Episodic Memory Retrieval from LLMs: A Neuromorphic Mechanism to Generate Commonsense Counterfactuals for Relation Extraction

Xin Miao[1], **Yongqi Li**[1], **Shen Zhou**[1], **Tieyun Qian**[1,2*]
[1] School of Computer Science, Wuhan University, China
[2] Intellectual Computing Laboratory for Cultural Heritage, Wuhan University, China
{miaoxin,liyongqi,shenzhou,qty}@whu.edu.cn

Abstract

Large language models (LLMs) have achieved satisfactory performance in counterfactual generation. However, confined by the stochastic generation process of LLMs, there often are misalignments between LLMs and humans which hinder LLMs from handling complex tasks like relation extraction. As a result, LLMs may generate commonsense-violated counterfactuals like 'eggs were produced by a box'.

To bridge this gap, we propose to *mimick the episodic memory retrieval*, the working mechanism of the human hippocampus, *to align LLMs' generation process with that of humans*. In this way, LLMs can derive experience from their extensive memory, which keeps in line with the way humans gain commonsense. We then implement two central functions in the hippocampus, i.e., *pattern separation* and *pattern completion*, to retrieve the episodic memory from LLMs and generate commonsense counterfactuals for relation extraction. Experimental results demonstrate the improvements of our framework over existing methods in terms of the quality of counterfactuals[1].

1 Introduction

Generating counterfactual augmented data to mitigate spurious correlations in neural networks is a rising trend in recent years (Wen et al., 2022; Zhang et al., 2023). The counterfactual is usually generated by flipping the label of an instance with minimal editing (Kaushik et al., 2019). Large language models (LLMs) (OpenAI, 2022, 2023) are proven to be proficient in generating counterfactuals for coarse-grained tasks (Ross et al., 2021; Wen et al., 2022) such as sentiment analysis and natural language inference. However, confined by the inherent stochastic generation process, LLMs confront challenges when generating counterfactuals for complex tasks like relation extraction (RE) (Li et al., 2023b) due to misalignments between LLMs and humans. Specifically, LLMs may disregard the commonsense constraint (Miao et al., 2023) between entities, resulting in the deviation of counterfactuals from real-world scenarios. As shown in Fig. 1 (a), the counterfactual generated by *davinci-002* violates commonsense because '*eggs*' cannot be produced by a '*box*'.

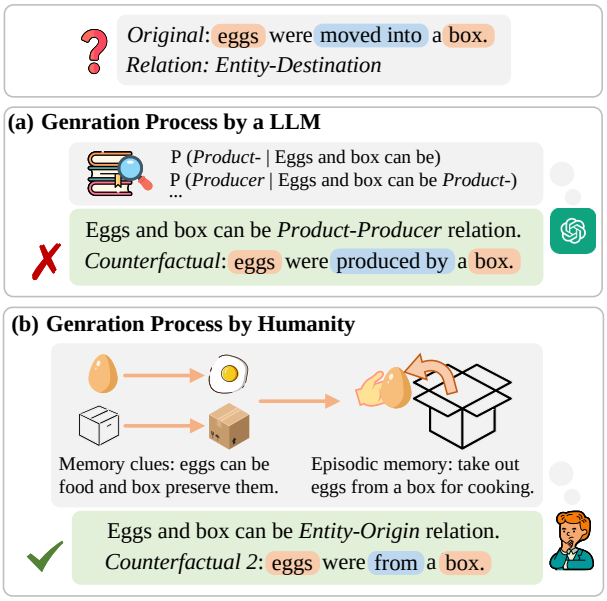

Figure 1: A comparison of the counterfactual generation process by *davinci-002* (a) and that by humans (b). The words in orange and blue denote entities and causal term, respectively.

In contrast to the stochastic process, when human beings engage in counterfactual thinking, the process could be activated by either a past event from episodic memory or a causal system from semantic memory (Nyhout and Ganea, 2019). Neuroscientists believe that the hippocampus supports core computations and representations of episodic memory systems (Knierim and Neunuebel, 2016). Moreover, two complementary computations–**pattern separation (PS)** and **pat-**

*Corresponding author.
[1]The code and data used in the experiment are available at: https://github.com/NLPWM-WHU/PSPC.

Findings of the Association for Computational Linguistics: ACL 2024, pages 2489–2511
August 11-16, 2024

tern completion (PC)–are viewed to be central to the function of the hippocampus (Kumaran et al., 2016), where PS refers to mnemonic discrimination of similar experiences and PC retrieves holistic experiences given a clue. For example, when we reason potential relations between '*eggs*' and '*box*', we initially retrieve a common scenario '*take out eggs from a box for cooking*' from our episodic memories (Tulving, 2002). We then uncover the potential relation '*entity-origin*' between two entities, as shown in Fig. 1 (b). Since the process is controlled by our past experiences, the generated counterfactual naturally follows commonsense.

In view of the fundamental difference, we propose to align LLMs' relation discovery process with that of humans by mimicking the neuromorphic mechanism in the human hippocampus, to generate commonsense counterfactuals for relation extraction. To this end, we realize the **P**attern **S**eparation (PS) and **P**attern **C**ompletion (PC) functions to retrieve episodic memory from LLMs. Specifically, we first perform PS by dividing entities with their typical attributes, e.g., *'Form', 'Usage', 'Purpose'*. We then perform PC by pairing the attributes of two entities and use the combined attribute as the clue to recall the complete scenario, e.g., *'eggs Usage: food'* and *'box Purpose: preserve'* retrieve the scenario of *'take out eggs from the box for cooking'*. We finally implement PS and PC in LLMs using in-context learning and chain-of-thought techniques.

We validate our proposed framework with both data augmentation and human evaluation approaches on three RE datasets. Experimental results demonstrate that with the assistance of an episodic memory retrieval mechanism, the counterfactual generation capability of the LLMs has surpassed current mainstream methods. The main contributions of this work include:

- We introduce a neuromorphic mechanism to align LLMs with humans during the counterfactual generation process, which is the first attempt towards this direction.
- We develop two functions, i.e., pattern separation and pattern completion, to realize this mechanism for generating commonsense counterfactuals for the relation extraction task.
- We extensively evaluate our framework on three typical RE datasets. The results prove the significantly increased performance and the quality of the generated counterfactuals.

2 Related Work

Generating counterfactual for data augmentation is proven to be simple and effective in mitigating shortcut learning in deep neural networks (Kaushik et al., 2019). Researches along this line in natural language understanding focus on sentiment classification or natural language inference tasks (Yang et al., 2021; Chen et al., 2021; Robeer et al., 2021; Ross et al., 2021; Wen et al., 2022), for which the label flipping targets are deterministic, i.e., from positive to negative, or vice versa. When generating counterfactuals for relation extraction (Zhang et al., 2023; Li et al., 2023b), the labels are just the relations with many choices, and an arbitrary relation between two entities may violate the commonsense. A recent method CCG (Miao et al., 2023) takes the commonsense constraint into account. However, CCG is a small language model based approach and it relies on external knowledge to keep consistency with commonsense. In contrast, our work is LLM-based and we make the first attempt to derive commonsense from LLMs themselves by the alignment of the generation process.

Guiding LLMs with prompting engineering for complex tasks is becoming a hot topic recently. Existing prompting methods can be roughly categorized into three lines: divide and conquer approach, self-reflection, and role-playing. The approaches based on chain-of-thought (Wei et al., 2022; Yao et al., 2022; Shinn et al., 2023; Yao et al., 2023; Qi et al., 2023) belong to the first line. They solve complex reasoning problems by generating intermediate reasoning steps. The approaches along the second line (Shinn et al., 2023; Huang et al.; Madaan et al., 2023) correct their previous outputs through self-feedback or external feedback. Lastly, the approaches along the third line (Li et al., 2023a; Xu et al., 2023; Wang et al., 2023) assign specific roles to LLMs based on task requirements, thus constraining their output behaviors. It is worth noting that our proposed method, which mimics the working mechanism of the hippocampus, is a new line totally different from all existing ones.

3 Our Proposed Neuromorphic Method

This section presents our proposed method, including the episodic memory retrieval (EMR) mechanism and its pattern separation (PS) and pattern completion (PC) functions, along with the specific implementation details on LLMs.

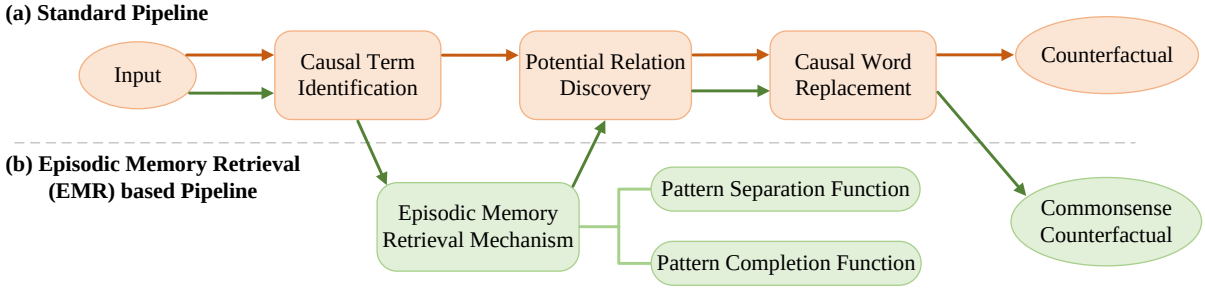

Figure 2: Schematic comparison between the standard pipeline of exiting methods (a) and our episodic memory retrieval based pipeline (b) for RE counterfactual generation.

3.1 Problem Definition

Let $\mathcal{X} = \{(x_i = (e_i, c_i), y_i)\}$ be the dataset, where $x_i \in \mathcal{X}$ is the i-th sentence containing a known entity pair $e_i = (e_i^1, e_i^2)$ and the unknown causal term c_i which determines the state between entities, and $y_i \in \mathcal{Y}$ is the corresponding relation between two entities e_i^1 and e_i^2. Given a sentence x_i, the relation extraction (RE) task aims to extract the relation y_i, and RE counterfactual generation aims to generate $\hat{x}_i = ((e_i, \hat{c}_i), \hat{y}_i)$, where $\hat{c}_i \neq c_i$, $\hat{y}_i \neq y_i$, i.e., altering the causal term to change the relation.

The process of existing RE counterfactual generation methods (Zhang et al., 2023; Li et al., 2023b; Miao et al., 2023) all adopt a standard pipeline with following three steps, as shown in Fig. 2 (a).

(1) **Causal term identification**:

$$c_i = \phi(x_i, e_i, y_i), \tag{1}$$

where ϕ is a strategy for recognizing the causal term c_i from the sentence x_i. Since this step is not the focus of our work, we follow the one used in (Li et al., 2023b), refer to Appendix A.4.

(2) **Potential relation discovery**:

$$\hat{y}_i = \sigma(x_i, e_i), \quad (\hat{y}_i \neq y_i), \tag{2}$$

where σ is a strategy to find all potential relations $\hat{y}_i$ which is suitable for the entity pair e_i.

