Investigating Annotation Noise for Named Entity Recognition

Yu Zhu[1], Yingchun Ye[1], Mengyang Li[1,2], Ji Zhang[3] and Ou Wu[1]

[1]National Center for Applied Mathematics, Tianjin University, Weijin Road, Tianjin, 300072, China.
[2]Jiuantianxia Inc., Jinguan North Second Street, Beijing, 100102, China.
[3]Zhejiang Lab, Wenyi West Road, Hangzhou, 311100, Zhejiang, China.

Contributing authors: yuzhu@tju.edu.cn; yingchunye@tju.edu.cn; limengyang99@gmail.com; zhangji77@gmail.com; wuou@tju.edu.cn;

Abstract

Recent studies revealed that even the most widely used benchmark dataset still contains more than 5% sample-level annotation noise in Named Entity Recognition (NER). Hence, we investigate annotation noise in terms of noise detection and noise-robust learning. First, considering that noisy labels usually occur when few or vague annotation cues appear in annotated texts and their contexts, an annotation noise detection model is constructed based on self-context contrastive loss. Second, an improved Bayesian neural network (BNN) is presented by adding a learnable systematic deviation term into the label generation processing of classical BNN. In addition, two learning strategies of systematic deviation items based on the output of the noise detection model are proposed. Experimental results of our proposed noise detection model show an improvement of up to 7.44% F1 on CoNLL03 than the existing method. Extensive experiments on two widely used but noisy benchmarks for NER, CoNLL03 and WNUT17 demonstrate that our proposed systematic deviation BNN has the potential to capture systematic annotation mistakes, and it can be extended to other areas with annotation noise.

Keywords: Information extraction, Named entity recognition, Noisy labels, Bayesian neural network

1 Introduction

1.1 Background

Deep neural models have achieved significant success on named entity recognition (NER) task [10]. However, the deep neural models can easily overfit the noisy labels, which negatively affects their generalization ability. Unfortunately, constructing a large dataset with absolutely clean labels is nearly impossible. To identify noisy labels and design a robust learning algorithm have received great attention in machine learning [7] and areas, such as computer vision [15], natural language processing [46], and so on. This problem is also severely serious, even if a compiled according to carefully designed professional instructions for manual annotation NER benchmarks, such as CoNLL03. Wang et al. [41] conducted a pilot study on the detection of noisy annotations for CoNLL03. Their study reveals that about 5.38% of test sentences contain incorrectly annotated entities. Therefore, how to detect noisy labels and design noise-robust learning methods is an urgent challenge for NER models.

Automatically detecting noisy labels in NER is a challenging task because identifying them manually is also actually difficult. The reasons for the difficulty of manually labeling are: (1) that both the entity itself and its context contain limited annotation cues, and (2) the vague annotation instructions. Recently, Wang et al. [41] released a corrected test set (CoNLL++[1]) for CoNLL03 by manually correcting 186 sentences containing label mistakes. However, according to our re-examination in their published test set, their annotations should be further discussed. Table 1 lists several examples. Taking the first sentence as an example, both the original and Wang et al. annotations consider that the words "U.S." and "British" belong to different categories. Nevertheless, their categories should be identical in our judgment. In the second example, we agree with the modification to "Chapman Golf Club" by Wang et al., but "South African" should be labeled as "LOC" because it is not a modifier. In the third sentence, another similar example can be found in the test set "*... and Zulu Chief Mangosuthu Buthelezi's ...*", in which the word "Zulu" is labeled as "MISC" by Wang et al. According to our understanding of the annotation instructions of CoNLL03, no annotation mistakes exist in the third to the sixth samples in the original test set. Furthermore, several entities have not been identified by Wang et al., which are discussed in Section 4.5.2.

In addition, there has been few research efforts on noise-robust NER model, and previous works mostly focuses on weak or distantly supervised [20, 23, 34]. Most of such methods typically depend on additional learning resources, which is different from our research on annotation noise for NER benchmarks themselves. CrossWeigh [41] aforementioned is a pioneering work that denoises NER benchmarks without using extra learning resources. This method is divided into two stages. In the first stage, it partitions the training data into several folds and trains independent NER models to detect potential noisy

[1] https://github.com/pfliu-nlp/Named-Entity-Recognition-NER-Papers/blob/master/ner_dataset.md

Table 1: Original and Wang et al. [41] corrected annotations.

Sentences from CoNLL03 test set	Original	Corrected (Wang et al.)
... **U.S.** and **British** reconnaissance plans had tracked ...	LOC/MISC	LOC/MISC
... **Chapman Golf Club** on Friday (**South African** unless stated):...	LOC/MISC	ORG/MISC
... in South Africa 's volatile **Zulu** heartland, police said on Friday.	MISC	LOC
Britain sets conditions to clear **American** alliance ...	MISC	LOC
And my mandate is also under **Chapter Seven** to operate in ...	O	MISC
The **BILO** stores are located in...	MISC	ORG

labels in each fold. In the second stage, it reduces the weight of samples on which the models disagree to train the final NER model. A method of this kind requires the training dozens of models in the first stage, leading to excessive space and time complexity. In addition, Xiao and Wang [43] adopted Bayesian neural networks (BNN) to quantify the model and data uncertainties for NLP tasks. Zhou and Chen [48] proposed a simple co-regularization framework for entity-centric information extraction, which consists of several neural models with identical structures but different parameter initialization. These models are jointly optimized and regularized to generate similar predictions based on an agreement loss.

1.2 Motivation and contribution

As discussed above, the test set corrected by Wang et al. [41] still has annotation mistakes, and the noise detection method proposed by them also requires the training of a dozen of models using the CrossWeigh strategy. Therefore, a more effective noise detection strategy with a lightweight training load is essential.

In addition, different from the existing noise-robust NER researches, we have an new observation that even on benchmarks systematic mislabels may occur when some annotation instructions are vague or easy to misunderstand. As shown in Table 1, “MISC” entities can be more easily labeled as “LOC”, but not vice versa. This phenomenon is also confirmed by Northcutt et al. [28]. They identify label errors in computer vision, natural language, and audio benchmark datasets and confirm the noise in common benchmark datasets is indeed primarily systematic mislabeling, not just random noise or lack of signal. As far as we know, however, the current noise-robust NER works [41, 43, 48] have not yet explicitly dealt with systematic mislabeling.

In this paper, we propose a novel more space-efficient and time-efficient annotation noise detection model and a BNN with systematic deviation for noise-robust NER to address the above challenges. In particular, our proposed annotation noise detection model consists of two sub-models. The first sub-model focuses on the annotation cues solely from the annotated entities themselves, whereas the second sub-model focuses on the cues solely from their contexts. We design a self-context contrastive loss function, which forces the characteristics to rely merely on the annotated entities themselves or the surrounding texts. We evaluate the proposed noise detection model on CoNLL03. The results show an improvement of up to 7.44% F1. For noise-robust NER, we propose an improved BNN by adding a learnable systematic deviation term into the label generation processing of vanilla BNN. We conduct extensive experiments on two prevalent but noisy benchmarks, CoNLL03 and WNUT17. The results of introducing our proposed BNN with systematic deviation to different NER baseline models indicate consideration of systematic mislabeling in NER task can bring significant performance improvements and the prior information provided by our proposed noise detection model can further improve performance. Our contributions are summarised as follows:

- We design a simple annotation noise detection network that constructs two sub-models based on our novel self-context contrastive loss. One submodel aims to detect annotation cues from annotated entities themselves and the other from contexts.
- We improve the classical BNN by adding a learnable systematic deviation term into the label generation processing.
- We evaluate our annotation noise detection model on the CoNLL03 dataset. Experiments reveal that our model achieves better detection performance, while the training load is lower than the existing CrossWeigh strategy.
- We evaluate the noise-robust NER model based on BNN with systematic deviation on various NER baseline models. Experiments indicate that the introduced systematic deviation does benefit NER model training, and the results of the noise detection model as prior information can further improve the performance of the noise-robust NER model.

