Prototypical Replay with Old-class Focusing Knowledge Distillation for Incremental Named Entity Recognition

Zesheng Liu[1,2], **Qiannan Zhu**[4,5], **Cuiping Li**[1,2*], **Hong Chen**[1,3]
[1]School of Information, Renmin University of China, Beijing, China
[2]Key Laboratory of Data Engineering and Knowledge Engineering, MOE, China
[3]Engineering Research Center of Database and Business Intelligence, MOE, China
[4]School of Artificial Intelligence, Beijing Normal University, Beijing, China
[5]Engineering Research Center of Intelligent Technology and Educational Application, MOE, China
{lzs2022,licuiping,chong}@ruc.edu.cn, zhuqiannan@bnu.edu.cn

Abstract

Catastrophic forgetting is a key challenge in incremental named entity recognition (INER). Existing methods often address this issue through distillation-based approaches, which involve transferring previously learned knowledge from the old model to the new one. However, these methods may not fully equip the new model with an adequate understanding of the characteristics about old entity types, leading to confusion when classifying tokens associated with these entity types. To address this challenge, we propose a novel method called **P**rototypical Replay with **O**ld-class **F**ocusing Knowledge Distillation (**POF**) for INER. Our approach focuses on preserving the main characteristics of each previous entity type by storing compact prototypes and replaying them with appropriate frequency. This replay strategy makes the new model review the knowledge of old entity types while minimizing storage needs. Additionally, we introduce an old-class focusing knowledge distillation (OFKD) loss, which distills features only in old-class regions to maintain the quality of old-class prototypes and prevent ineffective prototypical replay while preserving sufficient plasticity for learning new entity types. We conduct experiments on three benchmark datasets (i.e., Few-NERD, I2B2 and OntoNotes5), and the results demonstrate that our method outperforms all previous state-of-the-art methods.

1 Introduction

Named Entity Recognition (NER) plays a crucial role in the field of Natural Language Processing (NLP), benefiting various downstream applications such as question answering (Li et al. 2019; Longpre et al. 2021), web search (Fetahu et al. 2021; Guo et al. 2009; Mokhtari et al. 2019), sentiment analysis (Yasavur et al. 2014; Yang et al. 2018), and more. NER involves extracting entities from unstructured text and categorizing them into predefined entity types or non-entity type (Ma and Hovy 2016). In traditional NER approaches, it is typically assumed that no new entity types will emerge during the testing phase, and all predefined entity types are learned simultaneously during training. However, real-world scenarios often demand continuous recognition of newly encountered entity types, namely Incremental Named Entity Recognition (INER) (Parisi et al. 2019; Thrun 1998). A challenge arises as traditional NER methods struggle to handle entity types not seen during training, necessitating the development of effective INER techniques (Monaikul et al. 2021; Xia et al. 2022; Zheng et al. 2022) that can incrementally learn NER models using training samples containing only new entity types.

INER must address a major challenge: catastrophic forgetting (Goodfellow et al. 2013; Kirkpatrick et al. 2017; McCloskey and Cohen 1989; Robins 1995), which it inherits from incremental learning. This occurs because fine-tuning only on new data can cause previously learned knowledge to be forgotten. Consequently, previous INER methods (Zhang et al. 2023a; Zheng et al. 2022; Monaikul et al. 2021; Qiu et al. 2024) have adopted distillation-based approaches to effectively tackle this challenge by transferring knowledge from the old model to the new one. However, these methods fall short in enabling the new model to fully understand the characteristics of the old entity types. Because these methods mainly rely on constraints on top-level model parameters and lack the process of relearning low-level features of old entity types. As the learning steps increase, the new model will gradually weaken its memory about old entity types, leading to confusion when classifying tokens from these entity types and resulting in numerous false positives. That would mean that a token of an old entity type may be incorrectly recognized as non-entity type, other old entity types, or new entity types. A simple solution is to save raw texts of all previous steps and replay them in subsequent steps, but this way requires a significant amount of storage.

To address aforementioned issue, we propose a novel method called **P**rototypical Replay with **O**ld-class **F**ocusing Knowledge Distillation (**POF**) for Incremental Named Entity Recognition (INER). Our method employs a prototypical replay (PR) strategy, which involves the storage of key information after each training step. Specifically, we retain a compact prototype for each entity type, generated from associated features, as well as relevant statistical data linked to these prototypes. This enables the PR strategy to replay previously learned entity types at strategic intervals during subsequent steps, facilitating a revisit of old entity types. In contrast to save and replay raw texts, PR strategy also significantly reduces storage requirements duo to it only stores

*Corresponding author.

very little information. Additionally, to complement our PR strategy and prevent catastrophic forgetting, it is crucial to maintain the features of old entity types during subsequent steps. Otherwise, the features of old entity types may drift away from the stored prototypes, compromising the effectiveness of the PR strategy. To tackle this problem, we introduce an old-class focusing knowledge distillation (OFKD) loss, which constrains the changes in the features of old entity types, while still allowing enough flexibility to learn new entity types.

In summary, this paper makes several key contributions:

- We propose a prototypical replay (PR) strategy, which stores compact prototypes with essential statistical data for old learned entity types, enabling the new model to revisit the knowledge of old entity types while learning new entity types, thus eliminating the classification confusion towards the tokens belonging to old entity types.
- We introduce an old-class focusing knowledge distillation (OFKD) loss which is designed to preserve the learned knowledge or features of old entity types, effectively integrating with our PR strategy while ensuring sufficient flexibility for learning new entity types.
- We conduct extensive experiments on eight INER settings of three datasets (i.e., Few-NERD (Ding et al. 2021), I2B2 (Murphy et al. 2010) and OntoNotes5 (Hovy et al. 2006)). Experimental results demonstrate that our POF achieves significant improvements over the previous state-of-the-art methods in INER.

