, respectively.

7.3.1 Datasets

The evaluation datasets include COPA [108], e-Care [17], αNLI [5], Cloze [75], Swag [146], HellaSwag [147], and TimeTravel [98]. TimeTravel is a text-generation dataset, while others are multi-choice datasets. We evaluate our model on these datasets in the zero-shot setting. Note that the test sets of e-Care and HellaSwag are not released. So we evaluate our model on the validation set of the three datasets. The statistics of the datasets, as well as the experimental details are shown in Table 7.1.

7.3.2 Experimental Settings

For training the contrastive narratives generator, we use BART-base as the backbone. Batch-size is set to 16. We use the AdamW optimizer. lr is set to 5e-5. Weight-decay is set to 1e-4. We train the generator with 10 epochs and linearly decrease the lr to zero with no warmup. When the generation phase, we kept the $N = 60$ most qualified contrastive narratives for creating negative examples. For training our *CohEval*, we adopt RoBERTa-large as the backbone. We train our model for 5 epochs, and then evaluate it on downstream tasks. We set batch-size to 1 and gradient-accumulation-steps to 16. For each positive example, we sample 15 negative examples for contrastive training. lr is set to 5e-5. Weight-decay is set to 1e-4. We use the AdamW optimizer and linearly decrease the lr to zero with a 10% warmup ratio. The random seed is set to 42 for all experiments. All experiments are performed on a Ubuntu server with 4×RTX2080Ti GPUs.

7.3.3 Baselines and Metrics

For multi-choice tasks, the metric is Accuracy. We compare our method with Event-BERT [158], RankGen [47] and several large language models (LLMs), including

Alpaca-lora (7B)[3], ChatGLM2 (6B) [18, 148] and ChatGPT [84]. For LLMs, we use one-shot prompting for experiments, the used prompts are the same as in Chapter 6, Table 6.2 and 6.3.

For TimeTravel, we follow [11] and formulate this task in the MCMC-based sampling paradigm. Prior work in EDUCAT [71, 11] utilizes the MCMC-based sampling method for this endeavor. EDUCAT employs direct sampling from the sentence space employing three local operations: token replacement, deletion, and insertion. During the sampling process, upon identifying an edit position, the operation is randomly selected with uniform probability. Ultimately, the suggested sentence will undergo acceptance or rejection based on the computed acceptance rate determined by desired attributes $\pi(y)$. This iterative procedure continues until convergence is achieved. The stationary distribution $\pi(y)$ within EDUCAT is delineated as the product of the fluency score and coherence score, represented as follows:

$$\pi(y) = \mathcal{X}_{LM}(y) \cdot \mathcal{X}_{Coh}(y), \tag{7.11}$$

where the fluency score $\mathcal{X}_{LM}(y)$ is the probability of the generated ending based on GPT2. The coherence score $\mathcal{X}_{Coh}(y)$ is defined by:

$$\mathcal{X}_{Coh}(y') = \frac{P_{Coh}(Y = y'|z, x')}{P_{Coh}(Y = y'|z, x)}, \tag{7.12}$$

where $P_{Coh}(\cdot)$ is the conditional probability calculated by GPT2. This definition encourages the generated y' to be more coherent to x' instead of x. Following EDUCAT, we define the stationary distribution $\pi(y)$ as Equation 7.11. The difference is that we replace $\mathcal{X}_{Coh}(y)$ with our CohEval:

$$\mathcal{X}_{Coh}(y) = \text{CohEval}([z; x; y']), \tag{7.13}$$

where $[;]$ denotes the concatenation. Same as EDUCAT, we run our model and its variants for 100 steps for fairness.

[3]The checkpoint is at https://github.com/tloen/alpaca-lora.

We compare our method with DELOREAN [97], ClarET [159], CGMH [71], EDUCAT [11]. Automatic evaluation metrics include BLEU4 [87], BertScore [153], ENTScore [11], and HMean= $\frac{2 \cdot \text{BLEU4} \cdot \text{ENTScore}}{\text{BLEU4} \cdot \text{ENTScore}}$ [11]. Manual evaluation metrics include Fluency, Min-Edits [11], and Coherence.

7.3.4 Overall Results

Methods	COPA	e-Care	αNLI	Cloze	Swag	HS.
LLMs-based Prompting						
Alpaca-lora (7B)	57.4	54.5	52.6	66.1	36.0	30.2
ChatGLM2 (6B)	78.1	66.9	58.1	84.3	48.7	41.2
ChatGPT	96.2	81.8	75.5	94.7	70.7	76.4
Contrastive Training Based Methods						
RankGen(base)	63.8	70.3	52.2	50.7	46.3	33.9
RankGen(large)	70.2	72.1	54.8	54.4	49.2	40.5
EventBERT	N/A	N/A	59.5	75.6	N/A	N/A
CohEval (ours)	**77.8**	71.9	**67.6**	**77.6**	**67.4**	**44.9**
Ablation Study						
M_{ER}	73.4	**75.4**	65.3	77.1	61.8	38.9
M_{CC}	75.8	68.2	67.2	69.4	66.9	44.7

Table 7.2: The accurary (%) on multi-choice datasets. HS. denotes HellaSwag. Scores with **bold** denote the best results among contrastive training based methods.

Automatic Evaluation The automatic evaluation result can be seen in Table 7.2 and 7.3, respectively. We have the following observations.

- In Table 7.2, our model surpasses all contrastive training-based methods. This indicates that the negative samples we create are more qualified, which verifies the effectiveness of our method.
- Although there is still a significant gap compared to ChatGPT, our method surpasses smaller LLMs, e.g., ChatGLM2, on most datasets.
- In Table 7.3, our method outperforms EDUCAT. Since EDUCAT uses the off-the-shelf PLMs for evaluating coherence, the performance improvement proves that our CohEval is better at evaluating narrative coherence.
- Compared with our method, ChatGLM2 and ChatGPT achieve high ENTScore, but low BLEU4. This indicates that auto-regressive methods tend to generate coherence counterfactual ending with massive edits. These behaviors conflict with the requirements of the task.

Ablation Study To investigate the influence of the two kinds of negatives, we devise two ablated variants: (1) M_{ER} which means we create negatives via **e**vent-level **r**eplacement; (2) M_{CC} which means we create negatives via the **c**riss**c**rossing strategy. The ablation study result is shown in Table 7.2, 7.3. We have the following observations.

- Compared to CohEval, M_{ER} and M_{CC} achieve lower ENTScore, indicating their weaker coherence evaluation abilities. But both variants obtain higher BLEU4 and BertScore. In TimeTravel, there is a trade-off phenomenon between BLEU and EntScore. This is because the gold y' is obtained through editing the original y with minimal-edits. This leads to a high word overlap between y' and y. Due to the weaker coherence evaluation abilities of the two variants, the probability of accepting transitions is lower when adopting MCMC for rewriting. In other words, when using M_{ER} and M_{CC}, the number of rewritings is relatively low, resulting in higher BLEU4 and BertScore but lower ENTScore.