2 Related work

2.1 NER

NER is often formalized as a sequence tagging task [29] in which each word in a given text sample is associated with a categorical label.

Before deep learning became prevailing, the most popular sequence tagging methods for NER were hidden Markov models (HMM) [47] and conditional random field (CRF) [32, 44]. The hand-crafted features of each word were directly fed into HMM or CRF to infer tags of each word. As deep learning is widely used, LSTM [9, 42], CNN [19, 39] and Transformer [38] are used to better represent each word instead of hand-crafted features [25]. Bidirectional LSTM-CRF[29] is the first to introduce bidirectional Conditional Random Field layer in deep neural networks for NER. We use this classic network in the experiment. Recently, the vanilla Transformer is reported to perform poorly in the NER task [12], which is also confirmed by Yan et al. [13]. They propose a Transformer-like AdaTrans that incorporated the direction and relative distance aware attention and the un-scaled attention. Most neural named entity classifiers use pre-trained word embeddings, such as Word2vec [26], GloVe [30], and Flair [3]. As pre-trained model BERT [6] achieves state-of-the-art performances in most NLP tasks, BERT is used to replace LSTM and CNN in feature representation [23]. We adopt BERT series model in the experiment. External lexicon knowledge is fused into BERT by Guo et al.[11]. Heterogeneous knowledge from the linguistic, syntactic and semantic perspectives are incorporated to Chinese named entity recognition using graph by Nie et al[27].

Entity boundaries and types are detected simultaneously in common NER approaches. There has been some studies [22, 45] which decouples of boundary detection and named entity type classification and considers boundary detection as a sub-task in NER.

2.2 Noise-robust learning

Robust machine learning mainly deals with learning under noisy or error training labels. Several existing studies attempt to discover noisy labels in training data. Huang et al. [15] designed a novel under-over fitting procedure to detect label noise. The motivation is that the loss of samples with noisy labels is usually reduced at the overfitting stage. Shang et al. [35] explored the noisy factors in relation extraction task and designed an effective noise detection module in their entire network. Liu et al.[24] introduced flipping and class probability and utilized Expectation-Maximization algorithm to solve Gaussian mixture discriminant problem with label noise.

Other studies adopted a weighting strategy to reduce the negative effect of noisy labels. Jenni and Favaro [16] leveraged meta-learning to determine the sample weights. A sample with a lower weight is more likely to have a noisy label. Wang et al. [40] proposed an iterative learning approach, which integrated noisy label detection and discriminative feature learning in a closed loop. A reweighting scheme is used to reduce the negative effect of noisy labels during the loop. Shu et al. [36] proposed a method capable of adaptively learning an explicit weighting function from data directly. In the text classification, Jindal et al. [17] introduced a noise model that models the statistics of the label noise to CNN to better learn the CNN weights and prevent the network from overfitting to erroneous labels.

Currently, there are three researches on noise-label robust NER for benchmark datasets. CrossWeigh[41] partitioned the training data into several folds, trained independent NER models to identify the potential noisy labels, and adjusted the weights of training data accordingly to train the final NER model. Compared with the CrossWeigh method, our proposed method greatly alleviates the waste of space and time when identifying noisy labels. Xiao and Wang[43] adopted Bayesian neural networks to quantify the model and data uncertainties for NER task and sentiment analysis task. Due to the existence of systematic mislabels, we introduce a systematic deviation term into Bayesian neural networks, which can be initialized with the noise detection result or initialized from zero. Our proposed systematic deviation-based Bayesian neural network with zero initialization is as general as the traditional Bayesian neural network. Zhou and Chen[48] proposed a co-regularization framework consisting of several neural networks with the same structure but different initialization, which are jointly optimized for the task-specific losses and regularized to generate similar predictions based on an agreement loss.

The above studies suffer from three limitations: (1) While noise-robust learning in the image field has received much attention, the text field has not. According to our research, there are only three works mentioned above on noise-robust NER including CrossWeigh [41], BNN [43], and co-regularization [48]. (2) At present, most of the works focus on the study of sample-level noise instead of entity-level noise that is the type of noise in the NER task. (3) These studies mainly assume that the involved training set is clean, and added to simulated random noise.

2.3 BNN

Recently, some researchers viewed noisy labels as a type of data uncertainty for training data. BNN is then used in the entire framework to deal with noisy labels. BNN assumes two kinds of uncertainty in a standard supervised learning approach [8]. The first refers to the epistemic or model uncertainty, which assumes that the model parameters are distributed over on a prior distribution, such as Gaussian. Dropout variational inference is a practical approach for approximate inference when considering model uncertainty. The second refers to aleatoric or data uncertainty, which assumes that the predicted output (e.g., y) is distributed over on a distribution parameterized by the feature representation of the input (e.g., x). Taking the regression task as an example, the distribution of the predicted output y conforms to the following Gaussian distribution:

$$y \sim \mathcal{N}(f(x, W), \sigma(x)), \tag{1}$$

where $f(x, W)$ is the output of employed deep neural network, and $\sigma(x)$ is the corresponding standard deviation term to be learned for x, that is data uncertainty. Accordingly, the loss function can be written as:

$$\mathcal{L}(W) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i | f(x_i, W), \sigma(x_i)) = -\frac{1}{2N}\sum_{i=1}^{N} [\|\frac{y_i - f(x_i, W)}{\sigma(x_i)}| + \log \sigma(x_i)^2], \tag{2}$$

where $1/\sigma(x_i)$ can be viewed as a weight of sample x_i. As the value of $1/\sigma(x_i)$ decreases, y_i is more likely a noisy-label. From a cognitive perspective, the data with high confidence (low noise) should be paid more attention, so this parameter is trained to capture complex noise patterns in the data.

BNN has been widely studied in computer vision [4, 18], text mining [43], intelligence management [14], and others. In Section 3.2, BNN with data uncertainty in classification is briefly reviewed in a random deviation view.

3 Methodology

This section first introduces our proposed noise detection method in Section 3.1, then introduces an improved BNN by adding a systematic deviation term into the label generation processing of classical BNN in Section 3.2.

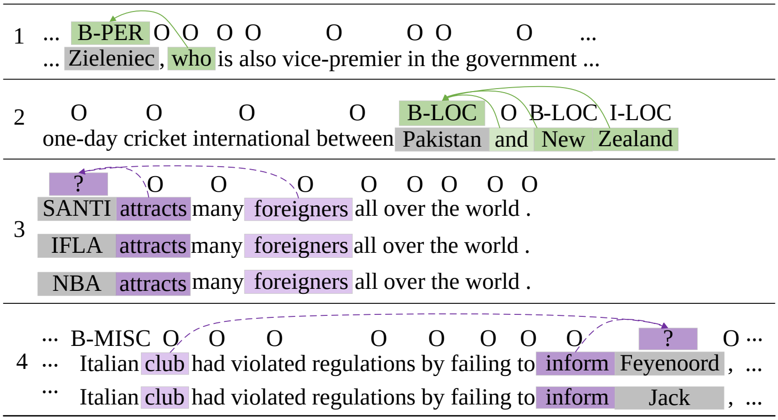

1 ... B-PER O O O O O O O O ...
... Zieleniec , who is also vice-premier in the government ...
2 O O O O B-LOC O B-LOC I-LOC
one-day cricket international between Pakistan and New Zealand
3 ? O O O O O O O O
SANTI attracts many foreigners all over the world .
IFLA attracts many foreigners all over the world .
NBA attracts many foreigners all over the world .
4 ··· B-MISC O O O O O O O O ? O ···
··· Italian club had violated regulations by failing to inform Feyenoord , ...
··· Italian club had violated regulations by failing to inform Jack , ...

Fig. 1: Examples of contexts as annotation cues. Green shade: clear cue information. Purple shade: vague cue information.

