Element Intervention for Open Relation Extraction

Fangchao Liu[1,3], **Lingyong Yan**[1,3], **Hongyu Lin**[1,*], **Xianpei Han**[1,2,*], **Le Sun**[1,2]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
{fangchao2017,lingyong2014,hongyu,xianpei,sunle}@iscas.ac.cn

Abstract

Open relation extraction aims to cluster relation instances referring to the same underlying relation, which is a critical step for general relation extraction. Current OpenRE models are commonly trained on the datasets generated from distant supervision, which often results in instability and makes the model easily collapsed. In this paper, we revisit the procedure of OpenRE from a causal view. By formulating OpenRE using a structural causal model, we identify that the above-mentioned problems stem from the spurious correlations from entities and context to the relation type. To address this issue, we conduct *Element Intervention*, which intervenes on the context and entities respectively to obtain the underlying causal effects of them. We also provide two specific implementations of the interventions based on entity ranking and context contrasting. Experimental results on unsupervised relation extraction datasets show that our methods outperform previous state-of-the-art methods and are robust across different datasets[1].

1 Introduction

Relation extraction (RE) is the task to extract relation between entity pair in plain text. For example, when given the entity pair *(Obama, the United States)* in the sentence *"Obama was sworn in as the 44th president of the United States"*, an RE model should accurately predict the relationship *"President_of"* and extract the corresponding triplet *(Obama, President_of, the United States)* for downstream tasks. Despite the success of many RE models (Zeng et al., 2014; Baldini Soares et al., 2019), most previous RE paradigms rely on the predefined relation types, which are always unavailable in open domain scenario and thereby limits their capability in real applications.

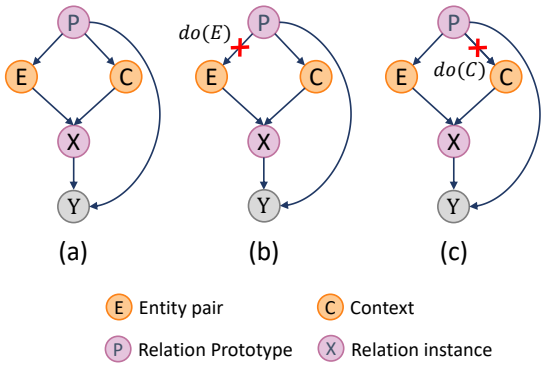

Figure 1: The Structural Causal Model demonstrates the procedure of OpenRE. (a) is the original SCM; (b) Entity intervention that fixes the entity pair and adjusts different contexts; (c) Context intervention that fixes the context and adjusts different entity pairs.

Open Relation Extraction (OpenRE), on the other hand, has been proposed to extract relation facts without pre-defined relation types neither annotated data. Given a relation instance consisting of two entities and their context, OpenRE aims to identify other instances which mention the same relation. To achieve this, OpenRE is commonly formulated as a clustering or pair-matching task. Therefore the most critical challenge for OpenRE is how to learn effective representations for relation instances and then cluster them. To this end, Yao et al. (2011) adopts topic model (Blei et al., 2003) to generate latent relation type for unlabelled instances. Later works start to utilize datasets collected using distant supervision for model training. Along this line, Marcheggiani and Titov (2016) utilizes an auto-encoder model and trains the model through self-supervised signals from entity link predictor. Hu et al. (2020) encodes each instance with pre-trained language model (Devlin et al., 2019; Baldini Soares et al., 2019) and learn the representation by self-supervised signals from pseudo labels.

Unfortunately, current OpenRE models are often unstable and easily collapsed (Simon et al., 2019).

*Corresponding authors.
[1]Code available at https://github.com/Lfc1993/EI_ORE

For example, OpenRE models frequently cluster all relation instances with context "was born in" into the relation type *BORN_IN_PLACE* because they share similar context information. However, "was born in" can also refer to the relation *BORN_IN_TIME*. Furthermore, current models also tend to cluster two relation instances with the same entities (i.e., relation instances with the same head and tail entities) or the same entity types into one relation. This problem can be even more severe if the dataset is generated using distant supervision because it severely relies on prototypical context and entity information as supervision signals and therefore lacks of diversity.

In this paper, we attempt to explain and resolve the above-mentioned problem in OpenRE from a causal view. Specifically, we formulate the process of OpenRE using a structural causal model (SCM) (Pearl, 2009), as shown in Figure 1. The main assumption behind the SCM is that distant supervision will generate highly correlated relation instances to the original prototypical instance, and there is a strong connection between the generated instance to the prototypical instance through either their entities or their context. For example, "[Jobs] was born in [California]" and "[Jobs] was born in [1955]" are highly correlated because they share similar context "was born in" and entity "Jobs". Such connection will result in spurious correlations, which appear in the form of the backdoor paths in the SCM. Then the spurious correlations will mislead OpenRE models, which are trained to capture the connection between entities and context to the relation type.

