Distill and Calibrate: Denoising Inconsistent Labeling Instances for Chinese Named Entity Recognition

Anonymous ACL submission

Abstract

Data-driving supervised models for named entity recognition (NER) have made significant improvements on standard benchmarks. However, such models often have severe performance degradation on large-scale noisy data. Thus, a practical and challenging question arises: Can we leverage only a small amount of relatively clean data to guide the NER model learning from large-scale noisy data? To answer this question, we focus on the inconsistent labeling instances problem. We observe that inconsistent labeling instances can be classified into five types of noise, each of which will largely hinder the model performance in our experiments. Based on the above observation, we propose a simple yet effective denoising framework named **D**istillation and **C**alibration for Chinese **NER** (DCNER). DCNER consists: (1) a Dual-stream Label Distillation mechanism for distilling five types of inconsistent labeling instances from the noisy data; and (2) a Consistency-aware Label Calibration network for calibrating inconsistent labeling instances based on relatively clean data. Additionally, we propose the first benchmark towards validating the ability of Chinese NER to resist inconsistent labeling instances. Finally, detailed experiments show that our method consistently and significantly outperforms previous methods on the proposed benchmark.

1 Introduction

Named entity recognition (NER), which is essential for natural language understanding, aims to detect and classify named entities in texts. For instance, given the sentence "He drove to the White House.", a NER system needs to detect the span of "the White House" and classify this mention to the Facility type. In recent years, data-driving supervised NER has made significant improvements on standard benchmarks (Ma and Hovy, 2016; Lample et al., 2016; Peters et al., 2018; Zhang and Yang, 2018; Li et al., 2020b,a, 2021; Yan et al., 2021). However, such models often have severe performance degradation on large-scale noisy data. Thus, a practical and challenging question arises: *Can we leverage only a small amount of relatively clean data to guide the NER model learning from large-scale noisy data?* To answer this question, we focus on the problem of inconsistent labeling instances.

Inconsistent labeling instances are widespread in human-annotated datasets. When building a large-scale NER dataset, annotators may have various standards for annotating a certain mention in their minds, leading to the problem of inconsistent labeling instances. This problem cannot be avoided by annotation guidelines because even the most detailed guideline cannot cover all entities. Due to the vagueness of Chinese word boundary, this problem is particularly prominent in Chinese NER (Zeng et al., 2021; Zhang et al., 2021). For example, OntoNotes 4.0 (Weischedel et al.) dataset, which is a standard Chinese NER benchmark, still cannot avoid this noise. In the statistics of (Zhang et al., 2021), the mention of "中国人民" (Chinese People) has two different labeling instances. It is labeled as "中国人民" (Chinese People) 23 times and "中国" (China) 13 times. We cannot arbitrarily assume that the majority is correct because both two labeling instances are reasonable and generally exist in the test set. Therefore, how to denoise inconsistent labeling instances is a challenging problem.

To tackle different types of noise, existing denoising methods can be roughly divided into three lines: methods towards the auto-annotated dataset (Hedderich and Klakow, 2018; Yang et al., 2018; Lange et al., 2019; Jie et al., 2019; Mayhew et al., 2019), methods towards instance-independent settings (Goldberger and Ben-Reuven, 2016; Zhou and Chen, 2021), and methods towards the human-annotated dataset (Wang et al., 2019; Jiang et al., 2021). Previous methods towards the auto-annotated dataset are confined to instances that cannot be auto-annotated due to the limited coverage of the dictionary. In fact, there are no inconsistent labeling instances in the auto-annotated dataset. For example, if "中国人民" (Chinese People) and "中国" (Chinese) exist in the dictionary at the same time, then the mention of "中国人民" (Chinese People) can only be labeled as "中国人民" instead of "中国人民". In contrast, methods for instance-independent settings generally do not consider the labeling instance, but directly randomly perturb the edge distribution of the label space. Our experiments show that, although these methods have achieved surprising effects on the target noise, they have to sacrifice the generalization on inconsistent labeling instances.

To tackle the problem of inconsistent labeling instances, we identify that inconsistent labeling instances can be classified into five types of noise, which are described in detail in Section 4. Further, our experiments demonstrate that each of the five noise types can seriously affect the model performance. Based on the above observation, we propose a two-stage denoising framework named **D**istillation and **C**alibration for Chinese **NER** (DCNER).

