Learning "O" Helps for Learning More: Handling the Unlabeled Entity Problem for Class-incremental NER

Ruotian Ma[1*], **Xuanting Chen**[1*], **Lin Zhang**[1], **Xin Zhou**[1], **Junzhe Wang**[1], **Tao Gui**[2†], **Qi Zhang**[1†], **Xiang Gao**[3], **Yunwen Chen**[3]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Institute of Modern Languages and Linguistics, Fudan University, Shanghai, China
[3]DataGrand Information Technology (Shanghai) Co., Ltd.
`{rtma19,xuantingchen21,tgui,qz}@fudan.edu.cn`

Abstract

As the categories of named entities rapidly increase, the deployed NER models are required to keep updating toward recognizing more entity types, creating a demand for class-incremental learning for NER. Considering the privacy concerns and storage constraints, the standard paradigm for class-incremental NER updates the models with training data only annotated with the new classes, yet the entities from other entity classes are unlabeled, regarded as "Non-entity" (or "O"). In this work, we conduct an empirical study on the "Unlabeled Entity Problem" and find that it leads to severe confusion between "O" and entities, decreasing class discrimination of old classes and declining the model's ability to learn new classes. To solve the Unlabeled Entity Problem, we propose a novel representation learning method to learn discriminative representations for the entity classes and "O". Specifically, we propose an entity-aware contrastive learning method that adaptively detects entity clusters in "O". Furthermore, we propose two effective distance-based relabeling strategies for better learning the old classes. We introduce a more realistic and challenging benchmark for class-incremental NER, and the proposed method achieves up to 10.62% improvement over the baseline methods.

1 Introduction

Existing Named Entity Recognition systems are typically trained on a large-scale dataset with predefined entity classes, then deployed for entity recognition on the test data without further adaptation or refinement (Li et al., 2020; Wang et al., 2022). In practice, the newly-arriving test data may include new entity classes, and the user's required entity class set might keep expanding. Therefore, it is in demand that the NER model can be incrementally updated for recognizing new entity classes.

[*]Equal contribution.
[†]Corresponding authors.

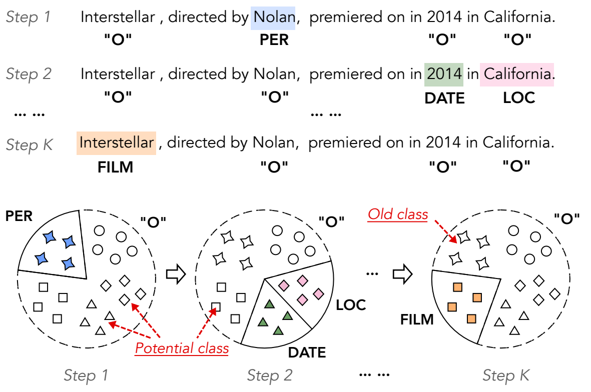

Step 1
Interstellar , directed by Nolan, premiered on in 2014 in California.
"O" PER "O" "O"
Step 2
Interstellar , directed by Nolan, premiered on in 2014 in California.
"O" "O" DATE LOC
... ...
Step K
Interstellar , directed by Nolan, premiered on in 2014 in California.
FILM "O" "O" "O"
PER
"O"
LOC
DATE
Old class
FILM
Potential class
Step 1
Step 2
Step K

Figure 1: Problems of class-incremental NER. In each incremental step, the data is only labeled with current classes, so the "O" class actually contains entities from old classes and entities from potential classes.

However, one challenge is that the training data of old entity classes may not be available due to privacy concerns or memory limitations (Li and Hoiem, 2017; Zhang et al., 2020). Also, it is expensive and time-consuming to re-annotate all the old entity classes whenever we update the model (Delange et al., 2021; Bang et al., 2021). To solve the problem, Monaikul et al. (2021) proposes to incrementally update the model with new datasets only covering the new entity classes, adopted by following studies as standard **class-incremental NER** paradigm.

However, as NER is a sequence labeling task, annotating only the new classes means entities from other entity classes are regarded as "Non-entity" (or "O") in the dataset. For example, in step 2 in Fig.1, the training data for model updating is only annotated with "LOC" and "DATE", while the entities from "PER" and "FILM" are unlabeled and regarded as "O" during training. We refer to this problem as the "Unlabeled Entity Problem" in class-incremental NER, which includes two types of unlabeled entities: (1) old entity classes (e.g., "PER" in step 2) that the model learned in previous steps are unlabeled in the current step, causing the model catastrophically forgetting these old classes. (Lopez-Paz and Ranzato, 2017; Castro et al., 2018) (2) potential entity classes that are not annotated till the current step, yet might be required in a future step. For example, the "FILM" class is not annotated till step 2, yet is required in step K.

In this work, we conduct an empirical study to demonstrate the significance of the "Unlabeled Entity Problem" on class-incremental NER. We observe that: (1) The majority of prediction errors come from the confusion between entities and "O". (2) Mislabeled as "O" leads to the reduction of class discrimination of old entities during incremental learning. (3) The model's ability to learn new classes also declines as the potential classes are unlabeled during incremental training. These problems attribute to the serious performance drop of incremental learning with the steps increasing.

To tackle the Unlabeled Entity Problem, we propose a novel representation learning method for learning discriminative representations for the unlabeled entity classes and "O". Specifically, we propose an entity-aware contrastive learning approach, which adaptively detects entity clusters from "O" and learns discriminative representations for these entity clusters. To further maintain the class discrimination of old classes, we propose two distance-based relabeling strategies. By relabeling the entities from old classes with high accuracy, this practice not only keeps the performance of old classes, but also benefits the model's ability to separate new classes from "O".

We also argue that the experimental setting of previous works Monaikul et al. (2021) is less realistic. Specifically, they introduce only one or two entity classes in each incremental step, and the number of total steps is limited. In real-world applications, it is more common that a set of new categories is introduced in each step (e.g., a set of product types), and the incremental learning steps can keep increasing. In this work, we provide a more realistic and challenging benchmark based on the Few-NERD dataset (Ding et al., 2021), following the settings of previous studies (Rebuffi et al., 2017; Li and Hoiem, 2017). We conduct intensive experiments on the proposed methods and other comparable baselines, verifying the effectiveness of the proposed method [1].

To summarize the contribution of this work:

- We conduct an empirical study to demonstrate the significance of the "Unlabeled Entity Problem" in class-incremental NER.
- Based on our observations, we propose a novel representation learning approach for better learning the unlabeled entities and "O", and verify the effectiveness of our method with extensive experiments.
- We provide a more realistic and challenging benchmark for class-incremental NER.

2 Class-incremental NER

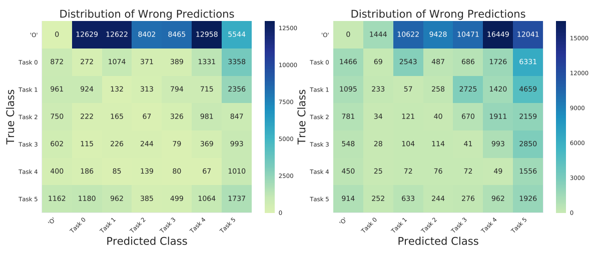
In this work, we focus on class-incremental learning on NER. Formally, there are N incremental steps, corresponding to a series of tasks $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_N\}$. Here, $\mathcal{T}_t = (\mathcal{D}_t^{tr}, \mathcal{D}_t^{dev}, \mathcal{D}_t^{test}, \mathcal{C}_{t,new}, \mathcal{C}_{t,old})$ is the task at the t^{th} step. $\mathcal{C}_{t,new}$ is the label set of the current task, containing only the **new classes** introduced in the current step (e.g., {"LOC", "DATE"} in Fig.1, step 2). $\mathcal{C}_{t,old} = \bigcup_{i=1}^{t-1} \mathcal{C}_{i,new} \cup \{\text{“}O\text{”}\}$ is the label set of **old classes**, containing all classes in previous tasks and the class "O" (e.g., {"PER", "O"} in Fig.1, step 2). $\mathcal{D}_t^{tr} = \{X_t^j, Y_t^j\}_{j=1}^n$ is the training set of task t, where each sentence $X_t^j = \{x_t^{j,1}, \dots, x_t^{j,l}\}$ and $Y_t^j = \{y_t^{j,1}, \dots, y_t^{j,l}\}, y_t^{j,k} \in \mathcal{C}_{t,new}$ is annotated with only the new classes. In each step t, the model $\mathcal{A}_{t-1}$ from the last step needs to be updated with only the data $\mathcal{D}_t^{tr}$ from the current step, and is expected to perform well on the test set covering all learnt entity types $\mathcal{C}_t^{all} = \mathcal{C}_{t,new} \cup \mathcal{C}_{t,old}$.

