Prompting Large Language Models for Counterfactual Generation: An Empirical Study

Yongqi Li[1,*], **Mayi Xu**[1,*], **Xin Miao**[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
{liyongqi, xumayi, miaoxin, shenzhou, qty}@whu.edu.cn

Abstract

Large language models (LLMs) have made remarkable progress in a wide range of natural language understanding and generation tasks. However, their ability to generate counterfactuals has not been examined systematically. To bridge this gap, we present a comprehensive evaluation framework on various types of NLU tasks, which covers all key factors in determining LLMs' capability of generating counterfactuals. Based on this framework, we 1) investigate the strengths and weaknesses of LLMs as the counterfactual generator, and 2) disclose the factors that affect LLMs when generating counterfactuals, including both the intrinsic properties of LLMs and prompt designing. The results show that, though LLMs are promising in most cases, they face challenges in complex tasks like RE since they are bounded by task-specific performance, entity constraints, and inherent selection bias. We also find that alignment techniques, e.g., instruction-tuning and reinforcement learning from human feedback, may potentially enhance the counterfactual generation ability of LLMs. On the contrary, simply increasing the parameter size does not yield the desired improvements. Besides, from the perspective of prompt designing, task guidelines unsurprisingly play an important role. However, the chain-of-thought approach does not always help due to inconsistency issues.

Keywords: Large Language Models, Counterfactual Generation, Natural Language Understanding

1. Introduction

Counterfactual generation, designed to eliminate spurious correlations in data, is a crucial technique used in causal intervention (Pearl, 1993). In recent years, many studies (Kaushik et al., 2020; Niu et al., 2021; Zhang et al., 2023) have attempted to enhance the robustness and performance of neural network models through counterfactual generation.

Large language models (LLMs) like ChatGPT are revolutionizing the field of natural language processing (Liu et al., 2023). Due to their power in understanding instructions, learning in context, and text generation, LLMs have attracted widespread attention in utilizing prompt engineering to generate text in specific scenarios. However, the potential of LLMs in generating counterfactuals remains unexplored systematically. This paper aims to bridge this gap by answering two key questions, 1) *What strengths and weaknesses do LLMs have in generating counterfactuals?* 2) *What factors influence the counterfactual generation ability of LLMs?*

To answer these two questions, we develop a comprehensive framework for evaluating LLMs' capability of generating counterfactuals on four typical natural language understanding (NLU) tasks, i.e., sentiment analysis (SA), natural language inference (NLI), named entity recognition (NER), and relation extraction (RE). Our framework covers all key factors in LLMs, including the inherent properties of LLMs themselves like the model size as well as the prompt designing for LLMs.

For the first question, we select the powerful GPT-3.5 as an example for evaluation. The experimental results show that LLMs can bring about promising enhancements under most settings. However, LLMs also have displayed some weaknesses when dealing with complex tasks such as RE. Further, to discover reasons for the weakness, we first examine the correlation between the quality of generated counterfactuals and the task-specific performance of LLMs. Then, we explore the factors that are crucial in determining the quality of counterfactuals in the RE task, regarding the satisfaction of entity constraints and the selection bias.

For the second question, we first employ the proposed evaluation framework on Llama-2 family of LLMs (Touvron et al., 2023), which includes {7,13,70}b, {7,13,70}b-chat versions, to investigate the impact of parameter sizes and alignment techniques. Second, we evaluate GPT-3.5 using different prompt variants to examine whether the task guidelines and chain-of-thought (CoT) (Wei et al., 2022b) are beneficial, and whether the counterfactual generation ability of LLMs is learned from the demonstration or is intrinsic.

Overall, this study makes two major contributions as follows:

(1) We are the first to present a comprehensive framework for systematically evaluating the counterfactual generation ability of LLMs. Our framework covers various types of NLU tasks and all key

[*] Equal contributions.
[†] Corresponding author.

LREC-COLING 2024, pages 13201–13221

factors in LLMs, including the parameter size, alignment technique, task guideline and CoT. This framework is then deployed to investigate the strengths and weaknesses of LLMs when generating counterfactuals.

(2) Our study reveals that LLMs can generate high-quality counterfactuals in most cases, but struggle to handle complex tasks such as RE. Moreover, the alignment technique can enhance the counterfactual generation capabilities of LLMs, whereas increasing the parameter size or applying CoT is not always beneficial.

2. Related Work

2.1. Large Language Models (LLMs)

Recently, there have been breakthrough advances in the capabilities of large language models (LLMs) (Zhao et al., 2023), especially in understanding instructions and in-context learning (Dong et al., 2022). The improvement of these capabilities mainly comes from the scaling up of the parameter size, also known as the emergence phenomenon (Wei et al., 2022a), and the inclusion of alignment techniques (Ouyang et al., 2022a), such as instruction-tuning and reinforcement learning with human feedback. Besides, when prompting LLMs for specific tasks, researchers have also found some ways to improve the performance, such as providing detailed task descriptions (Efrat and Levy, 2020), adopting chain-of-thought (Wei et al., 2022b; Kojima et al.) and selecting reasonable demonstration (Liu et al., 2022). In this study, we comprehensively examine these potentially affecting factors of LLMs for counterfactual generation.

2.2. Counterfactual Generation

Recent research on causal inference theory (Pearl, 2009; Rubin, 1974; Morgan and Winship, 2015; Pearl and Mackenzie, 2018; Feder et al., 2022) has gained increasing attention due to its potential to enhance the model performance and stability by mitigating spurious correlations in the data (Kaushik et al., 2020; Niu et al., 2021). In the area of natural language processing, counterfactual generation has emerged as a prominent area of interest and been employed for various tasks, such as text classification (Garg and Ramakrishnan, 2020; Wang and Culotta, 2021), question answering (Ou et al., 2022; Paranjape et al., 2022), sentiment analysis (Kaushik et al., 2020; Ross et al., 2021; Robeer et al., 2021; Chen et al., 2021; Yang et al., 2021; Howard et al., 2022; Wen et al., 2022), natural language inference (Dixit et al., 2022; Wen et al., 2022), named entity recognition (Zeng et al., 2020; Yang et al., 2022), and relation extraction (Zhang et al., 2023; Miao et al., 2023). These methods mainly follow the paradigm of causal identification, label-controlled generation and data augmentation, which is also adopted by our proposed evaluation framework.

Notably, there are very limited LLMs-based methods (Dixit et al., 2022; Chen et al., 2023) for counterfactual generation, which only focus on relatively simple SA and NLI tasks. Moreover, a comprehensive evaluation on LLMs for generating counterfactuals is missing from the literature. To fill this gap, we propose an evaluation framework and conduct a multi-perspective empirical study for counterfactual generation using LLMs, covering various types of NLU tasks including SA, NLI, NER and RE.

3. Methodology

3.1. Causal Theoretical Foundation

In this subsection, we use the structural causal model (SCM) (Pearl et al., 2000) to establish a causal theoretical foundation for counterfactual generation and augmentation. Here we use the SCM of the SA task as a representative for illustration.

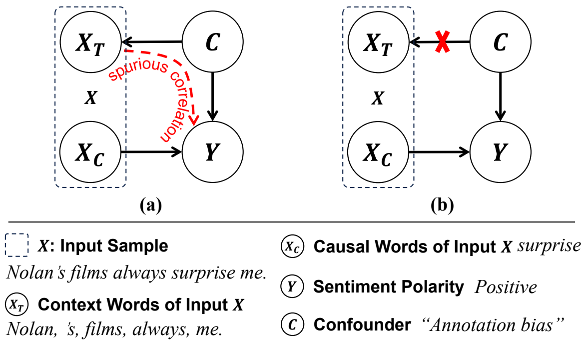

Figure 1: (a) Structural causal model of the SA task, (b) Intervention operation.

Structural Causal Model (SCM) As shown in Fig. 1 (a), the SCM mainly shows the relationships among the causal words (X_C), the context words (X_T) and the sentiment polarity (Y). $X_C \rightarrow Y$: the causal words (X_C), i.e., sentiment-related words, are the sole cause of sentiment polarity (Y). $C \rightarrow X_T$ and $C \rightarrow Y$: Since only the sentiment-related words X_C are attended during the collection of SA training samples, the distribution of the context words X_T and the sentiment polarity Y is ignored. Thus there may be an annotation bias, i.e., the hidden confounder C, affecting both X_T and Y.

Spurious Correlation Let y_+ and y_- denote the positive and negative sentiment polarity, respectively. Assume in a collection of training samples, the context word x_{t_1} = *"Nolan"* appears frequently in sentences with positive polarities and hardly

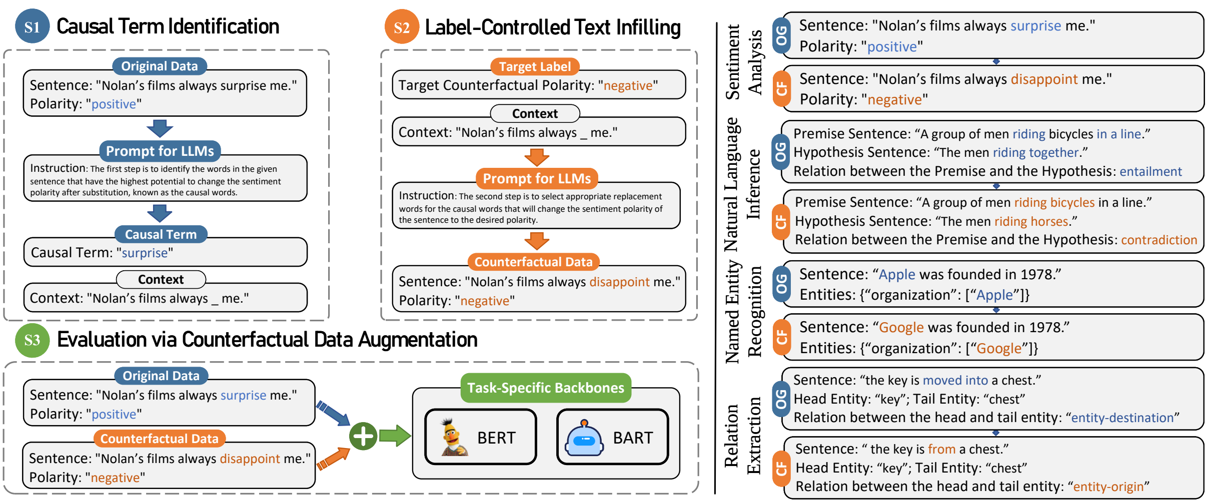

Figure 2: Left: The proposed framework for evaluating counterfactuals generated by LLMs (SA task). Right: Original (OG) samples and generated counterfactual (CF) samples on SA, NLI, NER and RE tasks.

