Dynamic Prefix as Instructor for Incremental Named Entity Recognition: A Unified Seq2Seq Generation Framework

Anonymous ACL submission

Abstract

The Incremental Named Entity Recognition (INER) task aims to update a model to extract entities from an expanding set of entity type candidates due to concerns related to data privacy and scarcity. However, conventional incremental learning methods for INER often suffer from the catastrophic forgetting problem, which leads to the degradation of the model's performance on previously encountered entity types. In this paper, we propose a parameter-efficient dynamic prefix method and formalize INER as a unified seq2seq generation task. By employing the dynamic prefix as a task instructor to guide the generative model, our approach can preserve task-invariant knowledge while adapting to new entities with minimal parameter updates, making it particularly effective in low-resource scenarios. Additionally, we design a generative label augmentation strategy and a novel self-entropy loss to balance the stability and plasticity of the model. Empirical experiments on NER benchmarks demonstrate the effectiveness of our proposed method in addressing the challenges associated with INER.

1 Introduction

Named Entity Recognition (NER) is a fundamental problem in information extraction tasks. Traditional NER systems typically require a large amount of annotated training data encompassing all predefined entity types. However, as new entity types emerge, retraining the entire model becomes impractical. Furthermore, obtaining sufficient supervised data for training is challenging due to concerns related to data privacy and scarcity (Ma et al., 2020). Consequently, continual learning (or incremental learning) for NER has been proposed (Monaikul et al., 2021) as a solution to train the model incrementally on new datasets labeled exclusively with new entity types, addressing the issues associated with retraining and data availability.

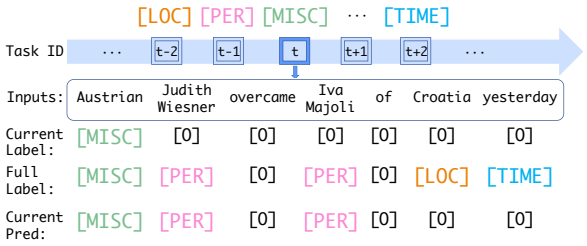

Figure 1: Challenges in class-incremental NER. At the current incremental step t, the data is only annotated with the current entity type [MISC], while previous entity types [LOC] and [PER] are annotated with [O]. [TIME] is a future entity type. "Current Pred" indicates that the model forgets previous entity type [LOC] after training at step t.

Continual learning aims to learn a sequence of tasks incrementally which mirrors the human capability of learning and accumulating knowledge continually without forgetting previously learned knowledge and and leveraging it to facilitate learning new tasks (Ke and Liu, 2022). However, catastrophic forgetting (McCloskey and Cohen, 1989) poses a significant challenge in continual learning where the model gradually forgets previous knowledge in the current learning step. In continual learning for NER, the information of previous and future entity types is missing in the current step. Ma et al. (2023) point out that the majority of prediction errors of INER stem from the confusion between pre-defined entities and other entities ("O"). As shown in Figure 1, the model learned to recognize "PER" (person) and "LOC" (location) in one step would be trained to annotate "PER" or "LOC" as "O" in current and subsequent steps. At step t, only the entity type "MISC" (miscellaneous) is labeled, which leads to the wrong prediction of the entity "Croatia". This indicates that the model has forgotten the entity information of "LOC" learned in previous tasks.

Directly training the model on the new data will exacerbate this problem with background shift

(Zhang et al., 2023), where old and future entity types are labeled as the non-entity type in the current task. This results in a significant performance drop on test data containing all encountered entities. To address this, we conduct an experiment to investigate the catastrophic forgetting problem in NER. As illustrated in Figure 2, we train our model with three different settings. The multi-task learning setting (the red line) serves as an upperbound since all the seen entity types are annotated in the new data. The naive method involves directly fine-tuning the model on the new task data (the blue line), leading to a sharp decline in F1 score for old entities. In contrast, when trained with continual learning methods, the model performance only decrease slightly compared to the upperbound, effectively alleviating the catastrophic forgetting problem.

Previous methods (Monaikul et al., 2021; Zheng et al., 2022; Zhang et al., 2023) treat INER as a sequence labeling classification task, which may encounter limitations, particularly in the era of Large Language Models (LLMs). Following traditional NER approaches, these methods use a text encoder to extract context representations, followed by a classification layer to assign each token an entity type label. When encountering new entity types, they need to expand the entity class set and initialize a new classification layer with old rows unchanged. Consequently, these approaches need to modify the model architecture, and both the parameters of the entire model and classification layer will be updated. Additionally, though sequence labeling methods have achieved outstanding performance on the INER task, they struggle to recognize nested entities and require task-specialized solutions or model modifications (Yan et al., 2021).

Motivated by them, in this paper, we formalize INER as a seq2seq generation task, which not only aligns well with the nature of NER but also facilitates prompt tuning in a more intuitive manner. Our proposed method leverages a parameter-efficient dynamic-prefix strategy for incremental learning in INER. By dynamically appending prefixes as instructors during the incremental process, our model inspires the model to acquire new knowledge while retaining old prefixes to maintain stability. Different from prior INER methods, all prefixes are pluggable, and no modifications are applied on the base model, making our method more practical. More importantly, the generative nature of the proposed model, as opposed to relying solely on classification, facilitates the adaptation to new entities while preserving task-invariant knowledge, especially in low-resource scenarios.

