Distantly Supervised Named Entity Recognition using Positive-Unlabeled Learning

Minlong Peng*, Xiaoyu Xing*, Qi Zhang, Jinlan Fu, Xuanjing Huang
School of Computer Science, Fudan University, Shanghai, China
{mlpeng16,xyxing18,qz,fujl16,xjhuang}@fudan.edu.cn

Abstract

In this work, we explore the way to perform named entity recognition (NER) using only unlabeled data and named entity dictionaries. To this end, we formulate the task as a positive-unlabeled (PU) learning problem and accordingly propose a novel PU learning algorithm to perform the task. We prove that the proposed algorithm can unbiasedly and consistently estimate the task loss as if there is fully labeled data. A key feature of the proposed method is that it does not require the dictionaries to label every entity within a sentence, and it even does not require the dictionaries to label all of the words constituting an entity. This greatly reduces the requirement on the quality of the dictionaries and makes our method generalize well with quite simple dictionaries. Empirical studies on four public NER datasets demonstrate the effectiveness of our proposed method. We have published the source code at https://github.com/v-mipeng/LexiconNER.

1 Introduction

Named Entity Recognition (NER) is concerned with identifying named entities, such as person, location, product and organization names in unstructured text. It is a fundamental component in many natural language processing tasks such as machine translation (Babych and Hartley, 2003), knowledge base construction (Riedel et al., 2013; Shen et al., 2012), automatic question answering (Bordes et al., 2015), search (Zhu et al., 2005), etc. In this field, supervised methods, ranging from the typical graph models (Zhou and Su, 2002; McCallum et al., 2000; McCallum and Li, 2003; Settles, 2004) to current popular neural-network-based models (Chiu and Nichols, 2016; Lample et al., 2016; Gridach, 2017; Liu et al., 2018; Zhang and Yang, 2018), have achieved great success. However, these supervised methods often require large scale fine-grained annotations (label every word of a sentence) to generalize well. This makes it hard to apply them to label-few domains, e.g., bio/medical domains (Delėger et al., 2016).

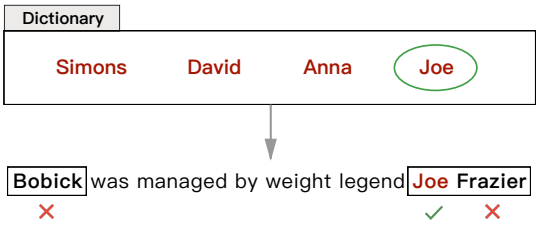


Figure 1: Data labeling example for person names using our constructed dictionary.

In this work, we explore the way to perform NER using only unlabeled data and named entity dictionaries, which are relatively easier to obtain compared with labeled data. A natural practice to perform the task is to scan through the query text using the dictionary and treat terms matched with a list of entries of the dictionary as the entities (Nadeau et al., 2006; Gerner et al., 2010; Liu et al., 2015; Yang et al., 2018). However, this practice requires very high quality named entity dictionaries that cover most of entities, otherwise it will fail with poor performance. As shown in Figure 1, the constructed dictionary of person names only labels one entity within the query text, which contains two entities "Bobick" and "Joe Frazier", and it only labels one word "Joe" out of the two-word entity "Joe Frazier".

To address this problem, an intuitive solution is to further perform supervised or semi-supervised learning using the dictionary labeled data. However, since it does not guarantee that the dictionary covers all entity words (words being of entities) within a sentence, we cannot simply treat a word

*Equal contribution.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2409–2419
Florence, Italy, July 28 - August 2, 2019.

not labeled by the dictionary as the non-entity word. Take the data labeling results depicted in Figure 1 as an example. Simply treating "Bobick" and "Frazier" as non-entity words and then performing supervised learning will introduce label noise to the supervised classifier. Therefore, when using the dictionary to perform data labeling, we can actually only obtain some entity words and a bunch of unlabeled data comprising of both entity and non-entity words. In this case, the conventional supervised or semi-supervised learning algorithms are not suitable, since they usually require labeled data of all classes.

With this consideration, we propose to formulate the task as a positive-unlabeled (PU) learning problem and accordingly introduce a novel PU learning algorithm to perform the task. In our proposed method, the labeled entity words form the positive (P) data and the rest form the unlabeled (U) data for PU learning. We proved that the proposed algorithm can unbiasedly and consistently estimate the task loss as if there is fully labeled data, under the assumption that the labeled P data can reveal the data distribution of class P. Of course, since words labeled by the dictionary only cover part of entities, it cannot fully reveal data distribution of entity words. To deal with this problem, we propose an adapted method, motivated by the AdaSampling algorithm (Yang et al., 2017), to enrich the dictionary. We evaluate the effectiveness of our proposed method on four NER datasets. Experimental results show that it can even achieve comparable performance with several supervised methods, using quite simple dictionaries.

Contributions of this work can be summarized as follows: **1)** We proposed a novel PU learning algorithm to perform the NER task using only unlabeled data and named entity dictionaries. **2)** We proved that the proposed algorithm can unbiasedly and consistently estimate the task loss as if there is fully labeled data, under the assumption that the entities found out by the dictionary can reveal the distribution of entities. **3)** To make the above assumption hold as far as possible, we propose an adapted method, motivated by the *AdaSampling* algorithm, to enrich the dictionary. **4)** We empirically prove the effectiveness of our proposed method with extensive experimental studies on four NER datasets.

