Named Entity Recognition via Noise Aware Training Mechanism with Data Filter

Xiusheng Huang[1,2], **Yubo Chen**[1,2], **Shun Wu**[1], **Jun Zhao**[1,2],
Yuantao Xie[3] and **Weijian Sun**[3]
[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Huawei Technologies Co., Ltd, Shenzhen, China
huangxiusheng2020@ia.ac.cn,
{yubo.chen,shun.wu,jzhao}@nlpr.ia.ac.cn,
{xieyuantao2,sunweijian}@huawei.com

Abstract

Named entity recognition (NER) is a fundamental task in natural language processing, these is a long held belief that datasets benefit the model. However, not all the data help with generalization, and some samples may contain ambiguous entities or noisy labels. The existing methods can not distinguish hard samples from noisy samples well, and becomes particularly challenging in the case of overfitting. This paper proposes a new method called Noise-Aware-with-Filter (NAF) to solve the issues from two sides. From the perspective of the data, we design a Logit-Maximum-Difference (LMD) mechanism, which maximizes the diversity between different samples to help the model identify noisy samples. From the perspective of the model, we design an Incomplete-Trust (In-trust) loss function, which boosts L_{CRF} with a robust Distrust-Cross-Entropy(DCE) term. Our proposed In-trust can effectively alleviate the overfitting caused by previous loss function. Experiments on six real-world Chinese and English NER datasets show that NAF outperforms the previous methods, and which obtained the state-of-the-art(SOTA) results on the CoNLL2003 and CoNLL++ datasets.

1 Introduction

Named entity recognition (NER) is a primary task and which identifies both types and spans in sentences. NER models are becoming more and more accurate in prediction tasks, the potential improvement of existing architectures in real-world applications is often inherently limited by data quality (Pleiss et al., 2020). However, not all samples are completely correct in the NER datasets (Nooralahzadeh et al., 2019; Lange et al., 2019). Many real-world datasets generally contain samples which are "weakly-labeled" (Derczynski et al., 2017; Peng and Dredze, 2015). Specifically, some

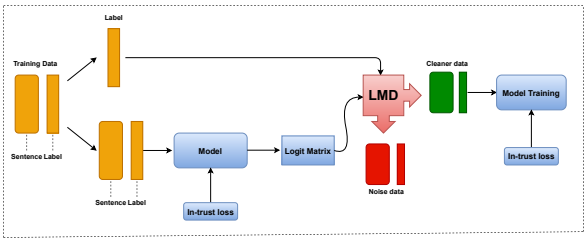

Figure 1: An overview of Noise-Aware-with-Filter(NAF). In-trust loss prevents model from overfitting and helps model generates logit matrices, which enters LMD mechanism with labels to filter noise data and get a cleaner data for model training.

datasets which are annotated based on distant supervision (Yang et al., 2018; Liang et al., 2020) contain more noise labels, and manual annotators, especially on crowdsourcing platform, are prone to making labeling mistakes. Meanwhile labeling a huge datasets is an expensive and fallible process.

As the increase of training iteration epochs, the model will overfit noisy samples and hinder the generalization of the model (Pleiss et al., 2020). In NER task, it is impractical to get an absolutely clean dataset, and the existing datasets generally exist mislabeled samples (Flor et al., 2019) and ambiguous entity (Nadeau et al., 2006), even if some classical datasets (e.g., CoNLL2003 (Tjong Kim Sang and De Meulder, 2003)) still contain noisy samples (Wang et al., 2019b). It makes sense to obtain a cleaner dataset, but it would be very difficult to correct these real-world datasets manually, and the existing methods can't solve the issues automatically , especially in the case of existing ambiguous entities in sentences (Wang et al., 2019b).

From the CoNLL2003 NER dataset, we divide the samples into **Easy samples** which are correctly labeled and do not contain ambiguous entities, **Hard samples** are correctly labeled but contain ambiguous entities, and **Noisy samples** are misla-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4791–4803

beled (Wang et al., 2019b). Such as:

- **Easy samples:{** third and final test between [England]{LOC} and [Pakistan]{LOC} at [the]{LOC}[Oval]{LOC} on **}**. The sample is labeled correctly and there is no ambiguous entity.
- **Hard samples:{** [Chicago]{ORG} won Game 1 with Derrick Rose scoring 25 points **}**. [Chicago] is a basketball team in NBA which is correctly marked as [Chicago]{ORG} here. Meanwhile [Chicago] is also a city in the United States, it is easy to mark that as [Chicago]{LOC} due to ambiguity.
- **Noisy samples:{** Soccer - [Japan]]{LOC} get lucky win, [China]{PER} in surprise defeat **}**. The [China]{PER} is mislabeled.

We can easily obtain the boundary between easy samples and noisy samples with utilizing loss values (Lin et al., 2017), but distinguishing hard samples from noisy samples still is a challenge (Wang et al., 2019b; Pleiss et al., 2020), and becomes particularly challenging in the case of overfitting (Wang et al., 2019b; Liu et al., 2020).

We propose a new method called Noise-Aware-with-Filter (NAF) to solve the issues from two sides. From the perspective of data, we design a Logit-Maximum-Difference (LMD) mechanism, which maximizes the diversity between different samples to help the model identify noisy samples. The difference between easy samples and noisy samples is very obvious in LMD score, meanwhile hard samples and noisy samples also can be well distinguished. From the perspective of model, we propose a noise tolerant term named Distrust-Cross-Entropy(DCE), which combines with L_{CRF} form the basis of the approach Incomplete-Trust (In-trust) loss function. In-trust not only improves the robustness of the model, but also helps LMD improve the accuracy of identifying noisy samples. Experiments on six real-world Chinese and English datasets show that NAF is more accurate than other methods in identifying noisy samples, meanwhile the datasets after filtering are cleaner.