Specifically, in the first distillation phase, we propose a **D**ual-stream **L**abel **D**istillation mechanism (DLD) to distill five types of inconsistent labeling instances from the noisy data. Therefore, this mechanism can preserve the potential labeling instances in noisy data as much as possible. In the second calibration phase, we propose a **C**onsistency-aware **L**abel **C**alibration network (CLC) to calibrate inconsistent labeling instances. This network can calibrate inconsistent labeling instances based on relatively clean data and outputs the final prediction. Besides, we propose the first Chinese benchmark towards the ability of the NER model to resist inconsistent labeling instances. To obtain noisy data similar enough to the real-world dataset, we heuristically amplify the original noise in two Chinese NER benchmarks at different scales. In this way, we get two synthetic datasets, which can validate the NER model under multiple proportions of inconsistent noise.

In summary, our main contributions are:

- This is the first NER work to focus on denoising inconsistent labeling instances in such a scenario where only a small amount of relatively clean data and large-scale noisy data are available. To this end, we propose the first denoising framework (DCNER) for handling the inconsistent labeling instances problem. Besides, we propose the first Chinese NER benchmark towards validating the ability of NER model to resist the inconsistent labeling instances.

- In DCNER, a novel Dual-stream Label Distillation mechanism is proposed to distill inconsistent labeling instances from the noisy data, and a novel Consistency-aware Label Calibration network is proposed to calibrate inconsistent labeling instances based on relatively clean data.

- Experiments show that our method consistently and significantly outperforms previous methods on the proposed benchmark, exceeding by 4.29% and 14.74% on F1 score (without pre-training). Besides, ablation experiments prove the effectiveness of each phase in our method.

2 Related Work

This section emphasizes some representative methods we use for comparison towards the following three mainstream scenarios.

2.1 Towards the Auto-annotated Dataset

Existing methods towards the auto-annotated dataset are confined to instances that cannot be auto-annotated due to the limited coverage of the dictionary. (Hedderich and Klakow, 2018) add a global noise adaptation matrix on top of a BiLSTM to correct noisy labels for English NER. (Lange et al., 2019) enhance confusion-matrix based methods to capture feature-dependent noise. (Yang et al., 2018) design an agent as an instance selector based on reinforcement learning to distinguish positive sentences. (Jie et al., 2019) and (Mayhew et al., 2019) use self-training to adjust the weights of wrong labels and correct labels iteratively.

2.2 Towards Instance-independent Settings

Methods for instance-independent settings generally do not consider the type of noise, primarily based on noise transition matrix and the memorization effect of neural network (Zhou and Chen, 2021). (Goldberger and Ben-Reuven, 2016) model the relationship between noisy and clean labels with a confusion matrix. (Luo et al., 2017) apply a

