A Class-Rebalancing Self-Training Framework for Distantly-Supervised Named Entity Recognition

Qi Li[1][2], Tingyu Xie[1][2], Peng Peng[2], Hongwei Wang*[1][2] and Gaoang Wang*[1][2]

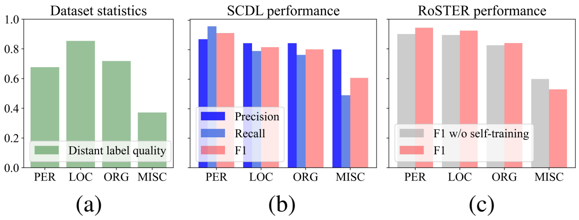
[1]College of Computer Science and Technology, Zhejiang University, China
[2]ZJU-UIUC Institute, Zhejiang University, China
liqi177@zju.edu.cn, 11921049@zju.edu.cn, pengpeng@intl.zju.edu.cn
hongweiwang@zju.edu.cn, gaoangwang@intl.zju.edu.cn

Abstract

Distant supervision reduces the reliance on human annotation in the named entity recognition tasks. The class-level imbalanced distant annotation is a realistic and unexplored problem, and the popular method of self-training can not handle class-level imbalanced learning. More importantly, self-training is dominated by the high-performance class in selecting candidates, and deteriorates the low-performance class with the bias of generated pseudo label. To address the class-level imbalance performance, we propose a class-rebalancing self-training framework for improving the distantly-supervised named entity recognition. In candidate selection, a class-wise flexible threshold is designed to fully explore other classes besides the high-performance class. In label generation, injecting the distant label, a hybrid pseudo label is adopted to provide straight semantic information for the low-performance class. Experiments on five flat and two nested datasets show that our model achieves state-of-the-art results. We also conduct extensive research to analyze the effectiveness of the flexible threshold and the hybrid pseudo label.

1 Introduction

The named entity recognition (NER) task recognizes the location and classification of the named entity. To reduce the reliance on the human annotation of the supervised NER, some works turn to distant supervision to generate large-scale labeled data automatically (Li et al., 2021; Zhou et al., 2022; Jie et al., 2019). Distant supervision is to match the words in sentences with labeled concepts in the collected knowledge bases (Liang et al., 2020). The distantly-labeled data obtained from rule-based matching is accompanied by noisy labels. Previous works in distant supervision mainly focus on the unlabeled entity (Liang et al., 2020; Li et al., 2021) and mislabeled entity (Zhang et al., 2021c).



(a) (b) (c)

Figure 1: The analysis among all entity classes on the CoNLL03 DS-NER benchmark. Green bars represent the class-wise statistics of the distantly-labeled training set. Red bars represent the class-level performance in self-training. (1a) The distant annotation shows different qualities among different classes. (1b) In SCDL, the recall is larger than the precision only in the high-performance class (Class PER, person). (1c) In RoSTER, the low-performance class (Class MISC, miscellaneous) shows performance degradation after self-training.

The class-level imbalanced distant annotation has been underestimated in the distantly supervised named entity recognition (DS-NER), where the distant label of the entity class varies in quality, as shown in Figure 1a. More specifically, the class-wise quality of the distant label depends on the coverage of class-related knowledge bases, and it is hard for the knowledge bases to include all the entities of the semantic-rich class comprehensively. The entity class with the high-quality distant annotation induces *the high-performance class*, and the low-quality distant annotation is related to *the low-performance class*.

While self-training (Hinton et al., 2015) is an effective method in the DS-NER task (Liang et al., 2020; Zhang et al., 2021b; Meng et al., 2021; Zhang et al., 2021c), they have not been thoroughly evaluated on the class-level imbalanced learning. Self-training uses the prediction of the model itself to train again, and effectively uncovers the unlabeled entity. The following works study the mislabeled entity from two aspects: candidate selection and label generation. For example, SCDL (Zhang et al.,

[*] Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2023, pages 11054–11068
July 9-14, 2023 ©2023 Association for Computational Linguistics

2021c) selects consistent and high-confidence data for model training; RoSTER (Meng et al., 2021) generates pseudo labels with the prediction of the contextualized augmented data. However, the initial model in self-training is trained on noisy data and is biased toward the high-performance class, then the subsequent training intensifies the bias and deteriorates the low-performance class, as shown in Figure 1.

In Figure 1b, the selected candidates are dominated by the high-performance class, as the recall is larger than the precision only in the high-performance class. This tendency selection can improve the generalization of the high-performance class, but impair the exploration of other low-performance classes. Actually, a predefined constant threshold struggles to handle the difference in class-wise learning ability (Zhang et al., 2021a), and limits the model to only focus on the high-performance class. In Figure 1c, the generated pseudo label fails to explore the low-performance class during self-training, as the performance degradation occurs in the low-performance class. When the generated pseudo label from the biased model misleads the semantic information of the low-performance class, the iterative update with the guide of pseudo label expands the negative impacts in the low-performance class (Wei et al., 2021).

In this work, we propose a unified self-training framework, called CLIM, to address the **cl**ass-level **im**balance learning in the DS-NER task. For the dominance from the high-performance class, we calculate the current learning ability for each entity class, and adjust the class-wise threshold to improve the candidate selection. For the degradation in the low-performance class, we leverage the semantic information in the distantly-labeled entities, and generate a hybrid pseudo label to improve the label generation. The above two parts of candidate selection and label generation are mutually beneficial. The generated hybrid pseudo label improves the feature capture for the low-performance class by injecting the distant label. And better feature representation improves the exploration of the low-performance class, as more candidates from the low-performance class are selected through the class-wise threshold. The contributions are as follows:

(1) The novel class-rebalancing self-training proposed in this work addresses the imbalance problem in the high-performance and low-performance classes by improving the candidate selection and label generation.

(2) Our method achieves state-of-the-art results on five flat and two nested datasets, and the exhaustive experimental analysis demonstrates the feasibility of addressing the class-level imbalance learning.

(3) Our work with the span-based schema extends the DS-NER task to the nested case, where two noisy nested datasets are additionally generated.