ever in those with negative polarities. Classical model training based on empirical risk minimization (ERM) (Vapnik, 1991), indiscriminately learns from both spurious correlation $X_T \leftarrow C \rightarrow Y$ and causal correlation $X_C \rightarrow Y$. Thus, we can obtain:

$$P(Y|X) = \frac{1}{||X_C \cup X_T||} \sum_{x_i \in (X_C \cup X_T)} P(Y|x_i), \quad (1)$$

where $||X_C \cup X_T||$ denotes the number of words in the input X and $Y = \{y_+, y_-\}$. Due to the bias factor mentioned before, $P(y_+|x_{t_1})$ is much larger than $P(y_-|x_{t_1})$, and thus $P(Y|x_{t_1})$ tends to dominate the overall distribution. That is, the model tends to learn $P(Y|X)$ from x_{t_1} rather than X_C during the training process.

Intervention and Counterfactual Generation To alleviate the issue above, one important way is to conduct causal intervention (Pearl, 1993) via counterfactual generation. Before performing causal intervention $do(X_T)$, one crucial step is to separate X_T from X_C, i.e., causal words identification. Next, we need to ensure that X_T is unchanged, e.g., $do(X_T) = x_{t_1}$, and flip the sentiment polarity by changing X_C, e.g., *"Nolan's films always **disappoint** me."*. After completing interventions for all samples, the counterfactual samples are augmented to the original samples so that both $P(y_+|do(X_T))$ and $P(y_-|do(X_T))$ are 1/2. Hence X_T has almost no contribution to $P(Y|X)$. As shown in Fig. 1 (b), $P(Y|X)$ can be rewritten as:

$$P(Y|X) = \frac{1}{||X_C||} \sum_{x_i \in X_C} P(Y|x_i), \quad (2)$$

which means that the model concentrates on learning $P(Y|X)$ from causal words X_C.

3.2. LLMs for Counterfactual Generation

As illustrated in Fig. 2, the proposed evaluation framework consists of three steps.

$S1$ (causal term identification): Separating causal words from context words.

$S2$ (label-controlled text infilling): Maintaining the context words unchanged, changing the label of the sample by altering the causal words.

$S3$ (counterfactual data augmentation): Combining the original and counterfactual samples as training samples.

Since we aim to evaluate the counterfactual generation ability of LLMs, $S1$ and $S2$ are performed by prompting LLMs for text completion. The combined samples after $S3$ are then used to train backbones for performing typical NLU tasks, e.g., SA.

Prompt Design For each labeled training sample, x_i, we construct a triplet prompt [T^p, D^p, x_i^p]. T^p denotes task guidelines, including a precise definition of the task along with step-by-step descriptions on how to generate counterfactuals. D^p represents the demonstration part used to clarify the format of inputs and outputs. x_i^p denotes the standardized format of the original sample x_i, like that in the D^p. We provide such triplet prompts to LLMs, and expect LLMs to identify causal words, replace causal words and generate desired counterfactuals.

3.3. Backbones for Data Augmentation

To measure the quality of the generated counterfactuals, we compare the performance of small language models (SLMs) trained with the original or counterfactually augmented data. We adopt SLMs like BERT (Devlin et al., 2018) and BART (Lewis et al., 2020) as backbones[1], which are typical for natural language understanding and generation tasks, respectively. For BERT-based SLMs, the output embeddings of BERT are inputted to the MLP or CRF for further classification or tagging. The backbone is trained to minimize the cross-entropy loss. For BART-based SLMs, the training goal of the model is to generate the target text following a pre-defined template and then we can de-linearize it into labels.

4. Evaluation of LLMs as the Counterfactual Generator

In this section, we choose GPT-3.5 as an example to evaluate and analyze the counterfactual generation ability of LLMs on four typical NLU tasks.

4.1. Evaluation Protocal

Datasets and Evaluation Metrics We conduct experiments across various datasets. Specifically, we adopt SST-2 (Socher et al., 2013) and IDMB (Maas et al., 2011) for the SA task, SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) for the NLI task, CoNLL2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotesV5 (Weischedel et al., 2013) for the NER task, SemEval2010 (Hendrickx et al., 2010) and TACRED (Zhang et al., 2017) for the RE task. We use accuracy as the evaluation metric for SA and NLI and the micro-F1 for NER and RE. We report mean accuracy or micro-F1 with standard deviation using 5 different seeds.

Few-shot Settings Spurious correlations are particularly prevalent in few-shot settings (Nan et al., 2021). To evaluate the generated counterfactuals for mitigating such negative impact, we conduct experiments using randomly sampled {5,10,20,50}-shot training set on each dataset. For the task where each sentence corresponds to a sample-label pair, i.e., SA, NLI, RE, we sample k samples for each class as the few-shot training set under the k-shot setting. For the task where each sentence corresponds to one or more sample-label pairs, i.e., the NER task, following (Yang and Katiyar, 2020), we adopt the greedy sampling algorithm[2].

Compared Methods for Typical NLU Tasks To investigate the efficacy of the generated counterfactuals by LLMs, we compare the performance of the following methods on typical NLU tasks.

LLMs: We test the performance of LLMs under few-shot settings as a comparative baseline. Specifically, we adopt the widely used in-context learning approach, i.e., performing specific tasks by understanding instructions and in-context demonstrations in the prompt. Generally, the prompt consists of three parts, i.e., [{task definition}, {demonstrations}, {test sample}].

SLMs (Original): We test the original few-shot performance of SLMs via the BERT-based or BART-based fine-tuning methods.

SLMs (Internal knowledge augmented): We augment the original SLMs with counterfactual data generated by internal knowledge tailored methods, including AutoCAD (Wen et al., 2022) for SA and NLI tasks, CFGen (Zeng et al., 2020) for NER, and CoCo (Zhang et al., 2023) for RE.

SLMs (LLMs augmented): We augment the original SLMs with counterfactual data generated by LLMs via the method introduced in Section 3.2. Notably, the purpose of including both SLMs (Internal knowledge augmented) and SLMs (LLMs augmented) here is to compare well-designed models with limited internal knowledge and general models with a large amount of external knowledge[3].

4.2. Discussion on Experimental Results

Fig. 3 shows the experimental results[4] of various compared methods under few-shot settings, which we will discuss next.

Do SLMs Have Chances to Outperform LLMs? 1) LLMs maintain clear advantages on relatively simple SA and NLI tasks, as well as on NER and RE tasks under extremely few-shot settings. 2) But for the NER and RE tasks, the advantage of LLMs seems to be not so obvious. Normally, the increase in the number of labels requires an increase in task-specific knowledge. However, the in-context learning approach may prevent LLMs from fully acquiring task-specific knowledge from the provided demonstrations. In other words, increasing the number of demonstrations may not notably improve the performance of LLMs (Ma et al., 2023). Therefore, for tasks with many labels, e.g., NER and RE[5], the performance of fine-tuned SLMs consistently improves since SLMs can acquire more task-specific knowledge through fine-tuning while LLMs cannot.

[1]We also experiment with LLMs as backbones for data augmentation. However, the time cost for testing is extremely expensive and the performance is not as good as that of SLMs. So we omit the evaluation on LLMs as backbones for data augmentation.

[2]Note that it is unavoidable that the NER few-shot training set obtained using the greedy sampling algorithm may contain more than k samples corresponding to some classes.

[3]Please refer to Appendix A for more descriptions about experimental details.

[4]Please refer to Appendix B.1 for detailed results.

[5]There are 2, 3, {4, 18}, and {10, 42} labels in the SA, NLI, NER, and RE tasks, respectively.

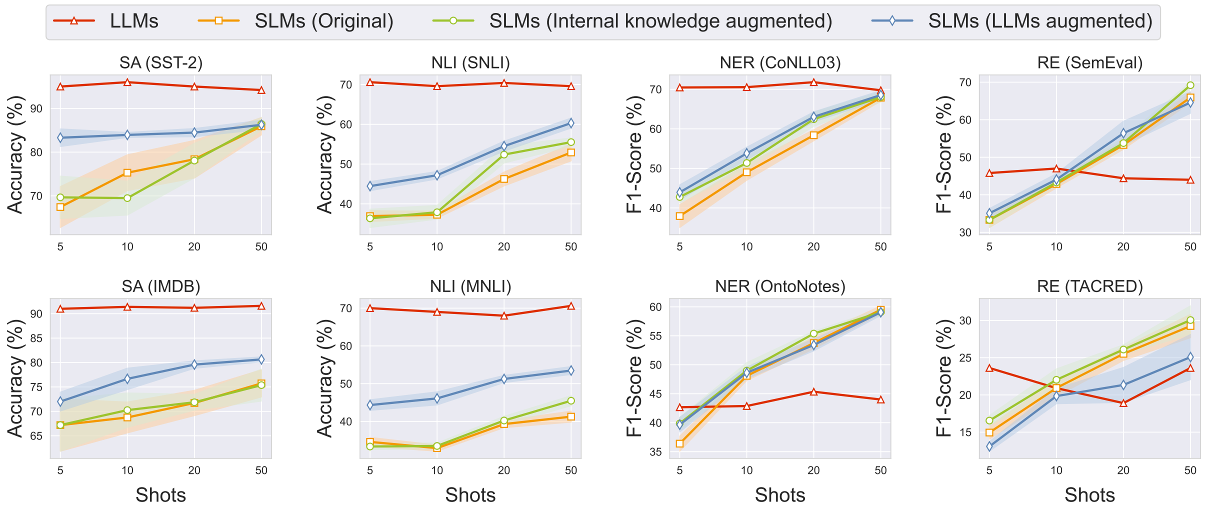

Figure 3: Performance comparison under few-shot settings. The LLMs refer to GPT-3.5. The results of SLMs are obtained by averaging the performance of BERT-based and BART-based fine-tuned models.

