Flexible Weight Tuning and Weight Fusion Strategies for Continual Named Entity Recognition

Yahan Yu[1] **, Duzhen Zhang**[2*] **, Xiuyi Chen**[2]**, Chenhui Chu**[1]
[1]Kyoto University, Japan
[2]Institute of Automation, Chinese Academy of Sciences, China
{yahan@nlp.ist.,chu@}i.kyoto-u.ac.jp, {duzhen.zhang972,chenxiuyi2017}@gmail.com

Abstract

Continual Named Entity Recognition (CNER) is dedicated to sequentially learning new entity types while mitigating catastrophic forgetting of old entity types. Traditional CNER approaches commonly employ knowledge distillation to retain old knowledge within the current model. However, because only the representations of old and new models are constrained to be consistent, the reliance solely on distillation in existing methods still suffers from catastrophic forgetting. To further alleviate the forgetting issue of old entity types, this paper introduces flexible Weight Tuning (WT) and Weight Fusion (WF) strategies for CNER. The WT strategy, applied at each training step, employs a learning rate schedule on the parameters of the current model. After learning the current task, the WF strategy dynamically integrates knowledge from both the current and previous models for inference. Notably, these two strategies are model-agnostic and seamlessly integrate with existing State-Of-The-Art (SOTA) models. Extensive experiments demonstrate that the WT and WF strategies consistently enhance the performance of previous SOTA methods across ten CNER settings in three datasets.[1]

1 Introduction

As a pivotal task in information extraction, Named Entity Recognition (NER) plays a crucial role in various applications, including question answering (Li et al., 2019; Longpre et al., 2021), web search queries (Guo et al., 2009; Zhang et al., 2021). Traditional fully-supervised NER endeavors to classify tokens in a sentence into fixed entity types (Ma and Hovy, 2016). However, real-world entity types typically emerge in a streaming manner, as seen in voice assistants like Siri, which need to recognize new entity types (*e.g.*, Band, Song) to understand new user intents (*e.g.*, GetMusic) (Zhang et al., 2023c). One straightforward solution involves retraining the model on the entire dataset by incorporating both old and new entity types. Nevertheless, this strategy incurs substantial training costs. Another simplistic approach is to fine-tune the previously learned model exclusively on the newly introduced entity types, a scenario known as Continual NER (CNER) (Monaikul et al., 2021; Ma et al., 2023). However, CNER commonly faces catastrophic forgetting, wherein the knowledge acquired from previous entity types is lost after learning new ones (McCloskey and Cohen, 1989; Robins, 1995; Goodfellow et al., 2013; Kirkpatrick et al., 2017; Dong et al., 2022, 2023, 2024; Zheng et al., 2023).

To address the issue of forgetting old entity types, prior CNER methods often employ knowledge distillation to preserve previous knowledge in the current model (Hinton et al., 2015; Zhang et al., 2023b; Chen and He, 2023). These approaches maintain output logits or intermediate features to prevent substantial changes in the parameters of the current model. Specifically, ExtendNER (Monaikul et al., 2021) distills output logits from the old model to the new model, encouraging the new model to generate logits closely resembling those produced by the old model. L&R (Xia et al., 2022) utilizes a two-stage learn-and-review framework. In the learning stage, it adopts a similar approach to ExtendNER; while during the reviewing stage, it integrates synthesized samples of old entity types to enhance the current dataset. Extending the ideas of ExtendNER, CFNER (Zheng et al., 2022) introduces a causal framework to distill causal effects from the non-entity type. CPFD (Zhang et al., 2023a) presents a feature distillation method to retain linguistic knowledge in attention weights, achieving State-Of-The-Art (SOTA) CNER performance. However, since only the output or internal representations of old and new models are constrained to be consistent, relying on knowledge distillation alone pro-

[*] Corresponding author.

[1] Our code is available at https://github.com/ku-nlp/CNER_WT-WF.

