De-Bias for Generative Extraction in Unified NER Task

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, Weiming Lu*
College of Computer Science and Technology, Zhejiang University
{zsss, luwm}@zju.edu.cn

Abstract

Named entity recognition (NER) is a fundamental task to recognize specific types of entities from a given sentence. Depending on how the entities appear in the sentence, it can be divided into three subtasks, namely, Flat NER, Nested NER, and Discontinuous NER. Among the existing approaches, only the generative model can be uniformly adapted to these three subtasks. However, when the generative model is applied to NER, its optimization objective is not consistent with the task, which makes the model vulnerable to the incorrect biases. In this paper, we analyze the incorrect biases in the generation process from a causality perspective and attribute them to two confounders: pre-context confounder and entity-order confounder. Furthermore, we design Intra- and Inter-entity Deconfounding Data Augmentation methods to eliminate the above confounders according to the theory of backdoor adjustment. Experiments show that our method can improve the performance of the generative NER model in various datasets.

1 Introduction

Named entity recognition (NER) is a task aimed at identifying distinct and independent entities from a given text while classifying them into predefined types. As a fundamental work in Natural Language Processing (NLP), its research facilitates the application of many downstream tasks (Ganea and Hofmann, 2017; Miwa and Bansal, 2016; Shen et al., 2021b). In previous work (Sang and Meulder, 2003; Pradhan et al., 2013a; Doddington et al., 2004; Kim et al., 2003; Karimi et al., 2015), three kinds of different NER subtasks were raised (as shown in Figure 1), which are Flat NER, Nested NER and Discontinuos NER.

The existing NER methods can be divided into three main categories, including labeling-based (Ju et al., 2018; Straková et al., 2019), span-based (Luan et al., 2019a; Shen et al., 2021a) and generative-based (Straková et al., 2019; Paolini et al., 2021; Yan et al., 2021a) methods. Non-generative methods have different problems when applied to all three different subtasks: the labeling-based methods need to design different tagging schema for various types (Ratinov and Roth, 2009; Metke-Jimenez and Karimi, 2016; Straková et al., 2019; Dai et al., 2020) while the span-based methods suffers from ambiguity of boundary when applied to discontinuous task. Although generative-based methods are able to model all NER subtasks uniformly (Yan et al., 2021a), the training objective differ significantly from NER task due to the autoregressive generation mannner, resulting in some incorrect biases learned by the model during the training process.

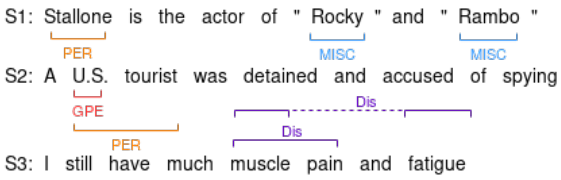

Figure 1: Examples involving flat, nested and discontinuous NER types, Entities are highlighted with colored markers

.

From a causal perspective, the incorrect biases stem from two confounders: pre-context confounder and entity-order confounder. Pre-context confounder means that the model is affected by pre-context words that may be extra-entity words when generating a particular entity word. For example, in S3 of Figure 1, the autoregressive generation mannner causes the model to generate the word *"fatigue"* of the entity *"muscle fatigue"* conditioned on the extra-entity words *"muscle"* and *"pain"*. This causes the model to mistakenly establish dependen-

* Corresponding author

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 808 - 818

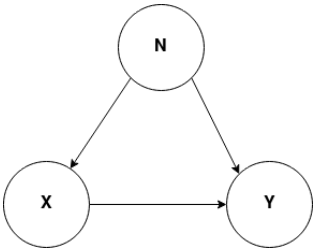

Figure 2: Structural Causal Model of the generative NER method. The confounder N causes the spurious correlation $X \leftarrow N \rightarrow Y$ to mislead the model from the true objective $X \rightarrow Y$.

