Distantly Supervised Named Entity Recognition via Confidence-Based Multi-Class Positive and Unlabeled Learning

Kang Zhou, Yuepei Li, Qi Li
Department of Computer Science, Iowa State University, Ames, Iowa, USA
{kangzhou, liyp0095, qli}@iastate.edu

Abstract

In this paper, we study the named entity recognition (NER) problem under distant supervision. Due to the incompleteness of the external dictionaries and/or knowledge bases, such distantly annotated training data usually suffer from a high false negative rate. To this end, we formulate the Distantly Supervised NER (DS-NER) problem via Multi-class Positive and Unlabeled (MPU) learning and propose a theoretically and practically novel CONFidence-based MPU (**Conf-MPU**) approach. To handle the incomplete annotations, Conf-MPU consists of two steps. First, a confidence score is estimated for each token of being an entity token. Then, the proposed Conf-MPU risk estimation is applied to train a multi-class classifier for the NER task. Thorough experiments on two benchmark datasets labeled by various external knowledge demonstrate the superiority of the proposed Conf-MPU over existing DS-NER methods. Our code is available at Github[1].

1 Introduction

Named Entity Recognition (NER) aims to detect entity mentions from text and classify them into predefined types. It is a fundamental task in information extraction and many other downstream tasks (Gábor et al., 2018; Luan et al., 2017; Giorgi et al., 2019). However, the necessity of extensive human efforts to annotate a large amount of training data imposes restrictions on the state-of-the-art supervised deep learning methods, especially in professional fields.

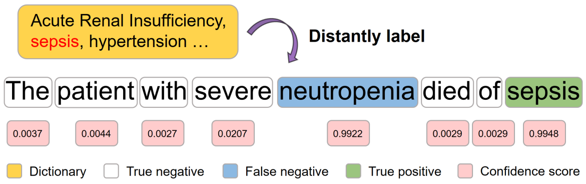

Figure 1: Distant labeling example for the entity type of `Disease` using a dictionary.

To address this problem, distantly supervised methods spring up (Yang et al., 2018; Shang et al., 2018; Mayhew et al., 2019; Cao et al., 2019; Peng et al., 2019; Liang et al., 2020; Liu et al., 2021; Zhang et al., 2021), which aim to train NER models using automatically annotated training data based on external knowledge such as dictionaries and knowledge bases. Observed by previous DS-NER methods (Shang et al., 2018; Peng et al., 2019), distant labels provided by reliable dictionaries are usually of high precision. However, such distant labeling suffers a major drawback — incomplete labeling. This is due to the fact that most existing dictionaries and knowledge bases have limited coverage on entities. Hence simply treating unlabeled samples (*i.e.*, unmatched tokens) as negative ones will introduce a high false negative rate (*e.g.*, "neutropenia" in Figure 1) compared with human-annotated training data, and further mislead a supervised NER model to overfit to false negative samples and seriously affect its recall.

Recently, binary Positive and Unlabeled (PU) learning is applied to DS-NER tasks for this challenge (Peng et al., 2019). PU learning performs classification using only limited labeled positive data and unlabeled data, thus naturally suitable for handling distant supervision, where external knowledge often has a limited coverage on positive samples. However, binary PU learning has several drawbacks in real DS-NER tasks. It applies the one-vs-all strategy to convert a multi-class classification problem into multiple binary classification problems, and thus suffers from two weaknesses. First, it is not efficient, especially in the case where there are many entity types. For a NER task with n entity types, n binary classifiers need to be trained. Second, the scale of predicted confidence values may differ among those binary classifiers, which may not guarantee a mutually beneficial inference for the final prediction (Bishop, 2006).

[1] https://github.com/kangISU/Conf-MPU-DS-NER

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7198 - 7211
May 22-27, 2022 ©2022 Association for Computational Linguistics

Furthermore, the PU learning theory is built on a fundamental assumption of data distribution that unlabeled data can accurately reveal the overall distribution (*i.e.*, the marginal distribution of the target field) (Bekker and Davis, 2020). In DS-NER tasks, the distantly annotated training data may not fit the assumption well: It depends on the coverage of used dictionaries or knowledge bases on the entities. Our empirical studies validate that violation of this assumption can significantly impact the performance of PU learning.

To address these challenges in DS-NER tasks and PU learning, we propose a CONFidence-based Multi-class Positive and Unlabeled (**Conf-MPU**) learning framework. The proposed Conf-MPU can handle different levels of false negative rates brought by dictionaries of various coverage and does not overfit to the distantly labeled training data. It consists of two steps. Specifically, given the distantly labeled training data, we first carry out a token-level binary classification to estimate the confidence score (a probability value in $[0,1]$) of a token being an *entity token* (*i.e.*, a token of a named entity). Then, we perform the NER classification using a neural network model with the proposed Conf-MPU risk estimator, which incorporates the confidence scores obtained from the first step in the risk estimation, to alleviate the impact of annotation imperfection. It is worth noting that the two-step strategy of Conf-MPU needs to train only two classifiers for any DS-NER tasks with arbitrary number of entity types, which is more efficient than previous binary PU learning.

In summary, our main contributions are:

- We propose Conf-MPU, a theoretically and practically novel approach for the DS-NER task. Conf-MPU enriches the PU learning theory with solid theoretical analysis.

- We verify that the practical use of traditional PU learning is subject to its theoretical assumption, which can be released by Conf-MPU. As far as we know, this is the first work specially dealing with such a practical problem.

- We empirically demonstrate that Conf-MPU with a two-step strategy can significantly alleviate the impact of incomplete annotations during the model training and outperform the state-of-the-art DS-NER methods on benchmark datasets.

2 Preliminaries

In this section, we briefly review the risk formulations of standard supervised learning and PU learning in the binary classification setting.

2.1 Standard Binary Supervised Learning

Suppose that the data follow an unknown probability distribution with density $p(\boldsymbol{x}, y)$. Let $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y} = \{0, 1\}$, where 0 and 1 indicate negative and positive classes, respectively. The goal is to learn a decision function $f : \mathcal{X} \to \mathcal{Y}$ by minimizing the expected classification risk:

$$\mathrm{R}(f) = \pi \mathrm{R}_{\mathrm{P}}^{+}(f) + (1 - \pi)\mathrm{R}_{\mathrm{N}}^{-}(f). \qquad (1)$$

In this function, $\pi = p(y = 1)$ is the prior of the positive class. $\mathrm{R}_{\mathrm{P}}^{+}(f) = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|y=1)} [\ell(f(\boldsymbol{x}), 1)]$ and $\mathrm{R}_{\mathrm{N}}^{-}(f) = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|y=0)} [\ell(f(\boldsymbol{x}), 0)]$ denote the expected classification risks on the positive and negative classes, respectively, where $\mathbb{E}$ denotes expectation and its subscript indicates the data distribution on which the expectation is computed, and the loss function is represented by ℓ.

In supervised learning setting, we are given both labeled positive and negative data that are sampled independently from $p_{\mathrm{P}}(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = 1)$ and $p_{\mathrm{N}}(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = 0)$ as $\mathcal{X}_{\mathrm{P}} = \{\boldsymbol{x}_j^{\mathrm{P}}\}_{j=1}^{n_{\mathrm{P}}}$ and $\mathcal{X}_{\mathrm{N}} = \{\boldsymbol{x}_j^{\mathrm{N}}\}_{j=1}^{n_{\mathrm{N}}}$, respectively. Then Eq. 1 can be estimated by $\hat{\mathrm{R}}_{\mathrm{PN}}(f) = \pi \hat{\mathrm{R}}_{\mathrm{P}}^{+}(f) + (1 - \pi)\hat{\mathrm{R}}_{\mathrm{N}}^{-}(f)$, where $\hat{\mathrm{R}}_{\mathrm{P}}^{+}(f) = \frac{1}{n_{\mathrm{P}}} \sum_{j=1}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_j^{\mathrm{P}}), 1)$ and $\hat{\mathrm{R}}_{\mathrm{N}}^{-}(f) = \frac{1}{n_{\mathrm{N}}} \sum_{j=1}^{n_{\mathrm{N}}} \ell(f(\boldsymbol{x}_j^{\mathrm{N}}), 0)$.

2.2 Binary PU Learning

In PU learning setting, we have only access to labeled positive data $\mathcal{X}_{\mathrm{P}}$ and unlabeled data $\mathcal{X}_{\mathrm{U}} = \{\boldsymbol{x}_j^{\mathrm{U}}\}_{j=1}^{n_{\mathrm{U}}}$ drawn from $p_{\mathrm{U}}(\boldsymbol{x})$ instead of labeled negative data $\mathcal{X}_{\mathrm{N}}$, which indicates that the classification risk Eq. 1 can not be directly estimated as done in supervised learning setting. For this problem, Du Plessis et al. (2014) propose the expected classification risk formulation of PU learning:

$$\mathrm{R}(f) = \pi \mathrm{R}_{\mathrm{P}}^{+}(f) + \mathrm{R}_{\mathrm{U}}^{-}(f) - \pi \mathrm{R}_{\mathrm{P}}^{-}(f), \qquad (2)$$

where $\mathrm{R}_{\mathrm{U}}^{-}(f) = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} [\ell(f(\boldsymbol{x}), 0)]$ and $\mathrm{R}_{\mathrm{P}}^{-}(f) = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|y=1)} [\ell(f(\boldsymbol{x}), 0)]$. Here $\mathrm{R}_{\mathrm{U}}^{-}(f) - \pi \mathrm{R}_{\mathrm{P}}^{-}(f)$ can alternatively represent $(1 - \pi)\mathrm{R}_{\mathrm{N}}^{-}(f)$ because $p(y = 0)p(\boldsymbol{x} \mid y = 0) = p(\boldsymbol{x}) - p(y = 1)p(\boldsymbol{x} \mid y = 1)$.

