CCPrefix: Counterfactual Contrastive Prefix-Tuning for Many-Class Classification

Anonymous ACL submission

Abstract

Recently, prefix-tuning was proposed to efficiently adapt pre-trained language models to a broad spectrum of natural language classification tasks. It leverages soft prefix as task-specific indicators and language verbalizers as categorical-label mentions to narrow the formulation gap from pre-training language models. However, when the label space increases considerably (i.e., many-class classification), such a tuning technique suffers from a verbalizer ambiguity problem since the many-class labels are represented by semantic-similar verbalizers in short language phrases. To overcome this, inspired by the human-decision process that the most ambiguous classes would be mulled over for an instance, we propose a brand-new prefix-tuning method, Counterfactual Contrastive Prefix-tuning (CCPrefix), for many-class classification. Basically, an instance-dependent soft prefix, derived from fact-counterfactual pairs in the label space, is leveraged to complement the language verbalizers in many-class classification. We conduct experiments on many-class benchmark datasets in both the fully supervised setting and the few-shot setting, which indicates that our model outperforms former baselines.

1 Introduction

Although fine-tuning paradigm has achieved great success in natural language processing, effectively transferring knowledge to specific tasks, there remains a considerable gap between pre-training and fine-tuning, which can inhibit the transfer and adaptation of knowledge in PLMs to downstream tasks. This gap primarily arises from the diverse objective forms that downstream tasks take on. To narrow this gap, Prompt-tuning (Brown et al., 2020; Schick et al., 2020) has been proposed to unify the objective of different tasks into a cloze-style task to predict target words. Compared to the prevalent fine-tuning, the prompt-tuning paradigm is consistent with language model pre-training and thus generalizable by with few learnable parameters (Brown et al., 2020; Trinh and Le, 2018; Petroni et al., 2019; Davison et al., 2019).

Instance:
As a stage actor, Greg has been a resident company member of the Alley Theatre in Houston, Texas.
Q: The type of Greg is ________.
A. Person-Actor **B**. Person-Employee

Why Person-Actor?
As a stage actor, Greg has been a resident company member of the Alley Theatre in Houston, Texas.

Why Person-Actor not Person-Employee?
As a stage actor, Greg has been a resident company member of the Alley Theatre in Houston, Texas.

Figure 1: An illustrative example of entity typing task from FewNERD (Ding et al., 2021) dataset. Option A is its ground-truth label, and Option B is the counterfactual. Red words are the related attributes for the question.

To bridge the gap to masked language models (MLMs), a task-specific template and verbalizers, are necessary to form a cloze-style task and achieve prompt tuning. Normally, the template can be a natural language prompt or a series of continuous tokens to query the language model, while the verbalizers are usually natural language phrases to represent task-specific labels. For example, in natural language inference (NLI), a training instance can be concatenated with a natural language prompt "[Premise] [MASK] [Hypothesis]". As such, a set of label words is designed as the candidate set for filling into that placeholder (e.g., `[MASK]`) in the designed template. Again, in NLI, the verbalizers are defined as {*Then*, *Maybe* and *But*}, corresponding the three-class categories {*entailment*, *neural* and *contradiction*}. Obviously, it is relatively tractable for experts to select valid label words as there are clearly semantic bounds among these mutual-exclusive labels.

However, with the increase of label space, the semantic boundary among many-class labels becomes obscure, which may overlap leading to the verbalizer ambiguity problem. This explains why some works (Webson and Pavlick, 2022; Cao et al., 2021) point out that the performance is quite sensitive to the choice of label words. For instance, as shown in Fig 1, "Person-Actor" and "Person-Employee" are the common classes in the entity typing task and share the same hypernym word "Person". To overcome the verbalizer ambiguity problem, Han et al. (2021) manually designs logic rules to merge several sub-prompts together as the final prompt for each class, however, limited by costly expert-required logic rules.

Taking inspiration from the social science research (Miller, 2019), we adopt the contrastive procedure of human explanation to generate diverse information prefixes for training instances. Concretely, rather than explaining "why A", it is more effective to explain "why A not B", where B serves as an implicit counterfactual of A within the current context. In Figure 1, we present an instance from the FewNERD (Ding et al., 2021) dataset, where the task is to classify the type associated with Greg. From a machine learning perspective, a well-trained model will recognize that Greg is associated with multiple attributes, including "Houston", "company" and "actor", all of which are deemed valuable for prediction. As illustrated in Figure 1, these contributed attributes can be redundant for prediction as highlighting. Hence, the contrastive explanation approach tends to overlook most similarity attributes between "Employee" and "Actor", focusing instead on the more salient semantics that are critical for the model's differentiation task.

