Re-Examine Distantly Supervised NER: A New Benchmark and a Simple Approach

Anonymous ACL submission

Abstract

Distantly-Supervised Named Entity Recognition (DS-NER) uses knowledge bases or dictionaries for annotations, reducing manual efforts but facing challenges like false positives and negatives in training data. In this paper, we re-examined existing DS-NER methods in real-world scenarios and found that many of them rely on large validation sets and some used test set for tuning inappropriately. We introduced a new dataset named QTL, where the training data is annotated using domain dictionaries and the test data is annotated by domain experts. This dataset has a small validation set, reflecting real-life scenarios. We also propose a new approach, token-level Curriculum-based Positive-Unlabeled Learning (CuPUL), which uses curriculum learning to order training samples from easy to hard. This method stabilizes training, making it robust and effective on small validation sets. CuPUL also addresses false negative issues using the Positive-Unlabeled learning paradigm, demonstrating improved performance in real-life applications.

1 Introduction

Distantly-Supervised Named Entity Recognition (DS-NER) is a task to leverage existing knowledge bases (KBs) or dictionaries to provide annotations for named entity recognition tasks. This approach significantly reduces the need for labor-intensive manual annotations, but it faces challenges due to issues in automated annotations, such as false positives and false negatives. To address the annotation errors, various methods are proposed. Some studies focus on false negative issues (Shang et al., 2018; Peng et al., 2019; Zhou et al., 2022). Others 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., 2021a; Liu et al., 2021).

Existing DS-NER approaches have been successfully applied to NER benchmark datasets, such as CoNLL2003, achieving competitive performances with fully supervised methods. However, the DS-NER scenarios mimicked by distantly labeling benchmark datasets often deviate from real-world scenarios. In the current DS-NER works, the training set annotations are often crudely replaced with distantly labeled annotations, thus converting a fully supervised (FS) dataset into a distantly supervised dataset, but the validation set remains the same. This approach overlooks the significant manual labor required to obtain a validation set for parameter tuning in real-life scenarios. Ignoring this issue leads to a decline in the performance of existing methods when applied to real-world problems, thereby undermining the reliability of existing approaches.

To assess the effect of the validation set, we re-examined existing DS-NER approaches and found several issues. Some approaches do not follow the DS-NER setting and directly use the test set for hyperparameter tuning, resulting in unreliable performance. Some approaches rely on large validation sets to achieve good performance. Some approaches train models using fixed hyperparameters, yet their models fail to perform well across all datasets. These findings demonstrate that current DS-NER approaches fall short in addressing real-life DS-NER problems effectively.

To further evaluate the effect, we introduce a real-life DS-NER dataset, QTL, which is annotated for trait entities in the animal science domain. Unlike previous datasets, QTL has a very small validation set, consisting of only 21 sentences, avoiding the significant manual effort required to obtain large validation sets in real-life scenarios. Additionally, unlike previous benchmark datasets, where entity mentions contain high portion of proper nouns, trait entities in QTL dataset are descriptive, such as "tail size" and "hoof color".

We further present a simple yet effective approach inspired by Curriculum learning and

Positive-Unlabeled (PU) learning , named CuPUL. The motivation behind 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 model performance. Curriculum learning rearranges the batches of training samples such that the model learns from easy to hard samples and revisits easier samples more frequently. With this new arrangement, models tend to converge to a better local optimum. Furthermore, we design a token-level curriculum arrangement to address token-level noise in DS-NER tasks. We observe that "easy samples" are usually cleaner, and learning from these first can initially avoid label noise, making the model more robust. To tackle false negative issues, we adopt the Positive-Unlabeled learning paradigm.

In summary, our main contributions are:

- We present a real-life DS-NER dataset, QTL, and test the performance of the existing state-of-the-art methods. We observe that many methods do not follow the practical DS-NER setting and have unsatisfactory performance.
- We propose a simple method CuPUL to address the noise issue in DS-NER. We empirically demonstrate that CuPUL can significantly outperform the state-of-the-art DS-NER method on the QTL dataset and different benchmark datasets.

2 Preliminary

2.1 DS-NER methods

We collected DS-NER methods published in major conferences in 2023 and their compared baselines. We categorize the existing DS-NER methods in three groups. 1)**DS-NER with Self-training**. To improve model performance, many DS-NER methods often incorporate a self-training step, utilizing mechanisms such as soft-label retraining and multi-model teacher-student frameworks. This group includes BOND (Liang et al., 2020), RoSTER (Meng et al., 2021), SCDL (Zhang et al., 2021b), ATSEN (Qu et al., 2023) and DesERT (Wang et al., 2023). 2)**DS-NER without Self-training**. This group of methods focuses on addressing the model's effectiveness in handling noise or false positives in DS-NER. While these methods can incorporate self-training mechanisms, it is not the primary focus of these methods.This group include AutoNER (Shang et al., 2018), Conf-MPU (Zhou et al., 2022), MProto (Wu et al., 2023) . 3) **Span-based DS-NER**. The final group of methods differs from the previous two, as it is based on span-based prediction rather than sequence labeling. These methods treat each span within a sentence as the prediction target. Previous work (Li et al., 2023) has shown that span-based NER models often outperform sequence-based NER methods in terms of effectiveness, albeit at the cost of increased algorithmic complexity. This group includes Top-Neg (Xu et al., 2023), CLIM (Li et al., 2023) and SANTA (Si et al., 2023). More details can be found in Appendix A.