PU learning assumes that unlabeled data $\mathcal{X}_{\mathrm{U}}$ can reflect the true overall distribution, that is, $p_{\mathrm{U}}(\boldsymbol{x}) = p(\boldsymbol{x})$, due to unlabeled data consisting of both positive and negative data, under which Eq. 2 can be approximated by $\hat{\mathrm{R}}_{\mathrm{PU}}(f) = \pi\hat{\mathrm{R}}_{\mathrm{P}}^{+}(f) + \hat{\mathrm{R}}_{\mathrm{U}}^{-}(f) - \pi\hat{\mathrm{R}}_{\mathrm{P}}^{-}(f)$, where $\hat{\mathrm{R}}_{\mathrm{U}}^{-}(f) = \frac{1}{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{U}}} \ell(f(\boldsymbol{x}_j^{\mathrm{U}}), 0)$ and $\hat{\mathrm{R}}_{\mathrm{P}}^{-}(f) = \frac{1}{n_{\mathrm{P}}}\sum_{j=1}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_j^{\mathrm{P}}), 0)$.

3 Methodology

In this section, we introduce the proposed Conf-MPU learning for DS-NER in the multi-class classification setting with solid theoretical analysis.

3.1 Conf-MPU Learning

Let $y \in \mathcal{Y} = \{0, 1, 2, ..., k\}$, where 0 refers to the negative class and $1, ..., k$ refer to k positive classes. The goal in multi-class classification is to minimize the following expected classification risk:

$$\mathrm{R}(f) = \sum_{i=1}^{k} \pi_i \mathrm{R}_{\mathrm{P}_i}^{+}(f) + (1 - \sum_{i=1}^{k} \pi_i)\mathrm{R}_{\mathrm{N}}^{-}(f), \quad (3)$$

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

Following PU learning setting, there are only labeled positive data $\mathcal{X}_{\mathrm{P}_i} = \{\boldsymbol{x}_j^{\mathrm{P}_i}\}_{j=1}^{n_{\mathrm{P}_i}}$ drawn from $p_{\mathrm{P}_i}(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = i)$ where $i \in \{1, ..., k\}$, and unlabeled data $\mathcal{X}_{\mathrm{U}}$. Thus we can not directly estimate Eq. 3. Here we adopt the same probability principle as applied in binary PU learning to alternatively compute the risk on negative data. Since $p(y = 0)p(\boldsymbol{x} \mid y = 0) = p(\boldsymbol{x}) - \sum_{i=1}^{k} p(y = i)p(\boldsymbol{x} \mid y = i)$, we can further derive Eq. 3 as:

$$\mathrm{R}(f) = \sum_{i=1}^{k} \pi_i \mathrm{R}_{\mathrm{P}_i}^{+}(f) + \mathrm{R}_{\mathrm{U}}^{-}(f) - \sum_{i=1}^{k} \pi_i \mathrm{R}_{\mathrm{P}_i}^{-}(f), \quad (4)$$

where $\mathrm{R}_{\mathrm{U}}^{-}(f) - \sum_{i=1}^{k} \pi_i \mathrm{R}_{\mathrm{P}_i}^{-}(f)$ theoretically plays the role of $(1 - \sum_{i=1}^{k} \pi_i)\mathrm{R}_{\mathrm{N}}^{-}(f)$, and $\mathrm{R}_{\mathrm{P}_i}^{-}(f) = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{P}_i}(\boldsymbol{x})}\left[\ell(f(\boldsymbol{x}), 0)\right]$. We denote this classification risk as MPU, whose estimation requires the same assumption of data distribution as binary PU learning does, namely, $p_{\mathrm{U}}(\boldsymbol{x}) = p(\boldsymbol{x})$. We refer this assumption to **PU assumption** hereinafter for convenience. Under PU assumption, $\mathcal{X}_{\mathrm{U}}$ can be used to estimate $\mathrm{R}_{\mathrm{U}}^{-}(f)$. Specifically, MPU risk estimator is given as:

$$\hat{\mathrm{R}}_{\mathrm{MPU}}(f) = \sum_{i=1}^{k} \frac{\pi_i}{n_{\mathrm{P}_i}} \sum_{j=1}^{n_{\mathrm{P}_i}} \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), i) + \max\Bigg\{0, \\ \frac{1}{n_{\mathrm{U}}} \sum_{j=1}^{n_{\mathrm{U}}} \ell(f(\boldsymbol{x}_j^{\mathrm{U}}), 0) - \sum_{i=1}^{k} \frac{\pi_i}{n_{\mathrm{P}_i}} \sum_{j=1}^{n_{\mathrm{P}_i}} \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), 0)\Bigg\}, \quad (5)$$

with a non-negative constraint inspired by Kiryo et al. (2017) ensuring the risk on the negative class is non-negative.

However, PU assumption can be violated in real distant supervision scenarios. The distribution of unlabeled data $p_{\mathrm{U}}(\boldsymbol{x})$ may be different from the overall distribution $p(\boldsymbol{x})$ especially when the distant supervision has a good coverage. In such cases, the unlabeled data will have a distribution closer to the distribution of the true negative data $p_{\mathrm{N}}(\boldsymbol{x})$ instead of the overall distribution $p(\boldsymbol{x})$. Thus, the risk estimation of $\mathrm{R}_{\mathrm{U}}^{-}(f)$ based on the assumption, in either MPU or binary PU, may be biased.

To alleviate such estimation bias, we derive a novel Conf-MPU risk function from MPU. With the context of NER tasks, we observe that almost any combination of characters could be part of a named entity. Based on this observation, mathematically, we define $\lambda(\boldsymbol{x}) = p(y > 0 \mid \boldsymbol{x})$ to determine the confidence score of a token being an entity token, no matter what entity type it belongs to, and further assume that $\lambda(\boldsymbol{x}) > 0$. Under this assumption, we can further decompose $\mathrm{R}_{\mathrm{U}}^{-}(f)$ in Eq. 4 by involving a threshold parameter $0 < \tau \leq 1$ as follows:

$$\mathrm{R}_{\mathrm{U}}^{-}(f) = \sum_{i=1}^{k} \pi_i \mathrm{R}_{\tilde{\mathrm{P}}_i}^{-}(f) + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f), \quad (6)$$

where $\mathrm{R}_{\tilde{\mathrm{P}}_i}^{-}(f) = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{P}_i}(\boldsymbol{x}|\lambda(\boldsymbol{x})>\tau)}\left[\ell(f(\boldsymbol{x}), 0)\frac{1}{\lambda(\boldsymbol{x})}\right]$ and $\mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\lambda(\boldsymbol{x})\leq\tau)}\left[\ell(f(\boldsymbol{x}), 0)\right]$. The detailed proof is shown as follows.

Proof. Since $\lambda(\boldsymbol{x}) > 0$ and $0 < \tau \leq 1$, we have

$$\begin{aligned}
&\mathrm{R}_{\mathrm{U}}^{-}(f) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\ell(f(\boldsymbol{x}), 0)\right] \\
&= \int_{\lambda(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}), 0)p(\boldsymbol{x})d\boldsymbol{x} + \int_{\lambda(\boldsymbol{x})\leq\tau} \ell(f(\boldsymbol{x}), 0)p(\boldsymbol{x})d\boldsymbol{x} \\
&= \int_{\lambda(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}), 0)\frac{p(\boldsymbol{x})p(\boldsymbol{x}, y > 0)}{p(\boldsymbol{x}, y > 0)}d\boldsymbol{x} + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f) \\
&= \int_{\lambda(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}), 0)\frac{p(\boldsymbol{x})p(\boldsymbol{x} \mid y > 0)p(y > 0)}{p(y > 0 \mid \boldsymbol{x})p(\boldsymbol{x})}d\boldsymbol{x} + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f) \\
&= \int_{\lambda(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}), 0)\frac{p(\boldsymbol{x} \mid y > 0)p(y > 0)}{\lambda(\boldsymbol{x})}d\boldsymbol{x} + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f) \\
&= \sum_{i=1}^{k} p(y = i) \int_{\lambda(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}), 0)\frac{p(\boldsymbol{x} \mid y = i)}{\lambda(\boldsymbol{x})}d\boldsymbol{x} + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f) \\
&= \sum_{i=1}^{k} \pi_i \int_{\lambda(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}), 0)\frac{1}{\lambda(\boldsymbol{x})}p(\boldsymbol{x} \mid y = i)d\boldsymbol{x} + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f) \\
&= \sum_{i=1}^{k} \pi_i \mathrm{R}_{\tilde{\mathrm{P}}_i}^{-}(f) + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f).
\end{aligned}$$

