# Noisy-Labeled NER with Confidence Estimation

**Kun Liu**[1*] **Yao Fu**[2*] **Chuanqi Tan**[1†] **Mosha Chen**[1]
**Ningyu Zhang**[3] **Songfang Huang**[1] **Sheng Gao**[4]
[1]Alibaba Group [2]University of Edinburgh [3]Zhejiang University
[4] Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University
{kun.liu624; sheng.gao.81}@gmail.com yao.fu@ed.ac.uk
{chuanqi.tcq; chenmosha.cms; songfang.hsf}@alibaba-inc.com
zhangningyu@zju.edu.cn

## Abstract

Recent studies in deep learning have shown significant progress in named entity recognition (NER). Most existing works assume clean data annotation, yet a fundamental challenge in real-world scenarios is the large amount of noise from a variety of sources (e.g., pseudo, weak, or distant annotations). This work studies NER under a noisy labeled setting with calibrated confidence estimation. Based on empirical observations of different training dynamics of noisy and clean labels, we propose strategies for estimating confidence scores based on local and global independence assumptions. We partially marginalize out labels of low confidence with a CRF model. We further propose a calibration method for confidence scores based on the structure of entity labels. We integrate our approach into a self-training framework for boosting performance. Experiments in general noisy settings with four languages and distantly labeled settings demonstrate the effectiveness of our method [1].

## 1 Introduction

Recent progress in deep learning has significantly advanced NER performances (Lample et al., 2016; Devlin et al., 2018). While most existing works assume clean data annotation, real-world data inevitably involve different levels of noise (e.g., distant supervision from the dictionary (Peng et al., 2019), or weak supervision from the web Vrandečić and Krötzsch, 2014; Cao et al., 2019a). Figure 1 gives an example of such noisy labels. To train robust models with high performance, it is fundamentally critical to tackle the challenges associated with noisy data annotation.

In this work, we propose a confidence estimation approach for NER with noisy labels. We motivate our approach with important empirical observations of the training dynamics of clean and noisy labels: usually, clean data are easier to fit with faster convergence and smaller loss values (Jiang et al., 2018; Han et al., 2018a; Arazo et al., 2019). Consequently, loss values (probabilities or scores of labels) can serve as strong indicators for the existence of noise, which we utilize to build our confidence estimation.

| | Brooklyn | and | Mary | live | in | New | York |
|---|---|---|---|---|---|---|---|
| Gold Labels | B-PER | O | B-PER | O | O | B-LOC | I-LOC |
| Noisy Labels | B-LOC | O | B-PER | O | O | O | B-LOC |

Figure 1: A noisy label example. *Brooklyn* and *York* are noisy positives. *New* is noisy negative.

The key contribution of this work is a confidence estimation method with calibration. We use probabilities of labels as confidence scores and apply two estimation strategies based on global or local normalization that assume different dependency structures about how the noisy labels are generated. We further calibrate the confidence score for positive labels (labels representing entity parts, e.g., *B-LOC*) based on the structure of these labels: we separately estimate scores for the *position part* (e.g., *B* in *B-LOC*) and the *type part* (e.g., *LOC* in *B-LOC*). Such fine-grained calibration leads to a more accurate estimation and better performance in our experiments.

We apply our method in a CRF model (Bellare and McCallum, 2007; Yang et al., 2018), marginalize out labels we do not trust, and maximize the likelihood of trusted labels. We use a self-training approach (Jie et al., 2019) that iteratively estimates confidence scores in multiple training iterations and re-annotates the data at each iteration. Experiments show that our approach outperforms baselines on a general noisy-labeled setting with datasets in four languages and shows promising results on a distantly-labeled setting with four datasets.

