DSP: Discriminative Soft Prompts for Zero-Shot Entity and Relation Extraction

Bo Lv[1,2,3], **Xin Liu**[2]*, **Shaojie Dai**[1,2,3]
Nayu Liu[1,2], **Fan Yang**[2,3], **Ping Luo** [1,2,3]* **and Yue Yu**[2]*
[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
[2]Peng Cheng Laboratory, [3]University of Chinese Academy of Sciences
{lvbo19,daishaojie22,liunayu18,yang22}@mails.ucas.ac.cn
hit.liuxin@gmail.com, luop@ict.ac.cn, yuy@pcl.ac.cn

Abstract

Prompt-based methods have shown their efficacy in transferring general knowledge within pre-trained language models (PLMs) for low-resource scenarios. Typically, prompt-based methods convert downstream tasks to cloze-style problems and map all labels to verbalizers. However, when applied to zero-shot entity and relation extraction, vanilla prompt-based methods may struggle with the limited coverage of verbalizers to labels and the slow inference speed. In this work, we propose a novel Discriminative Soft Prompts (DSP) approach to take advantage of the prompt-based methods to strengthen the transmission of general knowledge. Specifically, we develop a discriminative prompt method, which reformulates zero-shot tasks into token discrimination tasks without having to construct verbalizers. Furthermore, to improve the inference speed of the prompt-based methods, we design a soft prompt co-reference strategy, which leverages soft prompts to approximately refer to the vector representation of text tokens. The experimental results demonstrate that, our model outperforms baselines on two zero-shot entity recognition datasets with higher inference speed, and obtains a 7.5% average relation F1-score improvement over previous state-of-the-art models on Wiki-ZSL and FewRel.

1 Introduction

Zero-shot entity and relation extraction (Levy et al., 2017; Chen and Li, 2021) aim to extract novel entities and their relations by transferring semantic knowledge from seen classes to unseen ones. It is a fundamental problem in information extraction, which can be decomposed into two subtasks: zero-shot named entity recognition (ZSNER) (Li et al., 2020, 2022) and zero-shot relation extraction (ZSRE) (Sainz et al., 2021). Recent works (Li et al., 2020; Chen and Li, 2021) focus on fine-tuning PLMs with extra classifiers to leverage the rich lexical, syntactic, and factual knowledge (Petroni et al., 2019) within PLMs to compensate for the lack of domain knowledge in the task training. However, the significant objective gap between pre-training and fine-tuning may drive the parameters of the PLMs away from their initial values, resulting in a substantial loss of general knowledge.

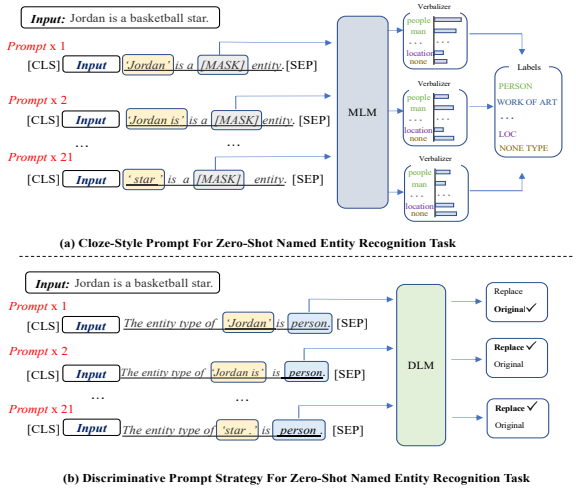

Figure 1: Cloze-style prompt with BERT (MLM) and discriminative prompt with ELECTRA (DLM) for ZS-NER task. The underlined text is the task-specific template. The yellow-filled token is the candidate entity, and the blue-filled token in Figure (b) is the label name. Predicting all labels in the sentence "Jordan is a basketball star. " requires enumeration over all spans.

Recent efforts (Ding et al., 2021) on probing knowledge have demonstrated that formalizing downstream tasks in the same form as pre-training is an efficient way to enhance the transmission of general knowledge. Inspired by this, prompt-based learning (Schick and Schütze, 2021) that reformulates downstream tasks as cloze questions has been introduced. Typically, for the entity type classification task, a template is used to convert [X] into a cloze task (e.g.,"[X] *E* is a [MASK] entity."), where [X] is the placeholder for input sentences, and *E*

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 5491–5505
July 9-14, 2023

is a candidate entity to be classified. The PLMs, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), are asked to infer the words to fill in [MASK], and the words are further mapped to corresponding labels through a verbalizer (e.g., "people" for label "PERSON").

However, two issues impede the application of cloze-style prompts to the zero-shot entity and relation extraction task as follows: (1) we can see from Figure 1(a) that multiple words or phrases can represent the same class. Building a verbalizer that can cover all candidate words comprehensively is challenging in a zero-shot setting, and a poorly designed verbalizer can limit the accuracy of predictions. (2) for the entity recognition task, the n-grams method needs to be employed for enumeration when generating candidate entities, resulting in a serious efficiency issue. Figure 1(b) shows that prompt methods need to run $n(n+1)/2$ times to recognize all entities in the sentence of length n during inference, which is unacceptable in real-world applications.

We argue that the primary reason for these limitations is that the existing prompt-based methods imitate masked language model (MLM), which needs to map the labels to verbalizers. Unlike MLM, the token discrimination task of discriminative pre-trained models (DLM) appears to be more compatible with zero-shot entity and relation extraction.

In this work, we introduce a Discriminative Soft Prompts approach, which utilizes prompt discriminative language models (e.g., ELECTRA (Clark et al., 2020)) to address the general knowledge forgetting problem caused by modifying the structure of PLMs. Specifically, we present a discriminative prompt strategy, which leverages the label information to construct a template to convert input sentences into a discriminative language modeling problem.