(3) **Causal term replacement**:

$$\hat{x}_i = \rho((e_i, \hat{c}_i), \hat{y}_i), \quad (\hat{c}_i \neq c_i), \tag{3}$$

where ρ is an operation to substitute c_i with a proper $\hat{c}_i$ such that the original label y_i can be changed into a reasonable $\hat{y}_i$. Note the meaning of **the commonsense constraint**[2] in the RE counterfactual generation task is twofold, i.e., both the causal term $\hat{c}_i$ is proper and the flipped label $\hat{y}_i$ is reasonable for the entity pair e_i.

[2]We provide detailed causal analysis in Appendix C.

For coarse-grained counterfactual generation tasks like sentiment analysis, they just need to assign an opposite label to the original sample. Their focus is to find and replace c_i with $\hat{c}_i$, which is much easier. For example, given '*The movie is wonderful*' with a '*positive*' label, a counterfactual could be '*The movie is bad*' with a '*negative*' label.

The RE counterfactual generation task is more challenging since there are a lot of labels, e.g., 42 relations on TACRED, and causal terms are closely related to two entities. For example, given '*eggs were moved into a box*', the counterfactual '*eggs were produced by a box*' with the '*product-producer*' label is wrong because the causal term '*produced by*' is not suitable for '*eggs-box*'.

Thanks to the powerful generation capability, an LLM like ChatGPT (OpenAI, 2022) can perform the causal term replacement well. However, given '*cheese is flowed into food banks*', ChatGPT may produce a counterfactual like '*cheese is donated to food banks*' with the '*content-container*' relation, which is still wrong because '*food banks*' cannot serve as the container, and an appropriate relation is '*entity-destination*'.

In light of this, we focus on the second step of the potential relation discovery to meet the commonsense constraint. To this end, we propose to incorporate the episodic memory retrieval (EMR) mechanism into the pipeline to align the generation process of LLMs with that of humans, as shown in Fig. 2 (b). The process can be formalized as:

$$s_i = \mathit{EMR}(e_i, \mathcal{M}_{llm}), \tag{4}$$

$$\hat{y}_i = \mathcal{M}_{llm}(e_i, s_i), \quad (\hat{y}_i \neq y_i), \tag{5}$$

where *EMR* is the neuromorphic mechanism to retrieve the episodic memory s_i from a LLM that is highly correlated with the entity pair e_i, and $\mathcal{M}_{llm}$ further changes the original label y_i for the entity pair e_i based on s_i.

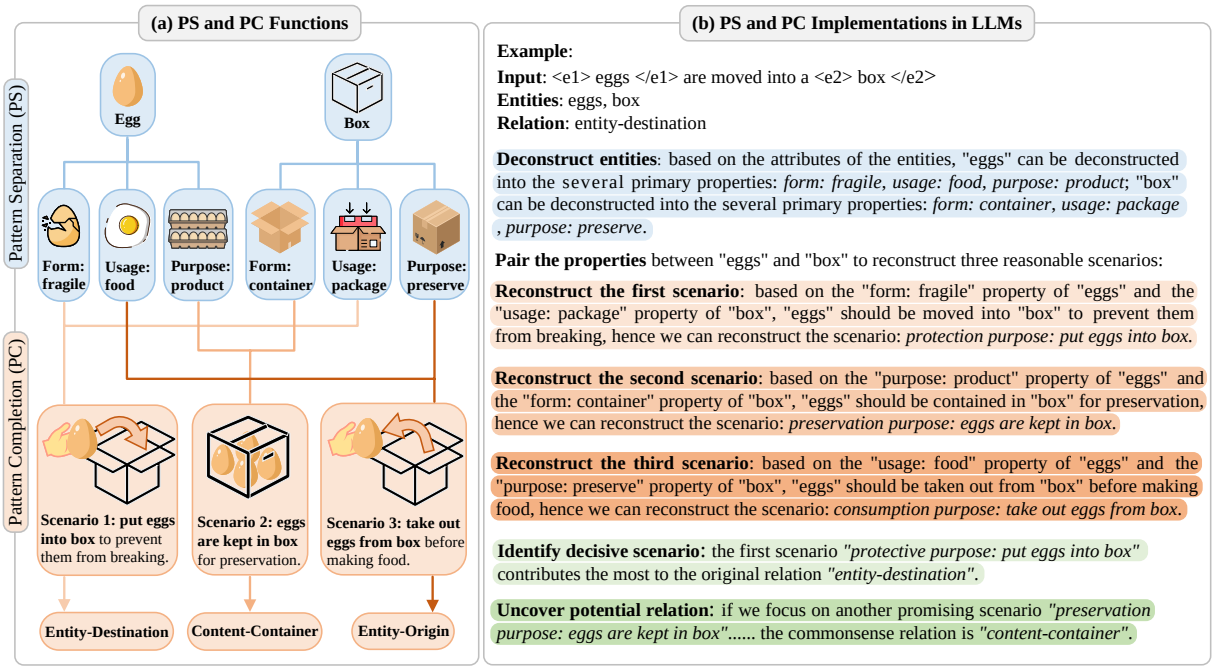

Figure 3: An illustration example of pattern separation (PS) and pattern completion (PC) functions (a) and their implementations in LLMs (b).

3.2 Retrieving Episodic Memory from Hippocampus

Episodic memory enables human beings to remember past experiences (Tulving, 2002). The fundamental property of episodic memory is to store and retrieve the memory of a particular single event involving an association between items such as the place and the object (Rolls, 2013). Therefore, episodic memory binds together the diverse co-occurring items that make up the specific events of our lives (Ngo et al., 2021). The hippocampus supports core computations and representations of episodic memory systems. Two complementary computations–**pattern separation (PS)** and **pattern completion (PC)**–are viewed to be central to the function of the hippocampus for storing and retrieving details of specific experiences (Kumaran et al., 2016).

Accurate episodic memory requires remembering details with high specificity so that they can be mnemonically discriminated from other similar memories (Ngo et al., 2021). Pattern separation aids mnemonic discrimination by reducing the degree of representational similarity among overlapping experiences (Ngo et al., 2021). On the contrary, pattern completion is a process that takes out a pattern fragment and fills in the remaining attributes (Kumaran et al., 2016). In short, pattern completion enables the recapitulation of an entire event from a partial cue (Ngo et al., 2021).

3.3 Connecting Episodic Memory Retrieval with RE Counterfactual Generation

Due to the entity-centric nature (Zhang et al., 2023), i.e., the given entities cannot be altered, the generation of RE counterfactuals can be mimicked by *performing pattern separation and pattern completion computations on entities* to retrieve similar episodic memories.

Specifically, the entity's attributes, which determine its usage scenarios and states, are critical to the mnemonic discrimination, thus we define pattern separation for our task as: *PS is used to decompose the attributes of the given entities*. Moreover, the entity's attributes serve as partial cues for reconstructing complete scenarios, and we define pattern completion as: *PC is used to associate the attributes between the given entities and then reason corresponding scenarios*. Below we present these two functions.

3.3.1 Pattern Separation (PS) Function

Since the essence of pattern separation is to extract distinctive properties between similar episodic memories, the design of the PS function should pay more attention to the discriminative property of the entity. To this end, we define three attributes for the entities in context: *form*, *usage*, and *purpose* [3], as shown in Fig. 3 (a). Formally, the pattern

[3]We keep the attributes consistent with ConceptNet (Speer et al., 2017), a carefully designed knowledge graph, to ensure

separation function is defined as:

$$(a_i^1, a_i^2) = \mathcal{PS}((e_i^1,\ e_i^2), \alpha), \tag{6}$$

where the $\mathcal{PS}$ function decomposes the head and tail entity e_i^1 and e_i^2 into the attribute a_i^1 and a_i^2 according to a specific aspect α, respectively.

3.3.2 Pattern Completion (PC) Function

Since pattern completion aims to compose relevant scenarios between entities, the PC function should pair the attribute of the head entity with the attribute of the tail entity. Formally, the pattern completion function is defined as:

$$s_i = \mathcal{PC}(e_i^1.a_i^1,\ e_i^2.a_i^2), \tag{7}$$

where the $\mathcal{PC}$ function combines the attribute a_i^1 of the head entity e_i^1 with the attribute a_i^2 of the tail entity e_i^2 and then generates the scenario s_i .

3.4 Retrieving Episodic Memory from LLMs

This section presents our method to retrieve episodic memory from LLMs by realizing the PS and PC functions.

3.4.1 Implementing PS Function in LLMs

We employ the widely used in-context learning technique to implement the PS and PC functions in LLMs with examples. The entity is decomposed into the 'attribute: value' pair, e.g., '*form: fragile entity*', '*usage: food*', and '*purpose: product*' of the entity '*eggs*', as shown in blue spans in Fig. 3 (b). The learning process can be formalized as:

$$(a_r^1, a_r^2) = \mathcal{M}_{llm}(\mathcal{PS}((e_r^1, e_r^2), \alpha)), \tag{8}$$

where the $\mathcal{PS}$ function is realized by using an in-context example, and $\mathcal{M}_{llm}$ is a specific LLM which decomposes the request entities e_r^1 and e_r^2 into attributes a_r^1 and a_r^2 according to a specific aspect α by imitating $\mathcal{PS}$. It is worth noting that LLMs can automatically adjust α based on the entity's context. The detail is given in Appendix A.4.

3.4.2 Implementing PC Function in LLMs

To reconstruct a reasonable scenario using entities' attributes, we utilize the chain-of-thought (CoT) approach (Wei et al., 2022) to constrain LLMs with the intermediate processes. For example, given the '*fragile*' attribute of '*eggs*' and the '*package*' attribute of '*box*', the reasoning process '*eggs should be moved into box to prevent them from breaking*' takes into account the attributes of two entities, as shown in the orange spans of Fig. 3 (b).

Furthermore, to ensure that the attribute combination and the reconstructed scenario are reasonable, we require the reconstructed first scenario to be equivalent to the original instance which is supposed to be sensible. By doing this, the first scenario can serve as a sample constraint to the rational planning of LLMs. We formalize the learning process as:

$$s_r = \mathcal{M}_{llm}(\mathcal{PC}(e_r^1.a_r^1,\ e_r^2.a_r^2)), \tag{9}$$

where the $\mathcal{PC}$ function is also represented as an in-context example, and $\mathcal{M}_{llm}$ reconstructs the scenario s_r for the request entities' attributes a_r^1 and a_r^2 through imitating the planning and reasoning process in $\mathcal{PC}$.

After acquiring all scenarios, we encourage $\mathcal{M}_{llm}$ to select a scenario which implies a relation different from the first scenario. Then, we let $\mathcal{M}_{llm}$ determine the relation between entities in the selected scenario. If it differs from the original relation, it is considered as the potential relation. Otherwise, it will be discarded to meet the counterfactual requirement, i.e., $\hat{y}_i \neq y_i$.

the generality and diversity of episodic memory.

Dataset	Train	Dev	Test	Relation
SemEval	7200	800	2715	19
TACRED	68124	22631	15509	42
ACE2005	576	192	1607/2015/2680	6

Table 1: Statistics of experimental dataset. Note that the vague relation 'Other' in SemEval and 'no_relation' in TACRED are excluded during the generation.

4 Experiments

This section presents the experimental settings, results, and corresponding analyses.