[*] Equal Contribution.
[†] Corresponding author.
[1]Our code can be found at https://github.com/liukun95/Noisy-NER-Confidence-Estimation



*Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3437–3445
June 6–11, 2021. ©2021 Association for Computational Linguistics

## 2 Method

Given a sentence $x = [x_1, ..., x_n]$ and its tag sequence $\hat{y}_1, ..., \hat{y}_n$, $n$ is the sentence length. We model the conditional probability of $y$ with a bi-directional LSTM-CRF (Huang et al., 2015):

$$h = \text{BiLSTM}(x) \quad \Phi_i = \text{Linear}(h_i) \tag{1}$$

$$p(y|x) = \Phi(y)/Z \quad \alpha, Z = \text{Forward}(\Phi) \tag{2}$$

Where $h$ denotes LSTM states, $\text{Linear}(\cdot)$ denotes a linear layer, $\Phi(y)$ denotes the potential (weight) evaluated for tag sequence $y$, $Z$ denotes the partition function, $\alpha$ denotes the forward variables, and $\text{Forward}(\cdot)$ denotes the Forward algorithm (Sutton and McCallum, 2006). The advantage of the CRF model is that it gives us a probabilistically uniform way to handle labels we do or do not trust by partial marginalization, which we discuss later.

### 2.1 Confidence Score Estimation

Our confidence estimation model reuses the base LSTM-CRF architecture and assigns a confidence score $s_i$ for each $\hat{y}_i$. A natural choice is to use the CRF marginal probability:

$$s_i = p(\hat{y}_i|x) \quad p(y_i|x) = \alpha_i \beta_i / Z \tag{3}$$

where $\beta$ is the backward variable and can be computed with the Backward algorithm (Sutton and McCallum, 2006). This strategy infers $s_i$ based on global-normalization and assumes strong dependency between consecutive labels. The intuition is that annotators are more likely to make mistakes on a label if they have already made mistakes on previous labels.

Our second strategy makes a stronger local independence assumption and considers a noisy label at step $i$ only relies on the word context, not the label context. To this end, we use a simple categorical distribution parameterized by a Softmax:

$$s_i = p(\hat{y}_i|x) \quad p(y_i|x) = \text{Softmax}(\Phi_i) \tag{4}$$

Here we reuse the factor $\Phi_i$ as the logits of the Softmax because in the CRF context it also means how likely a label $y_i$ may be observed given the input $h_i$. Intuitively, this strategy assumes that annotators make mistakes solely based on words, no matter whether they have already made mistakes previously.

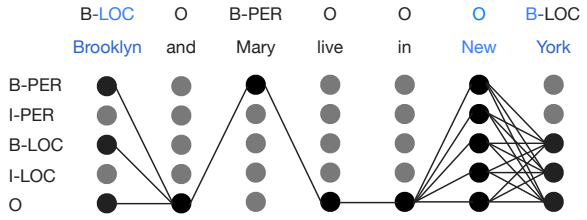



Figure 2: A partial marginalization example after confidence estimation. In this example, we do not trust any labels for *New* (so we marginalize all labels out), partially trust labels for *Brooklyn* (*B* part) and *York* (*LOC* part, so we sum over labels we trust), and fully trust labels for the rest words (so we simply evaluate and maximize their weights.).

### 2.2 Confidence Calibration and Partial Marginalization

We use $s_i$ to decide if we want to trust a label $\hat{y}_i$ and marginalize out labels we do not trust. Our marginalization relies on a threshold to determine the portion of trusted labels and the noise ratio that we believe the data contain. Given a batch of $(x, \hat{y})$ pairs, after confidence estimation, we collect all word-label-confidence triples into a set $\mathcal{D} = \{x_j, \hat{y}_j, s_j\}_{j=1}^{N}$, $N$ denotes total number of the triples.

We further separate the estimation for positive labels (entities) and negative labels (i.e., the *O* label) because we empirically observe that their probabilities are consistently different. To this end, we divide $\mathcal{D}$ into positive and negative groups $\mathcal{D}_\text{p} = \{(x_j, \hat{y}_j, s_j), \hat{y}_j \in \mathcal{Y}_\text{p}\}$ and $\mathcal{D}_\text{n} = \{(x_j, \hat{y}_j, s_j), \hat{y}_j \in \mathcal{Y}_\text{n}\}$, $\mathcal{Y}_\text{p}$ and $\mathcal{Y}_\text{n}$ denotes sets of positive and negative labels. We rank triples in $\mathcal{D}_l$ ($l \in \{\text{p}, \text{n}\}$) according to confidence scores and retain the most confident $r_l(e) \cdot |\mathcal{D}_l|$ triples at epoch $e$ as clean for which we do maximum likelihood. We view the remaining triples as noisy and marginalize them out. We update the keep ratio $r_l(e)$ at each epoch following Han et al. (2018b):