2.2 Method Analysis

We first analyze the feasibility and usability of existing DS-NER methods in real-life applications. For a method to be considered feasible, it must provide runnable code and instructions for hyperparameter tuning if necessary. Table 1 presents our feasibility analysis results base on the manuscripts and code repositories (accessed in April 2024). We find that 1) MProto and SANTA do not provide hyperparameter tuning instructions; 2) CLIM and Top-Neg do not provide runnable code; and 3) BOND, SCDL, and ATSEN selected their inference model based on performance on the test set according to their released repositories. Thus in our empirical studies, for a fair comparison, we only re-examine feasible methods and update some methods to select the inference model based on performance on the validation set only.

The motivation of DS-NER methods is that the manual annotations are too costly to obtain. Therefore, to reduce the amount of manual annotation, the annotations in the training set come from knowledge bases or dictionaries, and the validation set should not be large either. Existing methods focus on the first setting while neglecting the importance of the second setting. We analyzed the feasible methods in Table 1 based on these DS-NER settings and have the following observations. First, AutoNER and RoSTER use fixed hyperparameters. These approaches do not require hyperparameter tuning, thereby avoiding the need for a validation set. Second, Conf-MPU provides a strategy for preselecting hyperparameters, so it does not require a validation set either. However, the remaining methods (BOND, SCDL, ATSEN, and DesSERT) need a validation set for hyperparameter tuning. The size of the validation set may affect their performance. We present a detailed analysis of this impact in

Method	Code Provided	Code Runable	Hyperparameter	Tuning required	Tuning Instruction	Inference model	Feasible
DS-NER without Self-training							
AutoNER	✓	✓	Fixed	✗	-	Model at Final Epoch	✓
Conf-MPU	✓	✓	Not Fixed	✗	-	Model at Final Epoch	✓
MProto	✓	✓	Not Fixed	✓	✗	Model at Final Epoch	✗
DS-NER with Self-training							
BOND	✓	✓	Not Fixed	✓	✓	Best Model on Test	✓
RoSTER	✓	✓	Fixed	✗	-	Model at Final Epoch	✓
SCDL	✓	✓	Not Fixed	✓	✓	Best Model on Test	✓
ATSEN	✓	✓	Not Fixed	✓	✓	Best Model on Test	✓
DesERT	✓	✓	Not Fixed	✓	✓	First Student Model	✓
Span-based DS-NER models							
SANTA	✓	✓	Not Fixed	✓	✗	Model at Final Epoch	✗
Top-Neg	✓	✗	-	-	-	-	✗
CLIM	✗	-	-	-	-	-	✗

Table 1: Feasibility Analysis of Exist Methods for DS-NER tasks.

Section 5.

3 QTL Benchmark

We first present QTL, a real-life DS-NER application in the animal science domain. The entity type to recognize is "trait", an important task in the construction of genotype-phenotype databases for advancing livestock genomics research and breeding methodologies. Different from previous DS-NER benchmark datasets, where entities consist of many proper nouns, trait entities consist of descriptive expressions.

To establish the QTL dataset, we collected a corpus with 1,717 abstracts, which were meticulously selected from PubMed[1] by domain experts for quantitative trait locus (QTL) studies related to six species: cattle, pig, goat, sheep, chicken, and rainbow trout. For the distant annotation process, the domain experts gathered a specialized dictionary of 3,884 trait names from four established domain ontologies[2]. Among these abstracts, 1,609 were used in training data, which consisted of 18,706 sentences with 514,176 tokens.

A well-trained domain curator provided annotations for 108 randomly selected abstracts, which covered all six species of interest. A second domain curator randomly checked 10 abstracts and had a total agreement with the first curator. Therefore, we used the annotations as ground truth. More annotation details can be found in Appendix B. Among all the human-annotated sentences, we randomly selected 21 sentences (with 952 tokens) to form the validation set and the rest sentences formed the test set, which contains 1,044 sentences with 32,251 tokens and 1,219 entities.

[1] https://pubmed.ncbi.nlm.nih.gov/

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

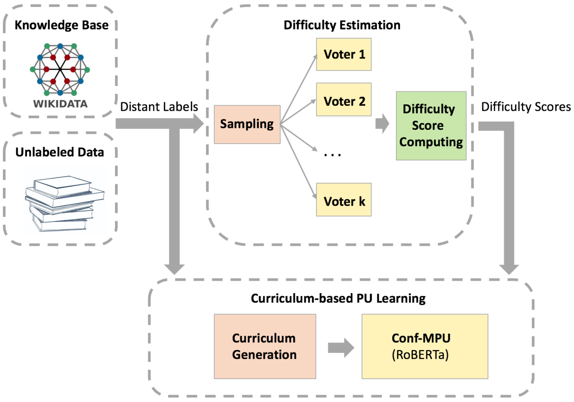

Figure 1: Overview of CuPUL

Notably, the validation set is quite small in the QTL dataset. This practice followed the motivation of DS-NER tasks, where the human effort should be minimized. This limited size of the validation set may impact the tuning of hyperparameters during the model training process, potentially affecting the model's performance. This issue reflects a realistic challenge encountered in DS-NER applications, which requires the model to be robust and not sensitive to hyperparameters.

