BETTER HANDLING UNLABELED ENTITY PROBLEM USING PU-LEARNING AND NEGATIVE SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

The NER task is largely developed based on well-annotated data. However, in many scenarios, the entities may not be fully annotated, leading to performance degradation. A common approach for this problem is to distinguish unlabeled entities from negative instances using labeled data. However, the vast differences between entities make such empirical approaches difficult to realize. Our solution is to treat unlabeled entities based on both empirical inference and random sampling. To this end, we propose a simple yet effective two-step method that consists of a novel Positive-Unlabeled (PU-learning) algorithm and negative sampling, in which PU-learning distinguishes part of the unlabeled entities from negative instances based on confidence threshold. In general, the proposed method can mitigate the impact of unlabeled entities at the outset and can be easily applied to any character-level NER model. We verify the effectiveness of our method on several NER models and datasets, showing a strong ability to deal with unlabeled entities. Finally, in real-world situations, we establish new state-of-the-art results on many benchmark NER datasets.

1 INTRODUCTION

Named entity recognition (NER) is a well-studied task in natural language processing (NLP), which has received significant attention (Huang et al., 2015; Ma & Hovy, 2016; Akbik et al., 2018; Li et al., 2020b). Previous methods have reached great success in the area of NER (Zhang & Yang, 2018; Gui et al., 2019; Jin et al., 2019). However, the majority of them rely on the well annotated data and ignore the potential unlabeled entities, which are commonly encountered in many cases. Li et al. (2020c) discovered that NER models suffer significantly from the lack of annotations and referred to this as the unlabeled entity problem.

Recently, numerous approaches have been developed to alleviate the unlabeled entity problem. These works can be divided into two groups. To begin with, Li et al. (2020c) used negative sampling and trained a span-based model to mitigate the misguidance of unlabeled entities. The random sampling method is flexible since it makes no assumptions or inferences about unlabeled entities. This line of work was further extended by Li et al. (2022) which used a new weighted sampling distribution to perform a better sampling. Another line of work makes full use of the labeled data to approximate the true label sequences or detect the potential unlabeled entities. For instance, Mayhew et al. (2019) proposed the Constrained Binary Learning method which adaptively trained a binary classifier and assigned weights to each token using the CoDL framework (Chang et al. (2007)). Peng et al. (2019) trained a PU-learning (Liu et al., 2002; 2003; Elkan & Noto, 2008) classifier to perform label prediction which can unbiasedly and consistently estimate the task loss. Jie et al. (2019) used the k-fold cross-validation to estimate a distribution in partial CRF model (Tsuboi et al. (2008)). Furthermore, Peng et al. (2021) trained a span selector using reinforcement learning in the process of negative sampling. These approaches mitigate the impact of unlabeled entities by rational use of labeled data.

The current methods have achieved great improvement in datasets with unlabeled entities. Despite the success, they also have some limitations. In particular, the approaches that use labeled data rely on the quality of labeled data and usually cannot fully recognize the unlabeled entities. Moreover, the random sampling approach does not consider the role of labeled data at all, thus will loss some helpful information. Therefore, entirely ignoring or relying on the labeled data may be suboptimal in handing unlabeled entities. We believe that some unlabeled entities are identifiable, but others are not. Thus, combining the advantages of both kinds of approaches is necessary to better solve the unlabeled entity problem.

In this work, our goal is to find a more proper way to deal with unlabeled entities. Through empirical studies, we found that the labeled data does have the ability to identify the unlabeled entities. However, such ability is limited since we can only detect a part of the unlabeled entities precisely and the others are still uncertain. Our idea is to handle unlabeled entities by steps to obtain a better performance. In general, we propose a two-step method that can be easily generalized to many existing NER models. To begin with, we generate a novel PU-learning algorithm based on self-supervision to detect some unlabeled entities with high confidence. Then, we apply negative sampling to mitigate the influence of the other unlabeled entities. Inspired by the empirical study, we detect unlabeled entities at the start of the training before fitting the noise. Furthermore, we use the angle-based technique ((Zhang & Liu (2014); Zhang et al. (2016); Fu et al. (2022)) to enhance our PU-learning algorithm, which is first encountered in deep neural networks.

We verify the effectiveness of the proposed method using four classic Chinese NER models and six benchmark Chinese NER datasets. The proposed method generally enables significant improvement of all baseline models on synthetic datasets. In real-world situations, our approach also delivers new state-of-the-art performances. Notably, the additional computational cost caused by our method is significantly small.

2 Preliminaries

2.1 Unlabeled Entity Problem

Unlabeled entity problem occurs when some ground truth entities are not annotated, and as result, are treated as negative instances. This problem may caused by the negligence of human annotator or the limited coverage of machine annotator.

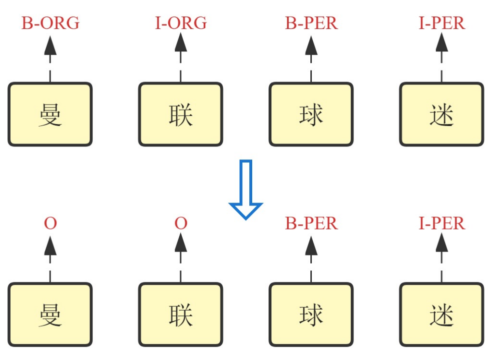


Figure 1: The unlabeled entity problem.

