Know-Adapter: Towards Knowledge-Aware Parameter-Efficient Fine-Tuning for Few-shot Named Entity Recognition

Binling Nie*, **Yiming Shao, Yigang Wang**
School of Digital Media and Arts, Hangzhou Dianzi University
binlingnie@hdu.edu.cn

Abstract

Parameter-Efficient Fine-Tuning (PEFT) is a promising approach to mitigate the challenges about the model adaptation of pretrained language models (PLMs) for the named entity recognition (NER) task. Recent studies have highlighted the improvements that can be made to the quality of information retrieved from PLMs by adding explicit knowledge from external source like KGs to otherwise naive PEFTs. In this paper, we propose a novel knowledgeable adapter, Know-adapter, to incorporate structure and semantic knowledge of knowledge graphs into PLMs for few-shot NER. First, we construct a related KG entity type sequence for each sentence using a knowledge retriever. However, the type system of a domain-specific NER task is typically independent of that of current KGs and thus exhibits heterogeneity issue inevitably, which makes matching between the original NER and KG types (e.g. Person in NER potentially matches President in KBs) less likely, or introduces unintended noises. Thus, then we design a unified taxonomy based on KG ontology for KG entity types and NER labels. This taxonomy is used to build a learnable shared representation module, which provides shared representations for both KG entity type sequences and NER labels. Based on these shared representations, our Know-adapter introduces high semantic relevance knowledge and structure knowledge from KGs as inductive bias to guide the updating process of the adapter. Additionally, the shared representations guide the learnable representation module to reduce noise in the unsupervised expansion of label words. Extensive experiments on multiple NER datasets show the superiority of Know-Adapter over other state-of-the-art methods in both full-resource and low-resource settings.

Keywords: Knowledge Graph, PEFT, few-shot NER

1. Introduction

Named entity recognition (NER) is a fundamental task in natural language processing (NLP). NER is often formulated as a sequence labeling task, where the goal is to assign a label to each entity in the input sequence. These labels are based on predefined categories such as location, organization, and person. Current state-of-the-art NER methods use pre-trained language models (PLMs) equipped with several NER paradigms, including label-specific classifiers (LCs) (Cui and Zhang, 2019), machine reading comprehension (MRC) (Yu et al., 2020), and unified generative models (Bart-NER) (Yan et al., 2021). However, these models are highly correlated with visible categories and often remember entities explicitly (Agarwal et al., 2021). This is because the output layers of these models must have a consistent set of labels between training and testing. Therefore, these models must be rebuilt from scratch to adapt to the target domain with new entity classes, making few-shot NER a challenging but practical research problem.

Few-shot NER is the task of training a NER model on a small number of labeled examples of the target domain. Recently, PEFTs (Wang et al., 2022b; Cui et al., 2021; Chen et al., 2022b; Ma et al., 2022; Shen et al., 2023) for few-shot NER have emerged as a surprisingly effective way to adapt PLMs to few-shot NER settings. A popular and straightforward PEFT approach is prompt learning, which aims to reduce the gap between pre-training and fine-tuning by reformulating downstream tasks into a pre-training style. The performance of these prompt models, such as PromptNER (Shen et al., 2023) and EntLM (Ma et al., 2022), in few- and zero-shot settings depends heavily on the provided prompts, which are contextual information fragments of the NER task. However, in many cases, the engineering design of prompts is naive and simplistic, providing PLMs with too little context to provide accurate answers. Furthermore, prompt design is often unsystematic, lacking a principled approach to combining cues. Recent researches () have highlighted that the quality of the information retrieved from PLMs can be improved by providing system designed prompts or adapters that contain explicit knowledge from external sources such as knowledge graphs (KGs).

Therefore, we propose Know Adapter, a novel KG-empowered PEFT method for few-shot NER. For each sentence, we construct a relevant KG type sequence and introduce it to guide the update process, injecting KG-based knowledge as inductive bias. However, there are heterogeneity issues between KB and NER type systems. Specifically, KB-type systems typically have thousands of types, while NER-type systems usually have no more than a hundred or even less, indicating a gap

* Corresponding Author

LREC-COLING 2024, pages 9777–9786

and leading to different design philosophies. To avoid introducing too much noise, we merge KG entity types and NER labels into a unified taxonomy based on KG ontology. We then build a learnable shared representation module that bridges the projection between ontology vocabulary and label space. KG ontology node words are not simple synonyms; they cover different granularities and perspectives, making them more comprehensive and unbiased than NER label class names. Additionally, the KG ontology-based shared representation module provides a shared representation for sentence-related KG type sequences and NER tag classes. These shared representations learn high semantic relevance knowledge, bridge the structural gap between context and KG, and refine the representation module itself by retaining high-quality words.

To verify the effectiveness of our approach, we conducted extensive experiments on a variety of few-shot NER tasks using PLMs. The results demonstrate that our proposed Know-Adapter can effectively address the absence of structural and semantic knowledge and excessive cost issues, significantly enhancing the performance.