2 Related Work

DS-NER with Self-training. To address the noise interference in the distantly labeled data, the previous works make the strong assumption that no mislabeled entity exists during the distant supervision, and mainly focus on the unlabeled entity (Chen et al., 2021; Zhou et al., 2022; Peng et al., 2019; Cao et al., 2019; Shang et al., 2018; Liang et al., 2020). Among them, self-training shows the effectiveness of uncovering unlabeled entities (Liang et al., 2020; Zhang et al., 2021b). On this basis, some works improve self-training to solve the mislabeled entity, from the two aspects of the candidate selection (Zhang et al., 2021c) and label generation (Meng et al., 2021). However, they take no consideration into the class-level imbalanced performance. The model is biased toward the high-performance class, and the subsequent training intensifies this imbalanced tendency. More importantly, this tendency significantly weakens the exploration of the low-performance class. In this way, our work advances self-training to tackle the class-level imbalanced learning.

Self-Training with Data Augmentation. Self-training (Hinton et al., 2015) consists of both candidate selection and label generation. Specifically, self-training only selects candidate whose largest class probability fall above a predefined threshold; the generated pseudo label comes from the prediction of the model itself. Referred to self-training in the semi-supervised learning (Sohn et al., 2020; Xie et al., 2020), the perturbed inputs with different augmentation is used to decouple the similar predictions on the same input. And also, this data augmentation improve the model robustness and achieves competitive performance (Gao et al., 2021; Chen et al., 2021). Different from the previous works that focus on the classification task with the external task-relevant unlabeled data, our work extends augmentation-driven self-training to the named entity recognition task with only noisy data.

3 Preliminary

Task Definition. Given an input sentence $\boldsymbol{x} = [x_1, x_2, ..., x_n]$ of n tokens, the NER task aims to detect all the entities of different types. Let $\boldsymbol{s} = \{s_1, s_2, \ldots, s_k\}$ be the set of possible spans in $\boldsymbol{x}$. The task of span-based NER is, for each span $s_i \in \boldsymbol{s}$, to produce its label $y_i \in \mathcal{E} \cup \{\epsilon\}$, where ϵ is the non-entity span[1], and $\mathcal{E}$ is the set of pre-defined entity classes. Denote the distantly-supervised NER dataset as $D = \{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$. And $\boldsymbol{y}_m$ is the set of distantly labeled spans, which includes mislabeled entities.

Backbone. For the contextual span representation $H(s_i) = [\mathbf{x}_{\text{START}(i)}; \mathbf{x}_{\text{END}(i)}; \phi(s_i)]$, $\mathbf{x}_{\text{START}(i)}$ and $\mathbf{x}_{\text{END}(i)}$ is the embedding of the start and end token in span s_i, $\phi(s_i)$ is the span width embedding with the random initialization. And the output of classifier G_θ is the probability distribution over entity classes, which is formulated as $F_\theta(s_i) = G_\theta(H(s_i)) \in \mathbb{R}^C$. Among them, θ represents the learnable parameters, and C is the number of entity classes. For simplicity, the probability distribution $F_\theta(s_i)$ is represented as p_i.

Augmentation-Driven Self-Training. The general self-training leverages the model itself to obtain pseudo labels with the loss function: $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\max p_i \geq \tau) \, \text{CE}(\hat{p}_i, p_i)$. Among them, $N = |\boldsymbol{s}|$, τ is the upper bound threshold, CE is the cross-entropy function. And $\hat{p}_i$ is the generated one-hot pseudo label, representing the class $\arg\max p_i$.

With the driven of the data augmentation, the random mask with two different probabilities is used to augment the same input in the attention matrix, which are represented as the strongly-augmented data $\mathcal{S}(s_i)$ and weakly-augmented data $\mathcal{W}(s_i)$. The strong augmentation function $\mathcal{S}$ is implemented with high masking probability to predict the probability distribution over classes, and the weak augmentation $\mathcal{W}$ is related to the low probability to derive the pseudo label. The loss function in self-training thereby has the form:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\max p_{wi} \geq \tau) \, \text{CE}(\hat{p}_{wi}, p_{si}), \quad (1)$$

where $p_{si} = F_\theta(\mathcal{S}(s_i))$, $p_{wi} = F_\theta(\mathcal{W}(s_i))$. And $\hat{p}_{wi}$ is the generated one-hot label, representing the class $\arg\max p_{wi}$.

4 CLIM

We advance self-training to tackle the class-level imbalance learning, with more detailed consideration in the candidate selection and label generation. The overview of our framework is illustrated in Figure 2, and the training algorithm is shown in Algorithm 1.

4.1 Flexible Threshold in Candidate Selection

To alleviate the dominance from the high-performance class, we improve the candidate selection in self-training, by adjusting the threshold for each class. In previous work (Zhang et al., 2021c), the constant threshold is biased towards the high-performance class, where the high-confidence class accounts for the majority of the selected candidates. And the low-performance classes can not be sufficiently explored during self-training, as the constant threshold masks out the samples of these low-performance classes. Therefore, we calculate the current learning ability for each entity class, and adjust the class-wise threshold dynamically to select the candidate. The basic idea agrees with curriculum learning (Zhang et al., 2021a), where candidates are gradually selected according to their learning ability.

The learning ability σ_c of an entity class c can be reflected by the number of entities whose prediction falls into the entity class $N_c = \mathbb{1}(\arg\max p_{wi} = c)$ and above the threshold $N_{\mathcal{T}} = \mathbb{1}(\max p_{wi} > \mathcal{T}(c))$, which is formulated as:

$$\sigma_c = \sum_{\boldsymbol{x} \in D} \sum_{s_i \in \boldsymbol{s}} N_{\mathcal{T}} \cdot N_c. \quad (2)$$

Then the class-wise flexible thresholds $\mathcal{T}(c)$ is formulated as

$$\mathcal{T}(c) = \mathcal{M}(\beta(\sigma_c)) \cdot \tau. \quad (3)$$

First, to reduce the bias of parameter initialization at the early stage, the warm-up process $\beta(\sigma_c) = \sigma_c / \max\{\max_{c'} \sigma_{c'}, \mathcal{N} - \sum_{c'} \sigma_{c'}\}$ is designed, where c' enumerates all entity classes

[1] The span-based schema enumerates all candidate spans, and classifies them into entity classes. Candidate spans that do not belong to any predefined entity classes are called non-entity spans.