2 Related Work

2.1 Incremental Learning

Incremental learning learns continuous tasks without compromising the performance of previous tasks (Dong et al. 2023a,b; Wang et al. 2024). We mainly classify existing incremental learning methods into three ways: replay-based, regularization-based, and dynamic architecture-based. The replay-based methods (Channappayya, Tamma et al. 2024; Liu et al. 2023; Lin et al. 2023; Qi, Zhao, and Li 2022) learn new tasks by integrating saved or generated old samples into the current training samples. Regularization-based methods impose constraints on network weights (Schwarz et al. 2018; Zenke, Poole, and Ganguli 2017), intermediary features (Zhai et al. 2024; Song et al. 2023; Han et al. 2023), or output probabilities (Zajac, Tuytelaars, and van de Ven 2023) to alleviate catastrophic forgetting. The methods based on dynamic architecture (Razdaibiedina et al. 2023; Gao et al. 2023) dynamically extends model architecture to learn new tasks.

2.2 Incremental Named Entity Recognition

INER seamlessly combines the principles of incremental learning with traditional NER approaches, resulting in a powerful framework that facilitates continuous learning and adaptation of the model to new entity types. ExtendNER (Monaikul et al. 2021) utilizes knowledge distillation by transferring output probabilities from the old model (teacher model) to the new model (student model). L&R (Xia et al. 2022) introduces a two-stage framework called learning and review: the learning phase uses knowledge extraction similar to ExtendNER, while the review phase uses replay-based methods to expand the current training set with composite samples of old entity types. CFNER (Zheng et al. 2022) introduces a causal framework for INER and uses the old model to identify non-entity type belonging to previous entity types to extract causal effects, and utilizing course learning strategies to mitigate recognition errors. DLD (Zhang et al. 2023b) divides the logits into positive and negative two parts for knowledge distillation. RDP (Zhang et al. 2023a) designs a task relation distillation scheme among different incremental steps and develop a prototypical pseudo-labeling strategy for classification. IS3 (Qiu et al. 2024) addresses both E2O and O2E semantic shifts in INER. However, they primarily rely on distillation-based methods to prevent catastrophic forgetting. Nonetheless, these methods may not fully enable the new model to grasp the nuances of old entity types, potentially leading to classification confusion in those cases.

In contrast, we design a strategy termed prototypical replay (PR) for INER. This approach aims to enable the new model to revisit the features of old entity types while learning new knowledge. Instead of replaying raw texts, PR only replays prototypes, which incurs a minimal storage cost and does not necessitate extra data. Additionally, we introduce an old-class focusing knowledge distillation (OFKD) loss to coordinate with our PR strategy.

3 Task Formulation

INER is a model training approach that aims to progressively learn and expand its knowledge of different entity types over a series of steps, denoted as $t = \{1, ..., T\}$. Each step has its own unique training set, denoted as D^t, which consists of (X^t, Y^t) pairs. Here, X^t represents an input token sequence with length of $|X^t|$, and Y^t represents the corresponding ground truth labels. It's important to note that the labels in Y^t only include the current entity types $\mathcal{E}^t$. Any labels for future entity types $\mathcal{E}^{t+1:T}$ or previous entity types $\mathcal{E}^{1:t-1}$ are collapsed into non-entity type e_o. It's worth mentioning that the datasets between different steps have no overlap, meaning that $\mathcal{E}^t \cap \mathcal{E}^{1:t-1} = \emptyset$ and $\mathcal{E}^t \cap \mathcal{E}^{t+1:T} = \emptyset$. At each step t ($t > 1$), the objective is to update a new model M^t that is capable of recognizing entities from all entity types seen up to that point, represented by $\bigcup_{i=1}^{t} \mathcal{E}^i$, and M^t consists of an encoder Φ_{Θ^t} with parameters Θ^t and a classifier Ψ_{Ω^t} with parameters Ω^t.

4 Methodology

This paper proposes a novel method called POF to address classification confusion of the new model towards old entity types in INER, shown in Figure 1. Our method includes three parts: Class-incremental Learning (CIL), Prototypical Replay Strategy (PR), and Old-class Focusing Knowledge Distillation (OFKD). Among them, CIL is the fundamental learning process of INER task, PR is to solve the above challenge, and OFKD is mainly to team up with our PR strategy.