cies between the intra-entity word *"fatigue"* and the extra-entity words *"muscle"* and *"pain"*, while ignoring the dependency between the intra-entity words *"muscle"* and *"fatigue"*. Therefore, when only the entity *"muscle fatigue"* is in the input sentence, the model cannot predict the entity accurately and completely due to the learned incorrect dependency bias. Entity-order confounder refers to the fact that the model is affected by a predetermined order of entities when generating an entity sequence. The entities in a sentence are essentially a set structure without decoding order among them. In contrast, the generative NER model pre-specifies the decoding order of entities, which introduces incorrect bias and ignores the bidirectional dependency between entities. As in S1 of Figure 1, after fixing the set of entities as *"Stallone"* $\rightarrow$ *"Rocky"* $\rightarrow$ *"Rambo"*, the model only models the unidirectional dependency of *"Rambo"* on *"Stallone"* and *"Rocky"*, without considering the reverse dependency of *"Stallone"* on *"Rocky"* and *"Rambo"*. In this case, if *"Rambo"* is decoded first, it is difficult for the model to decode the other two entities *"Stallone"* and *"Rocky"* due to the lack of the reverse dependency.

We can formulate the causalities in the process of entity sequence generation with a Structural Causal Model (SCM). As illustrated in Figure 2, the direct links denote the causality between the two nodes: cause $\rightarrow$ effect. $X \rightarrow Y$ represents the generation process of the target sequence, which can be divided into two cases according to the location of the generated words: intra-entity generation and inter-entity generation. In the former case, N denotes the pre-context words, which can affect the generation of the next word ($N \rightarrow Y$). While in the latter case, N denotes the entity decoding order, and can affect the generation of the next entity ($N \rightarrow Y$). In both cases, the representation of input X is contaminated by the backdoor path $X \leftarrow N \rightarrow Y$. Therefore, N is a confounder for the $X \rightarrow Y$ process which introduces a incorrect bias to the model.

In order to eliminate the bias caused by confounders N in both cases, we designed the Intra- and Inter-entity Deconfounding Data Augmentation method from the theory of backdoor adjustment. Our contributions are as follows:

- We analyzed the incorrect bias of the generative model on the NER task from a causal perspective, concluding that the pre-context confounder and the entity-order confounder are the main causes of the bias.
- Based on the backdoor adjustment theory, we designed the Intra- and Inter-entity Deconfounding Data Augmentation methods to remove the pre-context confounder and the entity-order confounder, respectively, to eliminate the incorrect bias of the generative model on the NER task.
- Experiments on three kinds of NER tasks show that our proposed method can de-bias the generative NER model and thus improve the model performance.

2 Prerequisite

For subsequent analysis, in this section we first illustrate how the NER task is modeled as a generative task, after which we illustrate the training and inference process of the generative model.

2.1 Problem Definition

The three kinds of NER tasks can all be formulated as follows, given an input sentence of l tokens $x = \{x_1, x_2, ..., x_l\}$, the target sequence $y = \{[ss], E_1, \cdots, E_M, [ee]\}$, where $E_i = \{[s], y_{e_1^i}, ..., y_{e_\mathcal{E}^i}, [e]\}$ is a word sequence of entity e^i, M denotes the number of the entities, $\mathcal{E}$ denotes the length of the entity, $[ss]$ and $[ee]$ are the start and end tags for the sequence, $[s]$ and $[e]$ are the start and end tags for the entity, and $y_{e_j^i}$ is the j-th word of i-th target entity.

2.2 Generative Model

In general, given an input sentence x, the generative model will return a sequence consisting of a collection of entities arranged in fixed order $y = \{[ss], E_1, \cdots, E_M, [ee]\}$. To this end,

we first computes the hidden vector representation $H = h_1, ..., h_l$ of the input via a multi-layer transformer encoder:

$$H = Encoder(x_1, ..., x_l) \quad (1)$$

where each layer of Encoder$(\cdot)$ is a transformer block with multi-head attention mechanism.

After the input sentence is encoded, the decoder predicts the output token-by-token according to the sequential inputs' hidden vectors. At the step i of generation, the self-attention decoder predicts the $i\text{-}th$ token y_i in the linearized form and decoder state h_i^d as:

$$y_i, h_i^d = Decoder([H; h_1^d, ..., h_{i-1}^d], y_{i-1}) \quad (2)$$

where each layer of Decoder$(\cdot)$ is a transformer block that contains self-attention with decoder hidden state $h_{<i}^d$ and cross-attention with encoder state H.