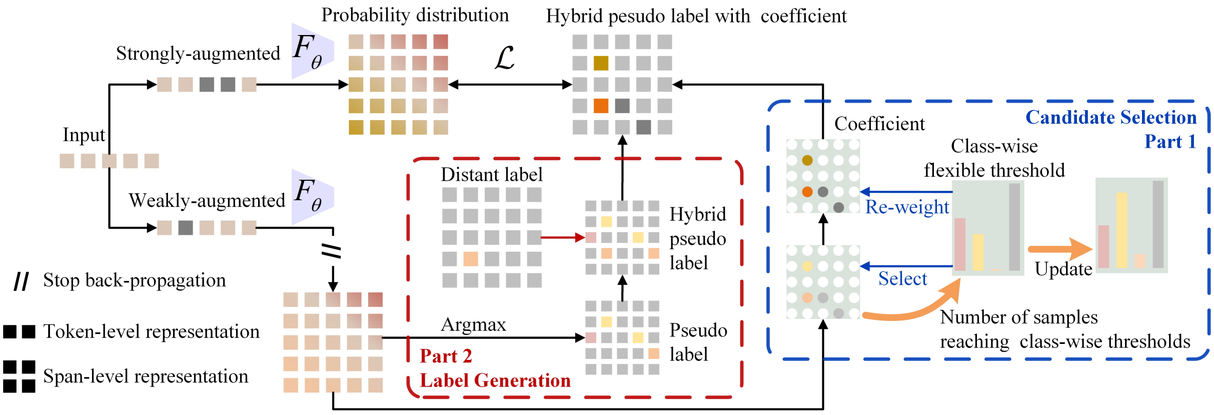


Figure 2: Overview of the proposed framework. Span-level probability distributions are produced, including the strongly-augmented (former) and weakly-augmented (latter) sample. The former is the prediction in the loss computation. In Part 1, the latter is used to select the candidate, of which the confidence is above the class-wise threshold. And also, the latter is converted to the generated one-hot pseudo label in Part 2, and the distant label is further introduced for the generation of the hybrid pseudo label.

and $\mathcal{N}$ represents the number of labeled entities in the distantly-labeled training set. Second, the non-linear mapping function $\mathcal{M}(x) = x/(2-x)$ is designed to make x be more sensitive to a large value and vice versa. In addition, we specially consider the pseudo labeling for the non-entity spans ϵ, since the non-entity spans take the majority of the span set $\boldsymbol{s}$. And we set $\mathcal{T}(c=\epsilon)$ with the same value as the upper bound threshold τ, to filter out non-entity spans ϵ in the early stage. With the class-wise threshold, we update the selection strategy with:

$$\mathbb{1}(\max p_{wi} > \mathcal{T}\left(\arg\max p_{wi}\right)). \tag{4}$$

Further, a re-weighting strategy by inversing the class-wise threshold is employed on each span. The coefficient of the span is defined as:

$$\alpha(c_i) = 2 - \mathcal{T}(c_i), \tag{5}$$

where $c_i = \arg\max p_{wi}$. And we also set the value of $\alpha(c=\epsilon)$ as the upper bound threshold τ, to reduce the attention in the predominant non-entity span.

4.2 Distant supervision in Label Generation

To tackle the degradation in the low-performance class, we advance the label generation, by injecting the semantic information of the distant label. The previous DS-NER work (Meng et al., 2021) leverages the prediction of the model itself to produce the pseudo label. Nevertheless, the model tends to capture information from the high-performance class, and the semantic information captured by the model is severely limited for the low-performance class. Thus the prediction based on the model causes a negative influence on the low-performance class, and the iterative update further expands this negative impact.

The hybrid pseudo label, injecting the distant label, can extraordinarily alleviate the capturing limitation for the low-performance class. More specifically, the distantly-labeled entities from the knowledge base contain useful information, since these knowledge bases are finely collected for the specific entity classes. Finally, the hybrid pseudo label is formulated as follows:

$$h_{wi} = \lambda_p \hat{p}_{wi} + \lambda_y y_i, \tag{6}$$

where y_i is the distant label of span s_i[2].

In different training stages, the model pays different attention to these labels. In the early stage, the model obtains entity features mainly from the distantly-labeled data. When the pseudo label with high confidence is generated, the model is more sensitive to the potential entity behind the noisy training data. Therefore, we dynamically adjust the weights of the distant label y_i and the pseudo label $\hat{p}_{wi}$. Then the dynamic weighting is formulated as follows:

$$\lambda_y = \left(\cos\left(0.5\cdot\pi\left(\hat{t}+1\right)\right)+1\right)^2, \tag{7}$$

$$\lambda_p = \left(\sin\left(0.5\cdot\pi\left(\hat{t}-1\right)\right)+1\right)^2, \tag{8}$$

[2] In practice, we also select the candidate span labeled in the distantly-labeled data, thus expanding the selection strategy in Eq. 4.

Algorithm 1 CLIM Training Algorithm

```
Input: Maximum iteration $T$; Training set $\{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$.
Initialize $\sigma_0(c) = 0$.
while $t = 1, 2, ..., T$ do
    Generate $\boldsymbol{s} = \{s_1, s_2, \ldots, s_i, \ldots\}$ from $\boldsymbol{x}_m$.
    Calculate $p_{si}$ and $p_{wi}$ with different augmentation.
    for $c$ in $\mathcal{E}$ do
        Update threshold $\mathcal{T}(c)$ via Eq. 3.
        Update learning ability $\sigma_c$ via Eq. 2.
    end for
    Selecte candidate via Eq. 4.
    Calculate coefficient $\alpha(c_i)$ via Eq. 5.
    Generate hybrid pseudo label $h_{wi}$ via Eq. 6.
    Back-propagation $\mathcal{L}$ via Eq. 9.
end while
Output: Model parameters.
```

where $\hat{t} = t/t_{total} \in [0, 1]$, t_{total} is the hyperparameter of total training steps.

