Counterfactual Generator: A Weakly-Supervised Method for Named Entity Recognition

Xiangji Zeng, Yunliang Li, Yuchen Zhai, Yin Zhang*
Zhejiang University, Hangzhou, China
`{zengxiangji,liyunliang,zhaiyuchen,zhangyin98}@zju.edu.cn`

Abstract

Past progress on neural models has proven that named entity recognition is no longer a problem if we have enough labeled data. However, collecting enough data and annotating them are labor-intensive, time-consuming, and expensive. In this paper, we decompose the sentence into two parts: entity and context, and rethink the relationship between them and model performance from a causal perspective. Based on this, we propose the Counterfactual Generator, which generates counterfactual examples by the interventions on the existing observational examples to enhance the original dataset. Experiments across three datasets show that our method improves the generalization ability of models under limited observational examples. Besides, we provide a theoretical foundation by using a structural causal model to explore the spurious correlations between input features and output labels. We investigate the causal effects of entity or context on model performance under both conditions: the non-augmented and the augmented. Interestingly, we find that the non-spurious correlations are more located in entity representation rather than context representation. As a result, our method eliminates part of the spurious correlations between context representation and output labels. The code is available at `https://github.com/xijiz/cfgen`.

1 Introduction

The natural language processing community has witnessed the paradigm shift from small data to big data, such as transformer (Vaswani et al., 2017) and its successors. It is not surprising that machine learning methods can easily surpass human performance if sufficient data is available (Wang et al., 2018). However, data acquisition is a challenging task for some special domains. For example, medical concept normalization, a basic subtask of named entity recognition (NER) in the medical area, has always been troubled by lack of enough Electronic Health Records due to the privacy protection. Small data with selection biases (Torralba and Efros, 2011) often induce the poor performance of machine learning models on inputs whose distribution is different from that of training data, which yet seems trivial to humans. The same issues are also mentioned in terms like *dataset bias*, *model robustness*, and *real understanding*. In natural language inference, models trained on hypotheses-only (vs hypotheses-premises) can outperform a majority-class baseline (Poliak et al., 2018; Gururangan et al., 2018). In reading comprehension, models trained on question-only or passage-only (vs question-passage) still achieve high accuracy (Kaushik and Lipton, 2018), models predicted on a broken question (vs original question) still make the same correct prediction (Feng et al., 2018a).

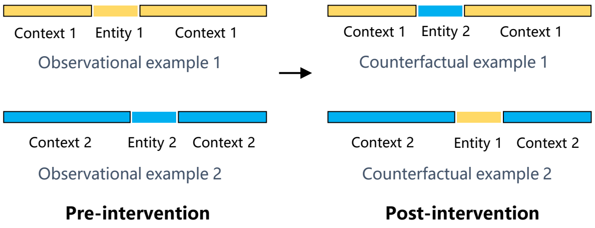


Figure 1: Interventions on observational examples of named entity recognition. More details can be found in Figure 3

The key challenge behind this phenomenon is caused by spurious correlations of statistical learning. Spurious correlations can be vividly explained by an example in computer vision (Arjovsky et al., 2019): If we consider an image dataset of cows and camels in their natural habitat, a classifier

---

[*] Corresponding Author

trained on this dataset will establish spurious correlations between the output labels (cows, camels) and the landscape of the image (green pastures, deserts). As a result, an image of cows taken on sandy beaches makes the classifier make a wrong prediction. In this background, we could not help thinking that *is there any way to eliminate spurious correlations except more data annotated by humans?* From a causal perspective, spurious correlations are caused by confounding factors rather than a direct or indirect causal path. If we directly intervene on the precursor variable in spurious correlations to create counterfactual data, we can eliminate the impact of spurious correlations in models to a certain degree (Volodin et al., 2020).

In this paper, based on the above analysis, we mainly focus on exploring the spurious correlations in NER from a causal perspective. We decompose the sentence into two different parts: entity and context, and rethink the relationship between them and the generalization ability of the NER model. Considering the sentence "John lives in New York", we observe that the location entity "New York" and the context "John lives in" are highly correlated but are not causal to each other. In other words, we can intervene on the location entity to set it to another different location entity without destroying the sentence correctness at the grammatical level.

Therefore, we propose the **Counterfactual Generator**, which generates new counterfactual examples by the interventions on the existing observational examples. Our method requires neither an additional entity dictionary nor a similar domain dataset. Figure 1 demonstrates the intervention process for observational examples. We utilize new counterfactual examples to enhance the existing observational examples. Experiments show that our method improves the generalization ability under limited observational examples. Before the enhancement, we find that the model performance is mainly driven by entity representation. After the enhancement, the importance of entity representation increases in most cases, and generalization ability improves in all cases. We conclude that the non-spurious correlations between input features and output labels do locate in context representation (Lin et al., 2020), but the previous two phenomena show that they are more located in entity representation.

In summary, our work have the following contributions:

- We provide a theoretical foundation from a causal perspective to describe the mechanism of the NER model inference and explore the spurious correlations between input features and output labels.
- Based on the interventions on the entity, we propose a weakly-supervised method for named entity recognition under limited observational examples. Experiments across three NER datasets demonstrate that our method boosts model performance.