In summary, our major contributions are the following:

- We propose a new method called Noise-Aware-with-Filter (NAF) to distinguish hard samples from noisy samples especially in the case of overfitting.
- To distinguish hard samples from noisy samples, we design a Logit-Maximum-Difference (LMD) mechanism. Meanwhile to alleviate the negative impact of overfitting, we propose Incomplete-Trust (In-trust) loss function, which utilizes both the incomplete correctness of labels and the relative correctness of the model output.
- We conduct extensive experiments on six real-world Chinese and English NER datasets show that NAF outperforms the previous methods, and which obtains the state-of-the-art(SOTA) results on the CoNLL2003 and CoNLL++ datasets. We release the source code publicly for further research [1].

2 Related Work

There are various approaches have been proposed to obtain a robust model. We summarize them into three categories: 1) Robust loss methods, 2) Training architectures methods, 3) Label correction methods.

Robust loss methods specifically design robust loss functions. They include Mean Absolute Error (MAE) (Ghosh et al., 2017), Improved MAE (Wang et al., 2019a) which is a reweighted MAE. Symmetric cross entropy (Wang et al., 2019b), by adding a symmetric reverse cross entropy after the cross entropy, makes the model have a certain noise tolerance, and Generalized cross entropy (Zhang and Sabuncu, 2018) is actually a new evolutionary form of MAE. Regularization (LSR) (Szegedy et al., 2016) is a technique using soft labels in place of one-hot labels to alleviate overfitting to noisy labels. ELR (Liu et al., 2020) is a kind of method that makes full use of early learning phenomenon to keep a large learning gradient for clean samples. But these methods can not effectively distinguish hard samples from noisy samples, and which are easy to confuse them.

Training architectures methods identify noisy samples from the perspective of model framework. Co-Tearching (Han et al., 2018; Yu et al., 2019) utilizes "early learning" phenomenon to maintain two networks in the process of training. All samples are sorted based on the loss values, and the noisy

[1] https://github.com/Huangxiusheng/Named-Entity-Recognition-via-Noise-Aware-Training-Mechanism-with-Data-Filter

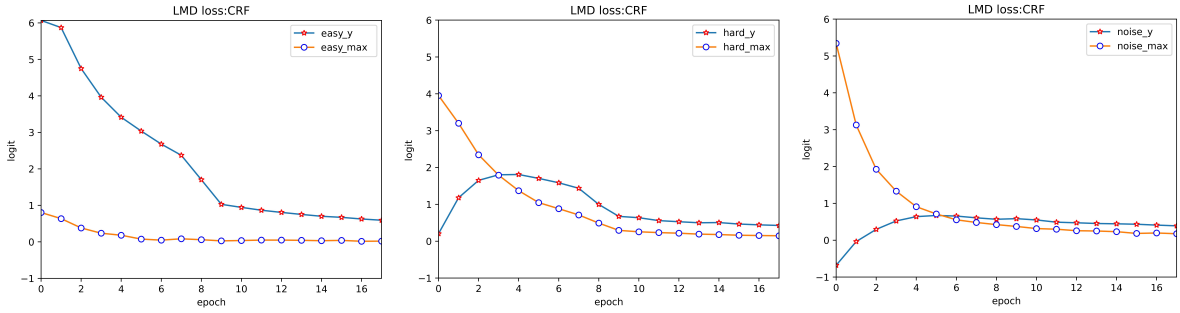

Figure 2: The graphs display logit value trajectories for easy sample (left), hard sample (middle) and noisy sample(right). The blue line: the logit value $Z_y^{(t)}$ corresponding to the label in logit matrix; the orange line: the maximal logit value $Z_{max}^{(t)}$ in logit matrix except $Z_y^{(t)}$. The x-axis refers to the number of training epochs, and the y-axis refers to the logit value. (Dataset: WUT-17)

samples are deleted according to the forgetting ratio (Jiang et al., 2018; Malach and Shalev-Shwartz, 2017). AUM (Pleiss et al., 2020) based on the output of the model to distinguish hard samples from noisy samples. CrossWeigh (Wang et al., 2019b) cover the label of a certain category in the datasets, and observes the model whether will predict the sample into another category. However, these methods always make the model to learn the easy samples and not consider the problem of overfitting (Chang et al., 2017).

Label correction methods are to improve the quality of raw labels. New labels equal to the probabilities estimated by the model (known as soft labels) or to one-hot vectors representing the model predictions (hard labels) (Tanaka et al., 2018; Yi and Wu, 2019). Another option is to set the new labels to equal a convex combination of the noisy labels and the soft or hard labels (Reed et al., 2015). However, these methods require the support from extra clean data or an expensive detection process to estimate the noise model.

3 Logit Maximum Difference Mechanism (LMD)

In this section, we propose a novel LMD mechanism. The LMD utilizes the tiny difference between hard samples and noisy samples in the model output. Meanwhile the LMD accumulates and expands the difference to identify noisy samples.

3.1 Preliminary

Easy samples and noisy samples are easy to distinguish(e.g., utilizing loss values (Han et al., 2018)), because of noisy samples are always contrary to the samples with correct tags. However, hard samples with ambiguous entity are difficult to distinguish from noisy samples, because hard samples also will produce large loss values (Song et al., 2020) in the early stage of training. This has become a major challenge in the denoising task (Song et al., 2020).

Utilizing Logit Matrix Neural network models will output a logit matrix in the training process, which goes through the softmax layer and then gets into loss function. The Softmax layer is a normalized exponential function, which will nonlinear increase the weight of maximum value in the logit matrix and bring unfairness for identifying noisy samples. LMD directly utilizes logit matrix to distinguish hard samples from noisy samples instead of loss values.

Given a sentence $x = [x_1, x_2, ..., x_n]$ and its tag sequence $y = [y_1, y_2, ...y_n]$, n is the sentence length. Every token x_i will obtain a corresponding logit matrix $Z = [z_1^i, z_2^i, ..., z_m^i]$, m denotes total number of tags. LMD utilizes the difference between the z_j corresponding to the class j and other values in the logit matrix.