4.1 Evaluation Protocol

Data Augmentation Evaluation Spurious correlations are particularly prevalent in low-resource (Nan et al., 2021) and out-of-domain (OOD) (Calderon et al., 2022) settings. Therefore, we use these two settings to validate counterfactuals' impacts on mitigating spurious correlations. Following previous work (Li et al., 2022; Miao et al., 2023), we employ SemEval (Hendrickx et al., 2019) and TACRED (Zhang et al., 2017) for low-resource, and ACE2005 (Grishman et al., 2005) for OOD experiments, respectively. The statistics are shown in Table 1. For details, see Appendix A.1.

Method	3%	5%	7%	10%	3%	5%	7%	10%
	R-BERT				R-RoBERTa			
SemEval								
Original	$59.31_{1.46}$	$68.66_{1.77}$	$69.90_{0.65}$	$76.47_{1.14}$	$64.27_{3.20}$	$69.99_{1.84}$	$72.37_{1.72}$	$78.27_{1.07}$
Synonym Rep.	$60.06_{1.15}$	$69.57_{1.80}$	$72.00_{1.27}$	$\underline{77.94}_{1.76}$	$62.89_{3.38}$	$71.24_{3.73}$	$72.57_{3.76}$	$78.51_{0.74}$
Back Trans.	$56.68_{1.69}$	$64.02_{2.28}$	$67.24_{2.56}$	$75.53_{1.53}$	$61.10_{2.85}$	$68.00_{2.79}$	$70.74_{1.84}$	$78.23_{1.13}$
BERT-MLM	$62.00_{2.38}$	$67.97_{1.51}$	$70.56_{1.05}$	$77.24_{0.90}$	$63.90_{3.69}$	$70.28_{3.06}$	$71.61_{1.84}$	$77.65_{1.43}$
MICE	$60.74_{0.47}$	$70.93_{1.79}$	$72.20_{0.68}$	$77.40_{1.33}$	$66.26_{1.24}$	$72.91_{1.46}$	$74.86_{1.00}$	$78.71_{0.55}$
AutoCAD	$62.09_{1.39}$	$\underline{71.18}_{1.36}$	$\underline{72.30}_{1.49}$	$77.86_{0.44}$	$\underline{66.98}_{0.79}$	$\underline{74.17}_{1.91}$	$\underline{75.26}_{1.04}$	$\underline{78.78}_{1.43}$
CoCo	$62.24_{1.10}$	$69.97_{1.61}$	$70.90_{2.10}$	$77.40_{0.66}$	$65.57_{2.73}$	$74.16_{2.16}$	$74.76_{0.71}$	$78.40_{0.88}$
CF-CoT	$\underline{63.08}_{1.43}$	$70.52_{0.90}$	$72.27_{0.85}$	$76.37_{0.40}$	$64.32_{1.35}$	$72.95_{0.63}$	$74.05_{0.99}$	$78.05_{0.68}$
PSPC (Ours)	$\mathbf{66.91}_{1.46}$	$\mathbf{72.11}_{0.78}$	$\mathbf{73.62}_{1.02}$	$\mathbf{78.17}_{0.65}$	$\mathbf{68.61}_{1.37}$	$\mathbf{75.15}_{1.48}$	$\mathbf{75.72}_{0.83}$	$\mathbf{79.20}_{1.09}$
TACRED								
Original	$15.59_{1.41}$	$23.97_{1.63}$	$30.16_{1.33}$	$35.65_{1.39}$	$21.39_{0.71}$	$32.25_{1.41}$	$34.41_{1.40}$	$40.13_{0.77}$
Synonym Rep.	$18.24_{0.43}$	$24.86_{1.49}$	$29.34_{0.83}$	$35.36_{0.57}$	$21.73_{0.69}$	$31.24_{1.31}$	$32.70_{2.06}$	$40.17_{1.05}$
Back Trans.	$17.67_{1.61}$	$25.41_{1.37}$	$28.04_{1.92}$	$35.42_{1.08}$	$22.08_{0.79}$	$32.08_{1.95}$	$33.78_{1.45}$	$40.79_{0.98}$
BERT-MLM	$\underline{18.45}_{1.08}$	$23.64_{1.65}$	$27.56_{1.26}$	$33.34_{1.21}$	$22.45_{1.35}$	$30.70_{1.11}$	$32.09_{0.50}$	$37.96_{0.64}$
MICE	$17.48_{2.43}$	$24.73_{0.67}$	$30.40_{1.43}$	$36.15_{2.08}$	$22.28_{2.23}$	$32.20_{1.23}$	$\underline{36.12}_{1.68}$	$\underline{41.10}_{0.32}$
AutoCAD	$17.29_{1.64}$	$25.51_{1.64}$	$29.95_{0.89}$	$\underline{36.83}_{0.98}$	$21.47_{1.26}$	$31.72_{1.35}$	$35.36_{1.30}$	$40.78_{0.47}$
CoCo	$17.17_{1.25}$	$24.39_{0.92}$	$29.25_{1.58}$	$35.61_{1.01}$	$22.09_{1.32}$	$31.32_{1.57}$	$34.17_{1.65}$	$40.63_{0.84}$
CF-CoT	$17.51_{0.84}$	$\underline{27.76}_{1.94}$	$\underline{30.78}_{0.39}$	$35.89_{0.85}$	$\underline{23.18}_{2.50}$	$\underline{34.33}_{1.24}$	$35.43_{0.63}$	$40.26_{0.85}$
PSPC (Ours)	$\mathbf{19.16}_{1.51}$	$\mathbf{28.19}_{1.43}$	$\mathbf{32.44}_{1.79}$	$\mathbf{36.87}_{0.80}$	$\mathbf{25.24}_{1.24}$	$\mathbf{35.33}_{1.14}$	$\mathbf{36.42}_{0.77}$	$\mathbf{41.17}_{1.05}$

Table 2: Results for data augmentation evaluation under low-resource settings on SemEval and TACRED. The values in **bold** and those underlined are the best and the second best scores. The subscript denotes the standard deviation.

Following previous work (Zhang et al., 2023; Miao et al., 2023), we employ R-BERT[4] (Wu and He, 2019) and RoBERTa[5] (Liu et al., 2019) as base models for relation extraction. These base models are applied to the original and augmented data generated by various counterfactual methods.

All hyper-parameters except the epoch are set to their default values. We use the validation set to select the optimal value for the epoch setting of the base model and report the mean and standard deviation of micro-F1 scores over 5 random seeds.

Human Evaluation We conduct the human study (Hao et al., 2021; Treviso et al., 2023) as a subjective assessment of commonsense evaluation. Specifically, we randomly select 100 generated counterfactuals from SemEval for each compared method and ask three annotators to give their scores based on the rationality of relations between entities. The scores follow a 3-point scale: reasonable (2), marginally reasonable (1), and not reasonable (0). Refer to Appendix A.2 for the criteria.

Baselines We employ three types of comparative methods as baselines. The first type is the conventional methods, including SYNONYM REPLACEMENT (Zhang et al., 2015), BACK TRANSLATION (Sennrich et al., 2015), and BERT-MLM (Jiao et al., 2019). The second type is the small language model (SLM) based counterfactual generation methods, including MICE (Ross et al., 2021), AUTOCAD (Wen et al., 2022), COCO (Zhang et al., 2023). The third type is the large language model (LLM) based counterfactual generation method CF-COT (Li et al., 2023b).

To ensure fairness in terms of quantity, all compared methods perform data augmentation once at most for each instance.

Implementation Details We utilize the API provided by the OpenAI[6] to evaluate our proposed mechanism. To prevent interference from version updates, we choose *gpt-3.5-turbo-0613* (GPT-3.5 for short) as a representative of LLMs to perform all experiments for CF-CoT and our PSPC, and we set the temperature to 0 for obtaining stable results.

Due to the space limit, the descriptions for datasets and baselines and the detailed PSPC implementations are given in Appendix A. Parameter and sensitivity analysis are given in Appendix B.

[4] https://github.com/monologg/R-BERT
[5] https://huggingface.co/roberta-base
[6] https://openai.com/

Method	WL → BC	WL → BN	WL → NW	WL → BC	WL → BN	WL → NW
	R-BERT			R-RoBERTa		
Original	$70.20_{2.13}$	$72.03_{2.27}$	$69.03_{1.91}$	$74.31_{1.25}$	$72.45_{1.47}$	$73.46_{1.73}$
Synonym Rep.	$71.63_{0.46}$	$73.37_{0.89}$	$70.03_{1.19}$	$74.32_{1.19}$	$73.32_{1.64}$	$74.29_{1.00}$
Back Trans.	$\underline{71.86}_{0.84}$	$\underline{73.52}_{0.78}$	$69.56_{0.63}$	$\underline{75.54}_{1.36}$	$73.42_{0.53}$	$74.96_{0.78}$
BERT-MLM	$71.42_{0.69}$	$72.85_{0.95}$	$\underline{70.14}_{1.01}$	$74.88_{1.01}$	$72.89_{0.12}$	$74.24_{1.15}$
MICE	$71.18_{1.11}$	$72.60_{1.63}$	$69.47_{1.60}$	$74.64_{0.99}$	$72.81_{0.81}$	$\underline{74.98}_{0.38}$
AutoCAD	$71.29_{1.41}$	$72.64_{0.93}$	$69.67_{0.72}$	$\mathbf{75.64}_{0.72}$	$\underline{73.49}_{1.20}$	$74.21_{0.83}$
CoCo	$70.76_{1.36}$	$72.02_{1.03}$	$69.49_{1.38}$	$75.28_{1.52}$	$72.41_{1.47}$	$74.65_{0.82}$
CF-CoT	$70.61_{1.34}$	$71.99_{1.04}$	$69.84_{1.07}$	$73.16_{0.47}$	$71.43_{0.96}$	$72.87_{0.33}$
PSPC (Ours)	$\mathbf{73.11}_{1.79}$	$\mathbf{73.99}_{1.43}$	$\mathbf{70.20}_{1.02}$	$\mathbf{75.64}_{1.33}$	$\mathbf{74.31}_{1.79}$	$\mathbf{75.09}_{0.57}$

Table 3: Results for data augmentation evaluation under out-of-domain settings on ACE 2005.