Specifically, we integrate manually constructed task instructions and entity type options in the input sentence (as shown in Figure 3). Then we introduce dynamic prefix as an instructor to guide the frozen Pre-trained Language Model (PLM) in learning new entity types incrementally. When training the model at each step, we dynamically increase the number of prefixes, where the newly appended prefixes are the only trainable parameters. This results in significantly fewer parameters to fine-tune compared to prior INER methods. During inference, all prefixes collaborate to generate a sequence of entity types from current options and their corresponding entities. Moreover, we integrate the generation-based label augmentation strategy and self-entropy loss to achieve a more refined equilibrium between stability and plasticity.

Our main contributions are summarized as follows:

- We propose a dynamic prefix method to retain task-invariant capabilities and preserve task-specific knowledge in INER.
- As an instructor, our proposed dynamic prefix method inspires the seq2seq model, demonstrating robustness and practicality, particularly in more realistic low-resource setting.
- Empirical experiments on INER benchmark demonstrate the effectiveness of our proposed DPI. Notably, our method based on generation architecture achieves better performance with significantly fewer fine-tuned parameters than prior sequence labeling INER methods.

2 Related Work

2.1 Class-Incremental Learning

Prior approaches to class-incremental learning can be divided into three categories: (1) **Architecture-based** methods dynamically adjust the model architecture to learn new knowledge while mitigating forgetting of previously learned tasks (Chen et al., 2016; Rusu et al., 2016; Mallya et al., 2018). (2) **Regularization-based** methods constrain the updates of parameters that are important to the learned tasks to retain previous knowledge (Li and Hoiem, 2017; Kirkpatrick et al., 2016; Aljundi et al., 2018). (3) **Rehearsal-based** methods keep

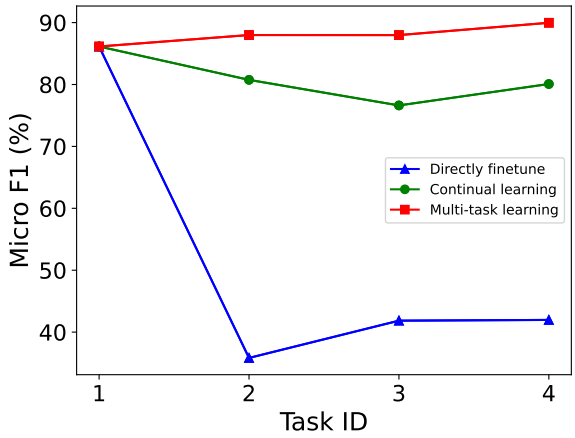

Figure 2: An illustration of catastrophic forgetting . We conduct the comparison with three different settings on the CoNLL03 (Sang and De Meulder, 2003) dataset.

exemplars from previous tasks in memory to alleviate forgetting (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019; de Masson d'Autume et al., 2019).

2.2 Prompt Tuning in Continual Learning

There have been some explorations in using prompts to enhance performance in continual learning of image classification. As a lightweight alternative to fine-tuning, prompt-based methods often learn a prompt pool or a series of soft prompts to instruct the model while keeping the base model frozen (Wang et al., 2022b;Razdaibiedina et al., 2023;Wang et al., 2022a). These prompts serve as both task-invariant and task-specific instructions. When learning new tasks, the prompt pool is updated, or new prompts are introduced, ensuring the preservation of knowledge from previous tasks. Some works have already demonstrated that prompts can alleviate the problem of catastrophic forgetting to a certain extent (Smith et al., 2023). For instance, Razdaibiedina et al. (2023) propose Progressive Prompts and demonstrate their efficacy across 15 text classification tasks.

2.3 Incremental Named Entity Recognition

Monaikul et al. (2021) introduce the incremental learning paradigm into NER (i.e., INER) and propose AddNER and ExtendNER to alleviate catastrophic forgetting. L&R (Xia et al., 2022) adopts a replay-based approach to synthesize samples of old entity types. CFNER (Zheng et al., 2022) and RDP (Zhang et al., 2023) focus on extracting information from non-entity type and task relationships. Ma et al. (2023) proposes an entity-aware contrastive learning method that adaptively detects entity clusters in the "O" class. In line with CFNER and RDP, our method is rehearsal-free and do not keep any exemplars from previous tasks.

2.4 Generation based Named Entity Recognition

A seq2seq architecture is introduced with a pointer mechanism in Yan et al. (2021) to generate entity index sequences. Lu et al. (2022) introduce a universal information extraction model based on a unified generation structure. Chen et al. (2023) propose a collaborative prefix method based on the generative paradigm for knowledge transfer. However, in INER, it is essential to consider not only the performance in the target domain but also across all tasks. As a consequence, these methods show limited performance when directly applied to INER since they are not designed for incremental scenarios.