In this paper, we propose Counter-factual Contrastive Prefix-tuning, dubbed CCPrefix [1], which aims to minimize semantic obscurity among verbalizers and mitigate the problem of verbalizer ambiguity. Our process begins by constructing all possible fact-counterfactual label pairs, with each class alternately assumed as the fact while the other classes are treated as counterfactuals. Each instance is then projected onto the subspaces spanned by these fact-counterfactual pairs, generating a range of potential contrastive attributes. These potential attributes are subsequently filtered through a global prototype alignment learning method, resulting in an instance-dependent soft prefix. Lastly, we employ a straightforward Siamese representation

[1] We will open our codes (uploaded), data, and models.

Algorithm 1 Contrastive Attributes Construction

Input: the class set $\mathcal{Y}$, instance x, a PLM model $\mathcal{M}$
Output: Contrastive attributes $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$

1: Initialize the verbalizer $\boldsymbol{V} = \phi(\mathcal{Y}) \in \mathbb{R}^{|R|\times d_e}$
2: Initialize the matrix $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$
3: Obtain instance representation $\boldsymbol{h}_x = \mathrm{Pool}(\mathcal{M}(x))$
4: **for all** $\boldsymbol{v}_i \in \boldsymbol{V}$ **do**
5:     **for all** $\boldsymbol{v}_j \in \boldsymbol{V}, i \neq j$ **do**
6:         Construct the contrastive subspace $\boldsymbol{u}_{i,j} = \boldsymbol{v}_i - \boldsymbol{v}_j \in \mathbb{R}^{d_e}$
7:         Project the instance onto the subspace $\boldsymbol{c}_{i,j} = \frac{\boldsymbol{u}_{i,j}\otimes\boldsymbol{u}_{i,j}^\top}{\langle\boldsymbol{u}_{i,j}^\top\boldsymbol{u}_{i,j}\rangle}\boldsymbol{h}_x$
8:     **end for**
9:     Form $\boldsymbol{C}_{i,*}$ representing the attributes between i-th fact and the other label
10: **end for**
11: **return** $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$

learning approach for each instance to ensure stability throughout the training process. This methodical multi-step approach strives to reduce ambiguity and enhance the effectiveness of prefix-tuning in the realm of natural language processing.

To comprehensively validate the efficacy of CCPrefix, we conduct extensive experiments on three many-class classification tasks in both fully supervised and few-shot settings, including relation classification, topic classification and entity typing. The experimental results suggest that Our work presents a promising step forward in the field, demonstrating the substantial potential of CCPrefix in handling complex classification tasks in natural language processing.

2 Methodology

In this section, we will detail our approach, whose overall architecture is shown in Figure 2.

Task Definition. First of all, we provide the task definition about the classification problem in fine-tuning paradigm. The classification tasks can be denoted as $\mathcal{T} = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X}$ is the instance set, $\mathcal{Y} = \{y_1, y_2, \ldots, y_{|R|}\}$ is the class set, and $|R|$ is the number of classes. The first token of the input is `[CLS]` which contains the special classification embedding. PLMs models take the hidden state $\boldsymbol{h}$ of the first token `[CLS]` as the representation of the whole sequence. A simple softmax classifier is then added to the top of PLMs to predict the probability of class y_c:

$$p(y_c|\boldsymbol{h}) = \mathrm{Softmax}(\boldsymbol{W}\boldsymbol{h}) \quad (1)$$

where $\boldsymbol{W}$ is the task-specific parameter matrix. Both the parameters from PLMs and $\boldsymbol{W}$ will