2 Preliminaries

2.1 Risk Minimization

Let $\mathbf{X} \in \mathcal{X}$ and $\mathrm{Y} \in \mathcal{Y}$ be the input and output random variables, where $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1\}$ denote the space of $\mathbf{X}$ and Y, respectively. Let $f : \mathcal{X} \to \mathbb{R}$ denote a classifier. A loss function is a map $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}^+$. Given any loss function ℓ and a classifier f, we define the ℓ-risk of f by:

$$R_\ell(f) = \mathbb{E}_{\mathbf{X},\mathrm{Y}}\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}) \tag{1}$$

where $\mathbb{E}$ denotes the expectation and its subscript indicates the random variables with respect to which the expectation is taken. In ordinary supervised learning, we estimate R_ℓ with the empirical loss $\hat{R}_\ell$:

$$\hat{R}_\ell = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i), \tag{2}$$

and update model parameters to learn a classifier f^* that minimizes $\hat{R}_\ell$:

$$f^* = \arg\min_f \hat{R}_\ell(f). \tag{3}$$

2.2 Unbiased Positive-Unlabeled learning

Unbiased positive-unlabeled learning (uPU) (du Plessis et al., 2014) aims to estimate R_ℓ when there are only a set of positive (P) examples and a set of unlabeled (U) examples, which contains both positive and negative examples. R_ℓ can also be formulated by:

$$R_\ell = \pi_n\mathbb{E}_{\mathbf{X},\mathrm{Y}=0}\ell(f(\boldsymbol{x}), 0) + \pi_p\mathbb{E}_{\mathbf{X},\mathrm{Y}=1}\ell(f(\boldsymbol{x}), 1), \tag{4}$$

where $\pi_p = \mathrm{P}(\mathrm{Y} = 1)$ and $\pi_n = \mathrm{P}(\mathrm{Y} = 0)$. Note that $\mathbb{E}_{\mathbf{X},\mathrm{Y}=1}\ell(f(\boldsymbol{x}), 1)$ can be effectively estimated using positive data. Therefore, the main problem of PU learning is how to estimate $\mathbb{E}_{\mathbf{X},\mathrm{Y}=0}\ell(f(\boldsymbol{x}), 0)$ without using negative labeled data. To this end, it further formulates:

$$\begin{aligned}\pi_n\mathbb{E}_{\mathbf{X},\mathrm{Y}=0}\ell(f(\boldsymbol{x}), 0) = \mathbb{E}_{\mathbf{X}}\ell(f(\boldsymbol{x}), 0) \\ - \pi_p\mathbb{E}_{\mathbf{X},\mathrm{Y}=1}\ell(f(\boldsymbol{x}), 0).\end{aligned}$$

This equation holds because:

$$\mathrm{P}(\mathrm{Y} = 0)\mathrm{P}(\mathbf{X}|\mathrm{Y} = 0) = \mathrm{P}(\mathbf{X}) - \mathrm{P}(\mathrm{Y} = 1)\mathrm{P}(\mathbf{X}|\mathrm{Y} = 1).$$

According to this equation, we can now estimate $\pi_n\mathbb{E}_{\mathbf{X},\mathrm{Y}=0}\ell(f(\boldsymbol{x}), 0)$ using only unlabeled data and positive data. Thus, R_ℓ can be effectively

estimated using only unlabeled data and positive data. In summary, we have that R_ℓ can be unbiasedly estimated by:

$$\hat{R}_\ell = \frac{1}{n_u}\sum_{i=1}^{n_u}\ell(f(\boldsymbol{x}_i^u),0)+ \frac{\pi_p}{n_p}\sum_{i=1}^{n_p}\left(\ell(f(\boldsymbol{x}_i^p),1)-\ell(f(\boldsymbol{x}_i^p),0)\right), \tag{5}$$

where $\boldsymbol{x}_i^u$ and $\boldsymbol{x}_i^p$ denotes an unlabeled and positive example, respectively, and n_u and n_p denotes the number of unlabeled and positive examples, respectively.

2.3 Consistent Positive-Unlabeled Learning

As we know, a good estimation should be not only unbiased but also consistent. The above induction has proved that $\hat{R}_\ell$ is an unbiased estimation of R_ℓ. In this section, we show that $\hat{R}_\ell$ can be also a consistent estimation of R_ℓ when the loss function ℓ is upper bounded. *We argue that this is the first work to give such a proof*, which is summarized in the following theorem:

Theorem 1. If ℓ is bounded by $[0, M]$, then for any $\epsilon > 0$,

$$\mathrm{P}\{\mathrm{S}\in\mathcal{D}|\sup_{\mathrm{f}\in\mathcal{H}_\mathrm{R}}|\mathrm{R}_\ell-\hat{\mathrm{R}}_\ell|\le\epsilon\} \ge 1-2N(\frac{\epsilon}{4(1+2\pi_p)L_M})e^{-\frac{\min(n_p,n_u)\epsilon^2}{8(1+2\pi_p)^2B^2}}, \tag{6}$$

where $B = L_M M + C_0$. Here, L_M denotes the Lipschitz constant that $L_M > \frac{\partial\ell(w,y)}{\partial w}, \forall w \in \mathbb{R}$, $C_0 = \max_y \ell(0, y)$, and $\mathcal{H}$ denotes a Reproducing Kernel Hilbert Space (RKHS) (Aronszajn, 1950). $\mathcal{H}_R$ is the hypothesis space for each given $R > 0$ in the ball of radius R in $\mathcal{H}$. $N(\epsilon)$ denotes the covering number of $\mathcal{H}_R$ following Theorem C in (Cucker and Smale, 2002).