Our contributions are summarized as follows:

- We propose a novel knowledgeable adapter to incorporate structure knowledge of entity type taxonomy in existing knowledge graph into PLMs for few-shot NER.
- We present the Know-adapter framework that can be flexibly combined with mainstream PLMs.
- We conduct extensive experiments to show the effectiveness of Know-adapter in both full-resource and low-resource scenarios.

2. Related Works

Few-shot NER. One important research direction in few-shot NER is the use of prototype-based methods. These methods, which incorporate meta-learning, have gained popularity as few-shot learning approaches in the NER field. However, most existing approaches (Ding et al., 2021; Henderson and Vulić, 2021; Yang and Katiyar, 2020; Ling et al., 2023) rely on the nearest-neighbor criterion to assign entity types, based on similar patterns between the source and target domains. These approaches are unable to fully leverage the capabilities of PLMs and may not perform well on cross-domain instances.

There is another line of work using PEFTs, mostly prompt tuning, to solve few-shot NER (Cui et al., 2021; Chen et al., 2022b; Ma et al., 2022; Shen et al., 2023). Especially, PromptNER (Shen et al., 2023) unifies entity locating and entity typing into prompt learning, and designs a dual-slot multi-prompt template with the position slot and type slot to prompt locating and typing respectively, which achieves state-of-the-art performance. However, both prototype-based methods and prompt learning methods don't take rich structure type information of knowledge graph ontology into consideration.

Knowledge Enhanced PLMs. To further inject structure knowledge of knowledge graph into PLMs, recent knowledge enhanced PLMs have been proposed to incorporate structure knowledge in the pre-training stage (Hongbin et al., 2022; Chen et al., 2022a; Ouyang et al., 2021; Zhang et al., 2019; Sun et al., 2020; Wang et al., 2022a). Additionaly, KP-PLM (Wang et al., 2022a) proposes a novel knowledge prompting paradigm to enhance PLMs, which learns factual knowledge from prompts. Inspired by this work, we introduce a novel knowledge enhanced adapter tuning method to leverage rich information of knoweldge graph and fully exploiting the potential of PLMs.

3. Preliminaries

NER Task. Followed by (Yan et al., 2021), we formulate the NER task as a generative framework to maintain the consistency of architecture and enable the model to handle different entity types. Given an input sentence of n tokens $X = [x_1, x_2, ..., x_n]$, the target sequence is $Y = [s_{11}, e_{11}, ..., s_{1j}, e_{1j}, t_1, ..., s_{i1}, e_{i1}, ..., s_{ik}, e_{ik}, t_i]$, where s, e are the start and end index of a span, each entity is represented as $[s_{i1}, e_{i1}, ..., s_{ij}, e_{ij}, t_i]$, where t_i is the entity tag index. We use $G = [g_1, ..., g_l]$ to denote the entity tag tokens (such as "Person", "Location", etc.), where l is the number of entity tags. We make $t_i \in (n, n + l]$, the n shift is to make sure t_i is not confusing with pointer indexes (pointer indexes will be in range $[1, n]$). The input and output sequence starts and ends with special tokens $< s >$ and $< /s >$. They should also be generated in Y, but we ignore them in equations for simplicity.

Few-shot NER Task. Traditional NER methods are typically trained in standard supervised learning settings, which require many labeled examples for each entity category. However, in real-world applications, only a few labeled examples are available for each entity category due to the high cost of annotation. This issue yields a challenging task of few-shot NER, where only a small number of labeled examples are available for each entity category. In the training phase of few-shot NER task, a pair-wise dataset $\mathcal{D}^L = \{(X_i, Y_i)\}_{i=1}^N$ are used, where N is the number of examples and is is a