For instance, given a phrase "曼联(Manchester United) 球迷(Football Fan)" which adopts BIO tagging scheme. The true label sequence is {B-ORG,I-ORG,B-PER,I-PER}. As shown in Figure 1, when the unlabeled entity problem occurs, the entity "曼联" is not annotated and the corresponding labels are replaced with tag O.

2.2 Motivation

As shown in Li et al. (2020c), there are two causes for performance degradation in unlabeled entity problem: the reduction of annotated entities and the misguidance of unlabeled entities. The second cause has far more influence than the first one and it can be mitigated by removing all unlabeled entities (Li et al. (2020c)). Ideally, we would be able to detect all unlabeled entities correctly.

However, the unlabeled entities are always confused with true O-chars. Here, O-char denotes the character with the tagging O in BIO or BMESO tagging scheme. Therefore, the current challenge is how to discriminate between unlabeled entities and true O-chars.

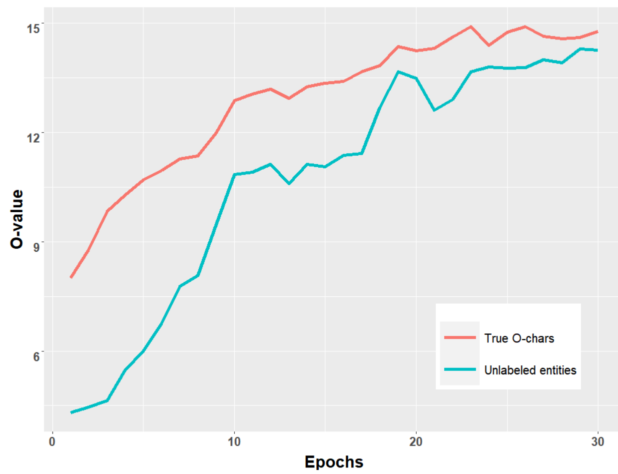
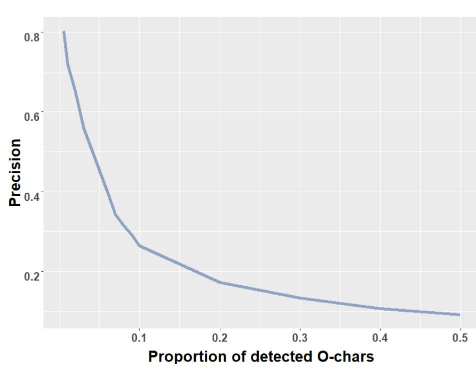


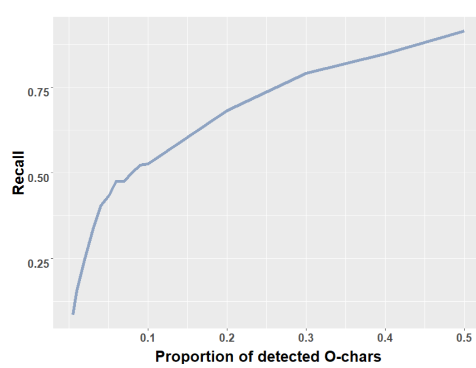
Figure 2: Difference between unlabeled entities and true O-chars.

Arpit et al. (2017) showed that the deep neural networks learn simple patterns first and subsequently fit the noise. Motivated by this, we conduct a simple study to understand the training process for unlabeled entities. To begin with, we introduce the definition of O-value. For character-level sequence labeling tasks, we always generate a k-dimensional decision vector for one character to match the classification, where k is the number of tags. Given the decision vector $\boldsymbol{h} \in \mathbb{R}^k$ of character c. The O-value of character c is defined as $\boldsymbol{h}[o]$, which is the value in $\boldsymbol{h}$ corresponding to the O tag. Then, we train 30 epochs in the synthetic Weibo dataset with 50% unlabeled entities and plot the average O-values for unlabeled entities and true O-chars. Note that we count every character to calculate the average. As shown in Figure 2, the average O-value for unlabeled entities is much smaller than the true O-chars in the first few epochs. With the increase of epochs, their difference becomes smaller and almost disappears. The result indicates that the unlabeled entities are learned by steps, which confirms the point of Arpit et al. (2017).

Since the difference in average O-values is significant, we try to detect unlabeled entities by their O-values. We fix the number of epochs at 2, select a portion of O-chars from small to large according to the O-values and analyze the results for detecting unlabeled entities. The results are in Figure 3. In general, the precision keeps decreasing while the recall keeps increasing as we pick more O-chars. As a result, we can only detect a portion of unlabeled entities correctly. The others are still confused with the true O-chars after training, and we need to pay more to handle them.



(a) Precision



(b) Recall

Figure 3: Precision and recall for detecting unlabeled entities.