2 Counterfactual Generator

In this section, we firstly define the NER problem. Then, we present our method by introducing a structural causal model to describe the mechanism of the NER model inference.

2.1 Task Definition

In this paper, we regard named entity recognition as a sequence labeling problem. In general, we let $\mathbf{x} = (x_1, x_2, ..., x_n)$ to denote a sequence of tokens. For each token x_i, we have a label y_i where $y_i \in \mathcal{Y}$. For example, $\mathcal{Y}$ can be $\{O, \textit{B-Diagnosis}, \textit{I-Diagnosis}\}$ in the medical area. The possible labels come from *BIO* tagging schema for labeling tokens from the sentence. For each sentence, we have an entity set $\mathcal{E}$ that contains all entities in this sentence. Finally, we have a labeled dataset $\mathcal{D} = \{(x, y)\}$.

2.2 Causal Model

What determines a certain segment in a sentence to be an entity mention? Why is this entity mention to be a diagnosis entity? These are causal questions because they require some information about the generation process of the data rather than observational data alone (Pearl et al., 2009). Observational data with selection biases often gives rise to the problem of the spurious correlations that results in low generalization ability of the NER model under limited data. For this problem, causality can provide an in-depth view of its essence.

To investigate the causal relationship between the NER model and data clearly, we introduce a Structural Causal Model (SCM) (Judea, 2000) to describe the mechanism of the inference process of the NER model. SCM is expressed visually by using directed acyclic graphs (DAGs). In the graph, vertices are random variables, and directed edges represent direct causation from variable A

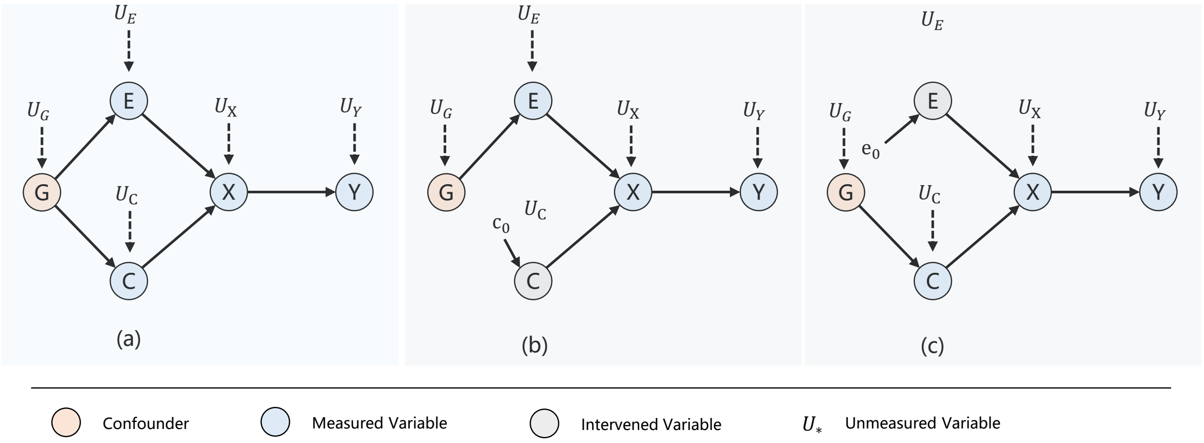


Figure 2: Structural Causal Models (SCMs) that describes the mechanism of the NER model inference. (a) Complete SCM without interventions. (b) Intervening on the variable ***C*** by the value c_0, denoted as $do(C = c_0)$. (c) Intervening on the variable *E* by the value e_0, denoted as $do(E = e_0)$.

to variable B. Here, for simplifying the problem, we decompose the sentence into the two variables: entity E and context C. As shown in Figure 2(a), we assume the following SCM:

$$\begin{aligned} g &:= f_G(U_G) \\ e &:= f_E(g, U_E) \\ c &:= f_C(g, U_C) \\ x &:= f_X(e, c, U_X) \\ y &:= f_Y(x, U_X) \end{aligned} \tag{1}$$

where G is a confounding variable that influences the generation of both entity E and context C, X is the input example that is generated by E and C, Y is the evaluation result (the $F1$ score) of the NER model, and U_* represents the unmeasured variable.

Causal effects help us better understand the causal relationship in a system. The basic method of estimating causal effects is simulating interventions in SCM. We use a mathematical operator $do(v_0)$ to simulate physical interventions by fixing the value of a variable v as v_0. For example, in order to simulate an intervention $do(c_0)$ in the structural causal model M, we fix the variable C to c_0 as shown in Figure 2(b), denoted as:

$$c := c_0 \tag{2}$$

This intervention blocks the influence of the variable G on the variable C. The post-intervention distribution $P(y|do(c_0))$ gives the proportion of individual that would attain response in level $Y = y$ under the hypothetical situation in which treatment $C = c_0$ is administered uniformly to the population (Pearl et al., 2009). Here, we have $P(y|do(c_0)) = 1$. More proof information can be found in the appendix.

