Generalizing Few-Shot Named Entity Recognizers to Unseen Domains with Type-Related Features

Zihan Wang¹,², Ziqi Zhao¹, Zhumin Chen¹, Pengjie Ren¹, Maarten de Rijke² and Zhaochun Ren³∗
¹Shandong University, Qingdao, China
²University of Amsterdam, Amsterdam, The Netherlands
³Leiden University, Leiden, The Netherlands
{zihanwang.sdu, ziqizhao.work}@gmail.com, chenzhumin@sdu.edu.cn, jay.ren@outlook.com, m.derijke@uva.nl, z.ren@liacs.leidenuniv.nl

Abstract

Few-shot named entity recognition (NER) has shown remarkable progress in identifying entities in low-resource domains. However, few-shot NER methods still struggle with out-of-domain (OOD) examples due to their reliance on manual labeling for the target domain. To address this limitation, recent studies enable generalization to an unseen target domain with only a few labeled examples using data augmentation techniques. Two important challenges remain: First, augmentation is limited to the training data, resulting in minimal overlap between the generated data and OOD examples. Second, knowledge transfer is implicit and insufficient, severely hindering model generalizability and the integration of knowledge from the source domain. In this paper, we propose a framework, prompt learning with type-related features (PLTR), to address these challenges. To identify useful knowledge in the source domain and enhance knowledge transfer, PLTR automatically extracts entity type-related features (TRFs) based on mutual information criteria. To bridge the gap between training and OOD data, PLTR generates a unique prompt for each unseen example by selecting relevant TRFs. We show that PLTR achieves significant performance improvements on in-domain and cross-domain datasets. The use of PLTR facilitates model adaptation and increases representation similarities between the source and unseen domains.¹

1 Introduction

Named entity recognition (NER) aims to detect named entities in natural languages, such as locations, organizations, and persons, in input text (Zhang et al., 2022; Sang and Meulder, 2003; Yang et al., 2017). This task has gained significant attention from both academia and industry due to its wide range of uses, such as question answering and document parsing, serving as a crucial component in natural language understanding (Nadeau and Sekine, 2007; Ma and Hovy, 2016; Cui and Zhang, 2019; Yamada et al., 2020). The availability of labeled data for NER is limited to specific domains, leading to challenges for generalizing models to new domains (Lee et al., 2022; Cui et al., 2021; Ma et al., 2022).

To overcome this issue, recent research focuses on enabling models to effectively learn from a few labeled examples in new target domains (Lee et al., 2022; Ma et al., 2022; Das et al., 2022; Chen et al., 2022a; Wang et al., 2022, 2023) or on exploring data augmentation techniques, leveraging automatically generated labeled examples to enrich the training data (Zeng et al., 2020). However, these methods still require manual labeling for target domains, limiting their applicability in zero-shot scenarios with diverse domains.

Recently, Yang et al. (2022) have explored a new task, few-shot cross-domain NER, aiming to generalize an entity recognizer to unseen target domains using a small number of labeled in-domain examples. To accomplish this task, a data augmentation technique, named FactMix, has been devised. FactMix generates semireal examples by replacing the original entity or non-entity words in training data. Despite its success, FactMix faces two challenges:

Augmentation is limited to the training data. Since the target domain is not accessible during training, FactMix exclusively augments the training data from the source domain. As a result, there is minimal overlap between the generated examples and the test instances at both the entity and context levels. For instance, only 11.11% of the entity words appear simultaneously in both the generated data (by FactMix) and the target dataset (target domain). At the context level, as demonstrated
To address this problem, we incorporate natural language prompts to guide the model during both training and inference processes, mitigating the gap between the source and unseen domains.

**Knowledge transfer is implicit and insufficient.** Intuitively, better generalization to unseen domains can be accomplished by incorporating knowledge from the source domain (Ben-David et al., 2022). However, in FactMix, the transfer of knowledge from the source domain occurs implicitly at the representation level of pre-trained language models. FactMix is unable to explicitly identify the type-related features (TRFs), i.e., tokens strongly associated with entity types, which play a crucial role in generalization. E.g., as illustrated in Fig. 1(b), the words “established” and “along with” exhibit a close relationship with organization and person entities, respectively, in both domains. This knowledge can greatly assist in recognizing organizations and persons in the target domain.

