Self-Cleaning: Improving a Named Entity Recognizer Trained on Noisy Data with a Few Clean Instances

Anonymous ACL submission

Abstract

To achieve state-of-the-art performance, one still needs to train NER models on large-scale, high-quality annotated data, an asset that is both costly and time-intensive to accumulate. In contrast, real-world applications often resort to massive low-quality labeled data through non-expert annotators via crowdsourcing and external knowledge bases via distant supervision as a cost-effective alternative. However, these annotation methods result in noisy labels, which in turn lead to a notable decline in performance. Hence, we propose to denoise the noisy NER data with guidance from a small set of clean instances. Along with the main NER model we train a discriminator model and use its outputs to recalibrate the sample weights. The discriminator is capable of detecting both span and category errors with different discriminative prompts. Results on public crowdsourcing and distant supervision datasets show that the proposed method can consistently improve performance with a small guidance set.

1 Introduction

Deep learning methods have notably improved the performance of named entity recognition (NER), but need large-scale high-quality labeled data (Lample et al., 2016; Devlin et al., 2018). In practice, collecting large-scale labeled data via crowdsourcing (Rodrigues and Pereira, 2018; Finin et al., 2010) or distant supervision (Liang et al., 2020) is far more cost-effective. However, such data is usually too noisy for direct use without further treatment (Hedderich et al., 2021; Liang et al., 2020; Chu et al., 2020).

Extensive efforts have been dedicated to develop data denoising techniques and learning strategies specifically tailored for noisy NER data. Liang et al. (2020) suggested fine-tuning pre-trained language models (PLMs) on such data, employing early stopping and self-training techniques to mitigate overfitting induced by noisy labels. Meng et al. (2021) extended the approach by using a frozen PLM to generate augmented pseudo labels for self-training. Liu et al. (2021a) further eliminated self-training labels with low estimated label confidence. Yet these denoising methods do not have a mechanism to guide error correction, thus suffer from *confirmation bias* (Tarvainen and Valpola, 2017; Arazo et al., 2020), where the learner struggles to correct its own mistakes.

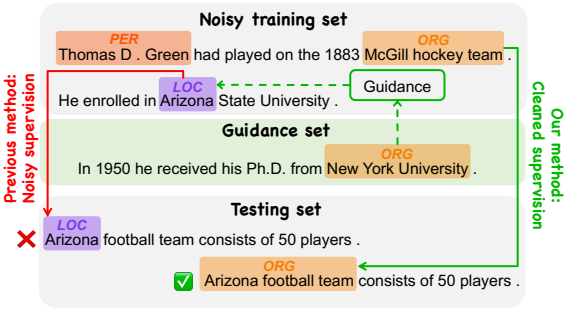


Figure 1: Illustration of the Guided Denoising Framework. The initial noisy label, `Arizona-LOC`, presents a deviation from the patterns observed in the guidance set, where geographical names preceding the term `University` are appropriately categorized into an organization entity (e.g., `New York University-ORG`). The depicted process of guided denoising (highlighted in green) ensures the retention of the accurately supervised label, `McGill hockey team-ORG`, thereby facilitating the acquisition of correct entity recognition patterns.

One natural idea to improve the performance of NER models trained on noisy data is to incorporate a small set of clean instances, which can be obtained at an acceptable cost. For example, one can let a financial professional manually label a subset of financial named entities and use them to guide the learning process on a larger, distantly-supervised financial NER dataset. We refer to the small clean set as the *guidance* set. There are a number of possibilities of how to effectively utilize the guidance set. The most straightforward method is to further fine-tune the model trained on the noisy NER data on the guidance set; we

treat this approach as a baseline to compare against. Jiang et al. (2021) augmented the noisy labels with a confidence score according to their probability of being correct given the clean data. Their heuristic-based approach is not tailored to the noisy NER problem, and as a result, it fails to identify particular types of noise in NER, such as span errors. We present a complementary approach that is effective in correcting NER-specific errors.

We propose a *Guided Denoising Framework* (shown in Figure 1) to better utilize the guidance data by eliminating noisy labels that conflict with the patterns in the guidance set. In this framework, in addition to the NER model, we also use a discriminator specifically designed to detect such conflicts.This discriminator is responsible for evaluating the accuracy of assigned labels and is trained in a few-shot manner (Brown et al., 2020; Liu et al., 2021b) with the small guidance set. Based on the analysis of real-world noisy NER datasets, we equip the discriminator, by designing different prompts, with the ability to detect two error types: span error and category error. The outputs of the discriminator are used to reweight the samples for the NER model's training. We also design a co-training strategy to improve the discriminator and the NER model in a collaborative manner. In short, we make the following contributions:

- We propose **Self-Cleaning**, a generic guided denoising framework for improving NER learning on noisy data with a small guidance set. To the best of our knowledge, this is the first instance of a denoising framework making use of an auxiliary model to correct noise in the data.
- We design a prompt-based discriminator to detect noisy NER labels. The discriminator is capable of identifying both span errors and category errors in the noisy NER data using distinct prompts.
- We report extensive experiments and ablation studies on NER benchmarks with crowdsourcing and distant supervision NER data. Results show that our approach boosts the performance.

2 Background

2.1 Named Entity Recognition

NER is the task of identifying named entities in plain text and classifying them into pre-defined entity categories, such as person, organizations, locations, etc (Li et al., 2020). Formally, we denote a sentence consisting of n tokens as $\boldsymbol{x} = [x_1, ..., x_n]$ and their corresponding labels as $\boldsymbol{y} = [y_1, ..., y_n]$. We define $D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{|D|}$ to be a labeled set. We use the `BIO` schema (Ramshaw and Marcus, 1999), where the first token of an entity with type `X` is labeled as `B-X`; the consecutive tokens of the entity are labeled as `I-X`; the non-entity tokens are labeled as `O`. An NER model $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\theta})$ takes a sentence $\boldsymbol{x}$ as input and outputs a predicted label sequence $\hat{\boldsymbol{y}}$, where $\boldsymbol{\theta}$ is its parameter set. We train the NER model by minimizing the following loss,

$$\mathcal{L} = \frac{1}{|D|} \sum_{i=1}^{|D|} \ell(\boldsymbol{y}_i, f(\boldsymbol{x}_i; \boldsymbol{\theta})), \tag{1}$$

where $\ell(\cdot, \cdot)$ can be the cross-entropy loss for token-wise classification model or negative likelihood for CRF model (Lafferty et al., 2001; Chu et al., 2019).