3 The Importance of Unlabeled Entity Problem in Class-incremental NER

In this section, we demonstrate the importance of the Unlabeled Entity Problem in Class-incremental NER with empirical studies. We conduct experiments on a challenging dataset, the Few-NERD dataset, to investigate the problems in class-incremental NER. We conduct experiments with two existing methods: (1) **iCaRL** (Rebuffi et al., 2017), a typical and well-performed method in class-incremental image classification. (2) **Continual NER** (Monaikul et al., 2021), the previous state-of-the-art method in class-incremental NER. More details of the dataset and the baseline methods can be found in Section 5.

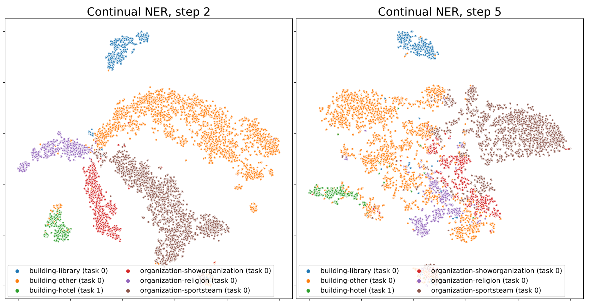
Observation 1: The majority of prediction errors come from the confusion between entities

[1]Our code is publicly available at `https://github.com/rtmaww/O_CILNER`.



(a) iCaRL (b) Continual NER

Figure 2: Distributions of prediction errors of different models in step 6. The first row represents the number of samples belonging to the "O" class wrongly recognized as entities, which shows the severe confusion between "O" and entity classes.



(a) Continual NER, step 2 (b) Continual NER, step 5

Figure 3: Visualization of the representation variation of the old classes during incremental learning. The class discrimination seriously decrease in step 5.

and "O". In Fig.2, we show the distributions of prediction errors of different models in step 6, where the y-axis denotes samples belonging to "O" or the classes of different tasks. The x-axis denotes the samples are wrongly predicted as "O" or as classes from different tasks. Each number in a grid denotes the number of error predictions. From the results, we can see that the majority of error predictions are samples belonging to "O" wrongly predicted as entities (the first row of each model), indicating serious confusion between "O" and entity classes, especially the old entity classes. As explained in Section 1, the training data of each new task is only annotated with the new entity classes and the entities from old classes are labeled as "O". As the training proceeds, the class variance between the true "O" and old entity classes will decrease, leading to serious confusion of their representations.

Observation 2: Old entity classes become less discriminative during incremental learning. We further investigate the representation variation of old classes during incremental learning. As shown in Fig.3, we select similar classes from step 0 and step 1, and visualize their representations after step 2 and step 5. The results show that the representations of these classes are discriminative enough in step 2. However, after a series of incremental steps, the representations of these old classes become less discriminative, leading to decreasing performance of old classes. This phenomenon also indicates the influence of the unlabeled entity problem on the unlabeled old classes.

Steps	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Full Data	72.7	69.2	68.3	67.0	67.3	69.1	68.8
iCaRL	71.3	56.9	52.6	48.8	53.4	48.1	39.6
Con. NER	72.4	63.5	56.9	52.5	56.8	51.8	42.2

Table 1: Performance of the new classes on dev set (only containing the new classes) keep decreasing during incremental learning. Here, Full Data is the model trained with datasets labeled with both old and new classes.

Observation 3: The model's ability to learn new classes declines during incremental learning. Finally, we conduct an experiment to investigate the model's ability to learn new classes. In Table 1, we test the results of new classes in each step on dev sets that only contain these new classes. Here, **Full Data** is a baseline that trains on datasets that both old and new classes are annotated. Surprisingly, we find that the performance of the new classes of iCaRL and Continual NER keeps decreasing during incremental learning, compared to the stable performance of Full Data. This phenomenon is also related to the Unlabeled Entity Problem. As explained in the introduction, the potential entity classes (i.e., the entity classes that might be needed in a future step) are also unlabeled and regarded as "O" during incremental learning. As a result, the representations of these classes become less separable from similar old classes (also labeled as "O"), thus hindering the model's ability to learn new classes.

Conclusion to the Observations: Based on above observations, we propose that appropriate representation learning are required to tackle the Unlabeled Entity Problems. The representations of entity and "O" are expected to meet the following requirements: (1) The "O" representations are expected to be distinct from the entity representations, so as to decline the confusion between "O" and entities (**Observation 1**). (2) The representations of old entity classes are expected to keep discrimina-

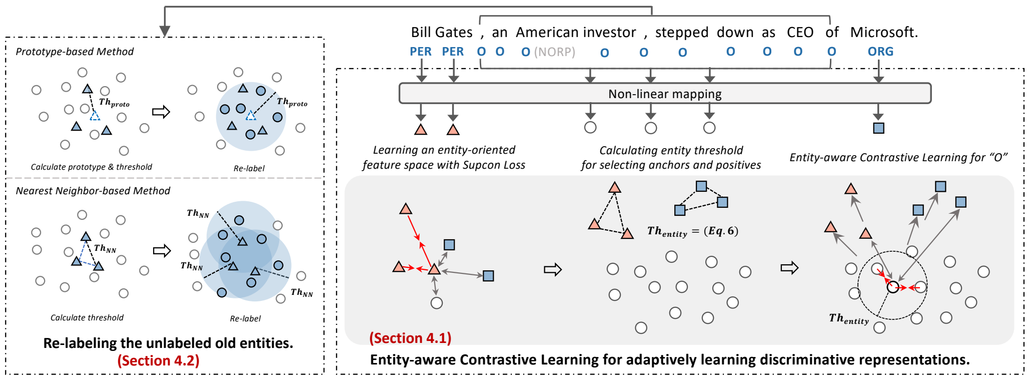

Prototype-based Method
Th_{proto}
Calculate prototype & threshold
Re-label
Nearest Neighbor-based Method
Th_{NN}
Calculate threshold
Re-label
Re-labeling the unlabeled old entities.
(Section 4.2)
Bill Gates , an American investor , stepped down as CEO of Microsoft.
PER PER O O O (NORP) O O O O O O O ORG
Non-linear mapping
Learning an entity-oriented feature space with Supcon Loss
Calculating entity threshold for selecting anchors and positives
Entity-aware Contrastive Learning for "O"
Th_{entity} = (Eq. 6)
Th_{entity}
(Section 4.1)
Entity-aware Contrastive Learning for adaptively learning discriminative representations.

Figure 4: Overview of the proposed representation learning method: (1) We propose an entity-aware contrastive learning method to adaptively detect entity clusters from "O" and learn discriminative representations for these entities. (2) We propose two distance-based relabeling strategies to further maintain the performance of old classes.

tive in spite of being labeled as "O" (**Observation 2**). (3) The potential entity class are expected to be detected and separated from "O", and also be discriminative to other entity classes (**Observation 3**). These observations and conclusions contribute to the motivation of the proposed method.

4 Handling the Unlabeled Entity Problem

In order to learn discriminative representations for unlabeled entity classes and the true "O" (connected to **Observations 1, 2, 3**), we propose entity-aware contrastive learning, which adaptively detects entity clusters in "O" during contrastive learning. To further maintain the class discrimination of old classes (connected to **Observation 2**), we propose two distance-based relabeling strategies to relabel the unlabeled entities from old classes in "O". Additionally, we propose the use of the Nearest Class Mean classifier based on learnt representations in order to avoid the prediction bias of linear classifier.

Rehearsal-based task formulation To better learn representations for entities and "O", in this work, we follow the memory replay (rehearsal) setting adopted by most of the previous works (Rebuffi et al., 2017; Mai et al., 2021; Verwimp et al., 2021). Formally, we retain a set of exemplars $\mathcal{M}_c = \{x_c^i, y_c^i, \overline{X}_c^i\}_{i=1}^K$ for each class c, where x_c^i refers to one token x labeled as class c and $\overline{X}$ is the context of x labeled as "O". In all our experiments, we set $K = 5$ [2].