3.1 Self-context contrastive loss-based annotation noise detection

Training and testing data or even validation data may contain label noise. To detect annotation noise in NER, Wang et al. [41] proposed a CrossWeigh strategy. CrossWeigh repeatedly divides the training data into ten folds. Each time, a model is trained from nine folds and the model is run on the rest fold. The prediction errors for each sample in the rest fold are recorded. After repeating thirty times (three cross-validation rounds), each training sample receives three predictions. The number of error predictions for each sample is used as the error indication value, which is calculated as follows:

$$r_i = \varepsilon^{k_i}, \tag{3}$$

where ε is set as 0.7 and k_i is the times of error predictions for the i-th sample during the cross-validation above. As k_i increases, r_i decreases, and thus, the sample label is more likely to be noisy.

The bottleneck of the CrossWeigh strategy is that thirty deep learning models are necessary to train, indicating that space and time consumption is expensive. We observed that noisy annotations usually occur for samples that are difficult to judge for humans. Alternatively, easy-labeling samples are less likely to have error annotations. Motivated by this intuition, we attempt to construct a simple model that solely relies on annotated entities or their contexts.

Fig. 1 shows some examples of contexts as annotation cues. For example, in the first sentence "*Zieleniec, who is also vice-premier in the government ...*", "*Zieleniec*" is easy to annotate because its next word "*who*" is an evident cue about "PERSON". In the same vein, "*and New Zealand*" in the second sentence also provides sufficient annotation cues. Nevertheless, in the third group of sentences "*SANTI attracts many foreigners all over the world*", "*SANTI*" is difficult to annotate because the word "*SANTI*" itself and its surrounding words do not contain evident cues (a game, an organization, or a location?) for human annotation. Obviously, "*IFLA*" (organization) and "*NBA*" (game) in the following two sentences are also feasible. The last group of sentences also lack clear annotation cues from contexts. The object of "*inform*" can be either a person or an organization, and the preceding "*club*" does not have a clear tendency. As a result, the possibility that its label is erroneous becomes higher than that of the word "*Zieleniec*" in the first sentence. The fourth sentence in Table 1 also indicates that if labeled entities and their surrounding texts have unclear or vague cues, then their labels have higher possibilities to be noisy labels.

Our noise detection model consists of two sub-models. The first sub-model focuses on the annotation cues solely from the annotated entities themselves, whereas the second sub-model focuses on the cues solely from their contexts (surrounding texts).

We assume that a sentence is represented by $s_i = \{s_{i,1}, \cdots, s_{i,t}, \cdots, s_{i,t+l}, \cdots, s_{i,L_i}\}$, where $s_{i,t}, \cdots, s_{i,t+l}$ are the annotated entity texts and L_i is the sentence length. Both sub-models will learn to predict the label of $s_{i,t}, \cdots, s_{i,t+l}$. Let $y_{i,t}$ be its true label (a one-hot vector), $ya^{(1)}_{i,t}$ be the predicted label by the first sub-model, and $ys^{(2)}_{i,t}$ be the predicted label by the second sub-model. Both sub-models share the same network structure as shown in Fig. 2. The input of the network is a sub-sentence "$s_{i,t-5} \cdots s_{i,t+l+5}$", which contains the annotated entity. Alternatively, the window size is a hyper-parameter and is set to eleven.

Taking the first sub-model as an example, at first, all the word embeddings in the input sub-sentence are inputted into an encoder (e.g., BiLSTM, Transformer [38]), and the corresponding hidden vectors are obtained:

$$h^{(1)}_{i,j} = Encoder(s_{i,j}), j = t-5, \cdots, t+l+5. \tag{4}$$

Subsequently, the hidden vectors $(h^{(1)}_{i,t}, \cdots, h^{(1)}_{i,t+l})$ for the annotated entity are fed into the attention and softmax layers as follows:

$$\begin{aligned} \alpha^{(1)}_{i,j} &= Attention(h^{(1)}_{i,j}), j = t, \cdots, t+l, \\ ya^{(1)}_{i,t} &= softmax[W^{(1)}_a \sum_{j=t}^{t+l} \alpha^{(1)}_{i,j} h^{(1)}_{i,j}], \end{aligned} \tag{5}$$

where $W^{(1)}_a$ is the dense layer parameter, and $ya^{(1)}_{i,t}$ is the prediction solely based on the annotated entity from the first sub-model. Similarly, the prediction solely based on surrounding texts $ys^{(1)}_{i,t}$ can also be calculated in the same manner.

We design a self-context contrastive loss function, which forces the characteristics to rely merely on the annotated entities themselves or the surrounding

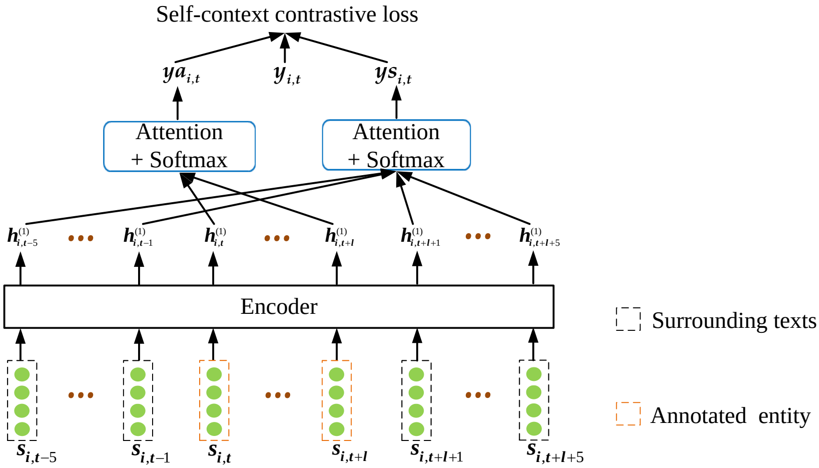


Fig. 2: Network structure used in our annotation noise detection.

texts. The self-context contrastive loss of the first sub-model is as follows:

$$\mathcal{CL}_{(1)} = \sum CE(ya^{(1)}_{i,t}, y_{i,t}) - \sum CE(ys^{(1)}_{i,t}, y_{i,t}), \tag{6}$$

where CE is the cross-entropy loss in this study. Eq.(6) indicates that the learning goal aims to train a sub-model, which relies merely on the annotated entities themselves.

Similarly, the self-context contrastive loss of the second sub-model is defined as follows:

$$\mathcal{CL}_{(2)} = \sum CE(ys^{(2)}_{i,t}, y_{i,t}) - \sum CE(ya^{(2)}_{i,t}, y_{i,t}). \tag{7}$$

The learning goal of the second sub-model aims to train a model that relies merely on the surrounding texts.

Once the two sub-models above are trained, their output $ya^{(1)}_{i,t}$ and $ys^{(2)}_{i,t}$ are fused to generate the final output. Let c be the true category of the current annotated entity. The final output of the entire model is as follows:

$$v_{i,t} = (1 - ya^{(1)}_{i,t,c})(1 - ys^{(2)}_{i,t,c}). \tag{8}$$

If both sub-models predict the correct label c, then v will be small; if both predict the incorrect one, then v will be large. v can be viewed as the score reflecting on what extent an annotation is a noise. If the score is high, then more likely, the annotation is an error. This score will be used in the improved BNN framework in Section 3.2 to further improve the effect of the noise-robust NER model.

3.2 Improved BNN for NER

Recently, experiments by Xiao and Wang [43] confirmed that only data uncertainty rather than model uncertainty matters in NER. Noisy label can be seen as a type of data uncertainty. Recent progress in BNN has made quantifying uncertainty and training more effective machine learning models possible [21]. BNN provides a theoretical tool to model the relationships among data uncertainty, model uncertainty, and deep neural networks. However, all existing BNNs assume that the noise obeys the Gaussian distribution with zero mean. We assume that the noise obeys the Gaussian distribution with non-zero mean. This non-zero mean is called systematic deviation. The reason lies in that annotators may systematically misunderstand the official annotation guidelines, leading to the occurrence of systematic deviation.