Observing The Difference In Figure 2, the logit value $Z_y^{(t)}$ corresponding to tag y and epoch t, and the $Z_y^{(t)}$ is evidently higher than other values in easy samples(left). In hard samples(middle), the $Z_y^{(t)}$ is small at the beginning of training, then $Z_y^{(t)}$ begins to increase and become the maximum in the logit matrix with increasing epoch t. In noisy samples(right), the $Z_y^{(t)}$ is relatively smaller than other values, and the $Z_y^{(t)}$ becomes the maximum in the logit matrix when epoch exceeds 5 even if y is a negative tag, which indicates that overfitting occurs.

3.2 Identify Noisy Samples

Defining LMD We propose a new statistic LMD score, which averages the difference between $Z_y^{(t)}$ and the other values $Z_{other}^{(t)}$ at each epoch t. The tiny difference between hard samples and noisy samples are gradually accumulated and maximized, which will effectively help model identify noisy samples. The LMD score can be defined as:

$$LMD(x,y)=\frac{1}{T}\sum_{t=1}^{T}(min(Z_y^{(t)}-\max_{i\neq y} Z_i^{(t)})) \tag{1}$$

Where T is the total number of epoch. A sentence is the minimum unit of the input in the NER task. If the tag corresponding to single token is mislabeled in a sentence, we can consider that the sentence is negative. Therefore we choose the smallest LMD score in each sentence as the LMD score of the sentence, where every token will obtain a LMD score in the sentence.

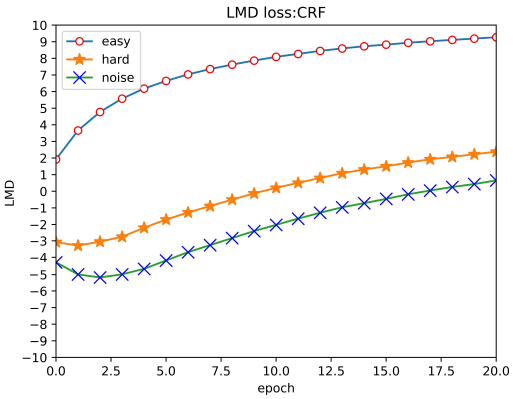

Figure 3: A record of LMD score in each epoch for easy, hard and noisy samples. We select 100 samples from the three sets respectively in WUT-17 dataset.

Working Mechanism In order to steadily enhance the discrimination between easy, hard and noisy samples, we stack the LMD scores of multiple epochs to get an average value. By utilizing the LMD mechanism, every sample will get a LMD score. The LMD scores of easy, hard and noisy samples have obvious differentiation in Figure 3. Then we sort the samples according to the LMD scores, and define samples under the noise ratio μ as noisy samples, finally delete them to get a cleaner training set. And the noise ratio μ is a hyperparameter. The model is trained again with utilizing a clean training set, which will obtain better performance without the interference of noisy samples.

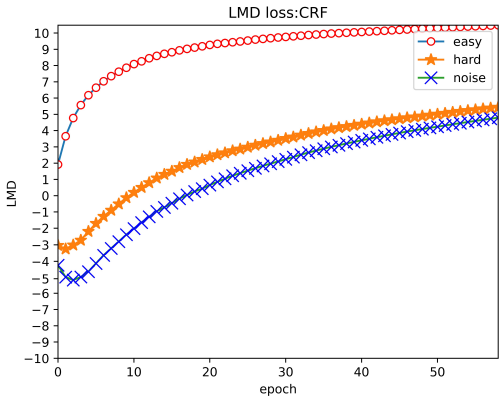

Figure 4: Compared with the experiment in Figure 3, we only adjusted the epoch to 60.

3.3 The Influence of Overfitting

Overfitting Appears We further explore the influence of overfitting in the denoising task from two sides. From the perspective of the LMD scores, the LMD scores of hard samples and noisy samples tend to be consistent with increasing epoch to 60 in the Figure 4. From the perspective of the logit values, the logit value $Z_y^{(t)}$ becomes the maximum in the logit matrix when epoch exceeds 5 even if y is a noise tag in the Figure 2.

Phenomenon Analysis As the increase of training iteration epochs, the model will overfit noisy samples. Meanwhile the model output of hard samples and noisy samples are almost consistent, which make it difficult to distinguish. This proposes another challenge that the model identifies noisy samples in the case of overfitting.

4 Incomplete-Trust Loss Function

In this section, we propose an Incomplete-Trust (In-trust) loss function. Previous loss functions (e.g., Cross Entropy) are easy to overfit noisy samples (Wang et al., 2019b), and they absolutely trust tags even if the tags are mislabeled. Meanwhile, neural networks have strong fitting ability, they can achieve zero training error even on datasets with randomly-assigned labels (Zhang et al., 2016). And deep neural networks have been observed to first fit the samples with clean tags during an "early learning" phase, before eventually memorizing the samples with mislabeled tags (Arpit et al., 2017; Zhang

et al., 2016). With exploiting the early-learning phenomenon, our proposed In-trust utilizes both the model output which obtain the relative correctness after enough practices and the incomplete correctness of tags which maybe are mislabeled. We also provide theoretical analysis about the formulation and behavior of In-trust.

4.1 Definition

KL-divergence Given two distributions p and q, the relationship between entropy, cross entropy and KL-divergence is as follows:

$$KL(q||p)=H(q,p)-H(q) \tag{2}$$

In NER task, $q=q(k|x)$ is the one-hot distribution of the label in sample x, and $p=p(k|x)$ is the prediction distribution of the model for sample x. The model makes the $p=p(k|x)$ gradually approach the $q=q(k|x)$, this is also to minimize the KL-divergence between the two distributions.