Methods	BLEU4	BertS.	ENTS.	HMean
LLMs-based Prompting				
ChatGLM2 (6B)	16.47	60.03	66.15	26.37
ChatGPT	36.41	69.81	82.62	50.55
Off-the-shelf small PLMs				
DELOREAN	23.89	59.88	**51.40**	32.62
ClarET	23.75	63.93	N/A	N/A
CGMH†	41.09	73.90	28.06	33.34
EDUCAT	44.05	74.06	32.28	37.26
EDUCAT†	43.57	74.00	33.41	37.82
CohEval (ours)	42.46	73.36	37.39	**39.77**
Ablation Study				
M_{ER}	**44.18**	**74.34**	34.63	38.82
M_{CC}	42.99	73.64	35.78	39.05

Table 7.3: The automatic result on TimeTravel. † denotes our implementation. BertS. denotes BertScore. ENTS. denotes ENTScore. Scores with **bold** denote the best results among off-the-shelf small PLMs.

- The best ENTScore is achieved by combining two kinds of hard negatives. This indicates the two kinds of negatives complement each other. The reason is that more diverse negative examples contribute to contrastive learning.

- M_{CC} generally performs better than M_{ER}. The possible reason is that, compared to the crisscrossing strategy, the event-level perturbation is more coarse-grained. Nevertheless, event-level replacement is an effective supplement to the crisscrossing strategies.

Manual Evaluation on TimeTravel We perform an A/B test to compare our method with several baselines. Following [98, 11], the human evaluation mainly focuses on three primary criteria: i) Fluency, whether a model produces fluent text; ii) Coherence, the logical consistency between the counterfactual context (z, x') and the generated endings y; and iii) Min-Edits, the extent of minimal revision between two endings. We carry out a pairwise comparison with CGMH, EDUCAT, and two ablated models: M_{exp} and M_{imp}. We randomly sample 100 cases for each pair of models. Three annotators are recruited to make a preference among win, tie, and lose given the counterfactual context and two outputs by our model and a baseline respectively. The annotators are research students from the field of text generation to make sure they have a fair judgment of used metrics. We calculate Fleiss's kappa reliability as the inter-annotator agreement.

As is shown in Table 7.4, LLMs are able to generate fluent and coherent counterfactual ending, but tend to massively edit the original ending, which coincides with the finding in automatic evaluation. Compared to EDUCAT and two ablated variants, CohEval achieves better fluency and coherence results. In addition, these four models achieve similar Min-Edits results, this is because they run for the same editing steps. The Fleiss's kappa reliability of Fluency, Min-Edits, and Coherence is 0.488, 0.507, and 0.428, respectively.

Human Correlation with our CohEval Same as [11], we analyze the correlation between our CohEval and human ratings in terms of coherence evaluation. We calculate three coefficients, including Pearson's r and Kendall's τ. The result is shown in Table 7.5. All results show a positive correlation. The result of our CohEval is close to that of ENTScore. Notice that ENTScore is trained with human-labeled counterfactual data, while our CohEval is trained in a self-supervised manner. This demonstrates the applicability of our CohEval.

Overall, the result demonstrates that our CohEval is a generic narrative coherence

Methods	Fluency		Min-Edits		Coherence	
	W(%)	L(%)	W(%)	L(%)	W(%)	L(%)
vs. EDUCAT†	27.0	13.7	23.0	24.7	33.7	4.7
vs. M_{ER}	25.7	16.7	22.3	23.3	28.0	6.7
vs. M_{CC}	20.0	12.0	23.7	22.3	23.0	7.0
vs. ChatGLM2	13.3	45.3	84.7	7.7	19.0	37.0
vs. ChatGPT	14.7	41.3	60.3	25.0	13.7	40.0

Table 7.4: Manual evaluation result on TimeTravel. Scores indicate the percentage of Win(W) and Lose(L).

Metrics	Pearson's r	Kendall's τ
ENTScore	0.25	0.24
CohEval	0.20	0.18

Table 7.5: The correlation between automatic metrics, e.g., ENTScore and CohEval, and human ratings. All of these numbers are statistically significant at $p < 0.01$.

evaluator, and can be applied to a wide range of downstream tasks.

7.3.5 Deeper Analysis about Contrastive Narratives Generation

Indirect Evaluation through Multi-choice Tasks We conduct an ablation experiment to explore the impact of different sub-modules in contrastive narratives generation. We compare our Brownian-Bridge based method (denoted as "BB") with the following variants. (1) "w/o prompt", in which we ablate the masked prompt when training. (2) "w/o trajectory", in which we ablate the latent trajectories sampled from the Brownian bridge. (3) "Infilling", in which we ablate the masked prompt

and the sampled latent trajectory when training. In this case, the ablated variant degenerates into a text-infilling model. We use the counterparts generated by different variants for crisscrossing to obtain negative examples, which are then used for contrastive learning. The result is shown in Table 7.6. We find: (1) Compared to "BB", "w/o prompt" and "w/o trajectory" get result drops, respectively; (2) "Infilling" gets a further performance drop.

The possible reasons lie in the following aspects.

- If contrastive narratives are incoherent, then the synthesized negatives are not "hard". The sampled latent trajectories help to maintain the coherence of generated contrastive narratives, which benefits the quality of synthesized negatives.
- The masked prompt helps to reduce the difficulty of the generation process, as a result, the obtained contrastive counterparts are similar to the original ones, making the resulting negatives more qualified.

Methods	COPA	e-Care	αNLI	Cloze	Swag	HS.	∇
BB (our M_{CC})	75.8	68.2	67.2	69.4	66.9	44.7	—
w/o prompt	79.0	65.4	68.5	75.6	59.2	39.9	-4.6
w/o trajectory	71.0	71.2	65.9	69.9	67.1	42.0	-5.1
Infilling	72.2	71.9	64.8	77.7	58.1	40.4	-7.1

Table 7.6: The result (%) of different kinds of counterparts for synthesizing negative examples.

Direct Evaluation through Manual Judgement We further conduct a manual evaluation to directly evaluate the quality of generated contrastive narratives. Since we want the generated narrative to be *similar* to the original one and reflect *subtle differences* (such as changes in opinions or entities) to make itself a different story, we

Methods	Coherence		Similarity		SubtleDiff.	
	W(%)	L(%)	W(%)	L(%)	W(%)	L(%)
vs. w/o prompt	43.0	19.0	46.0	6.3	27.3	7.0
vs. w/o trajectory	53.7	15.3	26.7	7.7	28.0	12.7
vs. Infilling	60.3	10.3	56.3	5.7	49.0	6.7
vs. ChatGLM2	40.0	20.0	39.0	20.3	24.3	28.3
vs. ChatGPT	21.0	26.0	30.7	17.0	18.0	23.0

Table 7.7: The manual evaluation on contrastive narratives generation. We compare "BB" with "w/o prompt", "w/o trajectory", "Infilling", ChatGLM2, and ChatGPT.

use Coherence, as well as Similarity and SubtleDifference (SubtleDiff.) as metrics. We randomly select 100 stories, which have no overlap with train data for experiment. For each story, we use different models to generate its contrastive variant. We also perform a pairwise comparison with "w/o prompt", "Infilling", and two LLMs: ChatGLM2 and ChatGPT. The same three annotators are asked to make a preference among win, tie, and lose for each pair of generation. We use Coherence, Similarity, and SubtleDiff. as metrics. Coherence reflect the logical consistency between the given (start,end) events and the generated middle events. Similarity reflects the similarity between the generated middle events and those of the original story. SubtleDiff. measures whether the generated example is a qualified contrastive narrative, which reflects subtle difference to the original story but actually a different story.