BNN is firstly briefly reviewed in a standard classification manner. A classification training set is denoted as $D = \{x_i, y_i\}_{i=1}^{N}$, where x_i is the i-th training sample and y_i is the target category. Let DNN be a deep neural network for expected logit representation, which is parameterized by Θ_f. The expected logit vector is fixed for each training sample once Θ_f is fixed. Therefore, the expected logit vector is represented as follows:

$$\mu(x_i) = DNN(x_i, \Theta_f). \tag{9}$$

BNN views the prediction as a label generation process based on the above expected logit vector. Specifically, BNN assumes that a random deviation (e_i) for $\mu(x_i)$ exists, which conforms to a Gaussian distribution with zero mean as:

$$e_i \sim \mathcal{N}(0;\ \sigma(x_i)), \tag{10}$$

where $\sigma(x_i)$ is the standard deviation for the random deviation for x_i in the logit space. Subsequently, a logit vector u_i is generated for x_i as follows:

$$u_i \sim \mu(x_i) + \mathcal{N}(0;\ \sigma(x_i)). \tag{11}$$

Given that u_i is the final logit vector for x_i, as the value of the standard deviation $\sigma(x_i)$ increases, the possibility that the difference between u_i and $\mu(x_i)$ is also high. With u_i, the final predicted category y_i is

$$\begin{aligned} p_i &= \mathrm{softmax}(u_i), \\ y_i &\sim categorical(p_i), \end{aligned} \tag{12}$$

where p_i is the probability over categories of x_i and y_i is the final predicted label.

During training, the above label generation process repeats many times for each training sample. K logit vectors are sampled, and the Monte Carlo approximation for predicted distribution is calculated as follows:

$$u_i^{(k)} \sim \mu(x_i) + \mathcal{N}(0;\ \sigma(x_i)), \tag{13}$$

where $\mu(x_i)$ and $\sigma(x_i)$ as mean and standard deviation functions that maps input x_i to the logit space. $u_i^{(k)}$ denotes the k-th logit vector sampled. Thereafter, it is transformed into probabilities using softmax operation. The specific formula is as follows:

$$\overline{p_i} = \frac{1}{K}\sum_{k=1}^{K} softmax(u_i^{(k)}), \tag{14}$$

where $\overline{p_i}$ is the average of the K softmax distributions calculated by the K sampled logit vectors. $\overline{p_i}$ is then used as the final predicted distribution. Consequently, the cross-entropy loss can be calculated as follows:

$$\mathcal{L}_{BNN}{=}\frac{1}{N}\sum_{i=1}^{N} CE(\overline{p_i}, y_i^{true}). \tag{15}$$

To apply the above label generation process into NER, Xiao and Wang [43] adopted an intuitive way that directly discarded the CRF layer in the standard NER network[2].

As mentioned above, in practical applications, noise deviation may be systematic rather than random. In NER annotation, systematic deviations may occur when some annotation instructions are vague or easy to misunderstand. In this study, a systematic deviation term is introduced into BNN. Specifically, we can assume that e_i conforms to a Gaussian distribution as:

$$e_i \sim \mathcal{N}\left(\Delta\mu_i;\ \sigma\left(x_i\right)\right), \tag{16}$$

where $\Delta\mu_i$ is the systematic deviation term for x_i.

Subsequently, given u_i and the above Gaussian distribution of e_i, a logit vector u_i is generated for x_i as follows:

$$u_i \sim \mu\left(x_i\right) + \mathcal{N}\left(\Delta\mu_i;\ \sigma\left(x_i\right)\right). \tag{17}$$

The sampled logit is then fed into the CRF layer/ softmax layer to infer the final label. The loss function can be written as

$$\mathcal{L}_{SdBNN}{=}\frac{1}{N}\sum_{i=1}^{N} CE(\overline{p_i'}, y_i^{true}) + \lambda\Delta{\mu_i}_2^2, \tag{18}$$

where $\overline{p_i'}$ is similar with $\overline{p_i}$ in Eq.(14) in which $u_i^{(k)}$ is sampled from Eq.(17); the regularized term is added to prevent the model from estimating meaningless systematic deviations for all input samples. Fig. 3 shows the main difference between conventional BNNs and our systematic deviation-based BNN (SdBNN). Mathematically, the difference between conventional BNNs and our improved BNN is the prior assumption on the expectation of the noise distribution. Fig.

[2] In our experiments, BiLSTM-CRF is not inferior to BiLSTM-BNN. Therefore, we conjectured that CRF can alleviate partial negative effects of random noise modeled by existing BNN.

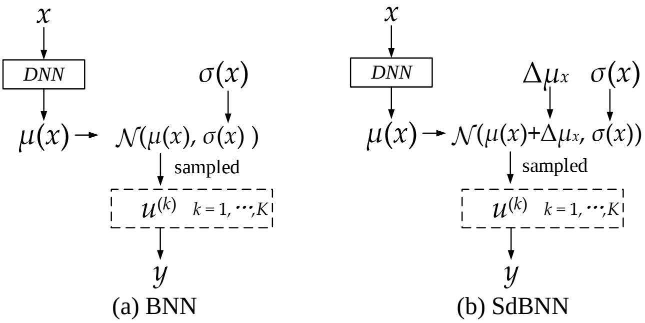


Fig. 3: Data (label) generation processes of BNN and SdBNN. Only data uncertainty is considered in this study.

4 shows the influence of systematic deviation on prediction. The black solid line in the figure represents the prediction result considering only random deviation. Assuming that the random deviation (variance) at sample x_i is a black dashed line, when x_i has a systematic deviation (bias), the actual deviation of sample x_i should be a blue dashed line. It can be seen that this will have a greater impact on the prediction of sample x_i. In many real applications (including NER), systematic deviation assumption used in our BNN is more reasonable.

Existing BNN assumes that the noisy labels appear randomly at each position of a sentence. Alternatively, no prior knowledge utilizes both the positions and quantities of noisy labels. Our proposed annotation noise detection model provides cues for possible positions and quantities of noisy labels. For

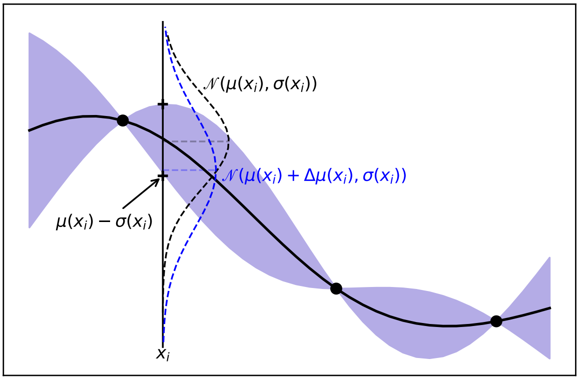


Fig. 4: The influence of systematic deviation on prediction. The black solid line represents the prediction result, and the purple area represents the confidence interval. The black and blue dashed lines represent random deviation and actual deviation (with systematic deviation), respectively.

each position, we can obtain a score (v in Eq. (8)) and a possible noisy indication vector (η), which is defined as follows:

$$\eta_{i,t} = Concat(ua_{i,t}^{(1)}, us_{i,t}^{(2)}), \tag{19}$$

where $ua_{i,t}^{(1)}$ and $us_{i,t}^{(2)}$ represent the logit vectors output by the two sub-models for annotated entity texts and their contexts, respectively. Consequently, we define the following two strategies to calculate the systematic deviation.

Strategy 1: Noise indication vector-based strategy.

$$\Delta\mu_{i,t} = W_\mu \eta_{i,t}, \tag{20}$$

where W_μ is the parameter. That is, $\Delta\mu_{i,t}$ is transformed from $\eta_{i,t}$.