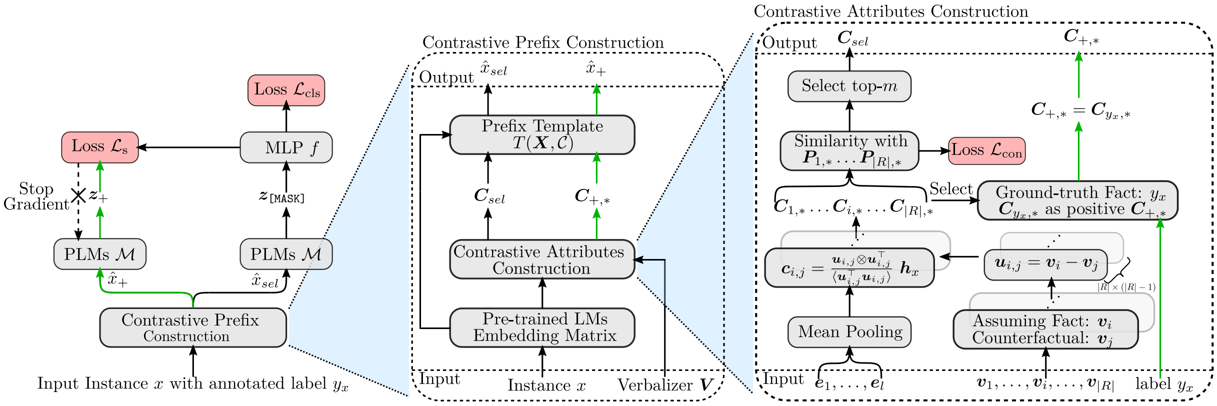

Figure 2: Our proposed model, CCPrefix. For easy comprehension, we zoom out contrastive prefix construction and contrastive attributes generation in Section 2.2. The losses $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{s}}$ and $\mathcal{L}_{\text{con}}$ are defined in Equation (9), Equation (8) and Equation (5). The black line is the forward path for both training and inference, while the green line is the training path with supervised signal.

be jointly fine-tuned by maximizing the log-probability of the correct label.

2.1 Prefix Tuning for Classification

Formally, prefix tuning consists of a series prefix tokens $\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_m\}$ and a verbalizer $\phi : \mathcal{V} \rightarrow \mathcal{Y}$ that bridges the class set $\mathcal{Y}$ and the set of answer words $\mathcal{V}$. To construct the cloze-style tasks, at least one placeholder [MASK] should be placed into the template for the PLMs, $\mathcal{M}$, as the following shows:

$$T(\boldsymbol{X}, \boldsymbol{C}) = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_l, \boldsymbol{c}_1, \dots, \boldsymbol{c}_m, \boldsymbol{e}_{\texttt{[MASK]}}\}, \tag{2}$$

where $\{\boldsymbol{e}_1, \dots, \boldsymbol{e}_l\}$ is the embedding of instance $\boldsymbol{X}$. With the soft prefix template $T(\cdot)$ and the verbalizer ϕ, the learning objective is to maximize $\frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X}} \log p(\texttt{[MASK]} = \phi(y_x) | T(x))$.

2.2 Contrastive Prefix Construction

We would elaborate on the process of exploring all potential contrastive attributes from each instance and the way we construct the prefix templates.

Contrastive Generation. Thus, for classification tasks, following (Jacovi et al., 2021), we construct all causal factors by projecting the sentence representation into the contrastive space. First of all, each instance x would be encoded by a deep neural encoder $f(\cdot)$ that transforms x into $\boldsymbol{X} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_l\} \in \mathbb{R}^{l \times d_e}$, where l is the sentence length and d_e the embedding dimension. Then, we use a multi-layer perception (MLP) with ReLU activation, and mean pooling over the sequence to get the whole sentence representation, $\boldsymbol{h}_x = \text{Pool}(\text{MLP}(\boldsymbol{X}))$.

Commonly, the prediction of the model $\boldsymbol{W}\boldsymbol{h}_x$ is linear in the latent input representation. The processor of prediction aims to map $\boldsymbol{h}_x$ to a specific direction $\boldsymbol{w}_i$ via dot product to obtain the logits of class i. As proposed by Jacovi et al. (2021) in terms of contrastive explanation, given two classes, y_p and y_q, if we are particularly interested in the contrastive attributes that the model predicts y_p rather than y_q, we can construct a new basis, $\boldsymbol{u}_{p,q} = \boldsymbol{w}_p - \boldsymbol{w}_q$, which represents a *contrastive space* for y_p and y_q. Thus, y_p is the fact while y_q is one of its counterfactuals. However, for each instance, the golden label is unavailable before prediction. Hence, we hypothesize that the i-th class y_i is the fact in turn while the rest in the finite-label space are counterfactuals to build fact-counterfactual pairs. Specifically, we employ the derivable vectors as the verbalizer $\boldsymbol{V} \in \mathbb{R}^{|R| \times d_e}$ to map to the class set $\mathcal{Y}$. Thus, supposing that i-th class y_i is the fact while one of the rest class y_j is the counterfactual, the contrastive subspace is:

$$\boldsymbol{u}_{i,j} = \boldsymbol{v}_i - \boldsymbol{v}_j \in \mathbb{R}^{d_e}, i \in |R|, j \neq i \tag{3}$$

Then, by projecting the instance representation $\boldsymbol{h}_x$ onto the subspace $\boldsymbol{u}_{i,j}$, the contrastive attribute between the specific fact-counterfactual pair is explored:

$$\boldsymbol{c}_{i,j} = \frac{\boldsymbol{u}_{i,j} \otimes \boldsymbol{u}_{i,j}^\top}{\langle \boldsymbol{u}_{i,j}^\top \boldsymbol{u}_{i,j} \rangle} \boldsymbol{h}_x \tag{4}$$

where $\otimes$ is the outer product and $\langle \cdot \rangle$ is the inner product. For the contrastive attributes generated between the same fact and the rest counterfactuals, we denote these attributes as $\boldsymbol{C}_{i,*} \in \mathbb{R}^{(|R|-1) \times d_e}$,