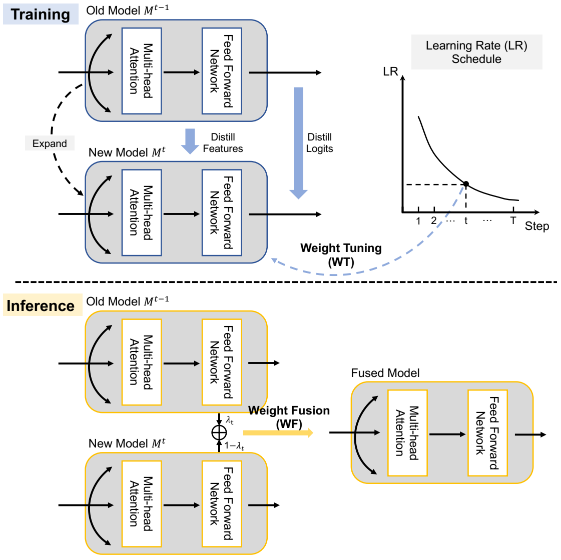

Figure 1: Our WT and WF strategies depict both the training and inference processes. Training benefits from a learning rate schedule, and fusion is employed at inference to consolidate all acquired knowledge.

vides limited gains in CNER performance.

To address this limitation, we propose flexible Weight Tuning (WT) and Weight Fusion (WF) strategies for CNER. Shown in Figure 1, WT applied at training, utilizes a learning rate schedule for the new model's parameters. The initial learning rate gradually decays with steps to effectively retain knowledge of historical entity types. At inference, WF incorporates knowledge from both the new and old models in the form of parameter addition with a dynamic factor. This process eventually yields a fused model for inference, striking a new balance between old and new knowledge without incurring additional computation costs, thereby further mitigating the forgetting of old entity types. Crucially, the WT and WF strategies are model-agnostic, seamlessly integrating with existing SOTA methods in a plug-and-play fashion.

Our contributions can be summarized as follows:

- We present flexible WT and WF strategies for CNER to further mitigate catastrophic forgetting. The former incorporates a learning rate decay schedule during training, while the latter effectively establishes a new balance between old and new knowledge for inference.
- We perform comprehensive experiments across ten CNER settings on three datasets. The outcomes demonstrate the efficacy of the WT and WF strategies, consistently enhancing the performance of SOTA CNER methods.

2 Preliminary

CNER endeavors to learn a model over $t = 1, ..., T$ steps, progressively acquiring proficiency in an expanding array of entity types. Each step involves a distinctive training set $\mathcal{D}^t$, consisting of multiple pairs $(\boldsymbol{X}^t, \boldsymbol{Y}^t)$, where $\boldsymbol{X}^t$ denotes an input token sequence with a length of $|\boldsymbol{X}^t|$ and $\boldsymbol{Y}^t$ signifies the corresponding ground truth label sequence encoded in a one-hot format. It is noteworthy that $\boldsymbol{Y}^t$ exclusively encompasses labels for the current entity types $\mathcal{E}^t$, while all other labels are masked as the non-entity type e_o. At step t (t>1), taking into account the old model $\mathcal{M}^{t-1}$ with parameters $\boldsymbol{\theta}^{t-1}$ and the current training set $\mathcal{D}^t$, we aim to train a new model $\mathcal{M}^t$ with parameters $\boldsymbol{\theta}^t$ capable of recognizing all entities types encountered thus far, as denoted by $\bigcup_{i=1}^{t} \mathcal{E}^i$.

Previous CNER methods (Monaikul et al., 2021; Xia et al., 2022; Zheng et al., 2022; Zhang et al., 2023a) commonly employ a Cross-Entropy (CE) term for acquiring knowledge about new entity types and a Knowledge Distillation (KD) term to preserve the knowledge acquired from previous entity types. This formulation is expressed as follows:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}_t) = -\boldsymbol{Y}^t \log \widehat{\boldsymbol{Y}}^t$$
$$\mathcal{L}_{\text{KD}}(\boldsymbol{\theta}_t) = \begin{cases} -\widehat{\boldsymbol{Y}}^{t-1} \log \widehat{\boldsymbol{Y}}^t \\ ||\boldsymbol{F}^{t-1} - \boldsymbol{F}^t||^2 \end{cases}, \qquad (1)$$

where $\widehat{\boldsymbol{Y}}^{t-1}$ and $\widehat{\boldsymbol{Y}}^t$ denote the output distributions of $\boldsymbol{X}^t$ generated by the old model $\mathcal{M}^{t-1}$ and the new model $\mathcal{M}^t$, respectively. Moreover, $\boldsymbol{F}^{t-1}$ and $\boldsymbol{F}^t$ represent the intermediate features of the old model $\mathcal{M}^{t-1}$ and the new model $\mathcal{M}^t$, respectively.