Proposing DCE Term However, if the sample x is a noisy sample and the $q=q(k|x)$ is an incorrect distribution, it will cause negative impacts for model, so the label distribution $q=q(k|x)$ is not worthy of full trust. According to the phenomenon of early learning, the model always tends to learn the correct samples in the early stage of training. It means that even if some samples are mislabeled, the model still may predict the correct results in the early stage of training. We exploit this phenomenon to trust that not only the label distribution $q=q(k|x)$, but also the prediction distribution $p=p(k|x)$ before the model overfit noisy samples. Therefore, we design the robust Distrust-Cross-Entropy L_{DCE} term as follows:

$$L_{DCE}=-p\log(\delta p+(1-\delta)q) \tag{3}$$

Where δ is a hyperparameter, and its size determines that the model whether trust labels or model output. When δ is larger, the model will trust prediction distribution $p=p(k|x)$ more, on the contrary, the model will trust label distribution $q=q(k|x)$ more.

Forming In-trust We proposed an Incomplete-Trust (In-trust) loss function, which boosts L_{CRF} with L_{DCE} term.

$$L_{In-trust}=\alpha L_{CRF}+\beta L_{DCE} \tag{4}$$

Where L_{DCE} term is an acceleration regulator term, which can effectively prevent model from overfitting noisy samples. That will be proved in Appendix A. The L_{CRF} term is not noise tolerant (Ghosh et al., 2017), but which benefits the convergence of the model (Zhang and Sabuncu, 2018). α and β are two decoupled hyperparameters, α regulates the overfitting issue of L_{CRF} while β aims to flexibly explore the robustness of L_{DCE}.

Contrasting Logit Values Figure 5 shows the result of comparative experiment with Figure 2. The logit value $Z_y^{(t)}$ corresponding to mislabeled tag y is no longer the maximum in the logit matrix, this means that $L_{In-trust}$ effectively prevents model from overfitting noisy samples.

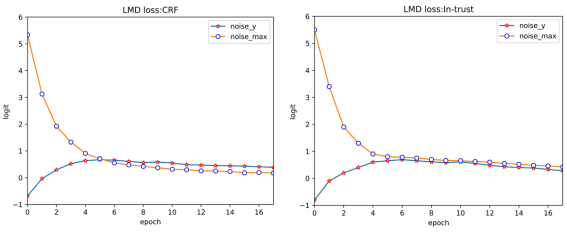

Figure 5: A comparative experiment of CRF (left) on noisy samples. We only replace the loss function to $L_{In-trust}$ (right), and other parameters are consistent.

Contrasting LMD Scores Figure 6 shows the result of comparative experiment with Figure 4. There is still obvious discrimination between hard samples and noisy samples when the epoch reaches 60, this indicates that $L_{In-trust}$ can help LMD mechanism identifies noisy samples accurately.

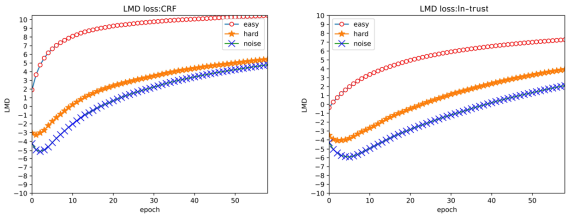

Figure 6: A comparative experiment between CRF (left) and $L_{In-trust}$ (right), other parameters are consistent.

4.2 Robustness Analysis

L_{DCE} Robustness Analysis: In order to simplify the calculation, we set α and β as 1 and derive the gradient of L_{DCE}. For brevity, we denote p_k, q_k as abbreviations for $p=p(k|x)$ and $q=q(k|x)$, the gradient of the L_{DCE} loss with respect to the

logits Z_j can be derived as:

$$\frac{\partial L_{DCE}}{\partial Z_j} = (\sum_{k=1}^{K} p_k \times \log(\delta p_k + (1-\delta) q_k))' \quad (5)$$

Where $\frac{\partial p_k}{\partial Z_j}$ can be further derived based on whether $k = j$:

$$\begin{cases} \frac{\partial p_k}{\partial Z_j} = p_k(1-p_k) & k=j \\ \frac{\partial p_k}{\partial Z_j} = -p_k p_j & k \neq j \end{cases} \quad (6)$$

For brevity, we denote $a_k = \delta p_k + (1-\delta) q_k$ and $b_k = p_k \log a_k + \frac{\delta p_k^2}{a_k}$. We know that $\frac{\partial p_k}{\partial Z_j}$ is a function of p_k from Eq.(5), let $L_{p_k} = \frac{\partial p_k}{\partial Z_j}$, and the gradient of the L_{p_k} with respect to the p_k can be derived as:

$$L'_{p_k} = \sum_{k=1}^{K} b_k + (p_j - 1) \frac{\partial b_j}{\partial p_j} \quad (7)$$

And the second derivative of L_{p_k} is:

$$L''_{p_k} = 2b' + (p_j - 1) b'' \quad (8)$$

Where L''_{p_k} is a monotone increasing function, when $q_j = 1$ and $\delta \in [0.0, 0.1, ...]$, we obtain the corresponding relation between L''_{p_k} and p_j in the Figure 7. It is concluded that L'_{p_k} is a decreasing and then increasing function, which also shows that the acceleration of L_{DCE} first decreases and then increases with the increase of p_j corresponding to the label.

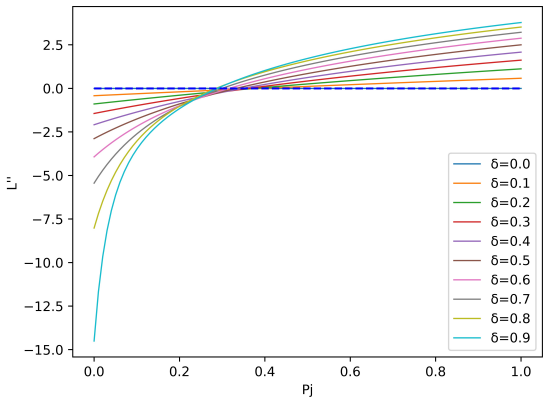

Figure 7: The acceleration record of LDC term.The LDC term produces different accelerations for different model outputs. The x-axis is p_j and the y-axis is the corresponding L''_{p_k} value. $\delta \in [0.0, 0.1, ...]$.