Following Meng et al. (2021), we build the NER model upon the RoBERTa model (Liu et al., 2019) by adding prediction heads. Specifically, we set an entity head f^e to predict whether a given token belongs to an entity and also a classification head f^c to predict the class of a given token. Both heads take the contextualized representations from a RoBERTa encoder. We decompose the original label sequence $\boldsymbol{y}$ into a sequence of binary span labels $\boldsymbol{e}$ and a sequence of category labels $\boldsymbol{c}$. The span labels are obtained by transforming `B-X` and `I-X` into positive labels (denoted as `E`), and `O` labels are remained as negative labels. In $\boldsymbol{c}$, only non-empty tokens have category labels (i.e., `B-X` and `I-X`). The entity head f^e is trained on $\boldsymbol{e}$ with the binary cross-entropy loss, while the classification head f^c is trained on $\boldsymbol{c}$ with the cross-entropy loss. This model design allows us to handle span and category errors with distinct treatments, further details of which will be provided in Section 3.3.

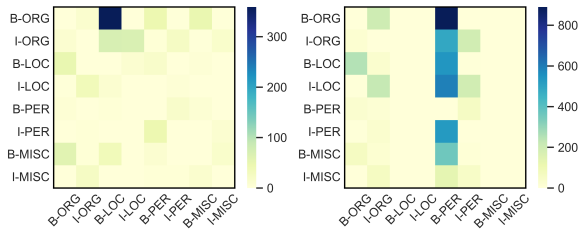
In inference, entities are first identified based on the outputs from the entity head, which are then classified using the classification head. The procedure is formalized as,

$$\hat{y} = \begin{cases} \texttt{O}, & f^e(x) \le t \\ \arg\max f^c(x), & f^e(x) > t \end{cases}, \tag{2}$$

where t is the threshold for entity identification, which is set to 0.5 by default.

2.2 NER with Noisy Data

In the noisy NER setting, the labels in D are typically collected via crowdsourcing (Rodrigues and



(a) Crowdsourcing. (b) Distant supervision.

Figure 2: Confusion matrices of CoNLL03 with crowdsourcing labels and distant supervisions. The x-axis refers to noisy labels while the y-axis are ground-truth labels. The value of each entry is the frequency of this confusion pair (e.g., mistakenly label `B-LOC` as `B-ORG`).

Pereira, 2018; Finin et al., 2010) or with distant supervisions from knowledge bases (Liang et al., 2020), which wrongly recognize many entities and often provide wrong categories for entities. Interartive self-training has proven effective in improving NER performance when learning from noisy data (Liang et al., 2020; Meng et al., 2021): the predicted label sequence $\hat{\boldsymbol{y}}_i$ from the current model iteration serves as pseudo labels for the subsequent iteration,

$$\mathcal{L}_{\text{Self}} = \frac{1}{|D|} \sum_{i=1}^{|D|} \ell(\hat{\boldsymbol{y}}_i, f(\boldsymbol{x}_i; \boldsymbol{\theta})). \quad (3)$$

In this paper, we also require a small guidance set C, the labels of which are examined by domain experts to ensure high quality. Typically, we only require $|C| \ll |D|$. It is both affordable and practical to obtain a small set of high-quality data while collecting large-scale noisy data via crowdsourcing or distant supervision. In Section 3, we will introduce our Self-Cleaning framework to guide the noisy NER learning with the guidance set.

2.3 Noise Pattern Analysis

We investigate the noise patterns on the CoNLL03 dataset with crowdsouring labels collected by Rodrigues et al. (2014) and distant supervisions collected by Liang et al. (2020). We find two types of errors: (1) **Span error**, where the span of the entity is not correctly recognized. For example, an error would occur if only `Arizona` was recognized in `Arizona State University`. The wrong entity span could either be shorter or longer than the span of the ground-truth entity. (2) **Category error**, where the entity is assigned an incorrect category.[1] An example of this would be labeling `Arizona State University` as a `location`.

We first calculated the proportion of entity spans that overlap with but do not perfectly match the ground-truth entity: it is **11.0%** for the crowdsourcing dataset and **12.8%** for the distant supervision dataset, a considerable amount of error that is likely to affect the resulting model. To analyze category errors, we present the confusion matrices on two datasets in Figure 2.[2] In the crowdsourcing dataset, `ORG` is often mislabeled as `LOC`, because the CoNLL dataset contains sports news in which team home cities or countries (locations) are also used as the name of the team (organizations), which easily confuses naive annotators. And due to the entity ambiguity in knowledge bases, all the classes could be mislabeled as `PER` in the distant supervision dataset, especially `ORG`.

Finally, a substantial proportion of ground-truth entities, **28.9%** and **25.3%**, are missing from the crowdsourcing and distant supervision datasets respectively. This finding underscores the importance of self-training, a crucial technique in previous noisy NER learning methods (Meng et al., 2021; Liu et al., 2021a; Liang et al., 2020), as it allows pseudo labels to recover these missing entities. However, in the absence of appropriate guidance, these pseudo labels may perpetuate both span and category errors. These errors, in turn, could be amplified due to the confirmation bias (Tarvainen and Valpola, 2017; Arazo et al., 2020; Chen et al., 2019), leading to a decline in performance.

3 Method: Self-Cleaning

In this section, we introduce Self-Cleaning in detail. We begin with the key component of Self-Cleaning: the prompt-based discriminator, explained in Section 3.1. We then present the training procedure (as shown in Figure 3), which consists of three stages:
Stage I: Demonstrative self-training. In this stage, high-confidence predictions from the NER model are used as pseudo labels to iteratively refine itself, a process often referred to as self-training (Liang et al., 2020; Meng et al., 2021). To enhance the robustness of the pseudo labels, we propose a mechanism called clean demonstration, in which entities from the guidance set serve as demonstrations to elicit robust predictions from the NER

[1] In the rest of the paper, we use the terms *class* and *category* interchangeably.

[2] The values of diagonal entries corresponding to correct labels are set to 0, otherwise the noise patterns in the non-diagonal entries are invisible. There are several crowdsourced annotations for each token, so we aggregate them into one label using majority voting.