3 Method

To further alleviate the forgetting of old entity types, we introduce model-agnostic WT and WF strategies to augment existing CNER methods.

The WT strategy implements a learning rate schedule for each continual training step t to adjust the parameters $\boldsymbol{\theta}^t$ of the current model $\mathcal{M}^t$. Specifically, during the initial step (t=1), the learning rate is set to lr_1. For subsequent steps (t>1), the learning rate is gradually reduced to better retain knowledge of historical entity types. Therefore, the learning rate lr_t at step t is expressed as:

$$lr_t = \begin{cases} lr_1 & \text{if } \ t = 1 \\ e^{-\alpha t} \cdot lr_1 & \text{if } \ t > 1 \end{cases}, \qquad (2)$$

Algorithm 1 Pseudo Code for WT and WF strategies in continual steps (highlighted in **red** font)

```
Require: $\theta^0, T, \mathcal{D}^t$, initial learning rate $lr_1$, and hyper-parameters $\alpha$ and $\beta$
  $t \leftarrow 1$                                   ▷ Continual training step
  while $t \leq T$ do
      $\theta^t \leftarrow \theta^{t-1}$
      if $t = 1$ then
          $\boldsymbol{lr_t \leftarrow lr_1}$
      else
          $\boldsymbol{lr_t \leftarrow e^{-\alpha t} \cdot lr_1}$
      end if
      $i \leftarrow 1$                               ▷ Iteration step
      while not converged do
          Sample mini-batch $\{\boldsymbol{X}_i^t, \boldsymbol{Y}_i^t\} \sim \mathcal{D}^t$
          $\theta_{i+1}^t \leftarrow \theta_i^t - \boldsymbol{lr_t} \nabla_{\mathcal{L}_{\text{CE}}+\mathcal{L}_{\text{KD}}}$ from Equation (1)
          $i \leftarrow i + 1$
      end while
      Initialize $\boldsymbol{C_{new}^t, C_{old}^t}$    ▷ Number of new and old entity types in the current step $t$
      $\boldsymbol{\lambda^t \leftarrow (\frac{C_{new}^t}{C_{new}^t + C_{old}^t})^\beta}$
      $\boldsymbol{\theta^{\text{fused}} \leftarrow \lambda^t \theta^t + (1 - \lambda^t)\theta^{t-1}}$
      $\boldsymbol{\theta^t \leftarrow \theta^{\text{fused}}}$
      $t \leftarrow t + 1$
  end while
```

where e^{-t} denotes the exponential decay of the learning rate, and α is a hyper-parameter.

Following each training step t (t>1), the WF strategy, instead of employing the new model directly, produces a fused model for inference. This fused model integrates knowledge from both the new and old models using a dynamic balance factor, establishing a refined equilibrium between old and new knowledge without introducing additional computation costs. The formulation is as follows:

$$\begin{aligned} \boldsymbol{\theta}^{\text{fused}} &= \lambda^t \boldsymbol{\theta}^t + (1 - \lambda^t)\boldsymbol{\theta}^{t-1} \\ \lambda^t &= \left(\frac{C_{\text{new}}^t}{C_{\text{new}}^t + C_{\text{old}}^t}\right)^\beta \end{aligned}, \quad (3)$$

where C_{new}^t and C_{old}^t represent the number of new and old entity types in the current step t, respectively, and β is a hyper-parameter.

The pseudo code for our WT and WF strategies is presented in Algorithm 1.

4 Experiments

4.1 Experimental Setup

To ensure a fair comparison, we adhere to the setup of CPFD (Zhang et al., 2023a) method as below:

	# Entity Type	# Sample	Entity Type Sequence (Alphabetical Order)
CoNLL2003	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 1: The statistics for each dataset.

Datasets We use three widely adopted NER datasets to assess the impact of **WT&WF** strategies, namely CoNLL2003 (Sang and De Meulder, 1837), I2B2 (Murphy et al., 2010), and OntoNotes5 (Hovy et al., 2006). The statistics for these datasets are shown in Table 1. We apply the greedy sampling algorithm, detailed in CFNER (Zheng et al., 2022), to split the training set into disjoint slices, each corresponding to different continual steps. Within each slice, we retain only the labels belonging to the entity types under learn, while masking others as the non-entity type.