Proof. Proof appears in Appendix **A**. □

Remark 1. *Let us intuitively think about what if ℓ is not upper bounded (e.g., the cross entropy loss function). Suppose that there is a positive example $\boldsymbol{x}_i^p$ not occurring in the unlabeled data set. Then, its corresponding risk defined in $\hat{R}_\ell$ is $V(\boldsymbol{x}_i^p) = \frac{\pi_p}{n_p}(\ell(f(\boldsymbol{x}_i^p),1)-\ell(f(\boldsymbol{x}_i^p),0))$. If ℓ is not upper bounded, to achieve a small value of $V(\boldsymbol{x}_i^p)$, f can heavily overfit $\boldsymbol{x}_i^p$ making $\ell(f(\boldsymbol{x}_i^p),0) \to +\infty$, and in turn $V(\boldsymbol{x}_i^p) \to -\infty$. From this analysis, we can expect that, when using a unbounded loss function and a flexible classifier, $\hat{R}_\ell$ will dramatically decrease to a far below zero value.*

Therefore, in this work, *we force ℓ to be bounded by replacing the common unbounded cross entropy loss function with the mean absolute error, resulting in a bounded unbiased positive-unlabeled learning (buPU) algorithm*. This slightly differs from the setting of uPU, which only requires ℓ to be symmetric.

We further combine buPU with the non-negative constraint proposed by Kiryo et al. (2017), which has proved to be effectiveness in alleviating overfitting, obtaining a bounded non-negative positive-unlabeled learning (bnPU) algorithm:

$$\hat{R}_\ell = \frac{\pi_p}{n_p}\sum_{i=1}^{n_p}\ell(f(\boldsymbol{x}_i^p),1)+ \max\left(0,\frac{1}{n_u}\sum_{i=1}^{n_u}\ell(f(\boldsymbol{x}_i^u),0)-\frac{\pi_p}{n_p}\sum_{i=1}^{n_p}\ell(f(\boldsymbol{x}_i^p),0)\right). \tag{7}$$

3 Dictionary-based NER with PU Learning

In the following, we first define some notations used throughout this work, and illustrate the label assignment mechanism used in our method. Then, we precisely illustrate the data labeling process using the dictionary. After that, we show the detail for building the PU classifier, including word representation, loss definition, and label inference. Finally, we show the adapted method for enriching the dictionary.

3.1 Notations

We denote $\mathrm{W} \in \mathcal{V}$ and $\mathrm{S} = \{\mathrm{W}\} \in \mathcal{S}$ be the word-level and sentence-level input random variables, where $\mathcal{V}$ is the word vocabulary and $\mathcal{S}$ is the sentence space. $\mathrm{D_e}$ denotes the entity dictionary for a given entity type and $\mathcal{D} = \{s_1, \cdots, s_N\} \subseteq \mathcal{S}$ denotes the unlabelled dataset. We denote $\mathcal{D}^+$ the set of entity words labeled by $\mathrm{D_e}$, and denote $\mathcal{D}^u$ the rest unlabeled words.

3.2 Label Assignment Mechanism

In this work, we apply the binary label assignment mechanism for the NER task instead of the prevalent BIO or BIOES mechanism. Entity words are mapped to the positive class and non-entity words are mapped to the negative class. This is because, as we have discussed in the §1, the dictionary cannot guarantee to cover all entity words within a sentence. It may only label the beginning (B), the internal (I), or the last (E) word

Algorithm 1 Data Labeing using the Dictionary

Input: named entity dictionary $\mathrm{D_e}$, a sentence $s = \{w_1, \cdots, w_n\}$, and the context size k
Result: partial labeled sentence
Initialize: $i \leftarrow 1$
while $i \leq n$ **do**
  for $j \in [k, \cdots, 0]$ **do**
    if $\{w_i, \cdots, w_{\max(i+j,n)}\} \in \mathrm{D_e}$ **then**
      label $\{w_i, \cdots, w_{\max(i+j,n)}\}$ as positive class.
      $i \leftarrow i + j + 1$
      break
    if $j == 0$ **then**
      $i \leftarrow i + 1$

of an entity. Therefore, we cannot distinguish which type, B, I, or E, the labeled entity word belongs to. Take the data labeling results depicted in Figure 1 as an example. With the dictionary, we know that "Joe" is an entity word. However we cannot know that it is the beginning of the person name "Joe Frazier".

3.3 Data Labeling using the Dictionary

To obtain $\mathcal{D}^+$, we use the maximum matching algorithm (Liu et al., 1994; Xue, 2003) to perform data labeling with $\mathrm{D_e}$. It is a greedy search routine that walks through a sentence trying to find the longest string, starting from a given point in the sentence, that matches with an entry in the dictionary. The general process of this algorithm is summarized in Alg. 1. In our experiments, we intuitively set the context size $k = 4$.

3.4 Build PU Learning Classifier

In this work, we use a neural-network-based architecture to implement the classifier f, and this architecture is shared by different entity types.

Word Representation. Context-independent word representation consists of three part of features, i.e., the character sequence representation $\boldsymbol{e}_c(w)$, the word embedding $\boldsymbol{e}_w(w)$, and some human designed features on the word-face $\boldsymbol{e}_h(w)$.

For the character-level representation $\boldsymbol{e}_c(w)$ of w, we use the *one-layer convolution network* model (Kim, 2014) on its character sequence $\{c_1, c_2, \cdots, c_m\} \in \mathcal{V}_c$, where $\mathcal{V}_c$ is the character vocabulary. Each character c is represented using

$$\boldsymbol{v}(c) = \mathbf{W}_c(c),$$

where $\mathbf{W}_c$ denotes a character embedding lookup table. The *one-layer convolution network* is then applied to $\{\boldsymbol{v}(c_1), \boldsymbol{v}(c_2), \cdots, \boldsymbol{v}(c_m)\}$ to obtain $\boldsymbol{e}_c(w)$.

For the word-level representation $\boldsymbol{e}_w(w)$ of w, we introduce an unique dense vector for w, which is initialized with Stanford's GloVe word embeddings[1] (Pennington et al., 2014) and fine-tuned during model training.