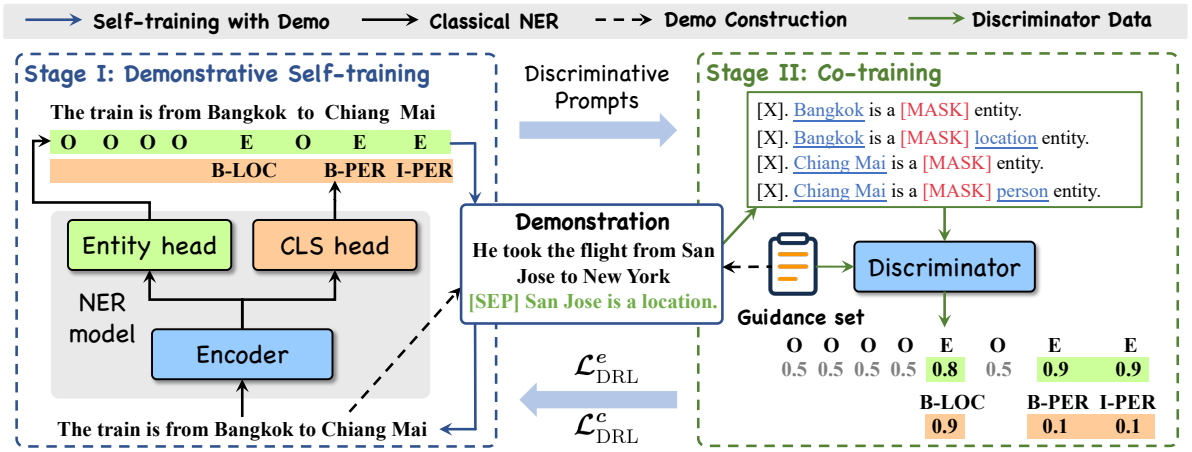


Figure 3: Overview of Self-Cleaning. In Stage I, we use the entities in the guidance set as clean demonstrations to augment the NER model's training. In Stage II, the discriminative prompts is filled with the predictions of the NER model, and then input into the prompt-based discriminator. The NER model is updated by Eq.(4) with the weights $\boldsymbol{w}^e$ and $\boldsymbol{w}^c$ provided by the discriminator. Conversely, the high-quality pseudo labels of the NER model are used to fine-tune the discriminator. In Stage III, we fine-tune the obtained NER model on the guidance set.

model. Details of the clean demonstration mechanism can be found in Section 3.2.

Stage II: Co-training. In this stage, we introduce a co-training strategy to fine-tune the NER model and the discriminator in a collaborative manner. Specifically, the discriminator's outputs are employed to guide the NER model's training by reweighting the training labels, while high-quality predictions from the NER model are chosen to augment the guidance set used for the discriminator's training. Details of the co-training strategy and the criteria for evaluating the quality of predictions are provided in Section 3.3.

Stage III: Fine-tuning. To further improve the performance, we fine-tune the obtained NER model only with the guidance set.

3.1 PLM as a Unified Discriminator

In Self-Cleaning, we use a discriminator g aims to evaluate the accuracy of assigned labels to guide the NER model's training. The rationale is that labels with low accuracy should be downweighted to mitigate their influence during model training, while the accurate labels should be retained.

We identified two error types in the noisy NER data in Section 2.3 which can be straightforwardly modeled by the descriminaror: span error and category error. Instead of training two separate discriminators to handle each type of error, we propose to train a unified discriminator using error-type-specific prompts to elicit different outputs. This approach not only saves memory space, but also leverages the power of prompt tuning, which has been shown to effectively utilize the knowledge embedded in the parameters of pre-trained language models (PLMs) (Brown et al., 2020; Li et al., 2020). With prompt tuning, it is possible to learn an effective discriminator with a small guidance set. In the following, we use RoBERTa (Liu et al., 2019) as the backbone model of the discriminator and respectivly prepare Masked Language Model (MLM) style prompts. It is important to note that other PLMs, such as generative language models (Radford et al., 2019), could be seamlessly integrated into our framework by modifying the prompts accordingly. We design the following two types of discriminative prompts,

- **Span**: `[X]. [Y] is a [MASK] entity.`
- **Category**: `[X]. [Y] is a [MASK] [Z] entity.`

`[X]` is the placeholder for a sentence $\boldsymbol{x}$, `[Y]` is the placeholder for an entity e and `[Z]` is the placeholder for a class c. The discriminator is trained to fill `correct` in the `[MASK]` token when the entity/class is appropriate given the context sentence, and `wrong` otherwise. The discriminative score of the evaluated entity or class is given by

$$w^e(e) = P_S(\texttt{correct}|\texttt{[X]} = \boldsymbol{x}, \texttt{[Y]} = e)\,,$$
$$w^c(c) = P_C(\texttt{correct}|\texttt{[X]} = \boldsymbol{x}, \texttt{[Y]} = e, \texttt{[Z]} = c)\,,$$

where P_S and P_C represent the probability associated with the span prompt and the category prompt, respectively. Given a sentence and its label sequence, we extract the entities in it and their corresponding classes from contiguous spans with `B-X`

and I-X labels in the data. For example, given [San Jose is a city] and [B-LOC, I-LOC, O, O, O], San Jose will be extracted as an entity and its class is LOC. We transform LOC and other category names into a meaningful word location which would fit naturally in a sentence. The details of the conversion can be found in Section B.

We first pre-train the discriminator to ensure a good starting point, treating the entities in the guidance set as positive samples. In this context, we will abuse the notation C to denote the set of positive samples drawn from the guidance set. We create incorrect entities and labels via data augmentations. Unlike in classification scenarios involving noisy label learning (Han et al., 2018), simulating noisy NER labels has to also provide negative examples for span errors. We investigated the datasets used in Section 2.3 and found that around 80% span-error entities deviate from the ground-truth entities by a single word. Thus, we create negative entities by randomly adding or removing a word around entities in the guidance set. For example, we transform Arizona State University into State University as a negative entity. For category errors, we randomly flip the classes of entities in the guidance set. We denote the set of negative samples as B. The discriminator is trained to minimize the following loss function,

$$\mathcal{L}_w = -\mathbb{E}_{e,c\sim C}\big[\log w^e(e) + \log w^c(c)\big] - \mathbb{E}_{\tilde{e},\tilde{c}\sim B}\big[\log(1-w^e(\tilde{e})) + \log(1-w^c(\tilde{c}))\big],$$

where $1-w^e(\tilde{e})$ and $1-w^c(\tilde{c})$ are essentially $P_S(\texttt{wrong}|\cdot,\tilde{e})$ and $P_C(\texttt{wrong}|\cdot,\tilde{c})$.