3 METHODOLOGY

In this section, we will introduce: (1) The proposed two-step learning approach, which encounters both PU-learning and negative sampling; (2) The application of angle-based technique; (3) Implement of CRF layer.

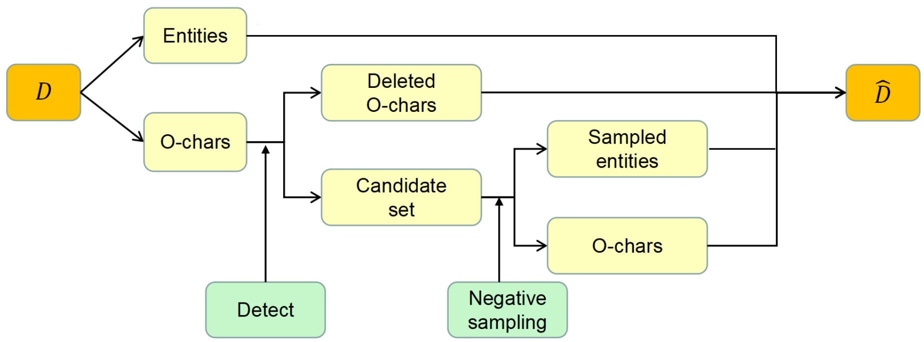


Figure 4: The architecture of our two-step method, in which the PU-learning and negative sampling approaches are encountered. Where D is the original dataset and $\hat{D}$ is the modified dataset.

3.1 ROBUST TWO STEP LEARNING

The main structure of our method is shown in Figure 4. In the first step, we aim to detect and then remove some identified unlabeled entities, in preparation for the following negative sampling step. Inspired by the empirical studies in Section 2.2, we propose our PU-learning algorithm. The details are as follows.

Given a NER model and the training set $D = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}), i = 1, \ldots, N\}$, where $\boldsymbol{y}^{(i)} = (y_1^{(i)}, \ldots, y_{s_i}^{(i)})$ and s_i is the length of i-th sentence. We first train the model for n epochs and get the corresponding model $\boldsymbol{f}$ and decision vectors $\boldsymbol{f}(\boldsymbol{x}^{(i)}) = \boldsymbol{h}^{(i)} = (\boldsymbol{h}_1^{(i)}, \ldots, \boldsymbol{h}_{s_i}^{(i)}) \in \mathbb{R}^{s_i \times k}$ for each sentence. Afterward, we summarize O-values for all O-chars in

$$\{\boldsymbol{h}_j^{(i)}[o] \mid 1 \leq i \leq N, 1 \leq j \leq s_i, y_j^{(i)} = O\}.$$

Then, we select the smallest $\lceil \lambda * m \rceil$ O-values and remove their corresponding O-chars in the training set. Here, $0 < \lambda < 1$ is a hyperparameter, m is the number of O-values, and $\lceil \rceil$ is the ceiling function. The hyperparameter λ is significant since it determines the range of the deleted O-chars. In practise, we recommend choosing a small λ to ensure accuracy.

Take the cross-entropy loss as example. After removing some potential unlabeled entities, the training loss becomes:

$$(\sum_i \sum_j -\log \boldsymbol{q}_j^{(i)}[y_j]) - (\sum_i \sum_{j \in \mathcal{A}_j} -\log \boldsymbol{q}_j^{(i)}[y_j]), \tag{1}$$

where $\boldsymbol{q}_j^{(i)} = \text{Softmax}\ (\boldsymbol{h}_j^{(i)})$ and $\mathcal{A}_j$ denotes the set of index for the deleted O-chars in j-th sentence.

After removing some unlabeled entities in the first step, we next apply random sampling on the remaining O-chars. We generate all the span candidates in $\boldsymbol{x}^{(i)}$ as

$$\mathcal{L}_i = \{(j, k) | \forall j \leq l \leq k, y_l^{(i)} = O, l \notin \mathcal{A}_i\}.$$

Then, we randomly sample $\lceil \gamma * s_i \rceil$ spans from $\mathcal{L}_i$ and give these spans a special non-entity label. Here, γ is a hyperparameter used to control the degree of sampling. The label of these spans are replaced with the corresponding BMES or BI tags and the modified label sequences are defined as $\hat{\boldsymbol{y}}^{(i)}$. Ultimately, the dataset we used is $\hat{D} = \{(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{y}}^{(i)}, \mathcal{A}_i), i = 1, \ldots, N\}$.

As a remark, Li et al. (2020c) used negative sampling in a span level NER model, which treated a span as the basic unit for labeling. Unlike Li et al. (2020c), we still treat the character as the basic unit. Thus, our method can be easily applied to many existing NER models since we only modify the training dataset and do no harm to the model. At the inference step, we treat the predicted non-entity spans as negative instances.