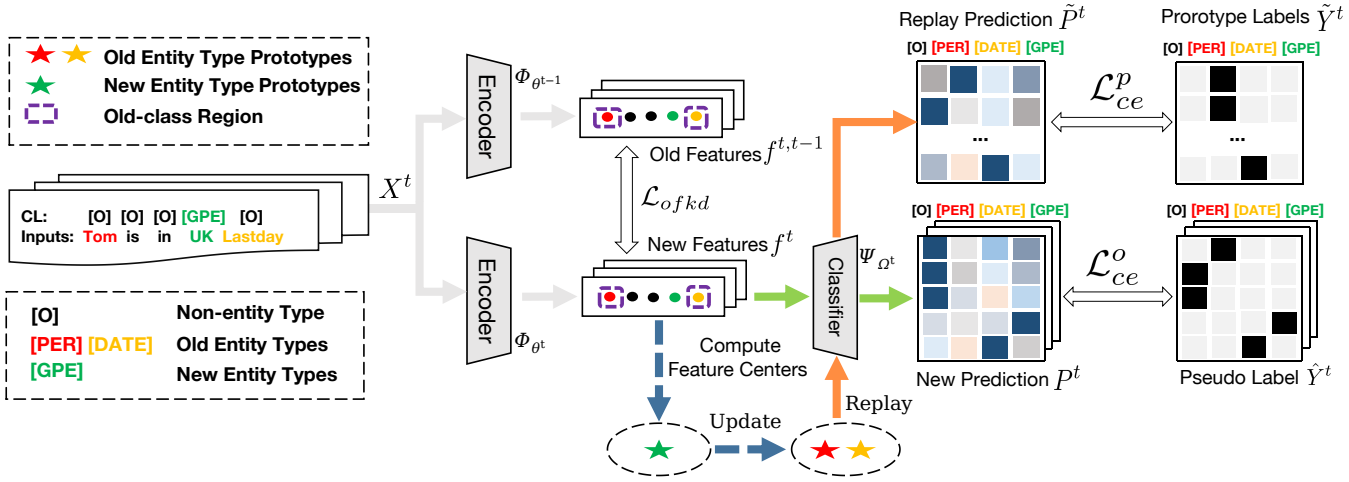

Figure 1: The overall framework of our POF, demonstrated by a simplified INER example. Firstly, old entity type prototypes have been saved in earlier steps. Nextly, pseudo label $\hat{Y}^t$ combined by old prediction and ground truth is to learn new entity types at current step t. Simultaneously, old entity type prototypes are replayed into the current classifier Ψ_{Ω^t}. Finally, we update parameters of the new model M^t with all cross entropy losses (e.g., $\mathcal{L}^o_{ce}$ and $\mathcal{L}^p_{ce}$) and old-class focusing knowledge distillation loss (i.e., $\mathcal{L}_{ofkd}$), and update the prototypes of new entity types to old ones.

And they will be introduced in detail separately in the following subsections.

4.1 Class-incremental Learning

At incremental step t, since the dataset D^t only contains the labels of the new entity types $\mathcal{E}^t$, if the model M^t is learning directly on D^t, the model will forget the recognition ability of the old entity types, resulting in catastrophic forgetting. Moreover, the learned old entity types are treated as non-entity type in D^t , which shifts the semantics of non-entity type, thus aggravating catastrophic forgetting. To address this problem, we adopt a naive pseudo-labeling which makes use of prediction of the old model in the last step. Specifically, given one sample pair (X^t, Y^t), we first feed X^t into the old model M^{t-1} and the new model M^t to generate the old prediction $P^{t,t-1}$ with old features $f^{t,t-1}$ and the new prediction P^t with new features f^t, respectively. Then, we re-label the tokens marked as e_o in the ground truth of current sample Y^t according to the entity types gained by $P^{t,t-1}$, and ultimately obtain the combined label $\hat{Y}^t$. Nextly, we use $\hat{Y}^t$ to supervise the current prediction P^t with the cross-entropy loss:

$$\mathcal{L}^o_{ce} = -\frac{1}{|X^t|}\sum_{i=1}^{|X^t|}\hat{Y}^t_i \log P^t_i \tag{1}$$

where P^t_i represents the probability score that the model predicts for i-th token in X^t.

4.2 Prototypical Replay Strategy

As previously mentioned, the new model struggles with classifying tokens from old entity types due to its limited knowledge or memory of them. To address this issue, we propose to create prototypes for each old entity type and replay these prototypes at an appropriate frequency during subsequent steps. This process makes the prototypes pass through the classifier alongside the features extracted from input texts, which allows the new model to revisit and reinforce its understanding for old entity types, thereby reducing its classification confusion. Thus, this strategy ensures that the new model can better distinguish between each old entity type and other entity types, ultimately improving the overall performance.

Specifically, after training at each step $\tau(\tau < t)$, we firstly calculate the number of tokens from each current entity type $e \in \mathcal{E}^\tau$ in the single-step training set D^τ to determine the replay frequency of subsequent steps:

$$\mathcal{N}_e = \sum_{i=1}^{|D^\tau|}\sum_{j=1}^{|X^\tau_i|}\mathbb{I}\{Y^\tau_{i,j} = e\} \tag{2}$$

where X^τ_i indicates the i-th token sequence in D^τ, j indicates the j-th token in X^τ_i, and $\mathbb{I}$ is the indicator function. Since each entity type is considered as a foreground class in a single step, $\mathcal{N}_e$ is the number of tokens that appear in entity type e running through the entire dataset, which can decide the frequency of replaying prototype of entity type e in subsequent steps.

Then, in order to construct replay samples for each entity type $e \in \mathcal{E}^\tau$ in future steps, we calculate the feature center of e to capture its main characteristics based on new features $f^\tau_{i,j}$ in current step τ:

$$\mu_e = \frac{\frac{1}{\mathcal{N}_e}\sum_{i=1}^{|D^\tau|}\sum_{j=1}^{|X^\tau_i|}(\hat{f}^\tau_{i,j} \odot \mathbb{I}\{Y^\tau_{i,j} = e\})}{\left\|\frac{1}{\mathcal{N}_e}\sum_{i=1}^{|D^\tau|}\sum_{j=1}^{|X^\tau_i|}(\hat{f}^\tau_{i,j} \odot \mathbb{I}\{Y^\tau_{i,j} = e\})\right\|_2} \tag{3}$$

where $\odot$ denotes the element-wise multiplication, $\|\cdot\|_2$ computes the L2-norm, and $\hat{f}^\tau_{i,j}$ refers to the L2-normalized fea-

ture vector given by $\hat{f}^{\tau}_{i,j} = \frac{f^{\tau}_{i,j}}{\|f^{\tau}_{i,j}\|_2}$. Meanwhile, μ_e indicates the mean direction of all feature vectors for e serving as a prototype that embodies the typical characteristic of e. However, relying solely on this prototype during the replay in subsequent steps may result in inadequately robust training outcomes, as a prototype only provides a single representation of entity type features without encompassing any variability. Therefore, we also compute the standard deviation of the feature vectors to introduce noise during the replay process. The calculation is given by:

$$\sigma_e = \sqrt{\frac{1}{\mathcal{N}_e}\sum_{i=1}^{|D^{\tau}|}\sum_{j=1}^{|X^{\tau}_i|}[(\hat{f}^{\tau}_{i,j} - \mu_e)^2 \odot \mathbb{I}\{Y^{\tau}_{i,j} = e\}]} \quad (4)$$