3.2 Stage I: Demonstrative Self-training

In this stage, we employ a self-training strategy enriched by demonstrations, to improve the performance of the NER model. Prior research (Zhang et al., 2022; Lee et al., 2021) has established that demonstrations can boost the robustness of PLMs. Consequently, we propose to incorporate clean entities from the guidance set into the input of the NER model to stimulate more robust outputs. These enhanced outputs are then used as pseudo labels for self-training, as specified in Eq.(3).

Technically, we follow the instance-oriented method in (Lee et al., 2021) to find demonstrations. For each sentence in the noisy training set, we first retrieve similar sentences from the guidance set using SBERT scores (Reimers and Gurevych, 2019). Then, the entities in the retrieved guidance sentences are used to form the clean demonstrations $\tilde{\boldsymbol{x}}$, which are appended as additional tokens to the original training sentence $\boldsymbol{x}$. The inputs of the NER model become $[\boldsymbol{x};\tilde{\boldsymbol{x}}]$. For example in Figure 3, San Jose-LOC is used to form the clean demonstration $\tilde{\boldsymbol{x}}$ =[SEP] San Jose is a location. During inference, we empirically found that demonstrations did not improve performance, hence we only input the original sentence $\boldsymbol{x}$ into the NER model.

3.3 Stage II: Co-training

In this stage, we fine-tune the NER model f and the discriminator g in a collaborative manner to improve the performance of both. On the one hand, the discriminator guides the NER model's training by reweighting the training labels. On the other hand, the high-quality pseudo labels generated by the NER model are used to augment the discriminator's training.

Discriminator-guided training for NER. Even though pseudo labels can effectively improve the performance, they may reproduce the noise present in the noisy training set, leading to confirmation bias (Tarvainen and Valpola, 2017) that impedes further model improvement. Therefore, we propose using the discriminator to guide self-training by reweighting the pseudo labels. As shown in Figure 3, we first extract the pseudo entities and their corresponding classes from the pseudo label sequences, and then insert them into the discriminative prompts. The outputs of the discriminator are used as weights for the pseudo labels, resulting in the following discriminative reweight loss (DRL),

$$\mathcal{L}^{e/c}_{\text{DRL}} = -\frac{1}{|D|}\sum_{i=1}^{|D|}\sum_{j=1}^{n} w^{e/c}_{ij}\log f^{e/c}_{\hat{e}_{ij}/\hat{c}_{ij}}(x_{ij};\boldsymbol{\theta}), \tag{4}$$

where $\hat{e}_{ij}$ and $\hat{c}_{ij}$ denote the pseudo span labels and category labels, respectively, for the j-th token in the i-th sentence; and $f^{e/c}_{\hat{e}_{ij}/\hat{c}_{ij}}$ refers to the entry of $\hat{e}_{ij}$ or $\hat{c}_{ij}$ in the corresponding probability distribution. Note that an entity could consist of several tokens, to which we allocate equivalent weights. We set the weights of negative span labels O to 0.5 by default to avoid overfitting on them.

Enhancing discriminator with high-quality pseudo labels. Conversely, we use high-quality pseudo labels generated by the NER model to enhance the discriminator's training. We assess the quality of pseudo labels based on two criteria: *accuracy* and *informativeness*. Firstly, We follow Yao et al. (2021) to use Jensen-Shannon divergence

(JSD) as a proxy to evaluate the accuracy of the pseudo labels of a token x_i,

$$q(\hat{e}_i) = 1 - \text{JSD}\big(f^e(x_i) \parallel \texttt{one\_hot}(e_i)\big),$$
$$q(\hat{c}_i) = 1 - \text{JSD}\big(f^c(x_i) \parallel \texttt{one\_hot}(c_i)\big),$$

where $\hat{e}_i$ and $\hat{c}_i$ are pseudo span label and category label for token x_i, while $f^e(x_i)$ and $f^c(x_i)$ are their corresponding probabilities from the entity head and classification head. e_i and c_i are observed labels in the training set, which are transformed into distributions by one-hot encoding.[3] However, the *mostly correct* pseudo labels selected by the above metric is not always helpful for the discriminator training, as they may not carry new information. Intuitively, if the discriminator shows uncertainty for its own prediction, that particular pseudo label becomes more informative. Similar to active learning (Chu and Wang, 2021; Schröder et al., 2021), we identify the most informative samples using the prediction entropy of the discriminator as a measure of uncertainty. The resulting token-level selection score $s(\cdot)$ is defined as,

$$s(\hat{e}_i) = \mathbb{H}\big(w^e(\hat{e}_i)\big) \cdot q(\hat{e}_i)\,,$$
$$s(\hat{c}_i) = \mathbb{H}\big(w^c(\hat{c}_i)\big) \cdot q(\hat{c}_i)\,,$$

where $\mathbb{H}$ is the entropy function while $w^e(\hat{e}_i)$ and $w^c(\hat{c}_i)$ are discriminative scores of pseudo labels. However, our discriminator works at the entity, not token, level. We form the entity-level selection score by averaging the token-level scores within an entity, $\frac{1}{L}\sum_i^L s(\hat{e}_i)$ and $\frac{1}{L}\sum_i^L s(\hat{c}_i)$, where L is the number of tokens in the entity. We select top-K entities as pseudo positive samples for discriminator, where K is set as a hyper-parameter. For each pseudo positive samples, we simulate pseudo negative samples in the same way as described in Section 3.1 to facilitate the discriminator training. To improve the few-shot ability of the discriminator (Gao et al., 2021), we use the approach described in Section 3.2 to generate demonstrations for the discriminator's inputs when fine-tuning and utilizing the discriminator in the co-training stage.

Lastly, in Stage III, we further fine-tune the obtained NER model only with the guidance set, as suggested in Jiang et al. (2021).

4 Experiments

4.1 Experiment Setup

Datasets. We conduct the experiments on two kinds of noisy English NER datasets:

[3]Label smoothing is used to avoid 0 entries.