Based on the above observations, we propose *element intervention*, which conducts backdoor adjustment on entities and context respectively to block the backdoor paths. However, due to the lack of supervision signals, we cannot directly optimize towards the underlying causal effects. To this end, we further propose two surrogate implementations on the adjustments on context and entities, respectively. Specifically, we regard the instances in the original datasets as the relation prototypes. Then we implement the adjustment on context through a Hierarchy-Based Entity Ranking (Hyber), which fixes the context, samples related entities from an entity hierarchy tree and learns the causal relation through rank-based learning. Besides, we implement the adjustment on entities through a Generation-based Context Contrasting (Gcc), which fixes the entities, generates positive and negative contexts from a generation-based model and learns the causal effects through contrastive learning.

We conduct experiments on different unsupervised relation extraction datasets. Experimental results show that our method outperforms previous state-of-the-art methods with a large margin and suffers much less performance discrepancy between different datasets, which demonstrate the effectiveness and robustness of the proposed methods.

2 OpenRE from Causal View

In this section, we formulate OpenRE from the perspective of Structural Causal Model and give the theoretical proof for intervention methods that block the backdoor paths from relation elements (i.e., context and entity pair) to the latent relation types.

2.1 Task Definition

Relation extraction (RE) is the task of extracting the relationship between two given entities in the context. Considering the sequence example: $\mathcal{S} = [s_0, ..., s_{n-1}]$ which contains n words, $e_1 = [i, j]$ and $e_2 = [k, l]$ indicate the entity pair, where $0 \leq i \leq j < k \leq l \leq n-1$, a relation instance X is defined as $X = (\mathcal{S}, e_1, e_2)$, (i.e. the tuple of entity pair and the corresponding context). The element of a relation instance is the entity pair and the corresponding context. Traditional RE task is to predict the relations type when given X. However, the target relation types are not pre-defined in OpenRE. Consequently, OpenRE is commonly formulated as a clustering task or a pair-matching task by considering whether two relation instances X_i and X_j refer to the same relation.

Unfortunately, current OpenRE models are often unstable and easily collapsed (Simon et al., 2019). In the next section, we formulate OpenRE using a structural causal model and then identify the reasons behind these deficiencies from the SCM.

2.2 Structural Causal Model for OpenRE

Figure 1 (a) shows the structural causal model for OpenRE. The main idea behind the SCM is distant supervision will generate highly correlated relation instances to the original prototypical instance, and there is a strong connection between the generated instance to the prototypical instance through

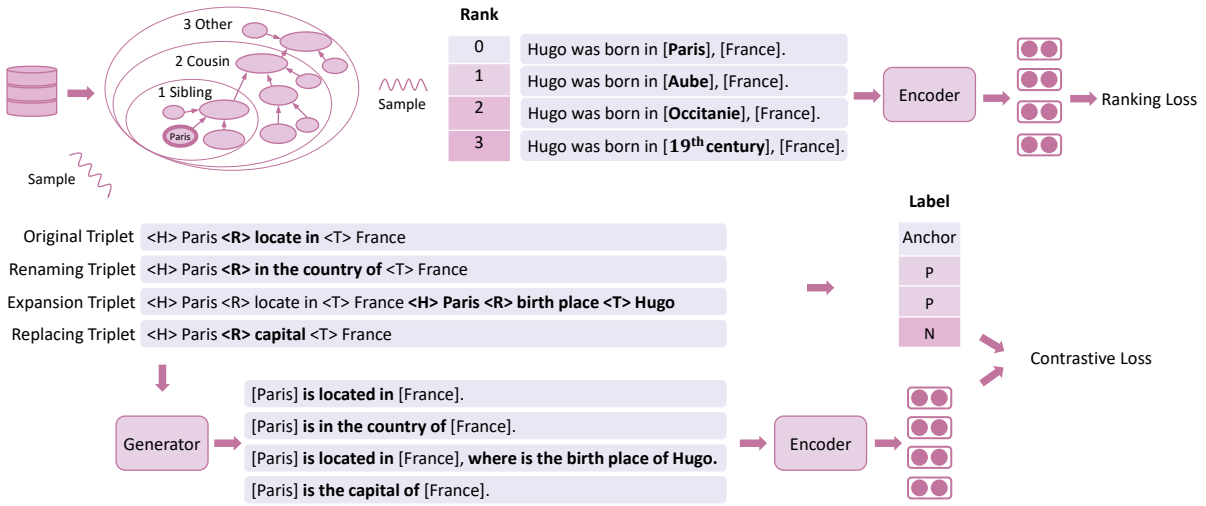

Figure 2: Framework of Element Intervention.