Specifically, the optimization objective of the generated model is to maximize the conditional probability of the entire output sequence $p(y|x)$, which is progressively combined by the probability of each step $p(y_i|y_{<i}, x)$:

$$p(y|x) = \prod_{i}^{|y|} p(y_i|y_{<i}, x) \quad (3)$$

3 The Proposed Solution

In the above, we have analyzed that the bias in the traditional generative NER model $P(Y|X)$ is introduced by two kinds of confounders: the pre-context confounder and the entity-order confounder. Now we need to perform the deconfounding using backdoor adjustment to obtain a debiased model $P(Y|do(X))$. Deconfounding seeks the true causal effect of one variable on another, and it is appealing to the objective of NER: given a sentence X, we hope Y extracted by the model being faithful only to the content of the input X itself. And the backdoor adjustment promotes the posterior probability $P(Y|do(X))$ from passive observation to active intervention as shown below:

$$P(Y \mid do(X)) = \sum_{n} P(Y \mid X, n)P(n) \quad (4)$$

where n is the stratum for the confounder N. This encourages the model to maximize $P(Y|X, n)$ for every stratum n, only subject to a prior $P(n)$ listening to no one, and hence the model is deconfounded.

In the next sections, we apply Equation 4 to design two data augmentation (DA) methods, Intra-entity Deconfounding DA and Inter-entity Deconfounding DA, for the pre-context confounder and the entity-order confounder, respectively.

3.1 Intra-entity Deconfounding DA

We first focus on the generation of words inside the entity. The autoregressive decoder needs to decode the word at the current step conditioned on the pre-context words, i.e., the already generated word sequence. The pre-context words may be in other entities that are not associated with the entity currently being generated. Thus it will learn the wrong dependencies and bring in bias to the model. From the SCM in Figure 2, the pre-context words are the confounder in the generation of words inside the entity, causing the spurious correlation $X \leftarrow N \rightarrow Y$ to mislead the model from the true objective $X \rightarrow Y$.

Next we implement Intra-entity Deconfounding by data augmentation to eliminate pre-context confounder. As the backdoor adjustment shown in Equation 4, we stratify the confounder N, pre-context words, and train the model on each stratum. To avoid the influence of other entity words, we split the target sequences of the samples by entity and construct separate target sequences for each entity. Specifically, we randomly sample a context word $[CW]$ of an entity e^i from X and concatenate it in front of the entity as a target sequence Y', denoted as:

$$\{[CW], y_{e_1^i}, y_{e_2^i}, \cdots, y_{e_{\mathcal{E}}^i}\}$$

where $\mathcal{E}$ denotes the length of the entity e^i. If there are M entities in a sentence X, we can construct M augmented samples (X, Y'). It is worth noting that, compared to the target sequence Y of the original sample, the target sequence in the augmented sample does not contain tags denoting the beginning and end of the sequence, i.e., $[ss]$ and $[ee]$. This is to tell the model to generate only a single entity on the augmented sample instead of all the entities, as a way to prevent the model trained by the augmented samples from exiting early in the practical prediction.

3.2 Inter-entity Deconfounding DA

Another generation case is that after the current entity is generated, the model is expected to generate the first word of the next entity. In traditional generative NER models (Paolini et al., 2021; Yan et al., 2021a), the target sequence is fixed in the order of entities, for example, Yan et al. (2021a) pre-specified entity order according to the occurrence. However, entities are essentially set structures and the decoding sequence is not supposed to be fixed. A pre-specified entity order can make the optimization target inconsistent with the task and introduce an incorrect bias to the model. As shown in the SCM of Figure 2, entity order is the confounder N who affects the generation $X \rightarrow Y$ through the backdoor path $X \leftarrow N \rightarrow Y$.

According to Equation 4, we design an Inter-entity Deconfounding data augmentation to eliminate entity-order confounder. Similar to Section 3.1, we construct augmented samples by sampling from all possible entity orders. Specifically, for the original sample (X,Y), we keep the last entity of its target sequence fixed and permute the order of the other entities. The target sequence Y' of the augmented sample can be represented as:

$$\{[ss], \mathrm{Perm}\left(E_1, \cdots, E_{M-1}\right), E_M, [ee]\}$$

where $\mathrm{Perm}(\cdot)$ represents the permutation operation. During the training, we only compute the loss for the first token of the last entity, while the other entities are fed directly to the decoder as decoded sequences.