As shown in Figure 1(b), our discriminative prompt method recognizes candidate entities by classifying entity type into binary categories (i.e., original, replaced), thereby bridging the gap between pre-training and task-training without the need for verbalizers. Furthermore, we design a soft prompt co-reference strategy, which leverages soft prompts to approximately refer to the vector representation of text tokens. By classifying soft prompts into binary categories, the inference speed of our model has significant improvement. Especially for entity recognition task, it only needs to run the model once to extract all entities of the same type in the sentence.

Extensive experiments are conducted on two zero-shot tasks, ZSNER and ZSRE. Specifically, our method advances the state-of-the-art SMXM(Aly et al., 2021) model on two ZSNER datasets and gains a 7.5% average relation F1-score improvement over the previous best model on WikiZSL and FewRel. Moreover, the inference speed of the DSP is up to 120 times faster than the cloze-style prompt method on ZSNER. Our main contributions are summarized as follows:

- We reformulate ZSNER and ZSRE as token discrimination tasks, taking advantage of the prompt method to strengthen the transmission of general knowledge without having to construct a verbalizer.
- We propose a soft prompt co-reference strategy, which significantly improves the inference efficiency of the discriminative prompt method for zero-shot entity and relation extraction tasks.
- Experiments on four datasets demonstrate the effectiveness of our model in both ZSNER and ZSRE tasks. Moreover, the inference speed of the DSP is up to 120 times faster than the cloze-style prompt method on ZSNER.

2 Method

In this section, we first formally define the problem of zero-shot entity and relation extraction. Then we introduce our Discriminative Soft Prompts (DSP) method. The following is a detailed description.

2.1 Problem Definition

In zero-shot entity and relation extraction, the goal is to learn from the seen dataset and generalize to the unseen dataset. The seen and unseen label sets are denoted as $C_s = \{c_1, c_2, ..., c_n\}$ and $C_t = \{\hat{c_1}, \hat{c_2}, ..., \hat{c_m}\}$, where $n = |C_s|$ and $m = |C_t|$ are the sizes of seen and unseen label sets, and $C_s \cap C_t = \emptyset$. Let $S = \left\{s_1, s_2, ..., s_{\frac{L(L+1)}{2}}\right\}$ be all the possible spans in sentence X up to length L. The problem can be decomposed into two subtasks:

Zero-Shot Named Entity Recognition The task is to predict an entity type $y_e(s_i) \in C_t$ or $y_e(s_i) = \emptyset$ for each span $s_i \in S$, indicating that span s_i is not an entity. The output of the task is $Y_{zsner} =$

$\{(s_i, e) : s_i \in S, e \in C_t\}$, where Y_{zsner} is the set of all (s_i, e) pairs such that span $_i$ is associated with entity type e, and $s_i \in S$ and $e \in C_t$.

Zero-Shot Relation Extraction The task is, for every pair of spans $s_i \in S$ and $s_j \in S$, to predict a relation type $y_r(s_i, s_j) \in C_t$, or to indicate that there is no relation between them with $y_r(s_i, s_j) = \emptyset$. The output of the task is $Y_{zsre} = \{(s_i, s_j, r) : s_i \in S, s_j \in S, r \in C_t\}$.

2.2 Discriminative Prompt strategy

Discriminative pre-trained language models (DLMs) (Yao et al., 2022; Xia et al., 2022) are a compelling alternative to mask pre-trained language models (MLMs) and have shown potential for low-resource scenarios. By casting NLP tasks as a discriminative language modeling problem, discriminative prompt strategies can help bridge the gap between pre-training and task-specific tuning. The inputs of DLM are formulated with an input sentence X, and a template $\mathcal{T}$. As shown in Figure 1(b), an example of ZSNER with the entity types set $C = \{c_1, c_2, ..., c_n\}$, we define a template that contains candidate entity $\mathcal{T}(\cdot, e, c)$. Given an input text X (e.g., "Jordan is a basketball star."), DLM fills the input text into the template as follows:

$$\mathcal{T}(X, e, c_i) = X, \text{The entity type of } e \text{ is } c_i \quad (1)$$

Where e refers to the candidate entity, i is i-th entity type belonging to C. After template filling, $\mathcal{T}$ is fed into DLM to obtain the hidden representations $\left\{h_{[CLS]}, h_1, ..., h_n, ..., h_e, ..., h_{c_i}, h_{[SEP]}\right\}$. The model then discriminates whether the entity type c_i is accurate, and the score of c_i is calculated as follows:

$$\mathcal{D}(T([c_i])) = 1 - \sigma(h_{DLM}^{\top} h_{[c_i]}) \quad (2)$$

where h_{DLM} is the reused classifier of DLM, and $\sigma(\cdot)$ is the sigmoid activation. DLM then rounds the output scores into binary categories, i.e., {0, 1} corresponding {replaced, original}. If an entity type name consists of multiple tokens, such as " WORK OF ART", we perform an *or* operation on the binary classification results of all tokens. Since the pre-training task of DLM is similar to this task, it bridges the gap between pre-training and downstream tasks without requiring a verbalizer design.

2.2.1 Problems of Prompt-based Methods For ZSNER

Nevertheless, prompt-based methods have high complexity in solving ZSNER task. During the process of inference, the candidate entities s_j^i that denote the span starting from x_i and ending with x_j need to be enumerated in order to obtain all possible spans:

$$s_j^i = Enumerate(\{x_i, ..., x_j\}, i, j \in \{1..n\}) \quad (3)$$

For example, the template of MLM can take the form as "X, The entity type of s_j^i is [MASK].", where the cloze-style prompt method predicts an entity label word at [MASK] (e.g., people) corresponding to an entity label (e.g., PERSON). As the sequence length increases, the decoding time also increases, rendering this decoding method time-consuming.