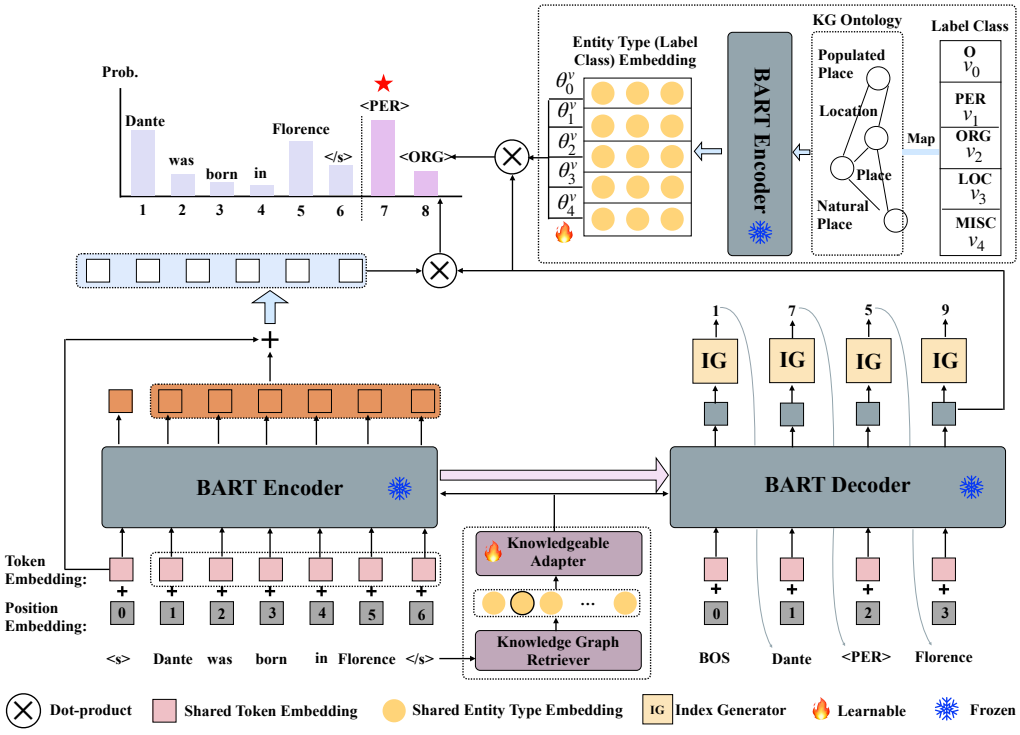


Figure 1: Overview of our proposed Know-Adapter module.

relatively small number compared to standard supervised learning settings. This means only a small amount of training examples are available. In this paper, we focus on the K-shot setting following (Yang and Katiyar, 2020) that only a small number $N = K$ of labeled examples are available for each entity type.

4. Methodology

In this work, we propose Know-Adapter, which uses a novel KG-empowered PEFT paradigm to resolve the few-shot NER task, shown as in Figure 1. Know Adapter follows the general Seq2Seq generation framework and uses the pointer mechanism to generate entity index sequences. To better utilize KG-based knowledge, it builds a shared representation of KG entity types and NER label classes between the two modules:(1) Knowledge adapter module: For a given input sentence X, a knowledge graph retriever is constructed to search for its corresponding KG entity type sequence, which is then provided to the adapter for guidance. (2) Knowledge-enhanced learnable shared representation module: To avoid introducing too much knowledge noise due to the heterogeneity between KG and NER type systems, NER label classes are mapped to KG entity type nodes of KG ontology. A learnable shared encoder is then constructed and refined to obtain shared representations of KG entity types and NER label classes.

4.1. Generative Framework

Since we formulate the NER task in a generative way, we use the Seq2Seq framework with the pointer mechanism to tackle this task. Our model consists of two components:

(1) **Encoder** reads the input sentence X and produces a hidden state representation $\mathbf{H}^e$:

$$\mathbf{H}^e = Encoder(X) \tag{1}$$

where $\mathbf{H}^e \in \mathcal{R}^{n \times d}$, and d is the hidden dimension. (2) **Decoder** gets the index probability distribution for each step $P_t = P(y_t|X, Y_{<t})$. However, since $Y_{<t}$ contains the pointer and tag index, it cannot be directly inputted to the Decoder. We use the Index2Token conversion to convert indexes into tokens

$$\hat{y_t} = \begin{cases} X_{y_t}, & if y_t \leq n \\ G_{y_t - n}, & if y_t > n \end{cases} \tag{2}$$

After converting each y_t this way, we can get the last hidden state $\mathbf{h}_t^d$ with $\hat{Y}_{<t} = [\hat{y_1}, ..., \hat{y_{t-1}}]$ as follows

$$\mathbf{h}_t^d = Decoder(\mathbf{H}^e; \hat{Y}_{<t}) \tag{3}$$

Then, we can use the following equations to compute the probability distribution p_t of token y_t:

$$\begin{aligned} \mathbf{E}_{seq} &= TokenEmbed(X) \\ \hat{\mathbf{H}}^e &= MLP(\mathbf{H}^e) \\ \tilde{\mathbf{H}}^e &= \alpha \cdot \hat{\mathbf{H}}^e + (1 - \alpha) \cdot \mathbf{E}_{seq} \\ p_{seq} &= \tilde{\mathbf{H}}^e \bigotimes \mathbf{h}_t^d \\ p_t &= Softmax([p_{seq}; p_{tag}]) \end{aligned} \tag{4}$$

where $\mathbf{E}^{seq}$, $\tilde{\mathbf{H}}^{e} \in \mathcal{R}^{n \times d}$; $\alpha \in \mathcal{R}$ is a hyperparameter; p_{seq} and p_{tag} refer to the predicted logits on index of entity span and entity categories respectively; $p_t \in \mathcal{R}^{n+m}$ is the predicted probability distribution of y_t on all candidate indexes; [:;:] denotes concatenation in the first dimension. In particular, the details of p_{tag} are in the following subsection.