Consequently, we obtain the expected classification risk of Conf-MPU by substituting $\mathrm{R}_{\mathrm{U}}^{-}(f)$ in Eq. 4 with Eq. 6 as follows:

$$\mathrm{R}(f) = \sum_{i=1}^{k} \pi_i \big(\mathrm{R}_{\mathrm{P}_i}^{+}(f) + \mathrm{R}_{\tilde{\mathrm{P}}_i}^{-}(f) - \mathrm{R}_{\mathrm{P}_i}^{-}(f)\big) + \mathrm{R}_{\tilde{\mathrm{U}}}^{-}(f). \quad (7)$$

Given a reliable λ and a proper τ, $\lambda(\boldsymbol{x}) > \tau$ indicates $\boldsymbol{x}$ being an entity token (a positive sample), otherwise a non-entity token (a negative sample), which further induces that $p_{\mathrm{P}_i}(\boldsymbol{x} \mid \lambda(\boldsymbol{x}) > \tau) \approx p_{\mathrm{P}_i}(\boldsymbol{x})$, and $p(\boldsymbol{x} \mid \lambda(\boldsymbol{x}) \leq \tau) \approx p_{\mathrm{U}}(\boldsymbol{x} \mid \lambda(\boldsymbol{x}) \leq \tau) \approx p_{\mathrm{N}}(\boldsymbol{x})$ even if $p_{\mathrm{U}}(\boldsymbol{x})$ is different from $p(\boldsymbol{x})$. Thus, empirically $\mathrm{R}^{-}_{\tilde{\mathrm{P}}_i}(f)$ and $\mathrm{R}^{-}_{\tilde{\mathrm{U}}}(f)$ can be estimated with less bias using $\mathcal{X}_{\mathrm{P}_i}$ and $\mathcal{X}_{\mathrm{U}}$, respectively, which further leads to a more precise estimation of $\mathrm{R}^{-}_{\mathrm{U}}(f)$. This is the mechanism that Conf-MPU can significantly reduce estimation bias in practice, even if PU assumption is violated. Specifically, Conf-MPU risk estimator can be expressed as:

$$\begin{aligned}\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) = \sum_{i=1}^{k} \frac{\pi_i}{n_{\mathrm{P}_i}} \sum_{j=1}^{n_{\mathrm{P}_i}} \max\Big\{0, \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), i) \\ + \mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i}) > \tau} \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), 0) \frac{1}{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})} - \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), 0)\Big\} \\ + \frac{1}{n_{\mathrm{U}}} \sum_{j=1}^{n_{\mathrm{U}}} \Big[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{U}}) \leq \tau} \ell(f(\boldsymbol{x}_j^{\mathrm{U}}), 0)\Big], \qquad (8)\end{aligned}$$

with a constraint to guarantee a non-negative loss on each labeled positive sample, where $\hat{\lambda}$ is an empirical confidence score estimator. In DS-NER tasks, we formulate the sub-task of estimating $\lambda(\boldsymbol{x})$ as a token-level binary classification problem which also uses distant labels. In practice, a classifier with a sigmoid output layer for this sub-task can guarantee $\hat{\lambda}(\boldsymbol{x}) > 0$.

3.2 Insights into Conf-MPU Risk Estimator

Targeting on the challenge of high false negative rates in training data, we give the following analysis to offer some insights into the Conf-MPU risk estimator. For ease of expression, we use letters to denote the terms in Eq. (8): $\mathbb{A} = \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), i)$, $\mathbb{B} = \mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i}) > \tau} \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), 0) \frac{1}{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})}$, $\mathbb{C} = \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), 0)$, $\mathbb{D} = \mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{U}}) \leq \tau} \ell(f(\boldsymbol{x}_j^{\mathrm{U}}), 0)$. The threshold τ is set to 0.5 by default. We assume that $\hat{\lambda}(\boldsymbol{x})$ of an entity token is close to 1 (*i.e.*, $\hat{\lambda}(\boldsymbol{x}) > \tau$), and $\hat{\lambda}(\boldsymbol{x})$ of a non-entity token is close to 0 (*i.e.*, $\hat{\lambda}(\boldsymbol{x}) \leq \tau$).

For a true positive sample (*e.g.*, "sepsis" distantly labeled in Figure 1), the loss is computed by $\mathbb{A}+\mathbb{B}-\mathbb{C}$, where $\mathbb{B}$ is involved because its confidence score is larger than the threshold. Since $1/\hat{\lambda}(\boldsymbol{x})$ is close to 1, $\mathbb{B}-\mathbb{C}$ is almost 0 but positive, and thus the loss on this sample approximately equals to $\mathbb{A}$, which is very similar with the loss on a positive sample in standard supervised learning. For a true negative sample (*e.g.*, "patient" unlabeled in Figure 1), the loss is calculated by $\mathbb{D}$ due to its confidence score is less than the threshold. So the minimization for $\mathbb{D}$ enables the model to learn from this true negative sample. For a false negative sample (*e.g.*, "neutropenia" unlabeled in Figure 1), the loss is not counted, because its confidence score is larger than the threshold and thus $\mathbb{D}$ is not calculated. It is the mechanism that Conf-MPU handles false negative samples from unlabeled data.

3.3 Estimation Error Bound

Here we establish an estimation error bound for the proposed Conf-MPU risk estimator (Eq. 8) to show the guaranteed performance.

Theorem 1. *Let* $f^* = \arg\min_{f \in \mathcal{F}} \mathrm{R}(f)$ *and* $\hat{f}_{\mathrm{Conf-MPU}} = \arg\min_{f \in \mathcal{F}} \hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f)$. *Assume that* $\ell(\cdot) \in [0, C_l]$ *and* ℓ *is Lipschitz continuous on the interval* $[-C_g, C_g]$ *with a Lipschitz constant* L_l, *where* $C_l, C_g > 0$. *Also suppose that* $\hat{\lambda}$ *is a fixed function independent of data used to compute* $\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f)$ *and* $\tau \in (0, 1]$. *Let* $\zeta = p(\hat{\lambda}(\boldsymbol{x}) \leq \tau)$ *and* $\epsilon = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}\Big[|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})|^2\Big]$. *Then for any* $\delta > 0$, *with probability at least* $1 - \delta$,

$$\begin{aligned}\mathrm{R}(\hat{f}_{\mathrm{Conf-MPU}}) - \mathrm{R}(f^*) \leq \sum_{i=1}^{k} 2\pi_i \frac{(\tau+1)C_l}{\tau} \\ + \sum_{i=1}^{k} 2\pi_i \left[\frac{2L_l}{\tau} \mathfrak{R}_{n_{\mathrm{P}_i}, p_{\mathrm{P}_i}(\boldsymbol{x})}(\mathcal{F}) + \frac{(\tau+1)C_l}{\tau} \sqrt{\frac{\log \frac{k+1}{\delta}}{2n_{\mathrm{P}_i}}}\right] \\ + 4L_l \mathfrak{R}_{n_{\mathrm{U}}, p(\boldsymbol{x})}(\mathcal{F}) + 2C_l \sqrt{\frac{\log \frac{k+1}{\delta}}{2n_{\mathrm{U}}}} + \frac{2C_l}{\tau} \sqrt{(1-\zeta)\epsilon}.\end{aligned}$$

In Theorem 1, $\mathcal{F}$ is the function class and $\mathfrak{R}_{n_{\mathrm{P}_i}, p_{\mathrm{P}_i}(\boldsymbol{x})}(\mathcal{F})$ is the Rademacher complexity of the function class $\mathcal{F}$ for the sampling of size n_{P_i} from the distribution $p_{\mathrm{P}_i}(\boldsymbol{x})$ and $\mathfrak{R}_{n_{\mathrm{U}}, p(\boldsymbol{x})}(\mathcal{F})$ follows a similar definition. We relegate this proof to the Appendix.

4 DS-NER Classification

In this section, we describe the setup for the DS-NER classification.

4.1 Generation of Distant Labels

In DS-NER tasks, professional dictionaries (*e.g.*, UMLS) and knowledge bases (*e.g.*, Wikidata) are used to automatically generate distant labels. Distant labeling by dictionaries employs some string matching algorithms to map training samples to dictionaries (Ren et al., 2015; Giannakopoulos et al., 2017; Peng et al., 2019), while knowledge bases utilize public APIs to perform such distant labeling.

4.2 Classifiers

The proposed Conf-MPU risk estimator can be applied on any NER classifiers where the task is to predict the label for each token. For example, BERT (Devlin et al., 2019) can be used as the underlying NER model, and then the Conf-MPU risk estimation can be used to calculate the classification risks. We use **Conf-MPU$_{\text{BERT}}$** to denote this method. BiLSTM (Chiu and Nichols, 2016) is another popular choice for NER models. Ratinov and Roth (2009); Passos et al. (2014); Chiu and Nichols (2016) demonstrate that using lexicons as external features can improve NER performance. With the dictionaries, we extract the lexicon features as follows. For each token, we match its contextual words within a window size against entries in the dictionaries. If there is any successful matching, a binary indicator is set to 1, otherwise 0. With the window size of n, we can form an n-bit vector, which is appended to the input embedding. We denote the model of BiLSTM with the lexicon feature engineering as LBiLSTM, and denote **Conf-MPU$_{\text{LBiLSTM}}$** as the LBiLSTM-based classifier with Conf-MPU risk estimation.