Additionally, since we remove the length information of feature vectors through L2-normalization, we need to restore it during replay. Hence, we also store the mean and standard deviation of the L2-norms of all features for entity type e, denoted as $\hat{\mu}_e$ and $\hat{\sigma}_e$, which are formulated by:

$$\hat{\mu}_e = \frac{1}{\mathcal{N}_e}\sum_{i=1}^{|D^{\tau}|}\sum_{j=1}^{|X^{\tau}_i|}(\left\|\hat{f}^{\tau}_{i,j}\right\|_2 \odot \mathbb{I}\{Y^{\tau}_{i,j} = e\}) \in \mathbb{R}^1 \quad (5)$$

$$\hat{\sigma}_e = \sqrt{\frac{1}{\mathcal{N}_e}\sum_{i=1}^{|D^{\tau}|}\sum_{j=1}^{|X^{\tau}_i|}(\left\|\hat{f}^{\tau}_{i,j}\right\|_2 - \hat{\mu}_e)^2 \odot \mathbb{I}\{Y^{\tau}_{i,j} = e\}} \in \mathbb{R}^1 \quad (6)$$

After obtaining n_e, μ_e, σ_e, $\hat{\mu}_e$ and $\hat{\sigma}_e$, robust replay can be implemented in subsequent steps. Specifically, in a future step t, the classifier is typically fed with features extracted solely from the input text. To ensure the new model reviews the knowledge of old entity types, we introduce replayed samples from these old entity type prototypes into the classifier. For an old entity type $e \in \mathcal{E}_{1:t-1}$, we represent one replayed sample using a random variable r_e, which is the product of two Gaussian-distributed random variables: the direction of the feature vector $\xi_e \sim \mathcal{G}(\mu_e, \sigma^2_e)$ and the L2-norm $\eta_e \sim \mathcal{G}(\hat{\mu}_e, \hat{\sigma}^2_e)$:

$$r_e = \xi_e \times \eta_e \quad (7)$$

Nextly, for each old entity type e, we replay $\mathcal{N}_e$ times for r_e per training epoch. Assuming each epoch consists of k iterations, we evenly distribute $\mathcal{N}_e$ among these iterations. This means r_e is sampled $\frac{\mathcal{N}_e}{k}$ times in each iteration. These random samples, denoted as $\mathcal{P}^t$, are then fed into the classifier Ψ_{Ω^t} to compute the total gradient for parameter updating with cross-entropy loss. Formally:

$$\mathcal{L}^p_{ce} = -\frac{1}{|\mathcal{P}^t|}\sum_{i=1}^{|\mathcal{P}^t|}\tilde{Y}^t_i \log \tilde{P}^t_i \quad (8)$$

where $\tilde{P}^t_i$ and $\tilde{Y}^t_i$ represent output probability and label corresponding to the i-th prototype vector in $\mathcal{P}^t$, respectively.

In this way, since each sampling of r_e produces different features and the distribution is controlled by the stored standard deviation, the new model can capture the ample latent characteristics of the old entity types to strengthen its memory of them and achieve robust discriminability. Consequently, we can rewrite the original loss in Eq.(1) by adding this loss to it, which is given by:

$$\mathcal{L}_{ce} = \mathcal{L}^o_{ce} + \alpha\mathcal{L}^p_{ce} \quad (9)$$

where α is a hyper-parameter to balance losses.

Datasets	#Entity Types	#Training Instances	# Test Instances
Few-NERD	66	131758	230025
I2B2	16	59376	41397
OntoNotes5	18	59922	23836

Table 1: The statistical information for each NER dataset.

4.3 Old-class Focusing Knowledge Distillation

The aforementioned PR strategy can enable the new model to revisit and retain knowledge of the old entity types, helping to prevent confusion of classification caused by faint memory. However, the success of this approach depends on the stability of the old-class features across different model training steps. If significant changes occur in these features between steps, discrepancies may arise between the features extracted by the current model and the replayed prototypes generated by the old model. In such cases, the PR strategy could become ineffective due to the outdated nature of the prototypes. Furthermore, since there are no constraints on parameter updates, the model may suffer from uncontrolled catastrophic forgetting of old knowledge. Consequently, our goal is to develop a strategy that maintains features of old entity types, cooperating with our PR strategy and resisting catastrophic forgetting. At the same time, we do not want the model to lose plasticity for learning new entity types.

By balancing above objectives, we propose an old-class focusing knowledge distillation (OFKD) loss, which focuses on distilling the features in the old-class regions. The OFKD loss is constructed to ensure that the consistency of feature representations in old-class regions is preserved. Simultaneously, features in other regions can be updated without constraints, ensuring that the flexibility to learn new entity types remains uncompromised. It achieves this by using the current ground truth to define the old-class regions and employing a constraint based on Mean Squared Error (MSE) in the distillation loss. This constraint helps maintain similarity measurement consistency through a hard loss function, such as L2-distance. The OFKD loss, designed around token features, is formulated as follows:

$$\mathcal{L}_{ofkd} = \frac{1}{N}\sum_{j=1}^{N} w_j * MSE(f^t_{i,j}, f^{t,t-1}_{i,j}) \quad (10)$$

where N represents the number of features, and

$$w_j = \begin{cases} 0, & \text{if} \quad \hat{Y}^t_{i,j} \in \mathcal{E}^t \quad or \quad \hat{Y}^t_{i,j} = e_o \\ 1, & \text{if} \quad \hat{Y}^t_{i,j} \in \mathcal{E}^{1:t-1} \end{cases} \quad (11)$$