4.2. Knowledgable Adapter Module

We present a strategy for tuning a large text model with knowledge graph on few-shot NER. Our strategy has two key properties: (i) It finds the best adapter that will make a pretrained masked language model predict the desired answer (shared embeddings) for a training example, and (ii) it adds only a small number of additional parameters. In other words, we want to find parameters $\theta = \{\theta_A, \theta_V\}$ for knowledgable adapter and shared embeddings respectively.

To achieve these desired characteristics, we propose a novel knowledgable adapter module that incorporates knowledge graph embeddings and a small number of additional parameters into the existing PLMs. These parameters are then trained on a few-shot NER task. We specifically apply this knowledgeable adapter-based tuning approach to text Transformers (Vaswani et al., 2017). In the Transformer, each layer consists of two main sublayers: an attention layer and a feed-forward layer. After each of these sub-layers, there is a projection that maps the feature size back to the input size of the layer. Additionally, a skip-connection is applied across each sub-layer. The output of each sub-layer is then passed through layer normalization. To incorporate knowledge graph embeddings, the knowledgeable adapter approach inserts small modules called adapters after each of these sublayers.

Knowledge Graph Retriever. Entities described by KG-based knowledge inject explicit information about the NEs in the sentence (e.g. using a description "location : country, mountain, hill ...", a connection between the mention "Buck mountain" and the entity type "Location" can be made without having seen the annotated example in training). This provides the flexibility to adapt for unseen entities or entities with variable surface forms. Typically it can automatically map novel entity types to the features, and adaptively recognize complex entities (e.g. movie titles) consisted of complex noun or word phrases on-demand. Overall, entities are easy to obtain from open source like Wikidata, which is a free and entity centric knowledge base. We integrate dynamic entity information into the adapter by utilizing entity retriever and encoder.

By leveraging knowledge from a KG, we can gather explicit information about NEs mentioned in a sentence. For example, if we have a description that states "Buck Mountain is a location in a country", we can establish a connection between the mention "Buck Mountain" and the entity type "Location" even without prior exposure to annotated examples during training. This approach allows us to handle new or variable entity forms more effectively. Additionally, our model is capable of automatically mapping new entity types to their respective features and adaptively recognizing complex entities, such as movie titles that consist of intricate noun or word phrases, as needed. Obtaining entity information is relatively straightforward as we can obtain it from open source like Wikidata. To incorporate dynamic entity information into our adapter, we utilize an knowledge graph retriever to search the concrete entity type sequence.

For a mention in the sentence, we design a retriever to find its closest match in Wikidata by using the following forms in order: i) original form; ii) lemma form by Spacy (Honnibal and Montani, 2017); iii) base word (last word). The retriever yields a sparse encoding $\mathbb{e}_s \in \mathbb{N}^{N \times k}$, where $\mathbb{e}_s = x_1, ..., x_N$ and $x_i \in (0, 1)_k$ is a binary vector of length k (k is the number of NEs in $\mathbb{E}$ in BIO format). More specifically, if our sequence of tokens is $\hat{s} = \{... \quad Adam \quad Smith \quad worked \quad ...\}$, the resulting retriever would yield the following matrix $\mathbb{e}_{\hat{s}}$:

$$
\begin{array}{r}
 \\
... \\
Adam \\
Smith \\
worked \\
...
\end{array}
\begin{array}{c}
\begin{array}{ccccc} ... & B-PER & I-PER & O & ... \end{array} \\
\begin{pmatrix}
... & ... & ... & ... & ... \\
... & 1 & 0 & 0 & ... \\
... & 0 & 1 & 0 & ... \\
... & 0 & 0 & 1 & ... \\
... & ... & ... & ... & ...
\end{pmatrix}
\end{array}
\tag{5}
$$

The sparse vectors $\mathbb{e}_{\hat{s}}$ are converted to dense representation by multiplying entities' corresponding type embeddings $\mathbf{E}_{tag}$,

$$g = \mathbb{e}_s \cdot \mathbf{E}_{tag} \tag{6}$$

where $\mathbf{E}_{tag}$ are the shared embeddings computed in Equation (8), the details are shown in corresponding section.

Single Knowledgeable Adapter. The adapter layer generally uses a down-projection with $\theta_A^{down} \in \mathcal{R}^{d \times r}$ to project the concatenation of the input h and its corresponding knowledge graph embeddings g to a lower-dimensional space specified by bottleneck dimension r, followed by a nonlinear activation function $f(\cdot)$, and a up-projection with $\theta_A^{up} \in \mathcal{R}^{d \times r}$. These adapters are surrounded by a residual connection, leading to a final form:

$$h \leftarrow h + f(hg\theta_A^{down})\theta_A^{up} \tag{7}$$

The output of the adapter is then passed directly into the following layer normalization.

4.3. Knowledge Enhanced Learnable Shared Representation Module

It is formal to devise a strategy to convert NER types into a natural language form for addressing the issue of limited information in few-shot examples. When incorporating KG type sequence into label semantic NER task, the heterogeneity issue between KG and NER type systems is the key challenge. To denoising knowledge and better empower advantages of label semantic strategy, we merge NER label classes into the categories in KG entity type taxonomy. With the unified taxonomy, shared representations $\mathbf{E}_{tag}$ between KG entity type sequence in the adapter module and label classes can be built. The mutal interaction can perceive semantic knowledge in entity type categories and boost the performance of few-shot NER.