For the first step of estimating confidence scores, we build a token-level binary classifier (*i.e.*, λ) based on LBiLSTM to output scores. To be consistent with PU learning setting, this classifier is equipped with a binary PU learning risk estimator (*i.e.*, $\hat{\mathrm{R}}_{\mathrm{PU}}(\lambda)$).

4.3 Prior Estimation

Unlike in supervised learning where priors (*i.e.*, π_i) can be easily obtained from human annotations, we cannot directly acquire them from distant annotations. In PU learning research, there are some methods proposed specifically for estimating the priors (Bekker and Davis, 2018; Jain et al., 2016; Du Plessis and Sugiyama, 2014). Here we adopt the most effective TIcE algorithm from Bekker and Davis (2018) to perform prior estimation.

4.4 Loss Function

Peng et al. (2019) point out that a bounded loss function can help avoid overfitting in PU learning setting. We also confirm this argument in our empirical studies. Thus, instead of using the common unbounded cross entropy loss function, we adopt the mean absolute error (MAE) as the loss function for Conf-MPU and other PU learning methods in our experiments. Given its label $\boldsymbol{y}$ in the one-hot form, the loss on a token $\boldsymbol{x}$ is defined by:

$$\ell(f(\boldsymbol{x}), \boldsymbol{y}) = \frac{1}{k+1} \sum_{i=0}^{k} |\boldsymbol{y}^{(i)} - f(\boldsymbol{x})^{(i)}|,$$

where $f(\boldsymbol{x})$ is the softmax output, and both $\boldsymbol{y}$ and $f(\boldsymbol{x})$ are in $k+1$ dimensions. Note that $\ell(f(\boldsymbol{x}), \boldsymbol{y}) \in [0, \frac{2}{k+1}]$ is bounded.

4.5 Post-Processing

In DS-NER tasks, self-training strategies as post-processing can often further improve the performance, such as iteratively enriching dictionaries based on the model predictions (Peng et al., 2019), or iteratively training a teacher-student framework (Liang et al., 2020). The discussion for self-training framework is out of the scope of this paper and we refer the readers to Zoph et al. (2020) for more information.

5 Experiments

In this section, we evaluate the proposed Conf-MPU and compare with other baseline methods.

5.1 Experimental Setup

5.1.1 Training Data and Evaluation Metrics

We consider two benchmark NER datasets from different domains: (1) **BC5CDR** comes from biomedical domain. It consists of 1,500 articles, containing 15,935 `Chemical` and 12,852 `Disease` mentions; (2) **CoNLL2003** is a well-known open-domain NER dataset. It consists of 1,393 English news articles, containing 10,059 `PER`, 10,645 `LOC`, 9,323 `ORG` and 5,062 `MISC` mentions.

We obtain the following distantly labeled datasets: (1) **BC5CDR (Big Dict)** is labeled using a dictionary[2] released by Shang et al. (2018); (2) **BC5CDR (Small Dict)** is labeled using a smaller dictionary constructed by selecting only the first 20% entries from the previous one; (3) **CoNLL2003 (KB)**[3] is labeled by the knowledge base Wikidata and released by Liang et al. (2020); (4) **CoNLL2003 (Dict)** is labeled using a refined dictionary released by Peng et al. (2019)[4]. For dictionary labeling, we use the strict string matching algorithm presented in Peng et al. (2019). The process of knowledge base labeling can be found in Liang et al. (2020).

All DS-NER methods are trained on the same distantly labeled training data and evaluated on the released human-annotated test sets in terms of span-level precision, recall and F1 score. To avoid the noise induced by the position tag in the distant labels, we do not consider the position of

[2] https://github.com/shangjingbo1226/AutoNER

[3] https://github.com/cliang1453/BOND

[4] https://github.com/v-mipeng/LexiconNER

each token in a named entity. During the prediction phase, a continuous span with the same label is considered as a single entity.

5.1.2 Baseline Methods

We compare the proposed Conf-MPU with different groups of baseline methods.

Fully Supervised Methods. We present the state-of-the-art (SOTA) performance of fully supervised methods on the two benchmark datasets, Wang et al. (2021) on BC5CDR and Wang et al. (2020) on CoNLL2003. For SOTA methods, we report the results from their original papers. We also evaluate the employed BiLSTM and BERT models in fully supervised setting. The performance in this group serves as upper-bound references.

Distantly Supervised Methods. We consider the following distantly supervised NER methods: (1) **Dict/KB Matching** distantly labels the test sets using dictionaries or knowledge bases directly, which is included here as references; (2) **AutoNER** (Shang et al., 2018) trains the model using a "tie-or-break" mechanism to detect entity boundaries and then predicts entity type for each candidate; (3) **BERT-ES** (Liang et al., 2020) adopts early stopping to prevent BERT from overfitting to noisy distant labels; (4) **BNPU** (Peng et al., 2019) built on LBiLSTM (**BNPU$_{\text{LBiLSTM}}$**) applies a binary PU learning risk estimation with MAE as the loss function to each entity type and then infers the final types; (5) **MPU** is the predecessor of the proposed Conf-MPU, which computes the empirical risk using Eq. 5. We also build MPU on both BERT and LBiLSTM models, denoted as **MPU$_{\text{BERT}}$** and **MPU$_{\text{LBiLSTM}}$**. Note that full models in Peng et al. (2019); Liang et al. (2020) contain self-training as post processing steps, which are omitted here. We focus on the evaluation of how well each model can handle incomplete labeling issues in DS-NER tasks.

5.1.3 Method Setups

To evaluate the efficacy of the DS-NER methods in real usage under distantly supervised settings, we do not use any human-annotated validation or test sets in any stage of the training process. The training stopping criteria are set as follows: 100 epochs for BiLSTM-based methods and 5 epochs for BERT-based ones. We report the performance of the final model instead of the best checkpoint. Consequently, the baselines have different performance from their reported results. We use the released code for AutoNER and BERT-ES to reproduce their results. For other methods, we report the results based on our implementations. BiLSTM-based models utilize pretrained `bio-embedding`[5] for BC5CDR and pretrained Stanford's `Glove`[6] embedding for CoNLL2003. BERT-based models use pretrained `biobert-base-cased-v1.1`[7] for BC5CDR and `bert-base-cased`[8] for CoNLL2003. The only exception is that BERT-ES uses `roberta-base`[9] for CoNLL2003 in the original implementation.

Dataset	Type	Precision	Recall
BC5CDR (Big Dict)	Chemical	97.99	63.14
	Disease	98.34	46.73
BC5CDR (Small Dict)	Chemical	98.66	11.43
	Disease	99.25	9.31
CoNLL2003 (KB)	PER	82.36	82.11
	LOC	99.98	65.20
	ORG	90.47	60.59
	MISC	100.00	20.07
CoNLL2003 (Dict)	PER	99.78	79.10
	LOC	97.56	34.69
	ORG	95.80	65.47
	MISC	99.24	57.22

Table 1: The quality of distant labels on training sets, stated in token-level precision and recall (in %).

5.2 Experimental Results

5.2.1 Main Results

We first examine the quality of the distantly labeled training data. Table 1 shows the detailed evaluation of distantly labeled training data. The results validate the assumption mentioned in previous work that distant labels generated by dictionaries are often of high precision but low recall.

Table 2 presents the overall span-level precision, recall, and F1 scores for all methods on the test sets. The proposed Conf-MPU shows a clear advantage over baseline methods, especially when accompanying with LBiLSTM. Almost all distantly supervised baselines perform better than Dict/KB Matching on these four datasets, except for a few cases of BNPU and MPU which will be discussed later. Among the baseline methods, AutoNER and BERT-ES show strong correlation with respect to the dictionary quality. On BC5CDR (Small Dict), where the dictionary suffers from extremely low coverage, the two methods have little improvement

[5] https://github.com/shangjingbo1226/AutoNER

[6] https://nlp.stanford.edu/projects/glove/

[7] https://huggingface.co/dmis-lab/biobert-base-cased-v1.1

[8] https://huggingface.co/bert-base-cased

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

Method	**BC5CDR (Big Dict)**	**BC5CDR (Small Dict)**	**CoNLL2003 (KB)**	**CoNLL2003 (Dict)**
Fully Supervised				
Existing SOTA	**90.99 (-/-)**		**94.60 (-/-)**	
BERT	83.88 (79.75/88.46)		89.03 (88.00/90.08)	
BiLSTM	75.60 (71.27/80.49)		86.19 (84.06/88.42)	
Distantly Supervised				
Dict/KB Matching	64.32 (86.39/51.24)	15.69 (80.02/8.70)	71.40 (81.13/63.75)	63.93 (93.12/48.67)
AutoNER	79.99 (82.63/77.52)	20.66 (81.47/11.83)	67.80 (73.10/63.22)	61.19 (82.87/48.50)
BERT-ES	73.66 (80.43/67.94)	17.21 (75.60/9.71)	72.15 (81.38/64.80)	63.68 (85.77/50.63)
BNPU$_{\text{LBiLSTM}}$	59.24 (48.12/77.06)	70.21 (64.93/76.43)	78.44 (74.38/82.97)	76.11 (73.68/78.70)
MPU$_{\text{BERT}}$	68.22 (56.50/86.05)	73.91 (70.08/78.18)	65.75 (58.79/74.58)	67.65 (63.63/72.22)
MPU$_{\text{LBiLSTM}}$	60.79 (48.28/82.06)	73.25 (67.50/80.07)	69.13 (59.46/82.54)	71.41 (63.41/81.71)
Conf-MPU$_{\text{BERT}}$	77.22 (69.79/86.42)	71.85 (81.02/64.54)	79.16 (78.58/79.75)	81.89 (81.71/82.08)
Conf-MPU$_{\text{LBiLSTM}}$	**80.07 (76.63/83.82)**	**76.18 (82.66/70.64)**	**80.02 (77.39/82.84)**	**83.34 (85.79/81.02)**

Table 2: The span-level results on test sets: F1 score (Precision/Recall) (in %), where the bests are in bold.