Annotation Limitations: Due to the cost of hiring domain curators, the majority of the annotations are provided by a single curator. Another observation from the curators is that there is a considerable amount of discontinued trait entities. For example, in "milk protein, lactose, and fat percentage", there are three entities: milk protein percentage, milk lactose percentage, and milk fat percentage. Due to the annotation software limitation, this example was annotated as "milk protein", "lactose", and "fat percentage".

4 Methodology

In this section, we introduce a simple DS-NER method that combines the advantages of curriculum learning and PU learning. Figure 1 shows the overview of the proposed method CuPUL. The method starts by training several *voters* using the distantly annotated data to calculate token difficulty scores. Then CuPUL trains a NER classifier following the curriculum scheduler using confidence-based positive-unlabeled learning risk estimation.

Problem Formulation: 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 the unlabeled type and $1, \cdots, k$ denote k entity types. For the models, a pre-trained language model such as RoBERTa is used to encode token representations and followed by a softmax function to forward the prediction of entity labels for each token in the sentence.

4.1 Difficulty Estimation

Curriculum learning has two main steps: difficulty estimation and curriculum scheduler (Kocmi and Bojar, 2017). More details and related work of curriculum learning are discussed in Appendix C.

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 noisy sample. The model can learn from "Peter lives in X" by ignoring the noise in the sentence. The token's difficulty score should reflect its inherent learnability. These scores are estimated using the disagreements between basic NER models or voters.

4.1.1 Voters

For training the voters, a neural network for NER classification is used. The design of the voters demands simplicity and variability. Thus, the voters are trained using a regular multi-class classification risk function. The training process follows the Positive-Negative setting, where 0 represents non-entity type. Label imbalance in NER tasks is mitigated by sampling negative samples. Note that the performance of the voter itself does not affect the final outcomes of CuPUL, which we will introduce in the section 4.2.

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:

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

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:

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

Eq.(2) 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

To avoid overfitting negative samples, we adopt Positive-Unlabeled (PU) learning based risk estimation, treating data labeled with 0 as unlabeled rather than non-entity. PU learning assumes the unlabeled data represents the entire dataset's distribution (Zhou et al., 2022). To meet this assumption, we include all unlabeled data in the first curriculum, scheduling only the labeled positive data.

Our curriculum is based on token difficulty scores H, which follow a long-tail distribution, making most tokens "easy" (Figure 3). Previous research (Platanios et al., 2019; Gnana Sheela and Deepa, 2013) indicates that a uniform difficulty range may render curriculum learning ineffective. Therefore, we propose a power-law selector for a more effective curriculum scheduler.

To build the curricula, we first arrange all T_u unlabeled tokens followed by T_p positive-labeled tokens sorted by their difficulty scores in ascending order. The first curriculum consists of all unlabeled tokens and the first τT_p labeled positive tokens, where τ $(0 < \tau < 1)$ is a selective factor. The second curriculum consists of the first $\tau^2 T$ tokens from the remaining $(1 - \tau)T_p$ tokens. This selection process continues until the penultimate curriculum. The remaining tokens are placed in the final curriculum. These curricula are denoted as $C_1, C_2, ..., C_\eta$. For example, suppose $T_p = 20$, $T_u = 80$, $\tau = 0.5$, and $\eta = 3$. Then, C_1 consists of tokens indexed from 1 to 90 (80 unlabeled tokens and the 10 easiest positive tokens), C_2 consists of tokens indexed from 91 to 95, and C_3 consists of tokens indexed from 96 to 100.

4.3 Curriculum-based PU Learning

We train the NER classifier across η curricula using the "Baby Step" training schedule(Spitkovsky et al., 2010; Cirik et al., 2017). Starting with C_1, we add each subsequent curriculum after a fixed number of epochs, training through all curricula until completion. 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. Each stage is treated as an independent training segment, with earlier curricula being reviewed more frequently, enhancing learning under PU assumptions and resulting in a robust 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. Details of Conf-MPU can be found in Appendix D. Instead of having entity confidence score $\lambda(x)$ estimated by another binary PU model, the only difference we make is to reuse the voters trained in Section 4.1 to ensemble the confidence score for each token x. We use the soft-label ensemble as

$$\mathbf{Pr}(x) = \frac{\sum_{j=1}^{V} f(x, \boldsymbol{\theta}_j)}{V}, \tag{3}$$

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

$$\lambda(x) = \sum_{j=1}^{k} \mathbf{Pr}_j(x). \tag{4}$$

For the neural network of the NER classifier, we choose the same structure with voters, which is defined at the beginning of Section 4.

4.4 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 a masked language model to conduct contextual data augmentation simultaneously. Self-training is used directly after CuPUL, and we call the classifier with self-training "CuPUL+ST".

5 Experimental Studies

5.1 Baseline Methods

We use feasible methods mentioned in Section 2 as baseline methods. First, we report distant supervision results as KB-Matching. We classify feasible DS-NER methods into two groups. 1) **DS-NER without Self-training** consists of AutoNER (Shang et al., 2018) and Conf-MPU (Zhou et al., 2022). CuPUL is directly comparable with these methods. We also include an ablation version of CuPUL (CuPUL-curr), which removes Curriculum Learning, as a baseline. 2) **DS-NER with Self-training** includes BOND (Liang et al., 2020), RoSTER (Meng et al., 2021), SCDL (Zhang et al., 2021b) and ATSEN (Qu et al., 2023) and DesERT (Wang et al., 2023). These methods apply teach-student or training augmentation steps to further boost the DS-NER performance. CuPUL+ST is directly comparable with these methods.