In Table 7.7, ChatGPT generally exhibits the best result, which reflects its powerful reasoning ability. Our "BB" is slightly inferior to ChatGLM2 on *SubtleDiff.* , but wins on the other two metrics. This indicates that our method is comparable to small LLMs. In addition, "BB" significantly surpasses the ablated variants. Specifically, we find that the masked prompt helps to improve Similarity, while latent trajectory helps to improve Coherence. This coincides with human intuition. The Fleiss's kappa

reliability of Coherence, Similarity, and SubtleDiff. is 0.369, 0.371, 0.244, respectively.

Generally, by utilizing the Brownian bridge process, we harvest qualified contrastive narratives, which contributes to contrastive learning.

7.3.6 Further Discussion

PLMs	Fluency (↓)	ENTScore (↑)
GPT2	2.8	58.2
T5	3.3	52.2
BART	3.4	66.7

Table 7.8: Impact of different backbones for contrastive narratives generation.

Influence of Different Backbones for Contrastive Narratives Generation We conduct a preliminary study on the influence of different backbones, including GPT2 [102] and T5 [104], and BART [50], for generating contrastive narratives. We use Fluency and ENTScore as metrics. Fluency evaluates whether the generated text is a fluent text sequence. We use off-the-shelf GPT2 to calculate Fluency. ENTScore evaluates the coherence of the generated stories. We randomly sample 2000 examples that do not exist in training for evaluation. We calculate the average result. As shown in Table 7.8, GPT2 is good at generating more fluent text, and BART generates more coherent text. A possible reason is that the contrastive narrative generation is more compatible with BART's pre-training task, e.g., masked auto-encoding. Finally, we choose BART as the backbone.

Different Backbones for Narrative Coherence Learning We additionally build our method on the BERT-base [15] and RoBERT-base backbones, as shown in Figure 7.3. RoBERTa-base has a better performance than BERT-base, and the RoBERTa-

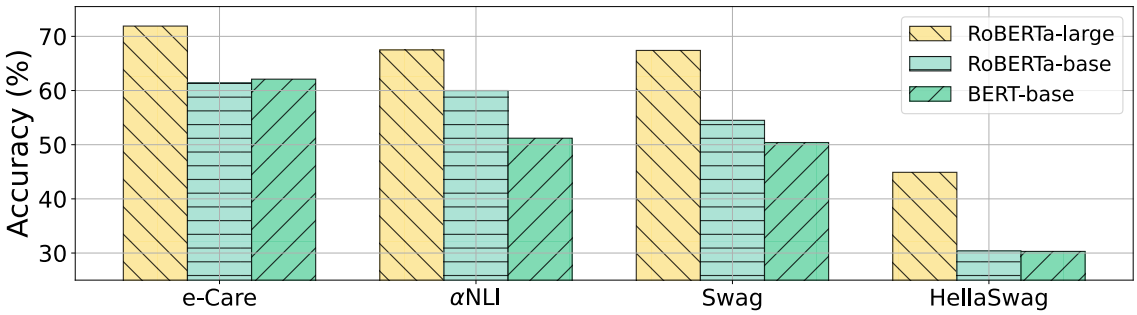


Figure 7.3: Results under different backbones for narrative coherence learning.

large tends to have a better result than RoBERTa-base. However, due to the limitation of computing resources, we are not able to evaluate our method under larger pre-trained models.

Strategies		COPA	e-Care	αNLI	Cloze	Swag	HS.
Mixup	Random	60.2	49.7	52.1	59.1	32.7	28.8
	w/o prompt	61.8	55.5	57.0	64.3	35.4	32.1
	BB	63.6	60.0	64.4	66.5	41.9	29.3
CrissC.	Random	72.6	71.8	58.8	70.0	53.6	37.4
	w/o prompt	79.0	65.4	68.5	75.6	59.2	39.9
	BB (our M_{CC})	75.8	68.2	67.2	69.4	66.9	44.7

Table 7.9: The result of different strategies for creating negatives. CrissC. denotes the crisscrossing strategy.

Influence of Different Strategies for Creating Negatives In our method, we crisscross a positive narrative with its contrastive counterparts to create negatives. Here, we further investigate the result when using Mixup [149] to create negatives. The mixup strategy creates negative examples via mixing-up a positive X and several

counterparts $\{X_c^k\}_{k=1}^K$ in the latent space:

$$\begin{aligned}
\mathbf{h}^+ &= \text{RoBERTa}(X) \\
\mathbf{h}_c^k &= \text{RoBERTa}(X_c^k), \\
\bar{\mathbf{h}}^k &= \alpha_k \mathbf{h}^+ + (1-\alpha_k)\mathbf{h}_c^k, \\
\alpha_k &\sim Uniform[0,1].
\end{aligned} \tag{7.14}$$

Then, the loss is:

$$\begin{aligned}
s^+ &= \mathbf{W}_c^T \mathbf{h}^+, \\
\bar{s}^k &= \mathbf{W}_c^T \bar{\mathbf{h}}^k, \\
\mathcal{L}_M &= -\frac{1}{|\mathcal{D}^+|} \sum_{\mathcal{D}^+} \log \frac{\exp(s^+)}{\exp(s^+) + \sum_{k=1}^K \exp(\bar{s}^k)}.
\end{aligned} \tag{7.15}$$

The experiment setting details are the same as those used in Section 7.3.2.

We additionally explore three ways of obtaining the counterparts: (1) "BB" denotes our Brownian-Bridge based contrastive narratives; (2) "w/o prompt" denotes we ablate the prompt when generating contrastive narratives; (3) Random denotes we randomly select different positive narratives as counterparts. The result is shown in Table 7.9. We observe that:

- The crisscrossing strategy is superior then Mixup by a large margin. We speculate that in the era of self-attention [130], using the transformer to directly learn the representation of negative samples is better than manipulating representations of samples in the hidden space.
- Whether adopting "CrissC." or Mixup, our BB-based contrastive narratives far surpass "random", which proves the strength of our method.