CNER Settings During training, we sequentially learn entity types in alphabetical order and train models with corresponding slices. For CoNLL2003, we employ two CNER settings: 1-1 and 2-1. Regarding I2B2 and OntoNotes5, we establish four CNER settings: 1-1, 2-2, 8-1, and 8-2. The notation a-b indicates that we utilize a entity types to learn a base model and every b entity type to train in each continual step. During validation, we retain only the labels of the current entity types under learning, masking others as the non-entity type within the validation set. At each step, we select the model that achieves the best validation performance for both testing and the subsequent step. In the testing phase, we preserve labels for all previously learned entity types, designating the remainder as the non-entity type within the test set.

Evaluation Metrics Given the challenge of entity type imbalance in NER, we employ Micro F1 (Mi-F1) and Macro F1 (Ma-F1) to assess the performance. The reported result denotes the mean across all steps, including the first one, serving as the final performance. To assess the significance of the performance improvement, we conduct a paired t-test with a significance level of 0.05 (Koehn, 2004).

Baselines We consider the following baselines for evaluation, encompassing SOTA CNER methods: ExtendNER (Monaikul et al., 2021), CFNER (Zheng et al., 2022), and CPFD (Zhang

Dataset	Baseline	1-1		2-2		8-1		8-2	
		Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
I2B2	Fine-tuning	17.43±0.54	13.81±1.14	28.57±0.26	21.43±0.41	20.83±1.78	18.11±1.66	23.60±0.15	23.54±0.38
	PODNet	12.31±0.35	17.14±1.03	34.67±2.65	24.62±1.76	39.26±1.38	27.23±0.93	36.22±12.90	26.08±7.42
	LUCIR	43.86±2.43	31.31±1.62	64.32±0.76	43.53±0.59	57.86±0.87	33.04±0.39	68.54±0.27	46.94±0.63
	Self-Training	31.98±2.12	14.76±1.31	55.44±4.78	33.38±3.13	49.51±1.35	23.77±1.01	48.94±6.78	29.00±3.04
	ExtendNER	**42.85±2.86**	**24.05±1.35**	57.01±4.14	35.29±3.38	43.95±2.01	23.12±1.79	52.25±5.36	30.93±2.77
	ExtendNER†	41.65±10.11	23.11±2.70	**67.60±1.15**	**42.58±1.59**	**45.14±2.91**	**27.41±0.88**	**56.48±2.41**	**38.88±1.38**
	ExtendNER† + **WT&WF**	**58.50±5.11**	**31.18±3.84**	**73.84±3.98**	**45.51±4.96**	**47.70±2.74**	**29.05±3.81**	**61.92±1.15**	**41.19±1.59**
	CFNER	62.73±3.62	36.26±2.24	71.98±0.50	49.09±1.38	**59.79±1.70**	**37.30±1.15**	69.07±0.89	51.09±1.05
	CFNER†	**64.79±0.26**	**37.79±0.65**	**72.58±0.59**	**51.71±0.84**	56.66±3.22	36.84±1.35	**69.12±0.94**	**51.61±0.87**
	CFNER† + **WT&WF**	**67.80±0.99**	**39.77±0.38**	**74.17±0.58**	**53.25±1.23**	**61.58±4.03**	**40.72±2.67**	**71.73±0.10**	**53.02±0.46**
	CPFD	**74.19±0.95**	**48.34±1.45**	78.19±0.58	56.04±1.22	74.75±1.35	56.19±2.46	**81.05±0.87**	65.04±1.13
	CPFD†	73.04±0.73	46.36±2.13	**78.25±0.29**	**56.09±0.57**	**75.06±1.79**	**57.47±2.33**	80.66±2.05	**65.08±2.23**
	CPFD† + **WT&WF**	**75.11±0.18**	**50.16±0.55**	**79.70±0.23**	**58.74±1.59**	**78.51±1.85**	**60.45±1.48**	**82.47±0.70**	**66.29±0.32**
OntoNotes5	Fine-tuning	15.27±0.26	10.85±1.11	25.85±0.11	20.55±0.24	17.63±0.57	12.23±1.08	29.81±0.12	20.05±0.16
	PODNet	9.06±0.56	8.36±0.57	34.67±1.08	24.62±0.85	29.00±0.86	20.54±0.91	37.38±0.26	25.85±0.29
	LUCIR	28.18±1.15	21.11±0.84	64.32±1.79	43.53±1.11	66.46±0.46	46.29±0.38	76.17±0.09	55.58±0.55
	Self-Training	50.71±0.79	33.24±1.06	68.93±1.67	50.63±1.66	73.59±0.66	49.41±0.77	77.07±0.62	53.32±0.63
	ExtendNER	50.53±0.86	32.84±0.84	**67.61±1.53**	**49.26±1.49**	73.12±0.93	49.55±0.90	76.85±0.77	54.37±0.57
	ExtendNER†	**51.36±0.77**	**33.38±0.98**	63.03±9.39	47.64±5.15	**73.65±0.19**	**50.55±0.56**	**77.86±0.10**	**55.21±0.51**
	ExtendNER† + **WT&WF**	**54.82±0.39**	**35.95±0.50**	**70.45±0.89**	**51.95±0.59**	**77.52±0.41**	**53.67±0.61**	**79.68±0.37**	**56.88±0.54**
	CFNER	**58.94±0.57**	**42.22±1.10**	**72.59±0.48**	**55.96±0.69**	**78.92±0.58**	57.51±1.32	**80.68±0.25**	60.52±0.84
	CFNER†	58.44±0.71	41.75±1.51	72.10±0.31	55.02±0.35	78.25±0.33	**58.64±0.42**	80.09±0.37	**61.06±0.37**
	CFNER† + **WT&WF**	**62.76±1.54**	**46.50±0.88**	**74.20±0.29**	**57.04±0.71**	**79.97±0.37**	**60.30±0.54**	**81.69±1.42**	**62.76±0.91**
	CPFD	**66.73±0.70**	**54.12±0.30**	74.33±0.30	57.75±0.35	81.87±0.47	65.52±1.05	83.38±0.18	66.27±0.75
	CPFD†	65.73±0.56	53.83±1.12	**74.36±0.33**	**57.75±0.49**	**82.07±0.30**	**65.79±0.36**	**83.49±0.18**	**66.66±0.69**
	CPFD† + **WT&WF**	**68.16±0.78**	**55.46±0.80**	**76.47±0.30**	**59.97±0.28**	**83.51±0.25**	**67.03±0.26**	**84.52±0.21**	**68.90±1.10**