A way to estimate the treatment effect or causal effect is to measure the average difference of the former distribution by using the expectation $\mathbf{E}$, called Average Causal Effect (ACE), denoted as:

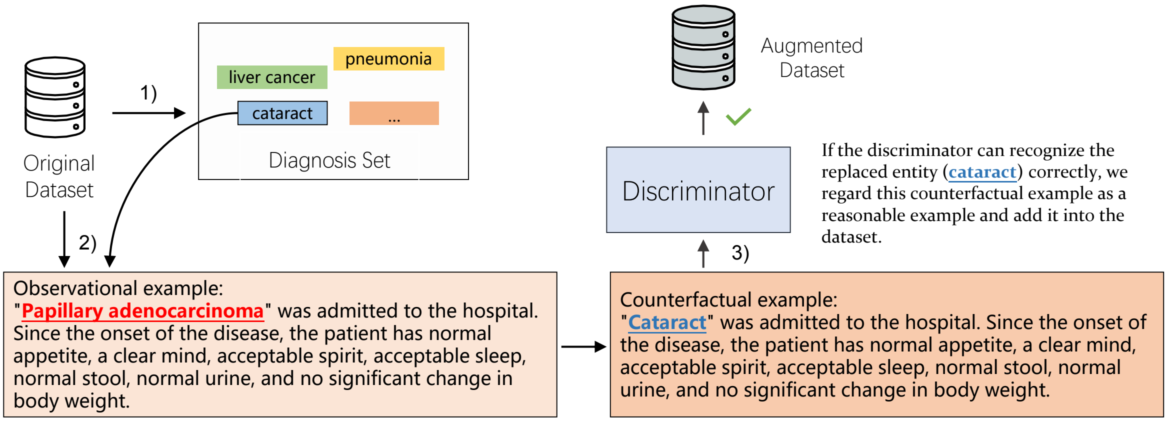
$$ACE_C = \mathbf{E}(y|do(c_0)) - \mathbf{E}(y|do(c)) \tag{3}$$

where c_0 and c are the intervened value and the original value. Similarly, in order to estimate the causal effects of the variable E on the variable Y, we can also intervene on the variable E, denoted as $do(e_0)$ (See Figure 2(c)).

2.3 Method

Our method tries to automatically replace an entity in an observational example with another different entity for creating a new counterfactual example. These counterfactual examples help our NER model deal with spurious correlations on limited observational examples and learn more invariant and stable features. As shown in Figure 3, our method mainly has the following three parts:

1) Set Preparation The core idea of our method is finding a different entity for intervening on an entity in the observational example. However, finding a new entity set in a specific domain needs human efforts to collect entities, which has no difference



If the discriminator can recognize the replaced entity (**cataract**) correctly, we regard this counterfactual example as a reasonable example and add it into the dataset.

Observational example:
"**Papillary adenocarcinoma**" was admitted to the hospital. Since the onset of the disease, the patient has normal appetite, a clear mind, acceptable spirit, acceptable sleep, normal stool, normal urine, and no significant change in body weight.

Counterfactual example:
"**Cataract**" was admitted to the hospital. Since the onset of the disease, the patient has normal appetite, a clear mind, acceptable spirit, acceptable sleep, normal stool, normal urine, and no significant change in body weight.

Figure 3: An example of the workflow of the Counterfactual Generator on the medical dataset. 1) We prepare the entity sets by the entity type (*diagnosis*) from the original dataset. 2) We randomly choose an entity (*papillary adenocarcinoma*) in the observational example and replace it with another different entity (*cataract*) from the entity set to form a new counterfactual example. It is noteworthy that the replaced entity and the candidate entity have the same entity type. 3) We send the counterfactual example to the discriminator for finding out the good one.

from annotating more data. Hence, as shown in Figure 3(1), we adopt local entities as the entity set, which is extracted from the original dataset. For example, we iterate all observational examples in the training dataset to collect all diagnoses to form a diagnosis set $\mathcal{E}_d$.

2) Entity Intervention We consider using the intervention on the entity to create new counterfactual examples. As shown in Figure 2(c) and Figure 3(2), for each observational example, we randomly select an entity $e \in \mathcal{E}$ with the entity type *diagnosis*, and replace it with another entity $e^{'} \in \mathcal{E}_d$. Importantly, in order to preserve the linguistic correctness of the new counterfactual example, we keep the replaced entity and the candidate entity have the same entity type.

3) Example Discrimination A key conflict is that not all counterfactual examples are correct or useful. We need a mechanism to discriminate which counterfactual example is good and make sure it does not bring in the noise. An intuitive solution is that we regard the NER model trained on the original dataset as the discriminator that provides well prior knowledge for inspecting our counterfactual examples. More specifically, as shown in Figure 3(3), the discriminator assists us to check whether the replaced entity is successfully predicted. If no, the counterfactual example will be discarded, otherwise, it will be outputted.

After executing all procedures, we have an augmented dataset that mixes observational examples and counterfactual examples. Afterwards, we can train the NER model on the augmented dataset.

Dataset	Train	Dev	Test	Total
CNER	1322	164	164	1650
IDiag	9274	1157	1157	11588
CLUENER	9674	1208	1208	12090

Table 1: Statistics of datasets: CNER, IDiag, and CLUENER. All datasets are divided into three parts of *train set* (80%), *dev set* (10%), and *test set* (10%).

3 Experiments

In this section, we mainly evaluate our method across three NER datasets, including two medical concept recognition datasets, and a conventional NER dataset.