3.3 Constrained Prediction

As the model uses a token-by-token approach for prediction, in order to reduce the search space and the impact of exposure bias, we restrict the model to generating only tokens from the original sentence at generation time, and control the entire generation process by limiting tokens that can be generated at each step.

Specifically, we add special start and end tokens for the generation of each entity and the generation of the whole sequence. At the time of prediction, the generation of the sequence must start from the sequence start token, and the generation of the entity must start from the entity start token, and when the end token of the entity is generated, the next token that could be generated can only be the sequence end token and entity start token. Also, when generating each entity, we restrict the category of the entity to be generated only after the entity is generated, and the category can only be followed by the [e].

4 Experiments

In this section, we first describe the dataset we used, then we present related implementation details and experimental results, after which we make an analysis based on the experimental results.

4.1 Datasets

As same as (Yan et al., 2021b), to show that our proposed method can be used in various NER subtasks, we conducted experiments on eight datasets.

4.1.1 Flat NER Datasets

We selected the CoNLL2003 (Sang and Meulder, 2003) and OntoNotes (Pradhan et al., 2013b) datasets to do the experiments of Flat NER subtask. For CoNLL2003, we follow (Lample et al., 2016; Yu et al., 2020) to train our model on the concatenation of the train and development sets. For OntoNotes, we use the same train, development and test splits as (Pradhan et al., 2012; Yu et al., 2020).

4.1.2 Nested NER Datasets

For Nested NER subtask, we adopt ACE2004 (Doddington et al., 2004), ACE2005 and Genia datasets (Kim et al., 2003). In experiment conducted on ACE2004 and ACE2005, we use the same data split as (Lu and Roth, 2015; Muis and Lu, 2017; Yu et al., 2020), the ratio between train, development and test is 8:1:1. For Genia, we follow (Wang et al., 2020b; Shibuya and Hovy, 2020) to use five types of entities and split the train, development and test as 8.1:0.9:1.0.

4.1.3 Discontinuous NER Datasets

We follow (Dai et al., 2020) to use CADEC (Karimi et al., 2015), ShARe13 (Pradhan et al., 2013a) and ShARe14 (Mowery et al., 2014) datasets to do our experiment. Since only the Adverse Drug Events (ADEs) entities include discontinuous annotation, only this kind of entity is considered. (Karimi et al., 2015; Metke-Jimenez and Karimi, 2016; Tang et al., 2018).

4.2 Implementation Details

Because of the use of special tokens, we use the pre-trained language model T5 (Raffel et al., 2020) as our encoder-decoder generative architecture. The

Model	CoNLL2003			OntoNotes		
	Prec.	Rec.	F1	Prec.	Rec.	F1
(Clark et al., 2018)[GloVe300d]	-	-	92.6	-	-	-
(Peters et al., 2018)[ELMo]	-	-	92.22	-	-	-
(Akbik et al., 2019)[Flair]	-	-	**93.18**	-	-	-
(Straková et al., 2019)[BERT-Large]	-	-	93.07	-	-	-
(Yamada et al., 2020)[RoBERTa-Large]	-	-	92.40	-	-	-
(Li et al., 2020b)[BERT-Large]	92.47	93.27	92.87	**91.34**	88.39	89.84
(Yu et al., 2020)[BERT-Large]	**92.85**	92.15	92.5	89.92	89.74	89.83
(Yan et al., 2021b)(BPE)[BART-Large]	92.60	93.22	92.96	90.00	89.52	89.76
Ours[T5-Base](Without-De)	92.68	93.49	93.08	89.58	90.71	90.14
Ours[T5-Base](Intra-De)	92.78	93.51	93.14	89.77	**91.07**	**90.42**
Ours[T5-Base](Inter-De)	92.68	**93.57**	93.12	89.75	91.02	90.38

Table 1: Results for the Flat NER datasets.