Method	3%	5%	7%	10%	3%	5%	7%	10%
	R-BERT				R-RoBERTa			
MICE	$60.74_{0.47}$	$70.93_{1.79}$	$72.20_{0.68}$	$77.40_{0.54}$	$66.26_{1.24}$	$72.91_{1.46}$	$74.86_{1.00}$	$78.71_{0.55}$
MICE w/ Random	$62.04_{1.06}$	$70.18_{1.32}$	$72.56_{1.17}$	$75.70_{0.56}$	$65.34_{2.64}$	$73.31_{2.00}$	$74.74_{1.11}$	$77.78_{1.04}$
MICE w/ CF-CoT	$63.60_{1.51}$	$71.35_{1.05}$	$72.31_{1.20}$	$76.53_{1.01}$	$67.01_{2.58}$	$75.25_{2.30}$	$74.90_{1.60}$	$78.02_{1.42}$
MICE w/ PSPC	$\mathbf{65.10}_{0.86}$	$\mathbf{72.10}_{1.23}$	$\mathbf{73.14}_{1.01}$	$\mathbf{77.48}_{1.15}$	$\mathbf{67.76}_{1.86}$	$\mathbf{75.86}_{1.60}$	$\mathbf{75.53}_{1.26}$	$\mathbf{78.89}_{0.99}$
AutoCAD	$62.09_{1.39}$	$71.18_{1.36}$	$72.30_{1.49}$	$77.86_{0.44}$	$66.98_{0.79}$	$74.17_{1.91}$	$75.26_{1.04}$	$78.78_{1.43}$
AutoCAD w/ Random	$62.47_{2.81}$	$70.26_{1.86}$	$72.34_{0.87}$	$76.99_{0.48}$	$65.68_{3.01}$	$72.79_{1.67}$	$74.75_{2.11}$	$77.73_{1.08}$
AutoCAD w/ CF-CoT	$63.84_{0.87}$	$71.21_{0.97}$	$73.89_{0.94}$	$78.00_{0.74}$	$67.19_{1.71}$	$74.62_{1.47}$	$75.74_{1.26}$	$78.91_{1.16}$
AutoCAD w/ PSPC	$\mathbf{65.25}_{1.11}$	$\mathbf{72.35}_{1.07}$	$\mathbf{74.09}_{0.65}$	$\mathbf{78.65}_{0.90}$	$\mathbf{68.09}_{2.19}$	$\mathbf{75.93}_{1.32}$	$\mathbf{76.21}_{0.56}$	$\mathbf{79.48}_{0.46}$

Table 4: Results of the synthetic experiment on SemEval.

4.2 Results for Data Augmentation Evaluation

The results for data augmentation evaluation under the low-resource and out-of-domain settings are shown in Table 2 and Table 3, respectively. From the results, we have the following observations.

(1) Our proposed PSPC method achieves the best performance with relatively small standard deviations across all different settings. This clearly demonstrates the capability of our episodic memory mechanism and its realized functions in generating counterfactuals for data augmentation.

(2) CF-CoT, which is also an LLM-based counterfactual method, is inferior to other baselines in many cases. Note that the only difference between CF-CoT and ours [7] is the pattern separation and pattern completion procedure in potential relation discovery. This shows that an LLM is prone to make errors when lacking proper guidance, but our PS and PC functions can effectively lead the LLM to recall similar scenarios around the entity pair.

(3) We observe that counterfactual augmentation methods are generally more effective in alleviating the impact of spurious correlations than conventional ones under low-resource settings. However, two conventional methods BACK TRANS. and BERT-MLM perform the second best in most cases under OOD settings. The reason might be that the relations in ACE05 are coarse-grained and not specific enough, which prevents counterfactual methods from finding reasonable potential relations.

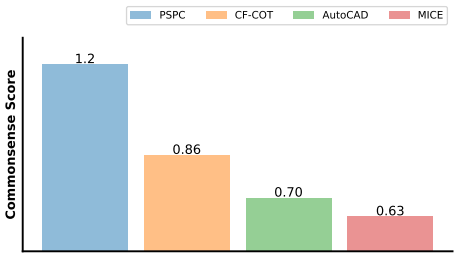

Figure 4: Results for human evaluation.

4.3 Results for Human Evaluation

To evaluate whether the generated counterfactuals conform to commonsense, we conduct the human study. The average scores by the aforementioned three annotators are calculated as the final commonsense score for each method. The results are shown in Fig. 4. We have the following observations.

(1) Our proposed PSPC method can discover po-

[7] The detailed comparison is provided in Appendix A.4.

Case 1	CSU Stanislaus students take **complaints** **to** President's **door** .	*Entity-Destination*
MICE	CSU Stanislaus students take **complaints** **caused** the President's **door** . ✘	*Cause-Effect*
AutoCAD	CSU Stanislaus students take **complaints** **caused** the President's **door** . ✘	*Cause-Effect*
CF-CoT	CSU Stanislaus students take **complaints** **are placed in** President's **door** . ✘	*Content-Container*
PSPC	CSU Stanislaus students take **complaints** **about** President's **door** . ✔	*Message-Topic*
Case 2	The final **chapter** **offers** a **theological survey** of the use of the formula.	*Message-Topic*
MICE	The final **chapter** **for** a **theological survey** of the use of the formula. ✘	*Instrument-Agency*
AutoCAD	The final **chapter** **into** a **theological survey** of the use of the formula. ✘	*Entity-Destination*
CF-CoT	The final **chapter** **reflects** a **theological survey** of the use of the formula. ✘	*Topic-Message*
PSPC	The final **chapter** is **part of** a **theological survey** of the use of the formula. ✔	*Component-Whole*

Table 5: Instances for case study. Entities are in **orange** and causal terms are in **blue** . ✔ denotes that the generated counterfactuals comply with the commonsense requirements and ✘ is the opposite.

tential relations that conform the best to commonsense. As can be seen, PSPC is the only method getting a score higher than 1, i.e., beyond the level of 'marginally reasonable'.

(2) The LLM-based model CF-CoT still lacks the ability to generate commonsense counterfactuals. The SLM-based models MICE and AutoCAD perform the worst in this experiment since their post-processing filtering mechanism cannot provide a sufficient commonsense guarantee during generation.

4.4 Empower SLM based Methods with PSPC

To further validate the rationality of the potential relations generated by our PSPC method, we design a synthetic experiment by examining its relations' impact on SLM based methods. Specifically, we first obtain the potential relations from PSPC, and we then let MICE and AutoCAD conduct controlled generation of the causal term using GPT-2 as the editor without applying their filtering strategies, e.g., *eggs were (entity-origin) <mask> a box* where the *'entity-origin'* relation is provided by PSPC. The results are shown in Table 4, from which we can draw the following conclusions.

- The relations from PSPC can provide effective guidance for current SLM methods. Both MICE and AutoCAD get remarkable improvements under all settings, and the enhancements are particularly pronounced on sparse data. This proves that the SLM based counterfactual methods can overcome their dependency on filtering mechanisms as long as they get proper potential relations.

- Small language models can be fine-tuned with flexibility, and thus they are more adept at addressing various issues in downstream tasks. Meanwhile, large models possess knowledge that small models lack. Therefore, the cooperation between a SLM based method and our PS and PC functions might be a promising way. For example, on the setting of 5%-10%, AutoCAD w/ PSPC outperforms PSPC using GPT-3.5 (the results in Table 2).

4.5 Case Study

We randomly select two samples from SemEval to have a close look. Table 5 shows the original and augmented instances by different methods. Clearly, our counterfactuals conform to commonsense and align well with labels. In contrast, the compared methods may produce unreasonable samples and relations. For example, in Case 1, *complaints* cannot '*caused*' the *door* for MICE and AutoCAD. In Case 2, there seems to be a very low probability for CF-CoT to form the *Topic-Message* relation between '*chapter*' and '*theological survey*'.

5 Conclusion

In this paper, inspired by cognitive neuroscience, we propose a novel neuromorphic mechanism to align LLMs with humans during the experience gaining process in counterfactual generation for relation extraction. We realize this mechanism with the pattern separation and pattern completion functions which first decompose entities into attributes and then combine the attributes to reconstruct a scenario. Our method enables the retrieval of entities' scenarios from the model's extensive memory, and thus provides factual basis for the relation between entities of the generated counterfactual.

Extensive experiments prove the effectiveness of our proposed mechanism and its implemented functions. We believe this is an interesting exploration of neuroscience based researches and wish it can inspire more studies along this direction.

Limitations

Despite the episodic memory retrieval (EMR) mechanism along with the pattern separation and pattern completion (PSPC) functions can effectively enhance the conformity to commonsense for the generated RE counterfactuals by LLMs, this mechanism relies on the vast knowledge related to episodic memories within LLMs and the prominent reasoning and planning capabilities of these models.

Ethics Statement

Our work aims to explore the commonsense counterfactual generation, which is entirely at the methodological level and therefore does not have any negative social impact.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China (NSFC) project (No. 62276193).

References

Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. 2022. Docogen: Domain counterfactual generation for low resource domain adaptation. *arXiv preprint arXiv:2202.12350*.

Hao Chen, Rui Xia, and Jianfei Yu. 2021. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 269–278.

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

Mor Geva, Tomer Wolfson, and Jonathan Berant. 2022. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. *Transactions of the Association for Computational Linguistics*, 10:111–126.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description. *ACE*, 5.

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

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models, july 2022b. *URL http://arxiv.org/abs/2207.05608*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

James J Knierim and Joshua P Neunuebel. 2016. Tracking the flow of hippocampal computation: Pattern separation, pattern completion, and attractor dynamics. *Neurobiology of learning and memory*, 129:38–49.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. 2016. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*.

Wanli Li, Tieyun Qian, Ming Zhong, and Xu Chen. 2022. Interactive lexical and semantic graphs for semisupervised relation extraction. *IEEE Transactions on Neural Networks and Learning Systems*.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. 2023b. Large language models as counterfactual generator: Strengths and weaknesses. *arXiv preprint arXiv:2305.14791*.

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Xin Miao, Yongqi Li, and Tieyun Qian. 2023. Generating commonsense counterfactuals for stable relation extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5654–5668.

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

Guoshun Nan, Jiaqi Zeng, Rui Qiao, Zhijiang Guo, and Wei Lu. 2021. Uncovering main causalities for long-tailed information extraction. *arXiv preprint arXiv:2109.05213*.

Chi T Ngo, Sebastian Michelmann, Ingrid R Olson, and Nora S Newcombe. 2021. Pattern separation and pattern completion: Behaviorally separable processes? *Memory & Cognition*, 49:193–205.

Angela Nyhout and Patricia A Ganea. 2019. The development of the counterfactual imagination. *Child Development Perspectives*, 13(4):254–259.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *https://openai.com/blog/ chatgpt/*.

OpenAI. 2023. Gpt-4 technical report. *https://arxiv.org/pdf/2303.08774.pdf*.

Judea Pearl. 2009. Causal inference in statistics: An overview.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

Jingyuan Qi, Zhiyang Xu, Ying Shen, Minqian Liu, Di Jin, Qifan Wang, and Lifu Huang. 2023. The art of socratic questioning: Recursive thinking with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4177–4199.

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

Marcel Robeer, Floris Bex, and Ad Feelders. 2021. Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3611–3625.

Edmund T Rolls. 2013. The mechanisms for pattern completion and pattern separation in the hippocampus. *Frontiers in systems neuroscience*, 7:74.

Alexis Ross, Ana Marasović, and Matthew E Peters. 2021. Explaining nlp models via minimal contrastive editing (mice). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852.

Indira Sen, Mattia Samory, Fabian Flöck, Claudia Wagner, and Isabelle Augenstein. 2021. How does counterfactually augmented data impact models for social computing constructs? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 325–344.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Marcos Treviso, Alexis Ross, Nuno M Guerreiro, and André FT Martins. 2023. Crest: A joint framework for rationalization and counterfactual text generation. *arXiv preprint arXiv:2305.17075*.

Endel Tulving. 2002. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1):1–25.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. *arXiv preprint arXiv:2205.03784*.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14024–14031.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. Autocad: Automatically generating counterfactuals for mitigating shortcut learning. *arXiv preprint arXiv:2211.16202*.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.

Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. *arXiv preprint arXiv:2106.15231*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.(may 2023). *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Mi Zhang, Tieyun Qian, Ting Zhang, and Xin Miao. 2023. Towards model robustness: Generating contextual counterfactuals for entities in relation extraction. In *Proceedings of the ACM Web Conference 2023*, pages 1832–1842.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

A Implementation Details

A.1 Datasets

Low-resource settings. We utilize two of the most popular RE datasets for low-resource settings, including SemEval (Hendrickx et al., 2019) and TACRED (Zhang et al., 2017). SemEval comprises 9 bidirectional relations and 1 *other* class. TACRED contains 41 relation types and 1 *no_relation* class. Due to their belonging to in-domain datasets (Zhang et al., 2023), they satisfy the independent and identically distributed (i.i.d.) assumption. Meanwhile, the ratio of the training set to the test set is significantly larger than 1 (SemEval at 2.65, and TACRED at 3.01), resulting in a high likelihood of spurious correlations in the test set being present in the training set. In this situation, the spurious correlations can assist the model in finding shortcuts and improving accuracy (Sen et al., 2021). For example, the model relies on non-causal words that appear in both the training and test sets for association. Unfortunately, when the counterfactuals block spurious correlations, they may not help the model in terms of accuracy and could even have a counterproductive effect under such an i.i.d. scenario (Kaushik et al., 2019; Sen et al., 2021; Wang and Culotta, 2021; Geva et al., 2022). To accurately validate the effects of generated counterfactuals, we introduce the continuous low-proportion low-resource settings (Li et al., 2022; Miao et al., 2023), including 3%, 5%, 7%, and 10%. For example, we randomly select 3% instances from the training set. Such low-resource settings can provide two assurances: (1) The issue of spurious correlations in low-resource settings is significant. (Nan et al., 2021), ensuring that there is room for counterfactuals to be effective. (2) The overlap of spurious correlations between the training and test sets is relatively fewer, ensuring that eliminating spurious correlations will not bring negative consequences. Therefore, low-resource settings are suitable for evaluating the effectiveness of counterfactuals (Miao et al., 2023).

Out-of-domain (OOD) settings. The spurious correlations under this scenario can be defined as correlations between domain-specific words and labels (Calderon et al., 2022). Such correlations cannot assist the model in establishing associations between cross-domains and may even have a counterproductive effect. Therefore, OOD evaluation can be used to validate the effectiveness of counterfactuals in mitigating such spurious correlations. The most popular OOD dataset in RE is ACE 2005 (Grishman et al., 2005), which contains 6 relatively coarse-grained relation types. In line with previous work (Miao et al., 2023), we select four sub-datasets from different domains, including weblogs (WL), broadcast conversation (BC), broadcast news (BN), and newswire (NW). Specifically, we employ the WL sub-dataset as the training set and other sub-datasets as test sets, respectively, which can be formalized as WL→BC, WL→BN, and WL→NW.

A.2 Rules of Human Evaluation

To provide an intuitive explanation, we utilize examples for illustration. Suppose the original example is *cheese is flowed into food banks*. The relation between *cheese* and *food banks* is *entity-destination*, depending on the causal term *flowed into*. The generated counterfactuals are scored based on the following rules.

(1) **Not Matching**: If the flipped relation does not match the replaced causal term, it should be given a score of 0. For example, the counterfactual of *cheese is donated to food banks*, with the new relation *content-container*. The *content-container* relation cannot be triggered by the causal term *donated to*. Such an erroneous example introduces noise and negatively affects the model training.

(2) **Not Reasonable**: If the flipped relation contradicts the entities in terms of commonsense, it should be given a score of 0. For example, the counterfactual of *cheese is caused by food banks*, with the new relation *effect-cause*. In our understanding, there is no causal relationship between *cheese* and *food banks*, hence it is not reasonable.

(3) **Marginally Reasonable**: If the flipped relation is somewhat associated with the properties of the entities but not entirely accurate, it should be given a score of 1. For example, the counterfactual of *cheese is contained in food banks*, with the new relation *content-container*. Although the *food banks* can be regarded as a container for storing *food*, it is not commonly expressed in that way.

(4) **Reasonable**: If the flipped relation aligns with the properties of the entities, it should be given a score of 2. For example, the counterfactual of *cheese is from food banks*, with the new relation *entity-origin*. Individuals can receive *food* from the *food banks*, hence the counterfactual is reasonable.

A.3 Baselines

We provide implementation details of the baselines, including conventional methods:

- SYNONYM REPLACEMENT (Zhang et al., 2015) is a synonym substitution based method that replaces 30% words in a sentence with their synonyms from WordNet (Miller, 1995).
- BACK TRANSLATION (Sennrich et al., 2015) is a translation based method that first translates sentences into another language and then back-translates them to the original language.
- BERT-MLM (Jiao et al., 2019) is a BERT-based (Devlin et al., 2018) substitution method that first masks 30% context words (excluding entities and causal terms) with `[MASK]`, then fills in the different top words predicted by BERT.

Small language model (SLM) based counterfactual generation methods:

- MICE (Ross et al., 2021) first identifies causal terms based on the gradient contributions obtained from the post-enhanced RE base model. Then it employs a trained editor to replace the identified causal terms with proper words to flip the original relation to the potential one.
- AUTOCAD (Wen et al., 2022) is similar to MICE, but it introduces the unlikelihood strategy (Welleck et al., 2019) for the editor, to ensure the editor does not generate the causal terms consistent with the original relation.
- COCO (Zhang et al., 2023) exploits syntactic and semantic dependency graphs to discover substitutable causal terms from other sentences with different relations and substitutes the original relation simultaneously.

The employed editors for the above methods are all uniformly based on GPT-2 (Radford et al., 2019). The large language model (LLM) based counterfactual generation method:

- CF-COT (Li et al., 2023b) requests LLMs to generate RE counterfactuals through the counterfactual CoT based on the standard pipeline. Compared with our method, the main distinction lies in its absence of the PSPC approach while keeping the rest consistent.

A.4 PSPC Implementations

The complete implementation process of PSPC is illustrated in Table 9. To ensure that LLMs fully comprehend our intentions, we first provide the task definition and instructions before in-context examples. The instruction aligns with Section 3.1, breaking down the overall process into six steps: (1) causal term identification, (2) entity deconstruction, (3) scenario reconstruction, (4) decisive scenario identification, (5) potential relation uncovering, and (6) causal term replacement. For each step, the instruction provides a detailed explanation. Afterward, we provide all the candidate relations depending on the given dataset for LLMs to select from. Most importantly, we design two in-context examples based on a fabricated instance. For SemEval, the fabricated instance is defined as: *eggs are moved into a box*. Note that we only need to fabricate a suitable instance for each dataset.

In example 1, we steer the operation process of LLMs explicitly step by step. After causal term identification, we deconstruct the attributes of *eggs* and *box* into three aspects, i.e., *form*, *usage*, and *purpose*. For example, *eggs* is decomposed as: *form: fragile entity*, *usage: as food*, and *purpose: farm product*, as shown in the blue spans of Table 9. These three aspects are referenced from ConceptNet (Speer et al., 2017), possessing broad generality among entities. Therefore, they are applied to all datasets. Benefiting from LLMs' extensive knowledge and reasoning capabilities, even when encountering unexpected entities, they can adapt by replacing the defined aspects with other appropriate aspects. For example, LLMs deconstruct *farmer* into the entity-adaptive aspects: *occupation: agricultural worker, action: erecting*, as shown in Table 6. LLMs can ultimately generate commonsense counterfactuals by adjusting the types and quantity of aspects.

Subsequently, we provided a detailed description of the scenario reconstruction process, as shown in the orange spans of Table 9. Specifically, we pair the decomposed attributes between entities and form three plausible scenarios. To construct scenarios that fit the attributes of the entities, we utilize chain-of-thought (CoT) (Wei et al., 2022) technology to deduce suitable scenarios starting from the attributes. For example, we pair the property *form: fragile entity* of *eggs* and the property *usage: package something* of *box*, then we can infer the reasonable scenario: *protective purpose:*

Input: <e1> farmer </e1> erected the <e2> disguise </e2>
Entities: farmer, disguise
Relation: producer-product
Identify causal term: the context word "erected" is causally related to the relation "producer-product".
Deconstruct entities: based on the attributes of the entities, "farmer" can be deconstructed into the several primary properties: occupation: agricultural worker , action: erecting ; "disguise" can be deconstructed into the several primary properties: form: a disguise , usage: conceal identity .
.....
Output: <e1> farmer </e1> uses the <e2> disguise </e2>
New Relation: agency-instrument

Input: <e1> engine </e1> powered a limited-production Mustang <e2> model </e2>
Entities: engine, model
Relation: component-whole
Identify causal term: the context word "powered" is causally related to the relation "component-whole".
Deconstruct entities: based on the attributes of the entities, "engine" can be deconstructed into the several primary properties: function: provide power , usage: in a vehicle , purpose: generate motion ; "model" can be deconstructed into the several primary properties: type: limited-production , usage: as a car , purpose: represent a specific design .
.....
Output: <e1> engine </e1> is used by a limited-production Mustang <e2> model </e2>
New Relation: instrument-agency

Table 6: The examples of aspect adaption in LLMs. The blue spans denote our defined aspects. The orange spans denote the aspects modified by LLMs to adapt to entities.

put eggs into a box, based on the analysis that eggs are fragile and the box can provide protection to prevent them from breaking. We do not specify nor can we specify the matching order of attributes, but the plausible scenarios themselves serve as target constraints. LLMs can autonomously decide on pairing schemes based on the situation, as long as they can form plausible scenarios. Additionally, to balance computational complexity and diversity, we set the number of reconstructed scenarios to 3.

Afterward, to ensure that the reconstructed scenarios are relevant to the instance yet imply different relations, we introduce decisive scenario identification and potential relation uncovering, as shown in the green spans of Table 9. We stipulate that LLMs must construct a scenario that adapts to the original instance and identify it. This constraint can guide pattern separation (PS) and pattern completion (PC) towards instance-relevant directions. After excluding this anchor scenario, we require LLMs to focus on another specific promising scenario, making it more likely to uncover a commonsense relation. We demonstrate the effectiveness of this strategy in supplementary experiments. Finally, we request LLMs to replace the identified causal term and satisfy the new relation.

To avoid the singularity in scenario selection, such as only selecting the second scenario, we introduce Example 2 to choose another scenario outside of the decisive scenario, while keeping the previous content consistent. Additionally, multiple examples can serve to standardize the format. In the counterfactual generation phase, we concatenate the input instance's sentence, entities, and relation after the in-context examples. LLMs will output the processed sentence and new relation based on the defined process above.

To ensure a fair comparison with CF-CoT (Li et al., 2023b), in the specific implementations, we only removed the pattern separation (PS) and pattern completion (PC) related processes while keeping other contents unchanged, as shown in Table 10. Furthermore, we provide a detailed example contrasting the differences in inference processes between PSPC and CF-COT, as shown in Table 11. In the inference process of CF-CoT, LLMs take a shortcut by simply reversing the relation. Due to the strong semantic correlation between reversed relations, LLMs fail to perceive the subtle differences, resulting in the incorrect causal term replacement that does not match the relation. While in the inference process of PSPC, under the guidance of reconstructed scenarios, LLMs uncover the potential relation that satisfies entity properties and generate a commonsense counterfactual.