3 Methodology

In this section, we introduce our dynamic prefix method designed to facilitate INER by seq2seq generation framework. We start with providing a formalized definition of INER in Section 3.1, followed by the working mechanism of prefix tuning for NER in Section 3.2. In Section 3.3 we propose a dynamic prefix method as a task-invariant and task-specific instructor based on seq2seq generation framework. Finally, Section 3.4 outlines the strategy employed to achieve a balance between stability and plasticity of INER.

3.1 Problem Definition

Following previous works (Monaikul et al., 2021;Xia et al., 2022;Zheng et al., 2022;Zhang et al., 2023;Ma et al., 2023), we focus on class-incremental learning on NER (INER). Formally, INER contains N incremental steps, each associated with its corresponding task $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$. Every task has its own dataset $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$. Specifically, the task at the t-th step can be described as $\mathcal{T}_t = (\mathcal{D}_t^{tr}, \mathcal{D}_t^{dev}, \mathcal{D}_t^{test}, \mathcal{C}_t^{new}, \mathcal{C}_t^{old})$, where $\mathcal{C}_t^{new}$ is the label set (i.e., new entity types) of the current task (e.g., {"PER", "ORG"}) and $\mathcal{C}_t^{old} = \bigcup_{i=1}^{t-1} \mathcal{C}_i^{new}$ represents the label set containing all seen entity types in old tasks. Each task has its unique training set $\mathcal{D}_t^{tr} = \{X_t^j, Y_t^j\}_{j=1}^n$, where $X_t^j = \{x_t^{j,1}, \ldots, x_t^{j,l}\}$ (with l as the se-

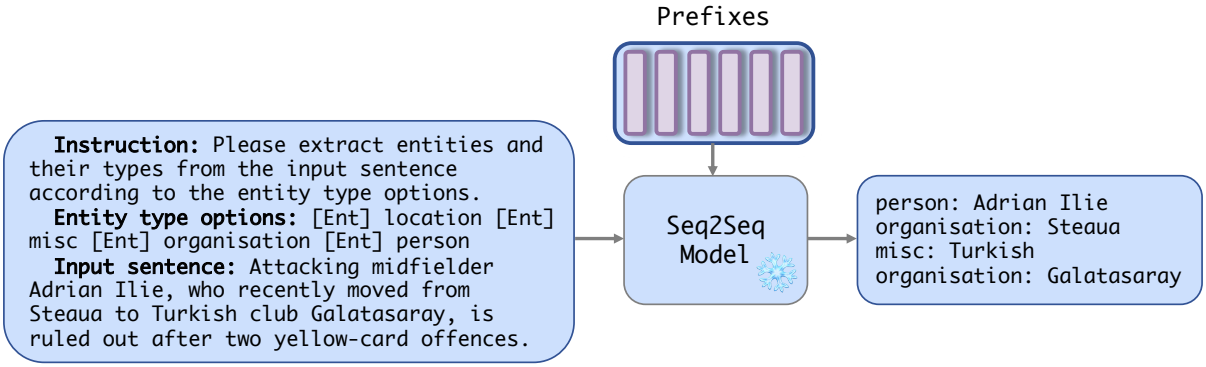

Figure 3: An illustration of the unified seq2seq approach for NER.

quence length) and $Y_t^j = \{y_t^{j,1}, \ldots, y_t^{j,l}\}, y_t^{j,k} \in \mathcal{C}_t^{new} (k = 1, \ldots, l)$ are annotated with only the new entity types or "O". At step t, with the model $\mathcal{M}_{t-1}$ trained at step $t-1$, we update $\mathcal{M}_{t-1}$ at $\mathcal{T}_t$ in order to train a model $\mathcal{M}_t$ which is expected to perform well on all seen entity types $\mathcal{C}_t^{all} = \mathcal{C}_t^{new} \cup \mathcal{C}_t^{old}$.

3.2 Prefix Tuning for Seq2Seq Generation in Named Entity Recognition

Prompt-based learning has been widely applied in NLP tasks, especially with the rise of LLMs. By providing manually designed hard prompts or attaching a set of soft prompts, they can serve as instructions for Pre-Trained Language Models (PLMs) in downstream tasks.

Specifically, given the input $(X^j, Y^j) \in \mathcal{D}^{tr}$, a sequence of soft prompts can be prepended to each layer of the transformer to obtain the input as: $Z^j = [\text{PREFIX}; X^j; \text{PREFIX}'; Y^j]$ (Li and Liang, 2021). The activations of the prefix are always in the left context and will therefore affect subsequent activations to the right.

