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 sequence labeling approaches to 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 formalize INER as a unified seq2seq generation task and propose a parameter-efficient dynamic prefix method. By employing the **D**ynamic **P**refix as a task **I**nstructor (DPI) 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 introduce a generative label augmentation strategy with dual optimization objectives including a self-entropy loss and a task-aware similarity loss to enable optimal balance between stability and plasticity. Through extensive empirical evaluation on standard NER benchmarks, we demonstrate that our approach significantly outperforms existing methods, achieving up to 13.6% improvement in low-resource scenarios while maintaining strong performance on previously learned entity types.

1 Introduction

Named Entity Recognition (NER) is a fundamental problem in information extraction. Traditional NER systems typically require extensive annotated training data encompassing all predefined entity types. However, as new entity types emerge, retraining the entire model becomes impractical. Furthermore, obtaining sufficient supervised training data 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.

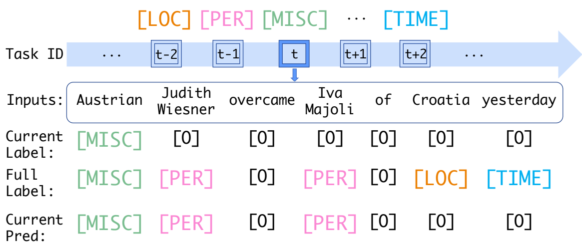

Figure 1: Challenges in INER. At the current incremental step t, the data is only annotated with the current entity type [MISC], while previous entity types [LOC] and [PER] are annotated with [O]. [TIME] is a future entity type. "Current Pred" indicates that the model forgets previous entity type [LOC]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

While incremental learning aims to mirror human capability in continuously acquiring knowledge (Ke and Liu, 2022), it faces the challenge of catastrophic forgetting (McCloskey and Cohen, 1989), where models lose previously acquired knowledge while learning new tasks. This is particularly problematic in NER, where information about previous and future entity types is absent during current learning steps. Ma et al. (2023) identify that most INER errors arise from confusion between pre-defined entities and non-entities ("O"). As shown in Figure 1, the model that successfully learned to recognize "PER" (person) and "LOC" (location) in one step would be trained to annotate "PER" or "LOC" as "O" in the 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.

Training directly on new data will exacerbate the background shift (Zhang et al., 2023) problem, where old and future entity types are mislabeled as the non-entity type. This results in a significant performance drop on historical entities. We validate the problem through experiments comparing three training paradigms (Figure 2). Multi-task learning (upperbound, green line) preserves all annotations, naive fine-tuning (blue line) shows catastrophic forgetting with F1 score plunging, while continual learning methods (orange line) maintain stable performance close to the upperbound.

Existing 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 entity types to individual tokens. This paradigm presents three critical challenges: 1. Structural Inflexibility: Adding new entity types requires expanding the classification layer and retraining the entire model, even when attempting to preserve existing weights. This architectural modification inevitably interferes with previously learned knowledge. 2. Parameter Inefficiency: The need to update both the encoder and classification layer parameters leads to significant computational overhead and increases the risk of catastrophic forgetting. 3. Limited Entity Modeling: The token-level classification paradigm struggles with complex scenarios such as nested or overlapping entities, often requiring specialized architectural modifications (Yan et al., 2021).

Motivated by these challenges, in this paper, we formalize INER as a seq2seq generation task, which aligns well with the generative nature of NER and facilitates prompt tuning more intuitively. Our proposed method employs a parameter-efficient dynamic prefix strategy tailored for incremental learning in INER. By dynamically assigning separate and task-specific prefixes as instructors during the incremental process, our model inspires the model to acquire new knowledge while retaining old prefixes to maintain stability. This structure inherently enables knowledge separation, where each prefix functions as a modular "expert" encoding specific entity-type patterns. During incremental learning, new prefixes are dynamically appended as lightweight instructors, while old prefixes remain intact, enabling parallel knowledge acquisition and retention. This approach is particularly effective for INER for two reasons: 1. Natural Entity Handling: The generative framework naturally supports complex scenarios such as overlapping entities by decoding entity spans autoregressively, overcoming the structural limitations of sequence labeling. 2. Efficient Knowledge Separation: The decoupled prefix architecture avoids overwriting shared parameters or expanding classification layers, ensuring smooth adaptation to new entity types.