4.1 Entity-aware Contrastive Learning

In this section, we introduce the entity-aware contrastive learning, which dynamically learns entity clusters in "O". To this aim, we first learn an entity-oriented feature space, where the representations of entities are distinctive from "O". This entity-oriented feature space is learnt through contrastive learning on the labeled entity classes in the first M epochs of each step. Based on the entity-oriented feature space, we further conduct contrastive learning on "O", with the anchors and positive samples dynamically selected based on an entity threshold.

Learning an Entity-oriented Feature Space. Firstly, we are to learn an entity-oriented feature space, where the distance between representations reflects entity semantic similarity, i.e., representations from the same entity class have higher similarity while keeping the distance from other classes. This feature space is realized by learning a non-linear mapping $F(\cdot)$ on the output representations $\mathbf{h}$ of PLM. We adopt cosine similarity as the similarity metric and train with the Supervised Contrastive Loss (Khosla et al., 2020):

$$L_{SCL} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{e^{s(\mathbf{z}_i, \mathbf{z}_p)/\tau}}{\sum_{a \in A(i)} e^{\mathbf{s}(\mathbf{z}_i, \mathbf{z}_a)/\tau}} \tag{1}$$

where $\mathbf{z} = F(\mathbf{h})$ denotes the representation after the mapping and $s(\cdot)$ is the cosine similarity.

Here, we apply contrastive learning only on the entity classes, thus we define:

$$\begin{aligned} I &= \{i \mid i \in Index(\mathcal{D}_t^{tr}),\ y_i \neq \text{“}O\text{”}\} \\ A(i) &= \{j \mid j \in Index(\mathcal{D}_t^{tr}),\ j \neq i\} \\ P(i) &= \{p \mid p \in A(i),\ y_p = y_i\} \end{aligned} \tag{2}$$

where the anchor set I **only includes entity tokens**. We train with L_{SCL} in the first K epochs, improving the representations of entities and obtaining an entity-oriented feature space.

[2]We set a small K because the class number can be large.

Calculating an entity threshold for anchors and positive samples selection. Based on the entity-oriented feature space, we are to dynamically select possible entity clusters in "O" and further optimize their representations via contrastive learning. This selection is realized by a dynamically adjusted *entity threshold*.

Specifically, we first define the *class similarity* S_c as the average of exemplar similarities inside each class:

$$S_c = \frac{1}{|\mathcal{M}_c|} \sum_{\substack{x_i, x_j \in \mathcal{M}_c, \\ x_i \neq x_j}} s(F(h(x_i)), F(h(x_j))) \quad (3)$$

Then, we sort the *class similarity* of all classes and choose the median as the *entity threshold* $\mathcal{T}_{ent}$ (here we simply choose the median for a modest threshold):

$$\mathcal{T}_{ent} = Sorted(\{S_1, \ldots, S_{|\mathcal{C}_t^{all}|}\})[i], i = \frac{|\mathcal{C}_t^{all}|}{2} \quad (4)$$

During contrastive learning for "O", we re-calculate $\mathcal{T}_{ent}$ before each epoch to dynamically adjust the threshold based on convergence degree.

Contrastive Learning for "O" with the entity threshold Based on entity threshold $\mathcal{T}_{ent}$, we then apply the entity-aware contrastive learning for "O" with auto-selected anchors and positive samples. Specifically, we re-define Eq.2 as:

$$\begin{aligned} I_O &= \{i \mid \exists j \neq i, y_j = y_i = \text{“}O\text{”}, s(\mathbf{z}_i, \mathbf{z}_j) > \mathcal{T}_{ent}\} \\ P_O(i) &= \{p \mid p \neq i, y_p = \text{“}O\text{”}, s(\mathbf{z}_i, \mathbf{z}_p) > \mathcal{T}_{ent}\} \\ A_O(i) &= P_O(i) \cup \{n \mid y_n \in \mathcal{C}_{t,new}\} \end{aligned} \quad (5)$$

Then, we define the entity-aware contrastive loss of "O" by adopting Eq.1 with the definition in Eq.5:

$$L_{SCL,O} = L_{SCL}(I_O, P_O, A_O) \quad (6)$$

In the last $N - K$ epochs, we jointly optimize the representations of entities and "O" by:

$$L = L_{SCL,O} + L_{SCL} \quad (7)$$

4.2 Relabeling Old Entity Classes

In order to further retain the class discrimination of old classes, we propose two distance-based relabeling strategies to recognize and relabel the unlabeled old-class entities in "O". These two strategies are designed to make use of the previous model $\mathcal{A}_{t-1}$ and the exemplar set $\mathcal{M}$.

Relabeling with Prototypes. This strategy relabels "O" samples based on their distance to the class prototypes. Specifically, we first calculate the prototype of each class based on the representations of exemplars from the old model $\mathcal{A}_{t-1}$.

$$\mathbf{p}_c = \frac{1}{|\mathcal{M}_c|} \sum_{x \in \mathcal{M}_c} h_{t-1}(x) \quad (8)$$

Then, we define a relabeling threshold, denoted as the *prototype relabeling threshold*, by calculating the lowest similarity of all exemplars with their prototypes:

$$\mathcal{T}h_{proto} = \beta \cdot \min_{\substack{(x,y) \in \mathcal{M}_c \\ c \in \mathcal{C}_{t,old}}} \{s(h_{t-1}(x), \mathbf{p}_y)\} \quad (9)$$

where β is a hyper-parameter to control the relabeling degree. Next, for each "O" sample x_i in $\mathcal{D}_t^{tr}$, we relabel it only if its highest similarity to prototypes is larger than $\mathcal{T}h_{proto}$:

$$\begin{aligned} \mathcal{S} &= \{s(h_{t-1}(x_i), \mathbf{p}_c) \mid c \in \mathcal{C}_{t,old}\} \\ y_i &= \arg\max_c \mathcal{S}, \quad if \max \mathcal{S} > \mathcal{T}h_{proto} \end{aligned} \quad (10)$$

Relabeling with Nearest Neighbors. In this approach, we relabel "O" samples based on their distance to the exemplars of each class. Similarly, we define the *NN relabeling threshold* $\mathcal{T}h_{NN}$ as:

$$\mathcal{T}h_{NN} = \beta \cdot \min_{\substack{(x_i,x_j) \in \mathcal{M}_c \\ c \in \mathcal{C}_{t,old}}} \{s(h_{t-1}(x_i), h_{t-1}(x_j))\} \quad (11)$$

For each "O" sample x_i, we then relabel it with $\mathcal{T}h_{NN}$ by:

$$\begin{aligned} \mathcal{S} &= \{s(h_{t-1}(x_i), h_{t-1}(x_c)) \mid x_c \in \mathcal{M}_c, c \in \mathcal{C}_{t,old}\} \\ y_i &= \arg\max_c \mathcal{S}, \quad if \max \mathcal{S} > \mathcal{T}h_{NN} \end{aligned} \quad (12)$$

Since the class discrimination of old entity classes keep declining during incremental learning, the older task needs a lower threshold for relabeling sufficient samples. Therefore, we set $\beta_i = 0.98 - 0.05 * (t - i)$ for each old task i, where t is the current step.

4.3 Classifying with NCM Classifier

To make full use of the learnt representations, we adopt the Nearest Class Mean (NCM) classifier used in (Rebuffi et al., 2017) for classification, which is also widely applied in few-shot learning (Snell et al., 2017). For each sample x, the class prediction is calculated by:

$$y^* = \arg\max_{c \in \mathcal{C}_t^{all}} s(h_t(x), \mathbf{p}_c) \quad (13)$$

where $\mathbf{p}_c$ is the prototype of class c calculated with the exemplars as the same in Eq.8.