Model	ACE2004			ACE2005			Genia		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
(Luan et al., 2019b)[ELMO]	-	-	84.7	-	-	82.9	-	-	76.2
(Straková et al., 2019)[BERT-Large]	-	-	84.33	-	-	83.42	-	-	76.44
(Shibuya and Hovy, 2020)[BERT-Large]	85.23	84.72	84.97	83.30	84.69	83.99	77.46	76.65	77.05
(Li et al., 2020b)[BERT-Large]	85.83	85.77	85.80	**85.01**	84.13	84.57	**81.25**	76.36	78.72
(Yu et al., 2020)[BERT-Large]	85.42	85.92	85.67	84.50	84.72	84.61	79.43	78.32	78.87
(Wang et al., 2020a)[BERT-Large]	86.08	**86.48**	**86.28**	83.95	85.39	84.66	79.45	78.94	**79.19**
(Yan et al., 2021b)(BPE)[BART-Large]	**86.69**	83.83	85.24	82.08	83.44	82.75	78.15	**79.06**	78.60
Ours[T5-Base](Without-De)	86.19	83.76	84.96	83.23	86.25	84.71	80.11	76.92	78.49
Ours[T5-Base](Intra-De)	86.36	84.54	85.44	83.31	86.56	84.90	81.04	77.21	79.08
Ours[T5-Base](Inter-De)	86.53	84.06	85.28	82.92	**87.05**	**84.93**	80.66	76.45	78.50

Table 2: Results for Nested NER datasets.

Model	CADEC			ShARe13			ShARe14		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
(Metke-Jimenez and Karimi, 2016)	64.4	56.5	60.2	-	-	-	-	-	-
(Tang et al., 2018)	67.8	64.9	66.3	-	-	-	-	-	-
(Dai et al., 2020)[ELMo]	68.9	69.0	69.0	80.5	75.0	77.7	78.1	81.2	79.6
(Yan et al., 2021b)(BPE)[BART-Large]	69.45	70.51	69.97	**82.07**	76.45	79.16	75.88	84.37	79.90
Ours[T5-Base](Without-De)	71.34	70.54	70.94	79.03	78.03	78.53	77.06	83.41	80.11
Ours[T5-Base](Intra-De)	**71.35**	**71.86**	**71.60**	81.09	**78.13**	**79.58**	**77.88**	**83.77**	**80.72**
Ours[T5-Base](Inter-De)	70.44	71.65	71.04	81.31	76.75	78.96	77.51	83.27	80.29

Table 3: Results for Discontinuous NER datasets.

T5 pre-trained model provides 100 default sentinel tokens for unsupervised training, here we use these special tokens to control the sequence generation process for avoiding the occupation of real tokens in the word list. Specifically, we use <*extra_id_2*> and <*extra_id_3*> to represent $[s]$ and $[e]$, <*extra_id_0*> and <*extra_id_1*> to represent $[ss]$ and $[ee]$, <*extra_id_11*> to <*extra_id_30*> to represent

different NER categories, and <*extra_id_50*> to mark the sample of inter-entity deconfounding samples. In addition, we use the AdamW (Loshchilov and Hutter, 2019) optimizer with a linear learning rate schedule (with peak learning rate of 1e-4). For simplicity, we assume that entities are unique, and for words with referential relations, such as *"we"*, which appears frequently in ACE2005, we tag each *"we"* with a different label in a sentence such as *"we_1"*, *"we_2"*, ... to distinguish them from each other.

4.3 Results

4.3.1 Comparision between Baselines

For simplicity of comparison, we use the results reproduced by (Yan et al., 2021b) on the dataset with different subtasks. Moreover, since we conducted the experiments on the subtoken-level, we only kept the experimental results of BPE in (Yan et al., 2021b). As can be seen from Tables 1 to 3, our model achieves similar or even better results on all three subtasks than the model in (Yan et al., 2021b). This may be caused by the fact that we use a different pre-trained model and not use pointer mechanism. Compared with other non-generative models, same as (Yan et al., 2021b), our method achieves comparable results with models focusing on only one subtask of NER on most of datasets, for exceptional cases, (Akbik et al., 2019) in Table 1 tags tokens at token-level; (Wang et al., 2020a) in Table 2 classifies candidate span, which integrates information of all subtokens in span, and is based on span-level; while our model only focuses on subtoken, which is based on subtoken-level.

4.3.2 Analysis of Intra-entity Deconfounding

In comparing the results of Without-De and Intra-De in Table 1-3, we can see that when intra-entity deconfounding are performed, the model has different degrees of improvement in all datasets. It is worth noting that the selection method we used to do augmentation differs slightly from dataset to dataset. Specifically, in each dataset we select entities considering on occurrence frequency, nesting status and character length of entity, in particular, we kick out some special entities that have referential relationships with others.