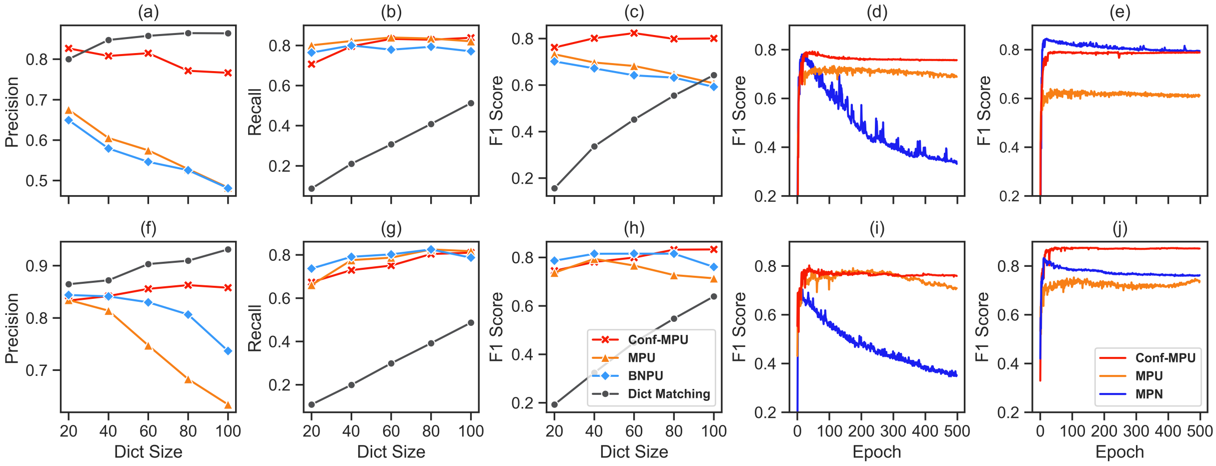

Figure 2: The performance of LBiLSTM-based methods under various settings. Figures in the first row (a - e) and the second row (f - j) show the results on BC5CDR and CoNLL2003, respectively.

on recall. Similarly, this phenomenon can also be observed by comparing their performance on CoNLL2003 (KB) and CoNLL2003 (Dict).

By contrast, all PU learning based methods demonstrate significantly higher recall on all datasets, showing more robustness to the issue of incomplete labeling. However, we can observe that compared with their performances on BC5CDR (Small Dict), BNPU and MPU suffer from low precision on BC5CDR (Big Dict) labeled by a dictionary with high coverage and precision, which is against one's intuition. We will extend the discussion in Section 5.2.3. Note that although BNPU and MPU are derived from the same probability principle, it is not necessary for BNPU and MPU to have similar performance. For example, BNPU$_{\text{LBiLSTM}}$ and MPU$_{\text{LBiLSTM}}$ perform similarly on BC5CDR datasets, but not on CoNLL2003 datasets. We suspect the cause is that they differ in the training process. BNPU is trained with the one-vs-all strategy where the distribution of unlabeled data is different for each entity type, while MPU is simultaneously trained with all types keeping the same distribution of unlabeled data. As mentioned earlier, the distribution of unlabeled data may significantly affect the risk estimation of PU learning. In addition, the inference step also has an unpredictable effect on the overall performance of BNPU.

As the results manifest, Conf-MPU can significantly improve precision compared with BNPU and MPU, and meanwhile maintain a high level of recall on all datasets, which shows that Conf-MPU can significantly alleviate the estimation bias.

We guide readers to the Appendix for the detailed evaluation of prior estimation and the empirical solution to the class imbalance problem.

5.2.2 Confidence Score Estimation

For Conf-MPU, another factor for its performance is the confidence score estimation for each token being an entity token. To evaluate the quality of the confidence scores, we first convert the results as labels where if $\hat{\lambda}(\boldsymbol{x}) > 0.5$ then label $\boldsymbol{x}$ as an

Method	**BC5CDR (Big Dict)**	**BC5CDR (Small Dict)**
Fully Supervised	85.00 (77.68/93.83)	
Binary PU	72.38 (61.89/87.17)	79.14 (81.85/76.60)
Method	**CoNLL2003 (KB)**	**CoNLL2003 (Dict)**
Fully Supervised	96.09 (92.77/99.65)	
Binary PU	88.88 (81.04/98.39)	88.08 (79.09/99.38)

Table 3: The results of confidence score estimation on test sets: F1 score (Precision/Recall) (in %).

Method	**LBiLSTM**	**BiLSTM**
BNPU	70.21 (64.93/76.43)	63.37 (57.92/69.97)
MPU	73.25 (67.50/80.07)	62.39 (56.50/69.66)
Conf-MPU	76.18 (82.66/70.64)	68.11 (71.68/64.88)

Table 4: Ablation study on lexicon features on BC5CDR (Small Dict).

entity token, otherwise label as a non-entity token. We present the results in terms of token-level F1 score, precision, and recall in Table 3, where Fully Supervised using human-annotated ground-truth labels provides upper-bound references for this estimation, while Binary PU uses distant labels. We can see that the classifier with the binary PU risk estimation achieves good recall on all of the distantly labeled datasets. High recall indicates that the classifier can recognize most of entity tokens, which can be taken advantage of in the Conf-MPU risk estimation to avoid overfitting to false negative samples in unlabeled data. We also evaluate the proposed Conf-MPU models with the confidence scores given by the fully supervised classifier on the four distantly labeled datasets, where the performances increased by 2 $\sim$ 5 percentage in terms of F1 score, indicating that the proposed Conf-MPU framework is robust to the confidence score estimation of lesser quality. We leave for future work the optimization of the confidence score estimation.

5.2.3 Impact of Dictionary Coverage

To have a solid recognition to the estimation bias in BNPU and MPU caused by the violation of PU assumption, we construct a series of dictionaries with different coverage on entities. We treat the dictionaries used to generate labels for BC5CDR (Big Dict) and CoNLL2003 (Dict) as two reference standard dictionaries. Then for each of the two benchmark datasets we build a group of dictionaries by selecting the first 20, 40, 60, 80, and 100 (%) entries from the standard ones. We train BNPU, MPU, and Conf-MPU on the distantly labeled datasets generated by these dictionaries. Here we show the results based on LBiLSTM in Figure 2 (a - c and f - h on corresponding test sets). Similar trend can be observed on BERT-based settings.

We can see a clear decreasing trend on precision for BNPU and MPU when dictionary size increases (Figures 2 (a&f)). These phenomena are caused due to the violation of PU assumption. When a dictionary has higher coverage, the distribution of unlabeled data is more and more similar to the distribution of true negative data, instead of to the overall data distribution. The BNPU and MPU risk estimations bring higher bias, leading to lower precision. Although their recalls remain high, the F1 scores still decrease. By contrast, the proposed Conf-MPU can effectively avoid this limitation and achieve good performance for all dictionary sizes.

5.2.4 Ablation Studies

To further evaluate the Conf-MPU risk estimation, we first conduct ablation studies comparing with MPN risk (Eq. 3) whose estimation simply treats unlabeled data as negative samples in distant supervision, and demonstrate the performance with different epochs. Sub-figures (d, e, i, j) in Figure 2 show the trends of F1 scores of LBiLSTM-based models on the validation sets using three risk estimations of Conf-MPU, MPU, and MPN, with respect to different number of epochs. (d, e) and (i, j) reflect the performance on BC5CDR and CoNLL2003, respectively. (d, i) and (e, j) reflect the performance based on 20% and 100% dictionaries, respectively. The results show that MPN risk estimation can lead to severe overfitting for the model when dictionaries have low coverage. Although MPN still causes overfitting on full dictionaries, its performances are more stable and generally good. By contrast, MPU and Conf-MPU consider the false negative issue during training, and do not overfit even on small dictionaries. Conf-MPU performs stably and consistently well for more scenarios comparing with the other risk estimations.

From Table 2, we can observe that Conf-MPU$_{\text{LBiLSTM}}$ outperforms BiLSTM with fully supervised setting on BC5CDR with both big and small dictionaries. To examine the performance gain, we implement three methods, BNPU, MPU, and Conf-MPU based on BiLSTM instead of LBiLSTM to evaluate the impact of lexicon features learned from the dictionaries. The results on BC5CDR (Small Dict) are shown in Table 4. We can see that the lexicon features used in DS-NER tasks can significantly improve the performance. The experiments performed on other distantly labeled datasets also exhibit similar trends. The results suggest that dictionaries in DS-NER tasks can also serve as external features in additional to the distant labels.