To ensure a fair comparison, we made some necessary code modifications to the baseline methods. For Conf-MPU, we updated the encoding model to RoBERTa. For BOND, SCDL, ATSEN, and DesSERT, we modified the hyperparameter tuning process to use the validation set instead of the test set. We employed early stopping to select the inference model. RoSTER uses fixed parameters, but the `max_seq_length` did not meet the requirements for some datasets, so we adjusted it accordingly. Specific parameters are detailed in Appendix F.

5.2 QTL Experiments

Evaluation Metrics: Due to the annotation limitation and the fact that none DS-NER methods can handle discontinued spans, we include relaxed

Method	QTL-strict	QTL-relax
DS-NER 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)
Conf-MPU	52.07 (76.30/45.37)	60.58 (91.15/51.28)
CuPUL-curr	54.75 (75.40/42.99)	62.94 (86.76/49.38)
CuPUL	**56.84** (73.03/**46.51**)	**66.18** (85.31/**54.06**)
DS-NER 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)
DesERT	54.41 (69.20/44.83)	64.23 (82.41/51.50)
CuPUL+ST	**58.87** (58.28/**59.47**)	**73.57** (73.07/**74.08**)

Table 2: Performance on QTL dataset: F1 Score (Precision/Recall) (in %). The best results are in **bold**, and the runner-up results are underlined.

Precision, Recall, and F1 scores to evaluate the performance on the QTL dataset, in addition to the strict span-level Precision, Recall, and F1 scores used in previous studies. For relaxed metrics, it deems a predicted span correct if there is at least one overlapping word with the ground truth annotation. According to the curator's feedback, the relaxed metrics can meet the practical need as identifying potential entities is more important than identifying precise boundaries.

Table 2 presents the results for all methods on the QTL dataset. Note that CuPUL without curriculum learning (CuPUL-curr) is essentially equivalent to Conf-MPU when there is one entity type. KB matching reveals that QTL annotations suffer from low recall but have relatively high precision. 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 compared to AutoNER, but still generally under-perform compared to PU-based methods. CuPUL+ST can further boost the recall compared to CuPUL, significantly outperforming all baseline methods. Specifically, strict F1 and relaxed F1 of CuPUL+ST outperform the runner-up by 5.79% and 8.00%, respectively.

5.3 Benchmark Experiments

We also re-examine all methods on existing benchmark datasets.

5.3.1 Datasets and Metrics

Datasets: We conduct experiments on six existing 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 are open-domain datasets, and BC5CDR is the biomedical domain. More details and the statistics of these datasets are summarized in Appendix B.

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

Settings: For the benchmark dataset, we use small subsets of the validation set to tune the hyperparameters including learning rate, epochs, etc, to simulate the real-life DS-NER application scenarios. Detailed settings and statistics of the validation set can be found in Appendix F.

5.3.2 Results on Benchmark Datasets

Table 3 presents the overall span-level F1 scores for all feasible and proposed methods on benchmark datasets. Note that RoSTER was tested on a different version of the OntoNotes5.0 dataset (Meng et al., 2021). Therefore, we re-run the code on OntoNotes5.0 too. We also add the results reported from previous papers for methods BOND, SCDL, ATSEN, and DesERT as a reference to the re-run results. We have the following observations.

DS-NER Without Self-training. From Table 3, it is obvious that KB-Matching generally exhibits low recall and on four of the benchmark datasets, low precision as well. In contrast, noise-aware DS-NER models, like CuPUL, significantly outperform KB-Matching. This is confirmed in the table where CuPUL achieves the best F1 scores on all datasets compared to all DS-NER models without self-training. The results of CuPUL-curr are very similar to those of Conf-MPU, except for the Twitter dataset. This difference is due to CuPUL using a different loss function to train the model obtaining the confidence score for each token. For NER tasks with more than 10 entity types (Twitter and OntoNotes5.0), we opted for cross-entropy, instead of MAE, as the loss function, which has proven to be effective. A detailed discussion can be found in Appendix E.

DS-NER With Self-training. The results for CuPUL+ST shown in Table 3 further verify that adding a self-training phase tends to enhance overall performance. When compared with DS-NER models that incorporate self-training, CuPUL+ST demonstrates superior performance on five datasets.