either their entities or their context. Specifically, in the SCM, we describe OpenRE with five critical variables: 1) the prototypical relation instance P, which is a representative relation instance of one relation type cluster; 2) the entity pair E, which encodes the entity information of one relation instance; 3) the context C, which encodes the context information of one relation instance; 4) a relation instance X (which can be generated from distant supervision or other strategies) and 5) the final pair-wise matching result Y, which corresponds to whether instance X and the prototypical relation instance P entail the same relation.

Given the variables mentioned above, we formulate the process of generating OpenRE instances based on the following causal relations:

- $E \leftarrow P \rightarrow C$ formulates the process of sampling related entities and context respectively from the prototypical relation instance P.

- $E \rightarrow X \leftarrow C$ formulates the relation instance generating process. Given the context C and entities E from the prototypical relation instance P, a new relation instance X is generated based on the information in C and E. This process can be conducted through distant supervision.

- $P \rightarrow Y \leftarrow X$ formulates the OpenRE clustering or pair-wise matching process. Given a prototypical relation instance P and another relation instance X, this process will determine whether X belongs to the relation cluster of P.

2.3 Spurious Correlations in OpenRE

Given a relation prototypical instance P, the learning process of OpenRE is commonly to maximize the probability $\mathcal{P}(y, P|X) = \mathcal{P}(y, P|E, C)$. However, as it can be observed from the SCM, there exists a backdoor path $P \rightarrow E \rightarrow X$ when we learn the underlying effects of context C. That is to say, the learned effect of C to Y is confounded by E (through P). For example, when we learned the effects of context "was born in" to the relation "BORN_IN_PLACE", the backdoor path will lead the model to mistake the contribution of the entities (PERSON, PLACE) to the contribution of context, and therefore resulted in spurious correlation. The same thing happens when we learn the effects of entities E, which is influenced by the backdoor path $P \rightarrow C \rightarrow X$. As a result, optimizing these spurious correlations will result in an unstable and collapsed OpenRE model.

2.4 Resolving Spurious Correlations via Element Intervention

To resolve the spurious correlations, we adopt the backdoor adjustment (Pearl, 2009) to block the backdoor paths. Specifically, we separately intervene on context C and entities E by applying the *do*–operation.

Entity Intervention. As shown in Figure 1 (b), to avoid the spurious correlations of entities to relation types, we conduct the *do*-operation by inter-

vening on the entities E:

$$\begin{aligned}\mathcal{P}(Y,P|do(E=e_0))&=\sum_{C,X}\mathcal{P}(C,P)\mathcal{P}(X,Y|e_0,C,P)\\&=\sum_{C}\mathcal{P}(C,P)\mathcal{P}(Y|e_0,C,P)\\&=\sum_{C}\mathcal{P}(P)\mathcal{P}(C|P)\mathcal{P}(Y|e_0,C,P)\end{aligned}\tag{1}$$

Since $\mathcal{P}(P)$ is uniformly distributed in the real world, this equation can be rewritten as:

$$\begin{aligned}&\mathcal{P}(Y,P|do(E=e_0))\\&\propto\sum_{C}\mathcal{P}(C|P)\mathcal{P}(Y|e_0,C,P)\end{aligned}\tag{2}$$

This equation means the causal effect from the entities E to its matching result Y can be estimated by considering the corresponding possibility of each context given the prototypical relation instance P. The detailed implementation will be described in the next section.

Context Intervention. Similarly, we conduct context intervention to avoid the spurious correlations of context to relation types, as shown in Figure 1 (c):

$$\begin{aligned}&\mathcal{P}(Y,P|do(C=c_0))\\&\propto\sum_{E}\mathcal{P}(E|P)\mathcal{P}(Y|c_0,E,P)\end{aligned}\tag{3}$$

which means the causal effect from the context C to its matching result Y can be estimated by considering the corresponding possibility of each entity E given P. The detailed implementation will also be described in the next section.