6 Related Work

DS-NER. Handling noisy labels (false positives and false negatives) in DS-NER has attracted extensive attention in recent years (Yang et al., 2018; Shang et al., 2018; Mayhew et al., 2019; Cao et al., 2019; Peng et al., 2019; Liang et al., 2020; Liu et al., 2021; Zhang et al., 2021). Here we briefly discuss a few representative approaches.

One line of work focuses on alleviating the impact of false negatives (or incomplete labeling). Our work belongs to this line. AutoNER (Shang et al., 2018) proposes a new tagging scheme to identify entity candidates by determining if the connection of two adjacent tokens should be `tied`, `broken`, or `unknown`, and then decides the type for entity candidates. To handle incomplete labeling, tokens with `unknown` tag are not counted for the loss calculation. Mayhew et al. (2019) introduce a constraint driven iterative algorithm learning to detect false negatives in the noisy data and down-weigh them, resulting in a weighted training set on which a weighted NER model is trained. Peng et al. (2019) employ PU learning to avoid the model overffiting to false negatives, and propose a bounded non-negative positive-unlabeled learning. However, the application of binary PU learning in DS-NER is limited to the underlying PU assumption and its efficiency. Our proposed Conf-MPU can release the limitations and allow PU learning to be utilized in wider distant supervision scenarios.

Another line of work considers the noise of both types, either explicitly or implicitly. Cao et al. (2019) design a data selection scheme to compute scores for annotation confidence and annotation coverage to distinguish high-quality sentences from noisy ones, and then propose a name tagging model that consists of two modules of sequence labeling and classification, focusing on high-quality and noisy portions, respectively. BOND (Liang et al., 2020), leveraging the power of pre-trained language model BERT, first adopts early stopping to prevent overfitting to noisy labels and obtains an initialized model, then further boosts the performance by a teacher-student self-training framework. Liu et al. (2021) propose a calibrated confidence estimation approach for DS-NER and integrate it in an LSTM-CRF model under a self-training framework to reduce the impact of noise. Zhang et al. (2021) study the noise in DS-NER from a novel perspective of dictionary bias. Specifically, they first formulate DS-NER using a structural causal model, then identify the causes of both false positives and false negatives, and finally debias via backdoor adjustment and causal invariance regularizer. We leave for future work enabling Conf-MPU to handle false positives.

PU Learning. PU learning learns a classifier from positive and unlabeled data (Elkan and Noto, 2008; Du Plessis et al., 2014). In a broad sense, PU learning belongs to semi-supervised learning. However, there is a fundamental difference between them: semi-supervised learning requires labeled negative data, but PU learning does not. Recently, a few works significantly enriched PU learning theory. Kiryo et al. (2017) propose a non-negative risk estimator for PU learning, which enables the usage of deep neural networks for classification given limited labeled positive data. Xu et al. (2017) first come up with the concept of multi-positive and unlabeled learning with a margin maximization goal for the multi-class classification problem. However, the objective of margin maximization cannot be easily extended to apply on popular deep learning architectures. Hsieh et al. (2019) propose a novel classification framework incorporating biased negative data in PU learning, which opens up a wider range of the applications of PU learning.

7 Conclusion

In this paper, we present a novel multi-class positive and unlabeled learning method called Conf-MPU for the DS-NER task. Conf-MPU estimates the empirical classification risks using the confidence estimation of a token being an entity token on the distantly labeled training data to prevent the model from overfitting to the false negatives. We empirically show that Conf-MPU can significantly reduce the potential risk estimation bias caused by PU assumption. The extensive experiments illustrate that compared with existing DS-NER methods, Conf-MPU is more robust to various types of dictionaries and can handle the incomplete labeling problem effectively.

Acknowledgments

The work is supported in part by NIFA grant no. 2022-67015-36217 from the USDA National Institute of Food and Agriculture.

References

Jessa Bekker and Jesse Davis. 2018. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760.

Christopher M Bishop. 2006. *Pattern recognition and machine learning*. Springer.

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

Jason PC Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, volume 27, pages 703–711.

Marthinus Christoffel Du Plessis and Masashi Sugiyama. 2014. Class Prior Estimation from Positive and Unlabeled Data. *IEICE TRANSACTIONS on Information and Systems*, 97(5):1358–1362.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 213–220.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688.

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

John Giorgi, Xindi Wang, Nicola Sahar, Won Young Shin, Gary D Bader, and Bo Wang. 2019. End-to-end named entity recognition and relation extraction using pre-trained language models. *arXiv preprint arXiv:1912.13415*.

Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. 2019. Classification from positive, unlabeled and biased negative data. In *International Conference on Machine Learning*, pages 2820–2829.

Shantanu Jain, Martha White, and Predrag Radivojac. 2016. Estimating the class prior and posterior from noisy positives and unlabeled data. In *Advances in neural information processing systems*, volume 29, pages 2693–2701.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. In *Advances in neural information processing systems*, pages 1675–1685.

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 *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

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.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651.

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning*. MIT Press.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86.

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.

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

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

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

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

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Improving named entity recognition by external context retrieving and cooperative learning. *arXiv preprint arXiv:2105.03654*.

Yixing Xu, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Multi-positive and unlabeled learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3182–3188.

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

Wenkai Zhang, Hongyu Lin, Xianpei Han, and Le Sun. 2021. De-biasing distantly supervised named entity recognition via causal intervention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4803–4813.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems*, 33.

Appendix

1 Proof of Theorem 1

Here we determine the difference between $\mathrm{R}(\hat{f}_{\mathrm{Conf-MPU}})$ and $\mathrm{R}(f^*)$ using the error bound between $\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f)$ and $\mathrm{R}(f)$. Let us first define the intermediate risk estimators as follows. Each one introduces a new estimation component.

Starting from $\mathrm{R}(f)$ (*i.e.*, Eq. 7), $\bar{\mathrm{R}}(f)$ introduces $\hat{\lambda}(\boldsymbol{x})$ as the estimation of $\lambda(\boldsymbol{x})$,

$$\begin{aligned}\bar{\mathrm{R}}(f) = \sum_{i=1}^{k} \pi_i \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=i)} \Big[&\ell(f(\boldsymbol{x}), i) \\ &+ \mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau} \ell(f(\boldsymbol{x}),0)\frac{1}{\hat{\lambda}(\boldsymbol{x})} - \ell(f(\boldsymbol{x}),0)\Big] \\ &+ \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})\le\tau}\ell(f(\boldsymbol{x}),0)\right].\end{aligned}$$

Then, $\hat{\mathrm{R}}(f)$ uses empirical means to estimate expectations,

$$\begin{aligned}\hat{\mathrm{R}}(f) = \sum_{i=1}^{k} \frac{\pi_i}{n_{\mathrm{P}_i}} \sum_{j=1}^{n_{\mathrm{P}_i}} \Big[&\ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), i) \\ &+ \mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})>\tau} \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}),0)\frac{1}{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})} - \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}),0)\Big] \\ &+ \frac{1}{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{U}}}\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{U}})\le\tau}\ell(f(\boldsymbol{x}_j^{\mathrm{U}}),0)\right].\end{aligned}$$

$\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f)$ restricts the loss on each labeled positive sample to be at least 0,

$$\begin{aligned}\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) = \sum_{i=1}^{k} \frac{\pi_i}{n_{\mathrm{P}_i}} \sum_{j=1}^{n_{\mathrm{P}_i}} \max\Big\{&0, \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), i) \\ - \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}),0) + \mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})>\tau} &\ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}),0)\frac{1}{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})}\Big\} \\ + \frac{1}{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{U}}}&\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{U}})\le\tau}\ell(f(\boldsymbol{x}_j^{\mathrm{U}}),0)\right].\end{aligned}$$

In the following proof, we will derive the error bounds from $\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f)$ to $\hat{\mathrm{R}}(f)$, from $\hat{\mathrm{R}}(f)$ to $\bar{\mathrm{R}}(f)$, and from $\bar{\mathrm{R}}(f)$ to $\mathrm{R}(f)$ in order.

Let us first derive the error bound from $\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f)$ to $\hat{\mathrm{R}}(f)$. For ease of notation, let

$$\begin{aligned}A &= \max\left\{0, \ell(f(\boldsymbol{x}), i) + \ell(f(\boldsymbol{x}),0)\left(\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\frac{1}{\hat{\lambda}(\boldsymbol{x})} - 1\right)\right\}, \\ B &= \ell(f(\boldsymbol{x}), i) + \ell(f(\boldsymbol{x}),0)\left(\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\frac{1}{\hat{\lambda}(\boldsymbol{x})} - 1\right).\end{aligned}$$

Then we have

$$\left|\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) - \hat{\mathrm{R}}(f)\right| = \sum_{i=1}^{k}\frac{\pi_i}{n_{\mathrm{P}_i}}\sum_{j=1}^{n_{\mathrm{P}_i}}|A-B|.$$

Since $\ell(\cdot) \in [0, C_l]$, we have $A \in \left[0, \frac{C_l}{\tau}\right]$ and $B \in \left[-C_l, \frac{C_l}{\tau}\right]$. Further, we have $|A-B| \le \left(1+\frac{1}{\tau}\right)C_l$. So we get the error bound

$$\left|\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) - \hat{\mathrm{R}}(f)\right| \le \sum_{i=1}^{k}\pi_i\frac{(\tau+1)C_l}{\tau}.$$

Then, we use the following lemma to establish the error bound from $\hat{\mathrm{R}}(f)$ to $\bar{\mathrm{R}}(f)$.

