Improving Distantly Supervised NER via Token-Level Curriculum-Based Positive-Unlabeled Learning

Anonymous ACL submission

Abstract

This paper studies the named entity recognition (NER) task under distant supervision. Distant supervision from existing resources can be used to annotate a training corpus instead of requiring a fully annotated corpus from domain experts, saving time and human effort. The drawback of distant supervision lies in the inferior label quality. Errors, including false positives, false negatives and positive type errors, are unavoidable. To address the different types of noises, we propose a token-level **Cu**rriculum-based **P**ositive-**U**nlabeled **L**earning (CuPUL) method. Using the proposed difficulty scoring function, the tokens are assigned to different curricula, with the easier tokens in the earlier curricula and the harder tokens in the latter curricula. Then CuPUL trains gradually with more curricula using the Conf-MPU loss function. Our experiments on seven datasets, including a newly collected dataset in animal science domain, show that the CuPUL can achieve superior performances, and extensive studies demonstrate the effectiveness of different components of the proposed CuPUL.

1 Introduction

Named Entity Recognition (NER) is an important task in natural language processing that aims to identify and classify named entities in text into predefined types, such as person, location, and organization. In recent years, supervised learning has been successful in NER tasks. However, it needs a large number of high-quality annotations to train a deep learning model, which can be costly and time-consuming to acquire. To address this issue, Distantly-Supervised Named Entity Recognition (DS-NER) has been proposed. This task uses existing knowledge bases (KB) or dictionaries to provide annotations, greatly reducing the need for manual annotations. However, the annotations from distant supervision suffer from annotation quality issues such as false positives, false negatives, and positive type errors.

To address the aforementioned issues in DS-NER, various methods are proposed. Some studies focus on false negative issues (Shang et al., 2018; Peng et al., 2019; Zhou et al., 2022). These methods adjust loss functions to reduce the impact of missing labels. These methods assume that KB or dictionaries are high quality, so false negative issues are the predominant issues. Recent studies relax the assumption and propose to tackle general noisy annotations through noise removal processes (Meng et al., 2021; Liang et al., 2020; Hedderich and Klakow, 2018; Zhang et al., 2021b; Liu et al., 2021). Some methods detect noisy annotations using model prediction confidence, where the assumption is if a moderately well-trained NER model strongly disagrees with a distant annotation, then this annotation is likely to be noisy. Some methods detect noisy annotations using the loss distribution, where the assumption is that the model converges slower on noisy annotations than on clean annotations.

The noise removal process faces several challenges. First, a moderately well-trained NER model is necessary to detect noise. However, it is hard to determine a proper threshold for when a NER model is moderately well-trained. Stopping the training too early, the model cannot produce an accurate enough model for noise detection. Stopping the training too late, the model will learn the noise and degrade the performance. Second, the moderately well-trained NER model is trained on noisy labels initially, so the noise detection methods may have unknown biases and cause irreparable damage.

In this paper, instead of removing noisy labels, we propose a token-level Curriculum-based Positive-Unlabeled Learning (CuPUL) method to tackle the challenge of noisy labels in DS-NER tasks. The motivation of curriculum learning is that

deep learning models are non-convex and trained using batches of samples, so the order of training data can significantly impact the model performance. Curriculum Learning rearranges the batches of training samples such that the model learns from easy to hard and learns from easy samples more times. With the new arrangement, the models tend to converge to a better local optimum. We follow the philosophy of curriculum learning and design a token-level curricula arrangement to address the token-level noise for DS-NER tasks, where we observe that "easy samples" are usually cleaner. Consequentially, learning from easy samples first can avoid label noise initially and make the model more robust. We further adopt Positive-Unlabeled (PU) learning paradigm to address the false negative issues.

Specifically, CuPUL first trains several voters to evaluate the difficulty level of each token for the NER task. Then, the tokens are assigned to different curricula based on their difficulty scores, with the easier tokens in the earlier curricula and the harder tokens in the latter curricula. CuPUL trains gradually with more curricula in each round using the Conf-MPU loss function (Zhou et al., 2022). We evaluate CuPUL on seven DS-NER datasets. Experimental results demonstrate that CuPUL consistently achieves better performance over existing state-of-the-art approaches. Ablation studies illustrate the effectiveness of curriculum learning procedures in DS-NER tasks.

In summary, our main contributions are:

- We propose CuPUL to tackle the challenge of noisy labels in DS-NER tasks following the curriculum learning philosophy. As far as we know, this is the first time that curriculum learning being applied to DS-NER tasks.
- We propose a token-level curriculum scheduler to tackle the positive type noises and adopt a PU loss function to tackle the false negative noises.
- We also provide an expert-labeled NER dataset in the animal science domain.
- We empirically demonstrate that CuPUL can significantly alleviate the impact of label noise during the model training and outperform the state-of-the-art DS-NER methods on benchmark datasets and the newly collected dataset.

2 Related Work

Fully supervised NER using deep neural networks always requires a large number of training data with human annotations, which is very costly. To alleviate the human efforts on annotating, DS-NER has been proposed and received increasing research interest recently, where annotations can be obtained from existing professional dictionaries or knowledge bases by some matching or query methods. However, because of the polysemy in language and the limited coverage of distant supervision resources, DS-NER often suffers from annotation errors like false positive, false negative, and positive type errors. Therefore, handling annotation errors in DS-NER has drawn special attention (Yang et al., 2018; Shang et al., 2018; Mayhew et al., 2019; Cao et al., 2019; Peng et al., 2019; Liang et al., 2020; Liu et al., 2021; Zhang et al., 2021a,c; Meng et al., 2021). Here we briefly discuss a few representative approaches.

One line of work assumes that distant supervision often has high-quality positive labels, therefore focusing on alleviating the impact of false negative errors. AutoNER (Shang et al., 2018) proposes a new tagging scheme to identify entity candidates and does not count the training loss on those candidates. Mayhew et al. (2019) introduce a constraint-driven iterative algorithm learning to detect false negative errors in the noisy data and down-weigh them, resulting in a weighted training set on which a weighted NER model is trained. More recently, positive and unlabeled learning has been adopted (Peng et al., 2019; Zhou et al., 2022) to tackle false negative errors from the loss function perspective without detection steps. Due to its superiority in tolerating false negative errors, we embed Conf-MPU (Zhou et al., 2022) into our proposed method. Top-Neg (Xu et al., 2023) selectively uses negative samples with high similarity to positives of the same entity type, improving performance by effectively distinguishing false negatives.

Another line of work simultaneously considers annotation errors of all types. Cao et al. (2019) design a data selection scheme to compute scores for annotation confidence and annotation coverage to distinguish high-quality sentences from noisy ones. BOND (Liang et al., 2020), leveraging the power of the pre-trained language model RoBERTa, first adopts early stopping to prevent overfitting to noisy labels. Liu et al. (2021) propose a calibrated confidence estimation approach for DS-NER and

integrate it in an LSTM-CRF model under a self-training framework to reduce the impact of noise. Zhang et al. (2021a) study the noise in DS-NER from the perspective of dictionary bias. SCDL (Zhang et al., 2021c) takes two teacher-student networks and a co-training paradigm to cope with noise and take full advantage of mislabeled samples. ATSEN (Qu et al., 2023) further develops the teacher-student networks and achieves better performance. RoSTER (Meng et al., 2021) proposes a noise-robust learning scheme consisting of a new loss function and a noisy label removal step to better model training with noisy data. SANTA (Si et al., 2023) deals with explicit and implicit errors separately. CLIM (Li et al., 2023) addresses the imbalance problem in different classes with high-quality candidate selection and label generation.