To tackle this limitation, we introduce mutual information criteria to extract informative TRFs from the source domain. Furthermore, we construct a unique prompt for each unseen instance by selecting relevant TRFs. Intuitively, these generated prompts serve as distinctive signatures, linking unfamiliar examples to the knowledge within the source domain.

**Contributions.** In this paper, we present a framework, named prompt learning with type-related features (PLTR) for few-shot cross-domain NER, to effectively leverage knowledge from the source domain and bridge the gap between training and unseen data. As Fig. 2 shows, PLTR is composed of two main phases: (i) type-related feature extraction, and (ii) prompt generation and incorporation. To identify valuable knowledge in the source domain, PLTR uses mutual information criteria to extract entity type-related features (TRFs). PLTR implements a two-stage framework to mitigate the gap between training and OOD data. Firstly, given a new example, PLTR constructs a unique sequence by selecting relevant TRFs from the source domain. Then, the constructed sequences serve as prompts for performing entity recognition on the unseen data. Finally, a multi-task training strategy is employed to enable parameter sharing between the prompt generation and entity recognition. Similar to FactMix, PLTR is a fully automatic method that does not rely on external data or human interventions. PLTR is able to seamlessly integrate with different few-shot NER methods, including standard fine-tuning and prompt-tuning approaches.

In summary, our contributions are: (i) to the best of our knowledge, ours is the first work to study prompt learning for few-shot cross-domain NER; (ii) we develop a mutual information-based approach to identify important entity type-related features from the source domain; (iii) we design a two-stage scheme that generates and incorporates a prompt that is highly relevant to the source domain for each new example, effectively mitigating the gap between source and unseen domains; and (iv) experimental results show that our proposed PLTR achieves state-of-the-art performance on both in-domain and cross-domain datasets.
2 Related work

Cross-domain NER. The task of cross-domain NER aims to transfer NER models across diverse text styles (Pan et al., 2013; Liu et al., 2021; Chen et al., 2021; Lee et al., 2018; Yang et al., 2017; Jia et al., 2019; Jia and Zhang, 2020; Zheng et al., 2022; Zhang et al., 2022; Hu et al., 2023; Wang et al., 2021). Yang et al. (2017) train NER models jointly in the source and target domains, while Jia et al. (2019) and Jia and Zhang (2020) leverage language models for cross-domain knowledge transfer. Zhang et al. (2022) introduce a modular learning approach that decomposes NER into entity span detection and type classification subtasks. However, these methods still rely on NER annotations or raw data in the target domain.

Few-shot NER and prompt-based learning. Few-shot named entity recognition (NER) is the task of identifying predefined named entities using only a small number of labeled examples (Wiseman and Stratos, 2019; Yang and Katiyar, 2020; Das et al., 2022; Zeng et al., 2020; Ma et al., 2023). Various approaches have been proposed to address this task. For instance, Huang et al. (2021) investigate the effectiveness of self-training methods on external data using distance-based approaches, where the label of the nearest neighbors is copied. Zeng et al. (2020) involves generating counterfactual examples through interventions to augment the original dataset. Additionally, prompt-based learning, which has gained prominence in natural language processing, has also been applied to few-shot NER (Cui et al., 2021; Ma et al., 2022; Lee et al., 2022; Das et al., 2022; Chen et al., 2022b; Li et al., 2022; Dong et al., 2023; Fang et al., 2023). In particular, Das et al. (2022) incorporate contrastive learning techniques with prompts to better capture label dependencies. Furthermore, Ma et al. (2022) develop a template-free approach to prompt NER, employing an entity-oriented objective. Recently, several studies have conducted analyses of the performance of current large language models (LLMs), such as the GPT series (Brown et al., 2020; OpenAI, 2023), in the context of the few-shot NER task (Gutierrez et al., 2022; Han et al., 2023; Sun et al., 2023). Nevertheless, these investigations have revealed a substantial performance gap between recent LLMs and state-of-the-art methods. Consequently, due to their high running costs and underwhelming performance, we do not consider recent LLMs as the basic model of our proposed framework (refer to Sec. 3.2). As mentioned in Sec. 1, previous few-shot NER methods primarily focus on in-domain settings and require manual annotations for the target domain, which poses a challenge for generalizing to OOD examples.