Based on prompt-based learning, we tackle the NER problem in a seq2seq paradigm, which offers an intuitive framework for integrating prompt-based techniques. Figure 3 shows the unified seq2seq procedure. The trainable prefixes serve as a guide for the seq2seq model, prompting it to extract all entities and the corresponding entity types in the input sentence. Formally, given the manually constructed task *instruction* ($\mathbf{s}$) specific to NER, at each step t the model takes the input sentence X_t with the *entity type options* ($\mathbf{o}_t$), and generates a sequence $\hat{\mathbf{y}}_t$ which is expected to contain all entity types and their corresponding entities:

$$\hat{\mathbf{y}}_t = \text{LM}_{\phi,\theta}(\mathbf{s}; \mathbf{o}_t; X_t), \tag{1}$$

where the language model parameters ϕ are frozen and the prefix parameters θ are the only trainable parameters in our continual steps. Note that we can obtain the label sequence $\hat{y}_t$ by post-processing the original output $\hat{\mathbf{y}}_t$.

3.3 Dynamic Prefix

When it comes to the incremental setting, the objective of the seq2seq INER is:

$$\max_{\phi,\theta} \sum_{t=1}^{N} \sum_{(x,y)\in\mathcal{T}_t} \log p(y|x, \phi, \theta) \tag{2}$$

To adapt our method to the incremental setting, we propose a *Dynamic Prefix* method as illustrated in Figure 4. We dynamically increase the number of prefixes which are expected to learn task-specific knowledge. Simultaneously, by concatenating newly added prefixes with the existing ones, we prevent forgetting knowledge pertaining to previous entity types, while adapting to new entities with minimal parameter updates and maximal knowledge acquisition. Specifically, when training the incremental task $\mathcal{T}_t$, a set of new prefixes $\mathbf{P}_t \in \mathbb{R}^{|L_t| \times d}$ with length of $|L_t|$ parameterized by θ_t are inserted into each layer while keeping the LM parameters (ϕ) and all old prefix parameters $(\theta_1, \ldots, \theta_{t-1})$ frozen. The objective of our dynamic prefix approach becomes:

$$\max_{\phi,\theta_1,\ldots,\theta_t} \sum_{t=1}^{N} \sum_{(x,y)\in\mathcal{T}_t} \log p(y|x, \phi, \theta_1, \ldots, \theta_t) \tag{3}$$

As shown in Figure 4, we concatenate the new prefixes with the old prefixes along the prefix length dimension. Then the entire set of prefixes $\mathbf{P}$ is split into $\mathbf{P}_k$ and $\mathbf{P}_v$, which are concatenated with the original keys $\mathbf{K}$ and values $\mathbf{V}$ to compute each head vector. The computation of the i-th head vector head_i can be written as:

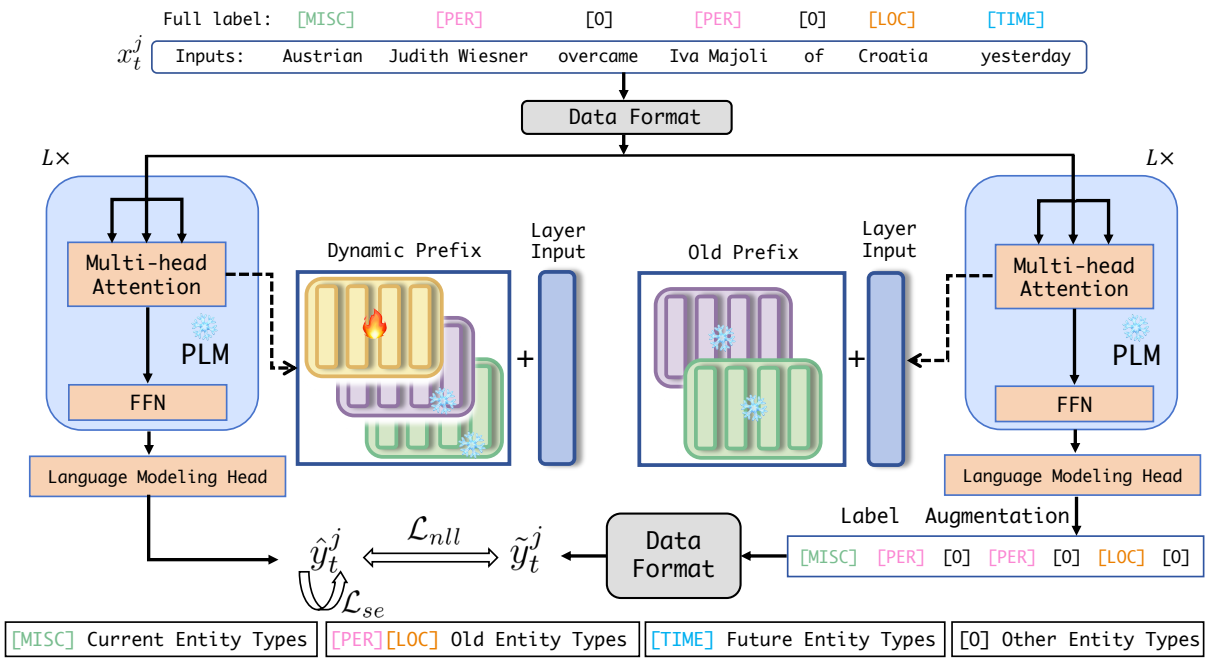

Figure 4: The overall architecture of our proposed DPI for INER. Here "+" denotes the concatenation operation.