For the human designed features $\boldsymbol{e}_h(w)$ of w, we introduce a set of binary feature indicators. These indicators are designed on options proposed by Collobert et al. (2011): *allCaps*, *upperInitial*, *lowercase*, *mixedCaps*, *noinfo*. If any feature is activated, its corresponding indicator is set to 1, otherwise 0. This way, it can keep the capitalization information erased during lookup of the word embedding.

The final word presentation independent to its context $\boldsymbol{e}(w) \in R^{k_w}$ of w, is obtained by concatenating these three part of features:

$$\boldsymbol{e}(w) = [\boldsymbol{e}_c(w) \oplus \boldsymbol{e}_w(w) \oplus \boldsymbol{e}_h(w)], \quad (8)$$

where $\oplus$ denotes the concatenation operation. Based on this representation, we apply a bidirectional LSTM (BiLSTM) network (Huang et al., 2015), taking $\boldsymbol{e}(w_t), w_t \in s$ as step input, to model context information of w_t given the sentence s. Hidden states of the forward and backward LSTMs at the t step are concatenated:

$$\boldsymbol{e}(w_t|s) = [\overrightarrow{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t], \quad (9)$$

to form the representation of w_t given s.

Loss Definition. Given the word representation, $\boldsymbol{e}(w|s)$, of w conditional on s, its probability to be predicted as positive class is modeled by:

$$f(w|s) = \sigma(\boldsymbol{w}_p^T \boldsymbol{e}(w|s) + b), \quad (10)$$

where σ denotes the sigmoid function, $\boldsymbol{w}_p$ is a trainable parameter vector and b is the bias term. The prediction risk on this word given label y is defined by:

$$\ell(f(w|s), y) = |y - f(w|s)|. \quad (11)$$

Note that $\ell(f(w|s), y) \in [0, 1)$ is upper bounded. The empirical training loss is defined by:

$$\hat{R}_\ell(f) = \pi_p \hat{R}_p^+(f) + \max\left\{0, \hat{R}_u^-(f) - \pi_p \hat{R}_p^-(f)\right\}, \quad (12)$$

[1] http://nlp.stanford.edu/projects/glove/

where

$$\hat{R}_p^+(f) = \frac{1}{|\mathcal{D}^+|} \sum_{w|s \in \mathcal{D}^+} \ell(f(w|s), 1),$$
$$\hat{R}_p^-(f) = 1 - \hat{R}_p^+(f),$$
$$\hat{R}_u^-(f) = \frac{1}{|\mathcal{D}^u|} \sum_{w|s \in \mathcal{D}^u} \ell(f(w|s), 0),$$

and π_p is the ratio of entity words within $\mathcal{D}^u$.

In addition, during our experiments, we find out that due to the class imbalance problem (π_p is very small), f inclines to predict all instances as the negative class, achieving a high value of accuracy while a small value of F1 on the positive class. This is unacceptable for NER. Therefore, we introduce a class weight γ for the positive class and accordingly redefine the training loss as:

$$\hat{R}_\ell(f) = \gamma \cdot \pi_p \hat{R}_p^+(f) + \max\left\{0, \hat{R}_u^-(f) - \pi_p \hat{R}_p^-(f)\right\}. \quad (13)$$

Label Inference. Once the PU classifier has been trained, we use it to perform label prediction. However, since we build a distinct classifier for each entity type, a word may be predicted as positive class by multiple classifiers. To address the conflict, we choose the type with the highest prediction probability (evaluated by $f(w|s)$). Predictions of classifiers of the other types are reset to 0.

At inference time, we first solve the type conflict using the above method. After that, consecutive words being predicted as positive class by the classifier of the same type are treated as an entity. Specifically, for sequence $s = \{w_1, w_2, w_3, w_4, w_5\}$, if its predicted labels by the classifier of a given type are $L = \{1, 1, 0, 0, 1\}$, then we treat $\{w_1, w_2\}$ and $\{w_5\}$ as two entities of the type.

3.5 Adapted PU Learning for NER

In PU learning, we use the empirical risk on labeled positive data, $\frac{1}{n_p}\sum_{i=1}^{n_p} \ell(f(x_i^p), 1)$, to estimate the expectation risk of positive data. This requires that the positive examples x_i^p draw identically independent from the distribution $\mathrm{P}(\mathrm{X}|\mathrm{Y} = 1)$. The requirement is usually hard to satisfy, using a simple dictionary to perform data labeling.

To alleviate this problem, we propose an adapted method, motivated by the AdaSampling (Yang et al., 2017) algorithm. The key idea of the proposed method is to adaptively enrich the named entity dictionary. Specifically, we first train a PU learning classifier f and use it to label the unlabeled dataset. Based on the predicted label, it extracts all of the predicted entities. For a predicted entity, if it occurs over k times and all of its occurrences within the unlabeled dataset are predicted as entities, we will add it into the entity dictionary in the next iteration. This process iterates several times until the dictionary does not change.

4 Experiments

In this section, we empirically study:

- *the general performance of our proposed method using simple dictionaries;*
- *the influence of the unlabeled data size;*
- *the influence of dictionary quality, such as size, data labeling precision and recall;*
- *and the influence of the estimation of π_p.*

4.1 Compared Methods

There are five indispensable baselines with which our proposed Adapted PU learning (**AdaPU**) algorithm should compare. The first one is the dictionary matching method, which we call **Matching**. It directly uses the constructed named entity dictionary to label the testing set as illustrated in Alg. 1. The second one is the supervised method that uses the same architecture as f but trains on fine-grained annotations (fully labeled $\mathcal{D}^u$ and $\mathcal{D}^+$). In addition, it applies the BIOES label assignment mechanism for model training. We call this baseline **BiLSTM**. The third one is the **uPU** algorithm, which uses cross entropy loss to implement ℓ. The fourth one is the bounded uPU (**buPU**) algorithm, which implement ell with mean absolute error. Compared with AdaPU, it does not apply the non-negative constraint and does not perform dictionary adaptation. The last one is the bounded non-negative PU learning (**bnPU**) algorithm, which does not perform dictionary adaptation compared with AdaPU.