The field of few-shot cross-domain learning is inspired by the rapid learning capability of humans to recognize object categories with limited examples, known as rationale-based learning (Brown et al., 2020; Shen et al., 2021; Chen et al., 2022a; Baxter, 2000; Zhang et al., 2020). In the context of NER, Yang et al. (2022) introduce the few-shot cross-domain setting and propose a two-step rationale-centric data augmentation method, named FactMix, to enhance the model’s generalization ability.

In this paper, we focus on few-shot cross-domain NER. The most closely related work is FactMix (Yang et al., 2022). FactMix faces two challenging problems: (i) augmentation is limited to the training data, and (ii) the transfer of knowledge from the source domain is implicit and insufficient.
In our proposed PLTR, to identify useful knowledge in the source domain, mutual information criteria are designed for automatic type-related feature (TRF) extraction. In addition, PLTR generates a unique prompt for each unseen example based on relevant TRFs, aiming to reduce the gap between the source and unseen domains.

3 Preliminaries

3.1 Task settings

A NER system takes a sentence \( x = x_1, \ldots, x_n \) as input, where \( x \) is a sequence of \( n \) words. It produces a sequence of NER labels \( y = y_1, \ldots, y_n \), where each \( y_i \) belongs to the label set \( \mathcal{Y} \) selected from predefined tags \( \{B_i, I_i, S_i, E_i, O\} \). The labels \( B, I, E, \) and \( S \) indicate the beginning, middle, ending, and single-word entities, respectively. The entity type is denoted by \( t \in \mathcal{T} = \{\text{PER}, \text{LOC}, \text{ORG}, \text{MISC}, \ldots\} \), while \( O \) denotes non-entity tokens. The source dataset and out-of-domain dataset are represented by \( D_{in} \) and \( D_{ood} \), respectively. Following Yang et al. (2022), we consider two settings in our task, the in-domain setting and the out-of-domain (OOD) setting. Specifically, we first train a model \( \mathcal{M}_{in} \) using a small set of labeled instances from \( D_{in} \). Then, for in-domain and OOD settings, we evaluate the performance of \( \mathcal{M}_{in} \) on \( D_{in} \) and \( D_{ood} \), respectively.

3.2 Basic models

Since our proposed PLTR is designed to be model-agnostic, we choose two popular NER methods, namely standard fine-tuning and prompt-tuning respectively, as our basic models. As mentioned in Sec. 2, due to their high costs and inferior performance on the NER task, we do not consider recent large language models (e.g., GPT series) as our basic models.

Standard fine-tuning method. We employ pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) to generate contextualized word embeddings. These embeddings are then input into a linear classifier with a softmax function to predict the probability distribution of entity types. The process involves feeding the input token \( x \) into the feature encoder PLM to obtain the corresponding contextualized word embeddings \( h \):

\[
h = \text{PLM}(x),
\]

where \( h \) represents the sequence of contextualized word embeddings derived from the pre-trained language models. To recognize entities, we optimize the cross-entropy loss \( L_{NER} \) as:

\[
L_{NER} = -\sum_{c=1}^{N} y_{o,c} \log(p_{o,c}),
\]

where \( N \) denotes the number of classes, \( y \) is a binary indicator (0 or 1) indicating whether the gold label \( c \) is the correct prediction for observation \( o \), and \( p \) is the predicted probability of \( c \) for \( o \).