$$r_l(e) = 1 - \min\left\{\frac{e}{K}\tau_l, \tau_l\right\}, l \in \{\text{p}, \text{n}\} \tag{5}$$

where $\tau_l$ is the ratio of noise that we believe in the training data. Basically this says we gradually decrease the epoch-wise keep ratio $r_l(e)$ to the full ratio $1 - \tau_l$ after $K$ epochs. We grid-search $\tau_l$ heuristically in experiments (results in Figure 3(b)).

For positive cases in $\mathcal{D}_\text{p}$ viewed as noisy according to the previous procedure, we do a further confidence calibration. Noting that a $y_i$ always take the form $y_i^p$-$y_i^t$ (position-type) (e.g. if $y_i =$ *B-LOC*,

| Method | General Noise | | | | Distant Supervision | | | |
|---|---|---|---|---|---|---|---|---|
| | En | Sp | Ge | Du | CoNLL | Tweet | Webpage | Wikigold |
| 1. BiLSTM-CRF | 73.3 | 61.9 | 57.7 | 58.3 | 59.5 | 21.8 | 43.3 | 42.9 |
| 2. BiLSTM-CRF (clean data upper bound) | 90.3 | 85.2 | 77.3 | 81.1 | 91.2 | 52.2 | 52.3 | 54.9 |
| 3. RoBERTa (clean data upper bound) | - | - | - | - | 90.1 | 52.2 | 72.4 | 86.4 |
| *Proposed for General Noise Setting* | | | | | | | | |
| 4. NA (Hedderich and Klakow, 2018) | 61.5 | 57.3 | 46.1 | 41.5 | - | - | - | - |
| 5. CBL (Mayhew et al., 2019) | 82.6 | 76.1 | 65.6 | 68.5 | 75.4 | 18.2 | 31.7 | 42.6 |
| 6. Self-training (Jie et al., 2019) | 84.0 | 71.4 | 66.5 | 59.6 | 77.8 | 42.3 | 49.6 | 51.3 |
| *Proposed for Distant Supervision Setting* | | | | | | | | |
| 7. AutoNER (Shang et al., 2018) | - | - | - | - | 67.0 | 26.1 | 51.4 | 47.5 |
| 8. LRNT (Cao et al., 2019a) | - | - | - | - | 69.7 | 23.8 | 47.7 | 46.2 |
| 9. BOND (RoBERTa Liang et al., 2020) | - | - | - | - | **81.5** | **48.0** | **65.7** | **60.1** |
| *Ours, best configurations* | | | | | | | | |
| 10. Ours (local, $\tau^*$) | **87.0** | 78.8 | 68.3 | 69.1 | 79.4 | 43.6 | 51.8 | 54.0 |
| 11. Ours (global, $\tau^*$) | 86.4 | 79.0 | **69.2** | **71.2** | 79.2 | 43.1 | 50.0 | 53.0 |
| *Ours, other possible configurations* | | | | | | | | |
| 12. Ours (local, $\tau^\star$) | 86.2 | **79.2** | 68.2 | 67.2 | - | - | - | - |
| 13. Ours (global, $\tau^\star$) | 85.4 | 75.4 | 68.4 | 69.0 | - | - | - | - |
| 14. Ours (local, $\tau^*$, w/o. calibration) | 85.8 | 77.3 | 67.2 | 68.0 | 79.9 | 40.8 | 46.9 | 50.0 |
| *Ours with pretrained LM* | | | | | | | | |
| 15. Ours (local, $\tau^*$, BERT) | - | - | - | - | 77.2 | 46.7 | 59.3 | 57.3 |
| 16. Ours (global, $\tau^*$, BERT) | - | - | - | - | 78.9 | 47.3 | 61.9 | 57.7 |