Finally, integrating the above two advanced components, the loss function in CLIM is represented as:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\Big[\mathbb{1}(\max p_{wi} > \mathcal{T}(c_i))\cdot \alpha(c_i)\text{CE}\left(h_{wi}, p_{si}\right)\Big]. \tag{9}$$

5 Experiment

5.1 Experimental Setup

Dataset. We evaluate on five **flat** benchmarks, including CoNLL03 (Tjong Kim Sang and De Meulder, 2003), Tweet (Godin et al., 2015), OntoNotes5.0 (Weischedel et al., 2013), Wikigold (Balasuriya et al., 2009), and Webpage (Ratinov and Roth, 2009). And we also implement two **nested** benchmarks, including ACE2004 (Doddington et al., 2004) and ACE2005 (Walker et al., 2006). For the flat case, the distant label is generated by matching entities in external knowledge bases, following BOND (Liang et al., 2020). For the nested case, the details of the distant label generation are described in Appendix C. Besides, the dataset statistics are provided in Appendix D.

Baseline. First, **KB Matching** is provided as the reference of the distant supervision quality. Second, we compare our method with the competitive baselines from the following two aspects.

(1) *No Labeling Denoising.* With the combination of the pre-trained language model RoBERTa (Liu et al., 2019) and classifier, both token-based (**RoBERTa-Token**) and span-based schema (**RoBERTa-Span**) are implemented.

(2) *Labeling Denoising.* In this part, we classify these baselines according to whether a self-learning process is used or not. On the one hand, **AutoNER** (Shang et al., 2018) designs modified tagging scheme, **LRNT** (Cao et al., 2019) uses Partial-CRFs with the non-entity sample strategy, **Co-Teaching** (Yu et al., 2019) adopts a advanced sampling strategy, **Comf-MPU** (Zhou et al., 2022) employs a multi-class positive and unlabeled learning method. On the other hand, the works with the self-training strategy are used as the strong baseline. **BOND** (Liang et al., 2020) basically implements the self-training with the teacher-student framework. **BA-CIR** (Zhang et al., 2021b) introduces the casual intervention into the self-training. With the schema of ensemble learning, **SCDL** (Zhang et al., 2021c) and **RoSTER** (Meng et al., 2021) study the mislabeled entity from the candidate selection and the label generation, respectively.

Implementation Detail. For fair comparison, the main result is the average value of 5 runs. We implement our code[3] with PyTorch based on huggingface Transformers[4], and employ the base-size RoBERTa (Liu et al., 2019) to obtain the contextual representation. In addition, the specific experimental settings are listed as follows: the maximum masking probability is 0.05 for the weakly-augmented sample, and 0.2 for the strongly-augmented sample; $\mathcal{T}(c = \epsilon)$ and $\alpha(c = \epsilon)$ are set to 0.9, and the confident threshold τ is set to 0.9; a cosine learning rate decay schedule with no warm-up step and 4 hard restarts is employed; the optimizer is AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$; the training batch size is 16, the maximum sequence length is 128. And more implementation details are listed in Appendix E.

5.2 Main Result

Flat Distantly Labeled NER Task. The span F1 scores on the flat case are listed in Table 1. Our method achieves SOTA results on all five benchmarks. Meanwhile, we conclude the results with the following aspects. (1) For non-denoising methods (the second part of Table 1), the span-based method (RoBERTa-Span) exhibits superior performance over the token-based method (RoBERTa-Token), implying the effectiveness of the span-based schema in DS-NER. (2) For denoising methods (the third part of Table 1), the models with

[3] https://github.com/liqi7797/CLIM/

[4] https://huggingface.co/transformers/

		CoNLL03	Tweet	OntoNote5.0	Webpage	Wikigold	Average
	KB Matching‡	0.714	0.358	0.595	0.525	0.478	0.534
No Lable Denoising	RoBERTa-Token*	0.759	0.465	0.682	0.610	0.526	0.608
	RoBERTa-Span†	0.781	0.525	0.691	0.628	0.526	0.630
Lable Denoising	AutoNER‡ (Shang et al., 2018)	0.670	0.261	0.672	0.514	0.475	0.518
	LRNT‡ (Cao et al., 2019)	0.697	0.238	0.677	0.477	0.462	0.510
	Co-Teaching‡ (Yu et al., 2019)	0.764	0.467	0.680	0.584	0.521	0.603
	Conf-MPU (Zhou et al., 2022)	0.800	-	-	-	-	-
	BOND (Liang et al., 2020)	0.815	0.480	0.684	0.657	0.601	0.647
	BA-CIR (Zhang et al., 2021b)	0.815	0.490	-	0.647	0.615	-
	RoSTER (Meng et al., 2021)	**0.854**	0.445†	**0.696**†	0.544†	0.678	0.643
	SCDL (Zhang et al., 2021c)	0.837	0.511	0.686	0.685	0.641	0.672
	CLIM (Ours)	**0.854**	**0.538**	**0.696**	**0.700**	**0.679**	**0.693**

Table 1: The main results in the flat DS-NER task, via span F1 scores. The baseline marked with ‡ is referred to (Liang et al., 2020), and the baseline marked with * is referred to (Zhang et al., 2021c). The baselines and results marked with † are our own runs. The best results are marked in **bold**.

	ACE04	ACE05
KB Matching	0.711	0.708
RoBERTa-Span	0.770	0.768
Tea-Stu (span-based)	0.782	0.791
Ensemble (span-based)	0.819	0.819
CLIM (Ours)	**0.831**	**0.822**

Table 2: The main results in the nested DS-NER task, via span F1 scores. We run all baselines using the span-based schema. The value of KB Matching is the result of manually-labeled noisy data in the training set. The best results are marked in **bold**.

	LOC	ORG	PER	MISC	ALL
RoSTER	0.923	0.839	0.942	0.528(0.861/0.380)	0.862
SCDL	0.817	0.803	0.913	0.609(0.802/0.491)	0.817
Ours	0.877	0.885	0.920	0.673(0.744/0.615)	0.864

Table 3: Class-level performance comparison with strong baselines on CoNLL03 training set, via span F1 (Precision / Recall).

self-training (BOND, BA-CIR, RoSTER, SCDL, and Ours) show better performance than other denoising methods, reflecting the superiority of self-learning methods in DS-NER. (3) Compared with the strong baseline RoSTER, our model shows better robustness among various data settings. (4) In extremely noisy data, our model significantly outperforms other methods. In the Tweet datasets with low KB matching values, our model boosts span F1 scores by 2.2%, compared with the previous SOTA method SCDL.