Eventually, the performance of SLMs catches up or even surpasses that of LLMs.

Can LLMs-Generated Counterfactuals Enhance the Performance of SLMs? 1) Counterfactual data generated by LLM significantly improve the performance of SLMs on SA and NLI tasks. 2) LLMs perform poorly on the more complex NER and RE tasks, where they only bring enhancements on some datasets (CoNLL2003 and SemEval), and even cause performance degradation on the TACRED (RE) dataset. This is likely due to the failure of LLMs to consider entity constraints when generating counterfactuals, which will be analyzed later.

Can LLMs Always Achieve Better Augmentation Results than Internal Knowledge Tailored Methods? 1) In most cases, LLMs demonstrate superior performance than internal knowledge tailored methods in generating counterfactuals, due to the vast inherent knowledge in them. 2) Nevertheless, when engaged with the RE task, the internal knowledge tailored method CoCo is more effective. This is largely attributable to its meticulous design and the set constraints that guide the counterfactual generation process.

4.3. Weaknesses Analysis of LLMs for Counterfactual Generation

The Quality of Generated Counterfactuals is Bounded by LLMs' Task-specific Performance We visualize the average performance of LLMs themselves and the average augmentation effects for SLMs on each dataset.

As shown in the Fig. 4, we find a strong correlation between the counterfactual generation capability of LLMs and their task-specific performance.

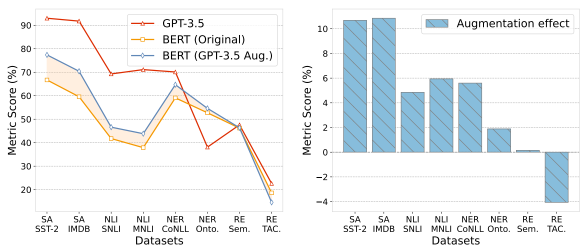

Figure 4: Task-specific performance (left) of LLMs and augmentation effects on SLMs (right).

Specifically, for the simpler SA task, LLMs can achieve up to about 93% accuracy, and the generated counterfactual data have the most significant augmentation effect on SLMs, with about 11% absolute increase. On the TACRED dataset of the hard RE task, LLMs can only achieve a 22% micro-F1 score. Correspondingly, the counterfactual data generated by LLMs have even a negative impact on the SLMs, i.e., a 4% absolute decrease. This finding indicates that the quality of generated counterfactuals is heavily bounded by the LLMs' task-specific performance, owing to the fact that we can only design prompts for counterfactual generation, which is far from expectations.

LLMs Fail to Fully Consider Entity Constraints when Generating Counterfactuals for RE In our previous experiments, we observe that for the RE task, the counterfactuals generated by GPT-3.5 might have a negative effect on SLMs. To investigate this issue, we select 100 generated counterfactuals for human evaluation. Specifically, we first determine whether the generated counterfactuals are reasonable, and then annotate the reasons for unreasonable ones.The results are presented in Fig. 5.

Case of Type B	The flight departs from an airport on territory of a member state to which the Treaty applies.	Entity-Origin
Counterfactual	The flight arrives at an airport on the territory of a member state to which the Treaty applies.	Destination-Entity
Case of Type C	The woods that furnish the best charcoal for painters are the beech and vine.	Instrument-Agency
Counterfactual	The beech and vine are the origin of the best charcoal for painters.	Entity-Origin

Table 1: Case study of noisy counterfactual samples generated by GPT-3.5 on the RE task. Cases are from the SemEval dataset. Entities are in blue.

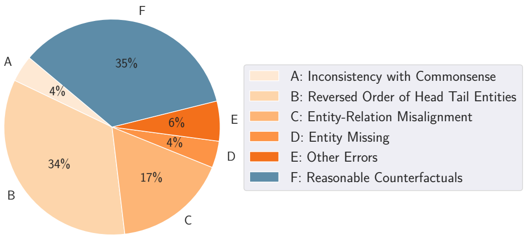

Figure 5: Reasons that lead to unreasonable counterfactuals and corresponding proportions.

From Fig. 5, it can be seen that type B and type C, i.e., “Reversed Order of Head Tail Entities” and “Entity-Relation Misalignment”, are the two dominant causes of unreasonable counterfactuals. We select two cases corresponding to these two types and present them in Table 1. In the case of type B, “flight” and “airport” should form the “Entity-Destination” relation, but not the reversed one, i.e., “Destination-Entity”. In the case of type C, the concerned entities are “charcoal” and “painters”. However, in the generated counterfactual sentence, the “Origin” part to form an “Entity-Origin” relation with the head entity “charcoal” is “beech and vine” rather than the tail entity “painters”. These findings indicate that LLMs still struggle with entity constraints such as the “head-tail order” and “alignment with the counterfactual relation”.

Selection Bias in LLMs Undermines Counterfactual Generation for the RE Task This section aims to investigate the potential selection bias in the choice of target counterfactual relations by LLMs. Specifically, we select 100 samples, with 10 samples for each relation from the SemEval dataset, to observe the frequency of relation transfer. To exclude potential biases introduced by the demonstration in the prompt, we average the results using 9 different prompts with different target counterfactual relations in the demonstration. We then visualize the average frequency matrix of the “original-counterfactual relation transfer” for every 100 samples in Fig. 6 (a). We also adopt the hypernyms in WordNet (Miller, 1995), which can group similar words into a high-level concept[6], to observe such selection bias. Fig. 6 (b) shows the number of hypernyms of the head and tail concepts within each relation.

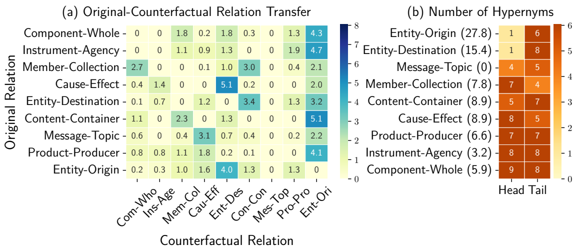

Figure 6: (a) Visualization of original-counterfactual relation transfer frequency. The number represents the frequency of the corresponding transition every 100 samples. (b) Visualization of the number of hypernyms for each head and tail concept. The number in () represents the average frequency of being the target counterfactual relation for every 100 samples.

From Fig. 6 (a), it is clear that LLMs tend to select certain relations as the target counterfactual ones. Further, together with Fig. 6 (b), we can see that, except for the “Message-Topic” relation, the frequency of a relation being chosen as a target counterfactual relation and the number of its hypernyms are negatively correlated. In other words, LLMs prefer to choose more abstract relation types as the target counterfactual ones such as “Entity-Origin”. Such selection bias leads to a serious imbalance of labels in the generated counterfactual sample set, which may result in a performance decrease of the counterfactually augmented model.

Promising Ways for Improving the Quality of Counterfactuals To address the two main causes of low-quality counterfactuals analyzed in previous paragraphs, i.e., inconsistency between entities and labels, and the selection bias of target counterfactual relations, there are three possible corresponding solutions. 1) *Consistency Filtering*: We can employ SLMs trained on specific tasks to filter counterfactual samples for consistency. For example, only samples that SLMs correctly predict can be retained. 2) *Consistency Correction*: We can utilize LLMs to conduct additional checks on the generated counterfactuals and correct them accordingly, such as whether the order of the head

[6] In WordNet, the fewer hypernyms a word has, the higher level of abstraction the word is.

Aug. Method	SA				Avg.	NLI				Avg.
	SST-2		IMDB			SNLI		MNLI		
	5-shot	10-shot	5-shot	10-shot		5-shot	10-shot	5-shot	10-shot	
None	59.67±5.39	66.88±6.68	57.05±1.40	57.14±2.15	60.19	37.67±1.29	38.77±1.23	35.03±1.72	32.40±1.00	35.97
Llama2-7b	56.68±4.18	65.45±6.73	56.03±1.65	54.77±1.76	58.23	38.22±1.02	43.39±1.73	35.25±1.67	32.45±1.02	37.33
Llama2-7b-chat	81.88±1.70	82.15±1.22	52.88±0.82	57.82±4.78	68.68	37.88±4.27	40.64±0.85	41.24±1.81	38.35±4.18	39.53
Llama2-13b	56.57±5.31	68.30±6.07	52.55±1.38	50.13±1.28	56.89	38.93±1.47	39.67±1.45	35.24±1.66	34.23±1.47	37.02
Llama2-13b-chat	81.86±1.97	83.54±0.23	54.41±3.05	64.07±3.60	70.97	34.03±1.25	43.82±1.53	36.23±1.90	38.35±3.67	38.11
Llama2-70b	65.14±6.10	71.38±5.52	53.93±1.57	52.16±1.26	60.65	41.00±1.83	47.28±2.56	37.87±1.35	34.24±1.73	40.10
Llama2-70b-chat	81.31±1.35	84.04±0.44	59.71±1.69	62.96±3.41	72.00	37.77±1.40	44.50±0.88	35.78±2.38	38.60±3.11	39.16

Table 2: Performance comparison of data augmentation experiments with counterfactuals generated by different LLMs. Here we use BERT-based SLMs. The better are in orange.

and tail entities is reversed, whether any entity is missing. 3) *Correcting Selection Bias of LLMs*: For example, we can restrict the choice of the target counterfactual relation based on commonsense knowledge related to the head and tail entities in advance. This prevents the LLMs from being influenced by their inherent bias when selecting the target counterfactual relation.

5. Analysis on What Affects LLMs for Counterfactual Generation

In this section, we will first analyze the impact of intrinsic properties of LLMs. Then, we will analyze the impact of prompt designing.