3 Preliminary

In this section, we briefly introduce the DS-NER task and curriculum learning.

3.1 NER Classifier and DS-NER Formulation

NER is the process of locating and classifying named entities in a corpus into predefined categories. We denote an input sentence with M tokens as $\boldsymbol{x} = [x_1, x_2, \cdots, x_M]$ and denote corresponding annotations as $\boldsymbol{y} = [y_1, y_2, \cdots, y_M]$, $y_i \in \{0, 1, \cdots, k\}$, where 0 denotes non-entity and $1, \cdots, k$ denote k entity types. In this paper, we consider token-level NER formulation, where an NER classifier predicts token labels. Formally, the contextual token representations of an input sentence $\boldsymbol{x}$ are represented as

$$[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_M] = Linear(Encoder(\boldsymbol{x})), \quad (1)$$

where the encoder can be a pre-trained language model (e.g., BERT). The final prediction is

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = Softmax([\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_M]), \quad (2)$$

$$\hat{\boldsymbol{y}} = Argmax(f(\boldsymbol{x}, \boldsymbol{\theta})), \quad (3)$$

where $\boldsymbol{\theta}$ denotes the parameters of the encoder and the linear layers, and $\hat{\boldsymbol{y}}$ is the prediction.

To construct distantly annotated training data, the corpus can be annotated with dictionaries by string matching (Ren et al., 2015; Giannakopoulos et al., 2017; Peng et al., 2019), or with knowledge bases by their provided APIs. However, the annotation process will introduce three types of noises, namely, false positives, false negatives, and positive type errors, where false positives refer to the noise where non-entity tokens are erroneously labeled as entities of a certain type, false negatives refer to the noises where entity tokens are mistakenly labeled as non-entity, and positive type errors refer to misclassifications of entity tokens (for instance, when a token of type PER is erroneously marked as type ORG).

3.2 Curriculum Learning

Curriculum learning was first proposed by Bengio et al. (2009) under the assumption that learning with reordering from "easy" samples to "hard" samples would boost performance. It has been applied in various applications, including neural machine translation (Zhou et al., 2020; Platanios et al., 2019; Zhou et al., 2020; Wang et al., 2018), relation extraction (Huang and Du, 2019), reading comprehension (Tay et al., 2019), natural language understanding (Xu et al., 2020) and named entity recognition (Jafarpour et al., 2021; Lobov et al., 2022; Wenjing et al., 2021).

Curriculum learning has two main steps: difficulty estimation and curriculum scheduler (Kocmi and Bojar, 2017). For a dataset $\mathbf{Z} = \{z_i\}_{i=1}^{T}$, the goal of difficulty estimation is to design a difficulty scoring function and compute a score for each sample z_i. Mathematically, the difficulty score of each sample is

$$H_i = D(z_i), 1 \leq i \leq T, \quad (4)$$

where $D(\cdot)$ is the difficulty scoring function. A higher H_i indicates that the sample z_i is more difficult to learn.

Curriculum scheduler includes creating curricula $C_1, C_2, \cdots, C_\eta$ based on difficulty scores and scheduling learning stages $S_1, S_2, \cdots, S_\sigma$. Each stage consists of some curricula.

Several studies aim to adopt curriculum learning philosophy for textual data and propose various difficulty-scoring functions and curriculum schedulers. Some methods measure sample difficulty with features derived from lexical statistics, e.g., sentence length and word rarity (Platanios et al., 2019; Jafarpour et al., 2021), where longer sentences and rarer words are considered "hard". Others use features from pre-trained language models (Zhou et al., 2020; Wang et al., 2018; Liu et al., 2020). Most schedulers select samples with difficulty scores lower than a threshold (Platanios et al., 2019). While Zhou et al. (2020) design a

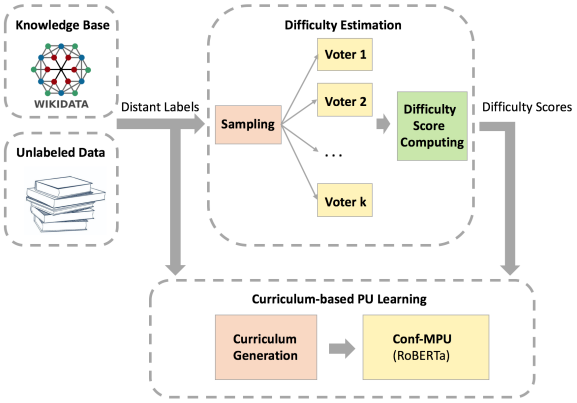

Figure 1: The CuPUL Framework

sample selecting function based on model uncertainty. Our approach, unique in applying token-level curriculum learning to DS-NER tasks, diverges from common sentence-level methods by utilizing Transformer-based models like BERT for context-aware token-specific predictions and gradient learning.

4 Methodology

This section introduces the proposed framework named CuPUL (Figure 1). The process starts by distantly labeling the corpus using knowledge bases, and then several *voters* are trained (Section 4.1) using this data to calculate token difficulty scores (Section 4.2). Finally, CuPUL trains a NER classifier following the curriculum scheduler using confidence-based positive-unlabeled learning (Section 4.3).

4.1 Difficulty Estimation

Motivated by the token-level noises in DS-NER tasks, we design the difficulty estimator and the curriculum scheduler at the token level as well. It allows the model to learn from one sentence by ignoring the noisy tokens. For example, in the sentence "Peter(PER) lives(O) in(O) America(ORG)", "Peter", "lives", and "in" are clean samples, and "America" is a noise sample. The model can learn from "Peter lives in X" by ignoring the noise in the sentence. The token's difficulty score reflects its inherent learnability. These scores are estimated using the disagreements between basic NER models or voters.

4.1.1 Voters

The design of the voters demands simplicity and variability. To balance efficiency and diversity, several voters are trained to increase prediction results variability. Label imbalance in NER tasks is mitigated by sampling from negative samples, introducing randomness to voters. Randomness is further enhanced with different random processes, such as shuffling and initialization.

For training the voters, a neural network for NER classification is used, as defined in Section 3.1. The voters are trained using a regular multi-class classification risk function. The risk of a classifier f is given by:

$$\mathrm{R}(f) = \sum_{i=1}^{k} \pi_i \mathrm{R}^{+}_{\mathrm{P}_i}(f) + (1 - \sum_{i=1}^{k} \pi_i) \mathrm{R}^{-}_{\mathrm{N}}(f), \quad (5)$$

where $\mathrm{R}^{+}_{\mathrm{P}_i}(f) = \mathbb{E}_{x \sim p(x|y=i)} \left[\ell(f(x, \boldsymbol{\theta}), i)\right]$ and $\pi_i = p(y = i)$ are the classification risk and the prior of the i-th positive class, respectively, and $\mathrm{R}^{-}_{\mathrm{N}}(f) = \mathbb{E}_{x \sim p(x|y=0)} \left[\ell(f(x, \boldsymbol{\theta}), 0)\right]$ is the classification risk of the negative class.