Table 1: Dataset statistics.

Dataset	#Types	#Train	#Test
CoNLL03-C	4	5,985	3,453
CoNLL03	4	14,041	3,453
OntoNotes	18	59,924	8,262
Wikigold	4	1,142	274

Crowdsourcing. We use a crowdsourced NER dataset (Rodrigues and Pereira, 2018) based on CoNLL03, denoted as CoNLL03-C, where 5,985 sentences are labeled by 47 non-expert annotators. Redundant crowdsourced annotations for each token are aggregated into a single noisy label using majority voting.

Distant supervision. We use three benchmarks for distant supervision datasets including CoNLL03 (Sang and De Meulder, 2003), OntoNotes5.0 (Weischedel et al., 2013) and Wikigold (Balasuriya et al., 2009). We follow BOND (Liang et al., 2020) to obtain distant supervisions using existing knowledge bases. The noise in these datasets is more systematic, as it is mainly caused by entity ambiguity or missing entities.

We randomly sample the small guidance set from the training set with ground-truth labels, ensuring that all types are covered in the guidance set at least $\left\lfloor \frac{|C|}{\#Types} \right\rfloor$ times. We use guidance sets of 200, 500, and 50 sentences on CoNLL03 and CoNLL03-C, OntoNotes5.0, and Wikigold, respectively. For each dataset the guidance sets are less than 5% of the size of the full set. The size of the guidance set C is an important hyperparameter that impacts the final performance, so we further study its influence in Section 4.3. We use `roberta-base` as the backbone model for both the NER model and the discriminator. More implementation details can be found in Appendix A. We also conduct a comprehensive study of different model designs in Appendix D, including using generative language models (Chung et al., 2022) as discriminator backbones and different combinations of backbone models for the NER model and the discriminator.

Baselines. We compare against two broad classes of related solutions as baselines. The first class of baselines is approaches that only use noisy labels and no clean data whatsoever:

- **Distant RoBERTa**, where a RoBERTa model is fine-tuned using noisy labels.

Table 2: Results on CoNLL03-C.

Methods	Pre.	Rec.	F1
Distant RoBERTa	0.824	0.796	0.805
BOND	0.775	0.806	0.787
RoSTER	0.790	0.822	0.804
Fine-tune RoBERTa	0.695	0.699	0.694
Fine-tune RoSTER	0.778	0.831	0.802
NEEDLE	0.822	0.863	0.842
GLC	0.803	0.791	0.790
Meta-Reweight	0.768	0.835	0.799
Self-Cleaning	**0.849**	**0.876**	**0.862**

- **BOND** (Liang et al., 2020) fine-tunes a RoBERTa model on noisy labels with early-stopping, and then self-trains the resulting model.
- **RoSTER** (Meng et al., 2021) combines a noise-robust loss and ensemble training to improve robustness on noisy NER data, and then utilizes a language model augmented self-training.

The second class of baselines covers approaches that similar to Self-Cleaning also utilize a guidance set when training, but the guidance set is used in different ways:

- **Fine-tune RoBERTa**, where a RoBERTa model is fine-tuned on the guidance set.
- **Fine-tune RoSTER**, where the final model of RoSTER is fine-tuned on the guidance set.
- **NEEDLE** (Jiang et al., 2021) estimates the confidence scores of pseudo labels in the self-training stage using the histogram binning heuristic.
- **GLC** (Hendrycks et al., 2018) estimates a class-level confusion matrix using the guidance set, which is used to calibrate the loss on noisy labels.
- **Meta-Reweight** (Wu et al., 2022; Shu et al., 2019) uses a bi-level optimization framework to learn label weights. It learns the weights of pseudo labels by minimizing the meta-loss on the guidance set in the upper level and updates the NER model with the weights in the lower level.

4.2 Main results

We report the results on CoNLL03-C in Table 2 and three distant supervision datasets in Table 3, where Self-Cleaning outperforms all baselines significantly. The performance of the second group of methods is generally better than the first group, which shows the necessity of the guidance from clean data. GLC and Meta-Reweight are directly borrowed from the Machine Learning community;[4] both of them fail to improve the performance with the guidance set. GLC estimates a confusion matrix of labels using the guidance set. However, in the NER scenario, label-level confusion is not meaningful, e.g., all non-empty labels can be labeled as O due to span error. NEEDLE uses the guidance set to estimate the confidence scores by heuristics. In contrast, informed by the analysis of noise that we presented, we design a discriminator to handle two types of errors in Self-Cleaning, which has shown to be a more effective way to provide guidance in the learning on noisy NER data. Please refer to Appendix E for a detailed case study elucidating the workings of both the NER model and the discriminator in Self-Cleaning.

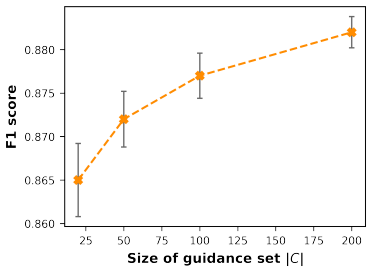


Figure 4: Results with different $|C|$.

4.3 Influence of $|C|$

The size of the guidance set influences the quality of the discriminator, and thus affects the final performance. We study the performance of Self-Cleaning with different sizes of guidance sets. For each $|C|$, we randomly sample 3 guidance sets. Due to the space limit, we report the mean and standard deviation of F1 score on CoNLL03, similar observations were also obtained on other datasets.

We show the results in Figure 4. With a smaller guidance set, the performance of Self-Cleaning drops as the quality of discriminator gets worse. Also, the performance becomes more unstable with smaller guidance sets, since the pattern distribution in different sets is different. The results show that the selection of the guidance set is crucial to the final performance. If the guidance set is of low quality or too small, the quality of the discriminator will be the bottleneck of the final performance.

In additional experiments we find that to reach

[4]GLC on OntoNotes5.0 is not reported due to its poor performance.

Table 3: Results on distant supervision NER datasets. p-value is reported to show the statistical significance.