B Supplementary Experiments

B.1 PSPC Parameter Analysis

B.1.1 Aspect Quantity Analysis

To study how the quantity of aspects affects experimental performance, we gradually increase the de-

Deconstruct entities: Based on the attributes of the entities, "eggs" can be deconstructed into several primary properties: form: fragile entity , usage: as food , purpose: farm product , function: hatch chick , craft: make handcraft ; "box" can be deconstructed into several primary properties: form: a container , usage: package something , purpose: preserve products , function: keep dry , craft: organize items.
Reconstruct the first scenario: Based on the "form: fragile entity" property of "eggs" and the "usage: package something" property of "box", "eggs" should be moved into a "box" to prevent them from breaking, hence we can reconstruct the scenario: protective purpose: put eggs into a box. **Reconstruct the second scenario**: Based on the "purpose: farm product" property of "eggs" and the "form: a container" property of "box", "eggs" should be contained in a "box" for preservation, hence we can reconstruct the scenario: preservation purpose: eggs are kept in a box. **Reconstruct the third scenario**: Based on the "usage: as food" property of "eggs" and the "purpose: preserve product" property of "box", "eggs" should be taken out from the "box" before making food, hence we can reconstruct the scenario: consumption purpose: take out eggs from the box. **Reconstruct the fourth scenario**: Based on the "function: hatch chick" property of "eggs" and the "function: keep dry" property of "box", "eggs" should be saved in "box" for keeping dry to hatch chicks, hence we can reconstruct the scenario: hatching purpose: eggs are saved in a box. **Reconstruct the fifth scenario**: based on the "craft: make handcraft" property of "eggs" and the "craft: organize items" property of "box", "eggs" should be taken out from the storage "box" before making handcrafts, hence we can reconstruct the scenario: production purpose: eggs are taken from the box.

Table 7: The pattern separation (PS) and pattern completion (PC) processes of the full set of aspects. The blue spans represent the defined aspects. The orange spans represent the reconstructed scenarios.

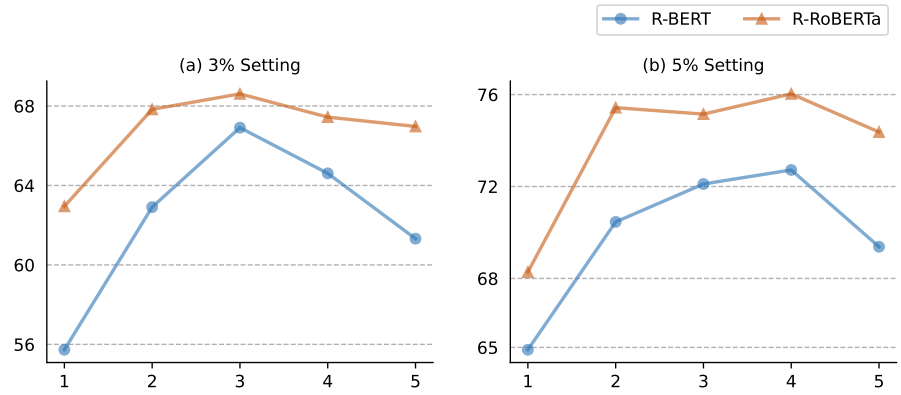

Figure 5: The F1-score performance of aspect quantity analysis in SemEval. The horizontal axis represents the number of aspects or reconstructed scenarios in in-context examples.

fined aspect in in-context examples. To validate the trend thoroughly, we supply two more aspects after the three already-defined aspects in Table 9. We demonstrate the pattern separation (PS) and pattern completion (PC) processes of the full set of aspects in Table 7. In the analysis process, we sequentially add the quantity of aspects, corresponding to an increase in the number of reconstructed scenarios. The trend of experimental performance is shown in Figure 5. The trend shown in the line graph indicates that as the number of aspects increases, the performance of LLMs initially rises gradually, but after a certain point, it starts to decline. Various aspects can provide a more comprehensive description of entities, but there may also be redundancy among aspects. Too many aspects can have the opposite effect. For example, there is a clear overlap between the *usage* aspect and the *function* aspect. Therefore, when defining aspects, both diversity and independence should be considered.

B.1.2 Instance Quantity Analysis

To study how the quantity of in-context instances affects experimental performance, we gradually increase the instances in in-context examples. In addition to the instance we have already demonstrated in Table 9, we additionally design two instances with different relations. The pattern separation (PS) and pattern completion (PC) processes of these two extra instances are shown in Figure 6. Following the implementations of Section A.4, each instance corresponds to two examples used to select the

Instance 1:
Input: <e1> man </e1> establishes the <e2> company </e2>
Entities: man, company
Relation: producer-product
Deconstruct entities: Based on the attributes of the entities, "man" can be deconstructed into several primary properties: form: social member , usage: use tools , purpose: create value ; "company" can be deconstructed into the several primary properties: form: an organization , usage: produce product , purpose: make money.
Reconstruct the first scenario: Based on the "form: social member" property of "man" and the "form: an organization" property of "company", "man" can be employed to become a member of "company", hence we can reconstruct the scenario: employment purpose: man become a member of a company.
Reconstruct the second scenario: Based on the "usage: use tools" property of "man" and the "usage: produce product" property of "company", "man" can utilize "company" to produce a product, hence we can reconstruct the scenario: utilization purpose: man utilizes a company.
Reconstruct the third scenario: Based on the "purpose: create value" property of "man" and the "purpose: make money" property of "company", "man" can establish "company" to make money to create value, hence we can reconstruct the scenario: establishment purpose: man establish a company.

Instance 2:
Input: <e1> accident </e1> is cased by the <e2> rumor </e2>
Entities: accident, rumor
Relation: effect-cause
Deconstruct entities: Based on the attributes of the entities, "accident" can be deconstructed into several primary properties: form: unexpected event , usage: cause injury , purpose: alert people ; "rumor" can be deconstructed into the several primary properties: form: unverified report , usage: incite emotion , purpose: spread information.
Reconstruct the first scenario: Based on the "form: unexpected event" property of "accident" and the "purpose: spread information" property of "rumor", "accident" can be integrated into "rumor" for spreading the event, hence we can reconstruct the scenario: integration purpose: accident is a part of the rumor.
Reconstruct the second scenario: Based on the "usage: cause injury" property of "accident" and the "usage: incite emotion" property of "rumor", "accident" can be triggered by "rumor" inciting emotion, hence we can reconstruct the scenario: incitement purpose: accident is triggered by rumor.
Reconstruct the third scenario: Based on the "purpose: alert people" property of "accident" and the "form: unverified report" property of "rumor", "accident" can be described in "rumor" report to alert people, hence we can reconstruct the scenario: warning purpose: accident is described in rumor.

Table 8: The pattern separation (PS) and pattern completion (PC) processes of two additional instances. The blue spans represent the deconstructed properties. The orange spans represent the reconstructed scenarios.

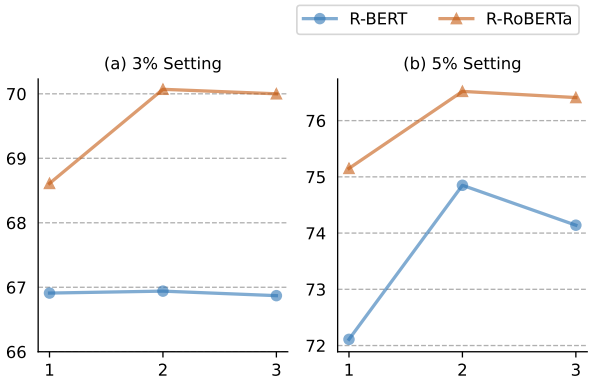

Figure 6: The F1-score performance of instance quantity analysis in SemEval. The horizontal axis represents the number of instances in in-context examples.

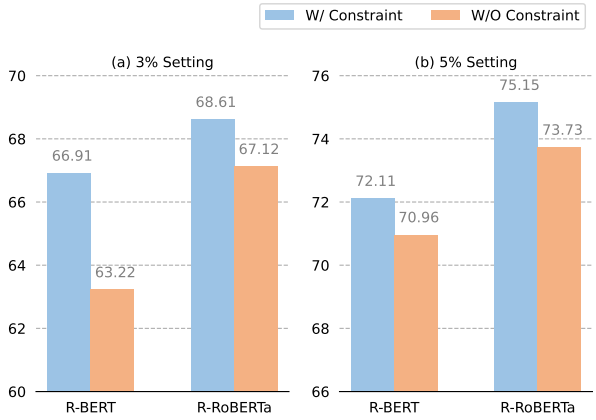

Figure 7: The F1-score performance of instance constraint analysis in SemEval.

other two scenarios apart from the decisive scenario, as shown in Table 9. During the analysis process, we sequentially superimpose instances in the in-context examples, and the experimental results are shown in Figure 6. When adding the second instance, LLMs show a significant improvement in most cases, but when adding the third instance, the improvement diminishes. The trend shown in the line graph illustrates that diverse instances can enhance the performance of our PSPC approach

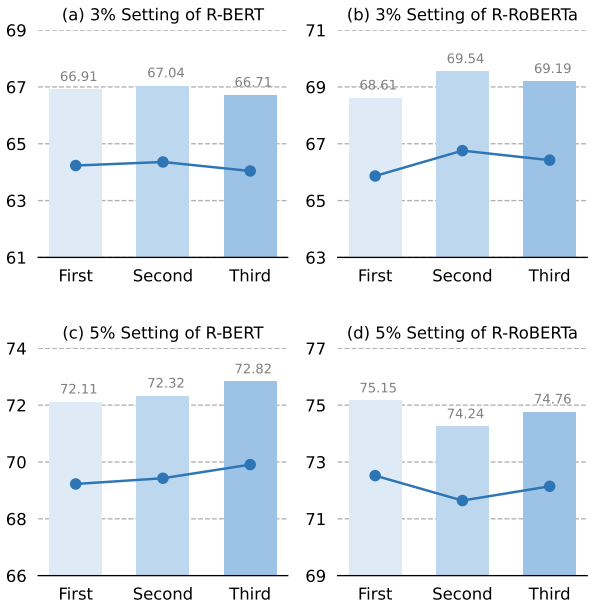

Figure 8: The F1-score of aspect sensitivity analysis in SemEval. The lines in the graph indicate the result fluctuation between each group.