4.1.2 Difficulty Scores

After training V voters, each token x receives V predicted class probabilities $f(x, \boldsymbol{\theta}_1), ..., f(x, \boldsymbol{\theta}_V)$, where $\boldsymbol{\theta}_1 ... \boldsymbol{\theta}_V$ are the voters' parameters. The prediction $f(x, \boldsymbol{\theta}_i)$ is a vector that represents the class distribution of each token x denoted as $\mathbf{Pr}_i(x)$. The difficulty of the token is assessed based on the disagreement among these distributions. Specifically, we use Kullback-Leibler (KL) divergence, a measurement for dissimilarities of two distributions $\mathbf{Pr}_i(x)$ and $\mathbf{Pr}_j(x)$, to calculate the disagreement level of two voters. Mathematically, it is:

$$H_{ij} = \frac{1}{2}\{D_{KL}(\mathbf{Pr}_i(x)||\mathbf{Pr}_j(x))) + D_{KL}(\mathbf{Pr}_j(x))||\mathbf{Pr}_i(x))\}, \quad (6)$$

where $D_{KL}(\cdot)$ denotes the KL divergence. KL divergence is asymmetric. By taking the average of H_{ij} and H_{ji}, we derive a symmetric difficulty score $H_{\{ij\}}$.

Given that there are V voters, the final difficulty score for each token x is defined as the average of the non-identical pairs among all voters as:

$$H = \frac{\sum_{i=1}^{V} \sum_{j=i+1}^{V} H_{\{ij\}}}{V \cdot (V-1)/2}. \quad (7)$$

Eq.(7) defines the token difficulty scores as an arithmetic mean of disagreements between pair-wise voters. Consequently, a token's difficulty score is low when all voters agree, and it increases with greater disagreement.

4.2 Curriculum Design

Since the prevalence of false negatives in distant labels can cause a Positive-Negative learning model to overfit, we set the model learning process as Positive-Unlabeled (PU) learning, where data labeled with 0 is considered unlabeled rather than non-entity. PU learning assumes that 1) labeled positive samples have the same distribution as the positive samples in the data, and 2) the unlabeled data follows the distribution of the entire dataset (Zhou et al., 2022). To fulfill the second assumption, we directly incorporate all unlabeled data into the curriculum to prevent too less unlabeled tokens in curricula. Different curriculum partitions are executed solely on the labeled positive data.

The curriculum design is based on token difficulty scores H. Empirical studies have shown that these scores follow a long-tail distribution (Figure 3), indicating that most tokens are relatively "easy". Previous work (Platanios et al., 2019; Gnana Sheela and Deepa, 2013) suggests that a curriculum with uniformly ranged difficulty scores might lead to most tokens belonging to the first curriculum, making curriculum learning ineffective. Hence, we propose using a power-law selector to construct a more effective curriculum scheduler.

For a corpus of T_p labeled **positive** tokens and T_u unlabeled tokens, we initially index the labeled positive tokens from the easiest to the hardest and put unlabeled tokens in front of them. The first curriculum is then populated with the first τT_p labeled positive tokens and all unlabeled tokens, where τ $(0 < \tau < 1)$ is a selective factor that indicates the proportion of tokens selected from the corpus. We then select the first $\tau^2 T$ tokens from the remaining $(1-\tau)T_p$ tokens as the second curriculum. This selection process continues until the penultimate curriculum. The remaining tokens are placed in the final curriculum. The token index of final curricula is denoted as $C_1, C_2, ..., C_\eta$.

$$
\begin{aligned}
&C_1 : 1 \sim T_u + \tau T_p \\
&C_2 : T_u + \tau T_p + 1 \sim T_u + (\tau + \tau^2) T_p \\
&\quad \cdots \\
&C_\eta : T_u + (\tau^{\eta-1} + \cdots + \tau) T_p + 1 \sim T_u + T_p.
\end{aligned}
$$

Note that the first curriculum C_1 starts at index 1, and the ending position of the last curriculum C_η is defined to be at index $T_u + T_p$.

For example, suppose $Tp = 20$, $T_u = 80$ and $\tau = 0.5$. A three-split containing positive curriculum would be (1 to 90), (91 to 95), and (96 to 100). Thus, the curricula cover the entire corpus.

4.3 Curriculum-based PU Learning

To train the NER classifier with η curricula, we employ the common discrete training scheduler, "Baby Step" (Spitkovsky et al., 2010; Cirik et al., 2017). Training begins with the first curriculum C_1, and the next curriculum is added after a set number of epochs. This process continues until all curricula are included and trained, terminating the training process. The training stages $(\{S_i, 1 < i \leq \eta\})$ correspond to the number of curricula, with the model trained over multiple epochs in each stage. Consequently, the tokens in easier curricula are learned more times. Each training stage S_i can be regarded as a standalone training process with the training subset $C_1, \cdots, C_i$. Therefore all unlabeled data and the majority of labeled positive data are used in each each training stage. Under these conditions, the two PU assumptions are maintained, allowing PU learning to be applied directly. Thus, CuPUL provides a more robust and effective curriculum learning framework.

Specifically, we adopt the Conf-MPU loss function, proposed by Zhou et al. (2022), as the backbone PU loss function in the curriculum-based training. Conf-MPU loss function has been shown to be more robust to PU assumption violation in practice. The risk function is

$$
\mathrm{R}(f) = \sum_{i=1}^{k} \pi_i \big(\mathrm{R}^+_{\mathrm{P}_i}(f) + \mathrm{R}^-_{\tilde{\mathrm{P}}_i}(f) - \mathrm{R}^-_{\mathrm{P}_i}(f) \big) + \mathrm{R}^-_{\tilde{\mathrm{U}}}(f), \tag{8}
$$

where $\mathrm{R}^-_{\tilde{\mathrm{P}}_i}(f) = \mathbb{E}_{x \sim p(x|y=i, \lambda(x)>\epsilon)} \left[\ell(f(x), 0) \frac{1}{\lambda(x)} \right]$, $\mathrm{R}^-_{\mathrm{P}_i}(f) = \mathbb{E}_{x \sim p(x|y=i)} \left[\ell(f(x), 0) \right]$, and $\mathrm{R}^-_{\tilde{\mathrm{U}}}(f) = \mathbb{E}_{x \sim p(x|\lambda(x) \leq \epsilon)} \left[\ell(f(x), 0) \right]$. $\lambda(x) = p(y > 0 \mid x)$ defines the confidence score of a token being an entity token. ϵ is a confidence score threshold in the range of $(0, 1]$. Due to the introduction of $\lambda(x)$ and ϵ, $\mathrm{R}(f)$ can be estimated by labeled positive data and unlabeled data in each stage with less bias compared with the traditional PU learning.