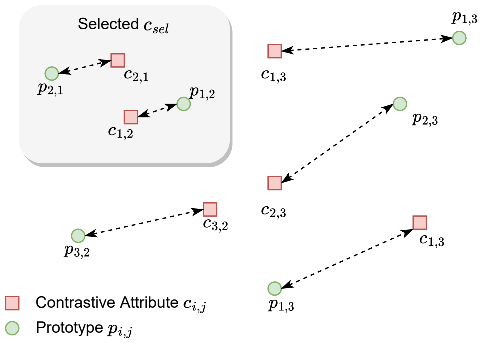

Figure 3: An illustration of the selection process of top-2 contrastive attributes $\boldsymbol{c}_{i,j}$ using the similarities between all possible $\boldsymbol{c}_{i,j}$ and their corresponding prototypes $\boldsymbol{p}_{i,j}$, where i-th class is fact and j-th class is its counterfactual.

where $i, *$ represents the fact-counterfactual pairs consisting of the i-th fact and the rest labels assumed as counterfactuals. Sequentially operating eq.3 and eq.4, we extract all contrastive attributes $\boldsymbol{C} \in \mathbb{R}^{|R| \times (|R|-1) \times d_e}$ from each instance. We summarize the former procedure of constructing contrastive attributes in Algorithm 1.

Prototype Constraint. Obviously, since we suppose each label as the fact to form fact-counterfactual pairs in turn, it is inevitable to face the noisy attributes projected by invalid fact-counterfactual pairs for each instance. Therefore, the contrastive attributes should be selected only if it is generated by the valid fact-counterfactual pairs formed by the accurate label. To distinguish valid contrastive attributes, we introduce a set of global prototypes $\{\boldsymbol{P}_{0,*}, \boldsymbol{P}_{1,*}, \dots, \boldsymbol{P}_{|R|,*}\} \in \mathbb{R}^{|R| \times (|R|-1) \times d_e}$ corresponding to contrastive attributes. Concretely, for the contrastive attributes $\boldsymbol{c}_{i,j}$ generated by projecting instance onto the subspace between i-th fact and j-th counterfactual, there is only one corresponding prototype $\boldsymbol{p}_{i,j}$. The fine-grained global prototypes can learn the common features of its corresponding fact-counterfactual attribute among the whole training instances. During training, according to the instance's ground-truth label, these prototypes can be split into two groups. One is the set of positive prototypes while the other is the rest negative prototypes $\boldsymbol{P}_{-,*} \in \mathbb{R}^{(|R|-1) \times (|R|-1) \times d_e}$. The positive prototypes represent the common knowledge of the corresponding attributes $\boldsymbol{C}_{+,*}$ generated by the valid fact-counterfactual pairs. These prototypes are trained with the following self-contrastive learning loss:

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(\langle \boldsymbol{W}\boldsymbol{C}_{+,*}, \boldsymbol{P}_{+,*} \rangle)}{\sum_{-} \exp(\langle \boldsymbol{W}\boldsymbol{C}_{+,*}, \boldsymbol{P}_{-,*} \rangle))} \quad (5)$$

where $\boldsymbol{W} \in \mathbb{R}^{d_e \times d_e}$ is the learning weight matrix and $\langle \cdot \rangle$ is the inner product to calculate the similarity. This objective forces the positive prototypes draw up the positive contrastive attributes. Simultaneously, the negative contrastive attribtues would be pushed away from the positive prototypes.

Prefix Construction. Thus, by calculating the similarities between instance's contrastive attributes and the corresponding prototypes, we select the top-m's most similar attributes $\boldsymbol{C}_{sel} \in \mathbb{R}^{m \times d_e}$ as additional prefix tokens, as shown in Figure 3. The selected contrastive attributes will be considered as a series tokens in the prefix template $T(\cdot)$, as Equation (2).

2.3 Siamese Prefix Tuning Objective

We note that some selected top-m contrastive attributes may inevitably take false classes as facts, thereby introducing unwanted noise. Therefore, it is crucial to force the PLMs to focus on the valid contrastive attributes and consequently stabilize the model performance. Hence, we leverage a simple Siamese representation learning method (Chen and He, 2021) to simultaneously train the PLMs, $\mathcal{M}$, via maximizing the similarity between the prefix templates with selected contrastive attributes $\boldsymbol{C}_{sel}$ and the same instance with all positive attributes $\boldsymbol{C}_{+,*}$. These two inputs with different contrastive attributes are fed into $\mathcal{M}$ to obtain the `[MASK]` representation $\boldsymbol{z}$ and $\boldsymbol{z}_{+}$:

$$\begin{aligned} \boldsymbol{z} &= \mathcal{M}(\hat{\boldsymbol{X}}) = T(\boldsymbol{X}, \boldsymbol{C}_{sel}), \\ \boldsymbol{z}_{+} &= \mathcal{M}(\hat{\boldsymbol{X}}_{+}) = T(\boldsymbol{X}, \boldsymbol{C}_{+,*}). \end{aligned} \quad (6)$$