$$\text{head}_i = \text{Attn}(xW_q^{(i)}, [\mathbf{P}_k^{(i)}; CW_k^{(i)}], [\mathbf{P}_v^{(i)}; CW_v^{(i)}]) \quad (4)$$

The activation vector $h_i \in \mathbb{R}^d$ at time step i is computed as:

$$h_i = \begin{cases} \mathbf{P}[i,:], & \text{if } i \in L \\ \text{LM}_\phi(Z_i, h_{<i}), & \text{otherwise} \end{cases} \quad (5)$$

where $\mathbf{P} \in \mathbb{R}^{|L| \times d}$ is a partially trainable matrix with $L = [L_1; \ldots; L_k; \ldots; L_t]$. L_k denotes the sequence of prefix indices of new prefixes at incremental step k.

Then we optimize the new prefix parameters θ_t by minimizing the negative log-likelihood over the training set D_t^{tr} of task $\mathcal{T}_t$.

$$\mathcal{L}_{\text{nll}}(\theta_t) = -\sum_{(x,y) \in D_t^{tr}} \log p(y|[\mathbf{P}_t, \ldots, \mathbf{P}_1, x], \phi, \theta_1, \ldots, \theta_t) \quad (6)$$

where the only trainable parameters are θ_t related to new prefixes.

3.4 Equilibrium Between Stability and Plasticity

The entities annotated with "O" at the current step may belong to the previous entity types $\mathcal{C}_t^{old}$ or the future entity types $\bigcup_{i=t+1}^{N} \mathcal{C}_i^{new}$. Obviously, the future entity types cannot be seen in the current task. For entities that belong to $\mathcal{C}_t^{old}$, we employ a generation-based label augmentation strategy. This strategy leverages the capabilities of the old model. By leveraging the old entity type information contained in tokens annotated with "O", the stability is enhanced when learning new entity types. Before training each task, we utilize the old model $\mathcal{M}^{t-1}$ to predict a "pseudo" entity type for entities annotated with "O". The augmented labels are then fused with the current labels for training the current task. As mentioned above, the original true label of the current task is denoted as $Y_t^j = \{y_t^{j,1}, \ldots, y_t^{j,l}\}$. To obtain the augmented label $\tilde{y}_t^{j,k}$ for the kth token of the jth input, we employ the strategy as follows:

$$\tilde{y}_t^{j,k} = \begin{cases} \hat{y}_{t-1}^{j,k}, & \text{if } y_t^{j,k} = \text{"}O\text{"} \\ y_t^{j,k}, & \text{otherwise} \end{cases} \quad (7)$$

where

$$\hat{y}_{t-1}^j = \arg\max_{o \in \mathbf{o}_{t-1}} \mathcal{M}_{t-1}(\mathbf{s}; \mathbf{o}_{t-1}; X_t) \quad (8)$$

After applying the label augmentation strategy, we obtain the final training set $\mathcal{D}'^{tr}_t$ for the current step t: $\mathcal{D}'^{tr}_t = \{X_t^j, \tilde{Y}_t^j\}_{j=1}^n$. The dynamic prefix approach and label augmentation strategy are expected to enhance the stability of our model.

To further extend the model's plasticity, we minimize the self-entropy loss to promote the model's

confidence in learning the new entity types:

$$\mathcal{L}_{\text{se}} = -\frac{1}{l}\sum_{k=1}^{l}\hat{y}^k \log \hat{y}^k \tag{9}$$

Here $\hat{y}^k$ denotes the output probability distribution.

With the augmented labels, the Equation (6) can be formulated as follows:

$$\begin{aligned}&\mathcal{L}_{\text{nll}}(\theta_t)\\ &= -\sum_{(x,y)\in D_t'^{tr}} \log p(y|[\mathbf{P}_t,\ldots,\mathbf{P}_1,x],\phi,\theta_1,\ldots,\theta_t)\end{aligned} \tag{10}$$

In summary, the objective function of our proposed method is:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{se}} \tag{11}$$

Dataset	# Entity Type	# Sample	Entity Type Sequence
CoNLL03	4	21k	LOCATION, MISC, ORGANISATION, PERSON
I2B2	16	141k	AGE, CITY, COUNTRY, DATE, DOCTOR, HOSPITAL,IDNUM, MEDICALRECORD, ORGANIZATION,PATIENT, PHONE, PROFESSION, STATE, STREET,USERNAME, ZIP

Table 1: Statistics of the NER datasets CoNLL03 (Sang and De Meulder, 2003) and I2B2 (Murphy et al., 2010).