For stage S^*, the number of token selected for class i is $T_i^{S^*}$. For simplification, we denote it as T_i^*. The empirical estimator of Eq.(8) is

$$
\begin{aligned}
\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) = \sum_{i=1}^{k} \frac{\pi_i}{T_i^*} \sum_{j=1}^{T_i^*} \max \Big\{ & 0, \ell(f(x_j^{T_i^*}, \boldsymbol{\theta}), i) \\
& + \mathbb{1}_{\hat{\lambda}(x_j^{T_i^*}) > \epsilon} \ell(f(x_j^{T_i^*}, \boldsymbol{\theta}), 0) \frac{1}{\hat{\lambda}(x_j^{T_i^*})} - \ell(f(x_j^{T_i^*}, \boldsymbol{\theta}), 0) \Big\} \\
& + \frac{1}{T_0^*} \sum_{j=1}^{T_0^*} \left[\mathbb{1}_{\hat{\lambda}(x_j^{T_0^*}) \leq \epsilon} \ell(f(x_j^{T_0^*}, \boldsymbol{\theta}), 0) \right],
\end{aligned} \tag{9}
$$

with a non-negative constraint inspired by Kiryo et al. (2017) ensuring the risk on the negative class. We follow Zhou et al. (2022) and set ϵ to 0.5 by default. But different from having $\lambda(x)$ estimated by another binary PU model, we reuse the voters trained in Section 4.1 to ensemble the confidence score for each token x. We use the soft-label ensemble as

$$\mathbf{Pr}(x) = \frac{\sum_{j=1}^{V} f(x, \boldsymbol{\theta}_j)}{V}, \tag{10}$$

where $\mathbf{Pr}(x)$ is the ensemble probability distribution over all classes.

The confidence score of a token x being an entity token is then calculated as

$$\lambda(x) = \sum_{j=1}^{k} \mathbf{Pr}_j(x). \tag{11}$$

For the neural network of the NER classifier, we choose the same structure with voters, which is defined in Section 3.1.

4.4 Loss Function

Two loss functions are popularly used for the DS-NER tasks. The first loss function is cross entropy (CE) loss:

$$\ell_{CE} = log\ f_{i,y_i}(x; \boldsymbol{\theta}), \tag{12}$$

where $f_{i,y_i}(x; \boldsymbol{\theta})$ is the prediction of token x_i on class j.

Another commonly used loss function is mean absolute error (MAE):

$$\ell_{MAE} = |\boldsymbol{y}_i - f_{i,y_i}(x; \boldsymbol{\theta})|, \tag{13}$$

where $|\cdot|$ is L-1 norm of the vector and $\boldsymbol{y_i}$ denotes the one hot vector of y_i. We leave the discussion of these two loss functions in the Appendix A.

4.5 Self-Training

Several studies (Liang et al., 2020; Peng et al., 2019; Meng et al., 2021) have shown that self-training can effectively upgrade the performance of a trained DS-NER model. We apply the self-training method in Meng et al. (2021), which uses soft labels to conduct self-training and uses a masked language model to conduct contextual data augmentation simultaneously. Self-training is used directly after CuPUL, and we call the final classifier after self-training CuPUL+ST.

5 Experimental Study

5.1 Baseline Methods

Two groups of baseline methods are shown below.

Fully supervised methods. We include a fully supervised NER method based on the RoBERTa-base model (Liu et al., 2019) as an upper-bound performance reference.

Distantly-supervised methods. First, we report distant supervision results as KB-Matching. We classify DS-NER methods into three groups. 1) *DS-NER without Self-training* consists of **AutoNER**(Shang et al., 2018), **Conf-MPU** (Zhou et al., 2022), and **RoBERTa-ES** (Liang et al., 2020). CuPUL is directly comparable with these methods. 2) *DS-NER with Self-training* includes **BOND** (Liang et al., 2020), **RoSTER** (Meng et al., 2021), **SCDL** (Zhang et al., 2021c) and **ATSEN** (Qu et al., 2023). These methods apply teach-student or training augmentation steps to further boost the DS-NER performance. The methods in these two groups are sequence-based models. 3) *Span-based DS-NER models*, including **CLIM** (Li et al., 2023) , **SANTA** (Si et al., 2023) , and **Top-Neg** (Xu et al., 2023) . Previous work (Li et al., 2023) shows that span-based NER models often outperform sequence-based NER methods in terms of effectiveness, albeit at the cost of increased algorithmic complexity.

We also include an ablation version of CuPUL (labeled as CuPUL-curr), which removes Curriculum Learning, as a baseline. More details of baselines can be found in Appendix C.

5.2 Datasets and Metrics

Datasets: We conduct experiments on seven DS-NER datasets. Six of them are benchmark datasets including CoNLL03 (Liang et al., 2020), Twitter (Liang et al., 2020), OntoNotes5.0 (Liang et al., 2020), Wikigold (Liang et al., 2020), Webpage (Liang et al., 2020), and BC5CDR (Shang et al., 2018). The first five datasets are open-domain datasets, and BC5CDR is in bio-medical domain.

We also collected a new dataset from the animal science domain named "QTL". The NER goal is to detect Trait Entities in animal science publications, an important task in building a comprehensive database for livestock trait research and animal breeding practice. For the corpus, domain experts gathered 1,716 PubMed abstracts from quantitative trait locus (QTL) studies for 6 species. For the distant annotation, we collected a dictionary with 3,884 curated trait names from four domain on-

Dataset		Train	Valid	Test	Types
CoNLL03	Sentence	14041	20	3453	4
	Token	203621	475	46435	
Twitter	Sentence	2393	50	3844	10
	Token	44076	719	58064	
OntoNotes5.0	Sentence	115812	50	12217	18
	Token	2200865	1090	230118	
Wikigold	Sentence	1142	20	274	4
	Token	25819	579	6538	
Webpage	Sentence	385	20	135	4
	Token	5293	120	1131	
BC5CDR	Sentence	4560	20	4797	2
	Token	118170	533	124750	
QTL	Sentence	18706	21	1044	1
	Token	514176	952	32251	

Table 1: The statistics of involved DS-NER datasets, the valid set comprises a small subset from the original dataset, whereas trainset and testset utilize the entire original dataset.

tologies[1]. For validation and testing, we randomly selected 107 abstracts and acquired ground truth annotations from a domain expert curator. We split the annotated sentences to form validation and testing sets with 21 and 1,044 sentences, respectively. The statistics of seven datasets are summarized in Table 1. More details can be found in Appendix B.

Metrics: We use span-level Precision (P), Recall (R), and F1 score as the evaluation metrics. These metrics require exact matches between predicted and actual entities. A continuous span with the same label is considered a single entity during inference.

In the QTL application, according to the curator's practical needs, identifying potential entities is more important than identifying precise boundaries. Therefore, we also introduce relaxed Precision (P), Recall (R), and F1 score to evaluate the performance of DS-NER methods for practical usage. For relaxed metrics, it deems a predict span correct if there is at least one overlapping word with the ground truth annotation.

5.3 Experiment Settings

We use the pre-trained RoBERTa as the backbone model for both the Voter and NER classifier[2]. For open-domain datasets, we use *roberta-base*[3]. For bio-domain datasets, we use *biomed-reberta-base*[4]. We employ PyTorch[5] and conduct all experiments on a server with a Tesla A100 GPU (32G).