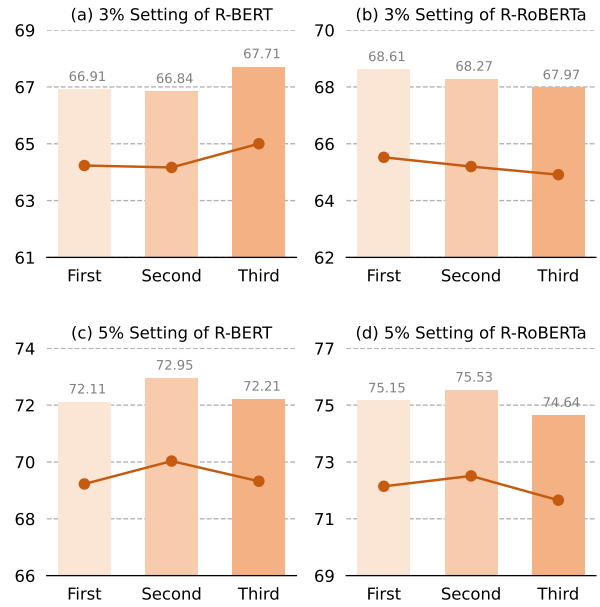

Figure 9: The F1-score of expression sensitivity analysis in SemEval. The lines in the graph indicate the result fluctuation between different editing versions.

in LLMs, but this improvement may not be continuous. Meanwhile, the increase in instances will significantly increase computational costs.

B.1.3 Instance Constraint Analysis

To validate the effectiveness of the instance constraint, which is the scenario reconstruction corresponding to the original instance, as shown in the green spans of Table 9, we conduct a comparative analysis. Specifically, we replace the decisive scenario with another plausible one that does not align with the original relation and remove the process of decisive scenario identification. The results are shown in Figure 7. We can observe that without this strategy, the performance of LLMs significantly decreases. Through the observations of the generated data, we find that the absence of this strategy results in a significant quantity decrease in the valid counterfactuals generated by LLMs. The main reason is the lack of an exclusion process; LLMs tend to choose the original relation as the potential one.

B.2 PSPC Sensitivity Analysis

To test the sensitivity of our PSPC method to implementation details, we analyze it in three respects: vocabulary, sentences, and examples. Ultimately, the experimental results show that it maintains stable and outstanding performance across different implementation approaches, which demonstrates that the effectiveness of our method stems from the strategy rather than implementation bias.

B.2.1 Aspect Sensitivity Analysis

To test the sensitivity of aspect terms, we modified the examples by replacing the aspect terms with their synonyms[8]. We select two synonyms for each aspect term, *appearance* and *structure* for the *form* aspect, *application* and *convention* for the *usage* aspect, *intent* and *target* for the *purpose* aspect. Then, we divide these aspect terms into three groups. The first group consists of the original aspect terms: *form*, *usage*, and *purpose*. The second group consists of the first synonym for each aspect: *appearance*, *application*, and *intent*. The third group consists of the second synonym for each aspect: *structure*, *convention*, and *target*. The experimental results for each group are shown in Figure 8. In all configurations, LLMs exhibit only minor fluctuations under the guidance of our PSPC method. The maximum decrease is only 0.91, and in most cases, there is still room for improvement. This suggests that our method is not sensitive to the choice of aspect terms.

B.2.2 Expression Sensitivity Analysis

To test the sensitivity to sentence-level expressions, we employ ChatGPT (OpenAI, 2022) to rewrite all sentences in Table 9. Specifically, while maintaining the format unchanged, we instruct ChatGPT to rewrite all sentences while ensuring semantic consistency. We conducted two times of rewrites. For

[8] https://www.thesaurus.com/browse/synonym

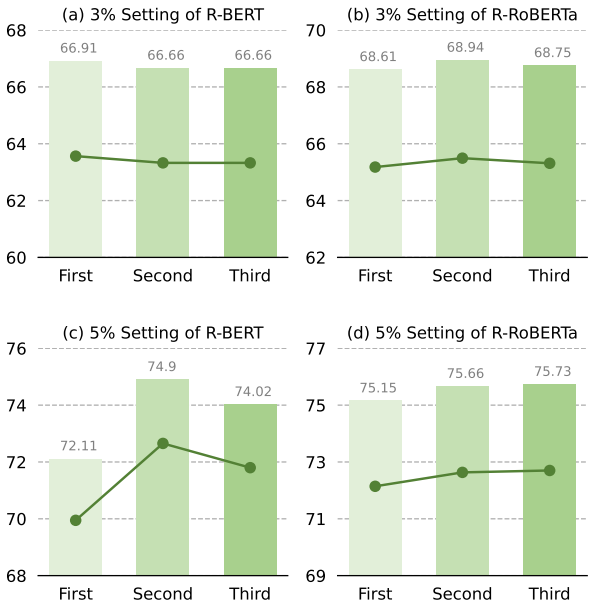

Figure 10: The F1-score of instance sensitivity analysis in SemEval. The lines in the graph indicate the result fluctuation between different instances.

clarity, the initial state is labeled as the first state, and the two rewrites are labeled as the second state and the third state, respectively. The experimental results are shown in Figure 9. Across all settings, LLMs exhibit minor fluctuations. The maximum decrease is just 0.64. This indicates that our method is insensitive to sentence expression.

B.2.3 Instance Sensitivity Analysis

To test the sensitivity of instance selection in in-context examples, we replace the original instance with the two additional ones shown in Table 8 respectively. For clarity, we label the original instance as the first instance, and the other two instances are labeled as the second and third instances respectively. The results of the comparative experiment are shown in Figure 10. The line chart indicates that LLMs generally exhibit relatively small fluctuations in most cases. The maximum decrease is only 0.25, which indicates that LLMs demonstrate overall stability. The scenario with relatively large fluctuations is the 5% setting of R-BERT, but these fluctuations represent growth. This indicates that we have not yet found the most suitable instances, and there is still significant room for improvement.

B.3 Version Effectiveness Analysis

To verify the effectiveness of our PSPC method across different versions of LLMs, we conduct tests on various released versions of GPT-3.5 Turbo. In addition to the *gpt-3.5-turbo-0613* version used in the experiments above, the additional versions we considered are listed below in chronological order:

- The *gpt-3.5-turbo-0301* is a snapshot of *gpt-3.5-turbo* from March 1th 2023. It is an early version of the model with relatively weak inference and alignment capabilities.
- The *gpt-3.5-turbo-instruct* has similar capabilities as GPT-3 era models. It is a versatile and powerful tool that has significant potential to transform professional workflows.
- The *gpt-3.5-turbo-1106* is the GPT-3.5 Turbo model with improved instruction following, JSON mode, reproducible outputs, parallel function calling, and more.
- The *gpt-3.5-turbo-0125* is the latest GPT-3.5 Turbo model with higher accuracy at responding in requested formats and a fix for a bug that caused a text encoding issue for non-English language function calls.

We directly applied the content of Table 9 to request the above-mentioned models, and the experimental results are shown in Figure 11. Through observation, we can draw the following conclusions: (1) Although no adjustments were made for each version, our PSPC method presents significant improvement across all released versions. This sufficiently demonstrates the effectiveness of our pattern separation (PS) and pattern completion (PC) strategies. (2) With version updates, GPT's counterfactual reasoning abilities gradually improve. Nonetheless, our PSPC approach still enables the model to generate higher-quality RE counterfactuals. (3) From the performance of the last two updates, it appears that GPT's counterfactual reasoning capability may have reached a bottleneck.

C Causal Perspective Analysis

To gain a more intuitive understanding of the importance of counterfactuals and commonsense counterfactuals, we first theoretically analyze the problem addressed by counterfactuals through a structural causal model (SCM) (Pearl and Mackenzie, 2018). Subsequently, through further causal analysis, we explain why counterfactuals need to align with commonsense and what problems may arise if they violate the commonsense constraint.

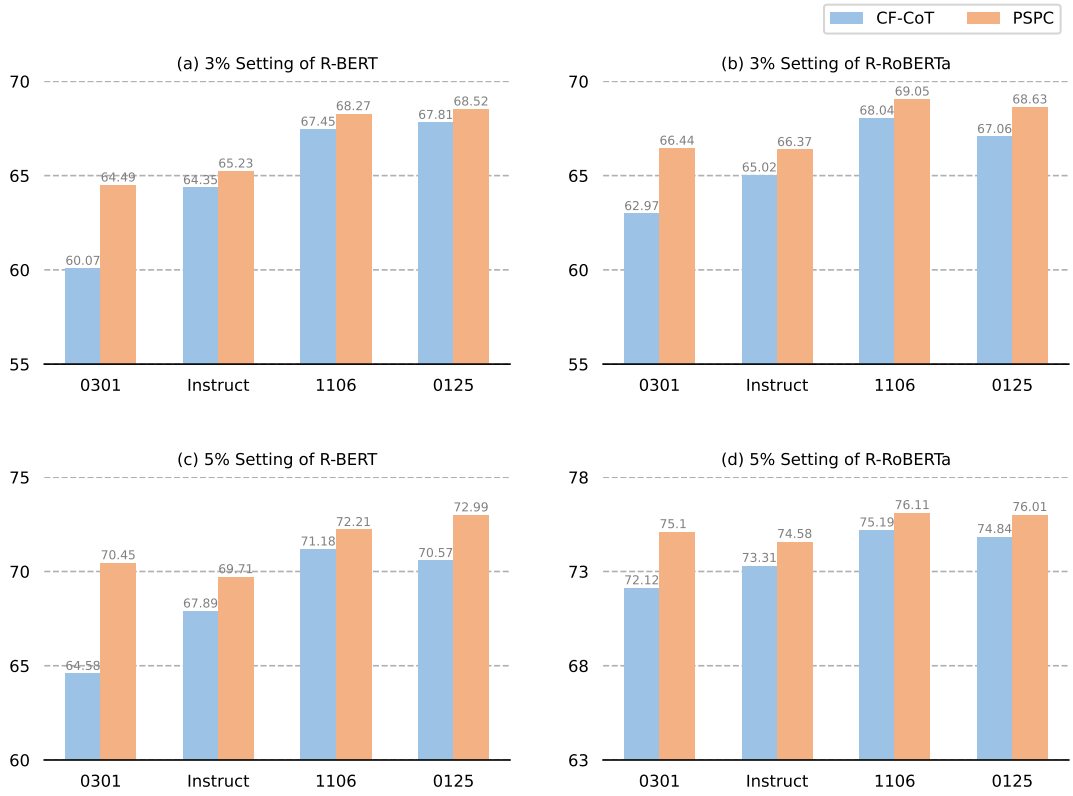

Figure 11: The F1-score of version effectiveness analysis in SemEval. For simplicity, we use name suffixes to identify each model. For example, the "0301" corresponds to *gpt-3.5-turbo-0301*.

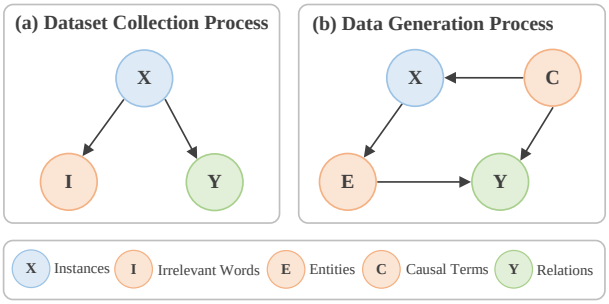

Figure 12: The structural causal model (SCM) of (a) dataset collection process and (b) counterfactual generation process. Nodes in the graph represent variables, and edges represent causal relationships between them.