Strategy 2: The combination of noise indication vector and score.

$$\Delta\mu_{i,t} = W_\mu \eta_{i,t}, \quad \textbf{if} \quad v_{i,t} > v_{threshold}. \tag{21}$$

In this strategy, only the $\Delta\mu_{i,t}$ of words whose corresponding $v_{i,t}$ values are larger than the $v_{threshold}$ are learned. The hyper-parameter $v_{threshold}$ is searched in 0.2, 0.4, 0.6, 0.8.

4 Experiments

We evaluate the proposed annotation noise detection and noise-robust NER models[3].

4.1 Datasets

Two benchmark datasets and one revision dataset are used, namely, CoNLL03 [37], WNUT17 [5] and CoNLL++ [41]. The standard train/dev/test splits follow the existing studies [2, 31, 41].

CoNLL03/CoNLL++. CoNLL03 (English) is one of the most widely used NER datasets, which is taken from Reuters news reports between August 1996 and August 1997. It contains four linguistic entity types: person (PER), location (LOC), organization (ORG), and miscellaneous names (MISC). Table 2 presents dataset statistics. Other detailed information such as annotation instructions can refer to the official website[4]. As described in Section 1, CoNLL++ is a revision of the CoNLL03 by Wang et al. [41]. They manually correct 186 sentences with mistake labels for test set. Table 3 shows the details of CoNLL03 and CoNLL++ test set labels.

WNUT17. WNUT17 is the dataset from the Shared task at the Workshop on Noisy User-generated Text 2017 with 6 entities: Person, Location, Corporation, Product, Creative work, and Group. The dataset takes from three

[3] Our code is available at https://github.com/ruby-yu-zhu/Annotation_Noise_NER

[4] https://www.clips.uantwerpen.be/conll2003/ner/

Table 2: Details of datasets

Dataset	Domain	Class	Type	Train	Dev	Test	Entities frequency
CoNLL03	News	4	Sentence	14987	3466	3684	11.64%
			Token	203621	51362	46435	
			Entity	23499	5942	5648	
WNUT17	Social Media	6	Sentence	3394	1009	1287	3.82%
			Token	62730	15733	23394	
			Entity	1975	836	1079	

different sources: Reddit, Twitter, YouTube, and StackExchange comments. Dataset statistics are listed in Table 2.

WNUT17 is more difficult to label than CoNLL03. The main reasons are as follows: 1) Lack of capitalization, because social media users unlike news editors tend to arbitrarily alter the character casing. 2) The classes are more heterogeneous. The target classes on the WNUT17 cover the CoNLL03 classes plus fine-grained classes such as Creative Work (e.g., movie titles, T.V. shows, etc.), Group (e.g., sports teams, music bands, etc.), and Product [1]. This may also be the reason why there is currently no research work to evaluate the label quality of the WNUT17 dataset.

Table 3: Details of CoNLL03 and CoNLL++ test set labels

Dataset	O	PER	LOC	ORG	MISC
CoNLL03	40787	1617	1668	1661	702
CoNLL++	40733	1618	1646	1715	723

4.2 Baseline models

For annotation noise detection, we compare our proposed method with the mistake estimation module of CrossWeigh [41], which detects the potential label mistakes through a cross checking process.

For noise-robust NER, the baseline models are as follows:

- **BiLSTM-CRF [29].** It is the bidirectional LSTM network with a CRF layer.
- **BiLSTM-CRF-CrossWeigh [41].** It is the noise-robust method particularly for NER. It assigns lower loss weights for possibly noisy samples.
- **BiLSTM-BNN [43].** It is the first method that utilizes BNN for NER. In this method, the CRF layer is discarded in the classical BiLSTM-CRF structure.
- **BiLSTM-CRF-BNN.** This method is based on the standard BNN (only random deviation is considered). In this method, CRF is not discarded.
- **BERT$_{\text{BASE/LARGE}}$[6].** It is a pre-trained language representation model based on Transformer.
- **BERT$_{\text{BASE/LARGE}}$-CrossWeigh.** This method is BERT$_{\text{BASE/LARGE}}$-based model with CrossWeigh framework.

- **BERT$_{\text{BASE/LARGE}}$-CR** [48]**.** This method is BERT$_{\text{BASE/LARGE}}$-based model with co-regularization framework.
- **BERT$_{\text{BASE/LARGE}}$-BNN.** This method is BERT$_{\text{BASE/LARGE}}$-based model with the standard BNN (only random deviation is considered).
- **DistilBERT**[33]**.** The model distilled from the checkpoint of BERT model.
- **DistilBERT-BNN.** This method is DistilBERT-based model with the standard BNN (only random deviation is considered).

4.3 Experimental Settings

For annotation noise detection, GloVe is used and the dimension is 1024. The dimensions of hidden and attention layers are set as 200 and 400, respectively. The learning rate is 0.001. The dropout rate is 0.5. $v_{threshold}$ is set as 0.4.

For BiLSTM-CRF-based model on noise-robust NER, both the pre-trained GloVe (300d)[5] and Flair[6] embeddings are used. All methods based on the BiLSTM-CRF network use the open-source FLAIR framework, which implements the standard BiLSTM-CRF sequence labeling architecture [29] and supports pre-trained various contextual string embeddings. For BERT-based model on noise-robust NER, the pre-trained language model BERT is used as the primary baseline algorithm. We use the `bert-base-cased`, `bert-large-cased` and `distilbert-base-cased` provided by HuggingFace[7]. The DistilBERT model distilled from the BERT model `bert-base-cased` checkpoint. The details of the `bert-base-cased` and `bert-large-cased` models are shown in Table 4. For hyper-parameters and optimization choices, we mostly follow Devlin et al. [6].

Table 4: Details of pre-trained model based on BERT

Pre-trained model	Layers	Hidden nodes	Heads	Parameters	Case-sensitive
`bert-base-cased`	12	768	12	110M	✓
`bert-large-cased`	24	1024	16	340M	✓
`distilbert-base-cased`	6	768	12	65M	✓

4.4 Evaluation

To evaluate our models, we report the standard metrics for noise detection: micro-averaged precision, recall, and F1-score. We use Exact-match F1[8] for the noise-robust NER model. The model that achieves the best performance on the development set is evaluated on the test set with the F1 score. The specific calculation method is as follows:

[5] https://nlp.stanford.edu/projects/glove/
[6] https://github.com/flairNLP/flair
[7] https://github.com/huggingface/transformers
[8] SeqEval package were used to calculate F1 metric.

$$P = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i}, \tag{22}$$

$$R = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i}, \tag{23}$$

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{24}$$

where N is the total number of samples. TP_i represents the number of correctly identified entities in the i-th sample. FP_i represents the number of misidentified entities in the i-th sample. FN_i represents the number of unrecognized entities in the i-th sample.

4.5 Results and analyses

4.5.1 Results on annotation noise detection

The sample-level F1 value the 186 manual corrections of Wang et al. [41] is the evaluation metric on CoNLL03. Table 5 lists the comparison results. The results of CrossWeigh are directly from the corresponding paper.

Table 5: Noise detection performance comparison on the manual correction set of Wang et al. [41].

Method	Precision (%)	Recall (%)	F1 (%)
Ours	32.48	75.27	**45.38**
CrossWeigh	25.13	77.42	37.94

The F1 score of our method is much higher than that achieved by CrossWeigh. The recall of our model is slightly lower than that of CrossWeigh. A possible reason is that some correct annotations in the original CoNLL03 test set are incorrectly judged by Wang et al. [41] as shown in Table 1. Furthermore, even for the incorrect annotations in the original test set, their manual corrections still contain errors according to our understanding of the annotation instructions.

4.5.2 Case study for annotation noise detection

Table 6 lists five examples. Taking the first sample as an example, “Mediterranean” is a “MISC” entity in the sentence because “Mediterranean” is an adjective. According to the annotation instructions, words derived from a word which is location should be labeled as “MISC”. The following two examples are similar to the above. In the fourth sample, according to our understanding, the words “Babri mosque” should be labeled as a whole. In the fifth example, the word “West” should be labeled as “ORG” because in the annotation instructions, “ORG” contains “political unions of countries”. Indeed, we are unsure about our corrections. For example, “Babri mosque” may be a “LOC”

Table 6: Error annotations in original CoNLL03.