Table 1: Results (F1%) on artificially perturbed datasets and distantly supervised datasets. $\tau^*$ = searched, $\tau^\star$ = oracle.

then $y_i^p = B$ and $y_i^t = LOC$), an important assumption is that annotators are unlikely to mistake both parts — mistakes usually happen on only one of them. So we calculate two calibrated confidence scores $s_i^p$ and $s_i^t$ for $\hat{y}_i^p$ and $\hat{y}_i^t$:

$$s_i^p = \frac{1}{|Y(\hat{y}_i^p)|} \sum_{y_i} p(y_i|x) \quad \text{where } y_i^p = \hat{y}_i^p \quad (6)$$

$$s_i^t = \frac{1}{|Y(\hat{y}_i^t)|} \sum_{y_i} p(y_i|x) \quad \text{where } y_i^t = \hat{y}_i^t \quad (7)$$

where $Y(\hat{y}_i^t)$ denotes the set of labels sharing the same $\hat{y}_i^t$ part, and $Y(\hat{y}_i^p)$ is defined similarly. If $s_i^p > s_i^t$, we trust the $\hat{y}_i^p$ (position) part of the label and marginalize out all labels with different positions except for the *O* label. For example, in Figure 2, for the word *Brooklyn* we trust the all labels with the position *B* (*B-PER* and *B-LOC*) and the *O* label, sum over the tag sequences passing these labels, and reject other labels. Similar operation applies for cases where $s_i^p < s_i^t$ (E.g., the word *York*). For labels we do not trust in the negative group $\mathcal{D}_{\text{n}}$, we simply marginalize all labels out (E.g., the word *New*). We maximize the partially marginalized probability (Bellare and McCallum, 2007):

$$\tilde{p}(\hat{y}|x) = \sum_{y \in \tilde{Y}} \Phi(y)/Z \quad (8)$$

where $\tilde{Y}$ denotes the set of tag sequences compatible with $\hat{y}$ after confidence estimation. A concrete example is given in Figure 2. The summation in equation 8 can be calculated exactly with Forward-styled dynamic programming (Sasada et al., 2016).

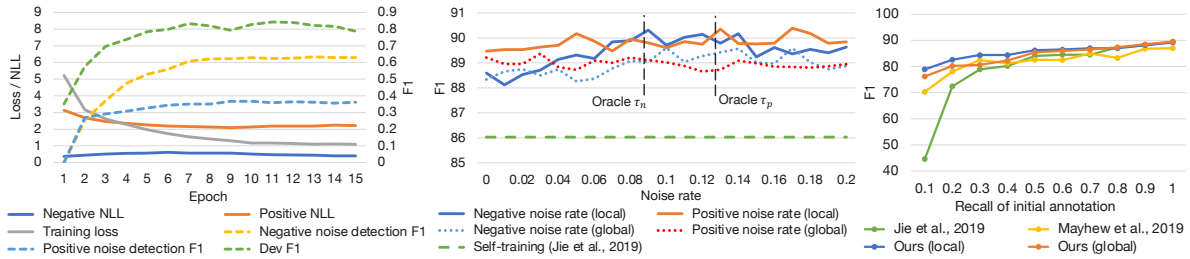
### 2.3 Self Training

We integrate our approach into a self-training framework proposed by Jie et al. (2019). At each round, the training set is randomly divided into two parts for cross-validation. We iteratively re-annotate half of the training set with a model trained on the other half. After a round, we use the updated training set to train the next round.