2.5 Optimizing Causal Effects for OpenRE

To effectively capture the causal effects of entities E and context C to OpenRE, a matching model $\mathcal{P}(Y|C,E,P;\theta)$ should be learned by optimizing the causal effects:

$$\begin{aligned}L(\theta)=&I(X,P)\cdot\mathcal{P}(Y=1,P|do(E=e(X))\\&+I(X,P)\cdot\mathcal{P}(Y=1,P|do(C=c(X))\\&+[1-I(X,P)]\cdot\mathcal{P}(Y=0,P|do(E=e(X))\\&+[1-I(X,P)]\cdot\mathcal{P}(Y=0,P|do(C=c(X))\end{aligned}\tag{4}$$

where $e(X)$ and $c(X)$ represents the entities and context in relation instance X, $I(X,P)$ is an indicator which represents whether X and P belong to the same relation. $\mathcal{P}(Y|C,E,P;\theta)=\mathcal{P}(Y|X,P;\theta)$ is a matching model, which is defined using a prototype-based measurements:

$$\mathcal{P}(Y|X,P;\theta)\propto -D(R(X;\theta),R(P;\theta))\tag{5}$$

where D is a distance measurement and $R(X;\theta)$ is a representation learning model parametrized by θ, which needs to be optimized during learning. In the following, we will use $D(X,P)=D(R(X;\theta),R(P;\theta))$ for short.

However, it is difficult to directly optimize the above loss function because 1) in unsupervised OpenRE, we are unable to know whether the relation instance X generated from (E,C) matches the prototypical relation instance P; 2) we are unable to traverse all possible E and C in Equation (2) and (3). To resolve these problems, in the next section, we will describe how we implement the context intervention via hierarchy-based entity ranking and the entity intervention via generation-based context contrasting.

3 Element Intervention Implementation

As we mentioned above, it is difficult to directly optimize the causal effects via Equation (4). To tackle this issue, this section provides a detailed implementation to approximate the causal effects. Specifically, we regard all relation instances in the original data as the prototypical relation instance P, and then generate highly correlated relation instances X from P via a hierarchy-based sampling and generation-based contrasting. Then we regard structural signals from the entity hierarchy and confidence score from the generator as distant supervision signals, and learn the causal effects via ranking-based learning and contrastive learning.

3.1 Hierarchy-based Entity Ranking for Context Intervention

To implement context intervention, we propose to formulate $\mathcal{P}(E|P)$ using an entity hierarchy, and approximately learn to optimize the causal effects of $\mathcal{P}(Y=1,P|do(C))$ and $\mathcal{P}(Y=0,P|do(C))$ in Equation (4) via a hierarchy-based entity ranking loss. Specifically, we first regard all relation instances in the data as prototypical relation instance P. Then we formulate the distribution $\mathcal{P}(E|P)$ by fixing the context in P and replacing entities by sampling from an entity hierarchy. Each sampled entity is regarded as the same $\mathcal{P}(E|P)$. Intuitively, the entity closer to the original entities in P tends to generate more consistent relation instance to P. To approximate this semantic similarity, we utilize the meta-information in WikiData (i.e., the *"instance_of"* and *"subclass_of"* statements, which

describe the basic property and concept of each entity), and construct a hierarchical entity tree for ranking the similarity between entities. In this work, we apply a three-level hierarchy through these two statements:

- **Sibling Entities**: The entities belonging to the same parent category as the original entity. For example, *"Aube"* and *"Paris"* are sibling entities since they are both the child entity of *"department of France"*, and both express the concepts of location and GPE. These sibling entities can be considered as golden entities to replace.
- **Cousin Entities**: The entities belonging to the same grandparent category but the different parent category from the original entity. For example, *"Occitanie"* and *"Paris"* is of the same grandparent category *"French Administrative Division"*, but shares different parent category. These entities can be considered as silver entities since they are likely to be the same type as the original one but less possible than the sibling entities.
- **Other Entities**: The entities beyond the grandparent category, which are much less likely to be the same type as the original one.

For the example in Figure 2, the prototypical relation instance *"Hugo was born in [Paris], [France]"* is sampled to be intervened. We first fix the context and randomly choose one of the head or tail entity to be replaced. In this case, we choose *"Paris"*. Then, entities that correspond to different hierarchies are sampled and to replace the original entity. In this case, *"Aube"* is sampled as the sibling entity, *"Occitanie"* to be the cousin entity and *"19^{th} century"* to be the other entity.

After sampled these intervened instances, we approximately optimize $\mathcal{P}(Y, P|do(C))$ using a rank-based loss function:

$$\mathcal{L}_E(\theta; \mathcal{X}) = \sum_{i=1}^{n-1} \max(0, D(P, X_i) - D(P, X_{i+1}) + m_E), \quad (6)$$

where θ is the model parameters, $D(X_i, P)$ is the distances between representations of generated relation instance X_i and prototypical relation instance P. X is the intervened relation instance set, m_E is the margin for entity ranking loss, and $n = 3$ is the depth of the entity hierarchy.