Lemma 1. *Let* $\hat{\lambda}(\cdot) : \mathbb{R}^d \to (0,1]$ *be a fixed function independent of data used to compute* $\hat{\mathrm{R}}(f)$ *and* $\tau \in (0,1]$. *For any* $\delta > 0$, *with probability at least* $1-\delta$,

$$\begin{aligned}&\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}(f) - \bar{\mathrm{R}}(f)\right| \le \\ &\sum_{i=1}^{k}\pi_i\left[\frac{2L_l}{\tau}\mathfrak{R}_{n_{\mathrm{P}_i},p(\boldsymbol{x}|y=i)}(\mathcal{F}) + \frac{(\tau+1)C_l}{\tau}\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{P}_i}}}\right] \\ &+ 2L_l\mathfrak{R}_{n_{\mathrm{U}},p(\boldsymbol{x})}(\mathcal{F}) + C_l\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{U}}}}.\end{aligned}$$

Proof of Lemma 1. For ease of notation, let

$$\begin{aligned}\bar{\mathrm{R}}_{\mathrm{P}_i}^{+} &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=i)}\Big[\ell(f(\boldsymbol{x}), i) \\ &\qquad + \ell(f(\boldsymbol{x}),0)\left(\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})}\frac{1}{\hat{\lambda}(\boldsymbol{x})} - 1\right)\Big], \\ \bar{\mathrm{R}}_{\mathrm{U}}^{-} &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})\le\tau}\ell(f(\boldsymbol{x},0))\right], \\ \hat{\mathrm{R}}_{\mathrm{P}_i}^{+} &= \frac{1}{n_{\mathrm{P}_i}}\sum_{j=1}^{n_{\mathrm{P}_i}}\Big[\ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}), i) \\ &\qquad + \ell(f(\boldsymbol{x}_j^{\mathrm{P}_i}),0)\left(\mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})>\tau}\frac{1}{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{P}_i})} - 1\right)\Big], \\ \hat{\mathrm{R}}_{\mathrm{U}}^{-} &= \frac{1}{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{U}}}\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x}_j^{\mathrm{U}})\le\tau}\ell(f(\boldsymbol{x}_j^{\mathrm{U}}),0)\right].\end{aligned}$$

From the sub-additivity of the supremum operator, we have

$$\begin{aligned}\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}(f) - \bar{\mathrm{R}}(f)\right| \le \sum_{i=1}^{k}\pi_i &\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}_{\mathrm{P}_i}^{+} - \bar{\mathrm{R}}_{\mathrm{P}_i}^{+}\right| \\ &+ \sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}_{\mathrm{U}}^{-} - \bar{\mathrm{R}}_{\mathrm{U}}^{-}\right|.\end{aligned}$$

It suffices to prove Lemma 1 if we can prove that with probability at least $1-\frac{\delta}{k+1}$, the following bounds hold separately:

$$\begin{aligned}\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}_{\mathrm{P}_i}^{+} - \bar{\mathrm{R}}_{\mathrm{P}_i}^{+}\right| &\le \frac{2L_l}{\tau}\mathfrak{R}_{n_{\mathrm{P}_i},p(\boldsymbol{x}|y=i)}(\mathcal{F}) \\ &\quad + \frac{(\tau+1)C_l}{\tau}\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{P}_i}}}, \qquad (9)\end{aligned}$$

$$\begin{aligned}\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}_{\mathrm{U}}^{-} - \bar{\mathrm{R}}_{\mathrm{U}}^{-}\right| &\le 2L_l\mathfrak{R}_{n_{\mathrm{U}},p(\boldsymbol{x})}(\mathcal{F}) \\ &\quad + C_l\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{U}}}}. \qquad (10)\end{aligned}$$

Next, we prove Inequation 9. Inequation 10 is proven similarly.

Let $\phi_{\boldsymbol{x}} : \mathbb{R} \to \mathbb{R}$ be the function defined by $\phi_{\boldsymbol{x}} : z \mapsto \ell(z,i) + \ell(z,0)\left(\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\frac{1}{\hat{\lambda}(\boldsymbol{x})} - 1\right)$. For $\boldsymbol{x} \in \mathbb{R}^d$, $f \in \mathcal{F}$, since $\ell(\cdot) \in [0, C_l]$ and $\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\frac{1}{\hat{\lambda}(\boldsymbol{x})} - 1 \in \left[-1, \frac{1}{\tau} - 1\right]$, we have $\phi_{\boldsymbol{x}}(f(x)) \in \left[-C_l, \frac{C_l}{\tau}\right]$. Following the proof of Theorem 3.3 in (Mohri et al., 2018), we can show that with probability at least $1 - \frac{\delta}{k+1}$, it holds that

$$\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}^+_{\mathrm{P}_i} - \bar{\mathrm{R}}^+_{\mathrm{P}_i}\right| \leq$$
$$2\mathbb{E}_{\mathcal{X}_{\mathrm{P}_i}\sim p(\boldsymbol{x}|y=i)^{n_{\mathrm{P}_i}}}\mathbb{E}_{\boldsymbol{\theta}}\left[\sup_{f\in\mathcal{F}}\frac{1}{n_{\mathrm{P}_i}}\sum_{j}^{n_{\mathrm{P}_i}}\theta_j\phi_{\boldsymbol{x}_j^{\mathrm{P}_i}}(f(\boldsymbol{x}_j^{\mathrm{P}_i}))\right]$$
$$+\frac{(\tau+1)C_l}{\tau}\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{P}_i}}},$$

where $\boldsymbol{\theta} = \{\theta_1, ..., \theta_{n_{\mathrm{P}_i}}\}$ and each θ_i is a Rademacher variable.

We notice that for all $\boldsymbol{x}$, $\phi_{\boldsymbol{x}}$ is a (L_l/τ)-Lipschitz function on the interval $[-C_g, C_g]$. Following the proof of Lemma 26.9 in (Shalev-Shwartz and Ben-David, 2014), we can show that, when the set $\mathcal{X}_{\mathrm{P}_i}$ is fixed, we have

$$\mathbb{E}_{\mathcal{X}_{\mathrm{P}_i}\sim p(\boldsymbol{x}|y=i)^{n_{\mathrm{P}_i}}}\mathbb{E}_{\boldsymbol{\theta}}\left[\sup_{f\in\mathcal{F}}\frac{1}{n_{\mathrm{P}_i}}\sum_{j}^{n_{\mathrm{P}_i}}\theta_j\phi_{\boldsymbol{x}_j^{\mathrm{P}_i}}(f(\boldsymbol{x}_j^{\mathrm{P}_i}))\right] \leq$$
$$\frac{L_l}{\tau}\mathbb{E}_{\mathcal{X}_{\mathrm{P}_i}\sim p(\boldsymbol{x}|y=i)^{n_{\mathrm{P}_i}}}\mathbb{E}_{\boldsymbol{\theta}}\left[\sup_{f\in\mathcal{F}}\frac{1}{n_{\mathrm{P}_i}}\sum_{j}^{n_{\mathrm{P}_i}}\theta_j f(\boldsymbol{x}_j^{\mathrm{P}_i})\right].$$

We can obtain Inequation 9 by substituting the Rademacher complexity. Lemma 1 is proven.

Then, we establish the error bound from $\bar{\mathrm{R}}(f)$ to $\mathrm{R}(f)$ by the following lemma.

Lemma 2. *Let $\hat{\lambda}(\cdot) : \mathbb{R}^d \to (0,1]$, $\tau \in (0,1]$, $\zeta = p(\hat{\lambda}(\cdot) \leq \tau)$ and $\epsilon = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})|^2\right]$. For all $f \in \mathcal{F}$, it holds that*

$$\left|\bar{\mathrm{R}}(f) - \mathrm{R}(f)\right| \leq \frac{C_l}{\tau}\sqrt{(1-\zeta)\epsilon}.$$

Proof of Lemma 2. We notice that the difference between $\bar{\mathrm{R}}(f)$ and $\mathrm{R}(f)$ is actually the difference between $\bar{\mathrm{R}}^-_{\mathrm{U}}$ and $\mathrm{R}^-_{\mathrm{U}}$, where

$$\begin{aligned}\bar{\mathrm{R}}^-_{\mathrm{U}} &= \sum_{i=1}^{k}\pi_i\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=i)}\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\frac{1}{\hat{\lambda}(\boldsymbol{x})}\right]\\&\quad+\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})\leq\tau}\ell(f(\boldsymbol{x}),0)\right],\\\mathrm{R}^-_{\mathrm{U}} &= \sum_{i=1}^{k}\pi_i\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=i)}\left[\mathbb{1}_{\lambda(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\frac{1}{\lambda(\boldsymbol{x})}\right]\\&\quad+\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathbb{1}_{\lambda(\boldsymbol{x})\leq\tau}\ell(f(\boldsymbol{x}),0)\right].\end{aligned}$$

We can rewrite $\bar{\mathrm{R}}^-_{\mathrm{U}}$ and $\mathrm{R}^-_{\mathrm{U}}$ in the form of integral

$$\begin{aligned}\bar{\mathrm{R}}^-_{\mathrm{U}} &= \sum_{i=1}^{k}\pi_i\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\frac{1}{\hat{\lambda}(\boldsymbol{x})}p(\boldsymbol{x}\mid y=i)d\boldsymbol{x}\\&\quad+\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})\leq\tau}\ell(f(\boldsymbol{x}),0)p(\boldsymbol{x})d\boldsymbol{x},\\\mathrm{R}^-_{\mathrm{U}} &= \sum_{i=1}^{k}\pi_i\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\frac{1}{\lambda(\boldsymbol{x})}p(\boldsymbol{x}\mid y=i)d\boldsymbol{x}\\&\quad+\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})\leq\tau}\ell(f(\boldsymbol{x}),0)p(\boldsymbol{x})d\boldsymbol{x},\end{aligned}$$

where for $\mathrm{R}^-_{\mathrm{U}}$, we replace the subscript $\lambda(\boldsymbol{x})$ of the indicator function with $\hat{\lambda}(\boldsymbol{x})$, which does not change the value of $\mathrm{R}^-_{\mathrm{U}}$.