For the benchmark dataset, we use the small subset of validation to adjust the learning rate. Other hyper-parameters are set according to data statistics. For QTL dataset, baselines are reproduced using their published codes. Baselines CLIM, SANTA, and Top-Neg are excluded due to reproduction obstacles. Hyperparameters of baseline methods and CuPUL are tuned on the validation set. Details can be found in Appendix D.

5.4 Main Results

Table 2 presents the overall span-level precision, recall, and F1 scores for all methods on benchmark datasets. Note that RoSTER was tested on a different version of the OntoNotes5.0 dataset (Meng et al., 2021). Therefore we exclude its reported results in Table 2 for a fair comparison. We have the following observations.

The KB-Matching results show that distant labels are often of low recall, and on four of the benchmark datasets, of low precision as well. The noise-aware DS-NER models significantly outperform the KB-Matching. Span-based DS-NER models tend to perform better than sequence-based models, which aligns with previous findings (Li et al., 2023). However, span-based NER models require innumerating all spans in the sentences, having higher complexity and longer training and inference time than sequence-based models.

The proposed methods CuPUL and CuPUL+ST achieve the best F1 scores on five out of six datasets compared with all DS-NER models and comparable results on OntoNotes5.0 dataset. For OntoNotes5.0 dataset, almost all noise-aware DS-NER models have similar performance, implying that the distant annotations may contain certain biases that is hard for the model to address. Except for OntoNotes5.0 dataset, compared with DS-NER Baselines without Self-training, CuPUL shows significant improvement on all metrics. Even compared with DS-NER Baselines with Self-training, CuPUL outperforms on four datasets. With the self-training step, CuPUL+ST in general can further improve the performance. Surprisingly, CuPUL and CuPUL+ST achieve higher F1 scores than the supervised baseline (RoBERTa) on Twitter and Webpage. We suspect that the distributions of the training and test data are inconsistent for these datasets,

[1]Vertebrate Trait (VT) Ontology, Livestock Product Trait (LPT) Ontology, Livestock Breed Ontology (LBO), Clinical Measurement Ontology (CMO).

[2]We will release code upon paper acceptance.

[3]https://huggingface.co/roberta-base

[4]https://huggingface.co/allenai/biomed_roberta_base

[5]https://pytorch.org/

Method	CoNLL03	Twitter	OntoNotes5.0	Wikigold	Webpage	BC5CDR
Fully Supervised						
RoBERTa$^{\#}$	90.11 (89.14/91.10)	52.19 (51.76/52.63)	86.20 (84.59/87.88)	86.43 (85.33/87.66)	72.39 (66.29/79.73)	90.99 (-/-)†
DS-NER Baselines without Self-training						
KB-Matching$^{\#}$	71.40 (81.13/63.75)	35.83 (40.34/32.22)	59.51 (63.86/55.71)	47.76 (47.90/47.63)	52.45 (62.59/45.14)	64.32 (86.39/51.24)†
AutoNER$^{\#}$	67.00 (75.21/60.40)	26.10 (43.26/18.69)	67.18 (64.63/69.95)	47.54 (43.54/52.35)	51.39 (48.82/54.23)	79.99 (**82.63**/77.52)†
RoBERTa-ES$^{\#}$	75.61 (83.76/68.90)	46.61 (53.11/41.52)	68.11 (66.71/69.56)	51.55 (49.17/54.50)	59.11 (60.14/58.11)	73.66 (80.43/67.94)†
Conf-MPU†	79.16 (78.58/79.75)	-	-	-	-	77.22 (69.79/86.42)†
Span-based DS-NER models						
CLIM$^{\diamond}$	85.4 (-/-)	53.8 (-/-)	69.6 (-/-)	70 (-/-)	67.9 (-/-)	-
SANTA$^{\diamond}$	86.59 (86.25/86.95)	-	**69.72** (**69.24**/70.21)	-	71.79 (78.40/66.72)	79.23 (81.74/76.88)
Top-Neg$^{\diamond}$	80.55 (81.07/80.23)	52.86 (52.30/53.55)	-	-	-	80.39 (82.09/78.90)
DS-NER Baselines with Self-training						
BOND$^{\#}$	81.15 (82.00/80.92)	48.01 (53.16/43.76)	68.35 (67.14/69.61)	60.07 (53.44/68.58)	65.74 (67.37/64.19)	-
RoSTER¶	85.40 (85.90/84.90)	-	-	67.80 (64.90/71.00)	-	-
SCDL‡	83.69 (**87.96**/79.82)	51.10 (59.87/44.57)	68.61 (67.49/69.77)	64.13 (62.25/66.12)	68.47 (68.71/68.24)	-
ATSEN‡	85.59 (86.14/85.05)	52.46 (62.32/45.30)	68.95 (66.97/**71.05**)	-	70.55 (71.08/70.55)	-
Proposed Methods						
CuPUL-curr	83.18 (83.69/82.68)	50.12 (47.48/53.07)	67.76 (65.66/70.00)	66.43 (58.89/76.18)	65.15 (62.89/67.57)	79.91 (75.07/85.43)
CuPUL	85.09 (84.64/85.53)	54.34 (54.47/**54.20**)	68.06 (66.31/69.91)	**70.53** (**67.06**/**74.39**)	73.10 (74.65/71.62)	**81.57** (77.02/86.70)
CuPUL+ST	**86.64** (86.02/**87.27**)	**54.78** (**57.32**/52.46)	68.20 (66.57/69.11)	70.19 (66.96/73.74)	**74.48** (**76.06**/**72.97**)	80.92 (75.45/**87.26**)

Table 2: Performance on benchmark datasets: F1 Score (Precision/Recall) (in %). # marks the row of results reported by Liang et al. (2020). ¶ marks the row of results reported by Meng et al. (2021), where results for Twitter, OntoNote5.0 and Webpage are not reported in Meng et al. (2021). ‡ marks the row of results reported by Zhang et al. (2021c). ◇ marks the row of results from the method proposed paper respectively. † marks the results from Zhou et al. (2022). Best results are in **bold**.

Method	QTL-strict	QTL-relax
DS-NER Baselines without Self-training		
KB-Matching	37.15 (82.95/23.93)	41.86 (93.46/26.97)
AutoNER	41.67 (69.07/29.83)	55.49 (83.17/41.64)
RoBERTa-ES	38.07 (76.30/25.37)	46.58 (91.15/31.28)
DS-NER Baselines with Self-training		
BOND	53.08 (60.89/47.04)	65.57 (77.97/56.57)
RoSTER	47.80 (73.12/35.51)	55.43 (**91.35**/39.79)
SCDL	43.62 (**79.57**/30.05)	50.18 (89.85/34.81)
ATSEN	46.23 (66.98/35.30)	51.64 (86.21/36.86)
Proposed Methods		
CuPUL-curr	54.75 (75.40/42.99)	62.94 (86.76/49.38)
CuPUL	58.02 (65.73/51.93)	70.24 (79.75/62.76)
CuPUL+ST	**61.83** (58.65/**65.38**)	**75.82** (72.96/**78.91**)

Table 3: Performance on QTL dataset: F1 Score (Precision/Recall) (in %). The best results are in **bold**.

and the supervised learning may have overfitted the training set due to the limited size.