5.1. Intrinsic Properties of LLMs

To explore what intrinsic properties of LLMs affect the quality of counterfactual generation, we employ the Llama-2 series LLMs (Touvron et al., 2023) for counterfactual generation. We mainly focus on two key factors, the parameter size of LLMs and whether using alignment techniques, e.g., reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022b). We choose the Llama-2 family of LLMs because there are six versions of Llama-2, i.e., `7b`, `13b`, `70b`, `7b-chat`, `13b-chat`, `70b-chat`. The only difference between "`{7,13,70}b`" and "`{7,13,70}b-chat`" versions is that the latter adopt instruction-tuning and RLHF techniques for aligning with humans. This provides good conditions for us to conduct controlled-variable experiments. Please note that even using the powerful GPT-3.5, it is hard to generate high-quality counterfactuals for the NER and RE tasks, making them unsuitable for comparing the counterfactual generation capabilities of different LLMs. Thus, we only select two relatively simple tasks, i.e., SA and NLI, for the experiments.

Increasing Parameter Size cannot Improve Counterfactual Generation of LLMs To explore whether the parameter size of LLMs is critical to the counterfactual generation capability, we conduct counterfactual data augmentation experiments on SLMs, using counterfactuals generated by LLMs with different parameter sizes. Fig. 7 illustrates the trend of counterfactually augmented SLMs performance with respect to the parameter size of LLMs.

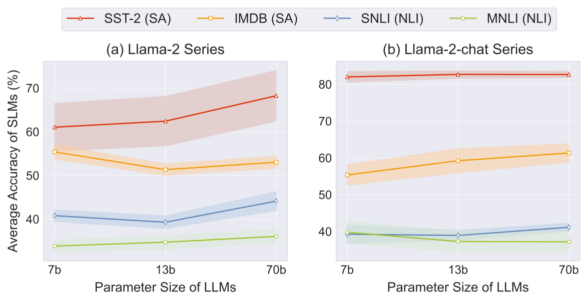

Figure 7: Performance comparison of counterfactually augmented SLMs. The counterfactuals are generated by Llama-2 (left) and Llama-2-chat (right) series with different parameter sizes. Results are obtained by averaging the performance of BERT-based SLMs under 5-shot and 10-shot settings.

It can be seen that despite the 10 times parameter size of LLMs (from 7b to 70b), the task performance of the counterfactually augmented SLMs is not significantly improved, i.e., the quality of generated counterfactuals is not notably improved. This suggests that the counterfactual generation ability of LLMs does not improve as the number of model parameters rises, which is quite different from the widely held findings of previous studies for other tasks (Wei et al., 2022a).

Alignment Techniques may Help Improving Counterfactual Generation of LLMs Table 2 presents the performance comparison of SLMs on SA and NLI tasks, which are augmented using counterfactuals generated by Llama-2 and Llama-2-chat series. For the SA task, the counterfactuals generated by Llama-2-chat series can bring a 10.4%-14.1% absolute accuracy increase averagely for SLMs than that of Llama-2 series. Since the only difference between the Llama-2 and Llama-2-chat series is that the latter employs alignment

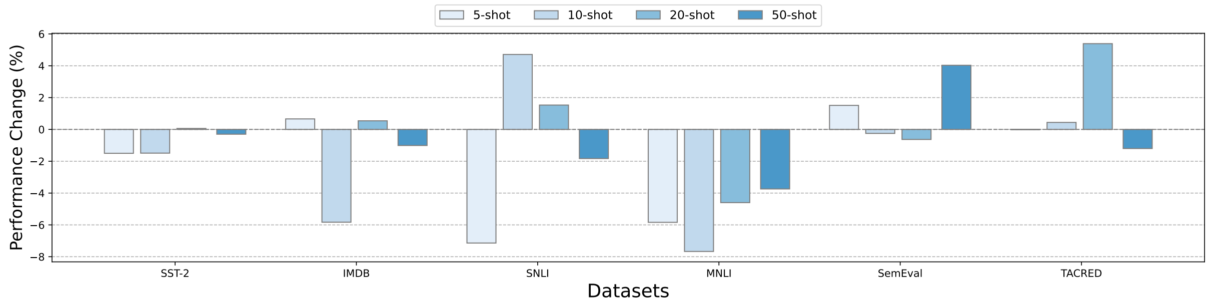

Figure 8: The impact of chain-of-thought. The vertical axis represents the change in the performance after using chain-of-thought for counterfactual generation. Note that for the NER task, the causal words are the entities themselves. There is no need to find the causal words first, so it is not discussed here.

Premise	The echoes of colonialism are in the Railway Station, which is locally nicknamed the Taj Mahal.			
Hypothesis	**Variant**			**Sentence**
	Original			The Railway Station is locally nicknamed the Taj hindra.
	w/o CoT			The Railway Station's Moorish architecture is locally nicknamed the Taj Mahal.
	w/ CoT	Process	*Causal Identification*	*The relationship "Contradiction" depends on "Taj hindra" in the hypothesis sentence.*
			Causal Replacement	*To change the relation from "Contradiction" to "Entailment", "Taj hindra" is replaced by "Mahal".*
		Generated Sentence		The echoes of colonialism are in the Railway Station, which is locally nicknamed the Mahal.

Table 3: An example of generated counterfactuals on the MNLI dataset without or with chain-of-thought (CoT). In this case, GPT-3.5 generates a counterfactual hypothesis sentence to change the relationship between the "Premise" and the "Hypothesis" sentence from "Contradiction" to "Entailment".

techniques, i.e., instruction-tuning and RLHF, it is reasonable to conclude that the alignment techniques make important contributions to the improvement of the counterfactual generation capabilities. But for the NLI task, the advantages of Llama-2-chat series over Llama-2 series are not significant. Upon more detailed analysis, we observe that when generating "Contradiction" counterfactuals, Llama-2-chat series tend to favor generating "Neutral" samples by changing part of the semantics of the sentence, which introduces much noise. This reflects the misalignment of label semantics understanding between LLMs and humans. These findings inspire us to improve the counterfactual generation of LLMs by further exploring alignment techniques, e.g., 1) aligning LLMs' causal discovery process with that of humans to further enhance LLMs' causal discovery ability, 2) aligning LLMs' understanding of domain-specific label semantics with that of humans.

5.2. Impact of Prompt Designing

In this section, we adopt GPT-3.5 for exploring the impact of prompt designing[7].

Task Guidelines are Critical for Counterfactual Generation To verify the necessity of providing detailed task guidelines in the prompt, we conduct comparison experiments of counterfactual generation with and without task guidelines.

Table 4 shows the results of the comparison experiments. It can be seen that the removal of the task guidelines leads to a performance drop on most datasets, e.g., a 2% absolute decrease on the NER task. The exception is the results of the RE task. One possible reason for this particular case is that the counterfactuals generated by GPT-3.5 for the RE task have a negative impact on SLMs. When the task guidelines are removed, the number of generated counterfactuals decreases, thus reducing this negative effect.

Task	Variant	Shot				Avg.
		5-shot	10-shot	20-shot	50-shot	
SA	w/ Task Guidelines	**70.54**	**75.64**	**77.84**	**79.16**	**75.80**
	w/o Task Guidelines	69.67	75.37	76.23	78.71	75.00
NLI	w/ Task Guidelines	**42.05**	44.91	**51.67**	**56.32**	**48.74**
	w/o Task Guidelines	39.05	**45.03**	51.52	54.05	47.41
NER	w/ Task Guidelines	**50.78**	**60.66**	**68.09**	**72.55**	**63.02**
	w/o Task Guidelines	46.79	60.14	65.39	72.00	61.08
RE	w/ Task Guidelines	18.84	28.38	35.74	41.44	31.10
	w/o Task Guidelines	**19.88**	**29.22**	**36.25**	**44.37**	**32.43**

Table 4: The impact of task guidelines.

Chain-of-thought does not Always Help Step-by-step task guidelines can help LLMs generate high-quality counterfactuals. Thus, intuitively, generating detailed explanations at each step may further improve the quality of generated counterfactuals. To verify this assumption, we borrow the idea of chain-of-thought (CoT) (Wei et al., 2022b) and use GPT-3.5 to generate counterfactuals in a chain-like manner. Specifically, we introduce two additional stages, "Causal Words Identification" and "Causal Words Replacement", and expect GPT-3.5 to generate explanations of these two steps before generating counterfactuals.

As shown in Fig. 8, counter-intuitively, the coun-

[7]Please refer to Appendix B.1 for detailed results.

terfactuals generated by CoT do not lead to big improvements and even have significant decreases under some settings. This might be due to the problem of inconsistency between the output and the process of counterfactual generation. To have a close look, we select a representative case from the MNLI dataset and present it in Table 3. It can be seen that, when using the prompt with CoT, the process of causal identification and replacement is correct. However, the generated sentence is not the result of only replacing causal words in the original sentence, showing an inconsistent phenomenon.

Even unreasonable demonstration can yield reasonable counterfactuals Another question we are curious about is whether LLMs' ability to generate counterfactuals is acquired by learning the provided demonstration in the prompt, or from its large-scale pre-training process. To answer this question, we replace the demonstration in the prompt with an unreasonable one for generating counterfactuals and show the results in Table 5.

Task	Variant	Shot				Avg.
		5-shot	10-shot	20-shot	50-shot	
SA	w/ Reasonable Demo	**70.54**	**75.64**	**77.84**	**79.16**	**75.80**
	w/ Unreasonable Demo	69.50	73.47	76.40	78.66	74.51
NLI	w/ Reasonable Demo	**42.05**	**44.91**	**51.67**	**56.32**	**48.74**
	w/ Unreasonable Demo	40.08	42.05	48.83	52.03	45.75
NER	w/ Reasonable Demo	50.78	**60.66**	**68.09**	**72.55**	**63.02**
	w/ Unreasonable Demo	**52.59**	60.22	66.03	72.24	62.77
RE	w/ Reasonable Demo	18.84	28.38	**35.74**	**41.44**	31.10
	w/ Unreasonable Demo	**19.71**	**28.86**	33.66	43.36	**31.40**

Table 5: Results of comparison experiments with reasonable and unreasonable demonstrations.