Method	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_5$	$\mathcal{A}_6$	$\mathcal{A}_7$	$\mathcal{A}_8$	$\mathcal{A}_9$	$\mathcal{A}_{10}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$
PODNet (Douillard et al. 2020)	48.73	11.77	10.33	9.66	7.14	4.89	5.16	4.73	5.18	3.99	3.79	10.49
LUCIR (Hou et al. 2019)	48.73	46.21	43.23	37.76	25.47	20.31	24.12	26.91	23.33	21.12	18.59	30.52
ST (De Lange et al. 2019)	48.73	41.15	35.50	32.94	27.77	26.03	30.34	32.73	32.85	32.14	30.71	33.71
ExtendNER (Monaikul et al. 2021)	48.73	40.40	38.42	32.20	27.26	25.48	30.38	33.68	33.69	32.73	31.61	34.05
CFNER (Zheng et al. 2022)	48.73	50.45	46.33	47.38	37.74	33.65	36.40	39.33	40.09	40.15	39.76	41.82
DLD (Zhang et al. 2023b)	48.73	<u>52.00</u>	46.40	40.92	34.74	32.22	35.55	37.81	36.94	37.24	37.56	40.01
RDP (Zhang et al. 2023a)	48.73	50.36	<u>46.40</u>	<u>45.20</u>	<u>40.19</u>	<u>37.57</u>	<u>38.02</u>	<u>39.56</u>	<u>39.76</u>	<u>38.96</u>	<u>38.68</u>	<u>42.13</u>
IS3 (Qiu et al. 2024)	48.73	50.74	45.87	39.61	34.44	33.71	37.20	39.44	37.57	37.60	36.82	40.16
POF(Ours)	48.73	**52.36**	**51.31**	**50.93**	**50.28**	**46.52**	**47.59**	**48.77**	**48.59**	**48.21**	**48.41**	**49.25**
Imp.	-	-	-	-	-	-	-	-	-	-	⇑9.73	⇑7.12

Table 2: Comparisons with baselines on Few-NERD at each incremental step (total 11 steps). The **bold** denotes the highest result, and the **underline** denotes the second highest result.

Based on the refined label $\hat{Y}_i^t$, we set the distillation loss weight w_j to 1 in the old-class region and to 0 in both the new-class and non-entity type regions. Hence, the final overall loss of our POF is:

$$\mathcal{L} = \mathcal{L}_{ce} + \beta \mathcal{L}_{ofkd} \tag{12}$$

with a hyper-parameter β for balancing losses.

5 Experimental Setup

Datasets We conducted experiments on three benchmark datasets, namely Few-NERD (Ding et al. 2021), I2B2 (Murphy et al. 2010), and OntoNotes5 (Hovy et al. 2006). We summarized the statistical data of these datasets in Table 1. The data processing process is consistent with CFNER (Zheng et al. 2022).

Baseline Methods We compare POF with recent INER methods in Sec 2.2, namely ExtendNER, CFNER, DLD, RDP and IS3. Following CFNER, we also compare POF with incremental learning methods used in the field of computer vision, such as Self-Training (ST) (De Lange et al. 2019; Rosenberg, Hebert, and Schneiderman 2005), LUCIR (Hou et al. 2019), and PODNet (Douillard et al. 2020).

INER Settings We utilize FG entity types to train the base model, and PG entity types are introduced at each subsequent incremental learning step, denoted as FG-*a*-PG-*b*. Specifically, we only employ the FG-6-PG-6 setting for Few-NERD. And for I2B2 and OntoNotes5, we employ four INER settings: FG-4-PG-2, FG-8-PG-1, FG-8-PG-2, and FG-12-PG-1.

Implementation Details Following the baseline methods, we also adopt the "BIO" labeling scheme across all datasets. Our model utilizes a BERT-based encoder (Devlin et al. 2018) and employs a fully connected layer as the classifier. We use the PyTorch (Paszke et al. 2019) framework to implement the model, which is built on top of the Huggingface (Wolf et al. 2019) implementation. We train the model for 20 epochs if PG is 2, and 10 epochs otherwise. Hyper-parameters α and β are set to 1.0 and 2.0, respectively. Meanwhile, we set the batch size and learning rate to 8 and 4e-4, respectively. Also, we introduce distillation loss in ExtendNER to better overcome forgetting. All experiments were conducted on NVIDIA GeForce RTX 3090 with 24GB of memory.

Metrics Following IS3, we compute the final performance report using the Macro-F1 score of the last task $\mathcal{A}_T$ and the average Macro-F1 score across all tasks $\bar{\mathcal{A}}$. Additionally, we present a line chart for task performance comparison to provide a more detailed analysis.

6 Results and Analysis

6.1 Main Results

In order to verify the effectiveness of our POF under various INER settings, we conducted experiments on Few-NERD, I2B2 and OntoNotes5 three datasets. The step-wise results of Few-NERD under FG-6-PG-6 setting is shown in Table 2. The results obtained from the I2B2 and OntoNotes5 are shown in Table 3, and the step-wise Macro-F1 score comparisons under eight INER settings on these two datasets are shown in Figure 2.

As shown in Table 2, we can observe that our POF achieves the best result at each incremental step under the FG-6-PG-6 setting of Few-NERD and achieves 9.73% improvement in $\mathcal{A}_T$ and 7.12% improvement in $\bar{\mathcal{A}}$ compared to the baseline RDP of SOTA. Meanwhile, as depicted in the upper part of Table 3, our POF achieves improvements over the best results of previous SOTA baselines ranging from 0.34% to 7.20% in $\mathcal{A}_T$, and 0.41% to 5.46% in $\bar{\mathcal{A}}$, under four INER settings of I2B2. Similarly, in the lower part of Table 3, our POF achieves improvements over the best results of previous SOTA baselines ranging from 1.45% to 6.50% in $\mathcal{A}_T$, and 3.08% to 3.85% in $\bar{\mathcal{A}}$, under four INER settings of OntoNotes5. These results quantitatively demonstrate that our POF outperforms all baselines, as we replay old entity type prototypes when learning new entity types, eliminating classification confusion for the new model on tokens of old entity type and improving the overall performance.