4.3.3 Analysis of Inter-entity Deconfounding

In comparing the results of Without-De and Inter-De in Table 1-3, we can see that when inter-entity deconfounding are performed, the model also have different degrees of improvement in all datasets. Here, it is worth noting that when selecting the sample for inter-entity deconfounding, we select samples based on factors with which the order confounder is most likely to have impact, such as the minimum order of last entity in the whole training dataset and whether it is easy to perform permutation such as the number of target entities. Besides, we have not select all samples for augmentation, and the results in Table 1-3 may not be the best.

CoNLL	Baseline			+ Intra-De		
	P	R	F1	P	R	F1
w/o attack	92.68	93.49	93.08	92.78	93.51	93.14
attack	91.43	93.38	92.39	92.51	93.27	92.88
Δ	-1.25	-0.11	-0.69	-0.27	-0.24	-0.26
				0.98↑	0.13↓	0.43↑
ACE04	**Baseline**			**+ Intra-De**		
	P	R	F1	P	R	F1
w/o attack	86.19	83.76	84.96	86.36	84.54	85.44
attack	80.24	83.32	81.75	83.01	84.34	83.67
Δ	-5.95	-0.42	-3.21	-3.35	-0.20	-1.77
				2.60↑	0.22↑	1.44↑
CADEC	**Baseline**			**+ Intra-De**		
	P	R	F1	P	R	F1
w/o attack	71.34	70.54	70.94	71.35	71.86	71.60
attack	69.46	67.91	68.67	71.14	70.14	70.64
Δ	-1.88	-2.36	-2.27	-0.21	-1.32	-1.06
				1.67↑	1.04↑	1.21↑

Table 4: Robustness Testing for the Pre-context Confounder

CoNLL	Baseline			+ Inter-De		
	P	R	F1	P	R	F1
w/o attack	98.42	98.48	98.45	98.19	98.31	98.25
attack	98.26	95.51	96.86	97.85	95.86	96.85
Δ	-0.26	-2.97	-1.59	-0.34	-2.45	-1.40
				0.08↓	0.52↑	0.19↑
ACE04	**Baseline**			**+ Inter-De**		
	P	R	F1	P	R	F1
w/o attack	94.51	92.09	93.28	93.95	91.41	92.66
attack	93.87	89.57	91.67	93.44	89.71	91.54
Δ	-0.64	-2.52	-1.61	-0.51	-1.70	-1.12
				0.13↑	0.82↑	0.49↑
CADEC	**Baseline**			**+ Inter-De**		
	P	R	F1	P	R	F1
w/o attack	93.65	90.53	92.06	93.94	91.30	92.60
attack	92.74	88.23	90.43	93.13	90.28	91.68
Δ	-0.91	-2.3	-1.63	-0.81	-1.02	-0.92
				0.10↑	1.28↑	0.71↑

Table 5: Robustness Testing for the Entity-order Confounder

4.4 Robustness Testing

To verify the effectiveness of the two data augmentation methods we designed for de-confounding, we conducted robustness testing experiments on CoNLL03, ACE04 and CADEC, respectively.

The pre-context confounder introduce error bias into the model by incorrectly relying on prefix sequences during entity sequence generation in the training phase. To verify the effectiveness of our Intra-entity Deconfounding Data Augmentation method in eliminating the pre-context confounder, we designed robustness testing experiments. In decoding, we randomly sample several words as pre-context sequences, and then require the model to continue decoding the entities. The experimental results are shown in Table 4. We can observe that the performance of both the baseline model and the Intra-entity Deconfounding model have different degrees of degradation after the attack of random fixed pre-context. However, the relative performance degradation of the Intra-entity Deconfounding model is less, and the Δ F1 on ACE04, CADEC and CoNLL are improved by+1.44%, +1.21% and +0.43% relative to the baseline model. This indicates that after Intra-entity Deconfounding Data Augmentation, the model can eliminate the pre-context confounder to some extent.