4 Experiment

4.1 Experimental Settings

Datasets. We conduct experiments on two widely used NER dataset: CoNLL03 (Sang and De Meulder, 2003) and I2B2 (Murphy et al., 2010) for evaluating the effectiveness of our method. The dataset statistics are shown in Table 1. Following CFNER (Zheng et al., 2022), for each dataset, a greedy sampling strategy is adopted to partition the training set into disjoint slices to better simulate realistic scenarios. Each slice corresponds to an incremental step. Specifically, FG entity types are used to train the initial model, and PG entity types are used for training in each subsequent incremental step. For example, under the "FG-8-PG-2" setting, 8 entity types are annotated in the first step and 2 entity types are annotated in each subsequent step. After dividing the original dataset into slices, we utilize UIE [1] for data pre-processing. Finally, the data annotated with "BIO" schema is converted into the UIE format (Lu et al., 2022) (i.e., the "Data Format" module in Figure 4) for seq2seq generation.

[1] https://github.com/universal-ie/UIE

Training. Different from previous works (Zheng et al., 2022;Zhang et al., 2023) using BERT-base (Devlin et al., 2018) for INER, we use T5-base (Raffel et al., 2019) as the backbone model for INER via seq2seq generation. Instead of fine-tuning almost all of the parameters, including the backbone model, at each incremental step as in previous methods, our dynamic prefix tuning method keeps the parameters of the backbone model frozen. The pluggable new prefixes are the only trainable parameters (approximately 0.1% of the backbone model). The implementation details can be found in Appendix A.

Baselines. We compare our method (DPI) with representative INER methods, including Extend-NER (Monaikul et al., 2021), CFNER (Zheng et al., 2022), and RDP (Zhang et al., 2023). Additionally, PODNet (Douillard et al., 2020) and LUCIR (Hou et al., 2019) are adapted to INER scenario by Zheng et al. (2022). We re-implement RDP which is the previous state-of-the-art INER method, while the results of the other baseline[2] are directly cited from Zheng et al. (2022).

5 Results and Discussion

5.1 Main Results

We report the results of our proposed method (DPI) on the CoNLL03 (Sang and De Meulder, 2003) and I2B2 (Murphy et al., 2010) datasets. We conduct experiments under INER settings and present the quantitative task-wise performance compared to the baselines. Notably, for a fair comparison, we re-implement the previous SOTA baseline RDP (Zhang et al., 2023) without knowledge distillation loss.

As shown in Table 2, the **Full Data** results, where all the seen entity types are annotated, are relatively stable, serving as an upperbound of our method. **Directly Fine-tune** represents the naive method where no incremental techniques are utilized, resulting in a sharp decline in performance. However, all the incremental learning methods show varying degrees of forgetting during the incremental process. Compared to the previous SOTA baselines CFNER (Zheng et al., 2022) and RDP (Zhang et al., 2023), our method demonstrates improvements in both average and task-wise results of CoNLL03 (Sang and De Meulder, 2003) under

[2] Please refer to Appendix C of CFNER (Zheng et al., 2022) for a more detailed introduction of the baselines and greedy sampling strategy.

Task ID Method	Trainable Param.	t=1 **[LOC]**	t=2 **+[MISC]**	t=3 **+[ORG]**	t=4 **+[PER]**	Avg.
Full Data	~0.1% of 220M	86.14	87.99	87.98	89.97	88.02
PODNet (Douillard et al., 2020)	~100% of 110M	85.96	11.13	24.16	25.49	36.74
LUCIR (Hou et al., 2019)		85.96	73.85	62.81	73.78	74.15
ExtendNER (Monaikul et al., 2021)		85.96	74.42	69.27	75.78	76.36
CFNER (Zheng et al., 2022)		85.96	80.63	76.10	**80.95**	80.91
RDP* †(Zhang et al., 2023)		84.53	77.31	**76.67**	79.22	79.43
DPI (Ours)	~0.1% of 220M	**86.14**	**81.90**	76.62	80.08	**81.19**
Directly Fine-tune		86.14	35.83	41.85	41.97	51.45

Table 2: Main results of the proposed method and baselines under the **FG-1-PG-1** setting of the CoNLL03 dataset (Sang and De Meulder, 2003). [LOC], [MISC], [ORG], and [PER] denote Location, Miscellaneous, Organization, and Person, respectively. Micro-F1 score is reported. * represents results from our re-implementation. † represents results without using knowledge distillation loss during continual learning. Other baseline results are directly cited from CFNER (Zheng et al., 2022).

Task ID Method	Trainable Param.	t=1 **[LOC], [MISC]**	t=2 **+[ORG]**	t=3 **+[PER]**	Avg.
PODNet (Douillard et al., 2020)	~100% of 110M	87.21	46.14	44.24	59.12
LUCIR (Hou et al., 2019)		87.21	74.59	80.02	80.53
ExtendNER (Monaikul et al., 2021)		87.21	67.93	74.84	76.66
CFNER (Zheng et al., 2022)		87.21	76.23	79.05	80.83
RDP*† (Zhang et al., 2023)		86.05	78.48	**82.45**	82.33
DPI (Ours)	~0.1% of 220M	**88.27**	**81.14**	78.69	**82.70**

Table 3: Comparison with baselines under the **FG-2-PG-1** setting of the CoNLL03 dataset (Sang and De Meulder, 2003).

the FG-1-PG-1 INER setting.