Results under Different Number of Retained Contrastive Narratives We explore the influence of the number of retained contrastive narratives. The result is shown in Figure 7.4. Our method generally achieves the best result when $N = 60$,

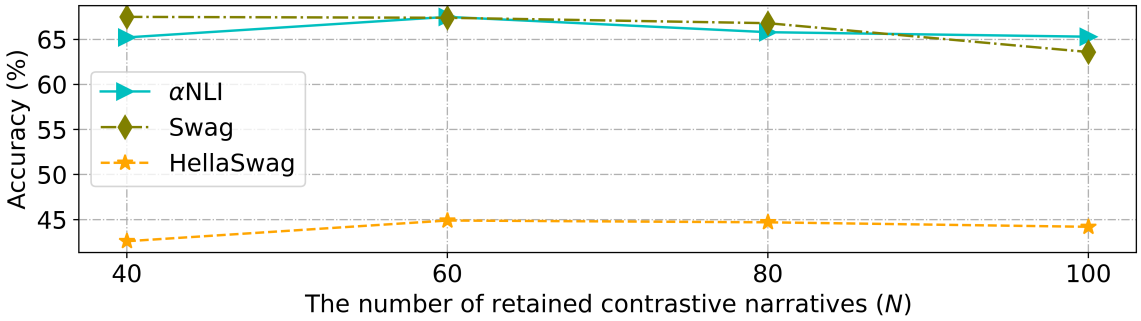


Figure 7.4: Results under the different number of retrained contrastive narratives.

and the result even decreases when N further increases. We speculate that as N increases, incoherent contrastive examples increase, which has a negative impact on the quality of synthesized negative examples. So, we set $N = 60$ by default.

Impact of the Mask Ratio ρ We investigate the impact of the different mask ratio ρ when generating contrastive narratives. In Table 7.10, the result is best when $\rho = 0.85$. As ρ decreases, the result gets worse. To investigate the reason, we manually examine the generated examples, and find the model tends to paraphrase the original story and generate duplicate examples when ρ decreases. This is because more information about the original story will be exposed when using a lower mask rate, making it easier to reconstruct the original story. We additionally calculate the diversity of the contrastive narratives generated at different ρ. We use Distinct-n [54] as the metric. As shown in Table 7.10, as ρ decreases, the corresponding Distinct scores also decrease. This indicates that a lower mask rate ρ may lead to duplicate samples when the generation phase, which harms the diversity of synthesized negative samples. Therefore, we proactively filter out duplicate items.

The Reliability of Created Negative Examples We further analyze whether the created negative samples are indeed "negative". On the training set, we first use ENTScore to directly evaluate the coherence of positive samples and two types of neg-

ρ	Accuracy(%)			Dist-2	Dist-3
	αNLI	Swag	HS.		
$\rho = 0.90$	65.2	**67.5**	42.6	26.4	41.0
$\rho = 0.85$	**67.5**	67.4	**44.9**	**27.1**	42.6
$\rho = 0.80$	66.2	66.5	43.3	26.8	**42.9**
$\rho = 0.70$	64.0	63.4	42.9	25.0	40.7

Table 7.10: The result under the different ρ. Dist-n denotes Distinct-n. Scores with **bold** denote the best result.

Types	ENTScore	FN Rate
Positive examples	94.6	N/A
Negatives via replacement	54.5	3.0%
Negatives via crisscrossing	65.9	4.3%

Table 7.11: The reliability evaluation of created negatives. FN denotes *false negative*.

atives. As shown in Table 7.11, the real positive examples receive an especially high ENTScore. However, the synthesized two types of negatives receive lower ENTScore, proving that they are obviously less coherent than positive examples. Next, we sample 100 cases and ask the annotators to make a judgment about whether the created 'negatives' are actually more coherent than positives, making them false negatives. As shown in Table 7.11, both types of negatives show a low FN rate.

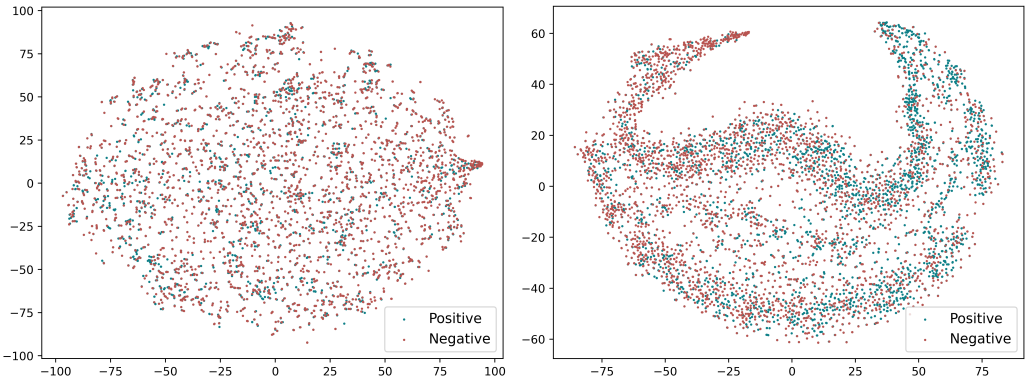
Error Analysis The most common error in event-level replacement is that the sampled event $\bar{e}$ from Q^e is especially similar to the original e, or is the paraphrase of the original e, as shown in Table 7.12, Case #1. The most common mistake in cross strategy is that the contrastive variant and the original story describe different actions for the same purpose, resulting in the false negative. An example is shown in

Table 7.12, Case #2. Overall, the proportion of errors is relatively low.

Visualize the Representations of Examples using t-SNE It is interesting to qualitatively visualize our model's ability to distinguish hard negatives. Based on the test set of TimeTravel, we are able to obtain positive examples and corresponding *hard negatives*. In TimeTravel, each example consists of an original story (z, x, y) and a counterfactual story (z, x', y'), where y' is similar to y. Motivated by [11], we obtain positive and negative samples from the perspective of natural language inference, i.e., the original context (z, x) entails by y but contradicts with y', and the counterfactual context (z, x') entails by y' but contradicts with y. Because y is similar to y', (z, x, y') and (z, x', y) tend to be hard negatives. Based on the test set of TimeTravel, we obtain 3742 positive examples and 3742 negative examples. Then, we use t-SNE to visualize representations of the examples that are encoded by different models.

We use our CohEval and the ablated variant M_{ER}, respectively, to obtain the representations of the examples, then we use t-SNE [129] to visualize the representations. As shown in Figure 7.5 (a), the representations of positive and negative examples obtained by M_{ER} entangle together, this shows that M_{ER}, a model that significantly outperforms baselines, still suffers from distinguishing the created positive and negative examples. But in Figure 7.5 (b), positive samples are concentrated on the right, while negative samples are concentrated on the left. This proves our CohEval's ability to distinguish positive examples from hard negatives, and confirms the effectiveness of the generated contrastive narratives.

Case Study Table 7.13 presents a case study for the task of TimeTravel. The counterfactual endings generated by ChatGLM2 and ChatGPT are very different from the original ending, which conflicts with the minimal-edits requirement of the task. On the contrary, based on the MCMC-sampling, our method produces the counterfactual ending, which is similar to the original ending, as well as coherent to



(a) M_{ER} (b) CohEval

Figure 7.5: Visualization of the representations of examples obtained from different models.

the counterfactual condition.