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. At each incremental step, we expand the prefix set while keeping new prefixes as the only trainable parameters (approximately 0.1% of the base model). All prefixes are pluggable and require no modifications to the base model. 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. We further enhance the framework with a generation-based label augmentation strategy with a self-entropy loss and a task-aware similarity loss to achieve a more refined equilibrium between stability and plasticity.

Our main contributions are:

- We propose a dynamic prefix method to retain task-invariant capabilities and preserve task-specific knowledge in INER, requiring updates to only 0.1% of model parameters.
- We propose a generation-based label augmentation strategy with a self-entropy loss and a task-aware similarity loss, achieving an equilibrium between stability and plasticity.
- Comprehensive empirical validation shows significant improvements over existing methods, particularly in low-resource scenarios, while using orders of magnitude fewer parameters than traditional sequence labeling INER approaches.

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

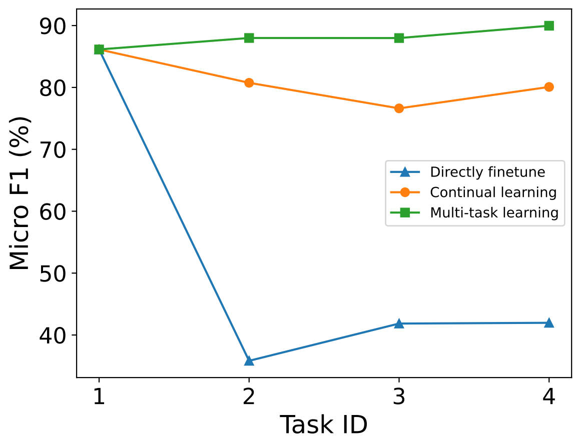

Figure 2: An illustration of catastrophic forgetting. We conduct the comparison with three different settings on the CoNLL03 (Sang and De Meulder, 2003) dataset.

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

2.2 Prompt Tuning in Continual Learning

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 does 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 the performance not only 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 contain-

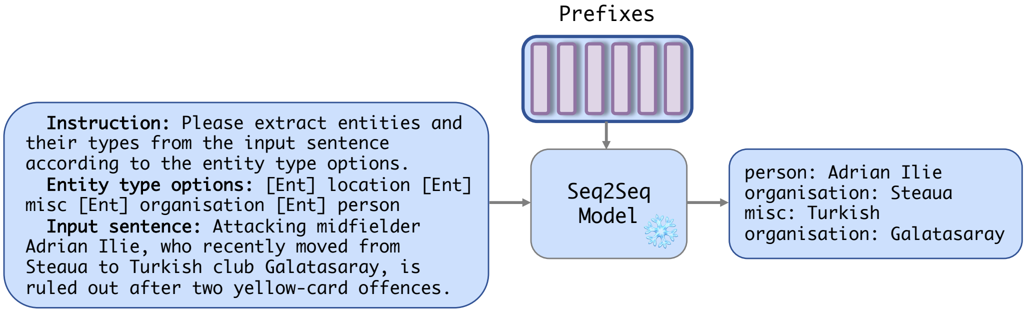

Figure 3: An illustration of the unified seq2seq approach for NER.

ing 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 $X_t^j = \{x_t^{j,1}, \dots, x_t^{j,l}\}$ (with l as the sequence length) and $Y_t^j = \{y_t^{j,1}, \dots, y_t^{j,l}\}, y_t^{j,k} \in \mathcal{C}_t^{new}(k = 1, \dots, l)$ 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:

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

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 ($\theta_1, \dots, \theta_{t-1}$) frozen. The objective of our dynamic prefix approach at step t becomes:

$$\max_{\theta_t} \sum_{(x,y)\in\mathcal{T}_t} \log p(y|x, \phi, \theta_1, \dots, \theta_t) \tag{3}$$

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:

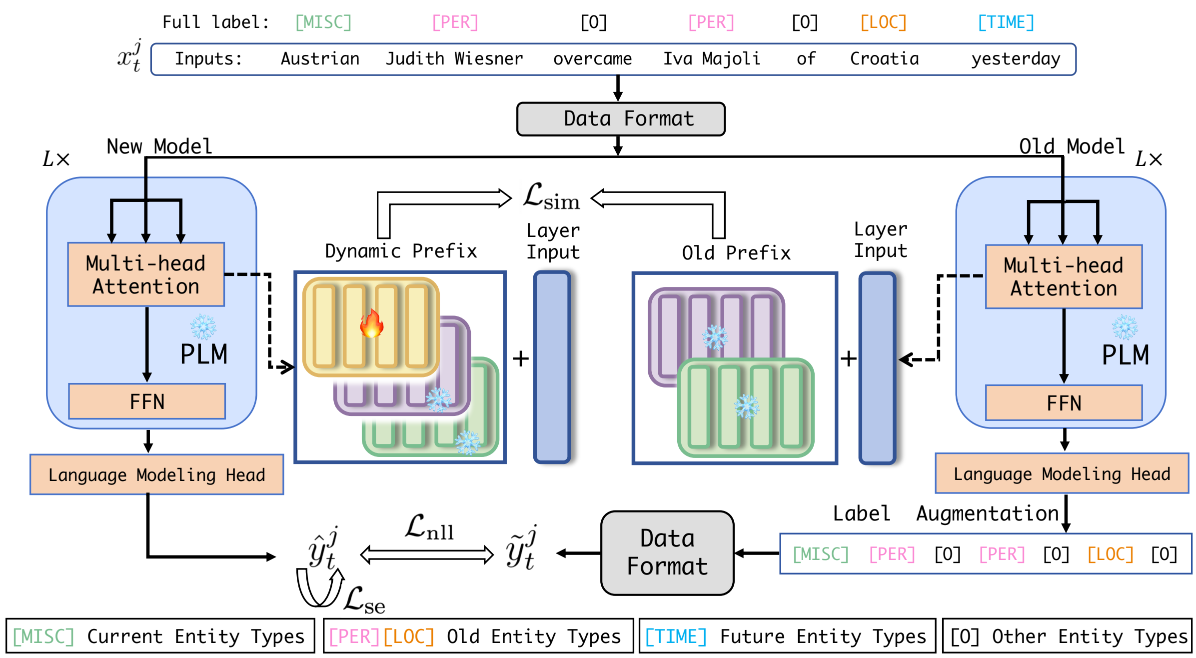

Figure 4: The overall architecture of our proposed DPI for INER. Here "+" denotes the concatenation operation.

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

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 $\mathcal{T}_t$.

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

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

3.4 Equilibrium Between Stability and Plasticity

In this section, we introduce the strategy employed to achieve a balance between stability and plasticity of INER.

Label Augmentation Strategy. 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 k^{th} token of the j^{th} input, we employ the strategy as follows:

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

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 $\mathcal{D}_t'^{tr} = \{X_t^j, \tilde{Y}_t^j\}_{j=1}^n$ for the current step t. The label augmentation strategy is expected to enhance the stability of our model. Then the Equation (6) can be formulated as follows:

$$\mathcal{L}_{\text{nll}}(\theta_t) = -\sum_{(x,y)\in D_t'^{tr}} \log p(y|[\mathbf{P}_t,\ldots,\mathbf{P}_1,x],\phi,\theta_1,\ldots,\theta_t) \tag{9}$$

Self-entropy Loss. To further extend the model's plasticity, we introduce a self-entropy loss to encourage the model to make more confident predictions. We minimize the self-entropy loss to promote the model's confidence in learning the new entity types:

$$\mathcal{L}_{\text{se}} = -\frac{1}{l}\sum_{k=1}^{l} \hat{y}^k \log \hat{y}^k \tag{10}$$

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

Task-aware Similarity Loss. To maintain distinction across tasks while enabling effective knowledge transfer, we introduce a dynamic regularization mechanism for prefix parameter optimization. The key insight is that tasks with similar entity types require more careful parameter separation to prevent interference. We quantify this through task-level semantic similarity, computed from entity type representations for simplicity. Specifically, we first obtain the representation $\mathbf{c}_i$ of the entity type definition c_i (as shown in Appendix A.2).

$$\mathbf{c}_i = \text{Encoder}(c_i) \in \mathbb{R}^d \tag{11}$$

For a task t containing K entity types, we compute its task-level semantic representation $\mathbf{T}_t$ by aggregating individual entity type embeddings:

$$\mathbf{T}_t = \frac{1}{K}\sum_{i=1}^{K} \mathbf{c}_i \tag{12}$$

Then we minimize the task-aware similarity loss to adaptively regulate the optimization of new prefix parameters based on semantic overlap with previous tasks.