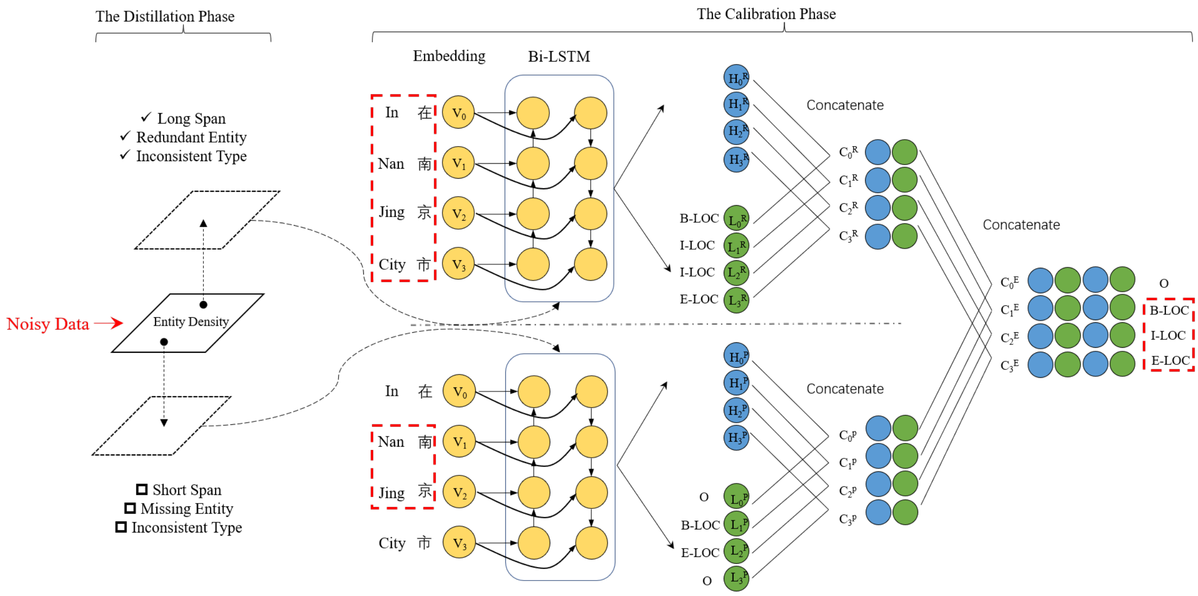

Figure 1: The architecture of our proposed DCNER.

dynamically generated matrices-based method to characterize clues about the noise patterns.

2.3 Towards the Human-annotated Dataset

This direction has received increasing attention in recent years. (Wang et al., 2019) aim to detect and down weight wrong labels based on self-training, resulting in a weighted training set. The first phase of our method is also based on self-training. However, our goal is fundamentally different from them. We aim to detect and separate inconsistent labeling instances entangled in the whole noisy data rather than arbitrarily asserting that a specific instance is more correct. (Jiang et al., 2021) have the same experimental settings as ours. Their idea is to use external resources for domain adaptation through pre-training models. Their method is limited to pre-trained models, but our method performs well with or without pre-training models.

3 Our Proposed NER Framwork

This section introduces in detail our proposed DC-NER. We first introduce the distillation phase with a self-training mechanism called Dual-stream Label Distillation. Then we describe the following calibration phase with a Consistency-aware Label Calibration network.

3.1 Dual-stream Label Distillation

The distillation phase is to detect and separate inconsistent labeling instances entangled in the whole noisy data. Our proposed Dual-stream Label Distillation is shown in Algorithm 1.

Algorithm 1: Dual-stream Label Distillation

Input: A NER network f, the noisy data $N = \langle \{x_1, \ldots, x_n\}, \{y_1, \ldots, y_n\} \rangle$, and hyper-parameters k.
Output: Two NER models: M^H and M^L.

Randomly partition N into k folds;
for *Each fold* N_k **do**
  $train\_set_k \longleftarrow N \setminus \mathrm{N}_k$;
  Train a NER model $M_k = f(train\_set_k)$;
  for *Each* $x_j \in N$ **do**
    $\hat{y}_j^k \longleftarrow$ M_k's prediction on x_j
$\hat{N}_k = \langle \{x_1, \ldots, x_n\}, \{\hat{y}_1^k, \ldots, \hat{y}_n^k\} \rangle$;
$D = \hat{N}_1 \cup, \ldots, \cup \hat{N}_k$;
N^H = High-Density Distillation (D, N);
N^L = Low-Density Distillation (D, N);
Train a NER model $M^H = f(N^H)$;
Train a NER model $M^L = f(N^L)$;
Return M^H and M^L;

Self-training: We randomly partition the noisy data N into k folds. when each fold is regarded as an independent development set dev_set_k, the other $(k-1)$ folds are combined as the corresponding training set $train\_set_k$. In this way, we get k new datasets for self-training. Then we train k NER models based on the k datasets. We use these

k NER models to predict the entire original noisy training data N. The sentences of N and the corresponding k prediction results are denoted as D integrally.

High-density Distillation: With k different labeling results in D for comparison, this part will automatically detect and separate the inconsistent labeling instances. We get all predicted entities for each sentence as a set E_k^i. Then we compare and merge the different labeling instances for each entity in E_1^i,...,E_k^i. Here, we tend to leave more non-entity labeling instances, which helps to recall more entities. As long as a non-entity labeling instance appears, we will leave it. Finally, for each sentence, we follow the following two rules: (1) keep the labeling instance that appears in $\hat{N}$ but does not appear in N; (2) always replace the shorter labeling instances N in with the longer ones in $\hat{N}$. Thus, we get the result of High-density Distillation N^H.

Low-density Distillation: Contrary to High-density Distillation, this strategy is to obtain lower entity density. Here, we tend to leave fewer non-entity labeling instances. As long as a non-empty labeling instance appears less than k times, we will leave remove it with an empty labeling instance. Finally, for each sentence, we follow the following two rules: (1) remove the labeling instance that appears in $\hat{N}$ but does not appear in N; (2) always replace the longer labeling instances in N with the shorter ones in $\hat{N}$. Thus, we get the result of High-density Distillation N^L.