Sentences from CoNLL03 test set	Original	Corrected	Our model
Mediterranean oil products were steady ...	O	LOC	MISC
Midcontinent prices were similarly lower ...	O	LOC	MISC
... Indonesia's **Busang** vast gold deposit.	ORG	LOC	MISC
... in remembrance of the **Babri mosque** ...	O	MISC (Babri)	MISC (Babri mosque)
... between the **West** and developing countries ...	O	LOC	ORG

Table 7: Missed annotations in original and corrected CoNLL03.

Sentences from CoNLL03 test set	Original/Corrected	Our model
NORDIC SKIING-WORLD CUP	O	MISC
NFL **AMERICAN** FOOTBALL-STANDINGS AFTER ...	O	MISC
... in distillate-hungry **Northeastern** markets...	O	MISC
Two ships loaded on the **East Coast**, three waited to load...	O	LOC
... as milder weather moved into the **Southwest**.	O	LOC

entity. Therefore, this study does not intend to provide a new correction set because the official annotation instructions are still vague for us in some places. Compiling an absolutely accurate NER training corpus is quite difficult.

In addition, our noise detection model locates missed annotated entities that are not detected by CrossWeigh. Table 7 lists some examples. Taking the second sample as an example, the word "AMERICAN" should be labeled as "MISC", whereas this word is labeled "O" in the original set. Among the missed annotations, "MISC" and "LOC" entities occupy the most. The reason is that the annotation instructions for "MISC" and "LOC" are highly similar in many cases.

4.5.3 Results on noise-robust NER

As describe in Section 3.2, we train and evaluate six main model variations:

- **BiLSTM-CRF-SdBNN.** This method is the classical method BiLSTM-CRF with our systematic deviation-based BNN.
- **BiLSTM-CRF-SdBNN1.** This method is the classical method with our proposed SdBNN when Strategy 1 (Eq. (20)) is used.
- **BiLSTM-CRF-SdBNN2.** This method is the classical method with our proposed SdBNN when Strategy 2 (Eq. (21)) is used.
- **BERT$_{\mathbf{BASE/LARGE}}$-SdBNN.** This method is standard BERT$_{\mathrm{BASE/LARGE}}$-based model with our systematic deviation-based BNN.
- **BERT$_{\mathbf{BASE/LARGE}}$-SdBNN1.** This method is standard BERT$_{\mathrm{BASE/LARGE}}$-based model with our proposed SdBNN when Strategy 1 (Eq.(20)) is used.
- **BERT$_{\mathbf{BASE/LARGE}}$-SdBNN2.** This method is standard BERT$_{\mathrm{BASE/LARGE}}$-based model with our proposed SdBNN when Strategy 2 (Eq. (21)) is used.

Table 8: The F1 scores (%) of competing NER methods (GloVe and Flair). The best results are in bold.

Methods	CoNLL03		CoNLL++		WNUT17	
	GloVe	Flair	GloVe	Flair	GloVe	Flair
BiLSTM-CRF [29]	90.11	92.98	91.10	93.61	37.38	46.16
BiLSTM-CRF-CrossWeigh [41]	90.28	93.02	91.19	93.81	37.28	46.46
BiLSTM-BNN [43]	87.57	92.09	88.29	92.07	31.58	43.67
BiLSTM-CRF-BNN	90.16	92.84	91.09	93.84	37.39	46.88
BiLSTM-CRF-SdBNN	90.38	93.05	91.33	93.93	37.49	**47.64**
BiLSTM-CRF-SdBNN1	90.42	93.13	91.41	**94.11**	37.48	46.68
BiLSTM-CRF-SdBNN2	**90.49**	**93.14**	**91.49**	93.84	**37.50**	47.00

- **DistilBERT-SdBNN.** This method is DistilBERT-based model with our systematic deviation-based BNN.
- **DistilBERT-SdBNN1.** This method is DistilBERT-based model with our proposed SdBNN when Strategy 1 (Eq. (20)) is used.
- **DistilBERT-SdBNN2.** This method is DistilBERT-based model with our proposed SdBNN when Strategy 2 (Eq. (21)) is used.

The first three methods are based on GloVe or Flair. In BiLSTM-CRF-SdBNN\SdBNN1\SdBNN2, the parameter λ is searched in $\{0.5, 1, 2, 5\}$.

The overall competing results on the two benchmark datasets (CoNLL03 and WNUT17) when GloVe and Flair embeddings are used are listed in Table 8. Overall, all the noise-robust methods (except BiLSTM-BNN) outperform the classical network BiLSTM-CRF. The poor performance of BiLSM-BNN indicates that enhanced character-level contextualized representations of the CRF are indeed important. Our three SdBNN-based methods achieve better results than CrossWeigh and BNN. When GloVe is used, BiLSTM-CRF-SdBNN2 yields the highest F1 score. Nevertheless, when Flair is used, the method BiLSTM-CRF-SdBNN achieves the highest F1 score on WNUT17, indicating that prior information according to the Strategies 1 and 2 may be not beneficial in some situations. In general, SdBNN-based methods with the noise detection results as prior information work better, but SdBNN without prior information is a new general BNN.

We also test the above models on CoNLL++. The results of methods with the two embeddings (GloVe and Flair) on this corrected test set, too, are shown in Table 8. The performances of nearly all involved competing methods are increased than on CoNLL03. The SdBNN-based methods still perform better than others.

To verify the effectiveness of our proposed BNN with systematic deviation, we further incorporate pre-trained language model $\text{BERT}_{\text{BASE/LARGE}}$ and DistilBERT. In order to reduce the impact of randomness, we ran all of our experiments three times, and an average F1 score is reported. For some hyper-parameters of BERT-based models, we mostly follow Devlin

Table 9: The F1 scores (%) of competing NER methods (BERT$_{\text{BASE}}$, BERT$_{\text{LARGE}}$ and DistilBERT). The results with * reported in Zhou et al.[48]. The best results are in bold.

Methods	CoNLL03	CoNLL03++	WNUT17
BERT$_{\text{BASE}}$ [6]	91.96*	92.91*	46.67
BERT$_{\text{BASE}}$-CrossWeigh	92.15*	93.03*	-
BERT$_{\text{BASE}}$-CR [48]	92.53*	93.48*	-
BERT$_{\text{BASE}}$-BNN	92.42	93.65	46.71
BERT$_{\text{BASE}}$-SdBNN	92.98	94.12	47.47
BERT$_{\text{BASE}}$-SdBNN1	**93.21**	94.30	47.16
BERT$_{\text{BASE}}$-SdBNN2	93.15	**94.48**	**47.79**
BERT$_{\text{LARGE}}$[6]	92.24*	93.22*	49.08
BERT$_{\text{LARGE}}$-CrossWeigh	92.49*	93.61*	-
BERT$_{\text{LARGE}}$-CR[48]	92.82*	94.04*	-
BERT$_{\text{LARGE}}$-BNN	93.03	94.44	49.24
BERT$_{\text{LARGE}}$-SdBNN	93.64	95.03	51.21
BERT$_{\text{LARGE}}$-SdBNN1	93.72	95.39	50.97
BERT$_{\text{LARGE}}$-SdBNN2	**93.78**	**95.46**	**51.26**
DistilBERT [33]	89.84	90.92	44.49
DistilBERT-BNN	90.28	91.51	44.83
DistilBERT-SdBNN	90.95	92.03	45.56
DistilBERT-SdBNN1	90.97	92.34	45.58
DistilBERT-SdBNN2	**91.01**	**92.44**	**45.86**

et al. [6]. In BERT$_{\text{BASE/LARGE}}$-SdBNN\SdBNN1\SdBNN2 and DistilBERT-SdBNN\SdBNN1\SdBNN2, the parameter λ is searched in {0.25, 0.5, 1, 2, 5}.