5 Experiment

Previous works (Monaikul et al., 2021; Xia et al., 2022; Wang et al., 2022) on class-incremental

Methods	Step 0	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10
Full Data	75.45	72.62	71.72	69.39	68.92	68.59	67.55	66.92	66.50	66.83	66.33
LwF	75.56	56.98	48.11	40.08	34.30	33.40	29.37	31.63	27.30	30.14	24.98
SCR	75.14	57.39	48.73	45.47	42.76	40.94	37.75	37.49	33.59	34.51	29.54
iCaRL	74.89	55.76	51.47	46.72	44.98	43.85	42.63	41.91	40.54	43.33	42.27
Con. NER	75.62	55.29	42.65	35.92	32.55	30.55	26.20	27.90	25.37	28.23	25.17
Con. NER*	75.63	59.89	49.82	42.23	36.02	36.44	33.92	32.15	31.09	31.68	28.05
Ours (NN)	75.73	**65.42**	62.17	56.98	55.55	52.79	51.10	49.85	47.15	49.40	47.59
Ours (Proto)	75.73	64.98	**62.19**	**57.08**	**55.56**	**54.47**	**52.90**	**52.16**	**51.05**	**52.73**	**51.16**

Table 2: Main results of the proposed method and baselines on Few-NERD dataset. For each model, we repeat incremental learning experiments on three different task orders and report the averages of the micro-f1 scores. Detailed results on each task order can be found in Appendix A.4.

NER conducted experiments on the CoNLL 2003 (Sang and De Meulder, 2003) and OntoNotes 5.0 (Weischedel et al., 2013) datasets. However, due to the limited class number of these datasets, the class number introduced in each step and the total number of incremental steps in these datasets are limited. For instance, there are only four classes in the CoNLL03 dataset, thus only one class is introduced in each step and there are only four incremental tasks to repeat. In more realistic situations, multiple classes can be introduced in each step (e.g., a set of product types) and there can be a larger number of incremental steps.

In this work, we provide a **more realistic and challenging benchmark** for class-incremental NER based on the Few-NERD dataset[3] (Ding et al., 2021), which contains 66 fine-grained entity types. Following the experimental settings of previous works (Rebuffi et al., 2017; Wu et al., 2019; PourKeshavarzi et al., 2021; Madaan et al., 2021), we randomly split the 66 classes in Few-NERD into 11 tasks, corresponding to 11 steps, each of which contains 6 entity classes and an "O" class. The training set and development set of each task $\mathcal{T}_t$ contains sentences only labeled with classes of the current task. The test set contains sentences labeled with all learnt classes in task $\{0 \dots t\}$. The statistics and class information of each task order can be found in Appendix A.6.

5.1 Experimental Settings

The main experiments in this work are conducted on the Few-NERD datasets. Specifically, for each model, we repeat incremental experiments on three different task orders and report the averages of the micro-f1 score. To further illustrate the proposed method on different datasets, we also conduct experiments on the OntoNotes 5.0 dataset (by splitting 18 classes into 6 tasks) in the same way.

We compare our method with 7 comparable baselines. **Full Data** denotes Bert-tagger (Devlin et al., 2019) trained with datasets annotated with both old and new classes, which can be regarded as an upper bound. **LwF** (Li and Hoiem, 2017) is a regularization-based incremental learning method. **iCaRL** (Rebuffi et al., 2017) is a typical rehearsal-based representation learning method. **SCR** (Mai et al., 2021) is also an effective rehearsal-based contrastive learning method with an NCM classifier. **Con. NER** or **Continual NER** (Monaikul et al., 2021) is the previous SOTA method on class-incremental NER. **Con. NER*** is Continual NER trained with exemplars and tested with NCM classifier. For our method, **Ours (NN)** and **Ours (Proto)** denote our method using NN-based and prototype-based strategies, respectively.

The implementation details of baselines and our method, the dataset details, and the detailed macro-f1 and micro-f1 results of different task orders can be found in Appendix A.1, A.4, A.5 and A.6.

5.2 Main Results

Table 2 show the results of the proposed method and baselines on the Few-NERD dataset. From the results, we can observe that: (1) The results of **Full Data**, which leverages all class annotations for training, is relatively consistent. (2) Although **Continual NER** has shown good performance on CoNLL03 or OntoNotes 5.0 datasets, its performance is limited on this more challenging benchmark, when encountering multiple classes and more incremental steps. (3) The proposed method shows up to 10.62% improvement over baselines, and consistently exceeded the baselines

[3]https://ningding97.github.io/fewnerd/

Methods	Step0	Step1	Step2	Step3	Step4	Step5
Full Data	92.42	90.93	89.71	89.41	88.30	87.40
LwF	92.15	82.79	74.60	65.96	57.90	56.37
SCR	92.04	84.80	80.53	78.67	76.48	76.05
iCaRL	**92.49**	84.89	79.65	80.03	78.76	77.14
Con. NER	92.02	79.90	69.89	69.48	64.47	60.49
Con. NER*	92.09	80.23	72.23	70.88	66.65	61.48
Ours (NN)	92.39	**87.72**	86.60	**86.83**	85.64	82.98
Ours (Proto)	92.39	87.41	**86.87**	86.42	**85.63**	**83.18**

Table 3: Main results of the proposed method and baselines on OntoNotes 5.0 dataset. We report the averages of the micro-f1 scores of three task orders.

by about 10% even in the later steps, verifying the advantages of the learnt representations. (4) The prototype-based relabeling strategy is more stable than the NN-based strategy especially in the later steps. A possible reason is that using the mean vector of exemplars for relabeling is more reliable than using each of the exemplars.

We also conduct experiments on the OntoNotes dataset to further illustrate our method. As shown in Table.3, the results of all methods improve on the less challenging setting, yet the proposed method still significantly outperforms all the baselines.

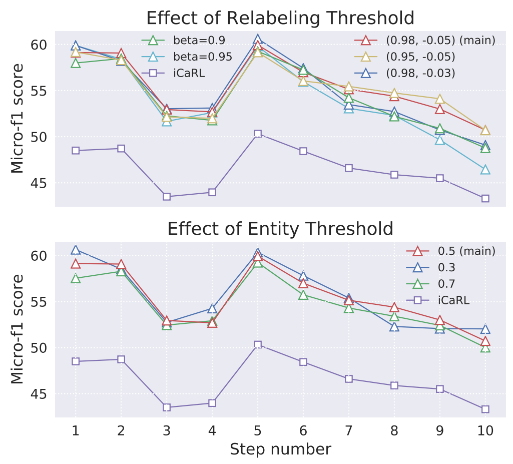


Figure 5: Effect of different relabeling threshold $\mathcal{T}_{proto}$ and entity threshold $\mathcal{T}_{entity}$. All results are based on the prototype-based method on Few-NERD task order 1.

5.3 Ablation Studies

To further illustrate the effect of each component on our method, we carry out ablation studies on Few-NERD task order 1 and show the micro-f1 and macro-f1 results in Figure 6. Here, *Normal SCL* means applying the normal SupCon Loss on both entity classes and "O" without the entity-aware contrastive learning. Similarly, *Normal SCL w/o "O"* means applying the normal SupCon Loss only on

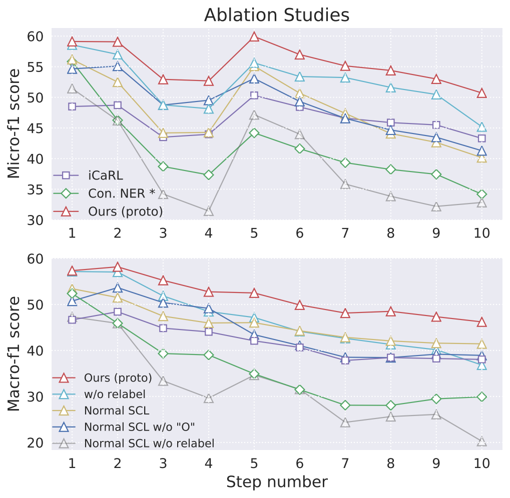


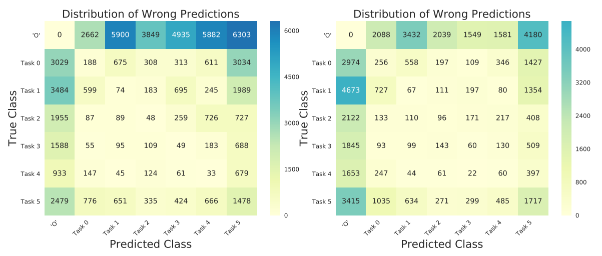
Figure 6: Ablation studies on each component of the proposed method on Few-NERD task order 1. The upper and the bottom figure shows the micro-f1 and macro-f1 score, respectively.

entity classes. *Normal SCL w/o relabeling* means applying the normal SupCon Loss without relabel (not using any of our methods). (Both *Normal SCL* and *Normal SCL w/o "O"* adopt prototype-based relabeling) *w/o relabel* denotes using the entity-aware contrastive learning without relabeling.