Table 7.14 presents a case study for the task of contrastive narratives generation. Due to the sampled different trajectories, in the case #1, our method shifts the topic of accent to personality, and produces a coherent story. In the case #2, our method exchanges the opinions of two participants. Due to the limited changes, the generated story is very similar to the original story and meets the requirements for contrastive narratives. On the contrary, the middle events generated by ChatGLM2 and ChatGPT show a significant difference from that of the original story, as a result, the synthesized negative samples should be easily distinguished.

7.4 Discussion

To automatically generate contrastive narratives, we made the following assumption: the observed story and its contrastive variants have the same start and end events. However, this assumption may not be consistent with reality. In addition, under limited computing resources, we are unable to explore our method on larger data

scales and larger pre-trained models. The experiment shows that our method is not able to surpass ChatGPT. But this does not mean that our work has no value in the era of large language models.

Our method is essentially a discriminative model, while LLMs are generative models. They have different advantages. For example, LLM is better at generating coherent text, and our CohEval is better at multi-choice tasks. In fact, on TimeTravel, we use MCMC to make our CohEval applicable to generating tasks. Therefore, the gap between our method and LLM has been magnified. On discriminative tasks, although our model is not as good as ChatGPT, it outperforms the smaller ChatGLM on most multi-choice tasks. On the other hand, it is inherently unfair to directly compare small models with LLMs, as large models are obtained with massive resources, e.g., data, hardware, funding, etc. Due to resource limitations, our method is not as good as ChatGPT, but it is superior to ChatGLM, which also indicates that our method is valuable in low-resource scenarios. Meanwhile, as data and methods can be shared, with sufficient computational resources, we can use a larger backbone and more data for training, which is expected to yield better results. We leave this in future works.

7.5 Chapter Summary

In this paper, we propose to use the Brownian Bridge process to generate contrastive narratives, then we crisscross a positive story and its contrastive variants to create negative examples for contrastive learning. In addition, we devise the event-level replacement, which is the effective supplement to the crisscrossing strategy. The experiment verifies that (1) the generated contrastive narratives are qualified, and (2) our CohEval is effective and is a general coherence evaluator that is applicable to many downstream tasks.

	Event-level replacement
#1	**Original story**: Rod was ironing his clothes. His iron is very old. The iron left a burn mark on his favorite shirt! Rod got very angry. He bought a new iron.
	Created negative: She even ironed her sheets! His iron is very old. The iron left a burn mark on his favorite shirt! Rod got very angry. He bought a new iron.
	Crisscrossing
#2	**Original story**: Katie needed to cross the street by herself. Being seven, she hadn't done it by herself before. She looked left then right. The coast was clear. She crossed the street successfully and was full of pride.
	Contrastive example: Katie needed to cross the street by herself. Being seven she knew it was possible. The nearest crosswalks were just behind her. She held her breath and did her best. She crossed the street successfully and was full of pride.
	Created negative: Katie needed to cross the street by herself. Being seven, she hadn't done it by herself before. The nearest crosswalks were just behind her. She held her breath and did her best. She crossed the street successfully and was full of pride.

Table 7.12: Error cases when creating negatives.

#1	Original context	Tim was mad. He saw roaches in his house.
	Original ending	He bought some cheap bug spray. However, it did not get rid of them. Tim called an exterminator.
	Counterfactual context	Tim was mad. He saw rats in his house.
	GhatGLM2	He found a rat trap and was able to catch them. Tim was relieved that the problem had been solved. He called an exterminator to make sure they were gone for good.
	ChatGPT	He bought some cheap rat traps. However, they did not catch any rats. Tim called a professional pest control service.
	CohEval(ours)	He bought rat poison. And it did not get rid of the rats. So he hired an exterminator.
#2	Original context	Jim worked at the local coffee shop. He was the best barista in town.
	Original ending	He always made sure his customers had the best drinks. Jim's boss saw him working one day and was impressed. Jim got a promotion for doing such a good job.
	Counterfactual context	Jim worked at the local coffee shop. He was the worst barista in town.
	GhatGLM2	He found a recipe online to make better drinks. Jim decided to try it out and impressed his boss. Jim got a raise for his excellent work.
	ChatGPT	He always made sure his customers had the worst drinks. Jim's boss saw him working one day and was disappointed. Jim got fired for doing such a bad job.
	CohEval(ours)	He always made sure his customers had the wrong drinks. Jim's boss saw him that night and was furious. He got a beating for doing a bad job.

Table 7.13: Some cases with the generated text by different models. The cases are from the test set of TimeTravel.

#1	Original story	Fanny is new at school. She is from England. Many kids do not understand her accent. This made Fanny sad. Fanny is having a difficult time making friends.
	ChatGLM2	Fanny is new at school. One of the kids at school does understand her accent, and they become her friends. Fanny meets a new student at her lunch table who helps her feel more comfortable. Fanny learns to embrace her accent and makes many new friends at school. Fanny is having a difficult time making friends.
	ChatGPT	Fanny is new at school. During lunchtime, Fanny sat alone at a table in the cafeteria. Some kids approached Fanny and asked her where she was from. Fanny tried her best to explain her accent, but they still had trouble understanding her. Fanny is having a difficult time making friends.
	CohEval(ours)	Fanny is new at school. She has just graduated from college. She is very shy. Fanny is not very outgoing. Fanny is having a difficult time making friends.
#2	Original story	Morgan and Timothy were having an argument. Morgan thought that all of The Beatles' music was good. Timothy thought their early stuff was not very good. Morgan made her case by simply playing a few songs. Timothy realized Morgan was probably right about the music.
	ChatGLM2	Morgan and Timothy were having an argument. Timothy tried to reason with Morgan, but she was too set in her opinion. Morgan refused to back down and continued to play her favorite songs. Timothy was left feeling frustrated, but eventually came to accept Morgan's love for her music. Timothy realized Morgan was probably right about the music.
	ChatGPT	Morgan and Timothy were having an argument. Morgan defended her stance by pointing out the innovative songwriting and harmonies in their later albums. Timothy reluctantly agreed to listen to a few songs from their later discography. As the music played, Timothy's opinion started to shift, realizing the depth and creativity in The Beatles' later work. Timothy realized Morgan was probably right about the music.
	CohEval(ours)	Morgan and Timothy were having an argument. Morgan said the Beatles' music was terrible. Timothy said their early music was good. Morgan disagreed. Timothy realized Morgan was probably right about the music.

Table 7.14: Case study for contrastive narratives generation.