Prompt-tuning method. The prompt-tuning method for NER tasks involves the use of mask-and-infill techniques based on human-defined templates to generate label words. We adopt the recent EntLM model proposed by Ma et al. (2022) as our benchmark for this method. First, a label word set \( \mathcal{Y}_i \) is constructed through label word engineering, which is connected to the label set using a mapping function \( \mathcal{M} : \mathcal{Y} \rightarrow \mathcal{Y}_i \). Next, entity tokens at entity positions are replaced with the corresponding label word \( \mathcal{M}(y_i) \). The resulting modified input is then denoted as \( x_{\text{Ent}}^i = \{x_1, \ldots, \mathcal{M}(y_i), \ldots, x_n\} \). The language model is trained by maximizing the probability \( P(x_{\text{Ent}}^i \mid x) \). The loss function for generating the prompt and performing NER is formulated as:

\[
\mathcal{L}_{NER} = -\sum_{i=1}^{N} \log P(x_i = x_{\text{Ent}}^i \mid x),
\]

where \( N \) represents the number of classes. The initial parameters of the predictive model are obtained from PLMs.

4 Method

In this section, we present the two primary phases of the proposed PLTR method, as depicted in Fig. 2: (i) type-related feature extraction (see Sec. 4.1), and (ii) prompt generation and incorporation (see Sec. 4.2).

4.1 Type-related feature extraction

As mentioned in Sec. 1, type-related features (TRFs), which are tokens strongly associated with entity types, play a crucial role in the few-shot cross-domain NER task. To extract these features, we propose a mutual information based method for identifying TRFs from the source domain. Here, we define \( S_i \) as a set that contains all sentences from the source domain where entities of the \( i \)-th type appear, and \( S \setminus S_i \) as a set that contains
sentences without entities of the $i$-th type. In our method, we consider a binary variable that indicates examples (texts) from $S_i$ as 1, and examples from $S_i \setminus S_i$ as 0. To find tokens closely related to $S_i$, we first calculate the mutual information between all tokens and this binary variable, and then select the top $l$ tokens with the highest mutual information scores. However, the mutual information criteria may favor tokens that are highly associated with $S_i \setminus S_i$ rather than with $S_i$. Thus, we introduce a filtering condition as follows:

$$\frac{C_{S_i \setminus S_i}(w_m)}{C_{S_i}(w_m)} \leq \rho, \quad C_{S_i}(w_m) > 0, \quad (4)$$

where $C_{S_i}(w_m)$ represents the count of the $m$-gram $w_m = x_p, \ldots, x_{p+m-1}$ in $S_i$. $C_{S_i \setminus S_i}(w_m)$ represents the count of this $m$-gram $w_m$ in all source domains except for $S_i$, and $\rho$ is a $m$-gram frequency ratio hyperparameter. By applying this criterion, we ensure that $w_m$ is considered part of the TRF set of $S_i$ only if its frequency in $S_i$ is significantly higher than its frequency in other entity types ($S_i \setminus S_i$). Since the number of examples in $S_i$ is much smaller than the number of examples in $S_i \setminus S_i$, we choose $\rho \geq 1$ but avoid setting it to a large value. This allows for the inclusion of features that are associated with $S_i$ while also being related to other entity types in the TRF set of $S_i$. In our experiments, we set $\rho = 3$ and only consider 1-gram texts for simplicity.

Note that the type-related feature extraction module is designed to be efficient with a computational complexity of $O(|D_m| \cdot l_{avg} \cdot |T|)$, where $|D_m|$, $l_{avg}$, and $T$ represent the number of sentences in the training dataset, the average sentence length, and the entity type set, respectively. This module is able to compute the mutual information criteria in Eq. 4 for all entity types in $T$ and each token by traversing the tokens in every training sentence just once.

### 4.2 Prompt generation and incorporation

To connect unseen examples with the knowledge within the source domain, we generate and incorporate a unique prompt for each input instance. This process involves a two-stage mechanism: first, relevant TRFs are selected to form prompts, and then these prompts are input into the PLM-based basic model for entity label inference.