Then, we minimize the negative cosine similarity between two outputs with an MLP $f(\cdot)$:

$$\mathcal{D}(\boldsymbol{z}, \boldsymbol{z}_{+}) = -\frac{f(\boldsymbol{z})}{||f(\boldsymbol{z})||_2} \cdot \frac{\boldsymbol{z}_{+}}{||\boldsymbol{z}_{+}||_2} \quad (7)$$

Following Chen and He (2021), we use a symmetrized loss with the stop-gradient operation:

$$\mathcal{L}_{\text{s}} = \frac{1}{2}\mathcal{D}(f(\boldsymbol{z}), \text{sg}(\boldsymbol{z}_{+})) + \frac{1}{2}\mathcal{D}(f(\boldsymbol{z}_{+}), \text{sg}(\boldsymbol{z})). \quad (8)$$

Here, $\boldsymbol{X}$ with attributes $\boldsymbol{C}_{+,*}$ receives no gradient from $\boldsymbol{z}_{+}$ in the first term, but it receives gradients from $f(\boldsymbol{z}_{+})$ in the second term, and vice versa.

Finally, the learning objective is to minimize the following loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{|\mathcal{X}|}\sum_{k=1}^{|\mathcal{X}|} \log p(\texttt{[MASK]} = \boldsymbol{v}_k | x_k) \quad (9)$$

where $p(\texttt{[MASK]} = v_k|x_k)$ is the predicted distribution for the k-th sample in dataset $\mathcal{X}$ and $\boldsymbol{v}_k$ is the answer word corresponding to its ground truth label y_k. Overall, our final training loss is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{s}} + \mathcal{L}_{\text{con}} \quad (10)$$

3 Experiments

We conduct experiments on several classification tasks, including relation classification (RC), topic classification (TC) and entity typing (ET).

3.1 Datasets

We adopt 4 popular datasets for relation classification, i.e., TACRED (Zhang et al., 2017), TACREV (Alt et al., 2020), ReTACRED (Stoica et al., 2021) and SemEval 2010 Task 8 (Hendrickx et al., 2009) (SemEval), one for topic classification, i.e., DBPedia (Lehmann et al., 2015), and one for entity typing, i.e., FewNERD (Ding et al., 2021).

- **TACRED**, **TACREV** and **ReTACRED** are used widely for relation classification. While TACRED is the origin, TACREV and ReTACRED are its revised versions with modifications in test sets and some relation tpyes.

- **SemEval** is a traditional dataset for RC.

- **DBPedia** is an ontology dataset with structured information extracted from WikiPedia. We privately set a 10% of the training dataset as the validation set.

- **FewNERD** is a manually large-scale dataset of entity typing containing 66 fine-grained entity types. We focus on the inter-task, where train/dev/test splits may share coarse-grained types while keeping the fine-grained entity types mutually disjoint.

More details of these datasets are shown in Table 1. For evaluation, we use F_1 scores as the metric for RC, and mean accuracy for TC and ET.

Dataset	#Class	Task	$\lvert\mathcal{D}_{\text{train}}\rvert$	$\lvert\mathcal{D}_{\text{dev}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$
TACRED	42	RC	68,124	22,631	15,509
TACREV	42	RC	68,124	22,631	15,509
ReTACRED	40	RC	58,465	19,584	13,418
SemEval	19	RC	6,507	1,493	2,717
DBPedia	14	TC	56,000	5,600	70,000
FewNERD	66	ET	338,753	48,667	96,901

Table 1: Basic statistics of the datasets, where RC stands for relation classification, TC stands for topic classification, and ET stands for entity typing.

3.2 Settings

To fairly compare with SoTA baselines, we evaluate CCPrefix under fully supervised and few-shot settings for RC tasks, and exclusively in few-shot settings for TC and ET, where for each class, K instances are sampled for training and validation. Following previous works (Han et al., 2021; Cui et al., 2022), we set K as 8, 16, 32 for relation classification and 1, 2, 4, 8, 16 for topic classification and entity typing. We use a fixed set of 5 random seeds to sample instances and take the average of all results as the final result.