To simulate a realistic scenario allowing the model to acquire sufficient "base-knowledge" before incremental learning, we conduct experiments where we initially learn half of all entity types. The results of CoNLL03 (Sang and De Meulder, 2003) under FG-2-PG-1 and I2B2 (Murphy et al., 2010) under FG-8-PG-2 are summarized in Table 3 and Table 4, respectively, demonstrating an improvement of approximately 0.4% and 6.9% compared to RDP (Zhang et al., 2023). To delve deeper into the performance of DPI, we conduct experiments with a broader range of incremental steps. As depicted in Figure 5, under the FG-2-PG-2 setting of I2B2, a total of 8 steps are considered. The performance of CFNER (Zheng et al., 2022) declines significantly with deeper incremental steps. However, our method consistently outperforms the previous SOTA method RDP (Zhang et al., 2023) throughout the incremental steps. Table 2 and Table 5 indicate that our method outperforms significantly with the previous methods when encountering more entity types and incremental steps.

These quantitative results indicate that our proposed method can achieve better performance and alleviate catastrophic forgetting by fine-tuning significantly fewer parameters (approximately ~0.1% size of the entire PLM).

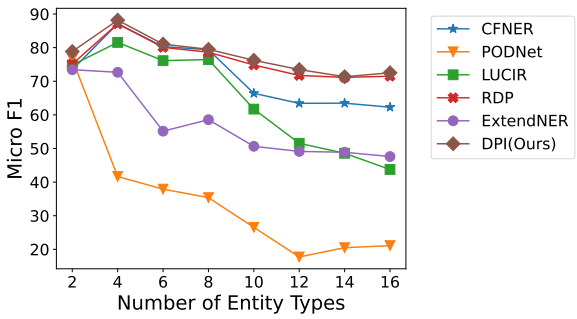

Figure 5: Task-wise results compared to baselines on the I2B2 (Murphy et al., 2010) dataset under the FG-2-PG-2 INER setting.

5.2 Low-resource Settings

Due to concerns related to data privacy and scarcity in realistic applications, INER often encounters low-resource scenarios. To further investigate the effectiveness of our method regarding the data scale, we conduct experiments with low-resource

Method	Trainable Param.	t=1	t=2	t=3	t=4	t=5	Avg.
PODNet (Douillard et al., 2020)		89.53	28.50	22.89	21.86	18.32	36.22
LUCIR (Hou et al., 2019)		90.23	72.0	63.18	60.96	56.32	68.54
ExtendNER (Monaikul et al., 2021)	~100% of 110M	89.39	53.84	42.25	39.31	36.47	52.25
CFNER (Zheng et al., 2022)		89.39	70.29	64.1	62.01	59.58	69.07
RDP*† (Zhang et al., 2023)		90.94	77.86	69.16	63.95	53.36	71.05
DPI (Ours)	~0.1% of 220M	**91.43**	**83.47**	**73.15**	**68.34**	**68.5**	**76.98**

Table 4: Comparison with baselines under the **FG-8-PG-2** setting of the I2B2 dataset (Murphy et al., 2010). Micro-F1 score is reported. * represents results from our re-implementation. † represents results without using knowledge distillation loss during continual learning. Other baseline results are directly cited from CFNER (Zheng et al., 2022).

settings. For each incremental step, we respectively sample 5% and 10% of the training set while adopting a greedy sampling strategy to partition the training set. We compare our DPI method with the previous SOTA approach RDP (Zhang et al., 2023) in Table 5. In the low-resource scenario with only 5% of the data available, our DPI method achieves an improvement over the previous SOTA baseline RDP (Zhang et al., 2023) by approximately 13.61% in Micro-F1 score. In a more stringent low-resource scenario, our method also outperforms RDP (Zhang et al., 2023) by approximately 12.19%. Our method demonstrates robustness in low-resource scenarios. While previous methods always fine-tune nearly all parameters during the incremental process, resulting in limited ability to extract useful information when the training data is scarce, our approach fine-tunes significantly fewer parameters at each step, effectively capturing the patterns of different entity types in low-resource scenarios.

Rate	Method	t=1	t=2	t=3	t=4	Avg.
10%	DPI (Ours)	60.63	50.85	55.07	66.90	**58.36**
	RDP	52.81	45.27	35.03	45.85	44.74
5%	DPI (Ours)	55.67	51.17	55.05	61.92	**55.95**
	RDP	54.00	36.27	39.73	45.03	43.76

Table 5: Performance in low -resource conditions on the CoNLL03 (Sang and De Meulder, 2003) dataset under the FG-1-PG-1 INER setting. The task-wise and average Micro-F1 scores are reported.

5.3 Ablation Studies

We conduct ablation studies to analyze the factors influencing the performance of our method. As shown in Table 6, all ablation factors degrade the INER performance of DPI. DPI w/o DP represents our method without the dynamic prefix strategy, where a fixed size of prefixes are trained throughout the incremental process. The results indicate that the fixed size of prefixes lack the continual ability, which is exacerbated with more incremental steps. DPI w/o LAS means no label augmentation strategy is employed and w/o $\mathcal{L}_{\text{se}}$ indicates the result without the self-entropy loss term. By employing LAS and introducing the self-entropy loss, we further achieve an equilibrium between stability and plasticity. Removing either of them will lead to a performance decline.