2.3 Discriminative Soft Prompts Co-reference For ZSNER

We propose a soft prompts co-reference strategy to solve the slow inference speed of DLM on ZSNER tasks. The main idea of our method is to perform binary classification twice for each token in the text, to identify whether it is the head or tail token of an entity. For example, given the input X as "Badaling Great Wall is located...", the model performs binary classification twice for each token. This yields a sequence [1,1,0], representing whether the token is the head of an entity, and a sequence [0,0,1], representing whether the token is the tail of an entity. Finally, the head and tail tokens are combined as nearest neighbors to obtain two entity span s_1^3 ("Badaling Great Wall") and s_2^3 ("Great Wall"), respectively. However, DLM has only one classification layer, and adding a new layer could disrupt the model's structure and potentially harm its performance. To address this, we design two soft prompts ($[s], [e]$), which can be easily incorporated into the input with minimal modification. It is worth noting that we only use two soft prompts, which are copied to refer to all tokens.

Figure 2 illustrates that we first link the position ID of each token with two soft prompts and assign them identical position embeddings. Through multi-layer Transformer calculations, the embedding of the soft prompts will be closest in proximity to the token that has the same position embedding.

To avoid damaging the fluency of sentences, we next modify the attention mask matrix, which is shown in Appendix B. Specifically, each soft prompt is only visible to partnering soft prompts that refer to the same span and is invisible to text tokens. At the same time, the soft prompts attend upon the text tokens to aggregate information from their corresponding spans. By classifying these soft prompts, we obtain a matrix of span positions to identify all entities.

Formally, we form a new sequence $\widehat{X}$ consisting of soft prompts, original text, and template:

$$\widehat{X} = x_1, ..., x_l, [s_1]..., [s_l], [e_1]..., [e_l], t_1, ...t_m \tag{4}$$

where X is a sequence of l text tokens, and $[s_l]$, $[e_l]$ represent the soft prompts that have the same position embedding as the l-th token x_l.

As illustrated in Figure 2, we input $\widehat{X}$ to the DLM and obtain:

$$\mathcal{D}(\widehat{X}([s1])) = 1 - \sigma(h_{DLM}^{\top} h_{[s_1]}) \tag{5}$$

The DLM outputs the "original" label corresponding to the positions $[s1]$, $[s2]$, $[e3]$, indicating that $([s1], [e3])$ $(Badaling\ Great\ Wall)$ and $([s2], [e3])$ $(Great\ Wall)$ belong to the entity type of "Work Of Art".

2.4 Discriminative Soft Prompts Co-reference For ZSRE

We adopt a pipeline approach to tackle the Zero-Shot Relation Extraction (ZSRE) task. Specifically, we employ DSP-ZSNER to identify entity mentions, and then perform Zero-Shot Relation Classification (ZSRC) task to classify the relations between all pairs of mentions. The discriminative prompt method can only determine if two entities belong to a particular relation in the prompt template at a time. Therefore, if there are K preset relations, the model needs to run K times to determine the relationship between the two entities. To improve inference speed on the ZSRC task, we employ a co-reference strategy, similar to the approach used in the ZSNER task.

Given an input sequence X and two entity spans e_s and e_o, we utilize a soft prompt $[r]$ to represent the relation between the two entities in the template. All labels in the relation label set share the same position embedding as $[r]$, enabling each label to obtain the contextual representation of $[r]$ approximately. Furthermore, to maintain semantic integrity, labels in the relation label set are not visible to each other in the mask matrix. We form a new sequence $\widehat{X}$ consisting of the soft prompt, relation label set, original text, and template, given by:

$$\widehat{X} = x_1, ..., x_n, e_o, ..., [r], ...e_s, r_1, ..., r_i, ...r_K \tag{6}$$

We input $\widehat{X}$ into the model and obtain:

$$\mathcal{D}(\widehat{X}(r_i)) = 1 - \sigma(h_{DLM}^{\top} h_{[r_i]}) \tag{7}$$

If $\mathcal{D}(\widehat{X}(r_i))$ outputs *original*, it indicates that the relation between e_s and e_o is r_i.

2.5 Training Loss Function

To facilitate optimization and prevent overfitting, the final training loss combines the cross-entropy (CE) loss with parameter regularization loss (λ is a hyper-parameter).

$$\mathcal{L} = \mathcal{L}(ce) + \lambda \mathcal{L}(w) \tag{8}$$

The cross-entropy loss is as follows:

$$\begin{aligned} \mathcal{L}(ce) = \sum_i (-y_i log \mathcal{D}(\widehat{X}(c_i)) \\ -(1 - y_i) log(1 - \mathcal{D}(\widehat{X}(c_i)))) \end{aligned} \tag{9}$$

The parametric regularization is defined as:

$$\mathcal{L}(w) = \frac{1}{2} \sum_{j \in S} (w_j - w_j^0)^2 \tag{10}$$

where w_j^0 represents the initial parameters of the j-th layer of the pre-trained language model, and w_j represents the parameters of the j-th layer of the discriminative prompt model during task-training.

3 Experiments

3.1 Setup

Datasets For the ZSNER task, we evaluate our approach on two popular zero-shot NER datasets: OntoNotes 5.0[1] (Pradhan et al., 2013) and MedMentions (Mohan and Li, 2019). To assess the model's performance in recognizing nested entities, we follow Aly et al. (2021) to gather all entities of each type from the dataset and mapped them into sentences based on their respective types. As shown in Table 8, the datasets are divided into a training set, development set and test set according to the entity type. For ZSRE task, we utilize the