3.1 Dataset

CNER[1] CNER is a Chinese clinical NER dataset in the CCKS-2019 challenge, including anatomy, disease, imaging examination, laboratory examination, drug, and operation. We extract 1650 available medical records from CNER, which contains entities of the disease type only.

IDiag For guaranteeing the diversity of the experimental data, we use Label Studio[2] to create a new medical NER dataset. We collect 12127 health record images from the hospital, which are converted into text paragraphs by optical character recognition (OCR). We hire some people to annotate diagnoses in these text paragraphs. To ensure the high quality of the dataset, we removed 539 data examples in the final dataset. It is worth noting that the distribution of IDiag, compared to CNER, has a big difference due to error text recognition from OCR.

---

[1] http://www.ccks2019.cn/?page_id=62

[2] https://labelstud.io/

CLUENER (Xu et al., 2020) In addition to the medical NER datasets, we also use a conventional NER dataset CLUENER released by CLUE organization[3], which is a well-defined and fine-grained dataset for named entity recognition in Chinese, including 10 categories like *Person Name*, *Organzation*, *Book*, etc. We extract 12090 available instances from this dataset.

All datasets are separately divided into three portions of 80% D_1, 10% D_2 and 10% D_3. D_1 is used to train models. D_2 is used to tune hyperparameters. D_3 is used to test the model performance (See Table 1).

3.2 Models

We conduct our experiments by using the following two classic models: LSTMTagger (Chiu and Nichols, 2016) and BERTTagger (Devlin et al., 2019). Our LSTMTagger consists of a bidirectional LSTM for encoding the input example and a dense layer (Tagger) for tagging all tokens. Each token is embedded by the pretrained word embedding (Song et al., 2018). Similarly, our BERTTagger consists of a pretrained BERT for encoding the input example and a dense layer (Tagger) for tagging all tokens.

3.3 Setup

In our experiments, we evaluate our method in two settings: *NoAug* and *Aug*. *NoAug* represents we train our models on the original dataset. *Aug* represents we train our models on the augmented dataset. We also set up five groups of experiments for each dataset. In each group, we only select N (*100, 200, 300, 400, and 500*) data from the *train set* to train models for evaluating performance under limited observational examples. At the same time, we always keep the *dev set* and *test set* unchanged in all experiments.

Additionally, we also conduct another experiment to calculate ACE of entity E or context C on the model performance Y. We design a special token [EMPTY] to replace tokens in entity E and tokens in context C separately, which is viewed as interventions. Once a token is replaced with [EMPTY] in an input example, all dimensions of token embedding will be zero. There are two intervened schemes corresponding to Figure 2(b) and Figure 2 (c) respectively, denoted as:

- $do(e_0)$ Replacing all tokens in entities with [EMPTY] and keeping context unchanged.
- $do(c_0)$ Replacing all tokens in the input example with [EMPTY] except tokens in entities.

3.4 Metrics

3.4.1 NER Evaluation

In this work, we mainly consider the performance at the entity level, which means the ground truth and the result have the same entity type and overlap boundaries are just taken into account. Hence, we use the relaxed metrics (Chinchor and Sundheim, 1993): micro-average F1 score (*F1*), precision (*P*), and recall (*R*). Besides, we also use the micro-average *F1* score at the token level for the later causal analysis, which evaluates predictions only by tokens.

3.4.2 RI Index

We design an index to evaluate the Relative Importance (RI) between entity E and context C, denoted as:

$$RI = ACE_C - ACE_E \tag{4}$$

This index indicates that the higher the RI is, the more important the entity representation is during the process of the model inference. Otherwise, the representation of context is more important. For example, they have the same importance when $RI = 0$. We adopt two different ways (Entity Level and Token Level) to calculate the variable Y (the $F1$ score) for attaining both the coarse-grained and the fine-grained results.

3.5 Main Results

As we can see, table 2 shows the comparisons between *NoAug* and *Aug*. We can see that our method achieves a huge improvement in almost all settings. For CNER, our method achieves the best results and yields a boost of 8.68% on average. Even

---

[3] https://www.cluebenchmarks.com/

CNER (%)

N	LSTM (*NoAug*)			LSTM (*Aug*)			BERT (*NoAug*)			BERT (*Aug*)		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
100	43.8	40.1	48.3	55.6 **(+11.8)**	51.0	61.0	47.2	43.2	52.1	47.6 **(+0.4)**	41.1	56.6
200	49.3	46.5	52.4	64.1 **(+14.8)**	60.8	67.8	57.1	52.4	62.7	68.3 **(+11.1)**	61.1	77.3
300	50.9	49.3	52.6	66.3 **(+15.4)**	61.4	71.9	61.1	53.8	70.8	71.4 **(+10.3)**	65.8	78.1
400	58.9	57.6	60.3	67.7 **(+8.8)**	65.8	69.7	73.7	70.2	77.5	77.4 **(+3.7)**	74.3	80.7
500	64.0	59.6	69.1	70.5 **(+6.5)**	67.4	74.0	74.9	71.4	78.7	78.8 **(+4.0)**	75.8	82.1

IDiag (%)