Soft Shared Encoder Construction. Most existing prompting methods design the learned-from-scratch verbalizer to leverage label semantics, which is lack of human prior knowledge. These methods typically only induce a few words or embeddings that are closely related to the class name in terms of word meaning or distance. As a result, they struggle to infer words that capture different levels of specificity (for example, going from "Person" to "baseball player"). However, if we create a sub-graph in the KG for each label class based on the corresponding entity type node in the KB ontology graph, we can significantly improve the accuracy of predictions. Therefore, to improve the coverage and reduce the bias of the shared encoder, we propose incorporating ontology knowledge into the encoder to assist with adapter tuning.

First, we scan all the entries in the KB ontology graph and extract their associated texts to create a large-scale corpus. Then, for each label class or entity type c in the set $\mathcal{C}$, and its corresponding ontology entity e_c, we generate a sub-graph $\mathcal{G}_c$ with a maximum of m hops, centered around e_c. This sub-graph includes all the relation paths starting from e_c in the graph G. Here, m is limited to a maximum of 3 hops. Taking the example of the label class c being "Loc", we obtain the set $\mathcal{V}_C$ as {“Location”,“Place”,“Natural Place”, ..., “Park”} based on the decomposition of c. Once the sub-graph $\mathcal{G}_c$ is constructed, we extract both semantic and structural knowledge from it. Following the approach proposed in (Hu et al., 2022), we employ continuous vectors to incorporate ontology knowledge into PLMs (Pre-trained Language Models). Specifically, we establish a mapping $\mathcal{M}$ from the KB sub-graph $\mathcal{G}$ of entity categories $\mathcal{C}$ to a learnable shared encoder $\mathcal{V}$. This mapping, denoted as $\mathcal{M} : \mathcal{C} \mapsto \mathcal{V}$, represents the random walk in the ontology graph. By traversing the graph, we obtain the corresponding ontology words for each entity type. Finally, we utilize a frozen BART model as an encoder to capture the semantic knowledge in the entity types.

Hyper	Value
Epoch	30
Warmup Step	0.01
Learning Rate	[1e-5, 2e-5, 4e-5]
Batch Size	16
BART	Large
α	0.5
Beam Size	[1,4]

Table 1: Hyper-parameters used for CoNLL-2003, OntoNotes5.0, WNUT-2017.

$$\mathbf{E}_{tag} = BART(\mathcal{M}(\mathcal{G})) \tag{8}$$

Shared Encoder Refinement. Although we have created a shared encoder that can understand and represent a wide range of words related to ontology, the words we collected may not be very accurate because the specific vocabulary of the knowledge base (KB) was not designed specifically for the PLM. Therefore, it is important to improve the quality of this encoder by refining the semantics of the words. Specifically, we assign a weight θ that can be learned to label classes (entity types) in order to enhance the accuracy of the prediction logit, which represents the ontology representation of the corresponding labels in the answer space.

$$p_{tag} = \theta_{\mathcal{V}} \star \mathbf{E}_{tag} \bigotimes \mathbf{h}_t^d \tag{9}$$

where $\theta_{\mathcal{V}}$ denotes the learnable parameters for the knowledge enhanced shared encoder. By utilizing the soft learnable shared encoder, Know-Adapter has the ability to understand ontology knowledge in entity categories without making any changes to the PLM.

5. Experiment

We conduct extensive experiments in standard and low-resource settings. We use CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) as the rich-resource domain. We use the WNUT-2017 (YLeon et al., 2017) and OntoNotes5.0 (Ralph et al., 2013) datasets as the cross domain low-resource datasets.

Type	Methods	CoNLL-2003			OntoNotes5.0			WNUT-2017		
		5	10	20	5	10	20	5	10	20
Knowledge Empowered FPFTs	ERNIE-M ⋆	42.15	58.63	65.19	34.76	55.62	60.42	16.80	23.73	30.89
	ERNIE⋆	41.92	58.16	65.05	34.45	55.03	59.41	16.18	23.21	30.05
	CoLAKE⋆	41.34	57.96	64.98	34.21	54.97	59.24	16.03	22.99	29.92
Prompt based Learning	Template	43.04	57.86	66.38	40.52	49.89	59.53	19.25	25.53	31.91
	LightNER	28.62	42.96	65.25	39.26	48.13	58.32	18.13	24.97	31.29
	EntLM	49.59	64.79	69.52	45.21	57.64	65.64	24.82	31.28	34.75
Knowledge Enhanced PEFTs	KP-PLM⋆	40.92	56.28	63.21	34.09	53.74	59.07	16.30	23.11	29.09
	Know-Adapter	60.28	69.19	73.42	58.93	66.83	68.67	29.30	39.91	41.03

Table 2: Model performance ($F1$ score) under different few-shot settings ($K = 5, 10, 20$). All of our experiments and baselines adopt large version of LMs. $\star$ indicates that we rerun their public code in this setting. FEFTs (Full Parameter Fine-tuning Methods), PEFTS (Parameter Transfer Fine-tuning Methods).