$$\mathcal{L}_{\text{sim}} = \sum_{i=1}^{t-1} \max(0, \cos(\mathbf{T}_t, \mathbf{T}_i))\text{Sim}(\mathbf{P}_t, \mathbf{P}_i)^2 \tag{13}$$

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

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{nll}} + \lambda_1\mathcal{L}_{\text{se}} + \lambda_2\mathcal{L}_{\text{sim}} \tag{14}$$

4 Experiment

4.1 Experimental Settings

Datasets. We conduct experiments on three widely used NER dataset: CoNLL03 (Sang and De Meulder, 2003), I2B2 (Murphy et al., 2010) and OntoNotes5 (Hovy et al., 2006) for evaluating the effectiveness of our method. The dataset statistics are shown in Table 7 in Appendix A.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.

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 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 B.

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 baselines are directly cited from CFNER (Zheng et al., 2022).

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

Method \ Task ID	Trainable Param.	t=1 [LOC]	t=2 +[MISC]	t=3 +[ORG]	t=4 +[PER]	Avg.
Directly Fine-tune	~0.1% of 220M	86.14	35.83	41.85	41.97	51.45
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)	~100% of 110M	85.96	73.85	62.81	73.78	74.15
ExtendNER (Monaikul et al., 2021)	~100% of 110M	85.96	74.42	69.27	75.78	76.36
CFNER (Zheng et al., 2022)	~100% of 110M	85.96	80.63	76.10	80.95	80.91
RDP* (Zhang et al., 2023)	~100% of 110M	84.53	77.31	76.67	79.22	79.43
DPI (Ours)	~0.1% of 220M	**86.14**	**82.10**	**76.81**	**81.46**	**81.63**

Table 1: Main results of the proposed method and baselines under the FG-1-PG-1 setting of the CoNLL03 dataset (Sang and De Meulder, 2003). Micro-F1 score is reported. * represents results from the provided code. Other baseline results are directly cited from CFNER (Zheng et al., 2022).

Method	Trainable Param.	t=1	t=2	t=3	t=4	t=5	Avg.
PODNet (Douillard et al., 2020)	~100% of 110M	89.53	28.50	22.89	21.86	18.32	36.22
LUCIR (Hou et al., 2019)	~100% of 110M	90.23	72.00	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)	~100% of 110M	89.39	70.29	64.10	62.01	59.58	69.07
RDP* (Zhang et al., 2023)	~100% of 110M	90.94	77.86	69.16	63.95	53.36	71.05
DPI (Ours)	~0.1% of 220M	**91.43**	**84.98**	**75.92**	**72.51**	**72.08**	**79.38**

Table 2: 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 the provided code. Other baseline results are directly cited from CFNER (Zheng et al., 2022).

5 Results and Discussion

5.1 Main Results

We conduct experiments under INER settings and present the quantitative task-wise performance compared to the baselines.

As shown in Table 1, 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 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 I2B2 (Murphy et al., 2010) and Ontonotes (Hovy et al., 2006) under FG-8-PG-2 are summarized in Table 2 and Table 3, demonstrating an improvement of approximately 7.5% and 0.5% respectively compared to RDP.

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. Figure 5 indicates that our method outperforms significantly over 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.

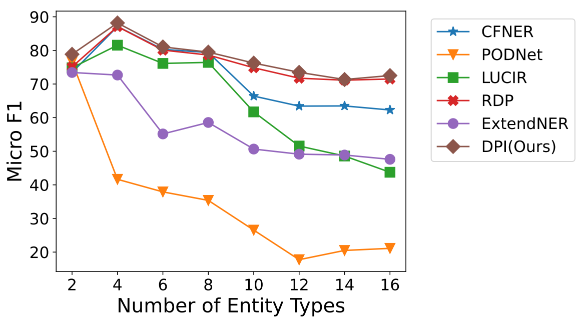

Figure 5: Task-wise results on the I2B2 (Murphy et al., 2010) dataset.