N	LSTM (*NoAug*)			LSTM (*Aug*)			BERT (*NoAug*)			BERT (*Aug*)		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
100	55.3	52.6	58.4	62.1 **(+6.8)**	57.3	67.7	58.3	52.4	65.8	67.9 **(+9.6)**	61.9	75.2
200	64.0	61.1	67.2	68.0 **(+4.0)**	64.0	72.4	67.9	63.4	73.2	72.6 **(+4.7)**	68.6	77.0
300	66.6	64.2	69.2	72.6 **(+6.0)**	69.7	75.7	71.8	67.8	76.5	76.1 **(+4.2)**	72.2	80.4
400	68.8	66.3	71.6	73.9 **(+5.1)**	70.5	77.6	73.7	70.2	77.5	77.4 **(+3.7)**	74.3	80.7
500	70.9	68.5	73.6	74.8 **(+3.9)**	72.6	77.2	74.9	71.4	78.7	78.8 **(+4.0)**	75.8	82.1

CLUENER (%)

N	LSTM (*NoAug*)			LSTM (*Aug*)			BERT (*NoAug*)			BERT (*Aug*)		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
100	7.8	6.5	9.8	11.3 **(+3.5)**	9.9	13.2	30.2	25.9	36.3	34.8 **(+4.6)**	27.8	46.4
200	15.2	13.5	17.4	20.4 **(+5.2)**	17.8	23.8	43.8	36.5	54.7	48.5 **(+4.7)**	40.9	59.6
300	19.4	17.1	22.4	23.4 **(+4.0)**	20.3	27.6	47.0	39.3	58.5	53.0 **(+6.0)**	46.3	61.8
400	21.8	18.1	27.6	26.7 **(+4.9)**	24.8	28.9	51.5	44.8	60.6	55.2 **(+3.7)**	47.9	65.2
500	25.4	23.1	28.1	29.1 **(+3.8)**	26.4	32.6	53.3	46.8	61.8	57.1 **(+3.9)**	50.2	66.3

Table 2: Results on datasets: CNER, IDiag, and CLUENER. N represents the number of training examples. *Aug* and *NoAug* represent whether we use our method for the augmentation or not.

Dataset	**ACE at Entity Level**						**ACE at Token Level**					
	LSTMTagger			BERTTagger			LSTMTagger			BERTTagger		
	ACE_C	ACE_E	RI	ACE_C	ACE_E	RI	ACE_C	ACE_E	RI	ACE_C	ACE_E	RI
CNER	-0.40	-0.49	0.08	-0.21	-0.71	0.50	-0.33	-0.66	0.33	-0.39	-0.85	0.46
IDiag	-0.28	-0.56	0.28	-0.04	-0.76	0.72	-0.20	-0.53	0.33	-0.11	-0.91	0.81
CLUENER	-0.20	-0.51	0.31	-0.14	-0.68	0.54	-0.16	-0.65	0.49	-0.23	-0.80	0.57

Table 3: Average Causal Effect (ACE) of entity E or context C on the model performance Y. Entity Level and Token Level are two different evaluation methods for the NER model (See Section 3.4). ACE_C denotes the ACE when intervening on the variable C by c_0. ACE_E denotes the ACE when intervening on the variable E by e_0. The RI index denotes the difference between ACE_C and ACE_E, which indicates the relative importance between entity representation and context representation. The higher the RI index, the more important the entity representation is during the process of the model inference.

for IDiag that has much noise, our method still achieves a huge gain of +5.2% improvement on average. We notice that our method only achieves a boost of 4.43% on average for CLUENER. Compared to the former two datasets, the reason for a lesser performance boost is that CLUENER contains more entity types than CNER and IDiag (10 vs 1, 1).

Table 3 demonstrates the ACE results for different combinations between datasets, models, and evaluation ways in *test set* without augmentation. Interestingly, we check the RI index and observe that the importance of entity representation is always far greater than the importance of representation of context.

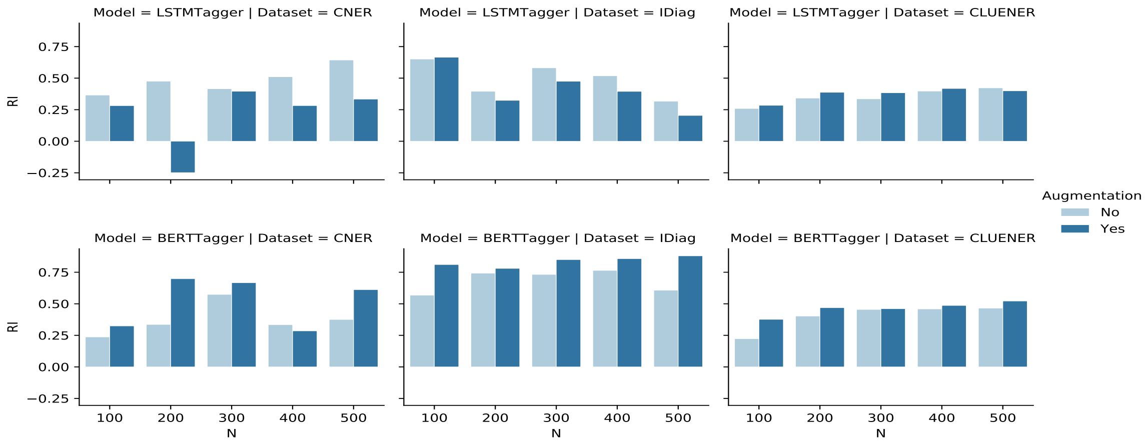


Figure 4: RI Index Changes between the non-augmented and the augmented. Different rows represent different models (LSTMTagger, BERTTagger); Different columns represent different datasets (CNER, IDiag, and CLUENER); N denotes the number of examples taken out from the *train set*.