According to the sub-additivity of the supremum operator, we have

$$\begin{aligned}&\left|\bar{\mathrm{R}}(f) - \mathrm{R}(f)\right| = \left|\bar{\mathrm{R}}^-_{\mathrm{U}} - \mathrm{R}^-_{\mathrm{U}}\right| \leq\\&\sum_{i=1}^{k}\pi_i\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\left|\frac{1}{\hat{\lambda}(\boldsymbol{x})} - \frac{1}{\lambda(\boldsymbol{x})}\right|p(\boldsymbol{x}\mid y=i)d\boldsymbol{x}\\&=\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\left|\frac{1}{\hat{\lambda}(\boldsymbol{x})} - \frac{1}{\lambda(\boldsymbol{x})}\right|p(\boldsymbol{x},y>0)d\boldsymbol{x}\\&=\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\ell(f(\boldsymbol{x}),0)\frac{\left|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})\right|}{\hat{\lambda}(\boldsymbol{x})\lambda(\boldsymbol{x})}p(\boldsymbol{x},y>0)d\boldsymbol{x}\\&\leq\frac{C_l}{\tau}\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\frac{\left|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})\right|}{\lambda(\boldsymbol{x})}p(\boldsymbol{x},y>0)d\boldsymbol{x}\\&=\frac{C_l}{\tau}\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\frac{\left|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})\right|}{p(y>0\mid\boldsymbol{x})}p(y>0\mid\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\&=\frac{C_l}{\tau}\int\mathbb{1}_{\hat{\lambda}(\boldsymbol{x})>\tau}\left|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})\right|p(\boldsymbol{x})d\boldsymbol{x}\\&\leq\frac{C_l}{\tau}\sqrt{\int\mathbb{1}^2_{\hat{\lambda}(\boldsymbol{x})>\tau}p(\boldsymbol{x})d\boldsymbol{x}}\sqrt{\int\left|\hat{\lambda}(\boldsymbol{x}) - \lambda(\boldsymbol{x})\right|^2p(\boldsymbol{x})d\boldsymbol{x}}\\&=\frac{C_l}{\tau}\sqrt{(1-\zeta)\epsilon},\end{aligned}$$

where the last inequality is obtained after applying the Cauchy-Schwarz inequality. Lemma 2 is proven.

Combining the above three error bounds, we know that with probability at least $1-\delta$, the following holds:

$$\begin{aligned}&\sup_{f\in\mathcal{F}}\left|\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) - \mathrm{R}(f)\right| \leq \sum_{i=1}^{k}\pi_i\frac{(\tau+1)C_l}{\tau}\\&+\sum_{i=1}^{k}\pi_i\left[\frac{2L_l}{\tau}\mathfrak{R}_{n_{\mathrm{P}_i},p(\boldsymbol{x}|y=i)}(\mathcal{F}) + \frac{(\tau+1)C_l}{\tau}\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{P}_i}}}\right]\\&+2L_l\mathfrak{R}_{n_{\mathrm{U}},p(\boldsymbol{x})}(\mathcal{F}) + C_l\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{U}}}} + \frac{C_l}{\tau}\sqrt{(1-\zeta)\epsilon}.\end{aligned}$$

Finally, with probability at least $1-\delta$,

$$\begin{aligned}&\mathrm{R}(\hat{f}_{\mathrm{Conf-MPU}}) - \mathrm{R}(f^*)\\&=\mathrm{R}(\hat{f}_{\mathrm{Conf-MPU}}) - \hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(\hat{f}_{\mathrm{Conf-MPU}})\\&\quad+\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(\hat{f}_{\mathrm{Conf-MPU}}) - \hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f^*)\\&\quad+\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f^*) - \mathrm{R}(f^*)\\&\leq\mathrm{R}(\hat{f}_{\mathrm{Conf-MPU}}) - \hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(\hat{f}_{\mathrm{Conf-MPU}})\end{aligned}$$

$$
\begin{aligned}
&+ \hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f^*) - \mathrm{R}(f^*) \\
\leq &\left|\mathrm{R}(\hat{f}_{\mathrm{Conf-MPU}}) - \hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(\hat{f}_{\mathrm{Conf-MPU}})\right| \\
&+ \left|\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f^*) - \mathrm{R}(f^*)\right| \\
\leq &\, 2 \sup_{f \in \mathcal{F}} \left|\hat{\mathrm{R}}_{\mathrm{Conf-MPU}}(f) - \mathrm{R}(f)\right| \\
\leq &\sum_{i=1}^{k} 2\pi_i \frac{(\tau+1)C_l}{\tau} + \sum_{i=1}^{k} 2\pi_i \Bigg[\frac{2L_l}{\tau}\mathfrak{R}_{n_{\mathrm{P}_i}, p(\boldsymbol{x}|y=i)}(\mathcal{F}) \\
&+ \frac{(\tau+1)C_l}{\tau}\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{P}_i}}}\Bigg] + 4L_l\mathfrak{R}_{n_{\mathrm{U}}, p(\boldsymbol{x})}(\mathcal{F}) \\
&+ 2C_l\sqrt{\frac{\log\frac{k+1}{\delta}}{2n_{\mathrm{U}}}} + \frac{2C_l}{\tau}\sqrt{(1-\zeta)\epsilon}.
\end{aligned}
$$

Theorem 1 is proven.

2 Evaluation of Prior Estimation

As mentioned in Section 4.3, we apply the TIcE algorithm to estimate the prior for each class without using ground-truth annotations. In Table 5, we compare the priors estimated on distantly annotated BC5CDR (Big Dict) and CoNLL2003 (Dict) with the true priors for each class of the two datasets. We also conduct the prior estimation on BC5CDR (Small Dict) and CoNLL2003 (KB), and we find that the priors estimated with different dictionaries are not significantly different. Table 5 shows that the estimated priors by TIcE algorithm are close to the true priors. We further trained Conf-MPU models using the true priors and do not observe significant differences in the performances. This experiment indicates that Conf-MPU is not sensitive to the prior estimations and TIcE algorithm can be applied for prior estimation without ground-truth labels.

Dataset	Type	Estimated Prior	True Prior
BC5CDR	Chemical	0.0503	0.0601
(Big Dict)	Disease	0.0504	0.0601
	PER	0.1052	0.0547
CoNLL2003	LOC	0.0331	0.0407
(Dict)	ORG	0.0630	0.0492
	MISC	0.0371	0.0226

Table 5: The results of prior estimation.

3 Study on Class Imbalance Problem

NER tasks often suffer from the problem of class imbalance, where most tokens are not entity tokens. In our experiments, we introduce a hyperparameter γ as a class weight in risk estimations to balance the risks on positive and unlabeled data.

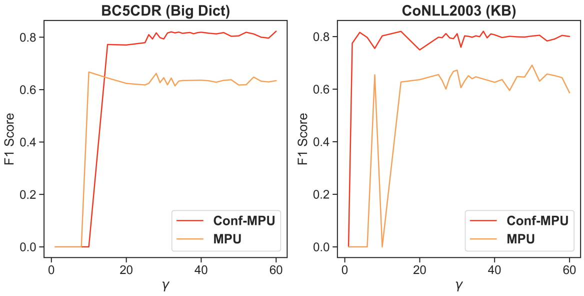

Figure 3: Empirical study on the class weight.

We empirically investigate the effect of this parameter by evaluating Conf-MPU and MPU with different values of γ on BC5CDR (Big Dict) and CoNLL2003 (KB), and show the span-level F1 scores on test sets in Figure 3. We can see that if the class imbalance issue is ignored (*i.e.*, $\gamma = 1$), the two methods achieve very low F1 scores on the two distantly labeled datasets. When γ increases to a certain value (*e.g.*, $\gamma = 15$), both methods can achieve good performances. As γ increases, the F1 scores fluctuate a little within a certain range but stay high. It indicates that the two methods are not sensitive to the value of the class weight γ if it is properly large. Similar results can also be observed on the other distantly labeled datasets. Therefore, for a fair comparison, we uniformly set γ to 28 and 15 for distantly labeled BC5CDR datasets and CoNLL2003 datasets, respectively, in our experiments.