3.2 Consistency-aware Label Calibration Network

The second calibrate phase is to select and edit the labeling instance consistent with the relatively clean data from the two sets of potential labeling instances of each entity. The workflow of our proposed Consistency-aware Label Calibration network is shown in Algorithm 2, and the architecture is shown in Figure 1.

Encoding: This paper treats NER as a sequence labeling problem for Chinese characters, which has achieved state-of-the-art performances. We convert NER into sequence labeling the BIEOS schema, following (Lample et al., 2016). In this way, each sentential character is assigned with one tag. We tag the entity with a single character by label "S-XX", the beginning character of an entity by "B-XX", the ending character of an entity by "E-XX", the internal character of an entity by "I-XX", and the non-entity character by the label "O", where "XX" denotes the type of an entity.

Algorithm 2: Consistency-aware Label Calibration Network

Input: This network F, the NER model with high-density labeling bias M^H, the NER model with low-density labeling bias M^L, the test set $test\_set$, and the relatively clean data $N^C = \left\langle \{x_1^F, \ldots, x_m^F\}, \{y_1^F, \ldots, y_m^F\} \right\rangle$

Output: Final NER prediction $\hat{Y}$.

$Y^H = M^H(N^C)$; ▷ high-density labels.
$Y^L = M^L(N^C)$; ▷ low-density labels.
Initialize $F(M^H, M^L)$;
Train $M^F = F(M^H, M^L, N^C, Y^H, Y^L)$;
$\hat{Y} = M^F(test\_set)$;
Return $\hat{Y}$;

Consistency-aware Label Calibration network has a pseudo-siamese structure, which models the sentence and its potential labeling instance information of each entity. The two parts of the pseudo-siamese structure are identical but with different parameters initialized from the two NER models. The two parts do not share parameters during the training process.

The input of the model is a sentence and the two sets of labels generated by the two NER models in the previous phase; its output is the new labels edited on the two sets of labels. We denote the sentence as $s = \{c_1, \ldots, c_n\}$ where c_i is the i-th character. By looking up the embedding vector from a pre-train character embedding matrix, each character c_i is represented as a vector, which denotes v_i.

$$\mathbf{v}_i = e^c\left(c_i\right) \tag{1}$$

e^c is a character embedding lookup table.

To capture contextual information around characters, we apply a bidirectional LSTM (BiLSTM) (Lampele et al., 2016) over $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. We then get the left-to-right hidden states and the right-to-left hidden states.

$$\overrightarrow{\mathbf{h}}_i^H = \overrightarrow{LSTM}^H\left(\mathbf{v}_i, \overrightarrow{\mathbf{h}}_{i-1}^H\right) \tag{2}$$

$$\overrightarrow{\mathbf{h}}_i^L = \overrightarrow{LSTM}^L\left(\mathbf{v}_i, \overrightarrow{\mathbf{h}}_{i-1}^L\right) \tag{3}$$

$$\overleftarrow{\mathbf{h}}_i^H = \overleftarrow{LSTM}^H\left(\mathbf{v}_i, \overleftarrow{\mathbf{h}}_{i+1}^H\right) \tag{4}$$

$$\overleftarrow{\mathbf{h}}_i^L = \overleftarrow{LSTM}^L\left(\mathbf{v}_i, \overleftarrow{\mathbf{h}}_{i+1}^L\right) \tag{5}$$

By concatenating left-to-right hidden states and the right-to-left hidden states of BiLSTM, we obtain the contextual representation $\mathbf{H}^H = \{\mathbf{h}_1^H, \ldots, \mathbf{h}_n^H\}$ and $\mathbf{H}^L = \{\mathbf{h}_1^L, \ldots, \mathbf{h}_n^L\}$.

$$\mathbf{h}_i^H = \overrightarrow{\mathbf{h}}_i^H \oplus \overleftarrow{\mathbf{h}}_i^H \tag{6}$$

$$\mathbf{h}_i^L = \overrightarrow{\mathbf{h}}_i^L \oplus \overleftarrow{\mathbf{h}}_i^L \tag{7}$$

We initialize the two BiLSTM by copying BiLSTM parameters of the NER model with high-density labeling bias M^H and the NER model with low-density labeling bias M^L, respectively. Note that in addition to BiLSTM, any structure that captures contextual information can be used in our network, but it must be consistent with the NER model used in the previous phase. We practice the two NER models to make predictions on the relatively clean training set N^C, generating high-density labels Y^H and low-density labels Y^L.