All experimental results of BERT$_{\text{BASE/LARGE}}$ and DistilBERT as baseline model on the two benchmark datasets (CoNLL03 and WNUT17) are shown in Table 9. Two main observations are obtained. Firstly, our SdBNN-based methods achieve better results than BERT. For example, our model BERT$_{\text{LARGE}}$-SdBNN2 incorporating systematic deviation to improve over the BERT$_{\text{LARGE}}$ model by 1.54% and 2.18% on ConLL03 and WNUT17, respectively. Secondly, our model is more suitable for datasets that are difficult to label. Compared with corresponding BNN models, our best model based on GloVe, Flair, BERT$_{\text{BASE}}$, BERT$_{\text{LARGE}}$ and DistilBERT increase by 0.33%, 0.3%, 0.79%, 0.75% and 0.73% on CoNLL03, and increase by 0.11%, 0.76%, 1.08%, 2.02% and 1.03% on WNUT17, respectively. Our models improves significantly on WNUT17 that are more difficult to label. We tend to attribute the improvements brought by our models as follows: the WNUT17 dataset collected from social media is more difficult to label than the CoNLL03 dataset collected from the news. Our model can correct the larger systematical deviation caused by the increasing labeling difficulty.

Fig.5 shows the model performance for different entity types on CoNLL03 dataset in BERT$_{\text{BASE}}$-based model. It can been seen that our proposed three models systematical deviation-based models (SdBNN, SdBNN1 and SdBNN2) achieve large improvements in MISC and ORG classes. This is consistent with

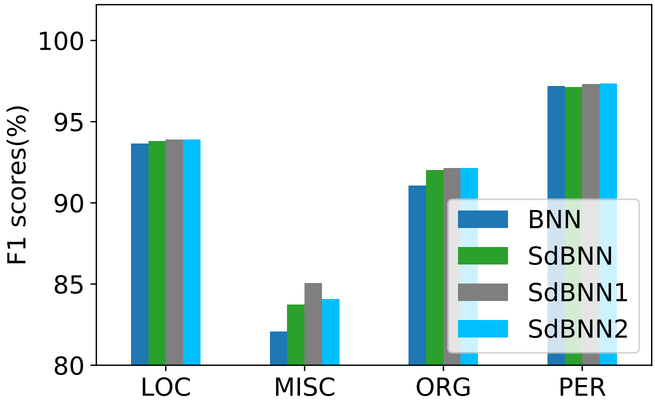


Fig. 5: The impact of structural changes on various entity types in BERT$_{\text{BASE}}$-based model

our observation in Section 4.5.2 that MISC and ORG classes are prone to systematic mislabels.

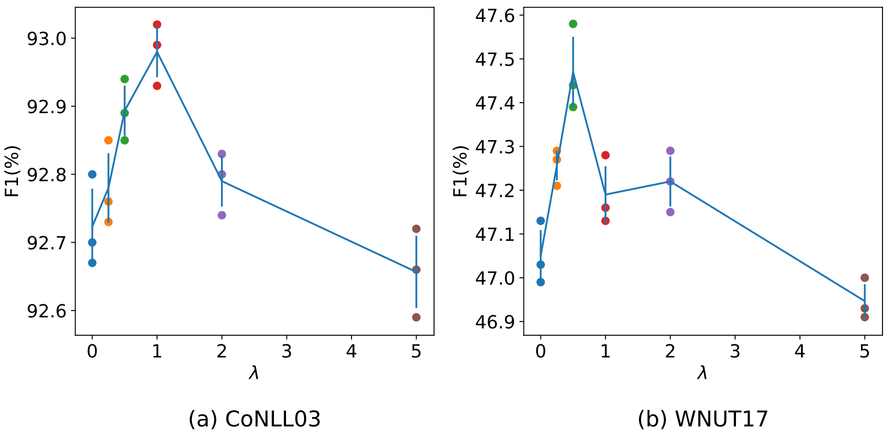
Table 9 also shows the test results of the competing methods on CoNLL++. The performances of almost all involved competing methods are increased than on CoNLL03. Our proposed BERT$_{\text{LARGE}}$-SdBNN2 achieves the best result on this corrected test set. Compared with the methods based on GloVe and Flair, the performances are greatly improved.

In summary, the consistent improvements to the baseline (BiLSTM-CRF and BERT) show that our method is optional in NER, especially for noisy texts that are difficult to label by human experts. In addition, compared with the BNN-based models that only consider the random deviation, the improvements of our models indicate that the introduced systematic deviation is indeed useful.

4.5.4 Impact of the regularization coefficient λ

λ is the regularization coefficient in Eq.(18). L2 regularization is often referred to as weight decay, which can prevent the model from estimating meaningless systematic deviation terms for all input samples. The larger the λ, the greater the inhibition effect on the systematic deviation terms. To verify the influence of λ on the experimental results of the model, we adjusted λ in the range of $\{0, 0.25, 0.5, 1, 2, 5\}$. The results based on the BERT$_{\text{BASE}}$-SdBNN model on the CoNLL03 and WNUT17 datasets are shown in Fig.6. Fig.6 (a) shows the experimental results under different λ values on the CoNLL03 dataset. It can be seen that when λ is less than 1, the model performance gradually increases with the increase of λ. When λ is greater than 1, the model performance shows a downward trend. The experimental results on the WNUT17 dataset (Fig. 6 (b)) also reflect a similar trend.

When the value of λ is small, increasing λ can suppress the system deviation term and prevent over-fitting. When the value of λ is large, the increase of λ



(a) CoNLL03

(b) WNUT17

Fig. 6: Experimental results with different λ based on the BERT$_{\text{BASE}}$-SdBNN model

causes the system deviation term to fail to function, which reduces the model's ability to model the systematic mislabels.

5 Discussion

5.1 Implications of annotation noise detection

The time consumption of our proposed annotation noise detection model is significantly lower than compared model (CrossWeigh), which requires multiple cross checking. On our NVIDIA 3080Ti GPU server, the training time for the 30 models in CrossWeigh is 44.45 hours, whereas that for our two sub-models is 20.25 hours contained parameter tuning. On a new corpus, if CrossWeigh requires tuning, our actual time can be less than 50% of that for CrossWeigh. If the corpus is larger, the saved time can be more meaningful. Therefore, our model is more efficient and occupies less space for model parameters.

5.2 Implications of improved BNN

We further discuss the user annotation noise in human labeling. An annotator can be viewed as an "annotation machine". Assuming that professional annotation instructions are given, if an annotator has a serious attitude for the task, the random deviation will be small. However, if the labeling task is difficult (professional instructions may also contain controversial or vague descriptions), a non-trivial systematic deviation may occur. This can also explain why the performance of conventional BNN with random deviation is worse than that with systematic deviation.

In addition, the improved BNN with learnable systematic deviation terms is a general model similar to the vanilla BNN. Our improved BNN model can be used in any other areas that are prone to systematic understanding bias, such as the classical image classification ImageNet dataset and object detection COCO dataset.

6 Conclusions

In this paper, an annotation noise detection model is constructed based on the annotation cues contained in annotated entities and their surrounding texts by leveraging a novel self-context contrastive loss. The annotation noise detection model we proposed is reduced by half in the time consumption compared to CrossWeigh's noise detection model, and the F1 score has increased by more than 7%. In addition, by adding a systematic deviation to the existing BNN, a new general BNN is presented. Experimental results on two prevalent but noisy NER benchmark datasets indicate the effectiveness of our proposed SdBNN. Compared with noise-robust NER models that do not consider systematic mislabels, such as CrossWeigh, CR and BNN models, the performance of our proposed SdBNN model is significantly improved under the BiLSTM-CRF and BERT baseline models. We analyze the annotation examples in the CoNLL03 test set and confirm that the named entity recognition task is prone to systematic mislabeling. The inconsistency among original annotations, CrossWeight corrections, and our corrections shows the huge challenge of NER annotations.