Especially compared to two SOTA baselines RDP and IS3, our POF performs better because these two methods mainly preserve the memory of old entity types mainly through knowledge distillation. However, as the learning steps continue to increase, they still lose or weaken their memory about old entity types, resulting in classification confusion. In the meantime, we can avoid this problem by replaying old entity type prototypes as a forgetting compensation. In addition, although the old entity type prototypes were also used in IS3, this method simply uses the mean

Dataset	Method	FG-4-PG-2		FG-8-PG-1		FG-8-PG-2		FG-12-PG-1	
		$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$
I2B2	PODNet (Douillard et al. 2020)	7.32	25.48	8.81	26.87	10.47	30.56	22.67	45.38
	LUCIR (Hou et al. 2019)	29.02	43.23	23.68	36.89	32.81	48.62	33.57	54.78
	ST (De Lange et al. 2019)	29.75	40.01	19.36	26.90	22.61	35.80	11.18	33.94
	ExtendNER (Monaikul et al. 2021)	28.53	38.52	18.99	27.78	25.65	39.16	10.27	32.03
	CFNER (Zheng et al. 2022)	32.82	47.39	19.77	37.18	37.18	49.60	31.50	48.52
	DLD (Zhang et al. 2023b)	43.93	51.46	28.47	38.41	38.76	50.61	36.68	55.49
	RDP (Zhang et al. 2023a)	44.56	55.21	50.53	62.73	58.50	64.92	56.25	67.03
	IS3 (Qiu et al. 2024)	45.97	55.88	52.24	60.25	56.06	63.33	69.14	72.74
	POF (Ours)	**46.31**	**56.29**	**59.44**	**66.21**	**65.49**	**70.38**	**72.78**	**75.70**
	Imp.	**⇑0.34**	**⇑0.41**	**⇑7.20**	**⇑3.48**	**⇑6.99**	**⇑5.46**	**⇑3.64**	**⇑2.96**
OntoNotes5	PODNet (Douillard et al. 2020)	16.46	23.73	11.06	21.32	15.50	27.56	6.54	33.29
	LUCIR (Hou et al. 2019)	49.17	53.44	35.55	48.51	50.47	58.25	37.22	57.20
	ST (De Lange et al. 2019)	52.47	52.84	43.38	49.43	46.29	54.02	48.55	60.72
	ExtendNER (Monaikul et al. 2021)	49.73	51.95	38.07	48.33	47.59	55.33	45.02	55.61
	CFNER (Zheng et al. 2022)	54.83	57.78	47.46	55.68	51.89	60.35	59.26	70.30
	DLD (Zhang et al. 2023b)	55.67	56.11	46.11	55.03	51.73	59.66	56.67	68.15
	RDP (Zhang et al. 2023a)	57.38	60.17	56.24	62.68	57.83	64.83	64.16	71.55
	IS3 (Qiu et al. 2024)	59.29	60.28	54.06	61.06	62.23	66.57	59.90	66.52
	POF (Ours)	**62.67**	**63.99**	**57.69**	**66.53**	**63.91**	**69.69**	**70.66**	**74.63**
	Imp.	**⇑3.38**	**⇑3.72**	**⇑1.45**	**⇑3.85**	**⇑1.68**	**⇑3.12**	**⇑6.50**	**⇑3.08**

Table 3: Comparisons with baselines on I2B2 and OntoNotes5. The **bold** denotes the highest result, and the **underline** denotes

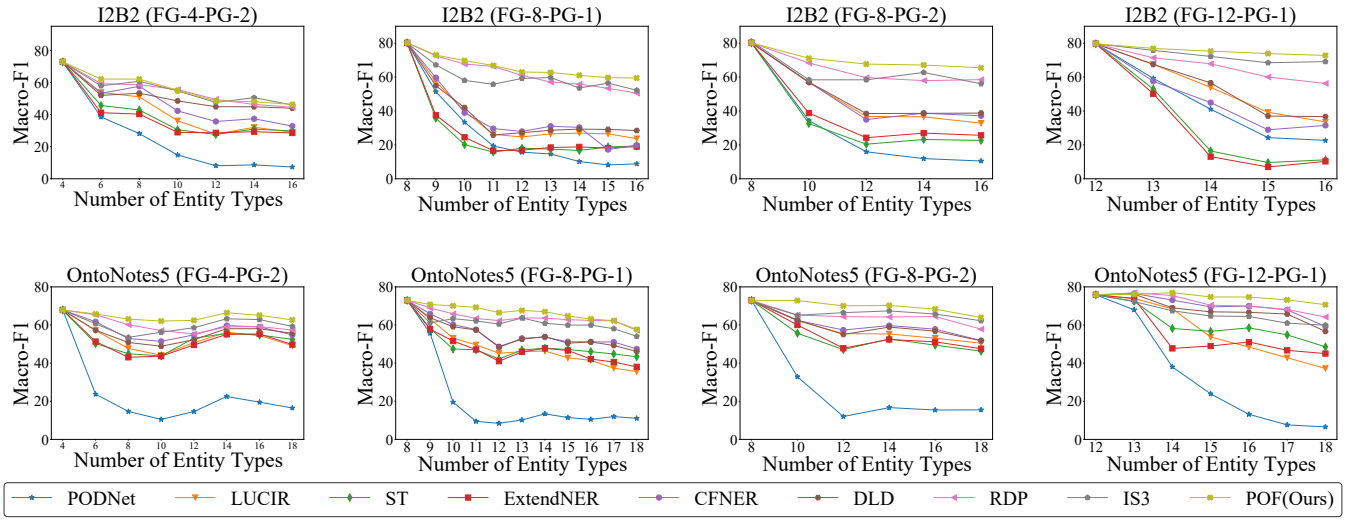

Figure 2: Comparison of the step-wise Macro-F1 on I2B2 and OntoNotes5 two datasets.

feature of each old entity type, lacking sufficient diversity to achieve the robust training outcome mentioned in Sec 4.2, so its discriminability is not as good as our POF.

Furthermore, as illustrated in Figure 2, our POF outperforms the INER baselines in step-wise Macro-F1 comparisons across the eight settings of I2B2 and OntoNotes5. These results qualitatively confirm the superiority and effectiveness of our POF in continuously learning new entity types compared to competitive baselines and the importance of reviewing the knowledge of old entity types through prototypical replay in our method.