[1] https://catalog.ldc.upenn.edu/LDC2013T19

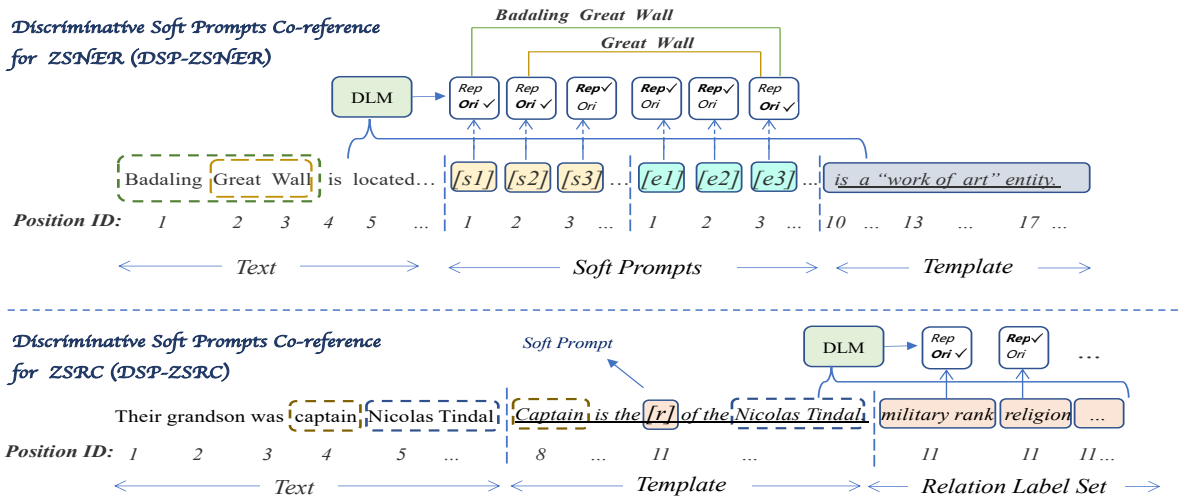

Figure 2: An overview of our discriminative soft prompts co-reference strategy for zero-shot named entity recognition task and zero-shot relation classification task. $[s]$, $[e]$, $[r]$ are the soft prompts. DSP-ZSNER performs binary classification for each $[s]$ and $[e]$ to obtain the position IDs of head and tail tokens of each entity, and then connects them to form a complete entity, (*e.g.* head tokens position id {1,2}, end token position id {3}, connect them we get two entities *Badaling Great Wall* (1,3) and *Great Wall* (2,3)). DSP-ZSRE performs binary classification for each label (*e.g.*, *military rank*) in relation label set to determine the relationship between entities (*e.g.*, *'caption'* and *'Nicolas Tindal'*).

Dataset	#Sents	#Ents (#Types)	#Rels (#Types)
OntoNotes-ZS	76.7k	58.1k (18)	-
MedMentions-ZS	46.9k	116.2k (21)	-
Wiki-ZSL	94.3k	-	77.6k (113)
FewRel	56.0k	-	72.9k (80)

Table 1: The statistics of the adopted datasets.

datasets released by Chia et al. (2022) and adhere to their prescribed splitting method[2] (consisting of five folds) for both training and evaluation. Additionally, we make use of their recommended data pre-processing methods. Table 1 shows the statistics of each dataset. For each dataset, we set the unseen label size to $m \in \{5, 10, 15\}$, while treating the remaining labels as seen labels during training in the experiments. Details of the datasets are described in Appendix C.

Evaluation metrics We follow the standard evaluation protocol and use F1-score as the evaluation metric. For ZSNER task, the unbalanced number of samples per class necessitates employing evaluation metrics that focus on per-class averaged scores to properly account for the imbalance. Therefore, like Aly et al. (2021), we evaluate our model using the macro average F1-score indicator. We evaluate ZSRC using the Macro F1 metric to be consistent with Chia et al. (2022). ZSRE first identifies entities, then predicts the relation between each pair of entities, which will predict a large number of negative samples. Thus we use the micro F1-score, which is standard in structured prediction tasks (Zhong and Chen, 2020) and report the precision (P.) and recall (R.). The details for evaluation metrics are in Appendix D.

Implementation details We adopt ELECTRA-Base as the backbone of our model and initialize with the corresponding pre-trained cased weights. Models are implemented using Pytorch framework[3] and Huggingface transformers[4]. DSP-ZSNER and DSP-ZSRC are optimized by AdamW (Loshchilov and Hutter, 2017) with the learning rate of 2e-5. The training batch size used is 16 for all models. For ZSNER task, the soft prompts and entity-type descriptions take up part of the input length, so the maximum length of the DSP is limited to 150 tokens. For the ZSRE task, the maximum length of input token is 256. For both tasks, we employ an early stopping scheduler to stop training when there is no improvement on the validation F1 score. We then conduct three runs of experiments to mitigate instability issues for all experiments[5].

[2]Available here: https://github.com/declare-lab/RelationPrompt

[3]https://pytorch.org

[4]https://github.com/huggingface/transformers

[5]See Appendix E for further details on the hyperparameter.

	Ontonotes-ZS		MedMentions-ZS	
Model	**Dev**	**Test**	**Dev**	**Test**
BEM	18.0	11.0	19.0	22.0
MRC	15.0	18.0	21.0	26.0
SMXM(base)	19.0	20.0	20.0	21.0
SMXM	23.0	25.0	23.0	27.0
DLM-Point	18.3	28.4	26.1	28.6
DSP-ZSNER	**27.0**	**31.6**	**29.8**	**32.7**

Table 2: The macro-averaged F1-score of ZSNER on OntoNotes-ZS and MedMentions-ZS, the best results are highlighted in bold. The SMXM model employs Bert-Large encoders, while our DLM-Point and DSP-ZSNER models adopt Electra-Base. The remaining models all use Bert-Base.