From Table 5, it is interesting that in most cases, the counterfactuals generated by GPT-3.5 using an unreasonable demonstration achieve comparable results to those by using a reasonable one. This suggests that the demonstration in the prompt does not always teach LLMs the task goal of counterfactual generation, i.e., the counterfactual generation capability of LLMs is "innate" in them.

6. Conclusion

This paper presents the first evaluation framework and a systematical empirical study on the capability of LLMs in generating counterfactuals. Experimental results on four typical NLU tasks including SA, NLI, NER, and RE demonstrate that LLMs can generate satisfactory counterfactuals in most cases. However, LLMs also have their weaknesses when dealing with complex tasks like RE due to the ignorance of entity constraints and inherent selection bias. Notably, we also discover that alignment techniques are crucial for improving the counterfactual generation capabilities of LLMs. This inspires us to explore alignment techniques for LLMs to generate high-quality counterfactuals in future work.

Acknowledgments

This work was supported by the grant from the National Natural Science Foundation of China (NSFC) project (No. 62276193). It was also supported by the Joint Laboratory on Credit Science and Technology of CSCI-Wuhan University.

7. Bibliographical References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. Disco: Distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528.

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*.

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

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

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

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Phillip Howard, Gadi Singer, Vasudev Lal, Yejin Choi, and Swabha Swayamdipta. 2022. Neurocounterfactuals: Beyond minimal-edit counterfactuals for richer data augmentation. *arXiv preprint arXiv:2210.12365*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Xin Miao, Yongqi Li, and Tieyun Qian. 2023. Generating commonsense counterfactuals for stable relation extraction. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

Stephen L Morgan and Christopher Winship. 2015. *Counterfactuals and causal inference*. Cambridge University Press.

Guoshun Nan, Jiaqi Zeng, Rui Qiao, Zhijiang Guo, and Wei Lu. 2021. Uncovering main causalities for long-tailed information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9683–9695.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. 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.

Jiao Ou, Jinchao Zhang, Yang Feng, and Jie Zhou. 2022. Counterfactual data augmentation via perspective transition for open-domain dialogues. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1635–1648, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022a. Training language models to follow instructions with human feedback.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022b. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Bhargavi Paranjape, Matthew Lamm, and Ian Tenney. 2022. Retrieval-guided counterfactual generation for QA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1670–1686, Dublin, Ireland. Association for Computational Linguistics.

Judea Pearl. 1993. [bayesian analysis in expert systems]: comment: graphical models, causality and intervention. *Statistical Science*, 8(3):266–269.

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.

Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2):3.

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.

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.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vladimir Vapnik. 1991. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4.

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.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. AutoCAD: Automatically generate counterfactuals for mitigating shortcut learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2302–2317, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122. Association for Computational Linguistics.

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

Linyi Yang, Lifan Yuan, Leyang Cui, Wenyang Gao, and Yue Zhang. 2022. Factmix: Using a few labeled in-domain examples to generalize to cross-domain named entity recognition. *arXiv preprint arXiv:2208.11464*.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7270–7280.

Mi Zhang, Tieyun Qian, Ting Zhang, and Xin Miao. 2023. Towards model robustness: Generating contextual counterfactuals for entities in relation extraction. WWW '23, page 1832–1842, New York, NY, USA. Association for Computing Machinery.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A. Experimental Details

A.1. Datasets

Table 6 shows statistics of various datasets used in our experiments and the number of augmented samples by various counterfactual generation methods. Note that we select `mnli_matched` as the test set for MNLI.

A.2. Internal Augment Methods

AutoCAD (Wen et al., 2022) formulates the task of counterfactual generation as a label-controlled text infilling task. In other words, it aims to generate candidate words that can replace the trigger words in the original sentence, based on the provided target counterfactual label. Since there is no design involved to find the target counterfactual label in this method, it can only be used for counterfactual generation for SA and NLI where the target counterfactual labels are determined.
CFGen (Zeng et al., 2020) employs substitution by using entity words from the same category present in the dataset to generate counterfactual samples.
CoCo (Zhang et al., 2023) generates counterfactuals by flipping contextual words with the assistance of entity guidance. It leverages syntactic and semantic dependency graphs to identify linguistically substitutable spans from other contexts, while also flipping the labels associated with those spans.

A.3. Implement Details of LLMs

We adopt the `gpt-3.5-turbo-0301` API from OpenAI for the experiments in Section 4. For all experiments, the parameter temperature of LLMs is set to `0`.

LLMs for Specific Tasks In the prompt for specific tasks, i.e., SA, NLI, NER, and RE, we set the number of demonstrations as 2, 2, 6, 6, 4, 18, 10, and 16 for the SST-2, IMDB, SNLI, MNLI, CoNLL03, OntoNotes, SemEval, and TACRED datasets, respectively. The demonstrations are randomly selected from the labeled data.

LLMs for Counterfactual Generation In the prompt for generating counterfactuals, we provide a manually written demonstration. Table 15-25 shows the prompts used in the main comparison experiments and the analysis experiments for counterfactual generation of LLMs.

A.4. Implement Details of SLMs

A.4.1. BERT-based

BERT (Devlin et al., 2018) equips multiple layers of Transformer encoder, which is trained by Masked Language Modeling and Next Sentence Prediction. BERT has achieved state-of-the-art results on a wide range of NLP tasks and has been widely adopted by researchers and industry practitioners. In this paper, we adopt the *bert-base-uncased* as the SLM. Specifically, samples of different tasks are inputted to BERT as follow:

- SA: [CLS] + *sentence* + [SEP]
- NLI: [CLS] + *premise sentence* + [SEP] + *hypothesis sentence* + [SEP]
- RE: [CLS] + $entity_1$ + [SEP] + $entity_2$ + [SEP] + *sentence* + [SEP]
- NER: [CLS] + *sentence* + [SEP]

For SA, NLI, and RE tasks, the embeddings corresponding to [CLS] token are inputted to an MLP and Softmax layer to predict the final class of sentiment polarities of sentences, logical relationships of sentence-pairs, and entity-pairs relations, respectively. For the NER task, the embeddings corresponding to all tokens are inputted to a CRF layer to tag the position of entities.

Furthermore, the learning rate, batch size, hidden size, and dropout rate of BERT-based backbones are set to $2 \times e^{-5}$, 64, 768, and 0.1, respectively. Notably, to ensure the reasonable of low-resource scenarios, we evaluate the backbones on test sets when their losses on train sets continuously increase over 3 epochs.

A.4.2. BART-based

With the continuous development of pre-trained generative models, many works try to convert various tasks into text generation tasks to utilize the label semantics better (Zhang et al., 2021; Lu et al., 2021). To verify the quality of counterfactual texts generated by LLMs more convincingly, we adopt the paradigm of text generation into the aforementioned tasks. We use BART-base (Lewis et al., 2020) as the backbone, a sequence-to-sequence pre-trained model composed of a 6-layer encoder and a 6-layer decoder. Generally, the core of tackling tasks by text generation lies in designing specific templates. At the training stage, we need to convert the input and output of each sample into text-to-text form based on the pre-defined templates. At the inference stage, we need to delinearize the label words from the target text generated by the model according to the pre-defined templates. The detailed templates are shown in Table 7. Besides, we map the given label of RE and NER to the natural words to make the model better understand the label semantics as shown in Table 26.

Furthermore, batch size and learning rate are 64 and 5e-5, respectively. We evaluate the model

Dataset	Settings	#Labels	#Train	#Internal Aug.	#GPT-3.5 Aug.	#GPT-3.5 (w/o Instruct.) Aug.	#GPT-3.5 (w/ CoT) Aug.	#GPT-3.5 (w/ Unreason. Demo) Aug.	#Test
SST-2	5-shot	2	10	3	10	10	10	10	1,821
	10-shot		20	4	20	20	20	20	
	20-shot		40	14	40	40	40	40	
	50-shot		100	6	100	100	100	100	
IMDB	5-shot	2	10	0	10	10	8	9	25,000
	10-shot		20	1	20	20	19	19	
	20-shot		40	1	39	40	39	39	
	50-shot		100	7	99	99	98	95	
SNLI	5-shot	3	15	10	26	28	30	30	10,000
	10-shot		30	13	58	56	60	60	
	20-shot		60	29	120	116	120	120	
	50-shot		150	81	300	280	300	300	
MNLI	5-shot	3	15	3	30	28	30	30	10,000
	10-shot		30	9	60	60	60	60	
	20-shot		60	26	120	114	120	120	
	50-shot		150	62	300	298	300	300	
CoNLL03	5-shot	4	11	60	60	9	-	42	3,453
	10-shot		22	174	166	28		101	
	20-shot		39	291	285	70		179	
	50-shot		96	707	673	189		423	
OntoNotes	5-shot	18	54	524	309	132	-	309	12,217
	10-shot		123	1,088	1,155	316		656	
	20-shot		209	1,997	2,073	481		1,120	
	50-shot		603	5,191	5,463	1,265		3,127	
SemEval	5-shot	10	50	0	40	29	14	37	2,717
	10-shot		100	1	71	50	25	70	
	20-shot		200	5	138	93	73	149	
	50-shot		500	71	370	242	147	372	
TACRED	5-shot	41	210	1	201	175	145	170	15,509
	10-shot		416	10	400	349	286	348	
	20-shot		826	16	775	686	572	671	
	50-shot		1,994	90	1,881	1,652	1,397	1,615	

Table 6: Statistics of datasets and the number of augmented samples for SA, NLI, NER and RE tasks by various counterfactual generation methods. “-” denotes that there is no corresponding experiments. Because the causal words for the NER task are the entity words themselves. There is no need to find the causal words first, so there is no need to discuss about the impact of CoT.