4 Discussion

In this section, we will firstly review our previous results, and then try to answer some potential questions that others may ask for a deep understanding of our method. Secondly, we will provide some real counterfactual examples to vividly illustrate our method. Finally, some limitations of our method that we have found so far are presented to guide future research. We hope these limitations can help readers understand our method better.

4.1 Analysis

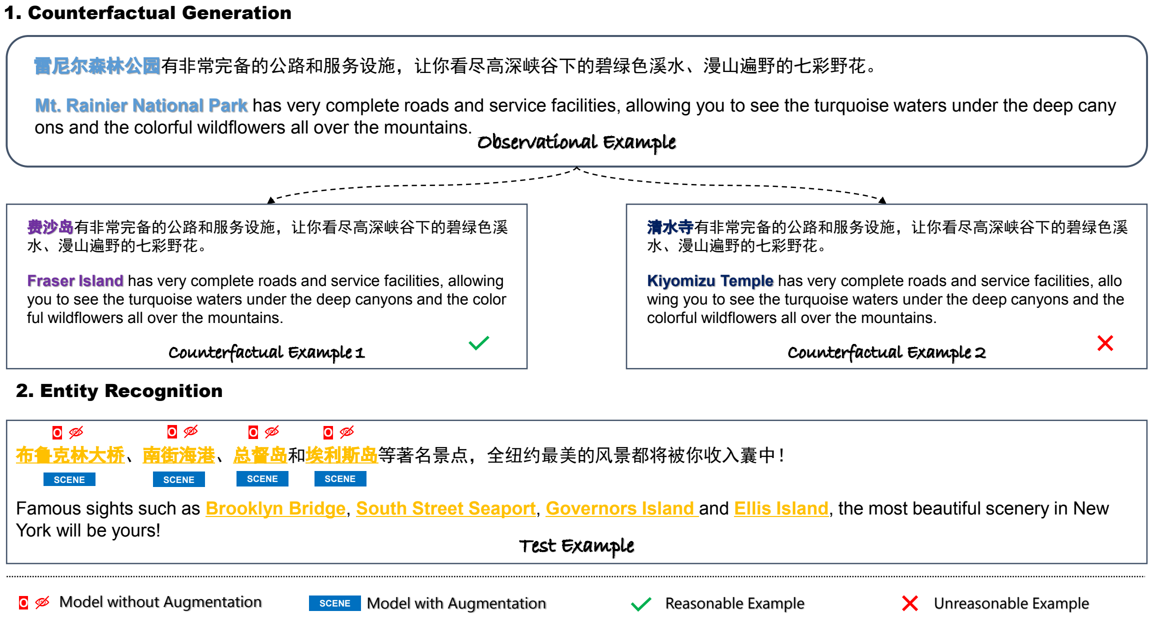
Our method achieves significant improvements across three datasets, but there are always a few mysteries that haunt us. **Q1**: *Do counterfactual examples change the causal effect between entity E and context C on model performance Y?* **Q2**: *Why does this simple method perform well on small training data?* **Q3**: *Are those counterfactual examples that were created out of air correct or reasonable?* A similar research also makes use of entity replacement to enhance the pretrained language model for improving zero-shot fact completion task. However, it considers the replaced entity as a negative sample (Xiong et al., 2020). **Q4**: *Why can making use of a counterfactual example as a positive sample here still improve performance?*

4.1.1 Answer for Q1

Since the RI index is huge under the circumstance that there is no data augmentation by using counterfactual examples, a question arises in our mind: how will the RI index change after using counterfactual examples to train NER model? Hence, we design another experiment to compare the RI changes between the non-augmented and the augmented. As shown in Figure 4, we observe that the RI index boosts in most cases after using counterfactual examples. Even for the RI index in those experimental groups which does not increase, it is almost always a positive number. Compared to context representation, entity representation dominates the performance of the NER model in most cases, especially for models based on BERT. This phenomenon suggests that the non-spurious correlations are more located in entity representation rather than context representation.

4.1.2 Answer for Q2

The essence of our method is forcing the disentanglement of entity E and context C in the input example, and to recombine them for generating new counterfactual examples. Before, we claim that the rationality of this operation is that entity and context are not a causal relationship. Our causal results show that the NER model will pay more attention to entities in the prediction process rather than context. Agarwal et al. also find that entity representation contributes more than context representation to system performance. Hence, to a certain extent, context representation may have more spurious correlations between the input features and output labels. The recombination of entity and context can increase the diversity of training



雷尼尔森林公园有非常完备的公路和服务设施，让你看尽高深峡谷下的碧绿色溪水、漫山遍野的七彩野花。

Mt. Rainier National Park has very complete roads and service facilities, allowing you to see the turquoise waters under the deep canyons and the colorful wildflowers all over the mountains.