Unseen Labels	Model	Wiki-ZSL			FewRel		
		$P_.$	$R_.$	F_1	$P_.$	$R_.$	F_1
m=5	R-BERT	39.2	43.2	41.1	42.2	48.6	45.2
	CIM	49.6	48.8	49.2	58.0	61.9	59.9
	ZS-BERT	71.5	72.4	72.0	77.0	78.9	77.9
	NoGen	51.8	46.8	48.9	72.4	58.6	64.6
	RelationPrompt	70.7	**83.8**	76.6	90.2	88.5	89.3
	DSP-ZSRC	**94.1**	77.1	**84.8**	**93.4**	**92.5**	**92.9**
m=10	R-BERT	26.2	29.7	27.8	25.5	33.0	28.2
	CIM	46.5	47.9	45.6	47.4	49.1	48.2
	ZS-BERT	60.5	61.0	60.7	56.9	57.6	57.2
	NoGen	54.9	36.5	43.8	66.5	48.3	55.6
	RelationPrompt	68.5	**74.8**	71.5	80.3	79.6	80.0
	DSP-ZSRC	**80.0**	74.0	**76.9**	**80.7**	**88.0**	**84.2**
m=15	R-BERT	17.3	18.8	18.0	17.0	19.4	18.1
	CIM	29.2	30.6	29.9	31.8	33.1	32.4
	ZS-BERT	34.1	34.4	34.3	35.5	38.2	36.8
	NoGen	54.5	29.4	37.5	66.5	40.1	49.4
	RelationPrompt	63.7	**67.9**	65.7	74.3	72.5	73.4
	DSP-ZSRC	**77.5**	64.4	**70.4**	**82.9**	**78.1**	**80.4**

Table 3: Zero-Shot Relation Classification, the best results are highlighted in bold.

3.2 Zero-Shot Named Entity Recognition

3.2.1 Baselines

We compare our DSP-ZSNER with current state-of-the-art models in both NER and related zero-shot tasks. **Binary Entailment Model(BEM)** is a ZSNER model obtained by modifying the state-of-the-art zero-shot text classification model (Yin et al., 2019) by Aly et al. (2021). They add a binary output layer based on BERT to generate binary output for each class, and the negative prediction of all classes predicts negative classes. **MRC** is an approach by Li et al. (2020) who construct queries for entity classes and modifie the model structure to transform NER into fully supervised machine reading comprehension tasks for flat and nested entities. Similar to MRC, **SMXM** (Aly et al., 2021) uses entity type descriptions to aid encoding, and subsequently feeds the entity encoding into a linearly transformed layer for classification. **DLM-Point** is a DLM-based sequence labeling method proposed by us, which is introduced in Appendix A.

3.2.2 Results

We show the ZSNER result in Table 2. Some observations are summarized from the experimental results: (1) Our approach outperforms the baselines based on fine-tuning by modifying the structure of PLMs on both Ontonotes-ZS and MedMentions-ZS datasets, and obtains a +6.3% F1-score improvement on MedMentions-ZS. MedMentions-ZS contains twice as many entities as Ontonotes-ZS and has a low correlation between training and testing data. It shows that the DSP-ZSNER can well preserve the initial general knowledge of the PLMs to better model the interrelation between entities and entity types. (2) With the same PLM (Electra-Base), DSP-ZSNER achieves an absolute F1 improvement of +8.7% over DLM-Point on Ontonotes-ZS dev, which shows the advantage of soft prompts co-reference strategy in identifying nested entities. In addition, the soft prompt, which explicitly represents the boundary of the span, is also a key factor for the improvement.

3.3 Zero-Shot Relation Classification

3.3.1 Baselines

There are four main categories of competing methods for the ZSRC task. **R-BERT** (Wu and He, 2019) is a relation classification model, but it can adapt to the zero-shot setting by designing a matching module based on BERT to perform the nearest neighbor search over the label embeddings. **CIM** (Rocktäschel et al., 2015) is an entailment-based method that takes sentences and each possible relation as input to determine whether the relation matches the sentence semantically. **ZS-BERT** (Chen and Li, 2021) learns the independent projection function to align input sentences with their candidate relations in the embedded space and to judge the relation between pairs of entities by measuring their distances in a new space. **RelationPrompt** (Chia et al., 2022) prompts GPT2 (Radford et al., 2019) to generate synthetic data, and modifies the Bart (Lewis et al., 2020) generation decoder to learn the ability to generate relation triplets from these data. **NoGen** indicates that it does not use generated synthetic samples for training and the other settings are the same as RelationPrompt.

3.3.2 Results

By providing entity-pair information in the prompt template, DSP can convert ZSRC task to the exact same task format as ELECTRA pre-training.

Unseen Labels	Model	Pre-trained Model	Wiki-ZSL			FewRel		
			$P.$	$R.$	F_1	$P.$	$R.$	F_1
m=5	TableSequence (Wang and Lu, 2020)	GPT-2	**43.7**	3.5	6.3	15.23	1.9	3.4
	NoGen (Chia et al., 2022)	BART	15.6	43.2	22.3	9.5	**36.7**	14.6
	RelationPrompt (Chia et al., 2022)	GPT-2& BART	29.1	31.0	30.0	20.8	24.3	22.3
	DSP-ZSNER & DSP-ZSRC	ELECTRA	42.7	**43.4**	**43.0**	**40.1**	27.0	**32.3**
m=10	TableSequence (Wang and Lu, 2020)	GPT-2	**45.3**	3.6	6.4	28.9	3.6	6.4
	NoGen (Chia et al., 2022)	BART	9.6	45.0	15.7	6.4	**41.7**	11.0
	RelationPrompt (Chia et al., 2022)	GPT-2& BART	30.2	32.3	31.2	21.6	28.7	24.6
	DSP-ZSNER & DSP-ZSRC	ELECTRA	26.3	**48.0**	**34.0**	**35.9**	27.1	**30.9**
m=15	TableSequence (Wang and Lu, 2020)	GPT-2	**44.4**	3.5	6.4	19.0	2.0	3.5
	NoGen (Chia et al., 2022)	BART	7.3	**43.7**	12.3	4.6	**36.4**	8.1
	RelationPrompt (Chia et al., 2022)	GPT-2& BART	26.2	32.1	28.9	17.7	23.2	20.1
	DSP-ZSNER & DSP-ZSRC	ELECTRA	27.7	32.4	**29.9**	**27.9**	25.4	**26.6**

Table 4: The results for Zero-Shot Relation Extraction (ZSRE), best results are highlighted in bold. DSP-ZSNER & DSP-ZSRC refers to that we utilize DSP-ZSNER model to recognize entities, and then classify the relation between every two entities using the DSP-ZSRC model.