From the result, we can see that: (1) Both the relabeling strategy and entity-aware contrastive learning contributes to high performance. (2) The performance of normal SCL without the entity-aware contrastive learning and the relabeling strategy is even worse than iCaRL, indicating that inappropriately learning "O" representations can harm performance. (3) Comparing the micro-f1 and macro-f1 results, we find that the relabeling strategy contributes less to the micro-f1 results. As the micro-f1 results are dominated by head classes with a larger amount of data, we deduce that entity-aware contrastive learning is more useful for head classes (which also appears more in "O"). Also, as the relabeling strategy is based on the distance between representations, the results indicate its effectiveness for both head classes and long-tailed classes.

5.4 Effect of Threshold Selection

Fig.5 shows the results of different hyperparameter choices for threshold calculation. The upper figure refers to the relabeling threshold $\mathcal{T}h_{proto}$, which we set $\beta_i = 0.98 - 0.05 * (t - i)$ for each task t in step i. In this experiment, we tried different strategies for setting the threshold (*bata=0.9* means $\beta = 0.9$, *(0.95,-0.05)* means $\beta_i = 0.95 - 0.05 * (t - i)$). We find that the performance is relatively



(a) The proposed method (b) Full Data

Figure 7: Distributions of prediction errors of the proposed method and the **Full Data** baseline in step 6. Compared to Fig.2, the confusion between "O" and entities is largely mitigated, even comparable to **Full Data**.

stable w.r.t different choices, and a lower threshold seems more helpful. [4]

In the bottom figure, we also tested for different $\mathcal{T}_{entity}$ choices, which we simply set as the median (0.5) of class similarities. As seen, the performance is also robust to different choices.

5.5 Mitigating the Unlabeled Entity Problem

To demonstrate the effectiveness of the proposed method on mitigating the Unlabeled Entity Problem, we conduct the same experiments as in Section 3. Comparing Fig.7 to Fig.2, we can see that the proposed method largely reduce the confusion between "O" and entities, contributing to much fewer error predictions. Comparing Fig.8 to Fig.3 (b), we find that the proposed method learns discriminative representations for the old classes despite the impact of incremental learning.

6 Related Works

6.1 Class-incremental Learning

There are two main research lines of class-incremental learning: (1) Rehearsal-based methods are the most popular and effective methods, which keeps a set of exemplars from the old classes. Typical researches include regularization-based methods that reduces the impact of new classes on old classes (Chaudhry et al., 2019; Riemer et al., 2019); methods that aim to alleviate the biased prediction problem in incremental learning (Zhao et al., 2020; Hou et al., 2019); methods that replay with generative exemplars (Kamra et al., 2017; Ostapenko et al., 2019; Ramapuram et al., 2020). (2) Regularization-based methods aim to regularize the model learning without maintaining any memory. Typical methods include

[4]We further test the relabeling accuracy in Appendix A.3.

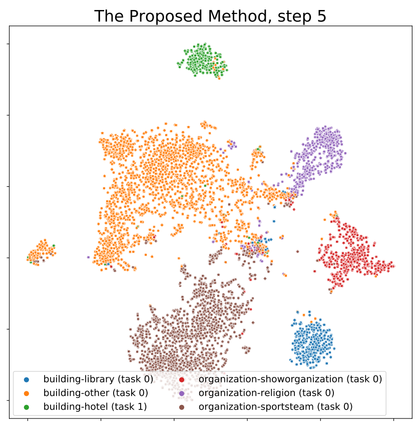


Figure 8: Visualization of the representation of the proposed method in step 5, as a comparison to Fig.3 (b). The proposed method learns much more discriminative representations for the old classes.

knowledge distillation-based methods (Zhang et al., 2020; Hou et al., 2019) and gradient-based methods that regularize the model parameters (Kirkpatrick et al., 2017; Schwarz et al., 2018; Aljundi et al., 2018). These methods, when directly applied to incremental-NER, do not consider the Unlabeled Entity Problem, thus show limited performance. Nonetheless, these methods are essential references for us to improve class-incremental NER.

6.2 Class-incremental Learning for NER

Previous works have explored the class-incremental problems in NER (Monaikul et al., 2021; Wang et al., 2022; Xia et al., 2022). These methods generally care about maintaining old knowledge. Monaikul et al. (2021) propose a knowledge distillation-based method for learning old classes in "O". Wang et al. (2022) and Xia et al. (2022) propose method to generate synthetic samples for old classes. Among these studies, we are the first to comprehensively investigate the Unlabeled Entity Problem and propose solutions that benefits both the old classes and new classes. We also provide a more realistic benchmark.

6.3 Learning "O" for NER

Many previous works have also explored "learning 'O'" in NER. There are three typical lines of work: (1) Tong et al. (2021) solves the "O" problem for few-shot NER. It proposes a multi-step undefined-class detection approach to explicitly classify potential entity clusters in "O", which is similar to our core idea. Different from (Tong et al., 2021), we integrate the clustering and detection of potential entity clusters implicitly into representation learning, through a novel design for anchor and positive selection in contrastive learning. To our best knowledge, we are the first to explore the "O" problem in NER with representation learning. (2) There also exist other works that study the unlabeled entity problem (Li et al., 2021, 2022) in NER. These works focus more on avoiding false-negative samples during training and are not specifically designed for distinguishing potential entity classes. (3) The 'O' problem is also considered by previous works in class-incremental NER (Monaikul et al., 2021; Wang et al., 2022), yet they mainly focus on distilling old knowledge from "O". Our work provides new insight on the "O" problem (or unlabeled entity problem) by comprehensively considers the old classes and new classes, with detailed experimental results.

7 Conclusion

In this work, we first conduct an empirical study to demonstrate the significance of the Unlabeld Entity Problem in class-incremental NER. Based on our observations, we propose a novel and effective representation learning method for learning discriminative representations for "O" and unlabeled entities. To better evaluate class-incremental NER, we introduce a more realistic and challenging benchmark. Intensive experiments demonstrate the effectiveness and show the superior of the proposed method over the baselines.

8 Limitations

The limitations of this work are: (1) In this work, we expect to consider more realistic and more applicable settings for class-incremental NER. Therefore, we consider the Unlabeled Entity Problem and provide a more realistic benchmark based on 66 fine-grained entity types. However, there remain some more serious situations unsolved in this work. First, the entity classes in each step might not be disjoint. For example, a new entity type "Director" might be included in an old entity type "Person". This problem is referred to as the coarse-to-fine problem existing in emerging types of NER. Second, the amount of data or labeled data introduced in each step can also be limited, referring to the few-shot class-incremental problem. Therefore, the proposed method can be further improved to solve these problems. Third, the current version of the proposed method cannot handle the nested NER or contiguous NER problems. In the current version, we simply followed typical works in NER and adopted the sequence labeling scheme to model the NER task, which is not suitable for more complicated NER tasks. Nonetheless, as the proposed representation learning and re-labeling methods are agnostic to the formation of representations, we believe our method can also be adapted to a span-level version, which might be future works. (2) The proposed method is a rehearsal-based method that requires keeping exemplar sets for each class. Although the number of exemplars for each class is really small, we believe there can be more data-efficient solutions that totally avoid the need of memorizing data and also achieve good results. (3) The proposed method includes several hyper-parameters such as the entity threshold $\mathcal{T}_{entity}$, relabeling threshold $\mathcal{T}h_{NN}$ and $\mathcal{T}h_{proto}$. Although we have shown that the choice of thresholds is relatively robust (Sec.5.4), it still requires efforts to explore the most suitable thresholds when applied to other datasets or situations. There can be further work to improve this problem by formulating an automatic threshold searching strategy.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by the National Natural Science Foundation of China (No.62076069,62206057,61976056), Shanghai Rising-Star Program (23QA1400200), and Natural Science Foundation of Shanghai (23ZR1403500).

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.

Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. 2021. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227.

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. 2018. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a unified classifier incrementally via rebalancing. *Computer Vision and Pattern Recognition*.

Nitin Kamra, Umang Gupta, and Yan Liu. 2017. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

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.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Yangming Li, lemao liu, and Shuming Shi. 2021. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*.

Yangming Li, Lemao Liu, and Shuming Shi. 2022. Rethinking negative sampling for handling missing entity annotations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7188–7197, Dublin, Ireland. 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. *Advances in neural information processing systems*, 30.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. 2021. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*.

Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. 2021. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3599.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. 2020. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*.

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.

Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. 2019. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11321–11329.

Mozhgan PourKeshavarzi, Guoying Zhao, and Mohammad Sabokrou. 2021. Looking back on learned experiences for class/task incremental learning. In *International Conference on Learning Representations*.

Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. 2020. Lifelong generative modeling. *Neurocomputing*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. 2019. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*.

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

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou, and Juanzi Li. 2021. Learning from miscellaneous other-class words for few-shot named entity recognition. 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 6236–6247, Online. Association for Computational Linguistics.

Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. 2021. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9385–9394.

Rui Wang, Tong Yu, Handong Zhao, Sungchul Kim, Subrata Mitra, Ruiyi Zhang, and Ricardo Henao. 2022. Few-shot class-incremental learning for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–582, Dublin, Ireland. Association for Computational Linguistics.

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

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382.

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, Dublin, Ireland. Association for Computational Linguistics.

Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. 2020. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13208–13217.

A Appendix

A.1 Implementation Details

We implemented the proposed method and all baselines based on the *bert-base-cased* pretrained model using the implementation of huggingface transformers [5]. For our method, we implement the SupCon loss based on the implementation in the *SupContrast* library[6]. For LwF and iCaRL, we follow the implementations of (Masana et al., 2020) [7]. For SCR, we follow the implementation of the *online-continual-learning* library[8]. There is no public source of Continual NER, so we implement based on the paper (Monaikul et al., 2021) and report the results of our implementation. At each step, we trained the model for 16 epochs and selected the best model on the dev set. For all methods, we use a learning rate of 5e-5, batch size of 16 and the max sequence length of 128. For our method, we start entity-aware contrastive learning for "O" with $L_{SCL,O}$ at the 10-th epoch and train it for 6 epochs at each step. We conducted all experiments on on NVIDIA GeForce RTX 3090.

Construction of the exemplar set For all rehearsal-based method, we keep 5 exemplars for each class, each of which consist of one entity word and its context. The exemplar words of each class are selected by picking the most high-frequency words of each class in the dataset. For each exemplar word, we randomly pick one sentence that contains this word as its context. We use the same exemplar set for all methods.

A.2 Performance on Old and New Classes

In figure 9, we show the performance change of different methods on old classes and new classes. As seen, the proposed method can maintain the performance of old classes in a higher degree, which mainly attributes to the relabeling strategy. Meanwhile, the entity-aware contrastive learning method also helps to keep the discrimination of old classes in "O". Also, the proposed method is more effective on learning the new classes than baseline methods, with a highest improvement of 6.01% in the last step. These results indicate the effectiveness of entity-aware contrastive learning, which helps learn fine-grained and entity-aware representations for "O", preventing the potential classes from confusing with "O" and other similar classes.

[5]https://github.com/huggingface/transformers

[6]https://github.com/HobbitLong/SupContrast

[7]https://github.com/mmasana/FACIL

[8]https://github.com/RaptorMai/online-continual-learning

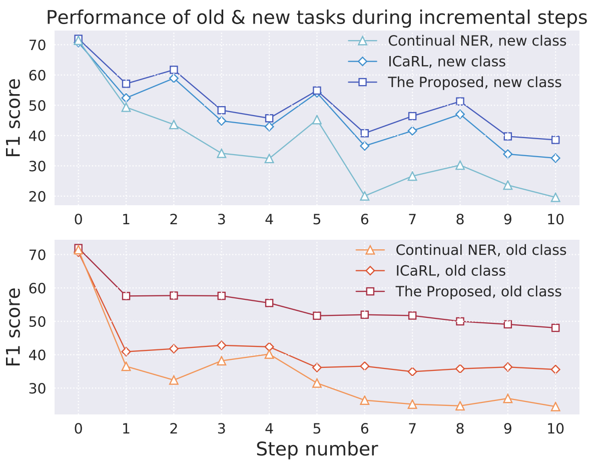


Figure 9: The performance change on old classes and new classes during incremental learning.

Steps	Precision	Recall	Micro-f1
Prototype-based relabeling			
Step 1	56.61	99.04	72.04
Step 4	62.24	84.29	71.61
Step 7	74.92	70.82	72.81
Prototype-based relabeling ($\beta = 0.9$)			
Step 1	52.52	99.16	68.67
Step 4	61.61	73.72	67.12
Step 7	79.40	67.63	73.05
NN-based relabeling			
Step 1	60.08	98.78	74.71
Step 4	64.81	81.32	72.14
Step 7	74.56	76.55	75.54

Table 4: Relabeling statistics of different strategies.

A.3 Relabeling Statistics

We examine the token-level micro-f1 scores of different relabeling strategies based on the gold labeled data of each step on Few-NERD task order 1. The results are shown in Table 4. We find that: (1) The proposed relabeling strategies can achieve acceptable relabeling accuracy, which greatly helps for retaining the knowledge of old classes and improving representation learning for potential classes. (2) Using a fixed β leads to higher recall and lower precision in earlier steps, as well as lower recall in later steps. This might because the convergence degree of old classes decrease in later step, thus a fixed threshold will relabel limited number of old class samples. (3) Compared to prototype-based method, the NN-based method has slightly lower recall and higher precision in earlier steps, which might correspond to its slightly higher performance in earlier steps on Few-NERD task order 1 (Table 5).

A.4 Detailed Results on Few-NERD

The detailed results on the Few-NERD datasets are shown in Table 5 (task order 1), Table 6 (task order 2), Table 7 (task order 3). In each table, the numbers in black denote the micro-f1 scores and the numbers in green denote the macro-f1 scores. The proposed method surpass all baseline methods in all task orders.

A.5 Detailed Results on OntoNotes 5.0

We also conduct experiments on OntoNotes 5.0[9] by randomly splitting the 18 entity classes into 6 tasks, each of which contains 3 entity classes and a "O" class. The detailed results on the OntoNotes datasets are shown in Table 8 (task order 1), Table 9 (task order 2), Table 10 (task order 3). In each table, the numbers in black denote the micro-f1 scores and the numbers in green denote the macro-f1 scores. The proposed method surpass all baseline methods in all task orders.

A.6 Dataset Details

The dataset details of Few-NERD are shown in Table 11 (task order 1), Table 12 (task order 2), Table 13 (task order 3). The dataset details of OntoNotes 5.0 are shown in Table 14 (task order 1), Table 15 (task order 2), Table 16 (task order 3).

[9]https://catalog.ldc.upenn.edu/license/ldc-non-members-agreement.pdf

Methods	Step 0	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10
Full Data	73.30 / 71.51	68.73 / 66.21	68.02 / 67.14	66.21 / 65.79	66.60 / 65.20	68.79 / 61.92	68.40 / 61.83	67.29 / 60.24	67.44 / 60.83	66.61 / 59.93	66.35 / 60.32
LwF	73.47 / 71.25	47.63 / 42.69	42.96 / 40.58	31.85 / 31.19	28.98 / 27.01	39.73 / 18.97	34.64 / 19.57	37.66 / 17.31	34.16 / 18.49	32.23 / 17.23	25.47 / 15.25
SCR	73.21 / 70.56	50.74 / 46.38	51.44 / 49.86	40.41 / 42.67	41.73 / 39.75	49.07 / 37.27	45.52 / 35.15	42.88 / 32.98	40.50 / 29.90	35.80 / 29.12	30.47 / 27.86
iCaRL	72.69 / 70.71	48.50 / 46.66	48.72 / 48.41	43.50 / 44.84	43.97 / 44.04	50.32 / 42.09	48.43 / 40.65	46.60 / 37.82	45.88 / 38.45	45.5 / 38.24	43.30 / 38.05
Con.NER	73.42 / 71.45	47.02 / 42.93	43.09 / 39.87	35.86 / 34.82	36.47 / 34.60	44.79 / 30.73	37.49 / 24.90	37.08 / 20.91	36.43 / 22.20	35.24 / 22.53	27.04 / 20.33
Con.NER*	73.56 / 71.90	55.84 / 52.37	46.27 / 45.91	38.71 / 39.33	37.34 / 39.02	44.20 / 34.91	41.61 / 31.49	39.34 / 28.10	38.23 / 28.06	37.44 / 29.48	34.19 / 29.91
Ours (NN)	**74.04** / **71.94**	**59.22** / **57.40**	**59.08** / **58.36**	52.18 / 54.64	**53.24** / **53.35**	**60.51** / **52.87**	**57.81** / **50.60**	**55.41** / **48.48**	53.61 / 47.97	49.44 / 45.30	46.93 / 44.80
Ours (Proto)	**74.04** / **71.94**	59.12 / 57.35	59.07 / 58.18	**52.94** / **55.24**	52.69 / 52.75	59.93 / 52.50	56.99 / 49.89	55.14 / 48.13	**54.39** / **48.50**	**53.00** / **47.33**	**50.72** / **46.21**

Table 5: Detailed results of Few-NERD task order 1. The numbers in black are the micro-f1 scores and the numbers in green are the macro-f1 scores.