3.2 Angle-based Decision Vector

We note that the current decision vector used in our PU-learning algorithm is suboptimal. For instance, the decision vector we have always used is undefinable since it does not satisfy the sum-to-zero constraint, which will degrade the performance. Take $\boldsymbol{h}_j^{(i)}$ as an example. If we add a constant to each value of $\boldsymbol{h}_j^{(i)}$, the classification results remain unchanged, but the accuracy of detecting small O-values is significantly reduced. Thus, we would like to implement the proposed method using the decision vector with sum-to-zero constraint. However, adding a sum-to-zero constrain directly in the deep neural network would be challenging to optimize.

Note that a single decision vector can be used to separate two classes. Analogously, a $(k-1)$-dimensional decision vector should be sufficient for a k-category classification problem (Zhang & Liu (2014)). As a result, using a k-dimensional decision vector to match a k-category classification problem is redundant. To handle these difficulties, we will then introduce the angle-based technique.

In the previous study of machine learning, Zhang & Liu (2014) proposed an angle-based technique for large margin classifiers, which uses $(k-1)$-dimensional decision vector and implicitly transfers the sum-to-zero constraint onto the newly defined functional margins. Thus, the angle-based classifiers can automatically satisfy the sum-to-zero constraint with a few parameters.

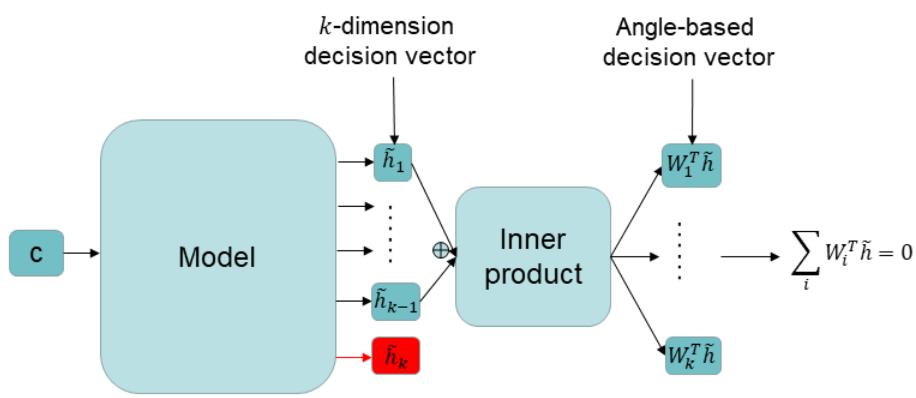
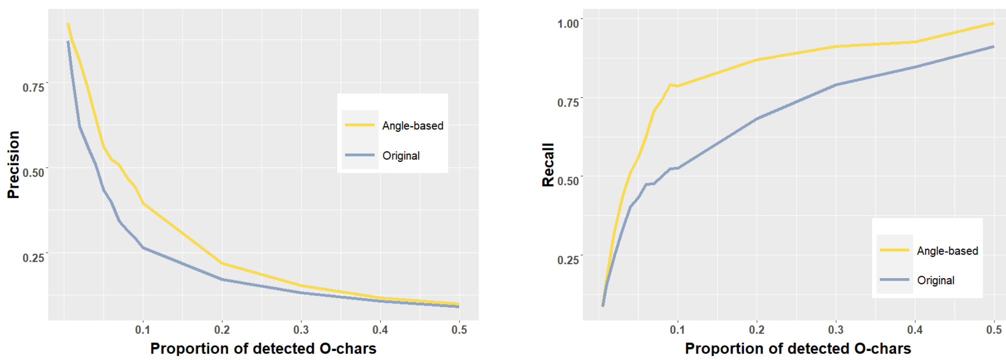


Figure 5: How angle-based technique works.

To begin with, consider a centered simplex in $\mathbb{R}^{k-1}$ with elements

$$\boldsymbol{W}_j = \begin{cases} (k-1)^{-1/2}\mathbf{1}, & j=1 \\ -\frac{1+\sqrt{k}}{(k-1)^{3/2}}\mathbf{1} + \sqrt{\frac{k}{k-1}}e_{j-1}, & 2 \le j \le k \end{cases}$$

where $e_j \in \mathbb{R}^{k-1}$ is a vector of 0's except its j-th element is 1, and $\mathbf{1} \in \mathbb{R}^{k-1}$ is a vector of 1. We only require a $(k-1)$-dimensional output to match the k-category classification in our tasks by the angle-based setting. For illustration, we define the required output for one character as $\widetilde{\boldsymbol{h}}$. Then we generate the k-dimensional angle-based decision vector by inner product, namely $(\widetilde{\boldsymbol{h}}^T\boldsymbol{W}_1, \widetilde{\boldsymbol{h}}^T\boldsymbol{W}_2, \ldots, \widetilde{\boldsymbol{h}}^T\boldsymbol{W}_k)$. One can verify that the sum-to-zero constraint, $\sum_i \widetilde{\boldsymbol{h}}^T\boldsymbol{W}_i$, is automatically satisfied. Moreover, the needed hidden outputs are $(k-1)$-dimensional, which can reduce the parameter size to a certain extent. Figure 5 shows the details about how angle-based technique works. See (Zhang et al., 2016; 2018; Yang et al., 2021; Fu et al., 2022) for more implements of angle-based technique in large-margin classifiers. To show effectiveness, we conduct a similar study as in section 2.2. From Figure 6, we can find that the angle-based model has higher precision and recall in detecting unlabeled entities than the original.