Model	Ontonotes-ZS		MedMentions-ZS	
	F_1	Speed (sent/s)	F_1	Speed (sent/s)
MLM (BERT)	6.3	0.3	3.7	0.3
SMXM	24.0	28.0	25.0	27.2
DLM	30.5	0.1	32.0	0.1
DLM-Point	23.4	86.4	27.4	78.4
DSP-ZSNER	29.3	41.6	31.3	36.0

Table 5: Average macro F1-score and efficiency of dev and test data on ZSNER benchmark. MLM is the cloze-style prompt method, which utilizes BERT-Base as encoder. The maximum length of MLM, SMXM, DLM and DLM-Point is 256, while the maximum length of DSP-ZSNER with soft prompts co-reference is 512.

As shown in Table 3, our approach outperforms previous methods by strict F1-score of +6.1% on Wiki-ZSL and +4.7% on FewRel. It is worth noting that our prompt-based method retains more general knowledge of PLM. When the invisible label set size m increases and the training data decreases, our prompt-based method can utilize this knowledge to maintain relatively high classification $F1$ performance. This trend suggests that our prompt-based method can be better extended to a larger set of invisible tags, which is more critical for real-world open domain applications.

3.4 Zero-Shot Relation Extraction

3.4.1 Baselines

For the ZSRE, we use several baseline methods provided by Chia et al. (2022) for comparison with our DSP method. **TableSequence** (Wang and Lu, 2020) is a table-based method that extracts entity relations by encoding different types of information in the learning process. Since it cannot directly solve ZSRE, Chia et al. (2022) used the composite samples from the relation generator to provide supervision data for it. Other methods have been described in Section 3.3.1.

3.4.2 Results

For ZSRE, we use a pipeline approach to train DSP-ZSNER and DSP-ZSRC models, respectively. During the inference phase, the DSP-ZSNER model extracts entities from the text and then classifies the pairs of entities using the DSP-ZSRC model. We compare DSP with the baselines on ZSRE for Wiki-ZSL and FewRel datasets in Table 4, our approach consistently outperforms the previous best methods in F1-score metrics. Compared to the previous state-of-the-art model, RelationPrompt, our approach achieves an absolute F1 improvement of +13% and +10.0% on Wiki-ZSL and FewRel, respectively, with fewer parameters, under the setting of $m = 5$. Such improvement from RelationPrompt indicates the effectiveness of modeling through pre-training tasks to limit excessive changes in model parameters during task-tuning.

3.5 Inference Speed

In this section, we compare the model's inference speed on an V100 GPU with a batch size of 32.

Speed of ZSNER We conduct an evaluation of the inference speed of BERT, SMXM, DLM, DLM-Point, and DSP on the Ontonotes-ZS and MedMentions-ZS datasets. The results are presented in Table 5, which indicate that our DSP-ZSNER model achieved a higher F1-score and faster inference speed than SMXM. Despite sacrificing 1.2% and 0.7% F1-score on Ontonotes ZS

Model	F_1	LM Loss	Parameter Variation
BERT	-	5.5	-
BERT+FFN	19.5	38.0	2609.5
ELECTRA	-	3.7	-
DSP-ZSNER	29.3	4.6	8.7
DSP(w/o PR)	27.1	7.9	70.8
ElECTRA+FFN	23.9	12.1	163.2

Table 6: After ZSNER task-tuning, the language model's cross-entropy loss on the pre-training task (LM Loss) and the variation of parameters compared with the original pre-training model. BERT and ELECTRA are the initial PLMs without task-tuning. BERT+FFN refers to adding two full connection layers on the basis of BERT to realize ZSNER tasks. The parameter variation is calculated using the L2 distance, and the abbreviation "w/o PR" refers to without parameter regularization.

and MedMentions-ZS, respectively, DSP-ZSNER obtained 416x and 360x speedup compared to the DLM model. Moreover, DSP-ZSNER achieved a speedup of up to 138.7x and 120x compared to MLM on the two datasets, with an F1-score increase of +23% and +27.6%. These results indicate that it is appropriate to utilize the soft prompts co-reference strategy to identify entities is an effective way to solve the problem of slow inference speed in prompt methods.

Speed of ZSRC We compare our methods to the best previous method, RelationPrompt. Table 7 shows that the inference speed of our DSP-ZSRC model is faster than RelationPrompt on both datasets. RelationPrompt needs to be trained at the inference stage using pseudo data generated by the GPT-2, which reduces its inference efficiency. Under the setting of an unseen label of 10, the DLM needs to run ten times to predict the relation between entities. DSP-ZSRC with soft prompts co-reference strategy can discriminate all candidate relations in one run, obtaining a 5.9× speedup on Ontonotes-ZS and a 6.5× speedup on MedMentions-ZS. On the other hand, this strategy only leads to a small performance drop and the F1-score decreases by only 0.2% and 0.3% on the two datasets.

3.6 Analysis of Parameter Variation and LM Loss

To investigate the impact of retaining the knowledge acquired during pre-training phase of the PLMs on zero-shot tasks, we conduct a ZS-

Model	Wiki-ZSL		FewRel	
	F_1	Speed (sent/s)	F_1	Speed (sent/s)
RelationPrompt	71.5	63.1	80.0	59.8
DLM	77.1	13.8	84.5	11.6
DSP-ZSRC	76.9	81.6	84.2	76.0

Table 7: Under the unseen label of 10 on the ZSRC task, the comparison between our DSP model, and RelationPrompt, in F1-score and speed.

NER task-tuning experiment on the Ontonotes-ZS dataset. BERT+FFN denotes the addition of two fully connected layers based on BERT to perform ZSNER tasks. Similarly, ElECTRA+FFN employs the hidden vector output by the transformer as input to a new fully connected layer for task tuning. BERT+FFN refers to adding two full connection layers based on BERT to implement ZSNER tasks. We randomly select 1000 pieces of training data and segregate them into 100 groups, each comprising ten pieces of data. For BERT and BERT+FFN, we replace words with [MASK] randomly with a a probability of 10% in each data group, calculate the loss value of the predicted [MASK] token, and finally average the loss values of 100 groups to obtain the LM loss. For models based on ELECTRA, we scramble the text order, replace the phrases randomly, and calculate the loss of whether the tokens in the text should be replaced.