We map Y^H and Y^L to a 50-dimensional type vector space, which is concatenated with $\mathbf{H}^H$ and $\mathbf{H}^L$ respectively.

$$\mathbf{C}^H = Y^H \oplus \mathbf{H}^H \tag{8}$$

$$\mathbf{C}^L = Y^L \oplus \mathbf{H}^L \tag{9}$$

Finally, we concatenate $\mathbf{C}^H$ and $\mathbf{C}^L$.

$$\mathbf{C}^E = \mathbf{C}^H \oplus \mathbf{C}^L \tag{10}$$

Decoding and Training: We use a standard CRF (Lafferty et al., 2001) layer to capture the dependencies between sentential labels. The input of the CRF layer is $\mathbf{c}^E = \{c_1^E, \ldots, c_n^E\}$. CRF involves two parts for prediction. First, we compute the scores for each label based $\mathbf{h}_t$, resulting in $\mathbf{W}^{y_i}$, whose dimension is the number of output labels. The other part is a transition matrix $\mathbf{T}$ which defines the scores of two successive labels. $\mathbf{T}$ is also a model parameter. Based on $\mathbf{W}^{y_i}$ and $\mathbf{T}$, we use the Viterbi algorithm to find the best label sequence. The probability of the ground-truth tag sequence $y = \{y_1, \ldots, y_n\}$ is

$$p(y \mid s) = \frac{\exp\left(\sum_i \left(\mathbf{W}^{y_i}\mathbf{c}_i + \mathbf{T}_{(y_{i-1}, y_i)}\right)\right)}{\sum_{y'} \exp\left(\sum_i \left(\mathbf{W}^{y'_i} c_i + \mathbf{T}_{(y'_{i-1}, y'_i)}\right)\right)} \tag{11}$$

Here y' is an arbitrary label sequence, $\mathbf{W}^{y_i}$ is used for modeling emission potential for the i-th character in the sentence, and $\mathbf{T}$ is the transition matrix storing the score of transferring from one tag to another.

Given a relatively clean training data $\{(s_i, y_i)\}|_{i=1}^N$. We optimize the model by minimizing the negative log-likelihood loss with L_2 regularization. The loss function is defined as:

$$L = -\sum_{i=1}^{N} \log\left(P\left(y_i \mid s_i\right)\right) + \frac{\lambda}{2}\|\Theta\|^2 \tag{12}$$

where λ denotes the L_2 regularization parameter and Θ is the all trainable parameters set.

Inference: The inference will practice the Consistency-aware Label Calibration network together with the two NER models preserved from the previous phase.

4 Construction Details of Our Benchmark

This section introduces construction details of our proposed Chinese NER benchmark.

4.1 Five Inconsistent Labeling Types

First of all, we conclude and define five inconsistent labeling types in the human-annotated dataset, which are shown in Figure 2. Long Span Noise means that an entity is incorrectly labeled as a longer labeling instance in some samples. Short Span Noise means that an entity is incorrectly labeled as a shorter labeling instance in some samples. Inconsistent Type Noise means that an entity has more than one labeling instance with different types. Missing Entity Noise means that an entity is incorrectly labeled as a non-entity labeling instance in some samples. Redundant Entity Noise means that a non-entity is incorrectly labeled as an entity labeling instance in some samples.

4.2 Two Original Benchmarks

We chose to build our benchmark based on OntoNotes 4.0 and MSRA (Levow, 2006) which are both the standard Chinese NER benchmarks. Statistics of original benchmarks are shown in

Annotator: A	Annotator: B	Noise Type
✓ 克林顿是越战以来[第一次]ORDINAL访问越南的总统. ✓ Clinton is the [first]ORDINAL president to visit Vietnam since the Vietnam War.	□ 这是查尔斯王子[第一] ORDINAL次展示他新妻子的机会. □ It was Prince Charles' [first]ORDINAL chance to show off his new wife.	Long Span Noise & Short Span Noise
✓ [去年]DATE总产值首次突破一千亿. ✓ [Last year]DATE, the total output value exceeded 100 billion for the first time.	□ 日本检方针对[去年]油燃加工厂外泄事故起诉 6 名工厂的员工. □ Japanese prosecutors charged 6 employees of an oil-burning plant for a leak [last year].	Redundant Entity Noise & Missing Entity Noise
✓ [美国]NORP总统克林顿上台之初. ✓ At the beginning of [U.S.]NORP President Bill Clinton's administration.	□ [美国]GPE国务卿奥尔布赖特随后也将抵达这里. □ [U.S.]GPE Secretary of State Albright will also arrive later.	Inconsistent Type Noise

Figure 2: Cases for inconsistent labeling instances. The mention that the annotator considers to be an entity is marked in red, and the green character next to it represents its entity type.