Method		CoNLL03	Twitter	OntoNotes5.0	Wikigold	Webpage	BC5CDR
DS-NER Without Self-training							
KB-Matching	$*$	71.40	35.83	59.51	47.76	52.45	64.32
AutoNER	$*$	67.00	26.10	67.18	47.54	51.39	79.99
Conf-MPU	†	82.39	43.21	66.04	66.58	63.32	80.06
CuPUL-curr		83.18	50.12	67.76	66.43	65.15	79.29
CuPUL		**85.09**	**54.34**	**68.06**	**70.53**	**73.10**	**80.19**
DS-NER With Self-training							
RoSTER		85.40*	43.91†	**69.10**†	58.34*	56.80†	79.78†
BOND	†	79.89	45.98	66.86	57.81	48.76	76.91
	$*$	81.15	48.01	68.35	60.07	65.74	-
SCDL	†	82.47	44.76	68.50	47.62	41.29	77.72
	$*$	83.69	51.10	68.61	64.13	68.47	-
ATSEN	†	79.39	49.38	68.22	60.72	43.03	79.95
	$*$	85.59	52.46	68.95	-	70.55	-
DesERT	†	80.57	48.21	67.94	60.32	62.88	78.21
	$*$	86.95	52.26	69.17	65.99	72.73	-
CuPUL+ST		**86.64**	**54.78**	68.20	**70.19**	**74.48**	**80.87**

Table 3: Performance on benchmark datasets with small validation: F1 Score (in %). $*$ marks the row of results reported from the original papers and † marks results we run. The best results are in **bold**.

This indicates that the CuPUL model benefits from the self-training approach, making it a versatile and effective tool for various datasets. On the OntoNotes5.0 dataset, almost all noise-aware DS-NER models have similar performances, implying that distant annotations may contain biases difficult for the models to address.

When comparing the results of BOND, SCDL, ATSEN, and DesSERT from their original papers with our re-run results, we can observe a significant decline, especially on Twitter, Wikigold, and Webpage datasets. Because these datasets are relatively small, it leads to instability in the training process and difficulty in selecting an appropriate inference model using a small validation set. The results indicate that these methods may not be robust in real-life applications. However, curriculum learning, which progresses from "easy" to "hard" samples, could stabilize the training process, making it more robust and less parameter sensitive.

5.4 Further Analysis

To further validate the effectiveness of CuPUL, we conduct additional analyses using benchmark datasets. We are unable to use the QTL dataset for this purpose due to the lack of ground truth annotations on training data.

5.4.1 Difficulty Score Estimation

For CuPUL, one assumption adopted is that difficulty scores can reflect the quality of distant supervision, where "easier" tokens have "cleaner" labels.

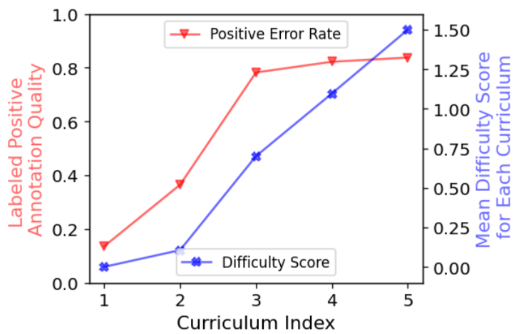

Figure 2: Token Level Positive Error Rate and Mean Difficulty Scores for Each Curriculum on Wikigold Dataset.

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 include false positives and positive type errors), and plot the rate using 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.

It is clear to see that both the average token difficulty scores and positive error rate have a clear increase with respect to the order of curricula. The figure also illustrates a strong correlation between the difficulty scores and the positive error rate of distant labels. Specifically, as the difficulty score

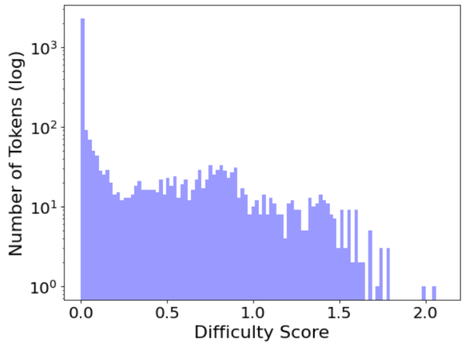

Figure 3: Distribution of the Difficulty Scores for Labeled Positives on Wikigold Dataset

increases, the quality of the distant labels 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 with 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, too. Due to the space limit, we omit the plots for other datasets.

The ablation study is discussed in Appendix I.

6 Related Work

6.1 Benchmark Datasets

To reduce the cost of human-annotated training data for NER tasks, DS-NER uses professional dictionaries or knowledgebases for annotations. Existing DS-NER benchmark datasets use NER benchmark datasets to simulate the distant supervision setting by replacing the human annotations on training datasets with knowledge base annotations (Liang et al., 2020; Shang et al., 2018; Zhou et al., 2022). There are some potential biases of current DS-NER benchmarks. 1) Only BC5CDR (Shang et al., 2018) dataset comes from professional domains where DS-NER tasks are in high demand. 2) A series of hand-crafted procedures were applied to the current DS-NER benchmarks. Such procedures are entity-type dependent and require substantial human effort, and thus may not be generalizable to other DS-NER tasks. 3) The major entity types in existing DS-NER benchmarks consist of many proper nouns (such as person's names, location, and gene names), but in many applications, the entities to recognize need not be proper nouns.

6.2 DS-NER Methods

Handling annotation errors in DS-NER tasks has drawn special attention. 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. Some methods address this issue by detecting potential entity candidates (Shang et al., 2018; Xu et al., 2023). Some methods adopt positive and unlabeled learning to tackle false negative errors from the loss estimation perspective without separate detection steps (Peng et al., 2019; Zhou et al., 2022). Due to its superiority in tolerating false negative errors, we embed Conf-MPU (Zhou et al., 2022) into our proposed method.