5.1. Implementation Details and Baselines

Considering the instability of the few-shot learning, we run each experiment 5 times with the random samples and report the averaged performance. We utilize Pytorch to conduct experiments with 1 Nvidia 3090 GPUs. We describe the details of the training hyper-parameters in Table 1. We choose the model performing the best on the validation set and evaluate it on the test set.

We refer to our model as Know-Adapter and compare it to the following groups of baselines. (1) knowledge empowered full parameter fine-tuning methods (FPFTs):ERNIE-M (Ouyang et al., 2021), ERNIE (Zhang et al., 2019) and CoLAKE (Sun et al., 2020). (2) promt based learning method: Template Cui et al., LightNER (Chen et al., 2022b), EntLM (Ma et al., 2022) and PromptNER (Shen et al., 2023). (3) knowledge enhanced PEFTs: KP-PLM (Wang et al., 2022a). (4)Standard supervised Methods: LC-BERT, LC-BERT and BARTNER (Yan et al., 2021).

5.2. Overall Performance

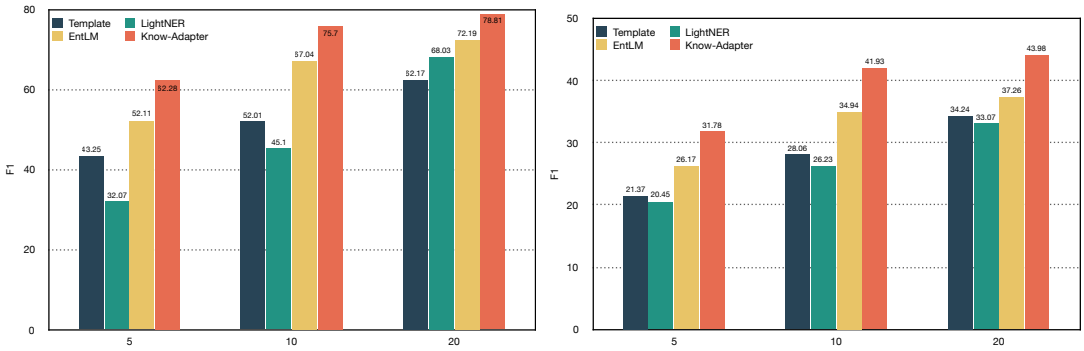
Few-shot NER. In the study of few-shot Named Entity Recognition (NER), our goal is to determine if the PEFT model effectively captures essential entity information from the knowledge graph in few-shot scenarios. We evaluate various methods based on their F1 scores, as shown in Table 2. Among all the training settings on the CoNLL-2003, OntoNotes5.0, and WNUT-2017 datasets, Know-Adapter consistently achieves the best overall results. Know-Adapter outperforms other methods as the number of shots decreases. Specifically, on the WNUT-2017 dataset with a 5-shot setting, Know-Adapter performs as the best-performing methods and surpasses prompt-based methods by 5.5% F1 score. The substantial large margin of Know-Adapter over other methods in the 5-shot setting demonstrates its effectiveness.

In the 20-shot experiments, we observe that the performance gap between different methods narrows as training data increases, but Know-Adapter consistently outperforms prompt learning-based methods. Surprisingly, the LightNER model does not outperform the BART-based prompt learning method with templates on any of the three benchmark datasets. Similarlyz, knowledge-enabled supervised fine-tuning approaches do not offer any advantages over our Know-Adapter model, suggesting their limitations in more realistic few-shot scenarios.

Notably, Know-Adapter significantly outperforms the knowledge-enhanced PEFT method KP-PLM, demonstrating the effectiveness of our external knowledge integration for few-shot NER. In contrast, KP-PLM gains little benefit, suggesting that its knowledge is not well-aligned with the NER task and may hinder PLMs performance.

Cross Domain Few-shot NER. In this paragraph, we evaluate the performance of the model in different scenarios where the target entity categories and textual style are either similar or different from the source domain. Additionally, we only have a limited amount of labeled data for training. The results of training models on the CoNLL-2003 dataset as a generic domain and evaluating them on other target domains are presented in Figure 2.

According to the data in Figure 2, when evaluating our model on two target domain datasets, Know-Adapter consistently outperforms all baselines significantly, even without leveraging any knowledge in the case of 5 instances per entity type. From the perspective of quantifying the knowledge transferred, when the number of instances is 5, our model achieves F1-scores of 62.28% and 31.78% on the OnotNotes5.0 and WNUT-2007 datasets, respectively. These results surpass the performance achieved by state-of-the-art methods such as EntLM and LightNER, indicating that our model successfully transfers learned knowledge from the source domain.