Chapter 8

Conclusion and Future Work

Narrative, as a cognitive construct, occupies a central position in human cognition, serving as a fundamental mechanism through which individuals make sense of their experiences, construct identities, and communicate meaning. Rooted in sociocultural frameworks and cognitive schemas, narrative structures imbue mundane events with significance, shape perceptions of reality, and influence decision-making processes. The advent of neural narrative reasoning systems represents a paradigm shift in computational cognition, harnessing the power of artificial intelligence to emulate and augment the intricacies of human narrative processing. By leveraging sophisticated algorithms and deep learning architectures, these systems facilitate a seamless integration of narrative comprehension, generation, and reasoning into the fabric of everyday life. Through the lens of behavioral decision-making, neural narrative reasoning systems offer a multifaceted toolkit for individuals navigating the complexities of choice and action in diverse contexts. From the creation of compelling narrative novels to the elucidation of intricate phenomena, and even the facilitation of colloquial exchanges, these systems afford users a versatile means of accessing, interpreting, and co-creating narratives that resonate with their cognitive and affective landscapes.

However, the proliferation of potential application scenarios also engenders a con-

comitant escalation in the performance expectations placed upon narrative reasoning systems. As users increasingly rely on these systems to inform, entertain, and engage, the imperative for accuracy, coherence, and adaptability becomes paramount. Whether tasked with crafting immersive narratives, elucidating complex concepts, or simulating naturalistic conversation, these systems must demonstrate robustness, fluency, and semantic fidelity to meet the exigencies of real-world usage. In essence, the efficacy of narrative reasoning systems hinges upon their ability to navigate the intricate interplay between computational processes and human cognition. As such, ongoing research endeavors must continue to interrogate and refine the underlying mechanisms driving narrative generation, comprehension, and reasoning, thereby ensuring that these systems remain at the vanguard of computational intelligence and human-computer interaction.

In this thesis, we comprehensively study the problem of narrative commonsense reasoning, which combines causal knowledge and causal theory. We propose and solve three important sub-problems, i.e., (1) how to automatically obtain large-scale causal knowledge to provide knowledge ground for narrative reasoning? (2) how can we ensure that narrative reasoning systems grasp the causal relationships within narrative events, enabling them to effectively address factual and counterfactual questions? (3) how can we devise robust quantitative methods to evaluate the coherence of AI-generated narrative content, thereby furnishing valuable tools for the community? We proposed a series of methods to constantly improve narrative reasoning from the aspects of knowledge-aware enhancement, causal-theory-based counterfactual reasoning, and hard negatives mining.

8.1 Summary of Thesis

The following sections summarize the main contributions of this thesis.

8.1.1 Automatically Causality Mining and De-biasing

- We propose a rule-based system to automatically extract causal pairs from free-form text, without any human efforts. We also demonstrate its use the the task of cause-to=effect generation.

- Not only the rule-based causal relationship extraction methods, we also develop a de-biased method to improve the precision of causal relationship extraction.

- Extensive experiments on several public benchmarks demonstrate the effectiveness of our proposed method.

8.1.2 Causal Knowledge Enhanced Factual and Counterfactual Reasoning in Narratives

- We devise the two-stage approach to make full utilization of multi-level causalities. Not only that, we have also proposed practical and feasible solutions to solve the sparsity problem of events, contributing to the field of knowledge-enhanced reasoning.

- In this part, we study counterfactual reasoning in narratives from the causal perspective, and formulate the problem with the structural causal model, which simulate the posterior information of background knowledge using the variational process. Different from the previous works that simply learns conditional distribution with the supervised paired data, we are the first to use the causal mechanism to robustly reason about counterfactuals. In addition, we introduce a pre-train classifier and external event causality to mitigate the posterior collapse problem in the variational process, and hence further improve the causality between the counterfactual condition and the generated counterfactual outcome.

- The experiment on several public benchmarks proves the effectiveness of our

method. We conduct a detailed ablation study to illustrate the significance of our proposed strategies.

8.1.3 Narrative Coherence Learning

- We review the problem of narrative coherence learning from the perspective of hard negatives mining. We propose a method of synthesizing negative samples using contrastive narrative.

- To obtain contrastive narrative, we innovatively introduce the Brownian bridge process to ensure the quality of obtained contrastive narrative, which is the key contribution of our work. We believe that our method can bring new research insights to this issue.

- The experiment verifies that (1) the generated contrastive narratives are qualified, and (2) our coherence evaluator is effective and is a general coherence evaluator that is applicable to many downstream tasks.

8.2 Future Directions

In this part, we point out the following potential directions that can further extend our previous work.

- For knowledge-enhanced narrative generation, it can be more abstractly described as retrieving relevant background knowledge from the knowledge base for the input text to enhance the generation process. This process can be induced into the retrieval augmented generation (RAG) framework. Considering the powerful reasoning and generation capabilities of current large models, it is promising to use large models as the base and adopt the RAG technology for

narrative reasoning. On the one hand, by retrieving relevant knowledge, the hallucination problem of large models can be mitigated. On the other hand, based on the large model as the foundation, the internal knowledge within the large model can be fully utilized. Overall, this direction has considerable research value.

- For counterfactual reasoning in narratives, our model can generate the counterfactual output for the counterfactual condition. This points out the potential direction. That is, it is possible to generate high-quality diverse counterfactual samples by perturbing the original story condition. It is relatively easy to only perturb the original conditions, as there is no need to consider the impact of disturbances. This step can be accomplished through many existing works. Next, based on the obtained counterfactual data, we can perform counterfactual data augmentation to obtain a better model. This process can be repeated in a bootstrapping manner.

- As for narrative coherence learning, due to limited computing resources, the model we are currently using is relatively small and the amount of data is also limited. In the future, we can use larger models and larger-scale data for training. Additionally, our model can be viewed as a reward model that evaluates the reward of a narrative text. Therefore, it has the potential to combine our reward model and reinforcement learning algorithms to fine-tune text generation models. The obtained generation model is expected to have a better ability to generate coherent narratives.

References

[1] Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 504–509, 2006.

[2] Mana Ashida and Saku Sugawara. Possible stories: Evaluating situated commonsense reasoning under multiple possible scenarios. *arXiv preprint arXiv:2209.07760*, 2022.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[4] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

[5] C. Bhagavatula, R. L. Bras, C. Malaviya, K. Sakaguchi, A. Holtzman, H. Rashkin, D. Downey, W. T. Yih, and Y. Choi. Abductive commonsense reasoning. In *International Conference on Learning Representations*, 2020.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Deng Cai, Yizhe Zhang, Yichen Huang, Wai Lam, and Bill Dolan. Narrative incoherence detection. *arXiv preprint arXiv:2012.11157*, 2020.

[8] Pengfei Cao, Xinyu Zuo, Yubo Chen, Kang Liu, Jun Zhao, Yuguang Chen, and Weihua Peng. 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, August 2021. Association for Computational Linguistics.

[9] Tommaso Caselli and Piek Vossen. 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, 2017.

[10] Eugene Charniak. *Toward a model of children's story comprehension.* PhD thesis, Massachusetts Institute of Technology, 1972.

[11] J. Chen, C. Gan, S. Cheng, H. Zhou, Y. Xiao, and L. Li. Unsupervised editing for counterfactual stories. 2021.

[12] Jiangjie Chen, Chun Gan, Sijie Cheng, Hao Zhou, Yanghua Xiao, and Lei Li. Unsupervised editing for counterfactual stories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10473–10481, 2022.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. Core: A retrieve-then-edit framework for counterfactual data generation. *arXiv preprint arXiv:2210.04873*, 2022.