PR	*OFKD*	I2B2 $\mathcal{A}_T$	I2B2 $\bar{\mathcal{A}}^{old}$	I2B2 $\bar{\mathcal{A}}^{new}$	I2B2 $\bar{\mathcal{A}}$	OntoNotes5 $\mathcal{A}_T$	OntoNotes5 $\bar{\mathcal{A}}^{old}$	OntoNotes5 $\bar{\mathcal{A}}^{new}$	OntoNotes5 $\bar{\mathcal{A}}$
✗	✓	55.01	66.06	57.27	67.98	64.45	73.47	53.64	72.57
✓	✗	66.75	70.10	73.32	72.07	67.77	73.72	53.85	72.81
✓	✓	**72.78**	**74.67**	**76.57**	**75.70**	**70.66**	**74.70**	**61.59**	**74.63**

Table 4: The ablation study of our POF under the FG-12-PG-1 setting of the I2B2 and OntoNotes5 datasets. For a more detailed analysis, we separately calculate the mean performance on old, new, and all entity types. Compared to our POF, all ablation variants significantly reduced the performance of INER, validating the importance of all components working together to solve INER.

Sentence	Global	plans	to	introduce	new	services	,	including	car	rental	and	plane	ticket	reservations	and	confirmation	.
RDP PL	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
IS3 PL	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
POF PL	B-ORG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
GL	B-ORG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

Sentence	Has	Iraq	been	overshadowed	by	Katrina	and	Tom	Delay	and	Harriet	Meyers	and	the	Pakistan	quake	?
RDP PL	O	B-GPE	O	O	O	B-PER	O	B-PER	I-PER	O	B-PER	I-PER	O	O	B-GPE	O	O
IS3 PL	O	B-GPE	O	O	O	B-PER	O	B-PER	I-PER	O	B-PER	I-PER	O	O	B-GPE	O	O
POF PL	O	B-GPE	O	O	O	B-EVE	O	B-PER	I-PER	O	B-PER	I-PER	O	O	B-GPE	O	O
GL	O	B-GPE	O	O	O	B-EVE	O	B-PER	I-PER	O	B-PER	I-PER	O	O	B-GPE	O	O

Sentence	Dongguan	Hsu	Fu	Chi	Foods	,	which	sells	candy	,	cakes	and	can	fruit	in	the	mainland	domestic	market	,	is	another	good	example	.
RDP PL	B-PER	I-ORG	I-WOA	I-ORG	I-ORG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
IS3 PL	B-PER	I-PER	I-WOA	I-PRO	I-ORG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
POF PL	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
GL	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

Figure 3: Three real NER cases sampled from the OntoNotes5 test set. **PL** and **GL** denote the predicted labels and golden labels, respectively. **B-** and **I-** distinguish begin/inside of named entities. O, EVE, GPE, PRO, WOA, ORG and PER denote non-entity type, Event, Countries, Cities, or States, Product, Work of art, Organization, and Person, respectively. All prediction results are from the last task of the FG-8-PG-2 setting. These visualization NER cases qualitatively demonstrate the superiority and effectiveness of our POF method.

6.2 Ablation Study

Table 4 shows the ablation study results of our POF under the FG-12-PG-1 setting, evaluating its two core components: prototypical replay (PR) and old-class focusing knowledge distillation loss (OFKD). Each component is individually removed from the model to evaluate its contribution to overall performance.

Effectiveness of the PR The first row of Table 4 presents the results without PR. Compared to the complete model's results shown in the last row, it is clear that PR boosts performance by 17.77% in $\mathcal{A}_T$ and 7.72% in $\bar{\mathcal{A}}$ of I2B2, and 6.21% in $\mathcal{A}_T$ and 2.06% in $\bar{\mathcal{A}}$ of OntoNotes5. Although PR involves replaying prototypes of old entity types, it also enhances the average recognition accuracy of new entity types, demonstrating 19.30% improvement of I2B2 dataset, and 7.95% improvement of OntoNotes5 dataset. This result stems from PR providing new model with sufficient access to the features of old entity types, which mitigate classifier bias and lessen the chance of misclassifying. Thus, we can observe significant performance improvements in the recognition of both old and new entity types.

Effectiveness of the OFKD The significance of OFKD can be recognized when comparing the second and the last row of Table 3. After removing OFKD loss, the results experience some breakdown, with 6.03% $\mathcal{A}_T$ and 3.63% $\bar{\mathcal{A}}$ drop of I2B2, and 2.89% $\mathcal{A}_T$ and 1.82% $\bar{\mathcal{A}}$ drop of OntoNotes5 compared to the full model. Meanwhile, the average recognition rates of both new and old entity types in two datasets have also significantly decreased. So that OFKD is a crucial mechanism in our method which can be used to maintain prior features while flexibly learning new knowledge. Its absence not only reduces the effectiveness of PR, but may also lead to uncontrolled parameter updates, which may result in unexpected catastrophic forgetting. Therefore, the learning of both old and new entity types will be affected, thereby having a marked impact on the overall performance.

6.3 Case Study

Figure 3 utilizes the challenging FG-8-PG-2 setting and performs qualitative comparisons on OntoNotes5, demonstrating the superiority of our POF over previous state-of-the-art INER methods. A common bad case in other competitive baselines is the occurrence of false positive samples of old entity types. For instance, in the first example, token *Global* labeled as old entity type *Organization* is incorrectly classified as non-entity type. In the second example, token *Katrina* of the old entity type *Event* is erroneously recognized as another old entity type *Person*. In the last example, tokens belonging to old entity type *Organization* are misclassified as other old entity types *Person* and *Product*, as well as new entity type *Work of art*. These errors stem from inadequate representation and limited knowledge of old entity types during the training of new entity types at each incremental step, leading to classification confusion. In contrast, our POF effectively addresses these issues by providing access to prototypes of old entity types as augmented training samples for the classifier of new entity types.

7 Conclusion

In this paper, we present a novel INER method named POF, which addresses the classification confusion associated with old entity types due to the new model's limited memory about them. POF employs a key strategy known as prototypical replay (PR), enabling the new model to revisit the knowledge of old entity types, thereby eliminating classification confusion to reduce recognition errors from them. By storing and replaying prototypes instead of raw texts, POF can minimize storage costs. Moreover, we propose an old-class focusing knowledge distillation loss to complement the PR strategy while preserving the flexibility to learn new entity types. Our extensive comparisons with state-of-the-art INER methods and ablation study demonstrate the superiority of our method, as well as significance of each individual component within our approach.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U23A20299, U24B20144, 62172424, 62276270, 62322214, 62472038, 62437001), National Key Research Develop Plan (2023YFB4503600), Fundamental Research Funds for the Central Universities (2233100004), and Engineering Research Center of Intelligent Technology and Educational Application, Ministry of Education.