3.3 Implementation Details

Our model is implemented based on PyTorch (Paszke et al., 2019) with V100 and the Transformer repository of Huggingface (Wolf et al., 2020). For RC and TC tasks, our model is based on RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019), while for ET, it is based on BERT$_{\text{BASE}}$ (Devlin et al., 2019). Adam optimizer (Kingma and Ba, 2015) is used for all datasets, where the learning rate is manually tuned $\in$ {1e-5, 3e-5, 5e-5 }, and the decay rate is set to 1e-2, and the batch size is set to 16. For the fully-supervised setting, the epoch is 5 while for few-shot setting, it is 30. The best model is selected based on the performance on the development set. We select top-m attributes as prefix, where $m = |R| - 1$.

3.4 Comparison Methods

We mainly compare CCPrefix with several representative methods in many-class classification tasks, including learning-from-scratch methods, fine-tuning methods and Prefix-tuning methods. 1) C-GCN (Zhang et al., 2018) is a learning-from-scratch based on graph neural networks for relation classification. 2) For fine-tuning vanilla PLMs, we directly select RoBERTa$_{\text{LARGE}}$ as our baselines for relation classification. 3) Since entity informa-

	Extra Data	TACRED	TACREV	ReTACRED	SemEval
C-GCN (Zhang et al., 2018)	-	66.3	74.6	80.3	-
RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019)	-	68.7	76.0	84.9	87.6
KnowBERT (Peters et al., 2019)	✓	71.5	79.3	-	89.1
SpanBERT (Joshi et al., 2020)	✓	70.8	78.0	85.3	-
LUKE (Yamada et al., 2020)	✓	**72.7**	80.6	90.3	-
PTR (Han et al., 2021)	-	72.4	81.4	90.9	89.9
CCPrefix (Ours)	-	72.6	**82.9**	**91.2**	**90.6**
w/o ConAtt in §2.2	-	70.0	80.9	90.6	90.1
w/o Prototypes in §2.2	-	71.9	81.2	90.5	90.4
w/o $\mathcal{L}_{\text{con}}$ in Eq.5	-	71.3	81.8	90.6	90.2
w/o Siamese in §2.3	-	72.0	81.8	90.8	90.1

Table 2: F_1 scores (%) for RC tasks on the 4 datasets in the fully supervised setting. "w/o ConAtt" denotes using manually Prefix template and soft verbalizer. "w/o Prototypes" denotes that the cluster is rely on the verbalizer. "w/o Siamese" denotes that the input of Prefixs template only maintain instance and selected contrastive attribute.

	TACRED			TACREV			ReTACRED		
	8	16	32	8	16	32	8	16	32
Fine-Tuning (Ours)	12.2	21.5	28.0	13.5	22.3	28.2	28.5	49.5	56.0
PTR (Han et al., 2021)	28.1	30.7	32.1	28.7	31.4	32.4	51.5	56.2	62.1
CCPrefix (Ours)	**30.1**	**33.4**	**37.6**	**29.8**	**33.0**	**34.0**	**54.5**	**61.4**	**65.2**
w/o ConAtt in §2.2	18.1	29.6	32.6	18.1	29.0	32.7	41.1	55.5	64.1
w/o Prototypes in §2.2	28.5	33.1	36.3	30.4	31.7	33.2	54.2	56.3	62.1
w/o $\mathcal{L}_{\text{con}}$ in Eq.5	28.2	33.2	37.3	28.9	32.1	33.8	53.5	59.7	64.4
w/o Siamese in §2.3	23.8	33.1	32.9	27.9	30.4	33.2	50.6	57.7	63.4

Table 3: F_1 scores (%) for RC tasks in the few-shot setting. We use $K = 8, 16, 32$ for few-shot settings.

tion is crucial in relation classification, we select SpanBERT (Joshi et al., 2020), KnowBERT (Peters et al., 2019) and LUKE (Yamada et al., 2020) as our baselines. 4) We select PTR (Han et al., 2021), a prompt augmentation model, for relation classification. 5) For topic classification and entity typing, our baselines are ProtoVerb (Cui et al., 2022) that uses manual prompts, and PETAL (Schick et al., 2020) that extracts words as prompts.

3.5 Main Quantitative Evaluation

We compare CCPrefix with several recent methods to conduct an in-depth analysis.

Fully Supervised Setting As indicated in Table 2, CCPrefix significantly outperforms former baselines, even surpassing KnowBERT and LUKE that leverage external task-specific knowledge to enhance models. Compared to PTR (Han et al., 2021), which manually constructs logic rules as the prompt, CCPrefix even outperforms. Such comparison indicates that the unique task-related information to form a unique prefix can better stimulate task-specific knowledge in PLMs.