Table 6 illustrates that the performance of PLMs on the pre-training task worsens as the parameters change, suggesting that the model tends to forget some of the general knowledge acquired during the pre-training stage while learning new tasks. Additionally, Figure 4 shows that the LM Loss is 6.9x larger than the initial BERT model, and we observe a decline in the F1-score of the model on the ZSNER task with the increase of LM Loss. This suggests that the knowledge acquired by PLMs during the pre-training stage is beneficial for zero-shot tasks.

4 Related Work

4.1 Zero-shot Entity and Relation Extraction

In recent years, zero-shot entity and relation extraction (Ma et al., 2016; Ye et al., 2022; Wang et al., 2021a) has attracted great attention from academia. Fine-tuning PLMs for ZSNER and ZSRE tasks has achieved promising performance. SMXM (Aly et al., 2021) achieves state-of-the-art results in ZS-NER by incorporating entity-type description into

entity encoding. There are other works converting ZSNER to machine reading comprehension framework (Li et al., 2020; Wang et al., 2021b). ZS-BERT (Chen and Li, 2021) learns the independent projection functions to predict relations and obtains a good performance on ZSRE task. However, ZS-BERT can only infer relations and assumes that the ground-truth entity pairs are readily available, which is unrealistic in real scenarios. RelationPrompt (Chia et al., 2022) is the first approach to extract the whole relation triplet under the zero-shot setting by modifying the BART (Lewis et al., 2020) generation decoder to generate relation triplets. Unlike them, DSP converts input sentences into a discriminative language modeling problem, which bridges the gap between pre-training and fine-tuning.

4.2 Prompt-based Learning

Stemming from the GPT models (Radford et al., 2018, 2019), the prompt-based learning has been widely discussed. The core idea of cloze-style prompt methods (Tam et al., 2021) is to transform a classification problem into a cloze-style task with textual templates, and then map label words to the verbalizer. Schick and Schütze (2021) use manually defined templates and verbalizers for prompting text classification tasks. To alleviate manual efforts, Jiang et al. (2020) propose a mining approach for automatically searching for templates. Meanwhile, several approaches have explored the designation of verbalizers. Cui et al. (2022) train prototype vectors as verbalizers by contrastive learning. Hu et al. (2021) expand the label word space of the verbalizer using external knowledge, and refine the verbalizer space with the training data. However, there is no training data to refine the verbalizer under the zero-shot setting, and it is still very difficult to map the label words to the verbalizer. In addition, the cloze-style prompt methods can predict only one token label per template, which is extremely slow for inference in token-level tasks, such as ZSNER. To solve the above problems, we propose DSP to formulate ZSNER and ZSRE tasks into label discrimination tasks without build verbalizers, while all entities of the same type in a sentence are extracted using only one inference.

5 Conclusion

This paper presents a novel Discriminative Soft Prompts method for zero-shot entity and relation extraction. Unlike the cloze-style prompt method that converts a specific task into an MLM problem, we reformulate ZSNER and ZSRC as a discriminative language modeling problem, which takes advantage of the prompt learning to strengthen the transmission of general knowledge without having to construct a verbalizer. Furthermore, we propose a soft prompt co-reference strategy, which significantly improves the inference efficiency of the discriminative prompt method. Experiments on four datasets demonstrate the effectiveness of our model in both ZSNER and ZSRE tasks. Also, the inference speed of the DSP is up to 120 times faster than the cloze-style prompt method on ZSNER.

Limitations

The main limitation of our work is that we can not use a unified model to complete the zero-shot entity and relationship extraction tasks. Specifically, our method trains two models, DSP-ZSNER and DSP-ZSRC, to extract the entities in the text first and then classify the relation of each pair of entities. This method needs to train and store two models, which is troublesome to maintain in practical applications. In addition, although our method has dramatically improved the inference speed of the previous prompt method, the method still affects the reasoning speed of the model. In the follow-up works, we will be committed to solving this problem.

Acknowledgements

We thank all the reviewers for their efforts to make the paper comprehensive and solid. This work is supported in part by the fund of National Key Research and Development Program of China (Grants No. 2021ZD0112905), National Natural Science Foundation of China (Grants No. 62206140, 62076231), and China Postdoctoral Science Foundation (Grants No. 2022M711726).

References

Rami Aly, Andreas Vlachos, and Ryan McDonald. 2021. Leveraging type descriptions for zero-shot named entity recognition and classification. 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 1516–1528, Online. Association for Computational Linguistics.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. *arXiv preprint arXiv:2104.04697*.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. *arXiv preprint arXiv:2203.09101*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. *arXiv preprint arXiv:2203.09770*.

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

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

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

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv: Computation and Language*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

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

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *Transactions of the Association for Computational Linguistics*, 10:607–622.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. 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. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *Learning*.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.

Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases. *Empirical Methods in Natural Language Processing*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021a. Zero-shot information extraction as a unified text-to-triple translation. *arXiv preprint arXiv:2109.11171*.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021b. Learning from language description: Low-shot named entity recognition via decomposed framework. *arXiv preprint arXiv:2109.05357*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. *conference on information and knowledge management*.

Mengzhou Xia, Mikel Artetxe, Jingfei Du, Danqi Chen, and Veselin Stoyanov. 2022. Prompting ELECTRA: Few-shot learning with discriminative pre-trained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11351–11361, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuan Yao, Bowen Dong, Ao Zhang, Zhengyan Zhang, Ruobing Xie, Zhiyuan Liu, Leyu Lin, Maosong Sun, and Jianyong Wang. 2022. Prompt tuning for discriminative pre-trained language models. *arXiv preprint arXiv:2205.11166*.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for entity and relation extraction. *north american chapter of the association for computational linguistics*.