Compared to CuPUL-curr, the ablation version of CuPUL by removing curriculum learning, CuPUL outperforms it across datasets, showing the efficacy of curriculum learning in boosting performance. Delving into precision and recall, we observe that CuPUL consistently achieves better precision than CuPUL-curr, indicating that curriculum learning does improve the model's robustness to false positives in training data.

Table 3 presents strict and relaxed precision, recall, and F1 scores for all methods on the QTL dataset. KB matching reveals that QTL annotations suffer from low recall but have relatively high precision, creating a notable imbalance between these metrics. We observe that DS-NER baselines without self-training have limited recall improvement, resulting in weak performance. DS-NER baselines with self-training improve recall, but CuPUL and CuPUL+ST can further boost the recall, significantly outperforming all baseline methods. Specifically, strict F1 and relaxed F1 of CuPUL+ST outperform the runner-up by 8.75% and 10.25%, respectively.

More experimental studies including Difficulty Score Estimation Efficiency Analysis, and Parameter Studies can be found in Appendix E H G.

6 Conclusion and Future Work

In this paper, we propose a token-level curriculum-based PU learning (CuPUL) method to improve distantly supervised named entity recognition tasks. We propose a difficulty scoring function that estimates the token difficulty based on disagreements between pair-wised voters. The tokens are then arranged into different curricula according to their difficulty scores. Finally, we propose a novel curriculum-based PU learning procedure and train CuPUL from easy to hard curricula. Experiments demonstrate the effectiveness of CuPUL on six benchmark datasets and the newly collected QTL dataset, and CuPUL outperforms state-of-the-art DS-NER models. Further studies illustrate the efficacy of each component in CuPUL.

Limitations

The "Baby Step" strategy in curriculum learning involves multiple repetitions of the first curriculum. Coupled with our power-law selector and curriculum scheduler, which tends to choose a larger initial curriculum, this may negatively impact efficiency if many curricula are established since the larger curriculum is repeatedly trained.

Ethics Statement

We comply with the ACL Code of Ethics.

References

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.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

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 EMNLP*, pages 261–270.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2017. Visualizing and understanding curriculum learning for long short-term memory networks.

Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised aspect term extraction with b-lstm & crf using automatically labelled datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 180–188.

K. Gnana Sheela and S.N. Deepa. 2013. Neural network based hybrid computing model for wind speed prediction. *Neurocomputing*, 122:425–429.

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.

Michael A Hedderich and Dietrich Klakow. 2018. Training a neural network in a low-resource setting on automatically annotated noisy data. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 12–18.

Yuyun Huang and Jinhua Du. 2019. Self-attention enhanced CNNs and collaborative curriculum learning for distantly supervised relation extraction. 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 389–398, Hong Kong, China. Association for Computational Linguistics.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active curriculum learning. In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 40–45, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in Neural Information Processing Systems*, 30.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386.

Qi Li, Tingyu Xie, Peng Peng, Hongwei Wang, and Gaoang Wang. 2023. A class-rebalancing self-training framework for distantly-supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11054–11068, Toronto, Canada. Association for Computational Linguistics.

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 and Data Mining*, pages 1054–1064.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021. Noisy-labeled ner with confidence estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3437–3445.

Xuebo Liu, Houtim Lai, Derek F Wong, and Lidia S Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436.

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. *arXiv preprint arXiv:1907.11692*.

Valeriy Lobov, Alexandra Ivoylova, and Serge Sharoff. 2022. Applying natural annotation and curriculum learning to named entity recognition for under-resourced languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4468–4480, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. Named entity recognition with partially annotated training data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, pages 645–655.

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 EMNLP*, pages 10367–10378.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuan-Jing 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.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*.

Xiaoye Qu, Jun Zeng, Daizong Liu, Zhefeng Wang, Baoxing Huai, and Pan Zhou. 2023. Distantly-supervised named entity recognition with adaptive teacher learning and fine-grained student ensemble. AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, pages 147–155.

Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, and Jiawei Han. 2015. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 995–1004. ACM.

Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on EMNLP*.

Shuzheng Si, Zefan Cai, Shuang Zeng, Guoqiang Feng, Jiaxing Lin, and Baobao Chang. 2023. SANTA: Separate strategies for inaccurate and incomplete annotation noise in distantly-supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3883–3896, Toronto, Canada. Association for Computational Linguistics.

Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

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-Volume 4*, pages 142–147.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the 3rd Conference on Machine Translation: Research Papers*, pages 133–143.

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.

Zhu Wenjing, Liu Jian, Xu Jinan, Chen Yufeng, and Zhang Yujie. 2021. Improving low-resource named entity recognition via label-aware data augmentation and curriculum denoising. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1131–1142, Huhhot, China. Chinese Information Processing Society of China.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Lu Xu, Lidong Bing, and Wei Lu. 2023. Sampling better negatives for distantly supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4874–4882, Toronto, Canada. Association for Computational Linguistics.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised ner with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169.

Wenkai Zhang, Hongyu Lin, Xianpei Han, and Le Sun. 2021a. 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.

Wenkai Zhang, Hongyu Lin, Xianpei Han, Le Sun, Huidan Liu, Zhicheng Wei, and Nicholas Yuan. 2021b. Denoising distantly supervised named entity recognition via a hypergeometric probabilistic model.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14481–14488.

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 EMNLP*, pages 10746–10757.

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.

Yikai Zhou, Baosong Yang, Derek F Wong, Yu Wan, and Lidia S Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944.

Appendix

A Discussion of Loss Function

Comparing the two loss functions, ℓ_{CE} is unbounded, and it grants better model convergence when trained with clean data (*i.e.,* y are ground truth labels) because more emphasis is put on difficult tokens. However, when the labels are noisy, training with the cross-entropy loss can cause overfitting to the wrongly labeled tokens. ℓ_{MAE} is more noise-robust than ℓ_{CE}. It is bounded and treats every token more equally for gradient update, allowing the learning process to be dominated by the correct majority in distant labels. However, using ℓ_{MAE} for training deep neural models generally worsens the convergence efficiency and effectiveness due to the inability to adjust for challenging training samples.

Considering the different characteristics of these two loss functions, in practice, we suggest using ℓ_{CE} loss for tasks with more entity types and using ℓ_{MAE} loss for tasks with fewer number of entity types.

B Datasets

Here, we give a short description of the six benchmark datasets as follows:

- CoNLL03 (Tjong Kim Sang and De Meulder, 2003) is built from 1393 English news articles and consists of four entity types: person, location, organization, and miscellaneous.
- Twitter (Godin et al., 2015) is from the WNUT 2016 NER shared task and consists of 10 entity types.
- OntoNotes5.0 (Weischedel et al., 2013) is built from documents of multiple domains like broadcast conversations, web data, etc. It consists of 18 entity types.
- Wikigold (Balasuriya et al., 2009) is built from a set of Wikipedia articles (40k tokens). They are randomly selected from a 2008 English dump and manually annotated with four entity types same as CoNLL03.
- Webpage (Ratinov and Roth, 2009) comprises personal, academic, and computer science conference web pages. It consists of 20 web pages that cover 783 entities with four entity types same as CoNLL03 too.
- BC5CDR comes from the biomedical domain. It consists of 1,500 articles, containing 15,935 Chemical and 12,852 Disease mentions.