**Automatic type-related feature selection.** Given an input sentence $x$ and the extracted TRF set $R$, we formulate the selection of relevant TRFs as a cloze-style task for our PLM-based basic model $M_b$ (refer to Sec. 3.2). Specifically, we define the following prompt template function $f(\cdot)$ with $K$ [MASK] tokens:

$$f(x) = \text{“}[SEP]\text{type-related features:[MASK]}\ldots[\text{MASK}]”.$$  

By inputting $f(x)$ into $M_b$, we compute the hidden vector $h_{[\text{MASK}]}$ of [MASK]. Given a token $r \in R$, we compute the probability that token $r$ can fill the masked position:

$$p([\text{MASK}] = r | f(x)) = \frac{\exp(r \cdot h_{\text{[MASK]}})}{\sum_{r \in R} \exp(r \cdot h_{\text{[MASK]}})}, \quad (6)$$

where $r$ is the embedding of the token $r$ in the PLM $M_b$. For each [MASK], we select the token with the highest probability as the relevant TRF for $x$, while discarding any repeating TRFs. For example, as illustrated in Fig. 2, for the sentence “Bolton’s spokesperson told CBS News.”, the most relevant TRFs include “Spokesmen”, “News” and “Corp”.

To train $M_b$ for TRF selection, we define the loss function $L_{gen}$ as follows:

$$L_{gen} = -\frac{1}{|D_m|} \sum_{x \in D_m} \sum_{i=1}^{K} \log p([\text{MASK}]_i = \phi(x,i) | f(x)), \quad (7)$$

where $\phi(x,i)$ denotes the label for the $i$-th [MASK] token in $x$. To obtain $\phi(x)$, we compute the Euclidean distance between the PLM-based embeddings of each $r \in R$ and each token in $x$, selecting the top-$K$ features. Note that our designed automatic selection process effectively filters out irrelevant TRFs for the given input sentence, substantially reducing human interventions in TRF extraction (refer to Sec. 7).

**Prompt incorporation.** To incorporate the entity type information into prompts, we generate a unique prompt given the selected relevant TRFs $R'(x) \subseteq R$ for input $x$. This is achieved using the following prompt template function $f'(x)$:

$$f'(x) = \text{“}[SEP]t_1:R'(x,t_1)[SEP]\ldots[SEP]t_{|T|}:R'(x,t_{|T|})”.$$  

where $t_i \in T$ is the entity type name (e.g., PER or ORG). Given sentence $x$, $R'(x,t_i) \subseteq R'(x)$ represents selected TRFs related to entity type $t_i$. Note that, if $R'(x,t_i) = \emptyset$, the entity type name, and relevant TRFs $R'(x,t_i)$ are excluded from $f'(x)$. For example, as depicted in
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Entity types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoNLL2003</td>
<td>14,987</td>
<td>3,466</td>
<td>3,684</td>
<td>4</td>
</tr>
<tr>
<td>OntoNotes</td>
<td>59,924</td>
<td>8,528</td>
<td>8,262</td>
<td>18</td>
</tr>
<tr>
<td>TechNews</td>
<td>-</td>
<td>-</td>
<td>2,000</td>
<td>4</td>
</tr>
<tr>
<td>AI</td>
<td>-</td>
<td>-</td>
<td>431</td>
<td>14</td>
</tr>
<tr>
<td>Literature</td>
<td>-</td>
<td>-</td>
<td>416</td>
<td>12</td>
</tr>
<tr>
<td>Music</td>
<td>-</td>
<td>-</td>
<td>456</td>
<td>13</td>
</tr>
<tr>
<td>Politics</td>
<td>-</td>
<td>-</td>
<td>651</td>
<td>9</td>
</tr>
<tr>
<td>Science</td>
<td>-</td>
<td>-</td>
<td>543</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the datasets used.

Fig. 2, the unique prompt \( f'(x) \) corresponding to \( x = \text{“Bolton’s spokesperson told CBS News.”} \) can be represented as follows:

\[
f'(x) = \text{“Bolton’s spokesperson told CBS News. [SEP]PER:Spokesmen[SEP]ORG:News, Corp”}. \tag{9}
\]

Then, we input \( f'(x) \) into \( M_b \) to recognize entities in the given sentence \( x \).