Dataset	Type	Train	Dev	Test
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

Table 1: Statistics of original benchmarks.

Dataset	Type	Noisy data	Clean data	Dev	Test
DC-OntoNotes	Sentence	10.2K	2.3K	2.3K	5.2K
	Entity	25.3K	6.1K	5.5K	12.3K
DC-MSRA	Sentence	37.5k	4.2K	4.6	4.4K
	Entity	63.1K	7.1K	4.3	6.2K

Table 2: Statistics of our benchmark.

Dataset	Noise Type	Noise Ratio
DC-OntoNotes	Inconsistent Span	15%
	Inconsistent Type	5%
	Missing Entity	5%
	Redundant Entity	5%
DC-MSRA	Inconsistent Span	15%
	Inconsistent Type	5%
	Missing Entity	5%
	Redundant Entity	5%

Table 3: Statistics of noise in our benchmark. We merge Long Span Noise and Short Span Noise that are difficult to manually distinguish into Inconsistent Span Noise.

Table 1. We take the same data split as (Chen et al., 2006) on OntoNotes. The development set is used for reporting development experiments. The OntoNotes and MSRA datasets are in the news domain which is the most commonly involved field in natural language understanding.

4.3 The Benchmark We Synthesized

Statistics of our benchmark are shown in Table 2. These five inconsistent labeling types are very tricky for both humans and models, especially when they are entangled in the dataset. When reviewing such a noisy training set, humans will get lost in various seemingly reasonable labeling instances, and do not know which one to believe. When feeding such a noisy training set to previous NER models, their structures cannot notice the inconsistency at the labeling instance level. As a result, models only learn the most frequently occurring labeling instances. Therefore, detecting inconsistent labeling in the dataset is undoubtedly a huge workload.

This benchmark provides both noisy data and a small amount of relatively clean data. To obtain sufficient noisy data by the crowd, we heuristically amplify the original inconsistent labels in two benchmarks at different scales. Specifically, we automatically matched the entire training set according to the definitions of these five types of noise. Then we hired three part-time annotators to filter manually. We heuristically split and reorganize the inconsistent labeling instances we selected and match the remaining data set to obtain more potentially inconsistent labeling instances. After multiple iterations, all inconsistent labeling instances in the entire training set are obtained. In this way, we get two synthetic datasets, each with multiple proportions of inconsistent noise. The two synthetic datasets can simulate the real inconsistent labeling instances in the human-annotated datasets to a certain extent.

In addition, for OntoNotes 4.0, the relatively clean data is randomly sampled from the original development set, and we leave the remaining half as a new development set. For MSRA, the relatively clean data is manually sampled from the original training set, and we try to avoid typical inconsistent labeling instances. Statistics of Noise

Method	DC-OntoNotes			DC-MSRA		
	P.	R.	F.	P.	R.	F.
Base-Clean	56.36	53.21	54.73	75.43	68.68	71.90
Base-Noise	50.23	47.12	48.62	71.57	66.56	68.97
Base-Mix	59.68	55.21	57.36	73.05	67.70	70.27
Towards the Auto-annotated Dataset						
(Yang et al., 2018)	-	-	56.25	-	-	72.60
(Jie et al., 2019)	54.43	47.10	50.50	70.05	69.82	69.93
(Mayhew et al., 2019)	60.01	54.13	56.92	73.35	67.94	70.54
Towards Instance-independent Settings						
(Veit et al., 2017)	35.16	34.03	34.59	52.58	42.65	47.10
(Luo et al., 2017)	34.80	35.35	35.07	56.40	44.40	49.69
(Hedderich and Klakow, 2018)	36.88	39.17	37.99	56.46	62.85	59.48
Towards the Human-annotated Dataset						
(Wang et al., 2019)	62.58	57.41	59.88	74.59	70.62	72.56
DCNER	**67.86**	**60.86**	**64.17(+4.29)**	**89.15**	**85.60**	**87.34(+14.74)**
(Jiang et al., 2021) w/ BERT	73.44	69.29	71.30	**90.49**	88.07	89.26
DCNER w/ BERT	**73.62**	**75.60**	**74.60**	89.72	**92.11**	**90.90**