5.2 Low-resource Settings

Due to concerns related to data privacy and scarcity in realistic applications, INER often encounters

Method	Trainable Param.	t=1	t=2	t=3	t=4	t=5	t=6	Avg.
PODNet (Douillard et al., 2020)		82.52	29.44	23.42	32.19	29.36	27.33	37.38
LUCIR (Hou et al., 2019)		82.67	78.80	75.43	76.14	75.03	68.94	76.17
ExtendNER* (Monaikul et al., 2021)	~100% of 110M	82.37	79.56	75.24	76.93	76.40	73.36	77.31
CFNER* (Zheng et al., 2022)		82.37	82.10	79.16	80.51	79.30	76.63	80.01
RDP* (Zhang et al., 2023)		85.01	83.68	82.08	**83.26**	**82.43**	79.1	82.59
DPI (Ours)	~0.1% of 220M	**86.13**	**84.21**	**83.45**	83.07	81.49	**80.28**	**83.11**

Table 3: Comparison with baselines under the FG-8-PG-2 setting of the Ontonotes5 dataset (Hovy et al., 2006). Micro-F1 score is reported. * represents results from the provided code. Other baseline results are directly cited from CFNER (Zheng et al., 2022).

low-resource scenarios. To further investigate the effectiveness of our method regarding the data scale, we conduct experiments on various datasets with low-resource settings. We report the results on the OntoNotes5 (Hovy et al., 2006) and the CoNLL03 (Sang and De Meulder, 2003) dataset in Table 4 and Table 5, respectively. 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 a low-resource scenario with only 10% of the data available, our DPI method improves over RDP by approximately 1.8% and 13.6% on OntoNotes5 and ConLL03, respectively. In a more stringent low-resource scenario, our method also outperforms RDP by approximately 3.7% and 12.2%. In comparison, our approach maintains the ability to identify entities effectively, by fine-tuning significantly fewer parameters at each step, and effectively capturing the patterns of different entity types in low-resource scenarios.

Rate	Method	t=1	t=2	t=3	t=4	t=5	t=6	Avg.
10%	DPI (Ours)	79.53	75.54	72.02	76.44	72.61	70.59	**74.46**
	RDP	78.50	74.79	70.92	73.43	70.10	68.25	72.67
5%	DPI (Ours)	72.79	70.33	65.81	66.58	66.09	65.49	**67.85**
	RDP	68.70	62.01	62.28	65.32	65.17	61.50	64.16

Table 4: Performance in low-resource conditions on the OntoNotes5 (Hovy et al., 2006) dataset under the FG-8-PG-2 INER setting.

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-8-PG-2 INER setting.

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 prefixes with fixed size are trained throughout the incremental process. The results indicate that prefixes with fixed size lack the continual ability, which is exacerbated with more incremental steps. DPI w/o LAS means no label augmentation strategy is employed. By employing LAS and introducing self-entropy loss and task-aware similarity loss, we further achieve an equilibrium between stability and plasticity.

Method	CoNLL03		I2B2	
	FG-1-PG-1	FG-2-PG-1	FG-2-PG-2	FG-8-PG-2
DPI	**81.63**	**83.10**	**80.16**	**79.38**
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 $\mathcal{L}_{\text{sim}}$	81.19	82.70	78.82	78.58
w/o LAS	59.45	61.40	54.26	57.03

Table 6: Ablation study of our DPI method. The average Micro-F1 score is reported.

6 Conclusion

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

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.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

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 Dataset

A.1 Dataset Statistics

The dataset statistics are shown in Table 7.

Dataset	# Entity Type	# Sample	Entity Type
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
OntoNotes5	18	77k	CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART

Table 7: Statistics of the NER datasets CoNLL03 (Sang and De Meulder, 2003), I2B2 (Murphy et al., 2010) and OntoNotes5 (Hovy et al., 2006).

A.2 Entity Type Definition

Entity types and their definitions are illustrated in Table 8.

B 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, prompt hidden dim, λ_1 and λ_2 is set to 7e-5, 32, 10, 1024, 0.1 and 0.2 respectively. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU with 24GB of memory.

C Additional Experimental Results

C.1 Visualization of Entity Type Similarity

Figure 6, Figure 7 and Figure 8 show the entity type similarity on the Ontonotes5 (Hovy et al., 2006), the I2B2 (Murphy et al., 2010) and the CoNLL03 (Sang and De Meulder, 2003) dataset.