Nested Distantly Labeled NER Task. For the nested ACE04 and ACE05, the span F1 values are listed in Table 2. Since the outstanding performance of the teacher-student framework (BOND) and the ensemble learning (RoSTER and SCDL) in the flat case, we implement two strong baselines for a fair comparison, which are Tea-Stu (span-based) and Ensemble (span-based), respectively. We conclude the nested results with two aspects. (1) Compared to KB matching, our model achieves higher F1 scores by significant margins, showing that our model is effective at handling noisy data in the nested NER task. (2) Consistent with the flat case, our model still achieves the best results among these self-training methods.

5.3 Denoising Performance Analysis

Based on the prediction and ground-truth label (not distant label) in the CoNLL03 training set, we discuss the denoising performance at the class level, compared to the strong baselines RoSTER and SCDL.

More Consistency with Flexible Threshold. In general (ALL in Table 3), the generated pseudo label in our model is more accurate, which is strongly related to the robustness under noisy data interference. Among different classes (LOC, ORG, PER, MISC in Table 3), our model shows more consistent performance, especially in the entity class MICS. The reason is that the class-wise flexible threshold considers the different learning abilities compared to the baselines, and pays more attention to other classes besides the high-performance class.

Better Exploration with Hybrid Pseudo Label. The low-performance class MISC (MISC in Table 3) shows a significantly higher recall in our model,

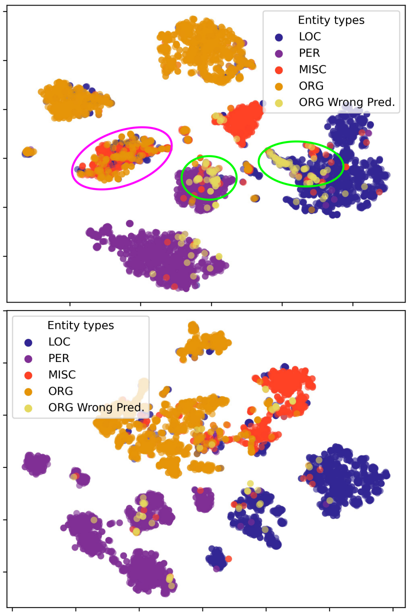
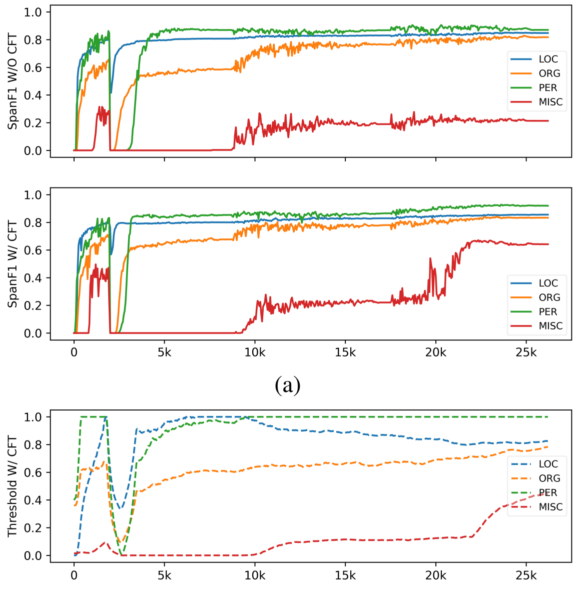


Figure 3: The representation visualization of entities on CoNLL03 testing set. The left subgraph represents strong baseline SCDL, and the right is our model. The markers with different colors represent different entity classes. And the yellow markers represent the wrong predictions of entity class ORG (ORG Wrong Pred.). The green and red circles are list in the left subgraph.



(a)

(b)

Figure 4: The class-wise analysis on CoNLL03 testing set. The horizontal axis represents the training steps. (4a) The vertical axis represents the span F1 scores. The upper subgraph denotes our model **w/o** CFT, and the lower subgraph denotes our model. (4b) The vertical axis represents the class-wise threshold based on our model.

implying that the special design of hybrid pseudo label improves the feature exploration of the low-performance class, by addressing the bias in label generation. In addition, our model further improves the performance of low-performance class MISC, proving that our model largely alleviates the performance degradation in the low-performance class.

5.4 Improvement from Hybrid Pseudo Label

We discuss the effects of the hybrid pseudo label with representation visualization, compared to the strong baseline SCDL. All entities are visualized in Figure 3, where different colors represent different entity classes. We take entity class ORG as an example to highlight its wrong predictions, where the wrong predictions have the ground true label ORG but are classified into other types.

Strong Classification Ability. Considering the highlight of green circles in Figure 3, the yellow markers (wrong predictions) in strong baseline SCDL are more widely distributed among different groups than in our model. Unlike SCDL, which only uses the prediction of the model itself, we additionally integrate the knowledge in the distantly-labeled entity into self-training. Since the distantly-labeled entities come from the entity-related knowledge bases, the distantly-labeled data contains abundant entity-related semantic information, which provides additional information for entity classification in self-training.

Clear Separation between Entity Classes. Considering the highlight of the red circle in Figure 3, markers of different entity classes (red, orange, and blue) are mixed, indicating that entity classes with similar semantics are wrongly clustered. This is presumably because the bias of the pseudo label further expands in self-training when the model is updated iteratively under the guide of this pseudo label. However, injecting the distant label, our model alleviates this bias with the semantic information of the distantly-labeled entities, and is better at identifying the difference between similar entity classes.

5.5 Boosting from Flexible Threshold

The effect of the Class-wise Flexible Threshold (CFT) is finely analyzed through the look into the training process. The F1 scores against the training iterations of each entity class are shown in Figure 4. And we mainly focus on the entity class MISC (represented by the red line), which contains com-

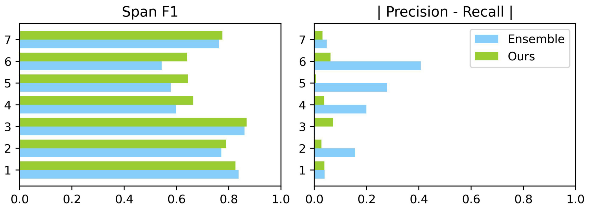


Figure 5: The class-level performance in the nested ACE05 testing set. The left subgraph is the span F1 score in all classes, and the right is the class-wise absolute value between precision and recall.

plex semantics (Tong et al., 2021) and shows low performance. The detailed characteristics of the training process are provided in Appendix A.