[17] L. Du, X. Ding, K. Xiong, T. Liu, and B. Qin. e-care: a new dataset for exploring explainable causal reasoning. 2022.

[18] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2021.

[19] Nouha Dziri, Ehsan Kamalloo, Kory W Mathewson, and Osmar Zaiane. Evaluating coherence in dialogue systems using entailment. *arXiv preprint arXiv:1904.03371*, 2019.

[20] Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. Modeling document-level causal structures for event causal relation identification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1808–1817, 2019.

[21] T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. 2021.

[22] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[23] Roxana Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83, 2003.

[24] Roxana Girju and Dan Moldovan. Mining answers for causation questions. In *Proc. The AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pages 67–82, 2002.

[25] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus, 2019.

[26] Travis Goodwin, Bryan Rink, Kirk Roberts, and Sanda Harabagiu. Utdhlt: Copacetic system for choosing plausible alternatives. 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 461–466, 2012.

[27] Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 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, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[28] Rujun Han, Xiang Ren, and Nanyun Peng. Deer: A data efficient language model for event temporal reasoning. *arXiv preprint arXiv:2012.15283*, 2020.

[29] Changying Hao, Liang Pang, Yanyan Lan, Yan Wang, Jiafeng Guo, and Xueqi Cheng. Sketch and customize: A counterfactual story generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12955–12962, 2021.

[30] Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 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, 2014.

[31] Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. In *IJCAI*, pages 5003–5009, 2019.

[32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[33] Pedram Hosseini, David A Broniatowski, and Mona Diab. Knowledge-augmented language models for cause-effect relation classification. *arXiv preprint arXiv:2112.08615*, 2021.

[34] Zhiting Hu and Li Erran Li. A causal lens for controllable text generation. *Advances in Neural Information Processing Systems*, 34:24941–24955, 2021.

[35] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR, 2017.

[36] Daniel D Hutto. Narrative understanding. In *The Routledge companion to philosophy of literature*, pages 291–301. Routledge, 2015.

[37] Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392, 2021.

[38] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[39] Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. Language generation with multi-hop reasoning on commonsense knowledge graph. *arXiv preprint arXiv:2009.11692*, 2020.

[40] Prathyusha Jwalapuram, Shafiq Joty, and Xiang Lin. Rethinking self-supervision objectives for generalizable coherence modeling. *arXiv preprint arXiv:2110.07198*, 2021.

[41] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus. Hard negative mixing for contrastive learning. 2020.

[42] Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. Adventure: Adversarial training for textual entailment with knowledge-guided examples. *arXiv preprint arXiv:1805.04680*, 2018.

[43] Randy M Kaplan and Genevieve Berry-Rogghe. Knowledge-based acquisition of causal relationships in text. *Knowledge Acquisition*, 3(3):317–337, 1991.

[44] Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. When choosing plausible alternatives, clever hans can be clever. *arXiv preprint arXiv:1911.00225*, 2019.

[45] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[46] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[47] Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. Rankgen: Improving text generation with large ranking models. *arXiv preprint arXiv:2205.09726*, 2022.

[48] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. 2023.

[49] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[50] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[51] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*, 2020.

[52] Dandan Li, Ziyu Guo, Qing Liu, Li Jin, Zequen Zhang, Kaiwen Wei, and Feng Li. Click: Integrating causal inference and commonsense knowledge incorporation for counterfactual story generation. *Electronics*, 2023.

[53] Dandan Li, Ziyu Guo, Qing Liu, Li Jin, Zequn Zhang, Kaiwen Wei, and Feng Li. Click: Integrating causal inference and commonsense knowledge incorporation for counterfactual story generation. *Electronics*, 12(19):4173, 2023.

[54] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[55] Zhongyang Li. *Eventic Graphs Construction and Application Methods for Textual Event Prediction*. PhD thesis, School of Computer Science and Technology, Harbin Institute of Technology, 2021.

[56] Zhongyang Li, Tongfei Chen, and Benjamin Van Durme. Learning to rank for plausible plausibility. *arXiv preprint arXiv:1906.02079*, 2019.

[57] Zhongyang Li, Xiao Ding, Kuo Liao, Bing Qin, and Ting Liu. Causalbert: Injecting causal knowledge into pre-trained models with minimal supervision. *arXiv preprint arXiv:2107.09852*, 2021.

[58] Zhongyang Li, Xiao Ding, and Ting Liu. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*, 2018.

[59] Zhongyang Li, Xiao Ding, Ting Liu, J Edward Hu, and Benjamin Van Durme. Guided generation of cause and effect. IJCAI, 2021.

[60] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*, 2019.

[61] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[62] Li Lin, Yixin Cao, Lifu Huang, Shuang Li, Xuming Hu, Lijie Wen, and Jianmin Wang. Inferring commonsense explanations as prompts for future event generation. *arXiv preprint arXiv:2201.07099*, 2022.

[63] Shih-Ting Lin, Nathanael Chambers, and Greg Durrett. Conditional generation of temporally-ordered event sequences. *arXiv preprint arXiv:2012.15786*, 2020.

[64] Jian Liu, Yubo Chen, and Jun Zhao. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3608–3614, 2020.

[65] Qi Liu, Matt Kusner, and Phil Blunsom. Counterfactual data augmentation for neural machine translation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 187–197, Online, June 2021. Association for Computational Linguistics.

[66] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[67] Zhiyi Luo, Yuchen Sha, Kenny Q Zhu, Seung-won Hwang, and Zhongyuan Wang. Commonsense causal reasoning between short texts. In *KR*, pages 421–431, 2016.

[68] Satya N Majumdar and Henri Orland. Effective langevin equations for constrained stochastic processes. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(6):P06039, 2015.

[69] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

[70] Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[71] N. Miao, H. Zhou, L. Mou, R. Yan, and L. Li. Cgmh: Constrained sentence generation by metropolis-hastings sampling. 2018.

[72] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

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

[74] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins:local descriptor learning loss. 2017.

[75] N. Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 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*, volume abs/1604.01696, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics.

[76] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*, 2016.

[77] Feiteng Mu and Wenjie Li. Enhancing text generation via multi-level knowledge aware reasoning. IJCAI, 2022.

[78] Feiteng Mu and Wenjie Li. Enhancing event causality identification with counterfactual reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 967–975, 2023.

[79] Feiteng Mu and Wenjie Li. A causal approach for counterfactual reasoning in narratives. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

[80] Feiteng Mu and Wenjie Li. Enhancing narrative commonsense reasoning with multilevel causal knowledge. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[81] Feiteng Mu and Wenjie Li. Generating contrastive narratives using the brownian bridge process for narrative coherence learning. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

[82] Feiteng Mu, Wenjie Li, and Zhipeng Xie. Effect generation based on causal reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 527–533, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[83] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710, 2021.

[84] https://chat.openai.com/chat.