(a) Label Similar: OntoNotes5.0 (b) Label Different: WNUT-2017

Figure 2: Model performance ($F1$ score) in the cross-domain low-resource setting (5/10/20/50/100-shot). The results of Know-Adapter are based on running the experiments five times on random samples and calculating the average of their scores.

Standard SFTs	**P**	**R**	**F**
LC-BERT	91.93	91.54	91.73
LC-BART	89.60	91.63	90.60
BARTNER	92.31	93.45	92.98
Knowledge Empowered FPFTs	**P**	**R**	**F**
ERNIE-M	-	-	93.28•
ERNIE⋆	92.18	93.11	92.64
CoLAKE⋆	91.79	93.21	92.49
Prompt Based Learning	**P**	**R**	**F**
Template	90.51	93.34	91.90
LightNER	92.39•	93.48•	92.93
PromtNER	92.48	92.33	92.41
Knowledge Enhanced PEFTs	**P**	**R**	**F**
KP-PLM ⋆	91.42	92.23	92.17
Know-Adapter	93.18	94.40	93.79

Table 3: Model performance (F1 score) under standard supervised NER setting. All the experiments adopt large version of LMs. • indicates the best performing baselines. ⋆ indicates that we re-run their public codes under this setting.

Notably, compared to the state-of-the-art baseline EntLM, the F1 increasment on OntoNotes5.0, a label-similar dataset, is much larger than that on WNUT-2017, a label-dissimilar dataset, under the 5-shot setting. While Even for knowledge transfer with different labels, Know-Adapter achieves an F1 score of 41.93% in the 10-shot setting of WNUT-2007, outperforming all prompt-based learning methods (Template, LightNER, and EntLM) in the 20-shot setting. This means our approach has the ability of knowledge transfer even when the source and target labels differ, requiring only a small amount of labeled data to achieve acceptable few-shot performance.

Standard Supervised NER. A comparison of the results obtained from Know-Adapter and the state-of-the-art methods can be found in Table 3. LC-BERT and LC-BART serve as strong baseline models. However, most knowledge-empowered full fine-tuning methods based on BERT outperform LC-BERT by a significant margin. Additionally, when compared to the BART-based prompt tuning methods LightNER and PromtNER, LC-BART shows weaker performance, while Know-Adapter achieves better results. This suggests that both the PEFT strategy and the integration of knowledge graph information in Know-Adapter contribute to notable performance improvements. Interestingly, we found that Know-Adapter, designed for few-shot Named Entity Recognition, remains highly competitive even in settings with abundant resources, indicating the effectiveness of our knowledgeable adapter module.

5.3. Ablation Study

We conduct comprehensive ablation studies to reveal the design choices in the proposed parameter-efficient transfer learning framework for few-shot NER and investigate the contributions of different components of our Know-Adapter.

From Table 4, we notice that both `Know-Adapter (w/o KA)` and `Know-Adapter (w/o KE)` drop in both vanilla few-shot setting and cross-domain few-shot setting on WNUT-2017. It demonstrates that the design of the knowledgeable adapter module is both parameter-efficient and beneficial for domain knowledge transfer and label class knowledge transfer like knowledge enhanced shared encoder, which is also essential for few-shot NER. We further compare Know-Adapter with several non-knowledge aware PEFTs variants. Know-

Methods	WNUT-2017		
	10	20	50
Know-Adapter	29.30	39.91	41.03
w/o KA	25.31	32.47	36.82
w/o KE	26.53	33.62	37.94
r KA w Adapter	27.02	34.86	39.75
r KA w p-tuning	27.08	34.91	40.06
Know-Adapter c	31.78	41.93	43.98
w/o KA c	26.41	36.33	38.42
w/o KE c	27.94	37.46	39.35

Table 4: Model performance (F1 score) in the cross-domain low-resource setting. `w/o`, `w`, `r` and `c` indicate without, with, replace and cross-domain, respectively. `KA` represents the knowledgable adapter module in our model. `KE` denotes the knowledge enhanced learnable shared representation module in our model.

Methods	WNUT-2017		
	10	20	50
Know-Adapter(KR)	29.30	39.91	41.03
Know-Adapter(DS)	27.31	33.16	37.97
w/o KG type sequences	27.02	34.86	39.75

Table 5: Model performance (F1 score) in the cross-domain low-resource setting. `KR` represents the KG type sequences used in the knowledgeable adapter module are generated by our knowledge retriever. `DS` represents the KG type sequences used in the knowledgeable adapter module are generated by the distant supervision method. `w/o` indicates without.

Adapter achieves the optimal performance among other non-knowledge aware PEFTs. We speculate that this may be because KG type sequences enhance information for few-shot NEs, allowing the model to derive background knowledge. Therefore, our model is not only remarkably expressive but also insensitive to knowledge graph information of NEs. This suggests that we do not have to design tailored knowledge graph information about NEs except for KG type sequences for the few-shot NER task. In addition, both `Know-Adapter (r KA w Adapter)` and `Know-Adapter (w/o KE)` behaves much worse than Know-Adapter in a low-resource setting, which reveals the unified type taxonomy do alleviate the heterogeneity issue and shared representations between additional KG entity type sequences and NER label classes have strong ability of knowledge transfer for NER.