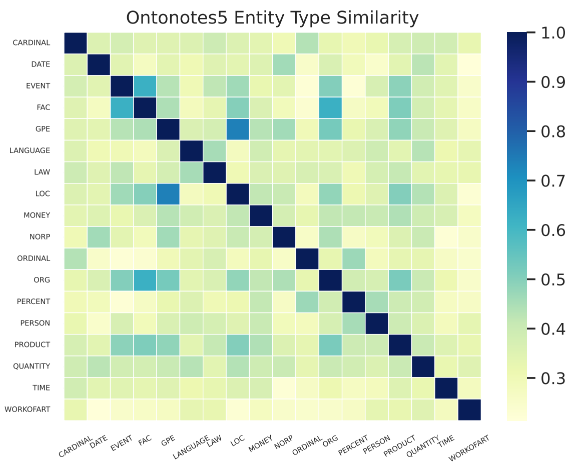

Figure 6: Entity type similarity of the Ontonotes5 (Hovy et al., 2006) dataset.

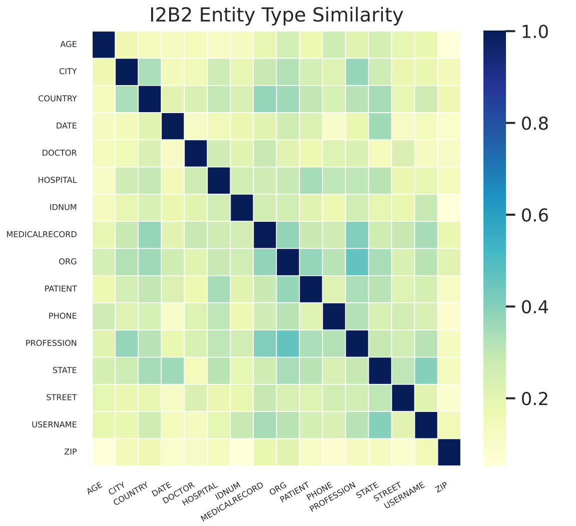

Figure 7: Entity type similarity of the I2B2 (Murphy et al., 2010) dataset.

C.2 Category-wise Results

To illustrate the performance variation of a single category throughout the incremental process, we present the results of four categories (“DATE”, “EVENT”, “GPE”, “LAW”) at step 1 in Figure 9, under the FG-8-PG-2 setting of the Ontonotes5 (Hovy et al., 2006) dataset.

C.3 Low-resource Results

The low-resource result of the I2B2 (Murphy et al., 2010) dataset is shown in Table 9. On the I2B2 dataset, RDP consistently fails to recognize almost all entities at every step. A possible reason is that it fine-tunes nearly all parameters during the incremental process, which hampers its ability to extract useful information when training data is limited.

Entity type	Definition
CARDINAL	Numerals that do not fall under another type.
DATE	Absolute or relative dates or periods.
EVENT	Named hurricanes, battles, wars, sports events, etc.
FAC	Facility (Buildings, airports, highways, bridges, etc.).
GPE	Geopolitical entities: countries, cities, states.
LANGUAGE	Any named language.
LAW	Named documents made into laws.
LOC	Locations excluding geopolitical entities, mountain ranges, bodies of water.
MONEY	Monetary values, including currency units.
NORP	Nationalities or religious or political groups.
ORDINAL	Ordinal numbers like "first", "second".
ORG	Organizations (Companies, agencies, institutions, etc.)
PERCENT	Percentage values (including "%").
PERSON	Person, including fictional characters.
PRODUCT	Commercial products (Vehicles, weapons, foods; excludes services).
QUANTITY	Measurements, as of weight or distance.
TIME	Sub-day time expressions.
WORK_OF_ART	Creative works.

Table 8: Entity type and definition of the Ontonotes5 dataset.

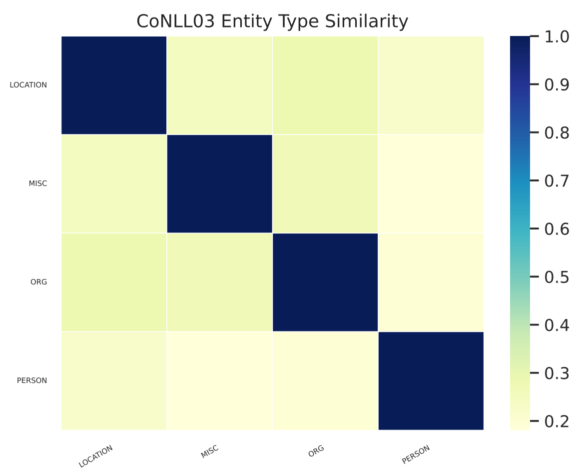

Figure 8: Entity type similarity of the CoNLL03 (Sang and De Meulder, 2003) dataset.