C Baselines

Here, we give a short description of all the baseline methods: **KB-Matching** distantly labels the test sets using distant supervision, serving as a reference to illustrate the performance improvements given by other advanced DS-NER methods.

AutoNER (Shang et al., 2018) trains the neural model with a "Tie or Break" tagging scheme for entity boundary detection and then predicts entity type for each candidate.

Conf-MPU (Zhou et al., 2022) treats the NER task as a Positive-Unlabeled learning problem and utilizes the pre-learned confidence scores to enhance the model's performance.

CLIM (Li et al., 2023) addresses the imbalance problem in the high-performance and low-performance classes by improving the candidate selection and label generation.

SANTA (Si et al., 2023) dealing with inaccurate and incomplete annotation noise in DS-NER by utilizing separate strategies.

Top-Neg (Xu et al., 2023) selectively uses negative samples with high similarity to positives of the same entity type, improving performance by effectively distinguishing false negatives.

RoBERTa-ES (Liang et al., 2020) trains a NER model using a RoBERTa-base model and adopts early stopping to prevent the model from overfitting to noisy distant labels.

BOND (Liang et al., 2020) trains a RoBERTa model on distantly labeled data with early stopping and then uses a teacher-student framework to iteratively self-train the model.

RoSTER (Meng et al., 2021) employs a noise-robust loss function and a self-training process with contextual augmentation to train a NER model.

SCDL (Zhang et al., 2021c) conducts self-collaborative denoising with teacher-student framework. It trains two teacher-student networks, and the final reports come from the best model (teacher or student).

ATSEN (Qu et al., 2023) develops a teacher-student framework with adaptive teacher learning and fine-grained student ensembling.

D Experiment Settings

For datasets CoNLL03, OntoNotes5.0, Webpage, Twitter, Wikigold, QTL and BC5CDR, the maximum sequence length is set as 150, 230, 120, 160, 120, 180, 280 respectively, to ensure the algorithm works correctly. Other parameters shows in Table

4. OntoNote5.0 and Twitter have more entity types, so we choose relatively small γ values. We apply cross-entropy loss to OntoNotes5.0 and Twitter since they have more entity types and apply MAE loss to CoNLL03, Webpage, and Wikigold. For all the datasets, we train them with a batch size of 32 sentences and apply Adam optimizer (Kingma and Ba, 2014). For all the datasets, the number of voters K and the number of curricula C are set as 5 and 5, respectively. The curriculum selective factor τ is set to 0.5 for all the datasets. We use the same random seeds for all datasets.

E Difficulty Score Estimation

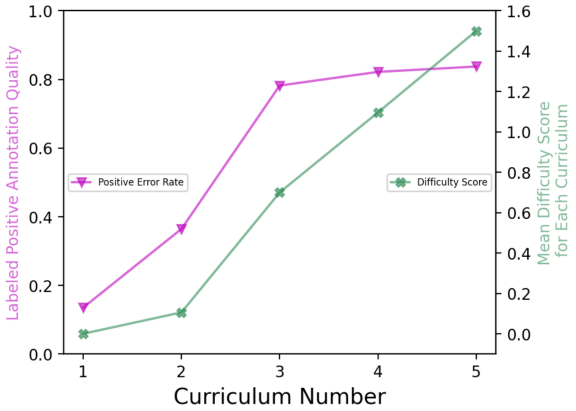

Figure 2: Distant Label Quality Token Level Positive Error Rate and Mean Difficulty Scores for Each Curriculum on Wikigold Dataset.

For CuPUL, one assumption adopted is that difficulty scores can reflect the quality of distant supervision, where "easier" tokens have "cleaner" labels. To validate this assumption and evaluate the quality of the difficulty score estimation, we examine the correlation between the difficulty scores and the quality of distant labels. We use Wikigold as the testbed, and the results are illustrated in Figure 2.

For each training curriculum, we compute the token-level positive error rate (positive errors includes false positives and positive type errors), and plot the rate for each curriculum use the left y-axis in Figure 2. We also compute the average difficulty scores for tokens in each curriculum shown with the right y-axis in Figure 2.

We can see that as the number of curricula increases, the average token difficulty scores and positive error rate have a clear increase. This illustrates a strong negative correlation between the difficulty scores and the quality of distant labels. Specifically, as the difficulty score increases, the quality

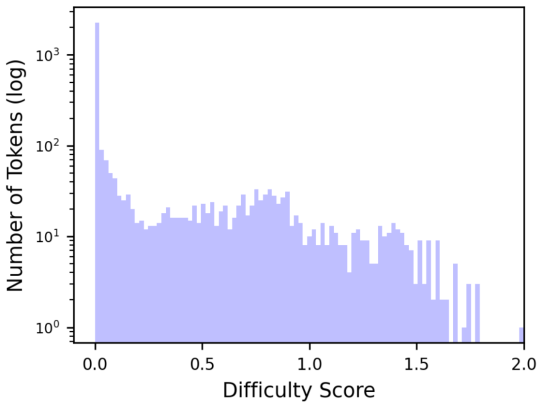

Figure 3: Distribution of the Difficulty Scores for Labeled Positives on Wikigold Dataset

of the distant labels significantly decreases. This result validates our assumption that "easy" data have cleaner labels and "hard" data have noisier labels. The clean data can initialize the model from a better starting point and improve the model's robustness to noise in the latter curricula.

Another important assumption we adopt for the design of curricula is that the difficulty scores follow a long-tail distribution. We illustrate the distribution of difficulty scores estimated on the Wikigold dataset in Figure 3. It clearly demonstrates the long-tail phenomenon, with most tokens having low difficulty scores. This phenomenon can be observed in other datasets. Due to the space limit, we omit the plots for other datasets.

F Ablation Study

To further evaluate the effectiveness of CuPUL, we conduct ablation studies based on Wikigold and Twitter datasets. The results are shown in Table 5.

To evaluate the effectiveness of the curriculum learning in CuPUL, we compare it with two variations of CuPUL. First, we use the five voters trained using positive and sampled negative examples and take the average of their soft label predictions as the result. The results are shown as voter ensemble in Table 5. Second, we include the result of CuPUL-curr from Table 2 since it is another variation. To evaluate the effectiveness of the Conf-MPU loss estimation for curriculum learning in CuPUL, we use the regular loss estimation, which considers unlabeled tokens as non-entity tokens, denoted as w/o Conf-MPU in Table 5.