Additionally, we compared our method with several representative supervised methods that have achieved state-of-the-art performance on NER. These methods include: **Stanford NER (MEMM)** (McCallum et al., 2000) a maximum-entropy-markov-model-based method; **Stanford NER (CRF)** (Finkel et al., 2005) a conditional-random-field-based method; and **BiLSTM+CRF**

Dataset	Type	# of l.w.	Precision	Recall
CoNLL (en)	PER	2,507	89.26	17.38
	LOC	4,384	85.07	50.03
	ORG	3,198	86.17	29.45
	MISC	1,464	92.13	30.59
CoNLL (sp)	PER	574	90.24	37.84
	LOC	272	84.93	16.39
	ORG	702	96.87	27.19
	MISC	157	68.15	11.94
MUC	PER	788	74.56	28.50
	LOC	511	89.43	43.33
	ORG	1,257	97.74	30.38
Twitter	PER	1,842	79.26	26.03
	LOC	1,109	90.96	34.15
	ORG	398	83.77	20.58

Table 1: Data labeling results using the dictionary: the number of labeled words (# of l.w.), the word-level precision ($\frac{\text{\# of true labeled words}}{\text{\# of total labeled words}}$) and recall.

(Huang et al., 2015) a neural-network-based method as the BiLSTM baseline, but additionally introducing a CRF layer.

4.2 Datasets

CoNLL (en). CoNLL2003 NER Shared Task Dataset in English (Tjong Kim Sang and De Meulder, 2003) collected from Reuters News. It is annotated by four types: PER, LOC, ORG, and MISC. We used the official split training set for model training, and testb for testing in our experiments, which contains 203K and 46K tokens, respectively. In addition, there are about 456k additional unlabeled tokens.

CoNLL (sp). CoNLL2002 Spanish NER Shared Task Dataset (Sang and Erik, 2002) collected from Spanish EFE News Agency. It is also annotated by PER, LOC, ORG, and MISC types. The training and test data sets contain 273k and 53k lines, respectively.

MUC. Message Understanding Conference 7 released by Chinchor (1998) for NER. It has about 190K tokens in the training set and 64K tokens in the testing set. For the sake of homogeneity, we perform entity detection on PER, LOC, and ORG in this study.

Twitter. Twitter is a dataset collected from Twitter and released by Zhang et al. (2018). It contains 4,000 tweets for training and 3,257 tweets for testing. Every tweet contains both textual information and visual information. In this work, we only used the textual information to perform NER and we also only performed entity detection on PER, LOC, and ORG.

Dataset	PER	LOC	ORG	MISC
CoNLL (en)	.055/.053	.041/.038	.049/.045	.023/.020
CoNLL (sp)	.019/.018	.019/.017	.030/.027	– – ––
MUC-7	.022/.019	.025/.023	.037/.034	– – ––
Twitter	.058/.055	.046/.044	.021/.018	– – ––

Table 2: True/Estimated value of π_p.

For the proposed method and the PU-learning-based baselines, we used the training set of each dataset as $\mathcal{D}$. Note that we did not use label information of each training set for training these models.

4.3 Build Named Entity Dictionary

For CoNLL (en), MUC, and Twitter datasets, we collected the first 2,000 popular English names in England and Wales in 2015 from ONS[2] to construct the PER dictionary. For LOC, we collected names of countries and their top two popular cities[3] to construct the dictionary. While for MISC, we turned country names into the adjective forms, for example, England → English, and China → Chinese, and used the resultant forms to construct the dictionary. For ORG, we collected names of popular organizations and their corresponding abbreviations from Wikipedia [4] to construct the dictionary. We also added names of some international companies[5], such as Microsoft, Google, and Facebook, into the dictionary. In addition, we added some common words occurring in organization names such as "Conference", "Cooperation", "Commission", and so on, into the dictionary.

For CoNLL (sp), we used DBpedia query editor[6] to select the most common 2000 names of the people who was born in Spain to construct the PER dictionary. We further used Google translator to translate the English LOC, ORG, MISC dictionary into Spanish.

The resultant named entity dictionaries contain 2,000 person names, 748 location names, 353 organization names, and 104 MISC entities. Table 1 lists some statistic information of the data labeling results with these dictionaries using Alg.

[2]http://www.ons.gov.uk/ons/index.html

[3]https://en.wikipedia.org/wiki/List_of_countries_by national_capital_largest_and_second-largest_cities

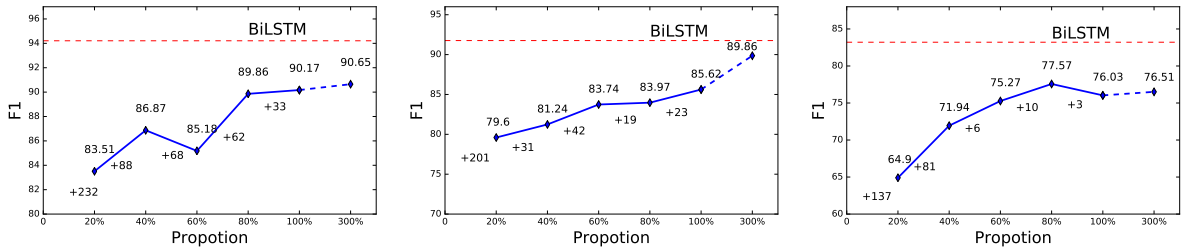
[4]https://en.wikipedia.org/wiki/List_of_intergovernmental_organizations