Effectiveness of Warm-up Process. Unlike the counterpart (**w/o** CFT), our model can quickly identify the low-performance class MISC in the early stage. We infer that the warm-up strategy in the flexible threshold design allows the candidate with low confidence to be selected in the early stage.

Attention for Complex Class. With the training progress, the line of the complex class MISC (Tong et al., 2021) in our model (the upper subgraph) is constantly rising until it reaches a steady state, but the model without CFT reaches a plateau prematurely. Therefore, our model effectively captures the complex feature of the class MISC. Besides, the increased capability for recognizing the class MISC happens at the late stage of model training. We conjecture this is due to the memorization mechanism of deep networks, that they would first memorize simple patterns than complex patterns (Arpit et al., 2017). And our model can fully adapt to the memorization mechanism, as the class-wise flexible threshold is dynamically adjusted according to the variant learning ability of the complex class during training.

5.6 Nested Case Study

Our work extends the DS-NER tasks to the nested case, and more detailed experimental results will be provided in the following part.

Class-Balancing Performance. We focus on the nested benchmark ACE05, and analyze the class-level performance with the strong baseline Ensemble. Totally, the class-level performance in the nested case agrees with that in the flat case. First, our model has improved significantly for the classes with low performance (Class 4, 5, and 6), as shown in the left subgraph of Figure 5, which exhibits more consistent performance among all classes. Second, our model tackles the large gap (between precision and recall) in the above classes compared to the Ensemble baseline, as observed in the right subgraph of Figure 5. And these two conclusions prove the validity of candidate selection and label generation in CLIM.

Statistics in training data	0.437	0.569	0.708
Predictions in testing data	0.774	0.808	0.822

Table 4: The model performance with different noise levels, via the span F1 score.

	CoNLL03	Wikigold	Tweet	ACE04
Our model	0.854	0.679	0.538	0.831 151†
Const.Thresh.(CT)	0.817	0.593	0.526	0.830 226†
LinearThresh. (LT)	0.826	0.565	0.537	0.829 191†
Const.Weight. (CW)	0.808	0.579	0.529	0.801
DataAug.(DA)	0.841	0.535	0.532	0.819

Table 5: The ablation study. The values marked with † denote the number of training epochs when the model reaches the optimal state, and other values denote the span F1 scores.

Robustness in Different Noise Levels. As mentioned in Appendix C, the distant label generation for the nested dataset is related to the statistics of CoNLL03. We then extend the distant label generation with the statistics of different flat benchmarks, including Wikigold and Twitter, and investigate the performance on different noise levels of the training set. As shown in Table 4, our model exhibits robustness towards varying degrees of noise.

5.7 Ablation Study

As shown in Table 5, we implement an exclusive ablation study to validate the effectiveness of each component, including the following aspects: (1) replacing the flexible threshold (Section 4.1) with the constant threshold (CT) and linearly-increased threshold (LT); (2) replacing the dynamic weighting of the pseudo label and distant label (Eq. 6) with the constant weighting (CW); (3) replacing the random masking in the attention matrix with the random masking in token input for data augmentation (DA).

Compared with different benchmarks, the nested case shows a more robust performance than the flat case. The flexible threshold strategy significantly accelerate the convergence speed in the nested case, as our model takes around 50 fewer training epochs to converge than its counterpart (CT and LT).

When each component is removed separately, the model shows different degrees of performance degradation, indicating the effectiveness of different components. We summarize the following aspects. (1) Compared to the constant threshold (CT), linearly-increased threshold (LT) shows higher performance, except for Wikigold. Although linearly-increased threshold can imitate the growth process of model learning ability, the class-level mismatch between the learning ability and threshold may deteriorate the performance. (2) The simple random masking, referred to the pre-training strategies in the pre-trained language model (Vaswani et al., 2017), shows the best performance. More advanced data augmentation strategies could be explored and applied in our framework, which is not in the scope of this paper. Further, we take a comprehensive parameter study in Appendix B.

6 Conclusion

This work advances the class-rebalancing self-training in the distantly-supervised named entity recognition. With the class-wise flexible threshold and the fine-grained hybrid pseudo label in self-training, our work tackles the dominance from the high-performance class and the degradation in the low-performance class. On this basis, the experiments show state-of-the-art results on seven benchmarks. And the comprehensive analysis further proves the more consistent performance in class-level learning and the stronger semantics classification ability. Our work, especially the advanced designs in self-training, positively impacts robust learning with noisy data. It provides a class-rebalancing method to explore the semantic information in distantly-labeled data.

Limitations

In the augmentation-driven self-training, we implement the data augmentation with random masking for simplicity, since augmentation is not the focus of this work. And Wang and Henao (2021) has explored more fine-grained data augmentation strategies, which may further improve performance.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and constructive suggestions. This research is supported by the National Key Research and Development Program of China (Grant No. 2020YFB1707803) and the Fundamental Research Funds for the Central Universities (Grant No. 226-2022-00051).

References

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 261–270, Hong Kong, China. Association for Computational Linguistics.

Yiming Chen, Yan Zhang, Chen Zhang, Grandee Lee, Ran Cheng, and Haizhou Li. 2021. Revisiting self-training for few-shot learning of language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9125–9135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ ACL WNUT NER shared task: Named entity recognition for Twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. 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 729–734, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, KDD '20, page 1054–1064, New York, NY, USA. Association for Computing Machinery.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han. 2021. Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10367–10378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419, Florence, Italy. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Rui Wang and Ricardo Henao. 2021. Unsupervised paraphrasing consistency training for low resource named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5308, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. 2021. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10857–10866.

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.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7164–7173. PMLR.

Bowen Zhang, Yidong Wang, Wenxin Hou, HAO WU, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021a. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 18408–18419. Curran Associates, Inc.