(a) Precision (b) Recall

Figure 6: Precision and recall for detecting unlabeled entities.

3.3 Implement of CRF.

Due to the successive structure of the CRF layer, we cannot directly remove the O-chars in the first step. Thus, we will introduce how to implement our method in the CRF layer. Given the decision vectors $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n$ for sentence $\boldsymbol{x}$ with length n. For original CRF, the probability of a label sequence $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ is

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp\left(\sum_i (\boldsymbol{h}_{i,y_i} + \boldsymbol{T}_{y_{i-1},y_i})\right)}{\sum_{\widetilde{\boldsymbol{y}}} \exp\left(\sum_i (\boldsymbol{h}_{i,\widetilde{y}_i} + \boldsymbol{T}_{\widetilde{y}_{i-1},\widetilde{y}_i})\right)},$$

where $\widetilde{\boldsymbol{y}}$ represents an arbitrary label sequence and $\boldsymbol{T}_{y_{i-1},y_i}$ is the transition score from y_{i-1} to y_i. Utilizing the CRF structure, we remove the O-chars by short circuiting and the modified probability is

$$\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \mathcal{A}) = \frac{\exp\left(\sum_{i \notin \mathcal{A}} (\boldsymbol{h}_{i,y_i} + \boldsymbol{T}_{y_{i-1},y_i})\right)}{\sum_{\widetilde{\boldsymbol{y}}} \exp\left(\sum_{i \notin \mathcal{A}} (\boldsymbol{h}_{i,\widetilde{y}_i} + \boldsymbol{T}_{\widetilde{y}_{i-1},\widetilde{y}_i})\right)},$$

where $\mathcal{A}$ is the set of O-chars deleted in the first step.

For training set $\hat{D} = \{(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{y}}^{(i)}, \mathcal{A}_i), i = 1, \ldots, N\}$, the sentence-level log-likelihood loss is:

$$L = -\sum_j \log(\hat{p}(\hat{\boldsymbol{y}}^{(j)} \mid \boldsymbol{x}^{(j)}, \mathcal{A}_j)).$$

For decoding, we use the Viterbi algorithm to find the label sequence with highest score.

4 Experiments

We conduct an extensive set of experiments on multiple classical Chinese NER models to investigate the effectiveness of our method. Experiments on both real-world datasets and synthetic datasets are available. We will demonstrate that our method can be robust against unlabeled entities in the synthetic datasets. For the real-world dataset, we achieve state-of-the-art performances in several datasets. Standard F1-score (F1) is used as evaluation metrics.

4.1 Experimental Setup

Datasets. We conduct our experiments on six benchmark Chinese NER datasets, which are (1) **Weibo** (Peng & Dredze (2015)) (2) **Resume** (Zhang & Yang (2018)) (3) **Ontonotes 4.0** (Weischedel et al. (2011)) (4) **MSRA** (Levow (2006)) (5) **EC** (Yang et al. (2018)) (6) **NEWS** (Yang et al. (2018)).

The statistics of the datasets can be found in Appendix A.1. We construct the synthetic datasets by contaminating two small datasets, Weibo and Resume. Specifically, we randomly select 10%,. . . ,70% entities in training set and flip their labels to O tags.

Baselines. To test the effectiveness of our method, we chose four different classical Chinese NER models, including

- **Bi-LSTM** A common Bi-LSTM+CRF (Huang et al., 2015) structure using the word2vec (Mikolov et al. (2013)) embedding pretrained by (Zhang & Yang, 2018).
- **FLAT** The Flat Lattice Transformer (Li et al., 2020a) using the same embedding as Bi-LSTM.
- **BERT+Word** A strong BERT base model proposed by Liu et al. (2021), which uses bilinear attention weighted word vector as a supplement to the BERT input, and uses LSTM and CRF as fusion layer and inference layer respectively.
- **LEBERT** The recommended model in Liu et al. (2021), which is a combination of Lexicon Adapter and Transformer.

Overall, our choice of models is diverse, with two using BERT, two using Transformer, and one using just Bi-LSTM. For synthetic datasets, we report the F1-scores of each model with and without using our method. In addition, we carry out ablation studies to examine the contribution of the angle-based technique in synthetic Weibo dataset. For real-world datasets, we test the performance of our method using BERT+Word and LEBERT models.

Hyperparameters. Recall that the parameter n is the number of epochs for the first training, λ stands for the proportion of removed O-chars and ratio γ represents the degree of negative sampling. We have found that the difference between unlabeled entities and true O-chars is huge in the first few epochs. Thus, the candidate set we used for n is $\{1, 2, 3\}$. Note that it is both inappropriate to set λ as too small or too large values. Empirically speaking, we select λ in $\{0.1 \times 2^{-5}, 0.1 \times 2^{-4}, \ldots, 0.1\}$. We tend to use a large λ when the proportion of unlabeled entities increases. For real-world datasets, we always use the minimum value, 0.1×2^{-5}. To select the best parameter γ, we use the grid search in $\{0, 0.1, 0.2, 0.3\}$. Other hyperparameters are the same as the original method.