Methods	CoNLL03			OntoNotes5.0			Wikigold		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
Distant RoBERTa	0.784	0.756	0.743	0.760	0.715	0.737	0.534	0.623	0.566
BOND	0.849	0.854	0.848	0.740	0.767	0.753	0.541	0.679	0.595
RoSTER	0.856	0.867	0.859	0.759	0.792	0.771	0.581	0.716	0.637
Fine-tune RoBERTa	0.695	0.699	0.694	0.744	0.822	0.779	0.493	0.551	0.509
Fine-tune RoSTER	0.850	0.872	0.860	0.756	0.797	0.773	0.620	0.755	0.675
NEEDLE	0.861	0.877	0.866	0.730	0.782	0.751	0.707	0.777	0.738
GLC	0.866	0.853	0.856	-	-	-	0.626	0.754	0.679
Meta-Reweight	0.839	0.866	0.851	0.737	0.781	0.755	0.609	0.746	0.665
Self-Cleaning	**0.883**	**0.882**	**0.882**	**0.809**	**0.846**	**0.826**	**0.761**	**0.798**	**0.778**
RoBERTa (Gold)	0.907	0.930	0.918	0.884	0.912	0.897	0.823	0.858	0.839
p-**value**	-	-	$<$0.005	-	-	$<$0.001	-	-	$<$0.001

the compareable performance (an F1 score of 0.880) as Self-Cleaning ($|C| = 200$), RoBERTa (Gold) needs 1,000 clean instances, five times more than Self-Cleaning. Directly fine-tuning a RoBERTa model only on the same guidance set C results in markedly worse performance as seen in Table 3. Noisy labels do effectively improve the sample efficiency of clean data.

4.4 Ablation Study

To evaluate the individual contributions of different components in Self-Cleaning, we conduct an ablation study and create the following variants: Firstly, we remove the span prompts and only reweight the category labels. Secondly, we remove the category prompts, which means the discriminator can only reweight the binary span labels. Thirdly, we remove clean demonstrations. Lastly, we remove the co-training strategy, and only use the pre-trained discriminator. Additionally, we also report the results on Stage I and Stage II.

We present the results in Table 4. As we discussed in Section 2.3, span error is a severe issue in the noisy NER data. Without the ability to detect span errors, the performance drops considerably. Also, without clean demonstrations, the few-shot ability of the discriminator is limited and the NER model lacks of guidance when generating pseudo labels, causing a drop in performance. By comparing the results of Stage I and RoSTER, we also observe that utilizing clean demonstrations leads to an improvement in the robustness of self-training. The co-training strategy is important to improve the discriminator, covering more patterns by involving pseudo positive labels from the NER model. Lastly, the improvement from Stage II to Self-Cleaning shows that fine-tuning on the guidance set can further improve the performance.

Table 4: Results of ablation study on CoNLL03.

Methods	**Pre.**	**Rec.**	**F1**
w/o Span Disc.	0.866	0.885	0.874
w/o Cat. Disc.	0.878	0.879	0.877
w/o Demonstration	0.888	0.873	0.878
w/o Co-training	0.882	0.877	0.878
Stage I	0.861	0.888	0.874
Stage II	0.881	0.879	0.880
Self-Cleaning	0.883	0.882	0.882

5 Conclusion

In this paper, we study how to improve NER models trained on noisy labeled data with a guidance set consisting of a small number of clean instances. Our research is grounded on the noise pattern analysis on the real-world noisy NER data. We identify two NER-specific error types: span error and category error. To address these errors, we propose to use a dedicated discriminator to guide the training of the NER model. This discriminator is tailored to detect the aforementioned errors using pre-defined discriminative prompts, and its outputs are used to reweight the samples for training the NER model. We design a three-stage training procedure to unleash the power of clean instances in guiding noisy NER learning. We evaluate the proposed method on a rich set of NER benchmarks with crowdsourcing labels and distant supervisions. The results show that with a few clean instances, the proposed method can boost the performance significantly.

Limitations

The discriminator is the key part of Self-Cleaning, however, it has several limitations. Firstly, the current version is able to handle noise within recognized entities but it falls short when dealing with noise in non-entity labels, i.e., `O` labels. Secondly, since the discriminator works at the entity level, an entity with even partially incorrect labels is fully downweighted. This approach could lead to the discarding of potentially useful labels, especially when category labels are very sparse. Future work might consider the development of discriminators that operate on the token level.

Additionally, it is also worth noting that in the current discriminator design, we did not make explicit assumptions about the underlying mechanisms generating span and categorical errors. The negative samples are simulated by randomly modifying tokens within entities and flipping their classes. Such negative samples may not be informative enough to capture the salient patterns needed to distinguish correct labels from incorrect ones, thereby limiting the final performance. For a more comprehensive understanding and identification of the root causes of errors in noisy NER data, future work might incorporate more advanced error modeling techniques, such as lexical analysis or cross-validation against external knowledge bases.

Ethics Statement

Learning from noisy NER data diminishes the necessity for large-scale, high-quality labeled data, thereby facilitating its use in domains where obtaining expert knowledge is costly, such as in legal and financial sectors. Our proposed method paves the way for achieving a model with reasonable performance while keeping the cost of expert-labeled data within an acceptable range. It has the potential to lower the entry barrier for novices who have limited data at their disposal.

However, we should notice our method makes it easier to attack the modeling training by poisoning the guidance set. Given the limited size of the guidance set, a subtle change could drastically alter its distribution, potentially leading to the collapse of the entire training pipeline.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jindong Chen, Ao Wang, Jiangjie Chen, Yanghua Xiao, Zhendong Chu, Jingping Liu, Jiaqing Liang, and Wei Wang. 2019. Cn-probase: a data-driven approach for large-scale chinese taxonomy construction. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1706–1709. IEEE.

Zhendong Chu, Renqin Cai, and Hongning Wang. 2019. Accounting for temporal dynamics in document streams. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1813–1822.

Zhendong Chu, Haiyun Jiang, Yanghua Xiao, and Wei Wang. 2020. Insrl: A multi-view learning framework fusing multiple information sources for distantly-supervised relation extraction. *arXiv preprint arXiv:2012.09370*.

Zhendong Chu and Hongning Wang. 2021. Improve learning from crowds via generative augmentation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 167–175.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, Mark Dredze, et al. 2010. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL Workshop on Creating Speech and Text Language Data With Amazon's Mechanical Turk*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Michael A Hedderich, Dawei Zhu, and Dietrich Klakow. 2021. Analysing the noise model error for realistic noisy label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7675–7684.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31.