A DLM-Point Method For ZSNER

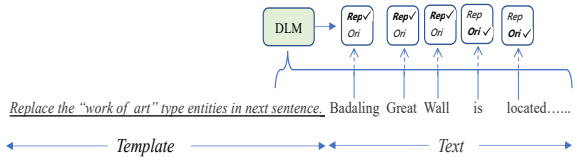

Figure 3: An example of our DLM-Point method for ZSNER.

We try to combine sequence labeling with a discriminative prompt method, and the template is *"Replace the 'work of art' type entities in next sentence."*. The model outputs "replace" of the tokens belonging to this entity type, and then decodes the entity span based on the output. However, this method does not recognize **nested entities**. For instance *Badaling Great Wall* is a "work of art" entity and *Great Wall* is also a "work of art" entity, but model can not recognize it.

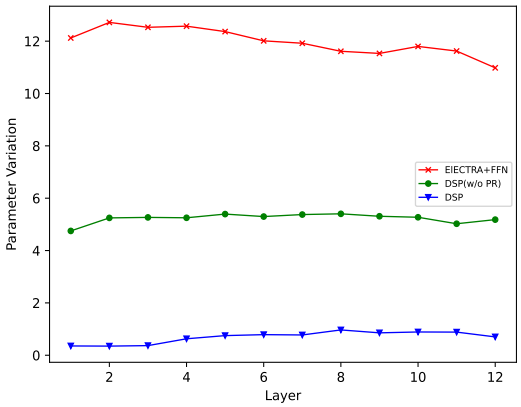

Figure 4: After task-tuning, the changes in each layer's parameters are referenced to the initial model. Longitudinal coordinates represent L^2 values that calculate parameter changes, and horizontal coordinates represent the layers.

B Examples Of the Attention Mask Matrixes

Figure 5 shows examples of the attention mask matrixes of soft prompts co-reference. The token marked with "1" in the matrix participates in the attention calculation, and the token marked with "0" is masked out and does not participate in the calculation.

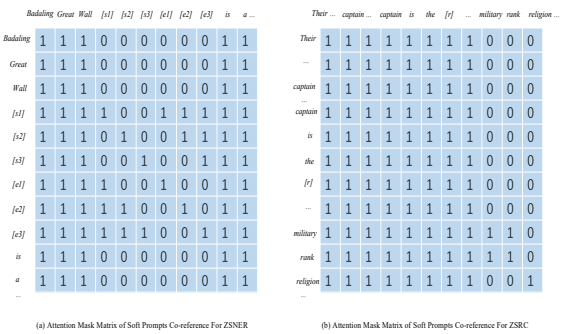

Figure 5: Examples of the attention mask matrixes of soft prompts co-reference.

C Datasets

OntoNotes 5.0 (Pradhan et al., 2013) is a large corpus comprising various genres of text (news, conversational telephone speech, weblogs, newsgroups, broadcast, talk shows) with structural information (syntax and predicate argument structure) and shallow semantics (word sense linked to an ontology and co-reference).

MedMentions (Mohan and Li, 2019) corpus consists of 4,392 papers (titles and abstracts) randomly selected from among papers released on PubMed in 2016 that were in the biomedical field, published in the English language, and had both a title and an abstract.

FewRel (Han et al., 2018) was hand annotated for few-shot relation extraction, and Chia et al. (2022) made it suitable for the zero-shot setting after data splitting into disjoint relation label sets for training, validation and testing.

Wiki-ZSL (Chen and Li, 2021) is constructed through distant supervision using Wikipedia articles and the Wikidata knowledge base.

D Evaluation metrics

We follow the standard evaluation protocol and use F1-score as the evaluation metric. For ZSNER task, the unbalanced number of samples per class necessitates the use of evaluation metrics that focus on per-class averaged scores to properly account for the imbalance. Therefore, we use the macro average F1-score to evaluate our model. We evaluate on ZSRC using the Macro F1-score to be consistent with Chia et al. (2022). In the ZSRE task, the model first identifies entities and then predicts the relation between each pair of entities, resulting in a large number of negative samples. Therefore, we use the micro F1-score which is standard in struc-

Train	PERSON, GPE, ORG, DATE	Biologic Function, Chemical, Healthcare Activity, Anotomical Structure, Finding, Spatial Concept, Intellectual Product, Research Activity, Eukaryote, Population Group, Medical Device
Dev	NORP, MONEY, ORDINAL, PERCENT, EVENT, PRODUCT, LAW	Organization, Injury or Poisoning, Clinical Attribute, Virus, Biomedical Occupation or Discipline
Test	CARDINAL, TIME, LOC, WORK OF ART, FAC, QUANTITY, LANGUAGE	Bacterium, Professional or Occupational Group, Food, Body Substance, Body System

Table 8: Zero-shot class splits and number of occurrences for OntoNotes-ZS and MedMentions-ZS.

tured prediction tasks (Zhong and Chen, 2020) and report the precision (P.) and recall (R.).

E Hyperparameters Choice

We select the learning rate with the best validation accuracy by conducting a grid search from the values of 1e-5, 2e-5, and 5e-5. The batch size is chosen based on the available GPU VRAM. For the weight λ in the regulation loss of Equation 10, we conduct a grid search experiment to determine the optimal value ($\lambda = 0.1$) from a set of values {10, 1, 0.1, 0.01}, based on the performance on the validation set for all models. For all other experiments, we follow the default settings of the ELECTRA (Clark et al., 2020).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section "Limitations"

☑ A2. Did you discuss any potential risks of your work?
Section "Limitations"

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section "1 Introduction"

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section "3.1 Setup"

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section "3.1 Setup"

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section "3.1 Setup"

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section "3.2.2 Results",Section "3.3.2 Results",Section "3.4.2 Results"

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section "3.1 Setup"

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.