4.2 Overall Results

Synthetic Datasets. Tables 1-4 summarize the results of synthetic datasets. For clarity reasons, ”/A” indicates that we do not employ the angle-based technique. In general, each baseline method achieves better performance in handing unlabeled entities. For instance, when the proportion of unlabeled entities in Resume increases from 0.1 to 0.7, the F1-score of the original LEBERT model decreases by 90.67. After applying the proposed method, the F1-score only decreases by 3. This demonstrates the effectiveness of our method, even with a very few labeled entities. Furthermore, the ablation studies show that the angle-based decision vector is a beneficial addition for our method. By comparing the results, one can find that our method is more effective on BERT+Word, LEBERT than FLAT, Bi-LSTM. This is likely because a strong NER model can acquire more precise underlying information during training, thus improving the performance of our method.

Prob.	BERT+Word				
	Weibo			Resume	
	Original	Our	Our/A	Original	Our
0.1	64.85	**67.03**	66.67	94.96	**95.66**
0.2	60.95	**66.36**	65.39	94.56	**95.36**
0.3	56.24	**65.39**	64.87	94.18	**95.18**
0.4	53.08	**65.09**	63.82	92.88	**95.09**
0.5	49.10	**63.88**	62.73	64.78	**94.45**
0.6	43.29	**63.63**	61.31	13.24	**94.13**
0.7	31.75	**63.38**	59.68	1.63	**85.10**

Table 1: The experiment results (F1-score) for BERT+Word on synthetic datasets.

Prob.	LEBERT				
	Weibo			Resume	
	Original	Our	Our/A	Original	Our
0.1	65.80	**69.34**	68.28	94.50	**95.16**
0.2	65.31	**68.75**	67.45	94.16	**94.91**
0.3	62.31	**68.17**	67.04	93.17	**94.95**
0.4	56.76	**67.71**	66.59	91.11	**94.46**
0.5	54.37	**65.73**	65.07	70.07	**94.02**
0.6	43.87	**64.32**	63.92	43.54	**93.27**
0.7	29.34	**62.88**	61.19	3.82	**92.40**

Table 2: The experiment results (F1-score) for LEBERT on synthetic datasets.

Prob.	FLAT				
	Weibo			Resume	
	Original	Our	Our/A	Original	Our
0.1	57.38	**59.97**	59.62	95.10	**95.27**
0.2	57.18	**59.68**	59.10	94.95	**95.22**
0.3	53.90	**59.00**	57.99	94.81	**95.14**
0.4	49.25	**56.09**	55.12	94.16	**94.70**
0.5	47.30	**53.53**	53.25	89.52	**92.50**
0.6	46.46	**49.85**	49.45	57.89	**72.39**
0.7	42.14	**49.19**	48.82	24.29	**68.58**

Table 3: The experiment results (F1-score) for FLAT on synthetic datasets.

Prob.	Bi-LSTM				
	Weibo			Resume	
	Original	Our	Our/A	Original	Our
0.1	47.73	**49.47**	48.67	93.96	**94.13**
0.2	46.32	**48.63**	46.71	93.93	**94.07**
0.3	41.91	**45.03**	44.78	93.10	**93.72**
0.4	33.58	**37.75**	36.56	92.01	**93.02**
0.5	28.68	**35.10**	33.22	83.53	**87.66**
0.6	19.54	**27.52**	27.26	39.81	**47.85**
0.7	8.24	**12.76**	9.83	5.33	**17.85**

Table 4: The experiment results (F1-score) for Bi-LSTM on synthetic datasets.

Real-world Datasets. As shown in Tables 5 and 6, our method helps the BERT+Word model outperforms its original procedure in each real-world dataset. For the LEBERT model, we achieve better results in Weibo, Resume, Msra, EC and NEWS, which are the state-of-the-art results. However, the F1-score on Ontonotes 4.0 is 0.12% worse than the original. These results indicate that our method is not only robust on synthetic datasets, but are also competitive on the real-world datasets. There are two possible reasons account for this. First, the real-world datasets may also have unlabeled entities, which can also lead to misguidance. Second, some true O-chars may also misguide the NER models. We will further discuss it in Appendix A.2.

In addition, we analyze the extra training time required after applying our method, which is also available in Appendix A.3.