We also verify the robustness of the Inter-entity Deconfounding Data Augmentation method against the entity-order confounder. We first randomly sample k entities as the prefix of the decoding sequence, and then let the model continue to generate entities. For convenience, we choose a sample of the test set with the number of entities greater than k for evaluation, and we do not consider the k randomly sampled correct entities in our evaluation. In our experiments, $k = 4$. From Table 5, we can observe that the performance of both models decreases after the attack of random entity order. However, after deconfounding the entity sequences by the Inter-entity Deconfounding Data Augmentation method, the model degradation is reduced, and the Δ F1 on ACE04, CADEC, and CoNLL are improved by +0.49%, +0.71%, and +0.19% relative to the baseline model. This indicates that the Inter-entity Deconfounding Data Augmentation method we designed can enhance the robustness of the model to cope with random entity order when generating entity sequences, i.e., the entity-order confounder are eliminated to some extent.

5 Related Work

5.1 NER Task

The existing models can be basically divided into sequence labeling formulation, span-based formulation and generative-based formulation. Among them, the sequence labeling formulation was earlier applied to solve the NER problem (McCallum and Li, 2003; Collobert et al., 2011; Huang et al., 2015; Chiu and Nichols, 2016; Lample et al., 2016; Straková et al., 2019; Yan et al., 2019; Li et al., 2020a). After Nested NER and Discontinuous NER were discovered and raised, inspired by the successful application of sequence labeling formulation on Flat NER subtask, Metke-Jimenez and Karimi (2016); Muis and Lu (2017) attempted to extend this approach to the new subtasks. Others chose a different path, based on the characteristics of Nested NER, Xu et al. (2017); Wang and Lu (2019); Yu et al. (2020) try to traverse all possible spans and do classification at the span-level. Shen et al. (2021a) try to reduce the number of candidate spans and Tan et al. (2021) make the left and right boundaries of the candidate spans completely unfastened. In addition, in order to apply span-based formulation to the Discontinuous NER, the concept of hypergraph was introduced to efficiently represent spans (Lu and Roth, 2015; Katiyar and Cardie, 2018; Muis and Lu, 2016).

Although sequence labeling formulation and span-based formulation can be applied to different subtasks separately, these formulations are difficult to be applied to them simultaneously. Among them, sequence labelling formulation needs to design different tagging schema for different NER subtasks (Ratinov and Roth, 2009; Metke-Jimenez and Karimi, 2016; Straková et al., 2019; Dai et al., 2020), while span-based formulation needs to sacrifice a certain degree of performance. For example, span-based methods need to set a maximum span length to avoid the number of candidate spans to be traversed (Xu et al., 2017; Luan et al., 2019b; Wang and Lu, 2018), since it is impossible to enumerate all possible spans, which is quadratic to the length of the sentence and fragment numbers of discontinuous entity.

Contrary to sequence-labeling and span-based formulation, generative-based formulation can be used to model these subtasks in a unified manner because it can generate variable-length sequences (Yan et al., 2021b). However, since the generative model uses autoregressive generation, its optimiza-

tion objective differs significantly from the extraction objective of the NER task, which results in the model being influenced by some confounders and thus reduces the performance of model.

5.2 Causal Inference

Causal inference is a science that studies the relationship between correlation and causality. It is not only an explanatory framework, but also a way to provide solutions to achieve desired goals by pursuing causal effects (Pearl et al., 2016; Fenton et al., 2020). So far, it has been achieved greatly success in various domains such as psychology, politics and epidemiology for years (Mackinnon et al., 2007; Luke, 2015; Alves et al., 2014). Recently, causal inference has also attracted increasing attention in nature language process for improving model's performance in various ways. For example, Gardner et al. (2020) constructs counterfactual samples by manually rewriting the rules, and Garg et al. (2019) frames counterfactual samples by heuristically replace some keywords. Compared to them, our method offers a fundamental way to remove the confounder in training phase for generative models which is applied to various tasks of essentially non-sequential problem.

6 Conclusion

In this paper, we analyze two kinds of confounder that generative models arised when applied to NER and use backdoor adjustment methods in causal inference to perform deconfounding. Specifically, for pre-context confounder and entity-order confounder, we respectively design Intra-entity and Inter-entity De-confounding Data Augmentation methods. Experiments show that the performance of the model improves on all datasets after deconfounding. In the future, we will continue to explore the application of causal inference to other tasks.

Acknowledgments

This work is supported by the Key Research and Development Program of Zhejiang Province, China (No. 2021C01013), the Chinese Knowledge Center of Engineering Science and Technology (CKCEST) and MOE Engineering Research Center of Digital Library.