Task	Templates
SA	**Input:** one long string of cliches **Output:** negative
NLI	**Input:** premise: The new rights are nice enough hypothesis: Everyone really likes the newest benefits **Output:** netural
RE	**Input:** And then, when right-wing icon Barry Goldwater vacated his U.S. Senate seat in 1986, McCain vaulted into it. The relation between Barry Goldwater and U.S. Senate is **Output:** employee of
NER	**Input:** For Mr. Sherwin, a conviction could carry penalties of five years in prison and a $ 250,000 fine on each count **Output:** person: Sherwin, date: five years, monetary: 250,000

Table 7: Templates of each task for BART-based SLMs.

on the training set per fixed steps {50, 100, 200}. When the performance on the training set reaches 1.0 or continuously decreases over three times, we will stop the model training and evaluate the model performance on the test set.

B. Detailed Experimental Results

B.1. Results of Main Evaluation Experiments

Tables 8-11 show the detailed experimental results of the main comparison experiments. We report the mean micro-F1 score with standard deviation using 5 different seeds.

B.2. Results of Analysis Experiments

Tables 12-14 show the detailed experimental results of the analysis experiments (Section 5.2). We report the mean micro-F1 score with standard deviation using 5 different seeds.

Backbone	Aug. Method	SST-2				IMDB			
		5-shot	10-shot	20-shot	50-shot	5-shot	10-shot	20-shot	50-shot
GPT-3.5	None	95.00	96.00	95.00	94.20	91.00	91.40	91.20	91.60
BERT	None	59.67±5.39	66.88±6.68	73.25±6.33	85.05±1.80	57.05±1.40	57.14±2.15	57.14±2.48	63.73±4.14
	AutoCAD	62.26±6.04	59.62±5.14	72.33±5.53	86.12±1.15	57.05±1.40	58.07±2.40	57.02±1.95	62.97±4.62
	GPT-3.5	83.60±1.32	84.29±0.25	85.58±0.49	85.98±0.60	57.48±2.40	66.98±2.00	70.09±0.80	72.33±0.25
BART	None	75.21±3.94	83.70±1.49	83.50±2.25	86.78±2.15	77.32±9.26	80.36±4.14	86.29±2.67	87.72±1.38
	AutoCAD	77.06±3.63	79.34±2.59	83.81±2.36	86.76±1.75	77.32±9.26	82.44±4.60	86.70±1.26	87.78±1.90
	GPT-3.5	82.99±2.67	83.58±0.87	83.35±1.36	86.52±1.45	86.53±1.43	86.33±2.35	89.06±0.69	88.94±0.75

Table 8: Detailed results under few-shot settings of the SA task. Accuracy and standard deviations are reported.

Backbone	Aug. Method	SNLI				MNLI			
		5-shot	10-shot	20-shot	50-shot	5-shot	10-shot	20-shot	50-shot
GPT-3.5	None	70.60	69.60	70.40	69.60	70.00	69.00	68.00	70.60
BERT	None	37.67±1.29	38.77±1.23	51.09±1.75	56.37±1.44	35.03±1.72	32.40±1.00	41.67±1.31	45.06±1.84
	AutoCAD	35.61±2.00	37.57±1.47	53.57±1.27	55.87±1.16	33.53±1.16	34.24±1.30	41.40±1.25	46.48±1.45
	GPT-3.5	41.81±1.79	45.11±0.69	52.52±1.45	59.91±0.96	42.28±1.58	44.72±1.94	50.83±0.94	52.73±1.23
BART	None	36.15±1.43	35.74±0.48	41.42±1.70	49.47±2.76	34.13±0.62	33.47±0.60	36.93±0.54	37.47±1.04
	AutoCAD	37.09±2.57	38.19±1.96	51.20±2.93	55.17±0.93	33.19±0.42	32.76±0.23	39.08±0.96	44.48±1.18
	GPT-3.5	47.10±0.48	49.27±1.16	56.50±1.04	60.72±1.51	46.39±1.19	47.47±1.50	51.66±0.92	54.22±0.84

Table 9: Detailed results under few-shot settings of the NLI task. Accuracy and standard deviations are reported.

Backbone	Aug. Method	CoNLL03				OntoNotes			
		5-shot	10-shot	20-shot	50-shot	5-shot	10-shot	20-shot	50-shot
GPT-3.5	None	70.46	70.54	71.80	69.75	42.67	42.88	45.35	44.00
BERT	None	40.62±2.23	56.53±1.81	66.48±1.21	75.91±0.83	45.98±1.52	58.11±0.83	62.40±0.90	68.03±1.03
	CFGen	47.79±1.94	58.50±0.92	69.88±2.49	76.18±1.55	50.50±1.08	58.49±1.22	65.06±0.32	67.37±0.82
	GPT-3.5	49.13±2.17	62.14±1.45	73.00±0.66	77.40±0.89	52.42±1.59	59.17±1.62	63.19±0.98	67.69±0.51
BART	None	35.26±3.52	41.50±2.20	50.29±1.55	59.83±0.87	26.79±1.06	38.07±0.76	45.10±1.09	50.95±0.32
	CFGen	37.73±0.99	44.23±2.00	55.00±1.03	59.93±1.01	29.24±0.99	39.47±1.54	45.70±0.51	50.80±0.25
	GPT-3.5	38.73±1.61	45.47±1.63	53.07±1.83	59.70±1.35	26.89±0.89	38.07±0.42	43.65±1.04	50.44±0.58

Table 10: Detailed results under few-shot settings of the NER task. micro-F1 scores and standard deviations are reported.

Backbone	Aug. Method	SemEval				TACRED			
		5-shot	10-shot	20-shot	50-shot	5-shot	10-shot	20-shot	50-shot
GPT-3.5	None	45.80	47.00	44.40	44.00	23.62	20.91	18.90	23.62
BERT	None	30.49±1.57	41.69±0.69	52.40±1.13	64.06±1.18	10.38±0.77	17.17±1.12	22.99±0.61	27.69±1.99
	CoCo	30.49±1.57	41.86±0.67	52.48±0.96	68.52±0.86	12.27±0.57	18.60±2.03	23.88±0.46	29.82±2.75
	GPT-3.5	28.89±1.22	41.51±0.99	55.65±4.12	62.02±4.66	7.80±0.59	15.24±1.45	15.83±3.27	20.86±5.02
BART	None	36.14±2.45	44.08±1.38	54.16±1.15	67.80±1.57	19.51±0.97	24.69±1.11	28.05±1.15	30.82±1.13
	CoCo	36.14±2.45	44.71±1.11	55.07±1.73	69.88±0.75	20.83±0.78	25.46±0.89	28.31±0.79	30.29±1.06
	GPT-3.5	41.25±1.27	46.62±2.09	57.16±2.23	67.16±1.05	18.36±0.67	24.45±0.58	26.85±1.41	29.30±1.02

Table 11: Detailed results under few-shot settings of the RE task. micro-F1 scores and standard deviations are reported.

Shots	Aug. Method	SA		NLI		NER		RE	
		SST-2	IMDB	SNLI	MNLI	CoNLL03	OnotoNotes	SemEval	TACRED
5-shot	**GPT-3.5 w/ Instructions**	83.60±1.32	57.48±2.40	41.81±1.79	42.28±1.58	49.13±2.17	52.42±1.59	29.89±1.22	7.80±0.59
	GPT-3.5 w/o Instructions	82.86±1.91	56.49±3.14	40.85±1.40	37.25±1.58	45.39±2.05	48.19±0.99	30.70±1.57	9.05±0.70
10-shot	**GPT-3.5 w/ Instructions**	84.29±0.25	66.98±2.00	45.11±0.69	44.72±1.94	62.14±1.45	59.17±1.62	41.51±0.99	15.24±1.45
	GPT-3.5 w/o Instructions	84.63±0.24	66.11±2.08	47.17±0.73	42.89±2.75	61.32±1.05	58.96±1.25	41.77±0.66	16.66±1.41
20-shot	**GPT-3.5 w/ Instructions**	85.58±0.49	70.09±0.80	52.52±1.45	50.83±0.94	73.00±0.66	63.19±0.98	55.65±4.12	15.83±3.27
	GPT-3.5 w/o Instructions	85.21±0.48	67.25±1.97	50.99±0.72	52.04±0.75	68.73±1.54	62.05±1.06	49.72±4.56	22.79±1.49
50-shot	**GPT-3.5 w/ Instructions**	85.98±0.60	72.33±0.25	59.91±0.96	52.73±1.23	77.40±0.89	67.69±0.51	62.02±4.66	20.86±5.02
	GPT-3.5 w/o Instructions	85.68±1.46	71.73±0.57	56.62±1.53	51.47±0.98	76.17±0.49	67.82±0.72	64.91±0.40	23.83±4.10

Table 12: Detailed results under few-shot settings for the analysis on the impact of removing instructions. Accuracy or micro-F1 scores and standard deviations are reported. Here BERT is used as the SLMs backbone.

Shots	Aug. Method	SA		NLI		RE	
		SST-2	IMDB	SNLI	MNLI	SemEval	TACRED
5-shot	**GPT-3.5 w/o CoT**	83.60±1.32	57.48±2.40	41.81±1.79	42.28±1.58	29.89±1.22	7.80±0.59
	GPT-3.5 w/ CoT	82.10±2.44	58.14±3.31	34.67±1.91	36.44±1.36	31.40±1.81	7.79±0.44
10-shot	**GPT-3.5 w/o CoT**	84.29±0.25	66.98±2.00	45.11±0.69	44.72±1.94	41.51±0.99	15.24±1.45
	GPT-3.5 w/ CoT	82.80±0.95	61.15±5.05	49.82±1.66	37.05±1.59	41.26±0.91	15.68±1.69
20-shot	**GPT-3.5 w/o CoT**	85.58±0.49	70.09±0.80	52.52±1.45	50.83±0.94	55.65±4.12	15.83±3.27
	GPT-3.5 w/ CoT	85.64±0.59	70.63±0.89	54.05±2.07	46.23±1.70	55.02±1.00	21.22±0.79
50-shot	**GPT-3.5 w/o CoT**	85.98±0.60	72.33±0.25	59.91±0.96	52.73±1.23	62.02±4.66	20.86±5.02
	GPT-3.5 w/ CoT	85.68±0.62	71.32±0.93	58.08±1.96	48.99±0.63	66.05±0.47	19.66±0.90

Table 13: Detailed results under few-shot settings for the analysis on the impact of using chain-of-thought. Accuracy or micro-F1 scores and standard deviations are reported. Here BERT is used as the SLMs backbone.