5 Related Work

NER is an indispensable component in many downstream NLP tasks. In Chinese NER, leveraging the word information can significantly improve the performance. A possible strategy is to perform word segmentation first, followed by the NER task. Unfortunately, because the cross-domain word segmentation is still an unsolved problem (Liu & Zhang, 2012; Jiang et al., 2013; Liu et al., 2014; Qiu & Zhang, 2015; Chen et al., 2017; Huang et al., 2017), this strategy may result in error

Model	Weibo	Resume	Ontonotes 4.0	Msra
Lattice LSTM(Zhang & Yang (2018))	63.34	94.51	75.49	92.84
CAN (Zhu et al. (2019))	59.31	94.94	73.64	92.97
WC-LSTM (Liu et al. (2019))	65.30	94.49	75.79	93.50
SoftLexicon (Ma et al. (2019))	69.11	95.35	81.34	95.54
FLAT	68.07	95.78	80.56	95.46
BERT+Word	68.32	95.46	81.03	95.32
LEBERT	70.75	96.08	**82.08**	95.70
Our /in Bert+Word	70.26	96.16	81.34	95.44
Our /in LEBERT	**71.00**	**96.24**	81.98	**95.73**

Table 5: The experiment results (F1-score) on Weibo, Resume, Ontonotes and Msra.

Model	EC	NEWS
Weighted Partial CRF (Jie et al. (2019))	61.75	78.64
Bert-MRC (Li et al. (2020b))	55.72	74.55
Negative Sampling (Li et al. (2020c))	66.17	85.39
Negative Sampling (Li et al. (2022))	67.03	86.15
BERT+Word	78.10	95.27
LEBERT	78.75	95.88
Our /in Bert+Word	79.95	96.08
Our /in LEBERT	**80.12**	**96.24**

Table 6: The experiment results (F1-score) on EC and NEWS.

propagation. Another line of work is enhancing lexicon information in character-based models which has demonstrated a significant benefit in merging the word information and preventing the error propagation, such as Lattice LSTM (Zhang & Yang (2018)), FLAT (Li et al. (2020a)), and LEBERT (Liu et al. (2021)).

However, the NER models suffer from the unlabeled entity problem in many scenarios (Zhang et al. (2020)). Recently, numerous works have been proposed to address this issue. Fuzzy CRF and AutoNER (Shang et al. (2020)) handle the unlabeled entities by learning from high-quality phrases. Another approach for solving this problem involves the use of PU-learning (Mayhew et al., 2019; Peng et al., 2019), which build a distinct binary classifier to detect unlabeled entities. Partial CRF (Yang et al., 2018; Jie et al., 2019) is an extension of common CRF which allows NER models to learning from incomplete annotations. Li et al. (2020c) discovered that the primary cause of performance degradation is misguidance of unlabeled entities and employed a span-level negative sampling model to mitigate the misguidance. The current methods are either rely entirely on the labeled data or do not encounter labeled data at all, which may be suboptimal to handle the unlabeled entity problem.

6 Conclusion

In this work, we propose a two-step method to handle the unlabeled entity problem. Our first step is based on the finding from empirical studies. We generate a novel PU-learning algorithm and it is proven to be effective in detecting unlabeled entities. The second step borrows the negative sampling method in the sequence labeling task, which is a helpful aid to the first step. Our method can be easily generalized to many NER models and only requires a few additional computing resources. Compared to the existing robust methods, our method is more effective and efficient. Furthermore, we are the first to employ the angle-based technique in deep neural networks which certainly enhances the effectiveness of our first step. Experiments on synthetic datasets have verified that our method is robust to the unlabeled entities and can improve the performance of each baseline model significantly. On multiple real-world NER datasets, we create new state-of-the-art results.

REFERENCES

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649, 2018.

Devansh Arpit, Stanisaw Jastrzbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242. PMLR, 2017.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 280–287, 2007.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuan-Jing Huang. Adversarial multi-criteria learning for Chinese word segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1193–1203, 2017.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 213–220, 2008.

Sheng Fu, Piao Chen, Yufeng Liu, and Zhisheng Ye. Simplex-based multinomial logistic regression with diverging numbers of categories and covariates. *Statistica Sinica*, Online, 2022. Doi: 10.5705/ss.202021.0082.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. CNN-based Chinese NER with lexicon rethinking. In *ijcai*, pp. 4982–4988, 2019.

Shen Huang, Xu Sun, and Houfeng Wang. Addressing domain adaptation for Chinese word segmentation with global recurrent structure. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 184–193, 2017.

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

Wenbin Jiang, Meng Sun, Yajuan Lü, Yating Yang, and Qun Liu. Discriminative learning with natural annotations: Word segmentation as a case study. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 761–769, 2013.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 729–734, 2019.

Yanliang Jin, Jinfei Xie, Weisi Guo, Can Luo, Dijia Wu, and Rui Wang. LSTM-CRF neural network with gated self attention for Chinese NER. *IEEE Access*, 7:136694–136703, 2019.

Gina-Anne Levow. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pp. 108–117, 2006.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuan-Jing Huang. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6836–6842, 2020a.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5849–5859, 2020b.

Yangming Li, Shuming Shi, et al. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*, 2020c.

Yangming Li, Lemao Liu, and Shuming Shi. Rethinking negative sampling for dandling missing entity annotations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7188–7197, 2022.

Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, volume 2, pp. 387–394. Sydney, NSW, 2002.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building text classifiers using positive and unlabeled examples. In *Third IEEE international conference on data mining*, pp. 179–186. IEEE, 2003.

Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. An encoding strategy based word-character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2379–2389, 2019.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. Lexicon enhanced Chinese sequence labeling using BERT adapter. 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)*, pp. 5847–5858, 2021.