Wenkai Zhang, Hongyu Lin, Xianpei Han, and Le Sun. 2021b. De-biasing distantly supervised named entity recognition via causal intervention. 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 4803–4813, Online. Association for Computational Linguistics.

Xinghua Zhang, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Jiawei Sheng, Xue Mengge, and Hongbo Xu. 2021c. Improving distantly-supervised named entity recognition with self-collaborative denoising learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10746–10757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kang Zhou, Yuepei Li, and Qi Li. 2022. Distantly supervised named entity recognition via confidence-based multi-class positive and unlabeled learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7198–7211, Dublin, Ireland. Association for Computational Linguistics.

A Training Process Analysis

We investigate the characteristic of the whole training process, with four representative observational variables in Figure 6. As the training loss decreases, the span F1 scores have experienced a significant fluctuation, mainly due to the rapid change between the number of predicted spans for non-entity class ϵ and entity classes in $\mathcal{E}$. We infer that the enhanced ability to recognize non-span entities induces this change, including the representation reconstruction for each entity class. After the model performance of identifying non-entity spans reaches a steady state, the number of the predicted spans for entity classes in $\mathcal{E}$ steadily increase.

In addition, the extreme imbalance of the entity classes can be seen intuitively by comparing the number of predicted spans for non-entity class ϵ (around 1250) and entity classes in $\mathcal{E}$ (around 25). Thus the design of class-wise thresholds is vital to alleviate the class imbalance problem.

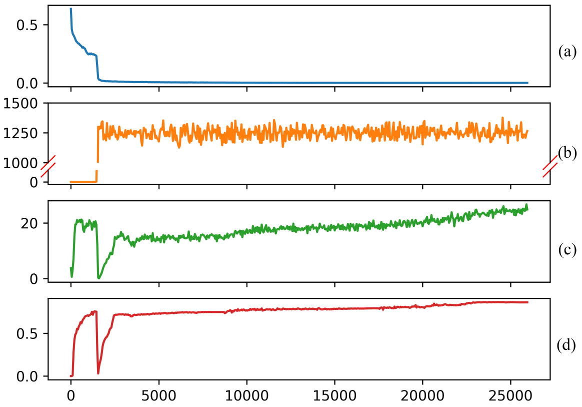


Figure 6: The training process analysis with four observational variables on CoNLL03. The horizontal axis represents the training steps, and the vertical axis represents different meanings with four subgraphs: (a) the total training loss $\mathcal{L}_t$; (b) the number of predicted spans for non-entity class ϵ; (c) the number of predicted spans for entity classes in $\mathcal{E}$; (d) the span F1 scores in the development set.

B Parameter Study

B.1 Upper Bond Threshold

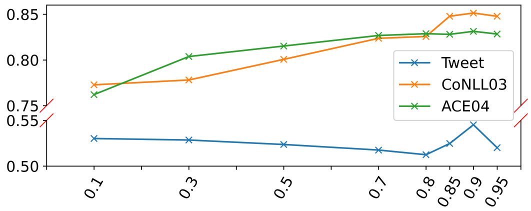


Figure 7: The parameter study of upper bond threshold on three benchmarks. The horizontal axis represents the values of the thresholds τ, and the vertical axis represents span F1 scores.

We investigate the effects of the threshold upper bond τ in Figure 7. In general, all three datasets achieve high-performance results around the threshold upper bond of 0.9. With higher values of the threshold, the amount of the predicted non-entity spans decreases, so the model training at the early stage concentrates more on entity classes in $\mathcal{E}$. In CoNLL03 and ACE04, the optimal results are achieved at high thresholds, which suggests that reducing the number of non-entity spans at the early stage helps the feature extraction of entity classes to some extent. However, the Tweet dataset obtains comparable performance with small thresholds. We assume this is because the Tweet dataset is inherently noisier. With a small threshold, the noise in pseudo labels is too heavy for the model to remember, making the model get a comparable performance by chance.

B.2 Masking Probability

Weak Strong	**0.05**	**0.10**	**0.20**
0.05	0.613	0.605	0.561
0.10	0.611	0.596	0.592
0.20	0.679	0.643	0.596

Table 6: The parameter study of masking probability on Wikigold, via the span F1 scores. The first row represents the maximum masking probability in the weak augmentation, and the first column represents the maximum masking probability in the strong augmentation.

We study the masking probability in Wikigold. Based on the experimental results in Table 6, we summarize that the combination of the weak augmentation with *low* masking probability and strong augmentation with *high* masking probability shows high performance. As in Table 6, the lower left cases (the values of 0.679, 0.643, 0.611) show high performance. And these results agree with the intuition. Since the weak augmentation with low masking probability explores more useful information about the input sentence, the pseudo label generated from the weakly-augmented data is more confident than the strong augmentation.

B.3 Dynamic Weighting

We explore three different designs of dynamic weighting, the results are shown in Figure 8a. The visualization of these mappings, from the training phase $\hat{t}$ to the distant label weights λ_y and the pseudo label weights λ_p, is provided in Figure 8b. And the definition of these mappings is shown as follows:

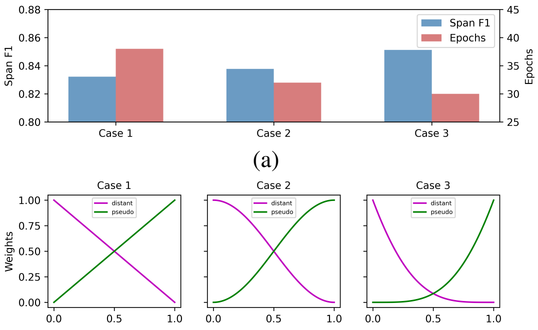
$$\text{Case 1} \quad \begin{cases} \lambda_y = \hat{t}, \\ \lambda_p = 1 - \hat{t} \end{cases}$$

$$\text{Case 2} \quad \begin{cases} \lambda_y = \left(\sin\left(0.5 \cdot \pi \left(\hat{t} - 1\right)\right)\right)^2 \\ \lambda_p = \left(\cos\left(0.5 \cdot \pi \left(\hat{t} + 1\right)\right)\right)^2 \end{cases}$$

$$\text{Case 3} \quad \begin{cases} \lambda_y = \left(\cos\left(0.5 \cdot \pi \left(\hat{t} + 1\right)\right) + 1\right)^2 \\ \lambda_p = \left(\sin\left(0.5 \cdot \pi \left(\hat{t} - 1\right)\right) + 1\right)^2 \end{cases}$$

where $\hat{t} = t/t_{total} \in [0, 1]$, t_{total} is the total training steps. And Case 3 is used in our work.