Table 2: Comparisons with baselines on the I2B2 and OntoNotes5 datasets. † represents our reproduced results with the open codebases. Other baseline results are directly cited from CPFD. **Red** and **blue** represent the maximum and second maximum values in ExtendNER, CFNER, and CPFD. ExtendNER, CFNER, and CPFD with **WT&WF** significantly outperform their corresponding vanilla methods (with $p < 0.05$).

et al., 2023a). Moreover, we include the lower bound method, Fine-tuning, directly employing new data for fine-tuning the model without using any anti-forgetting techniques. Furthermore, we consider continual learning methods from computer vision like PODNet (Douillard et al., 2020), LUCIR (Hou et al., 2019), and Self-Training (Lange et al., 2019). More details on these baselines can be found in Appendix A.

Implementation Details We employ the “BIO" labeling schema for all datasets. Our NER model utilizes the bert-base-cased (Kenton and Toutanova, 2019) model as the encoder and employs a fully-connected layer as the classifier. For each CNER setting, if each continual training step (t>2) learns an entity type, we train the model for 10 epochs; otherwise, for 20 epochs. We set the batch size, initial learning rate lr_1, hyper-parameters α and β to 8, $4e$-4, $1e$-2, and 0.5, respectively. All experiments are conducted on an NVIDIA A6000 GPU with 48GB of memory, and each experiment is run 5 times to ensure statistical significance.

4.2 Experimental Results

Main Results Tables 2 and 6 (Appendix B) present the performance of our **WT&WF** and baselines across 10 CNER settings on the I2B2, OntoNotes5, and CoNLL2003 datasets. Our findings show substantial improvements in both Mi- and Ma-F1 scores for ExtendNER, CFNER, and CPFD after applying **WT&WF** across nearly all CNER settings. Moreover, CPFD + **WT&WF** achieves new SOTA performance across all settings. These outcomes highlight the broad applicability and effectiveness of our **WT&WF** strategies.