References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 724–728. Association for Computational Linguistics.

L. S. Alves, C. Susin, N Damé-Teixeira, and M. Maltz. 2014. Tooth loss prevalence and risk indicators among 12-year-old schoolchildren from south brazil. *Caries Research*, 48(4):347.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Trans. Assoc. Comput. Linguistics*, 4:357–370.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1914–1925. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cécile Paris. 2020. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5860–5870. Association for Computational Linguistics.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Norman E. Fenton, Martin Neil, and Anthony C. Constantinou. 2020. The book of why: The new science of cause and effect, judea pearl, dana mackenzie. basic books (2018). *Artif. Intell.*, 284:103286.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer

Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1307–1323. Association for Computational Linguistics.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*, pages 219–226. ACM.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. 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 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *J. Biomed. Informatics*, 55:73–81.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 861–871. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. In *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia*, pages 180–182.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. The Association for Computational Linguistics.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020a. FLAT: chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6836–6842. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5849–5859. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 857–867. The Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019a. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019b. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3036–3046. Association for Computational Linguistics.

K. Luke. 2015. The statistics of causal inference: A view from political methodology. *Political Analysis*, 23(3):313–335.

D. P. Mackinnon, A. J. Fairchild, and M. S. Fritz. 2007. Mediation analysis. *Annual Review of Psychology*, 58(1):593.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 188–191. ACL.

Alejandro Metke-Jimenez and Sarvnaz Karimi. 2016. Concept identification and normalisation for adverse drug event discovery in medical forums. In *Proceedings of the First International Workshop on Biomedical Data Integration and Discovery (BMDID 2016) co-located with The 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 17,*

2016, volume 1709 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Danielle L. Mowery, Sumithra Velupillai, Brett R. South, Lee M. Christensen, David Martínez, Liadh Kelly, Lorraine Goeuriot, Noémie Elhadad, Sameer Pradhan, Guergana K. Savova, and Wendy W. Chapman. 2014. Task 2: Share/clef ehealth evaluation lab 2014. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, volume 1180 of *CEUR Workshop Proceedings*, pages 31–42. CEUR-WS.org.

Aldrian Obaja Muis and Wei Lu. 2016. Learning to recognize discontiguous entities. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 75–84. The Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2608–2618. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Sameer Pradhan, Noémie Elhadad, Brett R. South, David Martínez, Lee M. Christensen, Amy Vogel, Hanna Suominen, Wendy W. Chapman, and Guergana K. Savova. 2013a. Task 1: Share/clef ehealth evaluation lab 2013. In *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013*, volume 1179 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013b. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 143–152. ACL.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning - Proceedings of the Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes, EMNLP-CoNLL 2012, July 13, 2012, Jeju Island, Korea*, pages 1–40. ACL.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*, pages 147–155. ACL.

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, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021a. Locate and label: A two-stage identifier for nested named entity recognition. 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 2782–2794, Online. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. 2021b. A trigger-sense memory flow framework for joint entity and relation extraction. In *Proceedings of the Web Conference 2021*, WWW '21, page 1704–1715, New York, NY, USA. Association for Computing Machinery.

Takashi Shibuya and Eduard H. Hovy. 2020. Nested named entity recognition via second-best sequence learning and decoding. *Trans. Assoc. Comput. Linguistics*, 8:605–620.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of*

the Association for Computational Linguistics, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5326–5331. Association for Computational Linguistics.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3936–3942. ijcai.org.

Buzhou Tang, Jianglu Hu, Xiaolong Wang, and Qingcai Chen. 2018. Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF. *Wirel. Commun. Mob. Comput.*, 2018.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 204–214. Association for Computational Linguistics.

Bailin Wang and Wei Lu. 2019. Combining spans into entities: A neural two-stage approach for recognizing discontiguous entities. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6215–6223. Association for Computational Linguistics.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020a. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5918–5928. Association for Computational Linguistics.

Yu Wang, Yun Li, Hanghang Tong, and Ziye Zhu. 2020b. HIT: nested named entity recognition via head-tail pair and token interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6027–6036. Association for Computational Linguistics.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1237–1247. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: adapting transformer encoder for named entity recognition. *CoRR*, abs/1911.04474.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021a. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021b. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6470–6476. Association for Computational Linguistics.