5.4. Discussion:Do Know-Adapter have memorized knowledge or truly have the generlization ability?

Drawing from previous experiments, it is apparent that Know-Adapter are adept at swiftly extracting KG type knowledge via the knowledge retriever. This observation raises a question regarding the origin of the performance advantage in Know-Adapter: is it due to the substantial KG type sequences obtained by the knowledge retriever, enabling the models to acquire pertinent knowledge, or is it attributed to their robust inference and generalization capabilities based on the background knowledge of KG type sequence?

To delve into this question, we utilize a distant supervision method (denoted as DS) (Shang et al., 2018b,a) to generate KG type sequences instead of our knowledge retriever (denoted as KR), which will directly provide supervision signals but also introducing more noise. As is shown in Table 5, Know-Adapter(KR) achieves relatively best performance on the WNUT-2017 dataset in both 10-shot, 20-shot and 50-shot manners. Especially, Know-Adapter(DS) exihibits no improvement compared to `w/o KG type sequences`. This demonstrates Know-Adapter(KR) utilizes background knowledge retrieved by our knowledge retriever to enhance the information of few-shot examples and stimulate the generalization of PLMs. And this also proves that simply using the distant supervision labels for adapter tuning can introduce more noise that interferes with the model's performance.

Conclusion

In this work, we propose a knowledge-enhanced adapter tuning method, Know-Adapter, for few-shot NER. Specifically, we leverage KG entity type sequence to guide the updating process of the adapter. To boost the few-shot NER performance, we also construct a shared representation module to obtain shared representations of label classes and KG entity type sequence. Experimental results show that the proposed method can achieve significant improvement on few-shot NER over knowledge empowered full fine-tuning methods and prompt-based method. Also, compared general knowledge enhanced PEFT model KP-PLM, better results reveal that introducing appropriate knowledge instead of general factual knowledge of KG dominates for external knowledge sensitive task like few-shot NER. Moreover, utilizing KG type sequences generated by the distant supervision method is not effective for performance improvement, which prove that KG type sequence retrieved by our knowledge retriever provide background knowldge to stimulate the generalization of PLMs.

Acknowledgments

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No.LQ23F020016; the Fundamental Research Funds for the Provincial University in Zhejiang under Grant No.GK229909299001-022.

References

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2021. Interpretability Analysis for Named Entity Recognition to Understand System Predictions and How They Can Improve. *Computational Linguistics*, 47(1):117–140.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022a. Dictbert: Dictionary description knowledge enhanced language model pre-training via contrastive learning. In *Proceedings of the international joint conference on artificial intelligence*, pages 4086–4092.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2022b. LightNER: A lightweight tuning paradigm for low-resource NER via pluggable prompting. In *Proceedings of the International Conference on Computational Linguistics*, pages 2374–2387.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4115–4128.

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 Annual Meeting of the Association for Computational Linguistics*, pages 3198–3213, Online.

Matthew Henderson and Ivan Vulić. 2021. ConVEx: Data-efficient and few-shot slot labeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3375–3389, Online.

Ye Hongbin, Zhang Ningyu, Deng Shumin, Chen Xiang, Chen Hui, Xiong Feiyu, Chen Xi, and Chen Huajun. 2022. Ontology-enhanced prompt-tuning for few-shot learning. In *Proceedings of the World Wide Web Conference*, page 778–787.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the International Conference on Machine Learning*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240.

Ge Ling, Hu Chunming, Ma Guanghui, Zhang Hong, and Liu Jihong. 2023. Prokd: An unsupervised prototypical knowledge distillation network for zero-resource cross-lingual named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12818–12826, Online.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 27–38.

Weischedel Ralph, Palmer Martha, Marcus Mitchell, Hovy Eduard, Pradhan Sameer, Ramshaw Lance, Xue Nianwen, Taylor Ann, Kaufman Jeff, and Franchini Michelle. 2013. Ontonotes release 5.0 ldc2013t19. In *Linguistic Data Consortium Philadelphia, PA, 23*.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018a. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*.

Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. 2018b. Learning named entity tagger using domain-specific dictionary. In *EMNLP*.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating and typing for named entity recognition. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 12492–12507.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 3660–3670.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–147.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, pages 6000–6010.

Jianing Wang, Wenkang Huang, Minghui Qiu, Qiuhui Shi, Hongbin Wang, Xiang Li, and Ming Gao. 2022a. Knowledge prompting in pre-trained language model for natural language understanding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3164–3177.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022b. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 5808–5822.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375.

Derczynski YLeon, Nichols Eric, van Erp Marieke, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of WNUT*, pages 140–147.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.