Methods	8-1		8-2	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1
CPFD	75.06±1.79	57.47±2.33	80.66±2.05	65.08±2.23
CPFD + **WT**	77.39±2.98	59.01±3.25	81.45±0.59	65.92±1.25
CPFD + **WF**	76.49±2.34	58.49±2.46	81.33±0.52	65.64±1.16
w/o DBF (λ_t=0.5)	75.19±1.90	58.17±2.61	80.77±0.46	65.35±0.61
CPFD + **WT&WF**	**78.51±1.85**	**60.45±1.48**	**82.47±0.70**	**66.29±0.32**

Table 3: The ablation study of our **WT&WF** strategies on the I2B2 dataset under the 8-1 and 8-2 settings. **w/o DBF** denotes removing Dynamic Balance Factor λ_t in **WF** and fixing its value to 0.5.

Ablation Study Table 3 presents the results of ablation study on our **WT&WF** strategies based on the CPFD method. The best performance is observed when both strategies are used simultaneously. While utilizing either strategy alone results in some improvement compared to the original

Methods	Mi-F1	Ma-F1
CFNER + WT&WF ($\alpha = 1.0$)	20.41±3.82	18.23±1.18
CFNER + WT&WF ($\alpha = 1e$-1)	**61.17±3.42**	**40.61±1.89**
CFNER + WT&WF ($\alpha = 5e$-2)	**61.25±3.16**	**40.48±2.85**
CFNER + WT&WF ($\alpha = 1e$-2)	**61.58±4.03**	**40.72±2.67**
CFNER + WT&WF ($\alpha = 5e$-3)	**61.77±3.22**	**40.98±2.47**
CFNER + WT&WF ($\alpha = 1e$-3)	**61.45±3.97**	**41.23±2.15**
CFNER + WT&WF ($\alpha = 1e$-9)	58.79±3.62	38.39±2.91

Table 4: The hyper-parameter analysis of α on I2B2 dataset under 8-1 setting (β is fixed to 0.5). **Bold** represents the results not significantly different (not satisfy $p < 0.05$) from the results of $\alpha = 1e$-2.

Methods	Mi-F1	Ma-F1
CPFD + WT&WF ($\beta = 0.0$)	77.39±2.98	59.01±3.25
CPFD + WT&WF ($\beta = 0.5$)	**78.51±1.85**	**60.45±1.48**
CPFD + WT&WF ($\beta = 1$)	**78.29±1.94**	**60.33±1.28**
CPFD + WT&WF ($\beta = 2$)	**78.03±1.38**	**60.18±1.62**
CPFD + WT&WF ($\beta = 10$)	35.92±1.42	29.91±1.47

Table 5: The hyper-parameter analysis of β on I2B2 dataset under 8-1 setting (α is fixed to $1e$-2). **Bold** represents the results not significantly different (not satisfy $p < 0.05$) from the results of $\beta = 0.5$.

method, fixing λ_t to 0.5 in **WF** does not result in a discernible enhancement over the original method.

Sensitivity of Hyper-parameters Table 4 shows the sensitivity analysis on α. We set 7 optional values ranging from 1.0 to $1e$-9. It demonstrates that if we don't choose values that are too extreme ($\alpha = 1e$-9 or 1.0), results won't fluctuate much. We also perform a significance test, which shows the results of $\alpha = 1e$-1 to $\alpha = 1e$-3 don't have statistically significant difference (not satisfy $p < 0.05$) compared with the results of $\alpha = 1e$-2. On the contrary, when $\alpha = 1.0$, the decay is too rapid, resulting in the learning rate for new tasks approaching zero. And when $\alpha = 1e$-9, the results have a significant decrease (satisfy $p < 0.05$) compared with the results of $\alpha = 1e$-2, because $\alpha = 1e$-9 is almost equivalent to a fixed learning rate. Thus, our WT strategy is stable within a selection range of $\alpha = 1e$-1 to $1e$-3, and there is no need for intentional hyper-parameter adjustment.

Moreover, Table 5 shows the sensitivity analysis on β. We set 5 optional values ranging from 0.0 to 10. We perform a significance test, which shows the results of $\beta = 1.0$ and $\beta = 2.0$ don't have statistically significant differences (not satisfy $p < 0.05$) compared with the results of $\beta = 0.5$. On the contrary, results of $\beta = 0.0$ (using the new model directly for inference, as with the previous method) and $\beta = 10$ (approaching the scenario of using only the old model for inference) both have a significant decrease (satisfy $p < 0.05$) compared with the results of $\beta = 0.5$. Thus, our WF strategy is also stable within a selection range from $\beta = 0.5$ to $\beta = 2$, and there is also no need for intentional hyperparameter adjustment.