When p_j approaches to 1 with the q distribution is close to the p distribution, the model will believes correct tags more, and L_{DCE} has larger acceleration in learning correct samples. That benefits the model learns cleaner samples and prevents overfitting. On the contrary, the L_{DCE} term thinks that the model has relatively correct prediction result for noisy samples under the influence of learning other correct labels, and the acceleration is small which also effectively prevents the model overfitting and improves the noise tolerance. And other more detailed proofs are shown in the Appendix A.

$L_{In-trust}$ Robustness Analysis: According to Eq.(4), $L_{In-trust}$ consists of L_{CRF} and L_{DCE}. When $q_j = 1$, L_{DCE} term will provide a robust acceleration value, which benefits $L_{In-trust}$ obtains a correct loss value. Specifically, $L_{In-trust}$ will obtain a greater loss value when p_j approach to q, which will benefit the model learn this sample like L_{CRF}. On the contrary, L_{DCE} will prevent model from learning the sample unlike L_{CRF}. When $q_j = 0$, other loss functions will prevent the model from learning the direction, even if the model output p is greater in this direction. We believe that the model is relatively correct after learning a large number of samples. And L_{DCE} term will provide $L_{In-trust}$ with an acceleration to help the model to learn the direction. In addition, L_{DCE} term also prevent the model from learning when p is small. Therefore L_{DCE} term has no negative effect on the convergence of model.

5 Experiments

In this section, we verify the advantages of Noise-Aware-with-Filter (NAF) method by comparing experiments with other denoising methods.

5.1 Experimental Setup

NER Dataset

English NER Dataset We evaluate our method on English NER datasets include WUT-17 (Derczynski et al., 2017), CoNLL2003 (Tjong Kim Sang and De Meulder, 2003), CoNLL++ (Wang et al., 2019b) and OntoNotes5 (Pradhan et al., 2013). CoNLL2003 is in news domain and WUT-17 is user generated text. Compared with CoNLL2003, the test set of CoNLL++ is manually corrected. OntoNotes5 is a larger dataset and contains 18 entity types.

Dataset	Weibo NER	OntoNotes4	OntoNotes5	CoNLL2003	CoNLL++	WUT-17
CE	59.71	81.05	85.57	90.99	92.07	54.77
L_{CRF}	60.88	81.22	85.40	91.60	91.87	55.22
SCE	60.97	80.72	85.30	91.53	91.26	53.55
DSC	60.78	80.34	85.23	91.34	92.08	54.48
ELR	57.41	**81.54**	84.48	91.24	91.73	53.54
$L_{In-trust}$(our)	**61.97**	81.29	**86.23**	**91.67**	**92.68**	**56.13**
NAF(our)	**62.55**	**82.15**	**86.46**	**91.72**	**92.95**	**58.54**

Table 1: The result of NAF($L_{In-trust}$ + LMD) and other denoising methods in the **BERT** model.

Dataset	CoNLL2003	CoNLL++	WUT-17
CE	94.31	95.80	64.99
L_{CRF}	94.22	95.90	64.85
RCE	94.29	95.37	63.73
DSC	94.33	94.79	64.89
ELR	94.10	94.21	63.58
$L_{In-trust}$(our)	**94.45**	**96.13**	**66.44**
NAF(our)	**94.51**	**96.25**	**67.88**

Table 2: The result of NAF($L_{In-trust}$ + LMD) and other denoising methods in the **LUKE** model.

Chinese NER Dataset Chinese NER datasets include Weibo NER (Peng and Dredze, 2015) and OntoNotes4 (Pradhan et al., 2011). Weibo NER is in social domain, OntoNotes4 is in news domain.

In these six real-wold datasets, we use the same way of data segmentation as the original author. Since WUT-17 has no development set, we randomly select 10% samples from the training set as the development set.

Pre-trained Language Model **BERT** (Devlin et al., 2019) employs a Transformer encoder to learn a BiLM from large unlabeled text corpora and sub-word units to represent textual tokens. We use the $BERT_{base}$ model in our experiments. **LUKE** (Yamada et al., 2020)proposes new pretrained contextualized representations of words and entities based on the bidirectional transformer, which is the state-of-the-art(SOTA) model in English NER task.

Baseline We compare NAF with 3 recently proposed robust methods as well as the standard L_{CRF} : (1) CE:Cross Entropy; (2)SCE (Wang et al., 2019b): symmetric cross entropy loss; (3)DSC (Li et al., 2020): dice loss function; (4)ELR (Liu et al., 2020): early regularization; (5)In-trust (We proposed Incomplete-Trust loss function).

Evaluation Our primary evaluation metric is F1 score on the test set to compare the results of different methods.

5.2 Experimental Settings

In our experiments, we set the initial learning rate to $lr = 1e{-}5$ for all datasets. Since the scale of each dataset varies, we set different training batch size for different datasets. Specifically, we set the batch sizes of Weibo NER, OntoNotes4, WUT-17, CONLL2003 and CoNLL++ as 40, 40, 40,32 and 32 in **BERT**, and set the batch sizes of WUT-17, CONLL2003 and CoNLL++ as 2, 2 and 2 in **LUKE**. We stop the training when we find the best result in the development set.

5.3 Robustness Performance

Table 1 presents the results for the baseline and our methods in the **BERT**. Compared with other methods, NAF shows obvious advantage in the six real-wold datasets. Our method outperforms other methods by 1.58%, 0.61%, 0.29%, 0.87% and 3.77% in F1 score on Weibo NER, OntoNotes4, CoNLL2003, CoNLL++ and WUT-17 datasets. Table 2 presents the results in the **LUKE**, and our method has achieved new state-of-the-art (SOTA) with the F1 score reached to **94.51%** and **96.25%** on CoNLL2003 and CoNLL++ datasets.