4.3 Joint training

To enable parameter sharing between prompt generation and incorporation, we train our model using a multi-task framework. The overall loss function is defined as follows:

\[
L = \alpha \cdot L'_{NER} + (1 - \alpha) \cdot L_{gen}, \tag{10}
\]

where \( L'_{NER} \) denotes the normalized loss function for the NER task loss \( L_{NER} \) (refer to Sec. 3.2). \( \alpha \) is the weight assigned to \( L'_{NER} \) with prompts as inputs. The weight \( 1 - \alpha \) is assigned to the loss function \( L_{gen} \) for type-related feature selection. In our experiments, we optimize the overall loss function using AdamW (Loshchilov and Hutter, 2019). Sec. A.1 gives the detailed training algorithm of PLTR.

5 Experiments

We aim to answer the following research questions: (RQ1) Does PLTR outperform state-of-the-art fine-tuning methods on the few-shot cross-domain NER task? (Sec. 6.1) (RQ2) Can PLTR be applied to prompt-tuning NER methods? (Sec. 6.2) Micro F1 is adopted as the evaluation metric for all settings.

5.1 Datasets

Detailed statistics of both in-domain and out-of-domain datasets are shown in Table 1.

In-domain dataset. We conduct in-domain experiments on the CoNLL2003 dataset (Sang and Meulder, 2003). It consists of text in a style similar to Reuters News and encompasses entity types such as person, location, and organization. Additionally, to examine whether PLTR is extensible to different source domains and entity types, we evaluate PLTR using training data from OntoNotes (Weischedel et al., 2013) (refer to Sec. A.3). OntoNotes is an English dataset consisting of text from a wide range of domains and 18 types of named entities, such as Person, Event, and Date.

Out-of-domain datasets. We utilize the OOD dataset collected by Liu et al. (2021), which includes new domains such as AI, Literature, Music, Politics, and Science. The vocabulary overlaps between these domains are generally small, indicating the diversity of the out-of-domain datasets (Liu et al., 2021). Since the model trained on the source domain dataset (CoNLL2003) can only predict person, location, organization, and miscellaneous entities, we assign the label \( O \) to all unseen labels in the OOD datasets.

5.2 Experimental settings and baselines

We compare PLTR with recent baselines in the following two experimental settings:

Fine-tuning. Following Yang et al. (2022), we employ the standard fine-tuning method (Ori) based on two pre-trained models with different parameter sizes: BERT-base, BERT-large, RoBERT-base, and RoBERT-large. All backbone models are implemented using the transformer package provided by Huggingface.\(^2\) For fine-tuning the NER models in a few-shot setting, we randomly select 100 instances per label from the original dataset (CoNLL2003) to ensure model convergence. The reported performance of the models is an average across five training runs.

Prompt-tuning. Similar to Yang et al. (2022), we adopt the EntLM model proposed by Ma et al. (2022) as the benchmark for prompt-tuning. The EntLM model is built on the BERT-base or BERT-large architectures. We conduct prompt-based experiments using a 5-shot training strategy (Ma et al., 2022). Additionally, we select two representative datasets, TechNews and Science, for the OOD test based on the highest and lowest word overlap with the original training domain, respectively.

Additionally, we include a recent data augmentation method CF (Zeng et al., 2020) and the state-

\(^2\)https://huggingface.co/models
In-domain Fine-tuning Results

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Ori</th>
<th>CF</th>
<th>FactMix</th>
<th>PLTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-base-cased</td>
<td>54.03</td>
<td>77.71</td>
<td>80.10</td>
<td>82.05*</td>
</tr>
<tr>
<td>BERT-large-cased</td>
<td>65.38</td>
<td>81.11</td>
<td>83.04</td>
<td>83.75*</td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td>48.53</td>
<td>82.74</td>
<td>85.33</td>
<td>86.40*</td>
</tr>
<tr>
<td>RoBERTa-large</td>
<td>65.70</td>
<td>85.20</td>
<td>86.91</td>
<td>88.03*</td>
</tr>
</tbody>
</table>

Table 2: In-domain fine-tuning results (Micro F1) on CoNLL2003. * indicates a statistically significant difference (t-test, p < 0.05) when compared to FactMix.
are removed during test-time inference. In this case, the F1 scores across all datasets and settings suffer a significant drop compared to our proposed PLTR. This demonstrates the crucial role of incorporating prompts during both the training and inference processes. In the “RDW” and “REW” variants, prompts are constructed using randomly selected words from the source domain and the given example, respectively. The performance of both the “RDW” and “REW” model variants consistently falls short of PLTR, indicating that PLTR effectively identifies important knowledge from the source domain and establishes connections between unseen examples and the knowledge within the source domain.