[5]https://en.wikipedia.org/wiki/List_of_multinational_corporations

[6]http://dbpedia.org

Dataset	Type	MEMM	CRF	BiLSTM	BiLSTM+CRF	Matching	uPU	buPU	bnPU	AdaPU
CoNLL (en)	PER	91.61	93.12	94.21	95.71	6.70	74.22	85.01	87.21	90.17
	LOC	89.72	91.15	91.76	93.02	67.16	69.88	81.27	83.37	85.62
	ORG	80.60	81.91	83.21	88.45	46.65	73.64	74.72	75.29	76.03
	MISC	77.45	79.35	76.00	79.86	53.98	68.90	68.90	66.88	69.30
	Overall	86.13	87.94	88.30	90.01	44.90	72.32	79.20	80.74	82.94
CoNLL (sp)	PER	86.18	86.77	88.93	90.41	32.40	82.28	83.76	84.30	85.10
	LOC	78.48	80.30	75.43	80.55	28.53	70.44	72.55	73.68	75.23
	ORG	79.23	80.83	79.27	83.26	55.76	69.82	71.22	69.82	72.28
	Overall	81.14	82.63	80.28	84.74	42.23	73.84	74.50	74.43	75.85
MUC	PER	86.32	87.50	85.71	84.55	27.84	77.98	84.94	84.21	85.26
	LOC	81.70	83.83	79.48	83.43	62.82	64.56	72.62	75.61	77.35
	ORG	68.48	72.33	66.17	67.66	51.60	45.30	58.39	58.75	60.15
	Overall	74.66	76.47	73.12	75.08	50.12	63.87	69.89	70.06	71.60
Twitter	PER	73.85	80.86	80.61	80.77	41.33	67.30	72.72	72.68	74.66
	LOC	69.35	75.39	73.52	72.56	49.74	59.28	61.41	63.44	65.18
	ORG	41.81	47.77	41.39	41.33	32.38	31.51	36.78	35.77	36.62
	Overall	61.48	67.15	65.60	65.32	37.90	53.63	57.16	57.54	59.36

Table 3: Model performance by F1 on the testing set of each dataset. The first group of models are all fully-supervised, which use manual fine-grained annotations. while the second group of models use only named entity dictionaries to perform the NER task.



(a) PER (b) LOC (c) ORG

Figure 2: F1 of AdaPU on the testing set of CoNLL (en) using different portion of the training data set for model training. The red dot line denotes performance of BiLSTM. '+k' means that it labels k more unique words on the additional 20% (e.g., 40%-20%) of training data.

1. From the table, we can see that the precision of the data labeling is acceptable but the recall is quite poor. This is expectable and is a typical problem of the method using only dictionaries to perform NER.

4.4 Estimate π_p

Before disscussing the estimation of π_p defined in Eq. (12), let us first look at some statistic information of the four studied datasets. Table 2 lists the true value of π_p = (# of entity words)/(# of words of the training set) for different entity types over dataset. From the table, we can see that the variation of π_p cross different datasets is quite small. This motivates us to use the value of π_p obtained from an existing labeled dataset as an initialization. The labeled dataset may be from other domains or be out-of-date. In this work, we initially set $\pi_p = 0.04, 0.04, 0.05, 0.03$ for PER, LOC, ORG, and MISC, respectively. Starting from this value, we trained the proposed model and used it to perform prediction on the unlabeled dataset. Based on the predicted results, we re-estimate the value of π_p. The resulted values are listed in table 2 and were used throughout our experiments without further illustration.

4.5 Results

Following the protocol of most previous works, we apply the entity-level (exact entity match) F1 to evaluate model performance.

General Performance. Table 3 shows model performance by entity type and the overall performance on the four tested datasets. From the table, we can observe: **1)** The performance of the Matching model is quite poor compared to other models. We found out that it mainly resulted from low recall values. This accords with our

discussion in §1 and shows its inapplicability using such simple dictionaries. **2)** Those PU-learning-based methods achieve significant improvement over Matching on all datasets. This demonstrates the effectiveness of the PU learning framework for NER in the studied setting. **3)** buPU greatly outperforms uPU. This verifies our analysis in §2.3 about the necessity to make ℓ upper bounded. **4)** bnPU slightly outperforms buPU on most of datasets and entity types. This verifies the effectiveness of the non-negative constraint proposed by Kiryo et al. (2017). **5)** The proposed AdaPU model achieves further improvement over bnPU, and it even achieves comparable results with some supervised methods, especially for the PER type. This verifies the effectiveness of our proposed method for enriching the named entity dictionaries.

Type	Size	Precision	Recall
PER	10,159 (2,000)	89.65 (89.26)	19.08 (17.38)
LOC	10,106 (748)	71.77 (85.07)	56.42 (50.03)
ORG	10,039 (353)	83.42 (86.17)	28.59 (29.45)

Table 4: Statistic information of the extended dictionary v.s. (that of the original dictionary).

Model	PER	LOC	ORG	Overall
Matching	9.10 (6.70)	69.85 (67.16)	45.52 (46.65)	41.40 (39.39)
AdaPU	91.14 (90.17)	77.60 (85.62)	76.67 (76.03)	81.87 (82.94)

Table 5: F1 of the proposed method using the extend dictionary v.s. (that using the original dictionary) on CoNLL (en) testing set.

Influence of Unlabeled Data Size. We further study the influence of the unlabeled data size to our proposed method. To perform the study, we used 20%, 40%, 60%, 80%, 100%, and 300% (using additional unlabeled data) of the training data set of CoNLL (en) to train AdaPU, respectively. Figure 2 depicts the results of this study on PER, LOC, and ORG. From the figure, we can see that increasing the size of training data will, in general, improve the performance of AdaPU, but the improvements are diminishing. Our explanation of this phenomenon is that when the data size exceeds a threshold, the number of unique patterns becomes an sublinear function of the data size. This was verified by the observation from the figure, for example, on PER, it labeled 232 unique words on 20% of training data, while it only labeled 88 more unique words after introducing additional 20% of training data.