7 Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

[1] Aguilar G, López-Monroy AP, González FA, et al (2019) Modeling noisiness to recognize named entities using multitask neural networks on social media. Preprint at https://arxiv.org/abs/1906.04129

[2] Akbik A, Blythe D, Vollgraf R (2018) Contextual string embeddings for sequence labeling. In: Proceedings of the 27th international conference on computational linguistics, pp 1638–1649

[3] Akbik A, Bergmann T, Blythe D, et al (2019) FLAIR: An easy-to-use framework for state-of-the-art NLP. In: Proceedings of the 2019 annual conference of the North American chapter of the association for computational Linguistics, pp 54–59

[4] Apratim B. BSMario . (2018) Long-term on-board prediction of people in traffic scenes under uncertainty. In: Proceedings of the IEEE conferenceon computer vision and pattern recognition, pp 4194–4202

[5] Derczynski L, Nichols E, van Erp M, et al (2017) Results of the WNUT2017 shared task on novel and emerging entity recognition. In: Proceedings of the 3rd workshop on noisy user-generated text, pp 140–147

[6] Devlin J, Chang MW, Lee K, et al (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 annual conference of the North American chapter of the association for computational linguistics, pp 4171–4186

[7] Duan Y, Wu O (2017) Learning with auxiliary less-noisy labels. IEEE transactions on neural networks and learning systems 28(7):1716–1721. https://doi.org/10.1109/TNNLS.2016.2546956

[8] Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33th international conference on machine learning, pp 1050–1059

[9] Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks 18(5–6):602–610. https://doi.org/10.1016/j.neunet.2005.06.042

[10] Gui T, Ma R, Zhang Q, et al (2019) Cnn-based chinese ner with lexicon rethinking. In: In Proceedings of the 28th international joint conference on artificial intelligence, pp 4982–4988

[11] Guo Q, Guo Y (2022) Lexicon enhanced chinese named entity recognition with pointer network. Neural Computing and Applications

[12] Guo Q, Qiu X, Liu P, et al (2019) Star-transformer. 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 1315–1325

[13] Hang Y, Bocao D, Xipeng Q (2019) Tener: Adapting transformer encoder for name entity recognition. Preprint at https://arxiv.org/abs/1911.04474

[14] Hao Z, Wang H, Cai R, et al (2013) Product named entity recognition for chinese query questions based on a skip-chain crf model. Neural Computing and Applications 23(2):371–379. https://doi.org/10.1007/s00521-012-0922-5

[15] Huang J, Qu L, Jia R, et al (2019) O2u-net: A simple noisy label detection approach for deep neural networks. In: Proceedings of the IEEE international conference on computer vision, pp 3326–3334

[16] Jenni S, Favaro P (2018) Deep bilevel learning. In: Proceedings of the 15th european conference on computer vision, pp 618–633

[17] Jindal I, Pressel D, Lester B, et al (2019) An effective label noise model for dnn text classification. In: Proceedings of the 2019 annual conference of the North American chapter of the association for computational linguistics

[18] Kendall A, Gal Y (2017) What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems, pp 5574–5584

[19] Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the 26th annual conference on neural information processing Systems, pp 1097–1105

[20] Kun L, Yao F, Chuanqi T, et al (2021) Noisy-labeled ner with confidence estimation. In: Proceedings of the 2021 annual conference of the North American chapter of the association for computational linguistics

[21] Lee J, Bahri Y, Novak R, et al (2018) Deep neural networks as gaussian processes. In: Proceedings of the 6st international conference on learning representations

[22] Li J, Sun A, Ma Y (2020) Neural named entity boundary detection. IEEE Transactions on Knowledge and Data Engineering 33(4):1790–1795

[23] Liang C, Yu Y, Jiang H, et al (2020) Bond: Bert-assisted open-domain named entity recognition with distant supervision. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1054–1064

[24] Liu Jw, Ren Zp, Lu Rk, et al (2021) Gmm discriminant analysis with noisy label for each class. Neural Computing and Applications 33(4):1171–1191

[25] Ma X, Hovy E (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

[26] Mikolov T, Chen K, Corrado G, et al (2013) Efficient estimation of word representations in vector space. In: Proceedings of the 1st international conference on learning representations, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings

[27] Nie Y, Zhang Y, Peng Y, et al (2022) Borrowing wisdom from world: modeling rich external knowledge for chinese named entity recognition. Neural Computing and Applications 34(6):4905–4922

[28] Northcutt CG, Athalye A, Mueller J (2021) Pervasive label errors in test sets destabilize machine learning benchmarks. Preprint at https://arxiv.org/abs/2103.14749

[29] Panchendrarajan R, Amaresan A (2018) Bidirectional LSTM-CRF for named entity recognition. In: Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, Hong Kong

[30] Pennington J, Socher R, Manning C (2014) GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 1532–1543

[31] Peters ME, Ammar W, Bhagavatula C, et al (2017) Semi-supervised sequence tagging with bidirectional language models. In: Proceedings of the 55th annual meeting of the association for computational linguistics, pp 1756–1765

[32] Rodrigues F, Pereira F, Ribeiro B (2014) Sequence labeling with multiple annotators. Machine learning 95(2):165–181. https://doi.org/10.1007/s10994-013-5411-2

[33] Sanh V, Debut L, Chaumond J, et al (2019) Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:191001108

[34] Shang J, Liu L, Ren X, et al (2018) Learning named entity tagger using domain-specific dictionary. Preprint at https://arxiv.org/abs/1809.03599

[35] Shang Y, Huang HY, Mao X, et al (2020) Are noisy sentences useless for distant supervised relation extraction? In: Proceedings of the AAAI conference on artificial intelligence, pp 8799–8806

[36] Shu J, Xie Q, Yi L, et al (2019) Meta-weight-net: Learning an explicit mapping for sample weighting. In: Proceedings of the 33th annual conference on neural information lrocessing systems, pp 1917–1928

[37] Tjong Kim Sang EF, De Meulder F (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, pp 142–147

[38] Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008

[39] Wang J, Xu W, Fu X, et al (2020) Astral: adversarial trained lstm-cnn for named entity recognition. Knowledge-Based Systems 197:105,842. https://doi.org/10.1016/j.knosys.2020.105842

[40] Wang Y, Liu W, Ma X, et al (2018) Iterative learning with open-set noisy labels. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8688–8696

[41] Wang Z, Shang J, Liu L, et al (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, pp 5157–5166

[42] Wei W, Wang Z, Mao X, et al (2021) Position-aware self-attention based neural sequence labeling. Pattern Recognition 110:107,636. https://doi.org/10.1016/j.patcog.2020.107636

[43] Xiao Y, Wang WY (2019) Quantifying uncertainties in natural language processing tasks. In: Proceedings of the AAAI conference on artificial intelligence, pp 7322–7329

[44] Xu Z, Qian X, Zhang Y, et al (2008) Crf-based hybrid model for word segmentation, ner and even pos tagging. In: Proceedings of the sixth SIGHAN workshop on Chinese language processing, pp 167–170

[45] Zhai F, Potdar S, Xiang B, et al (2017) Neural models for sequence chunking. In: Proceedings of the AAAI Conference on Artificial Intelligence

[46] Zhang X, Wu X, Chen F, et al (2020) Self-paced robust learning for leveraging clean labels in noisy data. In: Proceedings of the AAAI conference on artificial intelligence, pp 6853–6860

[47] Zhou G, Su J (2002) Named entity recognition using an hmm-based chunk tagger. In: Proceedings of the 40th annual meeting of the association for computational linguistics, pp 473–480

[48] Zhou W, Chen M (2021) Learning from noisy labels for entity-centric information extraction. Preprint at https://arxiv.org/abs/2104.08656