Haoming Jiang, Danqing Zhang, Tianyu Cao, Bing Yin, and Tuo Zhao. 2021. Named entity recognition with small strongly labeled and large weakly labeled data. *arXiv preprint arXiv:2106.08977*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Dong-Ho Lee, Mahak Agarwal, Akshen Kadakia, Jay Pujara, and Xiang Ren. 2021. Good examples make a faster learner: Simple demonstration-based learning for low-resource ner. *arXiv preprint arXiv:2110.08454*.

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

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 & Data Mining*, pages 1054–1064.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021a. 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.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Filipe Rodrigues and Francisco Pereira. 2018. Deep learning from crowds. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine learning*, 95:165–181.

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

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2021. Revisiting uncertainty-based query strategies for active learning with transformers. *arXiv preprint arXiv:2107.05687*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

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.

Linzhi Wu, Pengjun Xie, Jie Zhou, Meishan Zhang, Chunping Ma, Guangwei Xu, and Min Zhang. 2022. Self-augmentation for named entity recognition with meta reweighting. *arXiv preprint arXiv:2204.11406*.

Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. 2021. Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5192–5201.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer.

Hongxin Zhang, Yanzhe Zhang, Ruiyi Zhang, and Diyi Yang. 2022. Robustness of demonstration-based learning under limited data scenario. *arXiv preprint arXiv:2210.10693*.

A Implementation Details

In Self-Cleaning, we use `roberta-base` as the backbone for both NER model and the discriminator. We use AdamW optimizer to optimize both NER model and the discriminator. We pre-train the discriminator with a learning rate $2e^{-5}$ and $5e^{-6}$ during co-training. The training batch size is fixed as 64. To update the NER model, we use learning rate $5e^{-6}$ for CoNLL03 and Wikigold, and $5e^{-7}$ for OntoNotes5.0. During co-training, we choose $K = 20$ pseudo entities per class to fine-tune the discriminator. We first use the noise-robust loss and ensemble training in Meng et al. (2021) to pre-train the NER model on noisy NER data, and then apply the proposed Self-Cleaning approach on the obtained model. GLC, Meta-Reweight and NEEDLE start with the same pre-trained model as Self-Cleaning. All experiments are repeated with 3 random seeds and 3 randomly sampled guidance sets. The averaged metrics are reported. We run our experiments on 2 NVIDIA GeForce RTX 2080Ti GPUs with 12 GB memory.

Table 5: Results on synthetic noisy CoNLL03.

Methods	Type	Noise Rate 0.2	0.4	0.6
RoSTER	**Span**	0.852	0.823	0.462
	Cat.	0.886	0.873	0.667
Self-Cleaning	**Span**	0.901	0.897	0.896
	Cat.	0.899	0.895	0.864

B Verbalizer

We list the mapping between NER labels and words used in our prompt-based discriminator.

- **CoNLL03**: `PER` - person, `LOC` - location, `ORG` - organization, `MISC` - other.
- **OntoNotes5.0**: `WORK_OF_ART` - work of art, `PRODUCT` - product, `NORP` - affiliation, `ORG` - organization, `FAC` - facility, `GPE` - geo-political, `LOC` - location, `PERSON` - person, `EVENT` - event, `LAW` - law, `LANGUAGE` - language, `PERCENT` - percent, `ORDINAL` - ordinal, `QUANTITY` - quantity, `CARDINAL` - cardinal, `TIME` - time, `DATE` - date, `MONEY` - money.
- **Wikigold**: `PER` - person, `LOC` - location, `ORG` - organization, `MISC` - other.

C Experiments on Synthetic Datasets

Settings. We also evaluate Self-Cleaning on synthetic data, where we manually create noisy NER data. We create two kinds of datasets based on CoNLL03 with span and category errors, respectively. For each error type, we control the noise rates. For the span error, the noise rate controls the probability to add or remove a token around a ground-truth entity. For the category error, the noise rate defines the probability of the class of a ground-truth entity to be flipped into a noisy class.
Results. We present the results in Table 5, where we also show the results of RoSTER to study the effect of noise rate. We can observe that with larger noise rate, the performance of RoSTER decreases significantly. But with our dedicated discriminator, both types of errors can be detected and down-weighted, leading to a robust performance.

Table 6: Results of various discriminator backbones.

Backbone	Pre.	Rec.	F1
`roberta-base`	0.883	0.882	0.882
`roberta-large`	0.885	0.883	0.884
`flan-t5-small`	0.878	0.873	0.875
`flan-t5-base`	0.884	0.877	0.878
`flan-t5-large`	0.889	0.877	0.881

D Experiments of different model designs

D.1 Study of Discriminator Backbones

To study the effect of different kinds of discriminators, we also incorporate Self-Cleaning with Generative Language Model (GLM) based discriminator. Specifically, we use Flan-T5, an instruction fine-tuned GLM family (Chung et al., 2022). Accordingly, we design the following two prompts,

- **Span**: `[X]`. `[Y]` is an entity. Is it correct?
- **Category**: `[X]`. `[Y]` is a `[Z]` entity. Is it correct?

The GLM-based discriminator is supposed to choose an answer from [`correct`, `wrong`]. We use the same method in Section 3.1 to create create both positive and negative samples for the pre-training of the discriminator. We consider three Flan-T5 variants with varying parameter sizes to understand the impact of model scaling. Additionally, we include results obtained by using `roberta-large` as the backbone of the discriminator.

In Table 6, we report the results on CoNLL03. We can observe that with a larger backbone model, the final performance is slightly better. Interestingly, both MLM-based and GLM-based discriminators achieve similar final performance. The success of GLM-based discriminators make it possible to introduce more powerful GLMs like the GPT family (Radford et al., 2019) in the future. However, the performance gains from larger models are marginal, suggesting a performance bottleneck. We hypothesize that the randomly generated negative samples may not be sufficiently informative. We leave how to create useful negative samples for the discriminator as an important future work.

D.2 Study of Encoder Configurations

In Self-Cleaning, we employ two `roberta-base` models as encoders for the NER model and the discriminator respectively. Additionally, we experimented with alternative designs, such as building both the NER model and the discriminator on top of a single `roberta-base` encoder. In this configuration, we added an entity head and a classification head for the NER model, while also incorporating an MLM head for the discriminator. We also conducted similar tests using the `roberta-large` model and have reported these results as well.