Our analysis reveals the critical role of each component, as removing any of them results in a significant drop in the F1 score. Compared CuPUL-curr

hyper-parameter	CoNLL03	Twitter	OntoNotes5.0	Wikigold	Webpage	BC5CDR	QTL
trainset sentence #	14041	2393	115812	1142	385	4560	18706
voter drop negative	0.3	0.1	0.3	0.1	0.1	0.3	0.3
voter learning rate	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5
voter learning epochs	1	5	1	10	15	5	1
Conf-MPU γ	20	10	20	10	10	20	20
curriculum learning stage epochs	1	2	1	2	2	1	1
curriculum learning learning rate	1e-5	7e-5	3e-5	1e-5	5e-5	1e-5	5e-5

Table 4: The hyper-parameters used in CuPUL

Method	Wikigold			Twitter		
	Precision	Recall	F1	Precision	Recall	F1
CuPUL	67.06	74.39	70.53	54.47	54.20	54.34
w/o Curriculum Learning						
voter ensemble	56.88	74.88	64.65	35.52	49.52	41.37
CuPUL-curr	58.89	76.18	66.43	47.48	53.07	50.12
w/o Conf-MPU	59.31	75.86	66.57	58.91	47.04	52.53

Table 5: Ablation study on Wikigold and Twitter datasets. CuPUL is compared with variations without Curriculum Learning (voter ensemble only and Conf-MPU only) and without Conf-MPU loss in Curriculum Learning.

with w/o Conf-MPU, we find that CuPUL-curr consistently achieves higher recall. This is attributed to Conf-MPU primarily addressing false positives and partial false positives (Zhou et al., 2022), leading to more tokens being predicted as entities, thereby enhancing recall. Conversely, w/o Conf-MPU exhibits higher precision since it tackles both false positives and positive type errors. Addressing positive type errors benefits both precision and recall, but the increase in precision is more pronounced compared to CuPUL-curr.

In previous methods, a moderately well-trained model is often used to detect label noise, and the confidently predicted soft labels from the moderately well-trained model are often used to replace the noisy distant labels. Based on our previous experiments, the ensembled voters can be viewed as a moderately well-trained model, and the earlier curricula are formed with data that the moderately well-trained model can confidently predict. Thus, following the previous methods, we study which labels should be used for curriculum learning in CuPUL, the voters' ensembled soft labels or the noisy distant labels. Note that the ensembled labels used here are the soft labels of the voters' ensemble. We use KL-divergence as the loss function in curriculum learning to learn from soft labels.

Figure 4 plots the results regarding F1 scores on test data with respect to incremental curriculum stages. We can see that CuPUL learns in almost all stages of the curricula, and the F1 value is steadily improving until the second last curriculum. However, using ensembled soft labels, the model has a good start but reaches the upper bound quickly. We have the following insights from this experiment. 1) A model that only learns from the confidently predicted labels and ignores the potential noisy data may converge faster but can be impacted by the performance bottleneck of the initial model. 2) the last curricula may contain high label noise (See Appendix E for more details), so training on the last curricula may degrade the performance slightly. However, thanks to the curriculum learning schedule, the model is overall robust to noise in the last curricula.

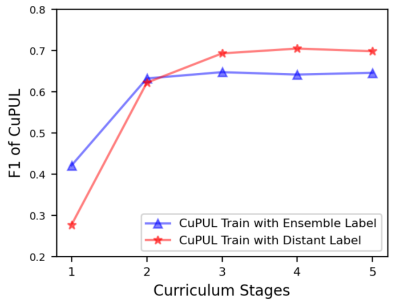

Figure 4: F1 scores of CuPUL on test data of Wikigold trained with Distant Labels (red) and Ensembled Labels from voters (blue) after each curriculum training stage.

G Parameter Study

Here, we perform parameter studies. Due to the simplicity of CuPUL, we mainly study two parameters: number of voters V and number of curricula η. To ensure comparability of experimental results, we keep all other parameters fixed and only change the corresponding parameter (V or η) to demonstrate their impact. The experiments are carried out on Wikigold.

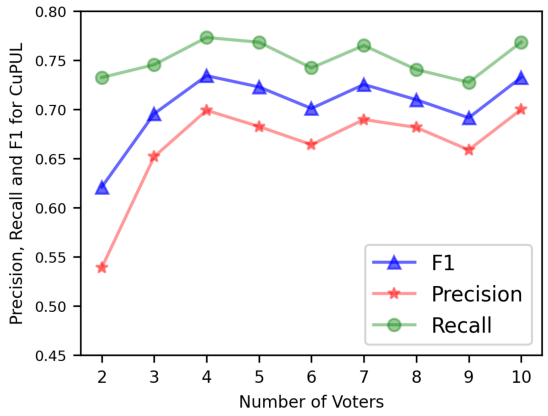

Figure 5: Span Level Precision, Recall, and F1 scores of CuPUL with respect to Number of Voters V.

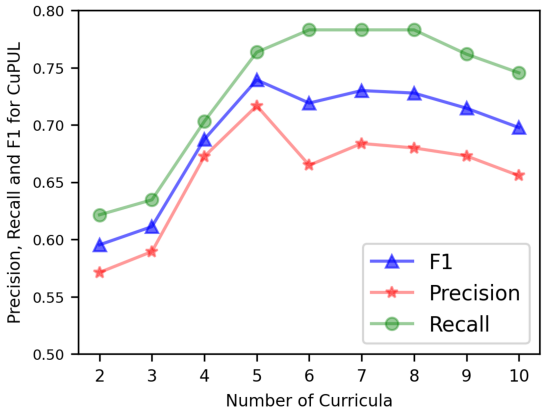

Figure 6: Span Level Precision, Recall, and F1 scores of CuPUL with respect to Number of Curricula η.

G.1 Number of Voters V

Figure 5 shows the effect of the number of voters V to CuPUL performance. From the figure, we can see that when there are only two voters, the performance of CuPUL is poor. This is understandable because, with too few voters, the difficulty scores estimated are unreliable, which leads to a low-quality curriculum scheduler. As the number of voters increases, the performance of CuPUL also rapidly improves. When the number of voters is 4, it reaches a local maximum. Then, as the number of voters increases, the new voters can no longer provide new information for difficulty estimation, and the results of CuPUL are stabilized around 0.7. Therefore, with the consideration of computation efficiency, a moderate number greater than or equal to 4 can be chosen for the number of voters.

G.2 Number of Curricula η

Figure 6 shows the effect of the number of curricula to CuPUL performance. Like the number of voters, when the number of curricula is small, the performance of CuPUL is poor. Too few curricula can reduce the ability to distinguish between easy and difficult tokens, leading to ineffective curriculum learning. With the increase of η, the performance of CuPUL also improves and reaches the best performance at $\eta = 5$. After that, as the number of curricula increases, the performance of CuPUL is relatively stable. The performance of CuPUL begins to decline after $\eta > 8$. The decline may be caused by the data having been trained too many rounds and the model starts to overfit to noisy labels.

	BOND	RoSTER	SCDL	Conf-MPU	CuPUL	CuPUL-ST
Run Time	978s	2397s	4319s	732s	819s	1733s
	16m18s	39m57s	71m59s	12m12s	13m39s	28m53s

Table 6: Efficiency analysis on CoNLL03, m means minute, s means second

H Efficiency Analysis

In order to evaluate the efficiency of CuPUL, we undertook performance timing of the principal methods on CoNLL03, with the results displayed in Table 6. All tests were performed on an identical computing infrastructure. The training epochs for BOND and SCDL were preset to 5, while the parameter configurations for RoSTER adhered strictly to those detailed in their respective paper. The data in the table reveals that Conf-MPU had the least time requirement. Our approach, CuPUL, demonstrated competitive performance in this regard. Even when the self-training procedure was incorporated into CuPUL-ST, it maintained a substantial efficiency advantage relative to both RoSTER and SCDL.