费沙岛有非常完备的公路和服务设施，让你看尽高深峡谷下的碧绿色溪水、漫山遍野的七彩野花。

Fraser Island has very complete roads and service facilities, allowing you to see the turquoise waters under the deep canyons and the colorful wildflowers all over the mountains.

清水寺有非常完备的公路和服务设施，让你看尽高深峡谷下的碧绿色溪水、漫山遍野的七彩野花。

Kiyomizu Temple has very complete roads and service facilities, allowing you to see the turquoise waters under the deep canyons and the colorful wildflowers all over the mountains.

布鲁克林大桥、南街海港、总督岛和埃利斯岛等著名景点，全纽约最美的风景都将被你收入囊中！

Famous sights such as Brooklyn Bridge, South Street Seaport, Governors Island and Ellis Island, the most beautiful scenery in New York will be yours!

Figure 5: Real examples of counterfactual example generation and entity recognition. The NER model augmented with reasonable counterfactual examples outperforms the same one without augmentation.

samples and eliminate the spurious correlations between variant features in context representation and output labels.

4.1.3 Answer for Q3&Q4

It would be interesting to ask a question: counterfactual examples have factual errors, and are they correct or reasonable? We have two answers from different views. On one hand, these counterfactual examples do have factual errors and are wrong. So we can regard the counterfactual example as a negative sample for improving the model performance at high-level tasks, such as semantic level or factual level (Xiong et al., 2020). On the other hand, these counterfactual examples are reasonable for named entity recognition because the task only focuses on better finding out entities and ignores the factual information. More importantly, our method preserves the linguistic correctness of the counterfactual example since the replaced entity and the candidate entity have the same entity type. These are the deep reason why the generalization ability of the NER model gets better after we treat the counterfactual example as a positive sample.

4.2 Case Study

As shown in Figure 5, we illustrate counterfactual generation and entity recognition, using the model LSTMTagger trained on the dataset CNER with training sample size $N = 200$.

When the original SCENE entity *Mt. Rainier National Park* is replaced by the other two SCENE entity, *Fraser Island* and *Kiyomizu Temple*, the discriminator judges the first counterfactual example to be reasonable, while the second is unreasonable. The second counterfactual example violates common sense because there are no canyons or colorful flowers all over the mountains in the temple. The real examples mean that our discriminator is able to filter out some extreme unreasonable counterfactual examples, and preserve some reasonable counterfactual examples that may appear in the real world.

Surprisingly, we can see the NER model augmented with counterfactual examples outperforms the NER model trained with only observational examples. Due to the help of the counterfactual example, *"Fraser Island has very complete roads and service facilities, allowing you to see the turquoise waters under the deep canyons and the colorful wildflowers all over the mountains."*, the NER model with counterfactual data augmentation can recognize all island entities with type SCENE, while the model without augmentation can not recognize these entities.

This illustration shows that our method can break the entanglement of the spurious features and the non-spurious features in the input example in the setting of limited observational examples.

4.3 Limitations

Although experimental results have shown the effectiveness of our method, our method can be further improved in terms of obtaining the most reasonable counterfactual examples. The capability of current discriminator is limited and the number of counterfactual examples regarded reasonable is large, which is not allowed especially for the large train set ($N > 500$). Although these examples increase the diversity of the combination between the entity and the context from the existing observational examples, there are lots of repeated text fragments in these examples. As far as we know, too many repeated text fragments would cause the CRF layer not to converge.

5 Related Works

We introduce the related works from two aspects:

Data Modification and Causality Recently, there is an increasing number of research works about data modification for providing the interpretability of neural models. For example, Feng et al. and Gururangan et al. reveal that neural models are overconfident in their predictions by reducing words or sentences; Ebrahimi et al. also find that adversarial examples generated by some manipulations at a character-level or a word-level can trick neural classifier. Additionally, considerable attention has been paid to utilize data modification for augmenting dataset or providing a supervised signal in the training process. For example, a rule-based data augmentation protocol has been proposed to provide a compositional inductive bias (Andreas, 2020); Kaushik et al. create new counterfactual sentences by modifying the original sentence for ameliorating the harm of spurious correlations; Xiong et al. introduce the type-constrained entity replacements to provide extra training signal for learning better factual knowledge. Interestingly, the above two points about data modification have high connections with causal inference because we can regard data modification as an intervention (Pearl et al., 2016) from a causal perspective. For instance, Ilse et al. provide a theoretical foundation from a causal perspective for explaining current data augmentation in computer vision. Kaushik et al. investigate the spurious correlations in two tasks: sentiment analysis and natural language inference.

Named Entity Recognition In this paragraph, we mainly focus on named entity recognition with limited supervision. One way to train the NER model with low-resource is dictionary-based distantly supervision (Fries et al., 2017; Shang et al., 2018; Yang et al., 2018; Liu et al., 2019) which builds a dictionary of entities for creating training data without too much effort. Few-shot learning is another promising way for training the NER model under limited supervision by transferring prior knowledge of the source domain to a new domain (Fritzler et al., 2019; Hou et al., 2019). There are also some works that focus on redefining NER as a different problem for reducing the need of hand-labeled training data. For example, Linking Rules (Safranchik et al., 2020) based on votes recognize entities through whether adjacent elements in a sequence belong to the same class; Lin et al. propose a new effective proxy of human explanation, “entity triggers”, for encouraging label-efficient learning of NER models.