Shots	Aug. Method	SA		NLI		NER		RE	
		SST-2	IMDB	SNLI	MNLI	CoNLL03	OnotoNotes	SemEval	TACRED
5-shot	**GPT-3.5 w/ Reason. Demo**	83.60±1.32	57.48±2.40	41.81±1.79	42.28±1.58	49.13±2.17	52.42±1.59	29.89±1.22	7.80±0.59
	GPT-3.5 w/ Unreason. Demo	82.15±1.57	56.85±3.90	39.61±1.54	40.54±2.08	52.80±1.00	52.38±0.83	31.27±1.80	8.16±0.85
10-shot	**GPT-3.5 w/ Reason. Demo**	84.29±0.25	66.98±2.00	45.11±0.69	44.72±1.94	62.14±1.45	59.17±1.62	41.51±0.99	15.24±1.45
	GPT-3.5 w/ Unreason. Demo	83.66±1.05	63.28±2.36	38.40±1.98	45.69±1.83	60.95±2.83	59.48±1.63	40.18±0.57	17.54±2.37
20-shot	**GPT-3.5 w/ Reason. Demo**	85.58±0.49	70.09±0.80	52.52±1.45	50.83±0.94	73.00±0.66	63.19±0.98	55.65±4.12	15.83±3.27
	GPT-3.5 w/ Unreason. Demo	85.34±0.62	67.45±0.73	46.91±1.86	50.75±1.52	69.11±4.06	62.96±1.1	49.94±1.02	17.39±1.52
50-shot	**GPT-3.5 w/ Reason. Demo**	85.98±0.60	72.33±0.25	59.91±0.96	52.73±1.23	77.40±0.89	67.69±0.51	62.02±4.66	20.86±5.02
	GPT-3.5 w/ Unreason. Demo	84.81±0.38	72.51±0.44	51.86±1.59	52.20±0.95	77.32±0.66	67.16±0.42	62.38±0.45	24.33±3.64

Table 14: Detailed results under few-shot settings for the analysis on the impact of providing unreasonable demonstration. Accuracy or micro-F1 scores and standard deviations are reported. Here BERT is used as the SLMs backbone.

Task Definition: Revise a given sentence with minimal changes to alter its sentiment polarity. Instruction: This process consists of two steps. The first step is to identify the words in the given sentence that have the highest potential to change the sentiment polarity after substitution, known as the causal words. The second step is to select appropriate replacement words for the causal words that will change the sentiment polarity of the sentence to the desired polarity. Demonstration: Given Sentence: "The movie is the best that I have ever seen." Current Sentiment Polarity: "positive" Target Sentiment Polarity: "negative" Revised Sentence: "The movie is the baddest that I have ever seen." Based on the given task definition and instruction, complete the following text by imitating the given demonstration. Given Sentence: "but it also has many of the things that made the first one charming ." Current Sentiment Polarity: "positive" Target Sentiment Polarity: "negative"

Table 15: Prompts for counterfactual generation for the SA task.

Task Definition: Revise a given sentence with minimal changes to alter its sentiment polarity. Instruction: This process consists of two steps. The first step is to identify the words in the given sentence that have the highest potential to change the sentiment polarity after substitution, known as the causal words. The second step is to select appropriate replacement words for the causal words that will change the sentiment polarity of the sentence to the desired polarity. Demonstration: Given Sentence: "The movie is the best that I have ever seen." Current Sentiment Polarity: "positive" Causal Words Identification: The sentiment polarity "positive" depends on words "best". Target Sentiment Polarity: "negative" Causal Words Replacement: To change the sentiment polarity of the given sentence from "positive" to "negative", causal words "best" are replaced by "baddest". Revised Sentence: "The movie is the baddest that I have ever seen." Based on the given task definition and instruction, complete the following text by imitating the given demonstration. Given Sentence: "This movie could not satisfy you." Current Sentiment Polarity: "negative" Causal Words Identification: Target Sentiment Polarity: "positive" Causal Words Replacement: Revised Sentence:

Table 16: Prompts for counterfactual generation for the SA task (with chain-of-thought).

Task Definition: Revise a given sentence with minimal changes to alter its sentiment polarity. Instruction: This process consists of two steps. The first step is to identify the words in the given sentence that have the highest potential to change the sentiment polarity after substitution, known as the causal words. The second step is to select appropriate replacement words for the causal words that will change the sentiment polarity of the sentence to the desired polarity. Demonstration: Given Sentence: "The movie is the best that I have ever seen." Current Sentiment Polarity: "positive" Target Sentiment Polarity: "negative" Revised Sentence: "The movie is the most wonderful that I have ever seen." Based on the given task definition and instruction, complete the following text by imitating the given demonstration. Given Sentence: "but it also has many of the things that made the first one charming ." Current Sentiment Polarity: "positive" Target Sentiment Polarity: "negative" Revised Sentence:

Table 17: Prompts for counterfactual generation for the SA task (with unresonable demonstration).

Task Definition: Revise the hypothesis sentence, using minimal changes, to alter the relationship between it and the premise sentence to either entailment, contradiction, or neutral. Instruction: This process consists of two steps. The first step is to identify the words in the given hypothesis sentence that have the highest potential to change the relationship with the premise sentence after substitution, known as the causal words. The second step is to select appropriate replacement words for the causal words that will change the relationship with the premise sentence to the desired relationship, either entailment, contradiction, or neutral. Demonstration: Given Premise Sentence: "A group of men riding bicycles in a line." Given Hypothesis Sentence: "The men riding together." Current Relationship between the premise sentence and the hypothesis sentence: "Entailment" Target Relationships: ["Contradiction","Neutral"] Generated Hypothesis Sentences: [{"target_relationship":,"Contradiction","revised_sentence":"The men riding horses."}, {"target_relationship":,"Neutral","revised_sentence":"The men are professionals."}] Based on the given task definition and instruction, complete the following text by imitating the given demonstration. Given Premise Sentence: "A group of young girls in a fenced in area." Given Hypothesis Sentence: "a group of sisters playing" Current Relationship between the premise sentence and the hypothesis sentence: "Neutral" Target Relationships: ["Entailment","Contradiction"] Generated Hypothesis Sentences:

Table 18: Prompts for counterfactual generation for the NLI task.

Task Definition: Revise the hypothesis sentence, using minimal changes, to alter the relationship between it and the premise sentence to either entailment, contradiction, or neutral.
Instruction: This process consists of two steps.
The first step is to identify the words in the given hypothesis sentence that have the highest potential to change the relationship with the premise sentence after substitution, known as the causal words. The second step is to select appropriate replacement words for the causal words that will change the relationship with the premise sentence to the desired relationship, either entailment, contradiction, or neutral.
Demonstration:
Given Premise Sentence: "A group of men riding bicycles in a line."
Given Hypothesis Sentence: "The men riding together."
Current Relationship between the premise sentence and the hypothesis sentence: "Entailment"
Causal Words Identification: The relationship "Entailment" depends on words "riding together" in the hypothesis sentence.
Target Relationship: "Contradiction"
Causal Words Replacement: To change the relationship between the premise sentence and the hypothesis sentence from "Entailment" to "Contradiction", causal words "riding together" are replaced by "riding horses".
Revised Sentence: "The men riding horses."
Target Relationship: "Neutral"
Causal Words Replacement: To change the relationship between the premise sentence and the hypothesis sentence from "Entailment" to "Neutral", causal words "riding together" are replaced by "are professionals".
Revised Sentence: "The men are professionals."
Based on the given task definition and instruction, complete the following text by imitating the given demonstration.
Given Premise Sentence: "A group of young girls in a fenced in area."
Given Hypothesis Sentence: "a group of sisters playing"
Current Relationship between the premise sentence and the hypothesis sentence: "Neutral"
Causal Words Identification:
Target Relationship: "Entailment"
Causal Words Replacement:
Revised Sentence:
Target Relationship: "Contradiction"
Causal Words Replacement:
Revised Sentence:

Table 19: Prompts for counterfactual generation for the NLI task (with chain-of-thought).

Task Definition: Revise the hypothesis sentence, using minimal changes, to alter the relationship between it and the premise sentence to either entailment, contradiction, or neutral.
Instruction: This process consists of two steps.
The first step is to identify the words in the given hypothesis sentence that have the highest potential to change the relationship with the premise sentence after substitution, known as the causal words. The second step is to select appropriate replacement words for the causal words that will change the relationship with the premise sentence to the desired relationship, either entailment, contradiction, or neutral.
Demonstration:
Given Premise Sentence: "A group of men riding bicycles in a line."
Given Hypothesis Sentence: "The men riding together."
Current Relationship between the premise sentence and the hypothesis sentence: "Entailment"
Target Relationships: ["Contradiction","Neutral"]
Generated Hypothesis Sentences: [{"target_relationship":,"Contradiction","revised_sentence":"The men riding bicycles."}, {"target_relationship":,"Neutral","revised_sentence":"The men are riding horses."}]
Based on the given task definition and instruction, complete the following text by imitating the given demonstration.
Given Premise Sentence: "A group of young girls in a fenced in area."
Given Hypothesis Sentence: "a group of sisters playing"
Current Relationship between the premise sentence and the hypothesis sentence: "Neutral"
Target Relationships: ["Entailment","Contradiction"]
Generated Hypothesis Sentences:

Table 20: Prompts for counterfactual generation for the NLI task (with unresonable demonstration).