The results on CoNLL03 are reported in Table 7. The variants utilizing `roberta-large` show better performance than those based on `roberta-base`, owing to the increased power of the backbone model. However, when the NER model and the discriminator share a single encoder, it negatively affects the final performance. Specifically, the RoBERTa encoder, when trained on noisy NER data, tends to propagate its noise to the discriminator, thereby affecting its quality. Therefore, to ensure the isolation between clean and noisy data, we recommend employing separate encoders for the NER model and the discriminator. This design is also more flexible as we are able to use different backbone models for the NER model and the discriminator, as we did in Section D.1.

E Case Study and Analysis

E.1 Case Study of the NER model

In Table 8, we perform case study to understand the advantage of **Self-Cleaning** with a concrete example, by comparing with the best baseline without guidance **RoSTER** and with guidance **NEEDLE**. Without the guidance about the span and category

Table 7: Results of different encoder configurations.

Encoder	Pre.	Rec.	F1
one `roberta-base`	0.881	0.875	0.878
two `roberta-base`	0.883	0.882	0.882
one `roberta-large`	0.889	0.879	0.884
two `roberta-large`	0.897	0.887	0.892

errors, RoSTER fails to detect the correct span of `Sheffield Shield` and classify `Bellerive Oval` even though the span is correct. NEEDLE estimates the confidence scores according to the NER model's outputs via the histogram binning heuristic (Zadrozny and Elkan, 2001), which is ineffective to handle both span and category errors. Self-Cleaning is able to downweight the noisy entities with wrong spans and classes, leading to the correct recognition of the testing sentence.

E.2 Case Study of the Discriminator

We present some example outputs of the discriminator in Table 9. Even when the class of an entity is incorrectly identified, the discriminator can still evaluate the span correctly. For instance, in the first example, `China` is correctly recognized as an entity, but is misclassified as `ORG`. The discriminator accurately assigns a low score to the category label and a high score to the span label. However, if the span label is incorrect, the category label will also be downweighted by the discriminator. For example, in the third case, both the span score and category score are low. Intuitively, correct entity recognition is a prerequisite for correct classification, making it meaningless to preserve the category label if the span label is incorrect.

We also investigate the quality of the discriminator in Figure 5. We rank the pseudo entities in ascending order based on discriminator scores and then report the mean accuracy by comparing these pseudo entities with their corresponding ground-truth entities. As seen in the figure, entities with low discriminator scores exhibit poor quality. For instance, the accuracy of the category labels for the bottom 10% of entities is approximately 0.4. As we incorporate more high-scoring entities, the mean accuracy shows a noticeable increase. This trend elucidates the discriminator's role in guiding the training of the NER model, primarily by accurately downweighting noisy labels.

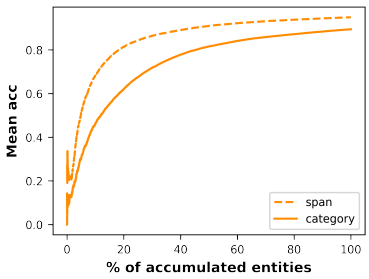


Figure 5: Mean accuracy of accumulated entities with ascending order of discriminator scores on CoNLL03 with $|C| = 200$.

Table 8: Case study of Self-Cleaning and baselines. The sentence is from CoNLL03.

Ground truth	Score on the first day of the four-day [Sheffield Shield]$_{\text{MISC}}$ match between [Tasmania]$_{\text{LOC}}$ and [Victoria]$_{\text{LOC}}$ at [Bellerive Oval]$_{\text{LOC}}$ on Friday.
RoSTER	Score on the first day of the four-day [Sheffield]$_{\text{MISC}}$ Shield match between [Tasmania]$_{\text{LOC}}$ and [Victoria]$_{\text{LOC}}$ at [Bellerive Oval]$_{\text{ORG}}$ on Friday.
NEEDLE	Score on the first day of the four-day [Sheffield]$_{\text{MISC}}$ Shield match between [Tasmania]$_{\text{LOC}}$ and [Victoria]$_{\text{LOC}}$ at [Bellerive]$_{\text{ORG}}$ Oval on Friday.
Self-Cleaning	Score on the first day of the four-day [Sheffield Shield]$_{\text{MISC}}$ match between [Tasmania]$_{\text{LOC}}$ and [Victoria]$_{\text{LOC}}$ at [Bellerive Oval]$_{\text{LOC}}$ on Friday.

Table 9: Case study of the discriminator. The sentences are from CoNLL03.

Ground truth
After the defeat of the resolution , drafted by the European Union and the United States , [China]$_{\text{LOC}}$'s Foreign Ministry thanked 26 countries for backing its motion for " no action " on the document .
After the defeat of the resolution , drafted by the European Union and the United States , China 's Foreign Ministry thanked 26 countries for backing its motion for " no action " on the document . [China] is a <mask> entity . Span score is 0.9999.
After the defeat of the resolution , drafted by the European Union and the United States , China 's Foreign Ministry thanked 26 countries for backing its motion for " no action " on the document . [China] is a <mask> [organization] entity . Category score is 0.0003.
Ground truth
Arafat subsequently cancelled a meeting between Israeli and PLO officials , on civilian affairs , at the Allenby Bridge crossing between Jordan and the [West Bank]$_{\text{LOC}}$.
Arafat subsequently cancelled a meeting between Israeli and PLO officials , on civilian affairs , at the Allenby Bridge crossing between Jordan and the West Bank . [West Bank] is a <mask> entity . Span score is 0.9993.
Arafat subsequently cancelled a meeting between Israeli and PLO officials , on civilian affairs , at the Allenby Bridge crossing between Jordan and the West Bank . [West Bank] is a <mask> [organization] entity . Category score is 0.0004.
Ground truth
At a news conference attended by approximately 50 players on Sunday , U.S. [Davis Cup]$_{\text{MISC}}$ player Todd Martin expressed the players ' outrage at the seedings .
At a news conference attended by approximately 50 players on Sunday , U.S. Davis Cup player Todd Martin expressed the players ' outrage at the seedings . [Davis] is a <mask> entity . Span score is 0.0009.
At a news conference attended by approximately 50 players on Sunday , U.S. Davis Cup player Todd Martin expressed the players ' outrage at the seedings . [Davis] is a <mask> [other] entity . Category score is 0.0346.