Few-Shot Setting To further assess our model, we evaluate CCPrefix in few-shot settings. For relation classification, as shown in Table 3, CCPrefix outperforms PTR, with an average improvement of 6.6% on ReTACRED. For topic classification, as shown in the left panel of Table 4, CCPrefix exceeds PETAL and ProtoVerb by a large margin. Specifically, in the extreme data scarce scenario ($K = 1, 2$), our model surpasses ProtoVerb by 15.3% and 9.1%. This demonstrates that, if the class labels are semantically diverse, our model is capable of acquiring sufficient knowledge from the PLM even in this limit. For entity typing, our model exceeds former baseline in several scenarios ($K = 4, 8, 16$) but not good when training instances are extremely scarce ($K = 1, 2$). We infer that for fine-grained entity typing, although our model can cancel out most of the attributes between two classes sharing the same coarse class with subtle differences in semantic (e.g., 'building-theater" and "building-library" are under type "building"), it is hard to discriminate such contrastive attributes in extreme data scarce scenario.

	DBPedia					FewNERD				
	1	2	4	8	16	1	2	4	8	16
PETAL (Schick et al., 2020)	60.06	78.21	86.40	88.41	92.90	20.88	31.28	43.10	50.78	55.49
ProtoVerb (Cui et al., 2022)	72.85	85.49	90.91	95.75	96.30	**25.00**	**35.72**	48.28	56.06	61.29
CCPrefix (Ours)	**84.02**	**93.26**	**95.17**	**97.66**	**98.45**	22.78	32.47	**51.49**	**58.54**	**63.38**

Table 4: Few-Shot TC & ET performance of F_1 scores (%) on the DBPedia and FewNERD datasets. We use $K = 1, 2, 4, 8, 16$ for few-shot settings.

Relation	Top selected counterfact
per:siblings	*per:title*
per:parents	*per:countries_of_residence*
org:dissolved	*org:member_of*
per:origin	*org:dissolved*
per:children	*per:country_of_birth*
per:city_of_birth	*per:city_of_death*
per:employee_of	*per:countries_of_residence*
per:religion	*per:city_of_death*
org:alternate_names	*org:founded_by*
per:cause_of_death	*per:country_of_death*
org:website	*org:members*

Table 5: The top selected counterfactual relation learned by the model for some relation types.

3.6 Ablation Study

We carry out an ablation study on relation classification datasets to further invetigate the effectiveness of each component in CCPrefix, as detailed in the bottom panel of Table 2 and Table 3. "w/o ConAtt" causes more performance degradation in the few-shot setting than in the fully supervised one, which indicates that contrastive attributes can further stimulate the knowledge in PLMs. For "w/o Prototypes", attribute-verbalizer similarities are used as the slection criteria, causing a significant performance drop due to noise attributes, although it slightly outperforms CCPrefix in TACREV under K=8. "w/o $\mathcal{L}_{\text{con}}$" has less performance reduction in few-shot setting than that in fully supervised setting. We infer that the unbalanced training data distribution may hurt the performance significantly. The performance of "w/o Siamese" drops severely in the extreme data scarce scenario ($K = 8$), indicating that simple representation learning can force the PLMs to focus on the valid contrastive attributes in prefix.

3.7 Selected Counterfact

Since the prefix are instance aware, we limit our analysis to a subset of 7K instances in the test set that could be correctly classified. For each relation type, we count the most frequently selected counterfactual relation. Part of the results are shown in Table 5. It is notable that most of the time the model can match a pair *per* relations, or a pair of *org* relations. Also, the model prefers to select two relation types semantically correlated but with subtle differences. For example, for relation *per:city_of_birth* or *org:dissolved*, the corresponding contrastive attribute factor is *per:city_of_death* or *org:member_of*, respectively.

3.8 Case Study

To analyze the influence of individual tokens on model prediction, we conduct a case study on the relation *per:city_of_birth* between entities "`he`" and "`Potomac`". "`Potomac`", as depicted in Figure 4. We compute the similarity between each word and the fact y^*=*per:city_of_birth*, as well as the contrastive attribution factor between y^*=*per:city_of_birth* and y'=*per:city_of_death*. For clarity, words with similarity scores exceeding the average are highlighted. Our results reveal that the contrastive attribute factor yields concentrated, key determinant highlights such as "native of". In contrast, using y^* alone results in scattered highlights, diverging from human expectations of the significant predictors.

3.9 Error Analysis

Our model operates under the strong assumption that all labels, save for the golden one, act as counterfactuals of the golden label. This hypothesis neglects the semantic correlations and overlaps among different classes, potentially impacting model performance. This issue is especially apparent in the entity typing task, where fine-grained entity types mayu semantically overlap, thereby challenging our assumption. When class labels possess subtly distinct semantics, more data is needed to construct valid contrastive attributes. This can cause model performance to drop in scenarios of extreme data scarcity, like with the FewNED dataset at $K = 1, 2$.