π_p	PER	LOC	ORG	MISC	Overall
True	90.21	85.06	77.17	69.85	83.13
Estimated	90.17	85.62	76.03	69.30	82.94

Table 6: F1 of the proposed method on CoNLL (en) when using True/Estimated value of π_p.

Influence of Dictionary. We then study the influence of the dictionary on our proposed model. To this end, we extended the dictionary with DBpedia using the same protocol proposed by Chiu and Nichols (2016). Statistic information of the resultant dictionary is listed in table 4, and model performance using this dictionary is listed in table 5. A noteworthy observation of the results is that, on LOC, the performance should decrease a lot when using the extended dictionary. We turn to table 4 for the explanation. We can see from the table that, on LOC, the data labeling precision dropped about 13 points (85.07 $\rightarrow$ 71.77) using the extend dictionary. This means that it introduced more false-positive examples into the PU learning and made the empirical risk estimation bias more to the expectation when using the extended dictionary.

Influence of π_p Value. Table 6 lists the performance of AdaPU when using the true or estimated value of π_p as listed in table 2. From the table, we can see that the proposed model using the estimated π_p only slightly underperforms that using the true value of π_p. This shows the robustness of the proposed model to a small variation of π_p and verifies the effectiveness of the π_p estimation method.

5 Related Work

Positive-unlabeled (PU) learning (Li and Liu, 2005) aims to train a classifier using only labeled positive examples and a set of unlabeled data, which contains both positive and negative examples. Recently, PU learning has been used in many applications, e.g., text classification (Li and Liu, 2003), matrix completion (Hsieh et al., 2015), and sequential data (Nguyen et al., 2011). The main difference between PU learning and semi-supervised learning is that, in semi-supervised learning, there is labeled data from all classes, while in PU learning, labeled data only contains examples of a single class .

AdaSampling (Yang et al., 2017) is a self-training-based approach designed for PU learning, which utilizes predictions of the model to iteratively update training data. Generally speaking, it initially treats all unlabeled instances as negative examples. Then, based on the model trained in the last iteration, it generates the probability $p(y = 0|\boldsymbol{x}_i^u)$ of an unlabeled example $\boldsymbol{x}_i^u$ to be a negative one. This value, in turn, determines the probability of $\boldsymbol{x}_i^u$ to be selected as the negative examples for model training in next iteration. This process iterates for an acceptable result.

6 Conclusion

In this work, we introduce a novel PU learning algorithm to perform the NER task using only unlabeled data and named entity dictionaries. We prove that this algorithm can unbiasedly and consistently estimate the task loss as if there is fully labeled data. And we argue that it can greatly reduce the requirement on sizes of the dictionaries. Extensive experimental studies on four NER datasets validate its effectiveness.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No. 2018YFB1005104, 2018YFC0831105, 2017YFB1002104,), National Natural Science Foundation of China (No. 61751201, 61532011), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), STCSM (No.16JC1420401,17JC1420200), ZJLab.

References

Nachman Aronszajn. 1950. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools*, pages 1–8. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Nancy Chinchor. 1998. Overview of muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association of Computational Linguistics*, 4(1):357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Felipe Cucker and Steve Smale. 2002. On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49.

Louise Delèger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferrė, Philippe Bessieres, and Claire Nėdellec. 2016. Overview of the bacteria biotope task at bionlp shared task 2016. In *Proceedings of the 4th BioNLP Shared Task Workshop*, pages 12–22.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Martin Gerner, Goran Nenadic, and Casey M Bergman. 2010. Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(1):85.

Mourad Gridach. 2017. Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91.

Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S Dhillon. 2015. Pu learning for matrix completion. In *ICML*, pages 2445–2453.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Empirical Methods in Natural Language Processing*.

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.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Xiao-Li Li and Bing Liu. 2005. Learning from positive and unlabeled examples with different data distributions. In *European Conference on Machine Learning*, pages 218–229. Springer.

Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pages 587–592.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. *AAAI Conference on Artificial Intelligence*.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2015. Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries. *Information*, 6(4):848–865.

Yuan Liu, Qiang Tan, and Kun Xu Shen. 1994. The word segmentation rules and automatic word segmentation methods for chinese information processing. *Qing Hua University Press and Guang Xi*, page 36.

Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

David Nadeau, Peter D Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 266–277. Springer.

Minh Nhut Nguyen, Xiao-Li Li, and See-Kiong Ng. 2011. Positive unlabeled learning for time series classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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*, pages 703–711.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. 2004. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076.

Tjong Kim Sang and F Erik. 2002. Introduction to the conll-2002 shared task: language-independent named entity recognition. *Computer Science*, pages 142–147.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 104–107. Association for Computational Linguistics.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. Linden: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st international conference on World Wide Web*, pages 449–458. ACM.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, 8(1):29–48.

Pengyi Yang, Wei Liu, and Jean Yang. 2017. Positive unlabeled learning via wrapper-based adaptive sampling. In *IJCAI*.

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.

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *AAAI*.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 1554-1564.*

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics.

Jianhan Zhu, Victoria Uren, and Enrico Motta. 2005. Espotter: Adaptive named entity recognition for web browsing. In *Biennial Conference on Professional Knowledge Management/Wissensmanagement*, pages 518–529. Springer.