Another line of work simultaneously considers annotation errors of all types. Some methods propose to train an initial model and apply a self-training framework to reduce the impact of noise (Liang et al., 2020; Liu et al., 2021; Zhang et al., 2021b; Qu et al., 2023; Li et al., 2023). Methods such as RoSTER (Meng et al., 2021) and SANTA (Si et al., 2023) endeavor to diminish noise effects using loss functions tailored for noise resilience.

These DS-NER approaches obtain promising performance on existing DS-NER benchmark datasets. However, in a real-life DS-NER application, we observe that they fail to obtain satisfactory performances. These methods are trained on noisy labels initially, so the noise detection may have unknown biases and cause irreparable damage.

7 Conclusion and Future Work

In this paper, we introduce a real-life DS-NER dataset, named QTL, from the animal science domain application. We reveal the limitations of current DS-NER methods in practical DS-NER settings on the QTL dataset. To solve this issue, we propose a simple yet effective token-level curriculum-based PU learning (CuPUL) method, which strategically orders the training data from easy to hard. Our experiments show that CuPUL not only mitigates the adverse effects of noisy labels but also achieves state-of-the-art DS-NER on many datasets. Through CuPUL, we demonstrate the effectiveness of curriculum learning in improving the performance of DS-NER systems.

Limitations

The limitations of the new benchmark dataset, QTL, are discussed in Section 3.

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.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2017. Visualizing and understanding curriculum learning for long short-term memory networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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.

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.

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.

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.

Haobo Wang, Yiwen Dong, Ruixuan Xiao, Fei Huang, Gang Chen, and Junbo Zhao. 2023. Debiased and denoised entity recognition from distant supervision. In *Advances in Neural Information Processing Systems*, volume 36, pages 16650–16672. Curran Associates, Inc.

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.

Shuhui Wu, Yongliang Shen, Zeqi Tan, Wenqi Ren, Jietian Guo, Shiliang Pu, and Weiming Lu. 2023. MProto: Multi-prototype network with denoised optimal transport for distantly supervised named entity recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2361–2374, Singapore. Association for Computational Linguistics.

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.

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.

Wenkai Zhang, Hongyu Lin, Xianpei Han, Le Sun, Huidan Liu, Zhicheng Wei, and Nicholas Yuan. 2021a. 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. 2021b. 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 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.

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., 2021b) 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.

MProto (Wu et al., 2023) represents each entity type with multiple prototypes to characterize the intra-class variance among entity representations and propose a noise-robust prototype network.

DesERT (Wang et al., 2023) propose a novel self-training framework which augments the NER predicative pathway to solve innate distributional-bias in DS-NER.

B Datasets

To annotate the QTL dataset, domain experts use an online tool named TeamTat[3]. The screenshot of the tool is shown in Figure 4.

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.

The statistics of the baseline datasets are shown in Table 4.

C 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).

[3] https://www.teamtat.org/

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 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.

D Conf-MPU Risk Estimation

Conf-MPU loss function has been shown to be more robust to PU assumption violation in practice. Conf-MPU estimates the risk as

$$\mathrm{R}(f) = \sum_{i=1}^{k} \pi_i \big(\mathrm{R}^{+}_{\mathrm{P}_i}(f) + \mathrm{R}^{-}_{\tilde{\mathrm{P}}_i}(f) - \mathrm{R}^{-}_{\mathrm{P}_i}(f)\big) + \mathrm{R}^{-}_{\tilde{\mathrm{U}}}(f), \tag{5}$$

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.(5) is

$$\begin{aligned}\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) = \sum_{i=1}^{k} \frac{\pi_i}{T_i^*} \sum_{j=1}^{T_i^*} \max\Big\{0, \ell(f(x_j^{T_i^*}, \boldsymbol{\theta}), i) \\ + \mathbb{1}_{\hat{\lambda}(x_j^{T_i^*}) > \epsilon} \ell(f(x_j^{T_i^*}, \boldsymbol{\theta}), 0) \frac{1}{\hat{\lambda}(x_j^{T_i^*})} - \ell(f(x_j^{T_i^*}, \boldsymbol{\theta}), 0)\Big\} \\ + \frac{1}{T_0^*} \sum_{j=1}^{T_0^*} \Big[\mathbb{1}_{\hat{\lambda}(x_j^{T_0^*}) \leq \epsilon} \ell(f(x_j^{T_0^*}, \boldsymbol{\theta}), 0)\Big],\end{aligned} \tag{6}$$

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.

E Discussion of Loss Function

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

$$\ell_{CE} = log\, f_{i,y_i}(x; \boldsymbol{\theta}), \tag{7}$$

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):

$$\ell_{MAE} = |\boldsymbol{y}_i - f_{i,y_i}(x; \boldsymbol{\theta})|, \tag{8}$$

where $|\cdot|$ is L-1 norm of the vector and $\boldsymbol{y_i}$ denotes the one hot vector of y_i.

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.