Table 4: Main results of our experiments. The training on noisy data may lead to a certain amount of variance in the evaluation scores. Therefore, we repeat all experiments of our main results five times and report the average.

in our benchmark are shown in Table . Among them, the reason for the higher proportion of Inconsistent Span Noise is that Chinese NER is prone to word segmentation confusion. In fact, 15% is close to the upper limit (18%) that we can achieve in this dataset with inconsistent labeling instances generated by humans.

5 EXPERIMENTS

In this section, we conduct a series of experiments to prove the effectiveness of our method. Besides, we also carry ablation experiments to prove the effectiveness of each phase in our method.

5.1 Experiment Setting

Character Embedding: In our experiments, We use the same character embeddings as (Zhang and Yang, 2018), which is pre-trained on Chinese Giga-Word. Its lexicon consists of 704.4k words, where the number of single-character, two-characters, and three-character words are 5.7k, 291.5k, 278.1k, respectively.

BERT Enhanced Character Embedding: Since pre-trained language models have been proven to be effective on several tasks, we also experiment with employing BERT (Devlin et al., 2018) to augment our model via BERT enhanced embedding. Note that in all experiments involving BERT, we used the Chinese BERT-Base model.

Hyper-parameter Setting: We implement our models in PyTorch (Paszke et al., 2019). Our models are optimized by Adam (Kingma and Ba, 2014) with a fixed learning rate of 0.01. The parameters are initialized by Xavier (Glorot and Bengio, 2010). We apply Dropout (Srivastava et al., 2014) with a 0.7 keep rate to our models. All runs are trained on GTX 1080Ti GPU with batch size 128. In the first phase of DCNER, we fix k as 5.

Evaluation: We use the strict F1 criteria as an evaluation metric, which is widely used for NER. In the strict F1 criteria, an entity is right only when the span and the type are consistent with the gold.

5.2 Baselines

We follow the setting of (Hedderich and Klakow, 2018), which uses a global confusion matrix for all noisy instances. We follow instructions by (Lange et al., 2019), adapting (Veit et al., 2017) and (Luo et al., 2017) models to the NER task. Thus, our method compares against them in our experiments. The work of (Wang et al., 2019), known as a very competitive general denoising framework based on self-training, has also been included in our comparison. We implement their methods based on the structure of BiLSTM. We use (Yang et al., 2018) as a comparison, which is also based on the structure of BiLSTM as instructions. Besides, We set up a series of BiLSTM based models. `Base-Clean` is trained only on the relatively clean data; `Base-Noise` is trained only on the noisy data; `Base-Mix` is trained on both the relatively clean data and the noisy data.

5.3 Main Results and Analysis

Table presents the comparisons among all approaches on our proposed benchmark. DCNER

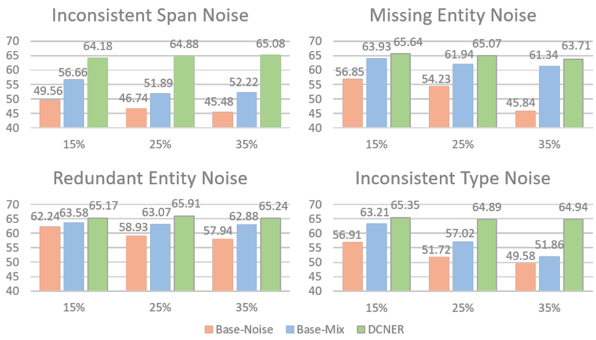

Figure 3: The effect of single noise. We merge Long Span Noise and Short Span Noise that are difficult to manually distinguish into Inconsistent Span Noise.