Specifically, NAF has made more obvious progress on Weibo NER, WUT-17 and CoNLL++ datasets, and our analysis shows that the noise ratio of Weibo NER and WUT-17 are greater than others and there is a cleaner test set after manual correction in CoNLL++ dataset.

5.4 Manual Verification

Results Statistics In order to prove the effectiveness of our method, we manually verify the noisy samples which are selected from CoNLL2003, OntoNotes4 and WUT-17 (Table 4). We randomly select 100 samples from "Original" train set and we manually verify the proportion of noisy samples. After utilizing LMD or NAF method, we will obtain a new datasets, and then we randomly select

Dataset	noisy samples
CoNLL2003	**Soccer - KEANE signs four-years contract with Manchester United{LOC} .**
	Soccer - sharpshooter knup back in swiss{MISC} squad .
	Little{PER} change from today 's weather expected .
	9/16 - Luo Yigang (China) beat Hwang Sun-ho{MISC} (South Korea) 15-3 .
WUT-17	**Federal lawyers{O} fly to Minneapolis to investigate shooting**
	@janzensational at least may date ka na hahaha{O} . Goodluck zen ! : *

Table 3: noisy samples in CoNLL2003 and WNUI-17. The mislabel entities are marked with red.

Dataset	CoNLL2003	Weibo NER
$\delta = 0.1$	90.34%	60.58%
$\delta = 0.2$	90.52%	61.12%
$\delta = 0.3$	91.21%	61.32%
$\delta = 0.4$	90.43%	61.78%
$\delta = 0.5$	**91.72%**	61.44%
$\delta = 0.6$	91.47%	62.21%
$\delta = 0.7$	91.28%	**62.55%**
$\delta = 0.8$	91.32%	62.32%
$\delta = 0.9$	90.20%	61.70%

Table 5: The effect of δ in **In-trust**. We set α=1 and β=1 here.

100 samples from the new datasets for manually verifying. The probability of true negative samples is 5% in the "Original" CoNLL2003 dataset, and which reaches to 76% and 82% respectively with utilizing LMD and NAF methods. While for OntoNotes4 dataset, the probability is 8% in "Original" dataset and which reaches to 72% and 80% with LMD and NAF. In the WUT-17 dataset, the probability is 18% in "Original" dataset and which reaches to 58% and 71% with LMD and NAF.

Result Analysis The accuracy of identifying noisy samples will greater with the dataset that model performs better, and the excellent datasets will benefit the model identify noisy samples.

Dataset	Original	LMD(our)	NAF(our)
CoNLL2003	5%	**76%**	**82%**
OntoNotes4	8%	**72%**	**80%**
WUT-17	18%	**58%**	**71%**

Table 4: The accuracy of identifying noisy samples is verified manually. We select 100 samples from the Original train set, LMD and NAF separately, then verify the proportion of noisy samples manually. And the "Original" means raw data.

Demonstration of Examples The real noisy samples in CoNLL2003 and WUT-17 datasets are shown in Table 3, such as {Soccer - KEANE signs four-years contract with Manchester United}, the { **Manchester United**} is wrongly marked **{LOC}**. The { **Manchester United** } is a football club in Manchester England, and which should be marked **{ORG}**. In addition, the more noisy samples in datasets are shown in the Appendix C.

5.5 Ablation Experiment

As mentioned in the previous Section 4.1, δ provides flexibility between the model output distribution p=$p(k|x)$ and the label distribution q=$q(k|x)$. In this section, we explore the influence of hyperparameter δ, and we conducted experiments on Weibo NER and CoNLL2003 with α=1 and β=1 to explore how it manipulates the tradeoff. Experimental results are shown in Table 5. The highest F1 on CoNLL2003 datatset is 91.72% when δ is set to 0.5, meanwhile for Weibo NER, the highest F1 is 62.55% when δ is set to 0.7. The optimal value of δ is different in different noise ratio datasets, and when there are more noisy samples in the datasets, δ should be set larger. Because of the noise ratio of Weibo NER is larger than CoNLL2003 dataset, the optimal δ value of Weibo NER is larger. The experiment result of the other hyperparameters α and β are show in the Appendix B.

6 Conclusion

In this paper, we observe that the existing denoising methods can not effectively distinguish hard samples from noisy samples, and we proposed a new method called Noise-Aware-with-Filter (NAF), which contains LMD mechanism and In-trust loss function to solve the issues. Specifically, NAF can effectively improve the discrimination between hard samples and noisy samples even in the case of overfitting. In addition, our proposed the Logit-Maximum-Difference(LMD) mechanism which maximizes the diversity between different samples to help the model identify noisy samples. Mean-

while we design an Incomplete-Trust ($L_{In-trust}$) loss function, which boosts L_{CRF} with a noise robust Distrust-Cross-Entropy(DCE) term. In order to verify the effectiveness of our method, we also conduct manual verification for noisy samples and the results show that our method has higher accuracy on identifying noisy samples. Experiments on six real-world Chinese and English NER datasets show that NAF outperforms the previous methods, and which obtained the state-of-the-art(SOTA) results on the CoNLL2003 and CoNLL++ datasets.

7 Acknowledgments

This work is supported by the National Key Research and Development Program of China (No.2020AAA0106400), the National Natural Science Foundation of China (No.61976211, No.61806201). This work is also supported by a grant from Huawei Technologies Co., Ltd.

References

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

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active bias: training more accurate neural networks by emphasizing high variance samples. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1003–1013.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Michael Flor, Michael Fried, and Alla Rozovskaya. 2019. A benchmark corpus of english misspellings and a minimally-supervised model for spelling correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 76–86.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1919–1925.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. 2018. Co-teaching: robust training of deep neural networks with extremely noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8536–8546.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR.

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.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced nlp tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 961–971.