Methods	Step 0	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10
Full Data	82.26 / 68.47	78.9 / 68.24	77.47 / 68.11	73.02 / 63.38	71.57 / 63.26	69.10 / 61.19	68.08 / 60.72	67.67 / 61.05	67.32 / 60.31	66.48 / 60.12	66.29 / 60.26
LwF	82.33 / 68.15	67.39 / 53.57	55.59 / 42.70	49.60 / 34.89	37.25 / 26.02	30.90 / 22.06	28.86 / 18.49	32.30 / 16.71	29.35 / 15.63	27.86 / 14.24	23.57 / 12.35
SCR	81.95 / 66.04	67.68 / 53.06	50.66 / 43.86	50.98 / 41.92	42.82 / 31.19	34.78 / 29.55	34.11 / 29.48	37.90 / 30.00	31.68 / 26.78	29.40 / 24.95	23.82 / 22.80
iCaRL	81.91 / 67.28	66.01 / 52.06	56.73 / 44.50	51.19 / 40.92	45.16 / 39.07	40.64 / 37.61	41.05 / 38.34	41.52 / 37.10	41.13 / 37.22	41.58 / 37.64	41.50 / 38.07
Con.NER	**82.44** / **68.91**	66.55 / 53.03	42.94 / 35.11	38.07 / 31.56	29.65 / 26.61	24.31 / 23.20	22.07 / 19.22	25.91 / 18.13	24.59 / 17.80	23.62 / 15.96	23.35 / 17.73
Con.NER*	82.38 / 68.88	68.89 / 55.10	56.38 / 44.16	48.42 / 39.21	37.71 / 35.69	33.76 / 32.16	33.81 / 30.21	29.43 / 25.66	29.84 / 28.06	28.89 / 26.39	26.75 / 26.92
Ours (NN)	82.32 / 68.29	**73.27** / **60.79**	67.96 / 55.97	**62.07** / **52.71**	**57.49** / **50.28**	47.27 / 46.88	48.95 / 47.08	52.33 / 45.34	49.75 / 46.58	49.44 / 44.72	49.75 / 44.17
Ours (Proto)	82.32 / 68.29	72.97 / 59.84	**68.38** / **56.07**	61.64 / 52.36	56.75 / 49.74	**51.30** / **47.26**	**53.33** / **47.73**	**53.44** / **47.67**	**52.75** / **46.72**	**52.20** / **46.50**	**52.10** / **46.06**

Table 6: Detailed results of Few-NERD task order 2. The numbers in black are the micro-f1 scores and the numbers in green are the macro-f1 scores.

Methods	Step 0	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10
Full Data	70.79	70.24	69.68	68.94	68.58	67.87	66.16	65.81	64.74	67.40	66.35
	72.21	69.30	68.62	63.63	64.17	63.84	63.30	63.25	62.18	61.09	60.47
LwF	70.87	55.91	45.79	38.79	36.68	29.58	24.61	24.93	18.39	30.32	25.90
	72.12	49.67	36.79	27.59	27.72	24.37	19.31	20.93	18.43	16.06	14.51
SCR	70.25	53.74	44.09	45.02	43.72	38.98	33.62	31.70	28.60	38.33	34.32
	70.47	54.34	43.22	35.97	36.57	33.32	31.56	31.11	28.49	27.76	26.76
iCaRL	70.07	52.78	48.98	45.48	45.81	40.57	38.42	37.60	34.60	42.92	42.00
	70.64	52.63	47.93	40.26	40.20	38.33	38.36	38.41	37.74	37.73	38.25
Con.NER	**70.98**	52.28	41.93	33.84	31.51	22.54	19.05	20.70	15.10	25.85	25.13
	72.82	48.64	36.63	21.51	22.83	20.34	16.07	17.23	15.81	11.25	11.51
Con.NER*	70.95	54.95	46.81	39.57	33.03	31.36	26.34	27.68	25.22	28.71	23.20
	72.71	54.01	44.32	32.84	31.78	29.89	27.08	27.84	26.25	19.18	20.75
Ours (NN)	70.84	**63.77**	**59.47**	**56.68**	55.93	50.58	46.54	41.80	38.09	49.31	46.10
	72.35	**63.73**	**58.15**	51.87	51.64	50.37	47.79	43.92	43.86	44.58	44.62
Ours (Proto)	70.84	62.85	59.12	56.66	**57.23**	**52.18**	**48.38**	**47.91**	**46.02**	**52.99**	**50.68**
	72.35	62.23	57.54	**52.32**	**53.80**	**51.79**	**49.97**	**50.42**	**49.21**	**49.25**	**48.11**

Table 7: Detailed results of Few-NERD task order 3. The numbers in black are the micro-f1 scores and the numbers in green are the macro-f1 scores.

Methods	Step 0	Step 1	Step 2	Step 3	Step 4	Step 5
Full Data	93.71	91.07	90.97	90.38	88.93	87.47
	84.82	82.45	78.11	78.36	77.02	76.97
LwF	93.28	86.63	73.58	73.60	71.70	64.24
	84.74	77.11	63.58	61.68	53.73	52.86
SCR	93.36	89.28	86.10	82.81	81.98	78.47
	81.71	75.59	71.50	69.06	67.28	66.75
iCaRL	93.62	87.78	78.91	79.60	76.52	75.33
	84.21	78.09	65.35	68.64	65.08	65.07
Con.NER	93.16	83.25	70.90	71.59	60.26	63.15
	83.62	71.99	59.90	59.01	50.13	48.28
Con.NER*	93.24	83.53	73.81	72.25	64.03	62.42
	83.46	72.51	60.29	59.04	51.00	52.38
Ours (NN)	93.69	89.23	88.47	**87.55**	**86.45**	83.15
	83.39	78.95	73.49	71.49	**71.28**	70.32
Ours (Proto)	93.69	**89.53**	**88.50**	87.50	86.20	**84.02**
	83.39	**79.84**	**74.33**	**72.92**	70.78	**72.19**

Table 8: Detailed results of OntoNotes task order 1. The numbers in black are the micro-f1 scores and the numbers in green are the macro-f1 scores.

Methods	**Step 0**	**Step 1**	**Step 2**	**Step 3**	**Step 4**	**Step 5**
Full Data	94.92	92.07	90.24	89.94	88.92	87.33
	92.85	78.90	78.54	77.84	77.97	77.27
LwF	95.29	80.10	77.75	78.88	58.99	56.81
	93.19	59.83	60.18	61.11	47.86	46.95
SCR	94.63	81.37	83.97	84.39	83.40	79.76
	91.33	61.60	63.79	63.47	62.55	61.78
iCaRL	**95.34**	84.37	81.09	81.86	82.60	78.91
	92.80	69.30	65.60	65.79	67.94	66.23
Con.NER	94.81	74.22	72.15	72.68	73.37	66.37
	92.13	55.44	52.84	55.07	53.51	51.20
Con.NER*	94.99	74.66	72.80	74.11	74.28	66.09
	92.45	55.63	53.23	56.07	55.70	52.45
Ours (NN)	94.65	**85.87**	86.17	**87.80**	**86.97**	**83.40**
	92.72	**69.93**	69.86	**72.34**	**72.68**	**70.21**
Ours (Proto)	94.65	85.33	**86.79**	87.13	86.71	82.75
	92.72	69.65	**71.23**	71.89	71.72	68.89

Table 9: Detailed results of OntoNotes task order 2. The numbers in black are the micro-f1 scores and the numbers in green are the macro-f1 scores.