Method	DC-OntoNotes		
	P.	R.	F.
DCNER	**67.86**	**60.86**	**64.17**
w/ BERT	**73.62**	**75.60**	**74.60**
w/o Label Emb	67.29	60.87	63.91
w/o DLD	66.70	58.33	62.24
w/o High-D Bias	67.18	54.47	60.16
w/o Low-D Bias	65.45	57.29	61.10

Table 5: Ablation experiments.

has achieved consistent and significant improvements on DC-OntoNotes and DC-MSRA. The results show that our method is more resistant to inconsistent labeling noise than previous methods. Experiments also show that our method has good compatibility with BERT, and the performance has been significantly improved. We notice that when faced with inconsistent labeling noise, the previous classic anti-noise method does not seem to be as effective as in the face of distant supervision noise. Models designed for distant supervision noise (Hedderich and Klakow, 2018; Jie et al., 2019; Mayhew et al., 2019; Yang et al., 2018), models designed for noise in the general sense (Wang et al., 2019), and models migrated from other tasks work not as well as before. Some even poorly when compared to Base-Clean, Base-Noise or Base-Mix. We draw two conclusions from our experiments as follows: Firstly, the noise types of the auto-annotated dataset and the human-annotated are different. Secondly, restrictions on application conditions. Some methods (Veit et al., 2017; Luo et al., 2017; Jie et al., 2019; Mayhew et al., 2019; Wang et al., 2019) are designed for noisy training data only (we feed the mix of noisy data and relatively clean data instead), while some methods (Hedderich and Klakow, 2018; Jiang et al., 2021) require additional data for initialization (we disable external resources). However, for the principle of a fair comparison, we have to modify some original limitations of these methods.

5.4 The Effect of Single Noise

We continue to dive into the impact of single inconsistent labeling noise on the model. Experiments prove that each of the five noise types can seriously affect the model performance and our method can resist any single type of inconsistent labeling noise. Experimental results are shown in Figure 3.

5.5 Ablation Study

As shown in Table 5, we also carry ablation experiments to prove the effectiveness of each phase in our method. w/ BERT means that we use BERT to enhance the character embedding of the Consistency-aware Label Calibration network (CLC). This experiment shows that BERT can very effectively enhance our method. w/o Label Emb means that we remove the label embedding in the CLC. This experiment shows that explicitly introducing label information can help the model understand inconsistent labeling instances better. w/o DLD means that we remove the Dual-stream Label Distillation (DLD) and use two Base-Noise models for the initialization of the network. This experiment shows the effectiveness of DLD. w/o High-D Bias means that we use two High-density Bias models to initialize the CLC. In contrast, w/o Low-D Bias means that we use two Low-density Bias models to initialize the CLC.

6 Conclution and Future Work

We propose the first NER work to study: relying on only a small amount of relatively clean data to denoise the inconsistent labeling instances in large-scale noisy data. To this end, we propose the first denoising framework named DCNER for handling the inconsistent labeling instances problem. Besides, we propose the first Chinese NER benchmark towards the ability of the NER model to resist the inconsistent labeling instances. Finally, detailed experiments have shown that our method consistently and significantly outperforms previous denoising methods on the proposed benchmark. In the future, we hope to continue to explore the inconsistent labeling problem in a broader language and task context.

References

Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. 2006. Chinese named entity recognition with conditional probabilistic models. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 173–176.

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

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer.

Michael A Hedderich and Dietrich Klakow. 2018. Training a neural network in a low-resource setting on automatically annotated noisy data. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 12–18.

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

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. 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)*, pages 729–734.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Lukas Lange, Michael A Hedderich, and Dietrich Klakow. 2019. Feature-dependent confusion matrices for low-resource ner labeling with noisy labels. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3545–3550.

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

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

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

Yinghao Li, Pranav Shetty, Lucas Liu, Chao Zhang, and Le Song. 2021. BERTifying the hidden Markov model for multi-source weakly supervised named entity recognition. 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 6178–6190, Online. Association for Computational Linguistics.

Bingfeng Luo, Yansong Feng, Zheng Wang, Zhanxing Zhu, Songfang Huang, Rui Yan, and Dongyan Zhao. 2017. Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 430–439.

Xuezhe Ma and Eduard Hovy. 2016. 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)*, pages 1064–1074.

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 (CoNLL)*, pages 645–655.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5157–5166.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, et al. Ontonotes release 4.0.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

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.

Qingkai Zeng, Mengxia Yu, Wenhao Yu, Tianwen Jiang, Tim Weninger, and Meng Jiang. 2021. Validating label consistency in ner data annotation. *arXiv preprint arXiv:2101.08698*.

Baoli Zhang, Zhucong Li, Zhen Gan, Yubo Chen, Jing Wan, Kang Liu, Jun Zhao, Shengping Liu, and Yafei Shi. 2021. Croano: A crowd annotation platform for improving label consistency of chinese ner dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 275–282.

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

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. *arXiv preprint arXiv:2104.08656*.