We design the above three mappings with the following consideration. (1) A general idea is to decrease the distant label weights and increase the pseudo label weights, with ongoing training. (2) Before the model obtains useful features for entity classes, the training mainly focuses on the distant label, thus slowing down the weight growth of the pseudo label. (3) And also, we accelerate the decline of distant label weights to avoid model overfitting.



(a)

(b)

Figure 8: (8a) The ablation study of the dynamic weighting on CoNLL03, via the span F1 score and training epoch; (8b) The visualization of the different mappings. The X-axis represents the training phase and the Y-axis represents the distant/pseudo label weights.

As seen from the results in Figure 8, there is a positive correlation between the more delicate design mapping and the higher model performance, including the span F1 scores and the convergence.

C Distant Label Generation in Nested Case

Though many works focus on distantly-supervised NER of the flat case, the study for the nested case is rare. Like the fully supervised NER task, recognizing the nested named entity is also essential for the downstream application. Hence, we extend the distantly-supervised NER with the nested case.

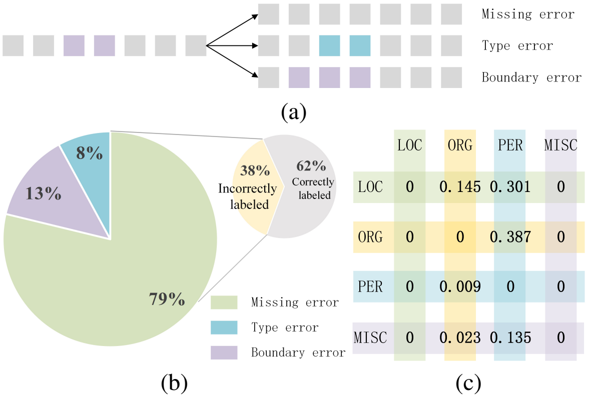
The span-based schema is to make a prediction on the entity level, and has shown high performance in the flat case. And we prove that our framework could further improve the ability to uncover the unlabeled entity and mislabeled entity in the nested case.

Distant Label generation with external knowledge bases is time-consuming, considering the collection of external dictionaries and the design of matching rules. In this work, we attempt to construct the noisy nested dataset by artificially adding

Dataset	CoNLL03	OntoNotes5.0	Tweet	Webpage	Wikigold	ACE2004	ACE2005
Learning Rate	3e-6	3e-6	3e-5	3e-5	3e-6	3e-6	3e-6
Max. Len. Span	9	9	9	9	9	12	12
Train Epoch	40	15	200	300	250	250	200

Table 7: Hyper-parameter settings in the DS-NER task. *Learning Rate* represents the initial learning rate with a cosine learning rate decay schedule; *Max. Len. Span* represents the maximum length of the candidate spans; *Train Epoch* represents the maximum epochs in the training process.

noise to ground-truth labels, which includes the following steps: (1) define the noisy type of named entity based on the ground-truth labels; (2) calculate the frequency of different noisy cases in a dataset; (3) generate the noisy labels according to the statistical results.



	LOC	ORG	PER	MISC
LOC	0	0.145	0.301	0
ORG	0	0	0.387	0
PER	0	0.009	0	0
MISC	0	0.023	0.135	0

Figure 9: (9a) Illustrate three types of the predefined noise; (9b) The statistic of the correctly/incorrectly labeled entity, and the more detailed statistic of the different noisy types in the incorrectly labeled case; (9c) The statistic of the type error, where the values represent the probability that one entity mislabels from true type (in raw) to wrong type (in column).

The incorrect annotations consist of missing, boundary, and type errors. Missing error means that an entity in the sentences (labeled in the ground-truth training set) is not identified during the rule-based matching process. Boundary error refers to the entities of the incorrectly labeled boundary and the correctly labeled type, and type error refers to the entities of the correctly labeled type and the incorrectly labeled boundary.

Taking CoNLL03 as an example, we statistic the incorrectly labeled entities in the ground-truth training set. Three predefined noisy types have already covered all incorrectly-labeled entities, as shown in Figure 9b. In addition, there are more incorrectly labeled cases of type error, when the semantic similarity between entity classes is relatively large, as shown in Table 9c. Then we generate the noisy label for the ACE04 and ACE05 datasets, with the statistics in Figure 9b and 9c.

D Dataset Statistics

Dataset	# types	# samples	# entities	# nested entities
CoNLL03	4	14041	17781	-
ON5.0	18	115812	125366	-
Tweet	10	2393	994	-
Webpage	4	385	393	-
Wikigold	4	1142	2282	-
ACE2004	7	6200	15745	3355
ACE2005	7	7292	17695	3438

Table 8: Statistics in the distantly-labeled training set. *# types*: the number of the pre-defined entity classes; *# samples*: the number of the training samples; *# entities*: the number of the distantly-labeled entities; *# nested entities*: the number of the distantly-labeled nested entities.

E Hyper-parameter and Baseline Setting

Detailed hyper-parameter settings for each dataset are shown in Table 7. Among then, we mainly fine-tune the parameters of the initial learning rate and training epoch, where the initial learning rate is chosen from {3e-5, 3e-6}, training epoch is chosen from {15, 30, 40, 50, 200, 250, 300}. The rest of the parameters are default in huggingface Transformers. We conduct the experiments on NVIDIA Tesla V100 GPU.

The baselines in the nested case are all implemented with the span-based schema. The average predictions of 2 ensemble models are used for the baseline Ensemble in Table 2.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In Limitations Section

☒ A2. Did you discuss any potential risks of your work?
No risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Left blank.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The data used in our work is open source.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The data used in our work is open source.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data used in our work is open source.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The data used in our work is open source.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In Experiment Section.

C ☑ Did you run computational experiments?

In Experiment Section.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Experiment Section.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Experiment Section.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
In Experiment Section.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
In Experiment Section.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.