3.2 Generation-based Context Contrasting for Entity Intervention

Different from the context intervention that can easily replace entities, it is more difficult to intervene on entities and modify the context. Fortunately, the rapid progress in pre-trained language model (Radford et al., 2019; Lewis et al., 2020; Raffel et al., 2020) makes the language generation from RDF data[2] available (Ribeiro et al., 2020). So in this work, we take a different paradigm named Generation-based Context Contrasting, which directly generates different relation instances from specifically designed relation triplets, and approximately learn to optimize the causal effects of $\mathcal{P}(Y = 1, P|do(E))$ and $\mathcal{P}(Y = 0, P|do(E))$ in Equation (4) via contrastive learning. Specifically, we first sample relation triplets from Wikidata as prototypical relation instance P, and then generates relation triplets with the same entities but different relation context using the following strategies:

- **Relation Renaming**, which contains the same entity pair with the original one, but an alias relation name for generating a sentence with different expressions. Then this instance is considered as a positive sample to prototypical relation instance.
- **Context Expansion**, which extends the original relation instance with an additional triplet. The added triplet owns the same head/tail entity with the original instance but differs in the relation and tail/head entity. This variety aims to add irrelative context, which forces the model to focus on the important part of the context and is also considered as a positive sample to prototypical relation instance.
- **Relation Replacing**, which contains the same entity pair as the original one, but with other relations between these two entities. This variety aims to avoid spurious correlations that extracts only based on the entity pair and is considered as a negative instance to the prototypical relation instance.

Then we use the generator to generate texts based on these triplets. Specifically, we first wrap the triplets with special markers *"[H], [T],[R]"* corresponds to head entity, tail entity, and relation name. Then we input the concatenated texts for relation instance generation. In our implementation, we

[2] https://www.w3.org/TR/WD-rdf-syntax-971002/

use T5 (Raffel et al., 2020; Ribeiro et al., 2020) as the base generator, and pre-train the generator on WebNLG data (Gardent et al., 2017). After sampled these intervened instances, we approximately optimize $\mathcal{P}(Y, P|do(E))$ using the following contrastive loss function:

$$\mathcal{L}_C(\theta; \mathcal{X}) = \sum_{X_p \in \mathcal{P}} \sum_{X_n \in \mathcal{N}} \max(D(P, X_p) - D(P, X_n) + m_C, 0), \quad (7)$$

where θ is the model parameters, $\mathcal{X}$ is the intervened instance set, $\mathcal{P}$ is the positive instance set generated from relation renaming and context expansion, $\mathcal{N}$ is the negative instance set generated from relation replacing, P is the original prototypical relation instance, m_C is the margin.

3.3 Surrogate Loss for Optimizing Causal Effects

Based on entity ranking and context contrasting, we approximate the causal effects optimized in Equation (4) with the following ranking and contrastive loss:

$$\mathcal{L}(\theta; \mathcal{X}) = \mathcal{L}_E(\theta; \mathcal{X}) + \mathcal{L}_C(\theta; \mathcal{X}). \quad (8)$$

which involves both the entity ranking loss and the context contrastive loss. During inference, we first encode each instance into its representation using the learned model. Then we apply a clustering algorithm to cluster the relation representations, and the relation for each instance is predicted through the clustering results.

4 Experiments

4.1 Dataset

We conduct experiments on two OpenRE datasets – T-REx SPO and T-REx DS, since these datasets are from the same data source but only differ in constructing settings, which is very suitable for evaluating the stability of OpenRE methods. These datasets are both from T-REx[3] (Elsahar et al., 2018) – a dataset consists of Wikipedia sentences that are distantly aligned with Wikidata relation triplets; and these aligned sentences are further collected as T-REx SPO and T-REx DS according to whether they have surface-form relations or not. As a result, T-REx SPO contains 763,000 sentences of 615 relations, and T-REx DS contains nearly 12 million sentences of 1189 relations. For both datasets, we use 20% for validation and the remaining for model training as Hu et al. (2020).