F Hyperparameters and Experiment Settings

Detailed hyper-parameter settings for each dataset are shown in Table 5. We tune hyperparameters with Grid-Search over the small validation sets shown in Table 4. Specifically, we first tune voter hyperparameters with one voter. The learning rates are set as 1e-5 for all datasets. Voter drop negative ratios are chosen from {0.1, 0.3, 0.5}, voter training epochs from {1, 5, 10, 15}, γ from {10, 20}. Then we tune curriculum learning hyperparameters. The stage epochs are chosen from {1, 2, 3} and learning rates are chosen from {1e-5, 3e-5, 5e-5, 7e-5, 9e-5}. Other hyperparameters are set without tuning accordingly. For example, 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. For all the datasets, we train them with a batch size of 32 sentences and apply Adam optimizer (Kingma and Ba, 2014). 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 and random seed to 42. We apply cross-entropy loss

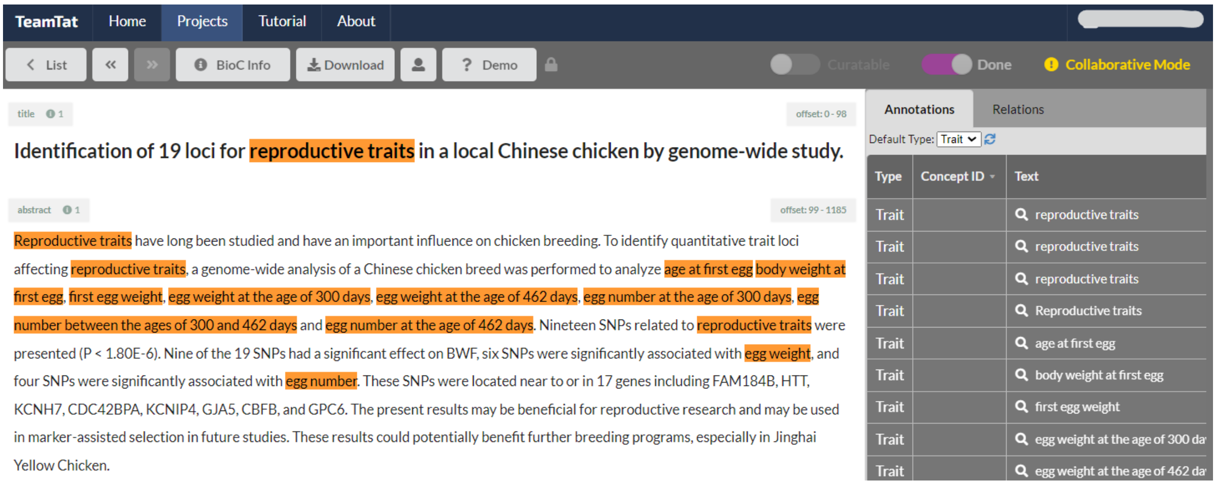

Figure 4: Screenshot for online annotation tool TeamTat.

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 4: The statistics of involved DS-NER datasets, the valid set comprises a small subset from the original dataset, whereas the train set and test set utilize the entire original dataset.

to OntoNotes5.0 and Twitter since they have more entity types and apply MAE loss to other datasets.

We use the pre-trained RoBERTa as the backbone model for both the Voter and NER classifier[4]. For all datasets, we use *roberta-base*[5]. We report single-run results for the model performance and the random seed is set to 42. We employ PyTorch[6] and conduct all experiments on a server with a Tesla A100 GPU (32G).

G Re-Examine Baseline Methods on QTL

We have explored various DS-NRE methods for QTL dataset. Our first attempt is AutoNER, which requires not only a dictionary for entity annotation but also a larger dictionary, called full-dict, for marking unknown labels, which leads to increased manual effort. To address this, we gathered a comprehensive dictionary of 26,620 potential trait entities. Unlike traditional machine learning approaches, AutoNER uses both a validation set and a test set during training and eliminates the need for hyperparameter tuning. In our exploration of RoBERTa-ES and BOND, we encountered the practice of using the test set for hyperparameter tuning during training. To rectify this, we modified the code to perform hyperparameter tuning on the validation set and conducted tests on the test set, focusing on hyperparameter tuning of early stop criteria and self-training period. For SCDL and ASTEN, we applied the hyperparameter tuning strategies outlined in the paper. Note that CuPUL without curriculum learning is essentially equivalent to Conf-MPU when there is one entity type. Therefore, Conf-MPU is not presented in the results.

H DS-NER with Small Validation Set

We evaluated the performances of BOND, SCDL, and ATSEN models when trained on a smaller validation set. RoSTER and AutoNER were excluded from this evaluation as they employ a uniform parameter set across all datasets. Additionally, Conf-MPU was not considered because its training strategy is stopping training after a predefined number of epochs, leading to it being unaffected by the validation set size.

Table 3 shows the performance of BOND, SCDL, and ATSEN. It indicates a noticeable decline for all datasets when utilizing small validation sets. The reductions in small datasets are more sig-

[4]We will release code upon paper acceptance.
[5]https://huggingface.co/roberta-base
[6]https://pytorch.org/

hyper-parameter	CoNLL03	Twitter	OntoNotes5.0	Wikigold	Webpage	BC5CDR	QTL
train set 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 5: The hyper-parameters used in CuPUL

nificant as we can observe on the wikigold and webpage. This phenomenon suggests that machine learning models trained on small datasets tend to be less stable and more susceptible to the influence of validation set size. In terms of methodology, BOND exhibited a smaller reduction in performance, possibly because a smaller validation set could lead to stopping at an incorrect position. But, the subsequent teacher-student training could mitigate this issue to some extent. The significant performance drop in SCDL and ATSEN can be attributed to their reliance on the validation set for selecting the optimal model, thereby increasing the likelihood of choosing a less effective model for the test set when the validation set is small.