5 Conclusion

In this paper, we introduce model-agnostic WT and WF strategies for CNER to mitigate catastrophic forgetting. WT employs a learning rate schedule within each training step to adjust the parameters of the new model, while WF dynamically fuses the new and old models to maintain a balance between new and old knowledge. Extensive experiments conducted on 10 CNER settings across 3 datasets illustrate that our strategies further enhance the previous SOTA methods.

Acknowledgments

We express our gratitude to the anonymous reviewers for their valuable and insightful comments. This work was supported by JSPS KAKENHI Grant Number JP23K28144.

Limitations

Existing CNER methods may suffer from instability, with the order of entity types significantly impacting performance. In real-world scenarios, the optimal entity type order is unknown beforehand, so an ideal CNER method should perform consistently regardless of this order. Previous experiments have maintained a constant entity type order (e.g., alphabetical order). Future research should explore the stability of CNER methods under varying entity type learning orders. Additionally, given the model-agnostic nature of our approach, future work can investigate combination with more CNER methods and encoders, including larger language models beyond BERT.

Ethical Considerations

In addressing ethical considerations, we wish to elucidate the following points: (1) This work does not involve the use of sensitive data or tasks; (2) We provide thorough descriptions of the dataset statistics and implementation details, and analyses are closely aligned with the experimental results; (3) All experiments use pre-existing datasets sourced from publicly available research.

References

Yi Chen and Liang He. 2023. SKD-NER: Continual Named Entity Recognition via Span-based Knowledge Distillation with Reinforcement Learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6689–6700.

Jiahua Dong, Hongliu Li, Yang Cong, Gan Sun, Yulun Zhang, and Luc Van Gool. 2024. No one left behind: Real-world federated class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2054–2070.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. 2022. Federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiahua Dong, Duzhen Zhang, Yang Cong, Wei Cong, Henghui Ding, and Dengxin Dai. 2023. Federated Incremental Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3934–3943.

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.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 267–274.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

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, Mitch 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.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. *CoRR*, abs/1909.08383.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1340–1350.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-Based Knowledge Conflicts in Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7052–7063.

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)*, pages 5959–5979.

Xuezhe Ma and Eduard Hovy. 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)*, pages 1064–1074.

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

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*, volume 35, 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.

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

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-Supervised Self-Training of Object Detection Models. In *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), 5-7 January 2005, Breckenridge, CO, USA*, pages 29–36.

Erik F Tjong Kim Sang and Fien De Meulder. 1837. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Development*, 922:1341.

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*, pages 2291–2300.

Duzhen Zhang, Wei Cong, Jiahua Dong, Yahan Yu, Xiuyi Chen, Yonggang Zhang, and Zhen Fang. 2023a. Continual Named Entity Recognition without Catastrophic Forgetting. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Duzhen Zhang, Hongliu Li, Wei Cong, Rongtao Xu, Jiahua Dong, and Xiuyi Chen. 2023b. 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.

Duzhen Zhang, Yahan Yu, Feilong Chen, and Xiuyi Chen. 2023c. Decomposing Logits Distillation for Incremental Named Entity Recognition. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1919–1923.

Ningyu Zhang, Qianghuai Jia, Shumin Deng, Xiang Chen, Hongbin Ye, Hui Chen, Huaixiao Tou, Gang Huang, Zhao Wang, Nengwei Hua, and Huajun Chen. 2021. AliCG: Fine-grained and Evolvable Conceptual Graph Construction for Semantic Search at Alibaba. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3895–3905.

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*.

Junhao Zheng, Shengjie Qiu, and Qianli Ma. 2023. Learn or Recall? Revisiting Incremental Learning with Pre-trained Language Models. *arXiv preprint arXiv:2312.07887*.

A Baselines

The detailed introductions about the baselines in our experiments are as follows.

PODNet (Douillard et al., 2020): PODNet is employed to address the challenge of catastrophic forgetting in continual learning for image classification, transferred to the CNER scenario. The overall loss of the model encompasses both classification loss and distillation loss. In the realm of classification, PODNet opts for the neighborhood component analysis loss as a substitute for the conventional cross-entropy loss. In the computation of distillation loss, PODNet imposes constraints on the output of each intermediate layer.