[3] https://hadyelsahar.github.io/t-rex/

4.2 Baseline and Evaluation Metrics

Baseline Methods. We compare our model with the following baselines: 1) **rel-LDA** (Yao et al., 2011), a generative model that considers the unsupervised relation extraction as a topic model. We choose the full rel-LDA with a total number of 8 features for comparison in our experiment. 2) **March** (Marcheggiani and Titov, 2016), a VAE-based model learned by self-supervised signal of entity link predictor. 3) **UIE** (Simon et al., 2019), a discriminative model that adopts additional regularization to guide model learning. And it has different versions according to the choices of different relation encoding models (e.g., PCNN). We report the results of two versions–UIE-PCNN and UIE-BERT (i.e., using PCNN and BERT as the relation encoding models) with the highest performance. 4) **SelfORE** (Hu et al., 2020), a self-supervised framework that bootstraps to learn a contextual relation representation through adaptive clustering and pseudo label.

Evaluation Metrics. We adopt three commonly-used metrics to evaluate different methods: B^3 (Bagga and Baldwin, 1998), V-measure (Rosenberg and Hirschberg, 2007) and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985).

Specifically, B^3 contains the precision and recall metrics to correspondingly measure the correct rate of putting each sentence in its cluster or clustering all samples into a single class, which are defined as follows:

$$B^3_{\text{Prec.}} = \mathop{\mathbb{E}}_{X,Y} P(g(X) = g(Y)|c(X) = c(Y))$$

$$B^3_{\text{Rec.}} = \mathop{\mathbb{E}}_{X,Y} P(c(X) = c(Y)|g(X) = g(Y))$$

Then B^3 F_1 is computed as the harmonic mean of the precision and recall.

Similar to B^3, V-measure focuses more on small impurities in a relatively "pure" cluster than less "pure" cluster, and use the homogeneity and completeness metrics:

$$V_{\text{Homo.}} = 1 - H(c(X)|g(X))/H(c(X))$$

$$V_{\text{Comp.}} = 1 - H(g(X)|c(X))/H(g(x))$$

ARI is a normalization of the Rand Index, which measures the agreement degree between the cluster and golden distribution. This metric ranges in [-1,1], a more accurate cluster will get a higher score. Different from previous metrics, ARI is

Dataset	model	B^3			V-measure			ARI
		F1	Prec.	Rec.	F1	Homo.	Comp.	
T-REx SPO	rel-LDA-full (Yao et al., 2011)*	18.5	14.3	26.1	19.4	16.1	24.5	8.6
	March (Marcheggiani and Titov, 2016)*	24.8	20.6	31.3	23.6	19.1	30.6	12.6
	UIE-PCNN (Simon et al., 2019)	36.3	28.4	50.3	41.4	33.7	53.6	21.3
	UIE-BERT (Simon et al., 2019)	38.1	30.7	50.3	39.1	37.6	40.8	23.5
	SelfORE (Hu et al., 2020)	41.0	39.4	42.8	41.4	40.3	42.5	33.7
	Our	**45.0**	46.7	43.4	**45.3**	45.4	45.2	**36.6**
	w/o Hyber	41.4	40.9	42.0	43.7	42.3	45.2	33.2
	w/o Gcc	42.2	44.2	40.4	45.2	44.7	45.7	34.7
T-REx DS	rel-LDA-full (Yao et al., 2011)*	12.7	8.3	26.6	17.0	13.3	23.5	3.4
	March (Marcheggiani and Titov, 2016)*	9.0	6.4	15.5	5.7	4.5	7.9	1.9
	UIE-PCNN (Simon et al., 2019)	19.7	14.0	33.4	26.6	20.8	36.8	9.4
	UIE-BERT (Simon et al., 2019)	22.4	17.6	30.8	31.2	26.3	38.3	12.3
	SelfORE (Hu et al., 2020)	32.9	29.7	36.8	32.4	30.1	35.1	20.1
	Our	**42.9**	40.2	45.9	**47.3**	46.9	47.8	**25.0**
	w/o Hyber	40.9	39.2	42.7	43.0	42.5	43.6	22.4
	w/o Gcc	41.5	40.1	42.9	45.2	44.8	45.6	21.7

Table 1: Results (%) on unsupervised relation extraction datasets. The results of * are reproduced in Simon et al. (2019), Hyber refers to our Hierarchy-based Entity Ranking methods and Gcc refers to Generation-based Context Contrasting method.

less sensitive to precision/homogeneity and recall/completeness.