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

Farhad Nooralahzadeh, Jan Tore Lønning, and Lilja Øvrelid. 2019. Reinforcement-based denoising of distantly supervised ner with partial annotation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 225–233.

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

Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. *arXiv preprint arXiv:2001.10528*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Sameer Pradhan, Lance Ramshaw, Mitch Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27.

Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *ICLR (Workshop)*.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560.

EF Tjong Kim Sang and F De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition.

Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. 2019a. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv preprint arXiv:1903.12141*.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019b. 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.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454.

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.

Kun Yi and Jianxin Wu. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7017–7025.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhilu Zhang and Mert R Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8792–8802.

A $L_{In-trust}$ Robustness Proof

We have know $L_{In-trust}$ is:

$$\begin{aligned} L_{In-trust}&=\alpha \times CRF+\beta \times L_{DCE} \\ &=\alpha \times CRF+\beta \times (-p \times \log(\delta p+(1-\delta)q)) \end{aligned} \tag{9}$$

And make:$a=\delta p+(1-\delta)q$

$$\begin{aligned} Eq9&=\alpha \times CRF+\beta \times (-p \times \log a \\ &=\alpha \times CRF \\ &+\beta \sum_{k=1}^{K}(-\frac{\partial p_k}{\partial Z_j}\log a_k+(-p_k\frac{a_k^{'}}{a_k})) \end{aligned} \tag{10}$$

Meanwhile we have know:

$$\begin{aligned} \frac{\partial p_k}{\partial Z_j}&=\frac{\partial(\frac{Z_k}{\sum_{j=1}^{K}e^{Z_j}})}{\partial Z_j} \\ &=\frac{\frac{\partial e^{Z_k}}{\partial Z_j}(\sum_{j=1}^{K})-e^{Z_k}\frac{\partial(\sum_{j=1}^{K}e^{Z_j})}{\partial Z_j}}{(\sum_{j=1}^{K}e^{Z_j})^2} \end{aligned} \tag{11}$$

When $k=j$:

$$\begin{aligned} \frac{\partial p_k}{\partial Z_j}&=\frac{\partial p_k}{\partial Z_k} \\ &=\frac{e^{Z_k}(\sum_{k=1}^{K}e^{Z_j})-(e^{Z^k})^2}{(\sum_{k=1}^{K}e^{Z_k})^2} \\ &=\frac{e^{Z_k}}{\sum_{k=1}^{k}e^{Z_k}}-(\frac{e^{Z_k}}{\sum_{k=1}^{K}e^{Z_k}})^2 \\ &=p_k-{p_k}^2 \\ &=p_k(1-p_k) \end{aligned} \tag{12}$$

When $k \neq j$:

$$\begin{aligned} \frac{\partial p_k}{\partial Z_j}&=\frac{0(\sum_{k=1}^{K}e^{Z_j})-e^{Z_k}e^{Z_j}}{(\sum_{k=1}^{K}e^{Z_j})(\sum_{k=1}^{K}e^{Z_j})} \\ &=-\frac{e^{Z_k}}{\sum_{k=1}^{K}e^{Z_j}}\frac{e^{Z_j}}{\sum_{k=1}^{K}e^{Z_j}} \\ &=-p_kp_j \end{aligned} \tag{13}$$

Gradient Calculation:

$$\begin{aligned} L_{DCE}&=-p \times \log(\delta p+(1-\delta)q) \\ \frac{\partial L_{DCE}}{\partial Z_j}&=(\sum_{k=1}^{K}p_k \times \log(\delta p_k+(1-\delta)q_k))^{'} \\ &=\sum_{k=1}^{K}{p_k}^{'}\log(\delta p_k+(1-\delta)q_k) \\ &+\frac{\delta p_k{p_k}^{'}}{\delta p_k+(1-\delta)q_k} \end{aligned} \tag{14}$$

For the convenience of calculation, we make $a=\delta p_k+(1-\delta)q_k$

$$\begin{aligned} \frac{\partial L_{DCE}}{\partial Z_j}&=\sum_{k=1}^{K}p_k^{'}\log a_k+\frac{p_ka_k^{'}}{a_k} \\ &=\sum_{k=1}^{K}p_k^{'}\log a_k+\frac{\delta p_kp_k^{'}}{a_k} \\ &=-p_j\sum_{k=1}^{k}(p_k\log p_k+\frac{\delta p_k^2}{a_k}) \\ &+p_j\log a_j+\frac{\delta p_j^2}{\delta a_j} \end{aligned} \tag{15}$$

Make $b_k=p_k\log a_k+\frac{\delta p_k^2}{a_k}$

$$\begin{aligned} \frac{\partial L_{DCE}}{\partial Z_j}&=-p_j\sum_{k=1}^{K}b_k+b_j \\ \frac{\partial L_{DCE}}{\partial Z_j}&=p_j\sum_{k=1}^{K}b_k-b_j \end{aligned} \tag{16}$$

We hypothesis $L=p_j\sum_{k=1}^{K}b_k-b_j$

$$\begin{aligned} L^{'}=\frac{\partial L}{\partial p_j}&=\sum_{k=1}^{K}b_k+p_j(\sum_{k=1}^{K}b_k)^{'}-\frac{\partial b_j}{\partial p_j} \\ &=\sum_{k=1}^{K}b_k+(p_j-1)\frac{\partial b_j}{\partial p_j} \end{aligned} \tag{17}$$

And $L^{''}=2b^{'}+(p_j-1)b^{''}$

Meanwhile we can get:

$$\begin{aligned} b^{'}&=\log a+\frac{pa^{'}}{a}+\frac{2\delta p-\delta p^2a^{'}}{a^2} \\ &=\log a+\frac{\delta p}{a}+\frac{2\delta p-\delta^2p^2}{a^2} \\ &=\log a+\frac{3\delta p}{a}-(\frac{\delta p}{a})^2 \end{aligned} \tag{18}$$

$$\begin{aligned} b^{''}&=\frac{\delta}{a}+3\delta\frac{(1-\delta)q}{a^2}-2\delta^2\frac{p}{a}\frac{(1-\delta)q}{a^2} \\ &=\frac{\delta}{a}+\frac{3\delta(1-\delta)q}{a_2}-\frac{2\delta^2p(1-\delta)q}{a^3} \end{aligned} \tag{19}$$