Rate	Method	t=1	t=2	t=3	t=4	t=5	Avg.
10%	DPI(Ours)	82.85	71.40	56.95	48.86	42.76	**60.56**
	RDP	1.21	0.04	0.32	0.28	0.23	0.42
5%	DPI(Ours)	77.43	65.28	49.12	40.65	38.16	**54.13**
	RDP	1.21	0.42	0.31	0.06	0.15	0.43

Table 9: Performance in low-resource conditions on the I2B2 (Murphy et al., 2010) dataset under the FG-8-PG-2 INER setting.

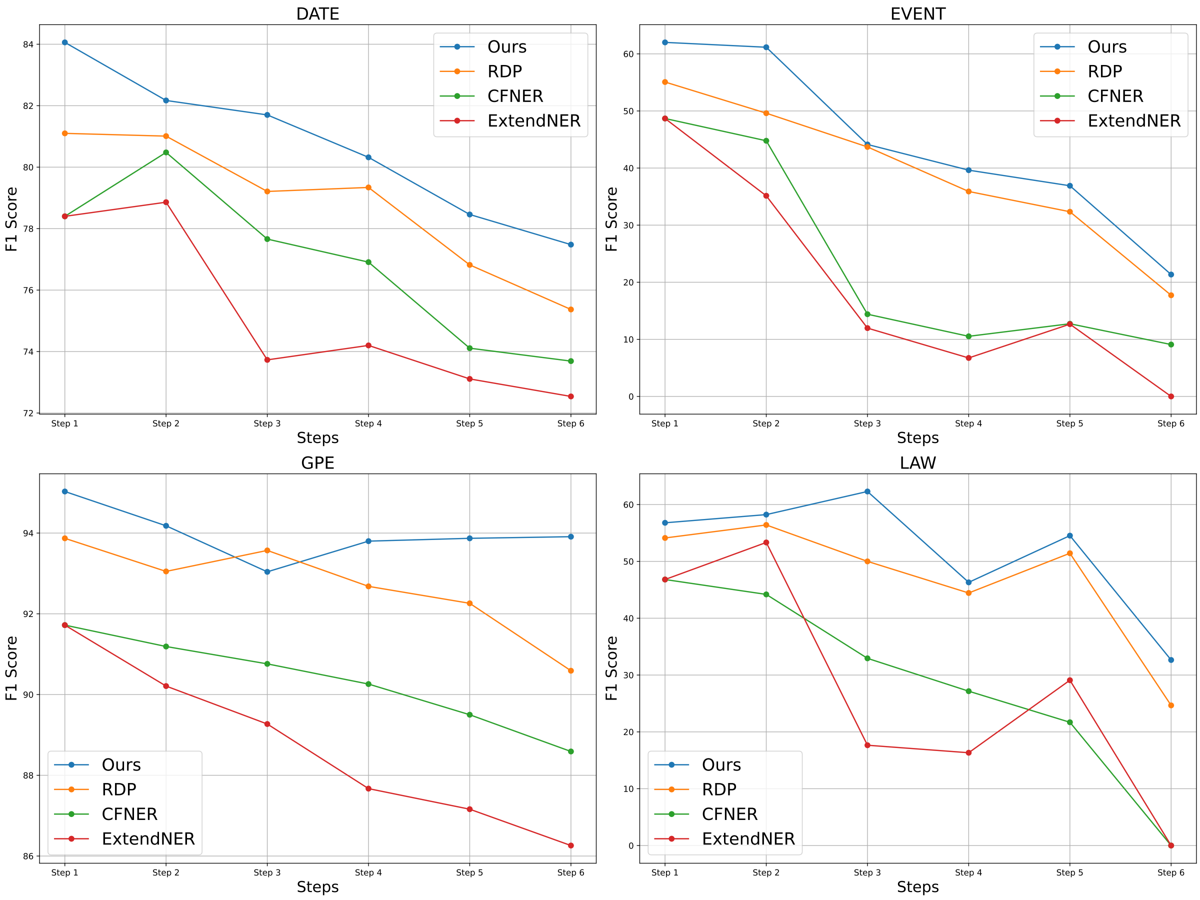

Figure 9: Category (Entity type) F1 of the Ontonotes5(Hovy et al., 2006) dataset under the FG-8-PG-2 setting.