[85] Jiao Ou, Jinchao Zhang, Yang Feng, and Jie Zhou. Counterfactual data augmentation via perspective transition for open-domain dialogues. *arXiv preprint arXiv:2210.16838*, 2022.

[86] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[87] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[88] Debjit Paul and Anette Frank. Social commonsense reasoning with multi-head knowledge attention. *arXiv preprint arXiv:2010.05587*, 2020.

[89] Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[90] Judea Pearl. *Causality*. Cambridge university press, 2009.

[91] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[92] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[93] Minh Tran Phu and Thien Huu Nguyen. Graph convolutional networks for event causality identification with rich document-level structures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3480–3490, 2021.

[94] Karl Pichotta and Raymond J Mooney. Learning statistical scripts with lstm recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[95] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The penn discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, 2008.

[96] Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. Counterfactual inference for text classification debiasing. 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 5434–5445, 2021.

[97] L. Qin, V. Shwartz, P. West, C. Bhagavatula, and Y. Choi. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. 2020.

[98] Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. Counterfactual story reasoning and generation. *arXiv preprint arXiv:1909.04076*, 2019.

[99] Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. Backpropagation-based decoding for unsupervised counterfactual and abductive reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, 2020.

[100] Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551, 2022.

[101] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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

[103] Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. Learning causality for news events prediction. In *Proceedings of the 21st international conference*

on World Wide Web, WWW '12, pages 909–918, New York, NY, USA, 2012. Association for Computing Machinery.

[104] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[105] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[106] Ali Razavi, Aäron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with delta-vaes. *arXiv preprint arXiv:1901.03416*, 2019.

[107] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*, volume 293. Springer Science & Business Media, 2013.

[108] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95, 2011.

[109] Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. Tailor: Generating and perturbing text with semantic controls. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Online, August 2022. Association for Computational Linguistics.

[110] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[111] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

[112] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[113] Shota Sasaki, Sho Takase, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui. Handling multiword expressions in causality estimation. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*, 2017.

[114] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[115] Karin Kipper Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

[116] Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. Creating causal embeddings for question answering with minimal supervision. *arXiv preprint arXiv:1609.08097*, 2016.

[117] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, 2016.

[118] Robyn Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, 2012.

[119] Ninke Stukker, Ted Sanders, and Arie Verhagen. Causality in verbs and in discourse connectives: Converging evidence of cross-level parallels in dutch linguistic categorization. *Journal of Pragmatics*, 40(7):1296–1322, 2008.

[120] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[121] Chris Sweeney and Maryam Najafian. A transparent framework for evaluating unintended demographic bias in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1662–1667, Florence, Italy, July 2019. Association for Computational Linguistics.

[122] Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. Debiasing nlu models via causal intervention and counterfactual reasoning. 2022.

[123] Ryoko Tokuhisa, Keisuke Kawano, Akihiro Nakamura, and Satoshi Koide. Enhancing contextual word representations using embedding of neighboring entities in knowledge graphs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3175–3186, 2022.

[124] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[125] Tom Trabasso. The role of causal reasoning in understanding narratives. *From orthography to pedagogy: Essays in honor of Richard L. Venezky*, pages 81–106, 2005.

[126] Yui Uehara, Tatsuya Ishigaki, Kasumi Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. Learning with con-

trastive examples for data-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2352–2362, 2020.

[127] Paul van Den Broek, Brian Linzie, Charles Fletcher, and Chad J Marsolek. The role of causal discourse structure in narrative writing. *Memory & Cognition*, 28(5):711–721, 2000.

[128] Paul van den Broek, Lisa Rohleder, and Darcia Narváez. Causal inferences in the comprehension of literary texts. 1996.

[129] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[130] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30 of *NIPS'17*, pages 5998–6008, Red Hook, NY, USA, 2017. Curran Associates Inc.

[131] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[132] Chaoqi Wang, Adish Singla, and Yuxin Chen. Teaching an active learner with contrastive examples. *Advances in Neural Information Processing Systems*, 34:17968–17980, 2021.

[133] Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. Language modeling via stochastic processes. *arXiv preprint arXiv:2203.11370*, 2022.

[134] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[135] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

[136] C. Y. Wu, R. Manmatha, A. J. Smola, and Philipp Krhenbühl. Sampling matters in deep embedding learning. *IEEE*, 2017.

[137] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288*, 2021.

[138] Yuqiang Xie, Yue Hu, Luxi Xing, Chunhui Wang, Yong Hu, Xiangpeng Wei, and Yajing Sun. Enhancing pre-trained language models by self-supervised learning for story cloze test. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, pages 271–279. Springer, 2020.

[139] Zhipeng Xie and Feiteng Mu. Boosting causal embeddings via potential verb-mediated causal patterns. In *IJCAI*, pages 1921–1927, 2019.

[140] Zhipeng Xie and Feiteng Mu. Distributed representation of words in cause and effect spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7330–7337, 2019.

[141] Wei Xu, Wenjie Li, Mingli Wu, Wei Li, and Chunfa Yuan. Deriving event relevance from the ontology constructed with formal concept analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 480–489. Springer, 2006.

[142] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. page 126–142, Berlin, Heidelberg, 2020. Springer-Verlag.

[143] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings*

of the AAAI Conference on Artificial Intelligence, volume 33, pages 7378–7385, 2019.

[144] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(5):1–46, 2021.

[145] Wenlin Yao and Ruihong Huang. Temporal event knowledge acquisition via identifying narratives. *arXiv preprint arXiv:1805.10956*, 2018.

[146] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. 2018.

[147] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? 2019.

[148] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

[149] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[150] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR, 2020.

[151] Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Xiaokang Yang, and Pinyan Lu. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. KDD '22, page 2461–2470, New York, NY, USA, 2022. Association for Computing Machinery.

[152] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models, 2022. *URL https://arxiv.org/abs/2205.01068*, 3:19–0, 2023.

[153] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[154] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*, 2019.

[155] Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. Constructing and embedding abstract event causality networks from text snippets. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 335–344, New York, NY, USA, 2017. ACM, Association for Computing Machinery.

[156] Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. Temporal common sense acquisition with minimal supervision. *arXiv preprint arXiv:2005.04304*, 2020.

[157] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018.

[158] Yucheng Zhou, Xiubo Geng, Tao Shen, Guodong Long, and Daxin Jiang. Eventbert: A pre-trained model for event correlation reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 850–859, 2022.

[159] Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. Claret: Pre-training a correlation-aware context-to-event transformer for event-centric generation and classification. *arXiv preprint arXiv:2203.02225*, 2022.

[160] Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*, 2017.

[161] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[162] Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. Improving event causality identification via self-supervised representation learning on external causal statement. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2162–2172, Online, August 2021. Association for Computational Linguistics.

[163] Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. LearnDA: Learnable knowledge-guided data augmentation for event causality identification. 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 3558–3571, Online, August 2021. Association for Computational Linguistics.

[164] Xinyu Zuo, Yubo Chen, Kang Liu, and Jun Zhao. 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), December 2020. International Committee on Computational Linguistics.