4.3 Hyperparameters and Implementation Details

In the training period, we manually search the Hyperparameters of learning rate in [5e-6,1e-5, 5e-5], and find 1e-5 is optimal, search weight decay in [1e-6, 3e-6, 5e-5] and choose 3e-6, and use other hyperparameters without search: the dropout rate of 0.6, a batch size of 32, and a linear learning schedule with a 0.85 decay rate per 1000 mini-batches. In the evaluation period, we simply adopt the pre-trained models for representation extraction, then cluster the evaluate instances based on these representations. For clustering, we follow previous work (Simon et al., 2019; Hu et al., 2020) and set K=10 as the number of clusters. The training period of each epoch costs about one day. In our implementation, we adopt Bert-base-uncased model [4] as the base model for relation extraction and a modified T5-base model [5] for text generation. The entity hierarchical tree is constructed based on WikiData and finally contains 589,121 entities. The generation set contains about 530,000 triplets, and each triplet corresponds to 5 positive/negative triplets and generated texts. We use one Titan RTX for Element Intervention training and four cards of RTX for text generation.

[4] https://github.com/huggingface/transformers
[5] https://github.com/UKPLab/plms-graph2text

Source	B^3	V-meas.	ARI
T-REx SPO	45.0	45.3	36.6
Generated	46.0	44.6	36.7

Table 2: The results (%) of entity ranking based on different data sources. These results are reported on T-REx SPO. And we only report the F_1 scores of B^3 and V-measure for simplicity.

4.4 Overall Results

Table 1 shows the overall results on T-REx SPO and T-REx DS. From this table, we can see that:

1. **Our method outperforms previous OpenRE models and achieves the new state-of-the-art performance.** Comparing with all baseline models, our method achieves significant performance improvements: on T-Rex SPO, our method improves the SOTA B^3 F_1 and V-measure F_1 by at least 3.9%, and ARI by 2.9%; on T-Rex DS, the improvements are more evident, where SOTA B^3 F_1 and V-measure F_1 are improved by at least 10.0%, and ARI is improved by 4.9%.

2. **Our methods perform robustly in different datasets.** Comparing the performances on these two datasets, we can see that almost all baseline methods suffer dramatic performance drops on all these metrics, which verifies that previous OpenRE methods can be easily influenced by the spurious correlations in datasets, as T-REx DS involves much more noisy instances without relation surface forms. As

Metrics	Both	Seen	Unseen
BLEU	60.9	65.9	54.9
chrF++	76.0	79.2	72.5

Table 3: Quantitative performance of our generator on WebNLG. Seen stands for generating from seen relation triplets, unseen stands for generating from unseen relation triplets. Both stands for a combination of seen and unseen relation triplets.

contrast, our methods have marginal performance differences, which indicates both the effectiveness and robustness of our methods.

4.5 Detailed Analysis

In this section, we conduct several experiments for detailed analysis of our method.

Ablation Study. To study the effect of different intervention modules, we conduct an ablation study on each intervention module by correspondingly ablating one. The other setting remains the same as the main model. From Table 1, we can see that, in both T-REx SPO and DS, combining these two modules can result in a noticeable performance gain, which demonstrates that both two modules are important to the final model performance and they are complementary on alleviating unnecessary co-dependencies: Hyber aims to alleviate the spurious correlations between the context and the final relation prediction, and Gcc aims to alleviate the spurious correlations between entity pair and the final relation prediction. Besides, in T-REx DS, we can see that Hyber or Gcc only is effective enough to outperform previous SOTA methods, which indicates that element intervention has clearly unbiased representation on either entity pair or context.

Entity Ranking on Generated Texts. This experiment studies the effect of different data sources for Hyber module. As shown in Table 2, we can see that Hyber based on T-REx SPO dataset or the generated texts has marginal difference. That means Hyber is robust to the source context. On the other hand, the quality of the generated texts satisfies the demand of this task.

Quality of Context Generation(unseen relations). This experiment gives a quantitative analysis of the generator used in our work. We select WebNLG (Gardent et al., 2017) to test the generator, and adopt the widely-used metrics including BLEU (Papineni et al., 2002) and chrF++ (Popović, 2017) for evaluation. As shown in Table 3, we can see that our generator is quite effective on seen relation generation. Though the generator suffers a performance drop in unseen relations, the scores are still receptible. Combined with results from other experiments, the generator is sufficient for this task.

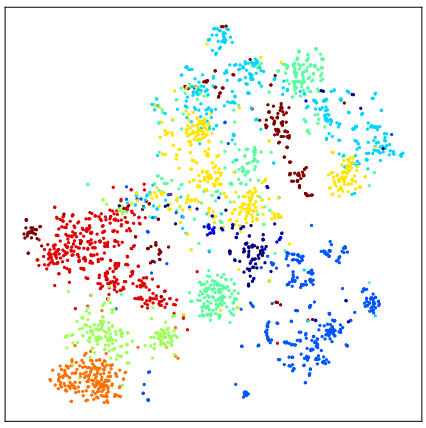

Figure 3: Visualization of relation representation learned by element intervention. Each relation instance is colored with the ground-truth label.