References

Channappayya, S.; Tamma, B. R.; et al. 2024. Augmented Memory Replay-based Continual Learning Approaches for Network Intrusion Detection. *Advances in Neural Information Processing Systems*, 36.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2(6): 2.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, N.; Xu, G.; Chen, Y.; Wang, X.; Han, X.; Xie, P.; Zheng, H.-T.; and Liu, Z. 2021. Few-NERD:A Few-shot Named Entity Recognition Dataset. In *ACL-IJCNLP*.

Dong, J.; Liang, W.; Cong, Y.; and Sun, G. 2023a. Heterogeneous forgetting compensation for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11742–11751.

Dong, J.; Zhang, D.; Cong, Y.; Cong, W.; Ding, H.; and Dai, D. 2023b. Federated incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3934–3943.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 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 16*, 86–102. Springer.

Fetahu, B.; Fang, A.; Rokhlenko, O.; and Malmasi, S. 2021. Gaze eer Enhanced Named Entity Recognition for Code-Mixed Web eries.

Gao, Q.; Zhao, C.; Sun, Y.; Xi, T.; Zhang, G.; Ghanem, B.; and Zhang, J. 2023. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11483–11493.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Guo, J.; Xu, G.; Cheng, X.; and Li, H. 2009. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 267–274.

Han, K.; Liu, Y.; Liew, J. H.; Ding, H.; Liu, J.; Wang, Y.; Tang, Y.; Yang, Y.; Feng, J.; Zhao, Y.; et al. 2023. Global knowledge calibration for fast open-vocabulary segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 797–807.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 831–839.

Hovy, E.; Marcus, M.; Palmer, M.; Ramshaw, L.; and Weischedel, R. 2006. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, 57–60.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; and Li, J. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1340–1350.

Lin, H.; Zhang, B.; Feng, S.; Li, X.; and Ye, Y. 2023. PCR: Proxy-based contrastive replay for online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24246–24255.

Liu, Y.; Cong, Y.; Goswami, D.; Liu, X.; and van de Weijer, J. 2023. Augmented box replay: Overcoming foreground shift for incremental object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11367–11377.

Longpre, S.; Perisetla, K.; Chen, A.; Ramesh, N.; DuBois, C.; and Singh, S. 2021. Entity-Based Knowledge Conflicts in Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7052–7063.

Ma, X.; and Hovy, E. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1064–1074.

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

Mokhtari, S.; Mahmoody, A.; Yankov, D.; and Xie, N. 2019. Tagging address queries in maps search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9547–9551.

Monaikul, N.; Castellucci, G.; Filice, S.; and Rokhlenko, O. 2021. Continual learning for named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13570–13577.

Murphy, S. N.; Weber, G. M.; Mendis, M.; Gainer, V. S.; Chueh, H. C.; Churchill, S. E.; and Kohane, I. S. 2010. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association : JAMIA*, 124–130.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural networks*, 113: 54–71.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qi, D.; Zhao, H.; and Li, S. 2022. Better Generative Replay for Continual Federated Learning. In *The Eleventh International Conference on Learning Representations*.

Qiu, S.; Zheng, J.; Liu, Z.; Luo, Y.; and Ma, Q. 2024. Incremental Sequence Labeling: A Tale of Two Shifts. In *Findings of the Association for Computational Linguistics: ACL 2024*, 777–791. Association for Computational Linguistics.

Razdaibiedina, A.; Mao, Y.; Hou, R.; Khabsa, M.; Lewis, M.; and Almahairi, A. 2023. Progressive Prompts: Continual Learning for Language Models. In *International Conference on Learning Representations*.

Robins, A. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146.

Rosenberg, C.; Hebert, M.; and Schneiderman, H. 2005. Semi-Supervised Self-Training of Object Detection Models. *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, 1: 29–36.

Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, 4528–4537. PMLR.

Song, Z.; Zhao, Y.; Shi, Y.; Peng, P.; Yuan, L.; and Tian, Y. 2023. Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 24183–24192.

Thrun, S. 1998. *Lifelong Learning Algorithms*, 181–209. Boston, MA: Springer US.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xia, Y.; Wang, Q.; Lyu, Y.; Zhu, Y.; Wu, W.; Li, S.; and Dai, D. 2022. Learn and review: Enhancing continual named entity recognition via reviewing synthetic samples. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2291–2300.

Yang, J.; Yang, R.; Wang, C.; and Xie, J. 2018. Multi-entity aspect-based sentiment analysis with context, entity and aspect memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yasavur, U.; Travieso, J.; Lisetti, C.; and Rishe, N. D. 2014. Sentiment analysis using dependency trees and named-entities. In *The Twenty-Seventh International Flairs Conference*.

Zajac, M.; Tuytelaars, T.; and van de Ven, G. M. 2023. Prediction Error-based Classification for Class-Incremental Learning. In *The Twelfth International Conference on Learning Representations*.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

Zhai, J.-T.; Liu, X.; Yu, L.; and Cheng, M.-M. 2024. Fine-Grained Knowledge Selection and Restoration for Non-exemplar Class Incremental Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6971–6978.

Zhang, D.; Li, H.; Cong, W.; Xu, R.; Dong, J.; and Chen, X. 2023a. 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*, 3319–3329.

Zhang, D.; Yu, Y.; Chen, F.; and Chen, X. 2023b. Decomposing logits distillation for incremental named entity recognition. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1919–1923.

Zheng, J.; Liang, Z.; Chen, H.; and Ma, Q. 2022. Distilling Causal Effect from Miscellaneous Other-Class for Continual Named Entity Recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3602–3615.