Method	CoNLL03		I2B2	
	FG-1-PG-1	FG-2-PG-1	FG-2-PG-2	FG-8-PG-2
DPI (Ours)	**81.19**	**82.70**	**77.64**	**76.98**
w/o DP	76.48	79.33	71.28	72.64
w/o $\mathcal{L}_{\text{se}}$	80.96	81.23	76.02	74.80
w/o LAS	59.45	61.40	54.26	57.03

Table 6: Ablation study of our DPI method under the FG-1-PG-1 and FG-2-PG-1 settings of the CoNLL03 (Sang and De Meulder, 2003) dataset and the FG-2-PG-2 and FG-8-PG-2 settings of the I2B2 (Murphy et al., 2010) dataset. The average Micro-F1 score is reported.

6 Conclusion

In this work, we introduce the dynamic prefix method and formalize INER as a seq2seq generation task. By employing the dynamic prefix based on seq2seq generation framework, our method retains task-invariant capabilities and preserves task-specific knowledge in INER. Additionally, we integrate the generation-based label augmentation strategy and self-entropy loss to achieve a refined equilibrium between stability and plasticity. Empirical experiments on CoNLL03 and I2B2 datasets on INER benchmark demonstrate the effectiveness of our proposed method. We further evaluate our method in low-resource settings, and the results indicate the robustness and practicality of our method in more realistic scenarios with limited training data.

7 Limitations

The limitations of this work include: (1) More complex NER problems are not considered in this work, such as coarse-to-fine INER. Our approach is not designed to address the problem that a new entity type might be entailed in an old entity type, for example, "Doctor" emerging after "Person". Additionally, while our seq2seq generation framework is capable of addressing nested or discontinuous NER problems, we do not evaluate its performance on nested or discontinuous NER datasets due to the absence of suitable split algorithms for the incremental setting. (2) Our proposed label augmentation strategy relies on the old model to predict "pseudo" entity types, which may lead to error propagation. More refined label augmentation strategies will be explored in our future work.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 144–161. Springer.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tianqi Chen, Ian J. Goodfellow, and Jonathon Shlens. 2016. Net2net: Accelerating learning via knowledge transfer. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Xiang Chen, Lei Li, Qiaoshuo Fei, Ningyu Zhang, Chuanqi Tan, Yong Jiang, Fei Huang, and Huajun Chen. 2023. One model for all domains: collaborative domain-prefix tuning for cross-domain ner. *arXiv preprint arXiv:2301.10410*.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13122–13131.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, pages 86–102.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a Unified Classifier Incrementally via Rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 831–839.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *CoRR*, abs/2211.12701.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6467–6476.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Ruotian Ma, Xuanting Chen, Zhang Lin, Xin Zhou, Junzhe Wang, Tao Gui, Qi Zhang, Xiang Gao, and Yun Wen Chen. 2023. Learning "o" helps for learning more: Handling the unlabeled entity problem for class-incremental NER. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5959–5979. Association for Computational Linguistics.

Xinyin Ma, Yongliang Shen, Gongfan Fang, Chen Chen, Chenghao Jia, and Weiming Lu. 2020. Adversarial

self-supervised data-free distillation for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6182–6192. Association for Computational Linguistics.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11208 of *Lecture Notes in Computer Science*, pages 72–88. Springer.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13570–13577.

Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. 2010. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *CoRR*, abs/1606.04671.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogério Feris, and Zsolt Kira. 2023. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 11909–11919. IEEE.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. 2022a. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, volume 13686 of *Lecture Notes in Computer Science*, pages 631–648. Springer.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. 2022b. Learning to prompt for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 139–149. IEEE.

Yu Xia, Quan Wang, Yajuan Lyu, Yong Zhu, Wenhao Wu, Sujian Li, and Dai Dai. 2022. Learn and review: Enhancing continual named entity recognition via reviewing synthetic samples. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2291–2300. Association for Computational Linguistics.

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 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.

Duzhen Zhang, Hongliu Li, Wei Cong, Rongtao Xu, Jiahua Dong, and Xiuyi Chen. 2023. Task relation distillation and prototypical pseudo label for incremental named entity recognition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3319–3329.

Junhao Zheng, Zhanxian Liang, Haibin Chen, and Qianli Ma. 2022. Distilling Causal Effect from Miscellaneous Other-Class for Continual Named Entity Recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

A Appendix

A.1 Implementation Details

The model is implemented in the PyTorch framework on top of the T5 Huggingface implementation. Consistent with RDP, we train the model for 20 epochs if PG=2, and 10 epochs otherwise. The learning rate, batch size, prompt length and prompt hidden dim are set to 7e-5, 32, 10, and 1024, respectively. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU with 24GB of memory, and each experiment is run 5 times to ensure statistical robustness.