6 Conclusion

In this paper, we propose a weakly-supervised method from a causal perspective and provide the interpretability of our method with the structural causal model. Our method improves generalization ability under limited observational examples. Our causal experiments suggest the spurious correlations are more located in entity representation rather than context representation. Importantly, our method eliminates part of the spurious correlations between input features and output labels.

7 Acknowledgments

We would like to thank anonymous reviewers, and Zehao Lin and Qianglong Chen for their valuable comments. This work was supported by National Key R&D Program of China (No. 2018AAA0101900), the NSFC projects (No. 61402403, No. U19B2042), Chinese Knowledge Center for Engineering Sciences and Technology, and Artificial Intelligence Research Foundation of Baidu Inc., MoE Engineering Research Center of Digital Library, and the Fundamental Research Funds for the Central Universities.

References

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2020. Interpretability analysis for named entity recognition to understand system predictions and how they can improve. *CoRR*, abs/2004.04564.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *CoRR*, abs/1907.02893.

Nancy Chinchor and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Shi Feng, Eric Wallace, Alvin Grissom, Mohit Iyyer, Pedro Rodriguez, and Jordan L. Boyd-Graber. 2018a. Pathologies of neural models make interpretation difficult. In *EMNLP*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018b. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Jason A. Fries, Sen Wu, Alexander Ratner, and Christopher Ré. 2017. Swellshark: A generative model for biomedical named entity recognition without labeled data. *CoRR*, abs/1704.06360.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, pages 993–1000.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Yutai Hou, Zhihan Zhou, Yijia Liu, Ning Wang, Wanxiang Che, Han Liu, and Ting Liu. 2019. Few-shot sequence labeling with label dependency transfer and pair-wise embedding.

Maximilian Ilse, Jakub M. Tomczak, and Patrick Forré. 2020. Designing data augmentation for simulating interventions. *CoRR*, abs/2005.01856.

Pearl Judea. 2000. Causality: models, reasoning, and inference. *Cambridge University Press. ISBN 0*, 521(77362):8.

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*.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015.

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020. Triggerner: Learning with entity triggers as explanations for named entity recognition. In *Proceedings of ACL*.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*.

Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Esteban Safranchik, Shiying Luo, and Stephen H. Bach. 2020. Weakly supervised sequence tagging from noisy rules. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5570–5578.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 175–180.

A. Torralba and A. A. Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Sergei Volodin, Nevan Wichers, and Jeremy Nixon. 2020. Resolving spurious correlations in causal models of environments via interventions. *CoRR*, abs/2002.05217.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

Liang Xu, Yu Tong, Qianqian Dong, Yixuan Liao, Cong Yu, Yin Tian, Weitang Liu, Lu Li, and Xuanwei Zhang. 2020. CLUENER2020: fine-grained named entity recognition dataset and benchmark for chinese. *CoRR*, abs/2001.04351.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169.

A Appendix

A.1 Proof of the Post-intervention Distribution

In this section, we give the proof of the post-intervention distribution $P(y|do(c_0)) = 1$.

Firstly, we review the definition of the structural causal model (SCM) (See Figure 2(a)):

$$\begin{aligned} e &:= f_E(g) \\ c &:= f_C(g) \\ x &:= f_X(e, c) \\ y &:= f_Y(x) \end{aligned} \tag{5}$$

where G is a confounding variable that influences the generation of both entity E and context C, X is the input example that is generated by E and C, and Y is the evaluation result (the $F1$ score) of the NER model. For clarity, we omit the unmeasured variables.

We use a mathematical operator $do(c_0)$ to simulates physical interventions by fixing the value of the variable c as c_0 (See Figure 2(b)). The post-intervention distribution $P(y|do(c_0))$ gives the proportion of individual that would attain response in level $Y = y$ under the hypothetical situation in which treatment $C = c_0$ is administered uniformly to the population. In order to calculate $P(y|do(c_0))$, based on Bayes' rule, we have

$$\begin{aligned} P(y|do(c_0)) &= \sum_x P(y|do(c_0), x)P(x|do(c_0)) \\ &= \sum_x P(y|do(x_0))P(x|do(c_0)) \end{aligned} \tag{6}$$

For gauging the effect of context C on the input example X, we need to calculate $P(x|do(c_0))$. However, there is a confounding variable G affects both entity E and context C. Fortunately, the variable E meets the backdoor criterion, and blocks the backdoor path $C \leftarrow G \rightarrow E \rightarrow X$. Using the adjustment formula, we have

$$P(x|do(c_0)) = \sum_e P(x|c_0, e)P(e) \tag{7}$$

In such condition, we have $P(e) = 1$ because our entity $E = e$ is unchanged and unique for each input example. Besides, we also have $P(x|do(c_0)) = 1$ though our input example $X = x$ is changed but unique. Therefore, we have $P(y|do(c_0)) = 1$ due to the certainty of the NER model for an input example. Similarly, as shown in Figure 2(c), we can also intervene on the variable E, denoted as $do(e_0)$ and have the same post-intervention distribution $P(y|do(e_0)) = 1$.