Additionally, to explore the efficacy of type-related feature selection (refer to Sec. 4.2), we conducted an evaluation of PLTR (BERT-base) using various frequency ratios $\rho$ (in Eq. 4). The results are presented in Table 7. As the value of $\rho$ increases, TRFs extracted using Eq. 4 become less closely associated with the specified entity type but become more prevalent in other types. When the value of $\rho$ is raised from 3 to 9, we observed only a slight decrease in the F1 scores of PLTR. When the value of $\rho$ is raised to 20, the F1 score of PLTR drops, but still surpasses the state-of-the-art baseline FactMix. These results indicate that PLTR effectively identifies relevant TRFs for OOD examples, considerably mitigating human interventions in the feature extraction process.

**The influence of training samples.** To examine the impact of the number of training samples, we compare the performance of PLTR and FactMix on few-shot cross-domain NER using 100, 300, and 500 training samples per entity type. Fig. 3 displays the results based on the BERT-base-cased model. PLTR exhibits the largest improvements over FactMix when the dataset comprises only 100 training instances per entity type, as opposed to the 300 and 500 training instances scenarios. Furthermore, PLTR consistently outperforms the prior state-of-the-art approach, FactMix, across all experimental settings with varying numbers of training examples, demonstrating its superiority.

**Analysis of sentence similarities.** In our analysis of sentence similarities, we investigate the impact of PLTR on the representation similarities between the source and target domains. We compute the
Table 5: OOD prompt-tuning results (Micro F1) on TechNews and Science. * indicates a statistically significant difference (t-test, p < 0.05) when compared to FactMix.

Table 6: Ablation studies on TechNews and Science.

Table 7: Influence of frequency ratio ($\rho$) on AI and Science (BERT-base, fine-tuning).

Table 8: Error analysis on sentence lengths in test sets (BERT-base, fine-tuning).

8 Conclusions

In this paper, we establish a new state-of-the-art framework, PLTR, for few-shot cross-domain NER. To capture useful knowledge from the source domain, PLTR employs mutual information criteria to extract type-related features. PLTR automatically selects pertinent features and generates a unique prompt for each unseen example, bridging the gap between domains. Experimental results show that PLTR not only effectively generalizes standard fine-tuning methods to unseen domains, but also demonstrates promising performance when incorporated with prompt-tuning-based approaches. Additionally, PLTR substantially narrows the disparity between in-domain examples and OOD instances, enhancing the similarities of their sentence representations.
Limitations

While PLTR achieves a new state-of-the-art performance, it has several limitations. First, the number of type-related features for prompt construction needs to be manually preset. Second, PLTR relies on identifying TRFs, which are tokens strongly associated with entity types. Extracting and incorporating more complex features, such as phrases, represents a promising direction for future research. In the future, we also plan to incorporate PLTR with different kinds of pre-trained language models, such as autoregressive language models.

Ethics statement

The paper presents a prompt-based method for recognizing named entities in unseen domains with limited labeled in-domain examples. However, the constructed prompts and model-predicted results still have a considerable amount of misinformation. Besides, the reliance on black-box pre-trained language models raises concerns. Hence, caution and further research are required prior to deploying this method in real-world applications.

Acknowledgement

This work was supported by the National Key R&D Program of China (2020YFB1406704, 2022YFC3303004), the Natural Science Foundation of China (62272274, 61972234, 62072279, 62102234, 62202271), the Natural Science Foundation of Shandong Province (ZR2021QF129, ZR2022QF004), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), the Fundamental Research Funds of Shandong University, the China Scholarship Council under grant nr. 202206220085, the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organization for Scientific Research, https://hybrid-intelligence-centre.nl, and project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO).