*y**=*per:city_of_birth*	(*y**, *y'*)=*per:city_of_birth, per:city_of_death*
Gross , a 60-year-old native of Potomac , Maryland , was working for a firm contracted by USAID when he was arrested Dec 3 , 2009 , and sent to Cuba 's high-security Villa Marista prison .	Gross , a 60-year-old native of Potomac , Maryland , was working for a firm contracted by USAID when he was arrested Dec 3 , 2009 , and sent to Cuba 's high-security Villa Marista prison .

Figure 4: The highlighted tokens of the same sentence where the two entities are underscored. On the left, the tokens are projected onto the ground truth *y**=*per:city_of_birth*, and on the right onto the contrastive space between *y** and the counterfactual *y'=per:city_of_death*.

4 Related Work

Prefix Tuning in Classification. The templates can be categorized into two groups, i.e., discrete prompt (Brown et al., 2020; Schick et al., 2020; Schick and Schütze, 2021) and continuous prefix (Lester et al., 2021; Li and Liang, 2021). Discrete prompts often manually designed for all training instances with task descriptions. Han et al. (2021) leverage manual logic rules to combine label-related sub-prompts together. Although it is a concrete manifestation of human's interpretation of the task, discrete prompts may not be the optimal solution. Continuous prefixes (Lester et al., 2021; Li and Liang, 2021), attached to instances, have proven useful but fail to fully capture the diversity of training instances. Our work inspired by the human decision process, introduces an instance-dependent prefix, better addressing the discrimination of label space.

Verbalier in Classification. Reformulating problems as language modeling tasks has been explored in few-shot scenarios (Brown et al., 2020; Trinh and Le, 2018; Petroni et al., 2019; Davison et al., 2019). Traditional manual verbalizer mappings demand expert knowledge, thus making automatic verbalizer search (Schick et al., 2020; Schick and Schütze, 2021) an appealing alternative. This approach iteratively enhances the label-to-word mapping in a greedy fashion.

Counterfactual Contrastive. Explanation of artificial intelligence is widely concerned in recent years. Miller (2019) presents the philosophical foundations of explanation that human relies on the contrastive explanations. Jacovi et al. (2021) highlights the attributes in the latent space to provide fine-grained explanation of model decision. Furthermore, Ross et al. (2021) produces contrastive explanations by editing the inputs for the contrast case while Gardner et al. (2020) uses it for evaluation. Paranjape et al. (2021) builds contrastive prompts with instance-specific information for explanation. Zhang et al. (2020) employs contrastive counterfactuals with the multi-instance framework for vision-language grounding. Kaushik et al. (2020) tasks humans with revising dataset to revise the dataset with counterfactuals. Meanwhile, Yang et al. (2021) produces high-quality augmented data with counterfactuals to overcome out-of-distribution data in the field. Due to the strong explanation of counterfactual, we leverage counterfactual to disambiguate the semantic overlap between labels.

5 Conclusion

In this paper, we propose a novel task-agnostic approach named CCPrefix. We sequentially construct fact-counterfacutal pairs to extract the attributes from the sample. With a set of global prototypes, the valid contrastive attributes will be selected as the prefix. A simple Siamese represeatation learning is employed to stable the training process. The experiment results verify the superiority of our model without extra data and human experts for manually designing Prefix templates. While we have shown that our method is flexible enough for a wide range of tasks in NLP, leveraging contrastive explanations in logic reasoning tasks remains an unveiled challenge for future work.

Limitations

A principal limitation of our CCPrefix model is the strong assumption it makes in the classification task: it regards all labels other than the gold standard as counterfactuals. This premise may not consistently hold true, particularly in scenarios involving hierarchical labels with overlapping semantics. This assumption may impact the performance.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1558–1569. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. 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 1860–1874. Association for Computational Linguistics.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7014–7024. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander M. Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. 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 3198–3213. 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.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, SEW@NAACL-HLT 2009, Boulder, CO, USA, June 4, 2009*, pages 94–99. Association for Computational Linguistics.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1597–1611. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes A difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597. Association for Computational Linguistics.

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

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Prompting contrastive explanations for commonsense reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4179–4192. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? 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 2463–2473. Association for Computational Linguistics.

Alexis Ross, Ana Marasovic, and Matthew E. Peters. 2021. Explaining NLP models via minimal contrastive editing (mice). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3840–3852. Association for Computational Linguistics.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the TACRED dataset. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13843–13850. AAAI Press.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2300–2344. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. 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.

Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 306–316. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2205–2215. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45. Association for Computational Linguistics.

Zhu Zhang, Zhou Zhao, Zhijie Lin, Jieming Zhu, and Xiuqiang He. 2020. Counterfactual contrastive learning for weakly-supervised vision-language grounding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.