Yang Liu and Yue Zhang. Unsupervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of COLING 2012: Posters*, pp. 745–754, 2012.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 864–874, 2014.

Ruotian Ma, Minlong Peng, Qi Zhang, and Xuanjing Huang. Simplify the usage of lexicon in Chinese NER. *arXiv preprint arXiv:1908.05969*, 2019.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1064–1074, 2016.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2409–2419, 2019.

Nanyun Peng and Mark Dredze. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 548–554, 2015.

Shi Peng, Yong Zhang, Zhengyun Wang, Dingkang Gao, Feng Xiong, and Haoyang Zuo. Named entity recognition using negative sampling and reinforcement learning. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 714–719. IEEE, 2021.

Likun Qiu and Yue Zhang. Word segmentation for Chinese novels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pp. 2054–2064. Association for Computational Linguistics, 2020.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 897–904, 2008.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 2011.

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

Yi Yang, Yuxuan Guo, and Xiangyu Chang. Angle-based cost-sensitive multicategory classification. *Computational Statistics & Data Analysis*, 156:107107, 2021.

Chong Zhang and Yufeng Liu. Multicategory angle-based large-margin classification. *Biometrika*, 101(3):625–640, 2014.

Chong Zhang, Yufeng Liu, Junhui Wang, and Hongtu Zhu. Reinforced angle-based multicategory supsport vector machines. *Journal of Computational and Graphical Statistics*, 25(3):806–825, 2016.

Chong Zhang, Minh Pham, Sheng Fu, and Yufeng Liu. Robust multicategory support vector machines using difference convex algorithm. *Mathematical Programming*, 169(1):277–305, 2018.

Haisong Zhang, Lemao Liu, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Jianchen Zhu, et al. Texsmart: A text understanding system for fine-grained NER and enhanced semantic analysis. *arXiv preprint arXiv:2012.15639*, 2020.

Yue Zhang and Jie Yang. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1554–1564, 2018.

Yuying Zhu, Guoxin Wang, and Börje F Karlsson. CAN-NER: Convolutional attention network for Chinese named entity recognition. *arXiv preprint arXiv:1904.02141*, 2019.

A Appendix

A.1 Statistics of Datasets

Dataset	**Type**	**Train**	**Dev**	**Test**
Weibo	Sentence	1.4k	0.27k	0.27k
	Char	73.8k	14.5k	14.8k
Resume	Sentence	3.8k	0.46k	0.48k
	Char	124.1k	13.9k	15.1k
OntoNotes	Sentence	15.7k	4.3k	4.3k
	Char	491.9k	200.5k	208.1k
MSRA	Sentence	46.4k	–	4.4k
	Char	2169.9k	–	172.6k
EC	Sentence	1.2k	0.4k	0.8k
	Char	8.6k	3.1k	6.1k
NEWS	Sentence	3.0k	3.33k	3.19k
	Char	139.8k	149.0k	132.1k

Table 7: The statistics of the datasets.

A.2 CASE STUDY

We have shown the validity of our method in real-world datasets. Note that the real-world datasets may contain very few or even no unlabeled entities. One natural question is how our method improves the performance of BERT+Word and LEBERT. This is demonstrated by analyzing some removed true O-chars on Weibo. We show a portion of the removed true O-chars and the corresponding sentence fragments in Table 8. These removed true O-chars may of three kinds. To begin with, they might be entities from other categories, such as “上海国际车展” and “卫生院”. Second, they may be close to the existing entities, such as “何主席” and “沈太太”. Note that if we substitute any other person entity for these O-chars, the sentences will continue to flow smoothly. Third, they may be fabricated or erroneous, such as “安大略省”, “淘宝元” and “吴小”. To conclude, we speculate that such O-chars may also misguide the NER model.

Sentence fragments	Removed true O-chars
上海国际车展	上海国际车展 Shanghai International Auto Show
安大略省旅游局	安大略省 Andalue Province
探访一下何主席	何主席 chairman He
卫生院的那个	卫生院 health center
淘宝元专区	淘宝元 Baoyuan Tao
吴小公民的微博	吴小 Xiao Wu
相当沈太太	沈太太 Mrs. Shen

Table 8: Examples of removed true O-chars.

A.3 EFFICIENCY OF OUR METHOD

Table 9 reports the extra training time required after applying our method in LEBERT. Note that parameter λ and γ has a negligible effect on computational speed. Hence we only report the results when n is varied. According to the table, our method adds no more than 8% additional training time when $n = 1$. If we use $n = 3$, the extra training time required can still be under 20%. Thus, the computational cost of our method is significantly small.

n	Weibo	Resume	Ontonotes 4.0	Msra	EC	NEWS
1	2.21%	5.15%	6.29%	7.31%	6.73%	8.32%
2	4.17%	11.97%	13.56%	14.47%	14.32%	15.88%
3	6.30%	15.73%	17.79%	19.72%	18.89%	20.75%

Table 9: The percentage of extra training time due to our method.