Methods	**Step 0**	**Step 1**	**Step 2**	**Step 3**	**Step 4**	**Step 5**
Full Data	88.64	89.64	87.91	87.93	87.05	87.39
	86.15	79.89	76.82	78.15	76.80	76.24
LwF	87.88	81.65	72.46	45.39	43.01	48.05
	85.36	65.78	55.52	44.41	47.94	46.83
SCR	88.13	83.76	71.53	68.83	64.05	69.93
	85.00	68.72	59.03	61.20	60.30	58.20
iCaRL	**92.49**	84.89	79.65	80.03	78.76	77.14
	87.56	72.97	65.95	67.17	67.09	65.86
Con.NER	88.08	82.24	66.63	64.16	59.78	51.94
	85.22	66.33	57.53	51.72	48.32	42.58
Con.NER*	88.03	82.48	70.09	66.28	61.64	55.93
	84.96	66.32	57.84	55.93	50.94	46.77
Ours (NN)	88.82	**88.05**	85.15	**85.15**	83.50	82.38
	86.33	**75.86**	**71.62**	**73.08**	73.04	69.71
Ours (Proto)	88.82	87.35	**85.31**	84.65	**83.99**	**82.78**
	86.33	73.51	70.08	71.21	**73.25**	**69.89**

Table 10: Detailed results of OntoNotes task order 3. The numbers in black are the micro-f1 scores and the numbers in green are the macro-f1 scores.

Task	Entity Class	# Train	# Dev	# Test
1	['building-library', 'organization-showorganization', 'other-award', 'building-other', 'organization-religion', 'organization-sportsteam']	18435	2656	5296
2	['person-politician', 'art-painting', 'event-disaster', 'organization-other', 'product-weapon', 'building-hotel']	18966	2788	10267
3	['event-sportsevent', 'other-chemicalthing', 'art-writtenart', 'product-game', 'location-mountain', 'other-livingthing']	11973	1652	13055
4	['location-island', 'person-scholar', 'building-restaurant', 'other-astronomything', 'building-airport', 'product-other']	9448	1326	15178
5	['location-road/railway/highway/transit', 'other-educationaldegree', 'building-sportsfacility', 'event-election', 'person-actor', 'art-film']	10295	1477	17254
6	['location-other', 'product-ship', 'organization-politicalparty', 'person-soldier', 'location-GPE', 'other-god']	47648	6941	24429
7	['event-attack/battle/war/militaryconflict', 'organization-sportsleague', 'building-theater', 'organization-education', 'product-train', 'other-medical']	13237	1852	25631
8	['event-protest', 'person-other', 'product-car', 'art-other', 'organization-company', 'other-disease']	30899	4416	29014
9	['other-biologything', 'person-artist/author', 'location-bodiesofwater', 'art-broadcastprogram', 'other-language', 'person-athlete']	21794	3114	31036
10	['product-airplane', 'art-music', 'product-software', 'event-other', 'location-park', 'organization-media/newspaper']	11963	1706	31874
11	['other-currency', 'person-director', 'building-hospital', 'other-law', 'organization-government/governmentagency', 'product-food']	9787	1443	32565

Table 11: Details of Few-NERD task order 1.

Task	Entity Class	# Train	# Dev	# Test
1	['location-GPE', 'event-sportsevent', 'organization-showorganization', 'event-attack/battle/war/militaryconflict', 'art-other', 'product-car']	48730	7060	13963
2	['location-bodiesofwater', 'person-scholar', 'person-artist/author', 'person-politician', 'other-livingthing', 'product-airplane']	21523	3183	17477
3	['product-other', 'art-music', 'location-island', 'person-athlete', 'building-airport', 'building-hotel']	15257	2169	19805
4	['person-soldier', 'event-other', 'product-software', 'event-election', 'organization-other', 'organization-politicalparty']	17967	2531	22192
5	['other-award', 'art-film', 'organization-government/governmentagency', 'other-astronomything', 'person-actor', 'person-director']	12258	1792	23841
6	['event-protest', 'building-library', 'art-broadcastprogram', 'other-educationaldegree', 'organization-sportsleague', 'location-other']	11191	1620	25125
7	['product-game', 'event-disaster', 'product-train', 'building-other', 'other-disease', 'building-hospital']	11243	1578	26473
8	['product-ship', 'other-currency', 'art-painting', 'product-weapon', 'organization-sportsteam', 'person-other']	28035	4055	29113
9	['other-god', 'art-writtenart', 'other-chemicalthing', 'organization-education', 'other-medical', 'building-restaurant']	11390	1631	30302
10	['building-sportsfacility', 'building-theater', 'organization-company', 'other-biologything', 'organization-religion', 'other-law']	16806	2333	31810
11	['location-mountain', 'location-road/railway/highway/transit', 'organization-media/newspaper', 'location-park', 'product-food', 'other-language']	11256	1594	32565

Table 12: Details of Few-NERD task order 2.

Task	Entity Class	# Train	# Dev	# Test
1	['organization-other', 'art-film', 'product-weapon', 'building-sportsfacility', 'person-soldier', 'organization-company']]	24344	3377	6860
2	['person-actor', 'product-other', 'person-athlete', 'building-theater', 'organization-media/newspaper', 'event-other']	18381	2603	11276
3	['event-attack/battle/war/militaryconflict', 'organization-showorganization' 'other-livingthing', 'other-language', 'art-broadcastprogram', 'product-ship']	102711	1504	13318
4	['other-award', 'location-road/railway/highway/transit','event-election', 'event-protest', 'person-other', 'art-painting']	25684	3748	18354
5	['other-medical', 'other-chemicalthing', 'product-airplane', 'art-music', 'organization-education', 'location-bodiesofwater']	14767	2148	20948
6	['other-astronomything', 'building-library', 'organization-sportsteam', 'product-food', 'building-restaurant', 'person-politician']	15831	2302	23476
7	['other-biologything', 'location-mountain', 'location-other', 'building-airport', 'other-currency', 'other-educationaldegree']	12075	1723	25509
8	['organization-politicalparty', 'product-car', 'building-hotel', 'location-island', 'person-artist/author', 'other-law']	14431	2089	27139
9	['product-train', 'organization-government/governmentagency', 'other-disease', 'person-director', 'location-park', 'event-disaster']	9792	1388	28142
10	['art-writtenart', 'other-god', 'art-other', 'organization-sportsleague', 'organization-religion', 'location-GPE']	47410	6902	31564
11	['product-game', 'product-software', 'person-scholar', 'event-sportsevent', 'building-hospital', 'building-other']	15549	2157	32565

Table 13: Details of Few-NERD task order 3.

Task	Entity Class	# Train	# Dev	# Test
1	['PRODUCT', 'GPE', 'CARDINAL']	15119	2149	2124
2	['QUANTITY', 'DATE', 'LANGUAGE']	9561	1335	2883
3	['PERSON', 'LAW', 'LOC']	13424	1725	3852
4	['ORDINAL', 'PERCENT', 'EVENT']	3259	460	4002
5	['NORP', 'FAC', 'TIME']	6913	949	4291
6	['MONEY', 'WORK_OF_ART','ORG']	11286	1480	4624

Table 14: Details of OntoNotes 5.0 task order 1.

Task	Entity Class	# Train	# Dev	# Test
1	['ORDINAL', 'PERSON', 'PERCENT']	14323	1814	1919
2	['WORK_OF_ART', 'PRODUCT', 'LAW']	1634	229	2073
3	['CARDINAL', 'EVENT', 'QUANTITY']	6786	904	2711
4	['GPE', 'MONEY', 'TIME']	12823	1887	3718
5	['NORP', 'LANGUAGE', 'DATE']	13090	1820	4318
6	['LOC', 'FAC', 'ORG']	11127	1500	4624

Table 15: Details of OntoNotes 5.0 task order 2.

Task	**Entity Class**	**# Train**	**# Dev**	**# Test**
1	['ORG', 'CARDINAL', 'QUANTITY']	14284	1898	1867
2	['LAW', 'FAC', 'GPE']	11335	1692	2877
3	['DATE', 'LANGUAGE', 'WORK_OF_ART']	9791	1386	3522
4	['PERCENT', 'NORP', 'EVENT']	6927	927	3815
5	['ORDINAL', 'TIME', 'MONEY']	4327	596	3979
6	['PERSON', 'LOC', 'PRODUCT']	13663	1747	4624

Table 16: Details of OntoNotes 5.0 task order 3.