LUCIR (Hou et al., 2019): LUCIR establishes a framework for incrementally learning in continual image classification tasks, transferred to the CNER scenario, sharing a conceptual similarity with PODNet in addressing catastrophic forgetting. Its overall loss comprises three components: (1) the cross-entropy loss on samples with new entity types; (2) the distillation loss between features extracted by the old model and the new one; and (3) the margin-ranking loss on reserved samples for old entity types.

Self-Training (Rosenberg et al., 2005; Lange et al., 2019): Self-Training initially employs the pre-existing model to label the non-entity type tokens with their respective old entity types. Subsequently, the novel model undergoes training on fresh data, incorporating annotations for all entity types. The ultimate objective is to minimize the cross-entropy loss across all entity types, ensuring comprehensive and effective model training.

ExtendNER (Monaikul et al., 2021): ExtendNER explores the application of knowledge distillation to CNER and shares similarities with Self-Training, but it computes cross-entropy loss for entity type tokens and KL divergence loss for non-entity type tokens. During the training process, ExtendNER minimizes the sum of cross-entropy loss and KL divergence loss.

CFNER (Zheng et al., 2022): Based on ExtendNER, CFNER proposes a causal framework for CNER, enabling the extraction of causal effects from the non-entity type. Specifically, it utilizes the old model to recognize non-entity type tokens belonging to previous entity types, extracting causal effects. It also employs a curriculum learning strategy to mitigate recognition errors.

CPFD (Zhang et al., 2023a): CPFD addresses two challenges in CNER: catastrophic forgetting and the semantic shift problem of the non-entity type. To tackle catastrophic forgetting, CPFD introduces a pooled feature distillation loss that balances stability and plasticity. Simultaneously, CPFD proposes a confidence-based pseudo-labeling strategy to reduce the impact of label noise and address the semantic shift problem.

B Supplementary Results

We also experiment with the effectiveness of our **WT&WF** strategies on the CoNLL2003 dataset and results are shown in Table 6. However, the improvement in results was not as substantial as in the I2B2 and OntoNotes5. Further analysis suggests that CoNLL2003 contains fewer (four) entity types, resulting in less noticeable forgetting of previous knowledge. CoNLL2003 is also less difficult compared to the other datasets, so the previous baseline can almost reach the performance upper-bound and leaves less room for improvements, especially for CPFD. Consequently, the enhancement on the previous SOTA methods is relatively minor.

Baseline	1-1		2-1	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1
Fine-tuning	50.84±0.10	40.64±0.16	57.45±0.05	43.58±0.18
PODNet	36.74±0.52	29.43±0.28	59.12±0.54	58.39±0.99
LUCIR	74.15±0.43	70.48±0.66	80.53±0.31	77.33±0.31
Self-Training	76.17±0.91	72.88±1.12	76.65±0.24	66.72±0.11
ExtendNER	**76.36±0.98**	73.04±1.80	76.66±0.66	66.36±0.64
ExtendNER†	76.07±0.35	**73.06±0.29**	**77.89±0.42**	**69.92±1.02**
ExtendNER† + **WT&WF**	**79.29±1.18**	**77.55±2.12**	**78.94±0.53**	**70.98±1.36**
CFNER	**80.91±0.29**	**79.11±0.50**	80.83±0.36	75.20±0.32
CFNER†	80.29±0.21	78.44±0.24	**81.52±0.43**	**77.20±0.82**
CFNER† + **WT&WF**	**82.37±0.61**	**80.59±0.63**	**82.58±0.61**	**78.37±0.88**
CPFD	**82.24±0.63**	**79.94±0.66**	85.70±0.19	**83.49±0.16**
CPFD†	82.18±0.15	79.42±0.19	**85.75±0.43**	83.41±0.31
CPFD† + **WT&WF**	**82.94±0.40**	**80.97±0.40**	**86.79±0.50**	**84.86±0.50**

Table 6: Comparisons with baselines on CoNLL2003. † represents our reproduced results with the open codebases. Other results are directly cited from CPFD. **Red** and **blue** represent the maximum and second maximum values in ExtendNER, CFNER, and CPFD. ExtendNER, CFNER, and CPFD with **WT&WF** outperform their corresponding vanilla methods (with $p < 0.05$).