Due to biases and limitations in dataset collection, neural networks are inevitably influenced by spurious correlations. The collection process can be illustrated as Fig. 12 (a), where each instance is annotated to acquire a relation ($X \rightarrow Y$) and contains several task-irrelevant words ($X \rightarrow I$). Although there is no causal relationship between task-irrelevant words and relations ($I \not\rightarrow Y$), the instances serve as a common cause ($I \leftarrow X \rightarrow Y$), resulting in a statistically spurious correlation between them. In the example of *eggs were removed into a box*, the correlation between word *were* or *a* and relation *entity-destination* is a spurious correlation. Counterfactuals emphasize decision boundaries by flipping the relations through replacing causal terms (Treviso et al., 2023), thus mitigating the influence of irrelevant words.

The RE counterfactual generation process can be formalized as Fig. 12 (b), where new instances are generated by causal term replacement ($C \rightarrow X$) and contain original entities ($X \rightarrow E$). Furthermore, causal terms and entities jointly determine the instances' relations ($C \rightarrow Y \leftarrow E$). If we focus on the effects between entities and relations, there exists a backdoor path containing two confounders ($E \leftarrow X \leftarrow C \rightarrow Y$). In an ideal state, we only seek to model their direct causal relationships ($E \rightarrow Y$), without being influenced by the confounders, which can be formalized as:

$$P(Y|E) = P(Y|do(E)), \tag{10}$$

where do represents the do-operator. Fortunately, we can implement the do-operator by back-door adjustment (Pearl, 2009), which can be represented as:

$$P(Y|do(E)) = \sum_{X} P(Y|E, X = x)P(X = x), \tag{11}$$

since X is a intermediate variable of C, the formula can be transformed into:

$$P(Y|do(E)) = \sum_{C} P(Y|E, C = c)P(C = c), \tag{12}$$

where $P(Y|E, C = c)$ represents that the relation distribution is the joint probability distribution of entities and causal terms. In simple terms, this formula only holds when considering the commonsense constraint on entities, as we previously announced. If this constraint is violated, the process is transformed into:

$$\begin{aligned} \hat{P}(Y|do(E)) &= \sum_{C} P(Y|C = c)P(C = c), \\ &= \sum_{C} P(Y, C = c), \end{aligned} \tag{13}$$

where $\hat{P}(Y|do(E))$ is only related to C. Therefore, counterfactuals that violate the commonsense constraint will ultimately render entities ineffective, while relations between entities should depend on the semantics of causal terms and the entities themselves (Wang et al., 2022). For example, in the erroneous counterfactual of *eggs were produced by a box*, relation *product-producer* only depends on causal term *produced by*, without considering entity *box* cannot be a *producer* in a general context.

Task definition: Change the relation between entities based on minimal context editing.

Instruction: The process can be divided into the following six steps. (1) Identify causal term: Find the context words that are causally related to the relation. (2) Deconstruct entities: Based on the attributes of the entities, deconstruct the entities into several primary properties. (3) Reconstruct scenarios: Based on the deconstructed properties, reconstruct the scenarios that the entities may constitute. (4) Identify the decisive scenario: Select a scenario that contributes the most to the relation. (5) Uncover potential relation: Select another promising scenario and match the most suitable relation from the candidate relations. (6) Replace causal term: Replace the identified causal term with suitable words to change the original relation to the potential one.
Candidate relations: message-topic, topic-message, entity-origin, origin-entity, entity-destination, destination-entity, content-container, container-content, cause-effect, effect-cause, component-whole, whole-component, member-collection, collection-member, instrument-agency, agency-instrument, product-producer, producer-product.

Example 1:
Input: <e1> eggs </e1> are moved into a <e2> box </e2>
Entities: eggs, box
Relation: entity-destination
Identify causal term: The context words "moved into" are causally related to the relation "entity-destination".
Deconstruct entities: Based on the attributes of the entities, "eggs" can be deconstructed into several primary properties: form: fragile entity, usage: as food, purpose: farm product; "box" can be deconstructed into several primary properties: form: a container, usage: package something, purpose: preserve products.
Pair the properties between "eggs" and "box" to reconstruct three reasonable scenarios:
Reconstruct the first scenario: Based on the "form: fragile entity" property of "eggs" and the "usage: package something" property of "box", "eggs" should be moved into a "box" to prevent them from breaking, hence we can reconstruct the scenario: protective purpose: put eggs into a box.
Reconstruct the second scenario: Based on the "purpose: farm product" property of "eggs" and the "form: a container" property of "box", "eggs" should be contained in a "box" for preservation, hence we can reconstruct the scenario: preservation purpose: eggs are kept in a box.
Reconstruct the third scenario: Based on the "usage: as food" property of "eggs" and the "purpose: preserve product" property of "box", "eggs" should be taken out from the "box" before making food, hence we can reconstruct the scenario: consumption purpose: take out eggs from the box.
Identify decisive scenario: The scenario "protective purpose: put eggs into a box" contributes the most to the relation "entity-destination".
Uncover potential relation: If we focus on another promising scenario "preservation purpose: eggs are kept in a box", in the scenario, entity 1 "eggs" is "content", hence the relation should start with "content-", entity 2 "box" is "container", hence the relation should be supplemented as "content-container", hence the commonsense relation is "content-container".
Replace causal term: To change the original relation to the potential one "content-container", the identified causal term can be replaced with "stored in".
Output: <e1> eggs </e1> are stored in a <e2> box </e2>
New Relation: content-container

Example 2:
Input: <e1> eggs </e1> are moved into a <e2> box </e2>
Entities: eggs, box
Relation: entity-destination
Identify causal term: The context words "moved into" are causally related to the relation "entity-destination".
.....
Uncover potential relation: If we focus on another promising scenario "consumption purpose: take out eggs from the box", in the scenario, entity 1 "eggs" is "entity", hence the relation should start with "entity-", entity 2 "box" is "origin", hence the relation should be supplemented as "entity-origin", hence the commonsense relation is "entity-origin".
Replace causal term: To change the original relation to the potential one "entity-origin", the identified causal term can be replaced with "from".
Output: <e1> eggs </e1> are from a <e2> box </e2>
New Relation: entity-origin

Exam (complete the remaining content and maintain consistency with the format of the above examples):

Table 9: The Implementations of PSPC in SemEval. The blue spans represent the pattern separation (PS) process. The orange spans represent the pattern completion (PC) process. The green spans represent the process of uncovering potential relations based on instance constraint and reconstructed scenarios.

Task definition: Change the relation between entities based on minimal context editing.

Instruction: The process can be divided into the following three steps. (1) Identify causal term: Find the context words that are causally related to the relation. (2) Uncover potential relation: Select another commonsense relation from the candidate relations for the entities. (3) Replace causal term: Replace the identified causal term with suitable words to change the original relation to the potential one.
Candidate relations: message-topic, topic-message, entity-origin, origin-entity, entity-destination, destination-entity, content-container, container-content, cause-effect, effect-cause, component-whole, whole-component, member-collection, collection-member, instrument-agency, agency-instrument, product-producer, producer-product.

Example 1:
Input: <e1> eggs </e1> are moved into a <e2> box </e2>
Entities: eggs, box
Relation: entity-destination
Identify causal term: The context words "moved into" are causally related to the relation "entity-destination".
Uncover potential relation: Apart from the original relation "entity-destination", entity 1 "eggs" can be "content", hence the relation should start with "content-", entity 2 "box" can be "container", hence the relation should be supplemented as "content-container", hence another commonsense relation is "content-container".
Replace causal term: To change the original relation to the potential one "content-container", the identified causal term can be replaced with "stored in".
Output: <e1> eggs </e1> are stored in a <e2> box </e2>
New Relation: content-container

Example 2:
Input: <e1> eggs </e1> are moved into a <e2> box </e2>
Entities: eggs, box
Relation: entity-destination
Identify causal term: The context words "moved into" are causally related to the relation "entity-destination".
Uncover potential relation: Apart from the original relation "entity-destination", entity 1 "eggs" can be "entity", hence the relation should start with "entity-", entity 2 "box" can be "origin", hence the relation should be supplemented as "entity-origin", hence another commonsense relation is "entity-origin".
Replace causal term: To change the original relation to the potential one "entity-origin", the identified causal term can be replaced with "from".
Output: <e1> eggs </e1> are from a <e2> box </e2>
New Relation: entity-origin

Exam (complete the remaining content and maintain consistency with the format of the above examples, if no other commonsense relation exists, use the original relation as the potential one):

Table 10: The CF-CoT Implementations in LLMs. The green spans represent the process of uncovering potential relations.

Exam:
Input: The final <e1> chapter </e1> offers a <e2> theological survey </e2> of the use of the formula.
Entities: chapter, theological survey
Relation: message-topic

PSPC:
Identify causal term: the context word "offers" are causally related to the relation "message-topic".
Deconstruct entities: Based on the attributes of the entities, "chapter" can be deconstructed into several primary properties: form: a section of a book, content: information about a specific topic, purpose: provide knowledge; "theological survey" can be deconstructed into several primary properties: form: a research paper, content: study of religious beliefs, purpose: analyze and present findings.
Pair the properties between "chapter" and "theological survey" to reconstruct three reasonable scenarios:
Reconstruct the first scenario: Based on the "form: a section of a book" property of "chapter" and the "content: study of religious beliefs" property of "theological survey", the "chapter" should describe the "theological survey", hence we can reconstruct the scenario: informative purpose: chapter describes theological survey.
Reconstruct the second scenario: Based on the "content: information about a specific topic" property of "chapter" and the "purpose: analyze and present findings" property of "theological survey", the "chapter" should provide information about the "theological survey", hence we can reconstruct the scenario: informative purpose: chapter provides information about theological survey.
Reconstruct the third scenario: Based on the "purpose: provide knowledge" property of "chapter" and the "form: a research paper" property of "theological survey", the "chapter" should contribute to the knowledge provided by the "theological survey", hence we can reconstruct the scenario: supportive purpose: chapter supports theological survey.
Identify decisive scenario: The scenario "informative purpose: chapter describes theological survey" contributes the most to the relation "message-topic".
Uncover potential relation: If we focus on another promising scenario "supportive purpose: chapter supports theological survey", in the scenario, entity 1 "chapter" is "component", hence the relation should start with "component-", entity 2 "theological survey" is "whole", hence the relation should be supplemented as "component-whole", hence the commonsense relation is "component-whole".
Replace causal term: To change the original relation to the potential one "component-whole", the identified causal term can be replaced with "is part of".
Output: The final <e1> chapter </e1> is part of a <e2> theological survey </e2> of the use of the formula.
New Relation: component-whole

CF-CoT:
Identify causal term: the context word "offers" are causally related to the relation "message-topic".
Uncover potential relation: Apart from the original relation "message-topic", entity 1 "chapter" can be "topic", hence the relation should start with "topic-", entity 2 "theological survey" can be "message", hence the relation should be supplemented as "topic-message", hence another commonsense relation is "topic-message".
Replace causal term: To change the original relation to the potential one "topic-message", the identified causal term can be replaced with "reflects".
Output: The final <e1> chapter </e1> reflects a <e2> theological survey </e2> of the use of the formula.
New Relation: topic-message

Table 11: An example comparing the inference processes between PSPC and CF-CoT. The blue spans represent the pattern separation (PS) process. The orange spans represent the pattern completion (PC) process. The green spans represent the process of uncovering potential relations.