When $q_j=0$:

$$\begin{aligned} L^{''}&=2\log(\delta+2)+\frac{p-1}{p} \\ &=2\log\delta p+4+1-\frac{1}{p} \\ &=2\log\delta p+5-\frac{1}{p} \end{aligned} \tag{20}$$

When p approaches 0 : $L^{''}<0$

When p approaches 1 : $L^{''}=2\log\delta+4$

$E:\delta 1\in[0,1]$, make $L^{''}(\delta 1)>0$

Because $L^{''}$ is continuous function, so $L^{''}$ is Monotone increasing function

$E:\delta 2\in[0,1]$, make $L^{''}(\delta 2)=0$

So $L^{'}$ is Decreasing then increasing function and the inflection point is only related to δ.

When q_j=1 :

$$b^{'}=\log a+\frac{3\delta p}{a}-(\frac{\delta p}{a})^2$$
$$b^{''}=\frac{\delta}{a}+\frac{3\delta(1-\delta)q}{a_2}-\frac{2\delta^2p(1-\delta)q}{a^3}$$
$$=\frac{\delta a^2+3\delta a-3\delta^2a-2\delta^2p+2\delta^3p}{a^3}$$

$$\begin{aligned}L^{''}&=2b^{'}+(p_j-1)b^{''}\\&=2\log a+\frac{6\delta p}{a}-\frac{2\delta^2p^2}{a^2}+\\&\frac{\delta a^2p+3\delta ap-3\delta^2ap-2\delta^2p^2+2\delta^3p^2}{a^3}+\\&\frac{-\delta a^2-3\delta a+a\delta^2a+2\delta^2p-2\delta^3p}{a^3}\end{aligned}\tag{21}$$

Because $L^{''}$ is monotone increasing function

So when p=0:

$L^{''}=2\log(1-\delta)+\frac{-\delta a^2-3\delta a+3\delta^2a}{(1-\delta)^3}<0$

When p=1:

In order to simplify the calculation, we make δ=0

So $L^{''}=0$

When $\delta>0$, and $L^{''}>0$

So $L^{'}$ is Decreasing then increasing function and the inflection point is only related to δ.

In summary, when q=0 or q=1 :

$L^{'}$ is Decreasing then increasing function and the inflection point is only related to δ.

We know $L^{'}(p=0)>0$, $L^{'}(p=1)>0$,

and $E:\mu\in[0,1]$, make $L^{'}(p=\mu)<0$

We observe that when δ is larger, the model tends to learn from the p of the model output, and when δ is smaller, the model tends to learn from the label q. Moreover, when the p_j corresponding to the label is larger and the model output is close to the label distribution, the acceleration of L_{DC} term is larger, which makes the model more inclined to learn the sample, which helps the model learn clean samples. When p_j is small, there is a big gap between the model output and label distribution. We think that the sample may be a noisy sample, and the acceleration of L_{DC} term is smaller, which makes the model more inclined to give up the learning of the sample, and prevents the model from over fitting the noisy sample.

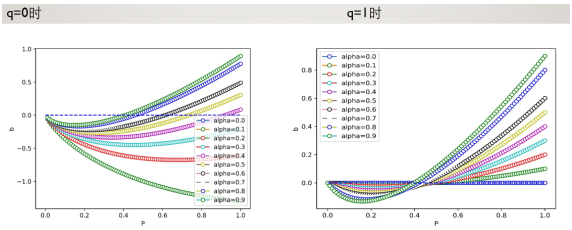

Figure 8: The relationship between b and p.

B Ablation Experiment Supplement

Dataset	CoNLL2003	OntoNotes4
$\alpha=0.1$	27.82%	29.56%
$\alpha=0.2$	30.37%	34.17%
$\alpha=0.3$	32.41%	46.71%
$\alpha=0.4$	45.32%	54.31%
$\alpha=0.5$	56.23%	68.32%
$\alpha=0.6$	74.10%	75.43%
$\alpha=0.7$	89.37%	78.32%
$\alpha=0.8$	90.21%	80.32%
$\alpha=0.9$	91.70%	82.13%

Table 6: Appendix B: The effect of α in **In-trust**. We set δ=0.5 and β=1 here.

Dataset	CoNLL2003	OntoNotes4
$\beta=0.1$	90.99%	81.05%
$\beta=0.2$	91.15%	80.07%
$\beta=0.3$	91.33%	80.07%
$\beta=0.4$	91.32%	80.09%
$\beta=0.5$	91.67%	81.10%
$\beta=0.6$	91.65%	82.09%
$\beta=0.7$	91.68%	82.15%
$\beta=0.8$	91.72%	82.10%
$\beta=0.9$	91.70%	82.13%

Table 7: Appendix B: The effect of β in **In-trust**. We set α=0.6 and δ=0.5 here.

C noisy samples

dataset	noisy samples
CoNLL2003	**TYPE FRN BASE 3M LIBOR PAY DATE S23.SEP.96** **O O O B-ORG O O O O**
	English County Championship cricket matches on Thursday : **B-MISC B-MISC I-MISC O O O O O**
	SOCCER - EUROPEAN CUP WINNERS ' CUP RESULTS . **O O B-MISC I-MISC I-MISC I-MISC I-MISC O O**
	Red Star - Vinko Marinovic (59th) **B-ORG I-ORG O B-MISC I-MISC O O O**
	SOCCER - SHARPSHOOTER KNUP BACK IN SWISS SQUAD . **O O O B-PER O O B-MISC O O**

Table 8: Appendix C: noisy sample display