A Proof of Theorem 1

Proof. Let denote $\hat{R}^s_\ell(f)$ the empirical estimation of $R_\ell(f)$ with k randomly labeled examples. Since ℓ is bounded, C_0, M, and B are finite. According to the Lemma in (Rosasco et al., 2004) we have:

$$\begin{aligned}&\mathrm{P}\{\mathrm{S}\in\mathcal{D}|\sup_{\mathrm{f}\in\mathcal{H}_\mathrm{R}}|\mathrm{R}_\ell(\mathrm{f})-\hat{\mathrm{R}}^\mathrm{s}_\ell(\mathrm{f})|\le\epsilon\}\\&\ge 1-2N(\frac{\epsilon}{4L_M})e^{-\frac{k\epsilon^2}{8B^2}}.\end{aligned}\tag{14}$$

Then, the empirical estimation error of $R_\ell(f) - \hat{R}_\ell(f)$ in PU learning can be written as:

$$\begin{aligned}&R_\ell(f)-\hat{R}_\ell(f)\\&=\left(\mathbb{E}_{\mathbf{X}}\ell(f(\boldsymbol{x}),0)-\frac{1}{n_u}\sum_{i=1}^{n_u}\ell((f(x_i^u),0)\right)\\&+\pi_p\left(\mathbb{E}_{\mathbf{X}|\mathrm{Y}=1}\ell(f(\boldsymbol{x}),1)-\frac{1}{n_p}\sum_{i=1}^{n_p}\ell(f(x_i^p),1)\right)\\&-\pi_p\left(\mathbb{E}_{\mathbf{X}|\mathrm{Y}=1}\ell(f(\boldsymbol{x}),0)-\frac{1}{n_p}\sum_{i=1}^{n_p}\ell(f(x_i^p),0)\right)\end{aligned}\tag{15}$$

Thus,

$$\begin{aligned}&|R_\ell(f)-\hat{R}_\ell(f)|\\&\le\left|\mathbb{E}_{\mathbf{X}}\ell(f(\boldsymbol{x}),0)-\frac{1}{n_u}\sum_{i=1}^{n_u}\ell((f(x_i^u),0)\right|\\&+\pi_p\left|\mathbb{E}_{\mathbf{X}|\mathrm{Y}=1}\ell(f(\boldsymbol{x}),1)-\frac{1}{n_p}\sum_{i=1}^{n_p}\ell(f(x_i^p),1)\right|\\&+\pi_p\left|\mathbb{E}_{\mathbf{X}|\mathrm{Y}=1}\ell(f(\boldsymbol{x}),0)-\frac{1}{n_p}\sum_{i=1}^{n_p}\ell(f(x_i^p),0)\right|\end{aligned}\tag{16}$$

Let $\mathrm{I}_\ell(\mathbf{X},0)$ denote

$$\mathbb{E}_{\mathbf{X}}\ell(f(\boldsymbol{x}),0)-\frac{1}{n_u}\sum_{i=1}^{n_u}\ell((f(x_i^u),0).$$

According to Eq. 14, we have:

$$\begin{aligned}&\mathrm{P}\{\mathrm{S}\in\mathcal{D}|\sup_{\mathrm{f}\in\mathcal{H}_\mathrm{R}}|\mathrm{I}_\ell(\mathbf{X},0)|\le\epsilon\}\\&\ge 1-2N(\frac{\epsilon}{4L_M})e^{-\frac{n_u\epsilon^2}{8B^2}}\end{aligned}\tag{17}$$

Similarly, let $\mathrm{I}_\ell(\mathbf{X}|\mathrm{Y}=1,1)$ denote

$$\mathbb{E}_{\mathbf{X}|\mathrm{Y}=1}\ell(f(\boldsymbol{x}),1)-\frac{1}{n_p}\sum_{i=1}^{n_p}\ell(f(x_i^p),1),$$

and $\mathrm{I}_\ell(\mathbf{X}|\mathrm{Y}=1,0)$ denote

$$\mathbb{E}_{\mathbf{X}|\mathrm{Y}=1}\ell(f(\boldsymbol{x}),0)-\frac{1}{n_p}\sum_{i=1}^{n_p}\ell(f(x_i^p),0),$$

we have:

$$\begin{aligned}&\mathrm{P}\{\mathrm{S}\in\mathcal{D}|\sup_{\mathrm{f}\in\mathcal{H}_\mathrm{R}}|\mathrm{I}_\ell(\mathbf{X}|\mathrm{Y}=1,1)|\le\epsilon\}\\&\ge 1-2N(\frac{\epsilon}{4L_M})e^{-\frac{n_p\epsilon^2}{8B^2}},\end{aligned}\tag{18}$$

and

$$\begin{aligned}&\mathrm{P}\{\mathrm{S}\in\mathcal{D}|\sup_{\mathrm{f}\in\mathcal{H}_\mathrm{R}}|\mathrm{I}_\ell(\mathbf{X}|\mathrm{Y}=1,0)|\le\epsilon\}\\&\ge 1-2N(\frac{\epsilon}{4L_M})e^{-\frac{n_p\epsilon^2}{8B^2}},\end{aligned}\tag{19}$$

Therefore,

$$\begin{aligned}&\mathrm{P}\{\mathrm{S}\in\mathcal{D}|\sup_{\mathrm{f}\in\mathcal{H}_\mathrm{R}}|\mathrm{R}_\ell(\mathrm{f})-\hat{\mathrm{R}}_\ell(\mathrm{f})|\le(1+2\pi_\mathrm{p})\epsilon\}\\&\ge\min(1-2N(\frac{\epsilon}{4L_M})e^{-\frac{n_p\epsilon^2}{8B^2}},\\&\qquad 1-2N(\frac{\epsilon}{4L_M})e^{-\frac{n_u\epsilon^2}{8B^2}})\\&=1-2N(\frac{\epsilon}{4L_M})e^{-\frac{\min(n_p,n_u)\epsilon^2}{8B^2}}\end{aligned}\tag{20}$$

The theorem follows replacing ϵ with $\frac{1}{1+2\pi_p}\epsilon$. □