References


Guanting Dong, Zechen Wang, Jinxu Zhao, Gang Zhao, Daichi Guo, Dayuan Fu, Tingfeng Hui, Chen Zeng, Keqing He, Xuefeng Li, Liwen Wang, Xinyue Cui, and Weiran Xu. 2023. A multi-task semantic decomposition framework with task-specific pre-training for few-shot NER. *CoRR*, abs/2308.14533.


Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and


Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In EMNLP, pages 10408–10423.


Ruotian Ma, Xin Zhou, Tao Gui, Yidong Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER. In NAACL, pages 5721–5732.


Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin,
Algorithm 1 Training Algorithm for PLTR.

Require: The source dataset $D_{in}$; the basic model $M_b$ with parameters $\Theta$; the number of selected type-related features $K$; the loss weight $\alpha$; the number of epochs $epoch$.

Ensure: The extracted type-related features $R$ and the trained basic model $M_b'$.

1: Establish the basic model $M_b$;
2: Initialize model parameters $\Theta$;
3: Extract type-related features $R$ for all entity types from the source dataset $D_{in}$ (Eq. 4);
4: while $i \leq epoch$ do
5: for Sample a batch $X \subseteq D_{in}$ do
6: for all sentences $x \in X$ do
7: Select relevant TRFs $R'(x)$ for input $x$ (Eq. 5 and 6);
8: Transform $x$ into the prompt template $f'(x)$ (Eq. 8);
9: Input $f'(x)$ into $M_b$ for prediction;
10: end for
11: end for
12: Update $\Theta$ by optimizing $L$ (Eq. 10);
13: end while
14: end for

$\theta'$ for each input $x$, and these prompts are then fed into $M_b$ for entity recognition (lines 8-9). Finally, we iteratively refine the parameters $\Theta$ by jointly optimizing two loss functions: the NER task loss function $L'_{\text{NER}}$ and the TRF selection loss function $L'_{\text{gen}}$ (line 11). Note that, during inference, PLTR generates a unique prompt for each sentence within the unseen target domain using extracted TRFs $R$. In this way, knowledge from the source domain is explicitly integrated into both the training and inference phases.

A.2 Influence of the number of selected type-related features

We evaluate PLTR based on BERT-base in fine-tuning setting, with the number of selected relevant type-related features $K$ varying from 10 to 60. The results are shown in Fig. 5. Our observations indicate that as the number of type-related features increases, the performance (F1 score) of PLTR initially improves because the model incorporated with more features is able to encode more useful knowledge from the source domain. But notice that the performance drops when the number of type-related features is too large. In our experiments, we set the number of type-related features to 40 on all
Table 9: Influence of source domains (BERT-base). In TechNews, there are no annotations for "EVENT" entities.

<table>
<thead>
<tr>
<th>Source: CoNLL2003</th>
<th>Source: OntoNotes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setting</strong></td>
<td><strong>Model</strong></td>
</tr>
<tr>
<td>OOD Fine-tuning</td>
<td>FactMix</td>
</tr>
<tr>
<td></td>
<td>PLTR</td>
</tr>
<tr>
<td>OOD Prompt-tuning</td>
<td>FactMix</td>
</tr>
<tr>
<td></td>
<td>PLTR</td>
</tr>
</tbody>
</table>

Figure 5: Influence of the number of selected relevant type-related features ($K$) on AI (BERT-base, fine-tuning).

A.3 Influence of source domains

We explore the performance of our proposed PLTR when trained on data from different source domains, i.e., CoNLL2003 and OntoNotes. Results in both the fine-tuning and prompt-tuning settings are shown in Table 9. Our observations indicate that our proposed PLTR consistently outperforms FactMix when trained on different domains. For instance, PLTR achieves an average improvement of 5.42% and 4.22% over FactMix for "LOC" entities when using CoNLL2003 and OntoNotes as source datasets, respectively. This highlights PLTR’s capacity to extend to various source domains and entity types.