Task definition: Generate words that can replace entities in the given sentence, whose type is the same as the original
entity type, and refer to the demonstration for the output format.
Demonstration:
Given Sentence: "Apple was founded in 1978."
Given Entities: [{"entity_span":"Apple","entity_type":"organization"},{"entity_span":"1978","entity_type":"date"}]
Replaceable Entity Words: [{"entity_span":"Apple","entity_type":"organization","replaceable_entity_words":["Google","OpenAI","Microsoft"]},
{"entity_span":"1978","entity_type":"date"},"replaceable_entity_words":["1978","1890","March"]]
Based on the given task definition and instruction, complete the following text by imitating the given demonstration.
Given Sentence: "Cargill thinks that even though the merchant has a contract stating that it wo n't bring this cocoa to
market until after March 1991 , there is some evidence the contract has been modified ."
Given Entities: [{"entity_span":"Cargill","entity_type":"PERSON"}, {"entity_span":"March 1991","entity_type":"DATE"}]
Replaceable Entity Words:

Table 21: Prompts for counterfactual generation for the NER task.

Task Definition: Generate words that can replace entities in the given sentence, whose type is the same as the original
entity type, and refer to the demonstration for the output format.
Demonstration:
Given Sentence: "Apple was founded in 1978."
Given Entities: [{"entity_span":"Apple","entity_type":"organization"},{"entity_span":"1978","entity_type":"date"}]
Replaceable Entity Words: [{"entity_span":"Apple","entity_type":"organization","replaceable_entity_words":["1970","1878","March"]},
{"entity_span":"1978","entity_type":"date","replaceable_entity_words":["Google","Microsoft","OpenAI"]}]
Based on the given task definition and instruction, complete the following text by imitating the given demonstration.
Given Sentence: "- Prime minister names former general Avraham Tamir to staff after failing to establish national security council ."
Given Entities: [{"entity_span":"Avraham Tamir","entity_type":"PER"}]
Replaceable Entity Words:

Table 22: Prompts for counterfactual generation for the NER task (with unresonable demonstration).

Task Definition: Revise a given sentence with minimal changes to change the relation between the head and tail entity.
Instruction: This process involves three steps. The first step is to identify context words (excluding entity words) in
the given sentence that are most likely to change the relation between the head and tail entity when replaced, known as
the causal words. The second step is to select a potential target relation from the candidate relation set, which must
conform the relevant commonsense of head and tail entity. The third step is to replace the causal words with appropriate
words to change the original relation into potential target relations.
Note: The found potential target relation must belong to the candidate relation set. If there are no potential target
relation that conforms the commonsense, just output None.
Demonstration: Given Sentence: "the key is moved into a chest."
Head entity: "key"
Tail entity: "chest"
Relation between the Head and Tail entity: "entity-destination"
Candidate Relation Set: {"message-topic", "topic-message", "destination-entity", "content-container", "container-content",
"effect-cause", "cause-effect", "whole-component", "component-whole", "collection-member", "member-collection",
"agency-instrument", "instrument-agency", "producer-product", "product-producer", "entity-origin", "origin-entity"}
Revised Sentence: {"target_relation":"entity-origin","revised_sentence":"the key is from a chest."}
Based on the given task definition and instruction, complete the following text by imitating the given demonstration.
Given Sentence: "The series comprises some re-issues of the previous books , as well as new titles ."
Head entity: "titles"
Tail entity: "series"
Relation between the Head and Tail entity: "Component-Whole"
Candidate Relation Set: {"Instrument-Agency", "Member-Collection", "Cause-Effect", "Entity-Destination", "Content-Container",
"Message-Topic", "Product-Producer", "Entity-Origin", "Whole-Component", "Agency-Instrument", "Collection-Member",
"Effect-Cause", "Destination-Entity", "Container-Content", "Topic-Message", "Producer-Product", "Origin-Entity", "Other"}
Revised Sentence:

Table 23: Prompts for counterfactual generation for the RE task.

Task Definition: Revise a given sentence with minimal changes to change the relation between the head and tail entity. Instruction: This process involves three steps. The first step is to identify context words (excluding entity words) in the given sentence that are most likely to change the relation between the head and tail entity when replaced, known as the causal words. The second step is to select a potential target relation from the candidate relation set, which must conform the relevant commonsense of head and tail entity. The third step is to replace the causal words with appropriate words to change the original relation into potential target relations. Note: The found potential target relation must belong to the candidate relation set. If there are no potential target relation that conforms the commonsense, just output None. Demonstration: Given Sentence: "the key is moved into a chest." Head entity: "key" Tail entity: "chest" Relation between the Head and Tail entity: "entity-destination" Candidate Relation Set: {message-topic, topic-message, destination-entity, content-container, container-content, effect-cause, cause-effect, whole-component, component-whole, collection-member, member-collection, agency-instrument, instrument-agency, producer-product, product-producer, entity-origin, origin-entity} Causal Words Identification: The relation type "entity-destination" depends on contextual words "moved into". Potential Relation Discovery: The relation between "key" and "chest" can be "entity-origin". Causal Words Replacement: To change the relation between "key" and "chest" from "entity-destination" to "entity-origin", causal words "moved into" can be replaced by "from". Revised Sentence: {"target_relation":"entity-origin","revised_sentence":"the key is from a chest."} Based on the given task definition and instruction, complete the following text by imitating the given demonstration. Given Sentence: "Tom Thabane , who set up the All Basotho Convention four months ago , said his party would do more against the poverty that wracks the southern African nation ." Head entity: "All Basotho Convention" Tail entity: "Tom Thabane" Relation between the Head and Tail entity: "org:founded_by" Candidate Relation Set: {"Instrument-Agency", "Member-Collection", "Cause-Effect", "Entity-Destination", "Content-Container", "Message-Topic", "Product-Producer", "Entity-Origin", "Whole-Component", "Agency-Instrument", "Collection-Member", "Effect-Cause", "Destination-Entity", "Container-Content", "Topic-Message", "Producer-Product", "Origin-Entity", "Other"} Causal Words Identification: Potential Relation Discovery: Causal Words Replacement: Revised Sentence:

Table 24: Prompts for counterfactual generation for the RE task (with chain-of-thought).

Task Definition: Revise a given sentence with minimal changes to change the relation between the head and tail entity. Instruction: This process involves three steps. The first step is to identify context words (excluding entity words) in the given sentence that are most likely to change the relation between the head and tail entity when replaced, known as the causal words. The second step is to select a potential target relation from the candidate relation set, which must conform the relevant commonsense of head and tail entity. The third step is to replace the causal words with appropriate words to change the original relation into potential target relations. Note: The found potential target relation must belong to the candidate relation set. If there are no potential target relation that conforms the commonsense, just output None. Demonstration: Given Sentence: "the key is moved into a chest." Head entity: "key" Tail entity: "chest" Relation between the Head and Tail entity: "entity-destination" Candidate Relation Set: {"message-topic", "topic-message", "destination-entity", "content-container", "container-content", "effect-cause", "cause-effect", "whole-component", "component-whole", "collection-member", "member-collection", "agency-instrument", "instrument-agency", "producer-product", "product-producer", "entity-origin", "origin-entity"} Revised Sentence: {"target_relation":"topic-message","revised_sentence":"the key causes a chest."} Based on the given task definition and instruction, complete the following text by imitating the given demonstration. Given Sentence: "Fine workmanship is the result almost entirely of the worker 's accurate eye and deft hand ." Head entity: "eye" Tail entity: "worker" Relation between the Head and Tail entity: "Component-Whole" Candidate Relation Set: {"Instrument-Agency", "Member-Collection", "Cause-Effect", "Entity-Destination", "Content-Container", "Message-Topic", "Product-Producer", "Entity-Origin", "Whole-Component", "Agency-Instrument", "Collection-Member", "Effect-Cause", "Destination-Entity", "Container-Content", "Topic-Message", "Producer-Product", "Origin-Entity", "Other"} Revised Sentence:

Table 25: Prompts for counterfactual generation for the RE task (with unresonable demonstration).

CoNLL2003	
PER	person
LOC	location
ORG	organization
MISC	miscellaneous
OntoNotes	
CARDINAL	cardinal
DATE	date
EVENT	event
FAC	facility
GPE	country city state
LANGUAGE	language
LAW	law
LOC	location
MONEY	monetary
NORP	nationality religious political group
ORDINAL	ordinal
ORG	organization
PERCENT	percent
PERSON	person
PRODUCT	product
QUANTITY	quantity
TIME	time
WORK_OF_ART	work of art
SemEval	
Entity-Destination	destined for
Cause-Effect	lead to
Content-Container	contained in
Other	other
Entity-Origin	originated from
Member-Collection	belongs to
Product-Producer	made by
Component-Whole	composed of
Message-Topic	pertaining to
Instrument-Agency	performed by
TACRED	
per:age	age
org:founded	founded
per:date_of_birth	date of birth
per:country_of_birth	country of birth
org:alternate_names	alternate names
org:founded_by	founded by
per:cause_of_death	cause of death
org:country_of_headquarters	country of headquarters
per:alternate_names	alternate names
org:members	members
per:cities_of_residence	cities of residence
org:city_of_headquarters	city of headquarters
org:political/religious_affiliation	political religious affiliation
per:employee_of	employee of
per:stateorprovinces_of_residence	state or provinces of residence
org:member_of	member of
org:stateorprovince_of_headquarters	state or province of headquarters
per:parents	parents
org:dissolved	dissolved
org:parents	parents
per:children	children
per:spouse	spouse
per:date_of_death	date of death
per:city_of_death	city of death
per:countries_of_residence	countries of residence
org:top_members/employees	top members employees
no_relation	no relation
per:title	title
per:schools_attended	schools attended
per:religion	religion
per:siblings	siblings
per:charges	charges
per:origin	origin
per:other_family	other family
per:stateorprovince_of_death	state or province of death
org:website	website
per:stateorprovince_of_birth	state or province of birth
org:shareholders	shareholders
org:subsidiaries	subsidiaries
per:city_of_birth	city of birth
org:number_of_employees/members	number of employees members
per:country_of_death	country of death

Table 26: The label-word mapping of NER and RE tasks for BART-based SLMs.