Visualization of Relation Representations. In this experiment, we visual the representations of the validation instances. We sample 10 relations from the T-REx SPO validation set and each relation with 200 instances for visualization. To reduce the dimension, we use t-sne (van der Maaten and Hinton, 2008) to map each representation to the dimension of 2. For the convenience of comparison, we color each instance with its ground-truth relation label. Since the visualization results of only Hyber or Gcc are marginally different from the full model, so we only choose the full model for visualization. As shown in Figure 3, we can see that each relation is mostly separate from others. However, there still be some instances misclassified due to the overlapping in the representation space.

5 Related Work

Current success of supervised relation extraction methods (Bunescu and Mooney, 2005; Qian et al., 2008; Zeng et al., 2014; Zhou et al., 2016; Velikovi et al., 2018) depends heavily on large amount of annotated data. Due to this data bottleneck, some weakly-supervised methods are proposed to learn relation extraction models from distantly labeled datasets (Mintz et al., 2009; Hoffmann et al., 2011; Lin et al., 2016) or few-shot datasets (Han et al., 2018; Baldini Soares et al., 2019; Peng et al., 2020). However, these paradigms still require pre-defined

relation types and therefore restricts their application to open scenarios.

Open relation extraction, on the other hand, aims to cluster relation instances referring to the same underlying relation without pre-defined relation types. Previous methods for OpenRE can be roughly divided into two categories. The generative method (Yao et al., 2011) formulates OpenRE using a topic model, and the latent relations are generated based on the hand-crafted feature representations of entities and context. While the discriminative method is first proposed by Marcheggiani and Titov (2016), which learns the model through the self-supervised signal from entity link predictor. Along this line, Hu et al. (2020) propose the SelfORE that learns the model through pseudo label and bootstrapping technology. However, Simon et al. (2019) point out that previous OpenRE methods severely suffer from the instability, and they also propose two regularizers to guide the learning procedure. But the fundamental cause of the instability is still undiscovered.

In this paper, we revisit the procedure of OpenRE from a causal view. By formulating OpenRE using a structural causal model, we identify the cause of the above-mentioned problems, and alleviate the problems by Element Intervention. There are also some recent studies try to introduce causal theory to explain the spurious correlations in neural models (Feng et al., 2018; Gururangan et al., 2018; Tang et al., 2020; Qi et al., 2020; Zeng et al., 2020; Wu et al., 2020; Qin et al., 2020; Fu et al., 2020). However, to the best of our knowledge, this is the first work to revisit OpenRE from the perspective of causality.

6 Conclusions

In this paper, we revisit OpenRE from the perspective of causal theory. We find that the strong connections between the generated instance to the prototypical instance through either their entities or their context will result in spurious correlations, which appear in the form of the backdoor paths in the SCM. Then the spurious correlations will mislead OpenRE models. Based on the observations, we propose *Element Intervention* to block the backdoor paths, which intervenes on the context and entities respectively to obtain the underlying causal effects of them. We also provide two specific implementations of the interventions based on entity ranking and context contrasting. Experimental results on two OpenRE datasets show that our methods outperform previous methods with a large margin, and suffer the least performance discrepancy between datasets, which indicates both the effectiveness and stability of our methods.

Acknowledgements

We thank all reviewers for their insightful suggestions. Moreover, this work is supported by the National Key Research and Development Program of China under Grant No.2019YFC1521200, the National Natural Science Foundation of China under Grants no. U1936207 and 61772505, and in part by the Youth Innovation Promotion Association CAS(2018141).

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge

base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Tsu-Jui Fu, Xin Wang, Scott Grafton, Miguel Eckstein, and William Yang Wang. 2020. SSCR: Iterative language-based image editing via self-supervised counterfactual reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4413–4422, Online. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. SelfORE: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682, Online. Association for Computational Linguistics.

L. Hubert and P. Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

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, Online. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. 2020. Two causal principles for improving visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 697–704, Manchester, UK. Coling 2008 Organizing Committee.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, Online. Association for Computational Linguistics.

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

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

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *CoRR*, abs/2007.08426.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1378–1387, Florence, Italy. Association for Computational Linguistics.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*.

Petar Velikovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Li, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. 2020. De-biased court's view generation with causality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 763–780, Online. Association for Computational Linguistics.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and 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, Online. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.