I Ablation Study

Curriculum Learning To evaluate the effectiveness of curriculum learning in CuPUL, we compare it with two variations of itself. 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 7. Second, we include the result of CuPUL-curr from Table ?? 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 7.

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 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.

Distant Labels. 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. 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 5 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, 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.

J Parameter Study

Here, we perform parameter studies. Due to the simplicity of CuPUL, we mainly study two parameters: the number of voters V and the number of

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 (-/-)†
Span-based DS-NER models						
SANTA◇	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◇	80.55 (81.07/80.23)	52.86 (52.30/53.55)	-	-	-	80.39 (82.09/78.90)
CLIM◇	85.4 (-/-)	53.8 (-/-)	69.6 (-/-)	70 (-/-)	67.9 (-/-)	-
DS-NER 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)†
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**)
DS-NER 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)	-
CuPUL+ST	**86.64** (86.02/**87.27**)	**54.78** (57.32/**52.46**)	68.20 (66.57/69.11)	**70.19** (**66.96/73.7**4)	**74.48** (**76.06/72.97**)	**80.92** (**75.45/87.26**)

Table 6: 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. (2021b). ◇ marks the row of results from the method proposed paper respectively. † marks the results from Zhou et al. (2022). The best results are in **bold**, second best results are in underline.

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 7: 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.

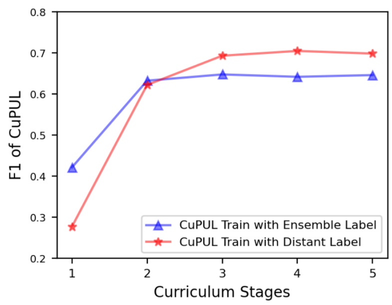

Figure 5: F1 scores of CuPUL on test data of Wikigold trained with Distant Labels (red) and Ensembled Labels from voters (blue) after each curriculum training stage.

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.

J.1 Number of Voters V

Figure 6 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.

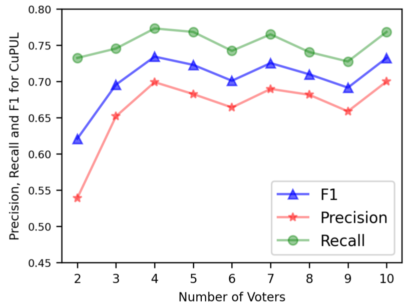

Figure 6: Span Level Precision, Recall, and F1 scores of CuPUL with respect to Number of Voters V.

J.2 Number of Curricula η

Figure 7 shows the effect of the number of curricula to CuPUL performance. Like the number of voters, when the number of curricula is small, the perfor-

Index	1	2	3	4	5	6	7	8	9	
Token	the	regiment	was	attached	to	Howe	’s	Brigade	of	···
Ground Truth	O	O	O	O	O	ORG	ORG	ORG	O	
Distant Label	O	O	O	O	O	ORG	ORG	ORG	ORG	
Curriculum #	0	0	0	0	0	2	3	2	4	
Index	10	11	12	13	14	15	16	17	18	
Token	the	IV	Corps	of	the	Army	of	the	Potomac	
Ground Truth	O	ORG	ORG	O	O	ORG	ORG	ORG	ORG	
Distant Label	O	ORG	ORG	ORG	O	ORG	ORG	O	O	
Curriculum #	0	2	2	2	0	2	2	0	0	

Table 8: Case study on Wikigold. The selected sentence is "After burying the dead on the field of Second Battle of Bull Run, the regiment was attached to Howe ’s Brigade of Couch ’s Division of the IV Corps of the Army of the Potomac where it replaced De Trobriand ’s 55th New York, Gardes Lafayette regiment on September 11, 1862." This table shows two pieces of this sentence.

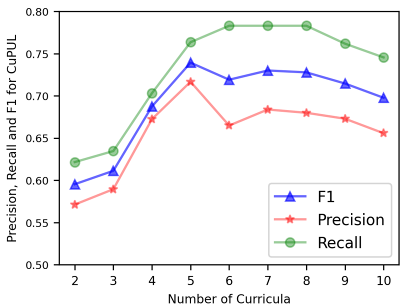

Figure 7: Span Level Precision, Recall, and F1 scores of CuPUL with respect to Number of Curricula η.

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

Table 9: Efficiency analysis on CoNLL03, m means minute, s means second

mance 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.

K 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 9. 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.

L Case Study

To gain an intuitive understanding of how the curriculum helps CuPUL, we selected a sentence from the Wikigold corpus to show how CuPUL learns. As shown in Table 8, we give the tokens, ground truth labels, the distant labels, and the Number of Curricula for each token in the sentence. We assign each token an index for ease of discussion. We display a sentence in two lines and omit some repeated parts. As can be seen from Table 8, the two “of” (token 9 and token 16) are learned in different curricula. The one with the false positive label (token 9) is arranged in the fourth curriculum, whereas the one with the correct label (token 16) is learned early (the second curriculum). This shows that the pre-trained language model has the capability of providing prediction results for each token while retaining context information, and thus, the difficulty scores can be determined at the token level.