## 3 Experiments

### 3.1 Datasets and Baselines

**General Noise.** Following Mayhew et al. (2019), we first consider general noise by artificially perturbing the CoNLL dataset (Sang and De Meulder, 2003) on four languages including English, Spanish, German, and Dutch. Gold annotations are per-




(a) Training curve (b) Noise rate search curve (c) Results w.r.t. noise level

Figure 3: Analysis on English CoNLL03 dataset. (a) Dev performance strongly correlates to loss values (confidence scores) and noise detection performance. (b) An over-estimate of noise tends to give better performance. (c). Our approach is particularly effective under larger noise (lower recall = larger noise).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | ... | Edhen | Efendija | Camdzic, | Doboj's | Islamic | ... |
| Noisy Labels | | O | O | O | S-LOC | O | |
| Gold Labels | | B-PER | I-PER | E-PER | S-LOC | S-MISC | |
| 2 | ... | Norm | Charlton | retired | the | final | ... |
| Noisy Labels | | O | S-PER | O | O | O | |
| Gold Labels | | B-PER | E-PER | O | O | O | |
| 3 | ... | cruise | through | the | Pacific | depths | ... |
| Noisy Labels | | O | O | O | S-PER | O | |
| Gold Labels | | O | O | O | S-LOC | O | |
| 4 | ... | including | the | Bharat | Ratna | ... | |
| Noisy Labels | | O | O | O | O | | |
| Gold Labels | | O | O | B-MISC | E-MISC | | |

Figure 4: Confidence estimation case study. Red fonts = noisy positive, blue fonts = noisy negatives. Green shade = correct noise detection, red shade = wrong noise detection.

turbed by: (a) tagging some entities to *O* to lower the recall to 0.5; (b) introducing some random positive tags to lower the precision to 0.9. We compare our methods with Noise Adaption (NA, Hedderich and Klakow, 2018), Self Training (Jie et al., 2019), and CBL (Mayhew et al., 2019). This setting is for testing our approach in a controlled environment.

**Distant Supervision.** We consider four datasets including CoNLL03 (Sang and De Meulder, 2003), Tweet (Godin et al., 2015), Webpage (Ratinov and Roth, 2009), and Wikigold (Balasuriya et al., 2009). In this setting, the distantly supervised tags are generated by the dictionary following BOND (Liang et al., 2020). We compare our methods with AutoNER (Shang et al., 2018), LRNT (Cao et al., 2019a), and BOND. This setting aims to test our approach in a more realistic environment.

### 3.2 Results

Table 1 shows our primary results. We use *local* and *global* to denote locally / globally normalized confidence estimation strategies. We use *oracle* (unavailable in real settings) / *searched* $\tau$ to denote how we obtain the prior noise ratio $\tau$. We note that the Self-training baseline (Jie et al., 2019, line 6) is the most comparable baseline since our confidence estimation is directly integrated into it. We primarily compare this baseline with our best configurations (line 10 and 11). We focus on the shaded results as they are the most informative for demonstrating our method.

**General Noise.** Our methods (both local and global) outperforms the state-of-the-art method (Jie et al., 2019) by a large margin in three datasets (En, Sp, Du, line 10 and 11 v.s. 6), showing the effectiveness of our approach. We observe the oracle $\tau$ does not necessarily give the best performance and an over estimate of confidence could leave a better performance. Ablation results without calibration further show the effectiveness of our calibration methods (line 10 v.s. 14). We note that the CoNLL dataset is an exception where the calibration slightly hurts performance. Otherwise the improvements with calibration is clear in the other 7 datasets.

**Distant Supervision.** Our method outperforms AutoNER and LRNT without pre-trained language models. Reasons that we are worse than BOND (line 16 v.s. 6) are: (a) many implementation aspects are different, and it is (currently) challenging to transplant their settings to ours; (b) they use multiple tailored techniques for distantly-labeled data (e.g., the adversarial training), while ours is more general-purpose. Though our method does not outperform BOND, it still outperforms AutoNER and

LRNT (under the setting all without pretrained model, line 10 and 11 v.s. 7 and 8) and shows promising gain.

### 3.3 Further Analysis

We conduct more detailed experiments on the general noise setting for more in-depth understanding.

**Training Dynamics (Figure 3(a)).** As the model converges, as clean data converge faster, the confidence gap between the clean and the noisy is larger, thus the two are more confidently separated, so both noise detection F1 and dev F1 increase.

**Noise Rate Search (Figure 3(b)).** Our method consistently outperforms baseline without confidence estimation. Lines tend to be higher at the right side of the figure, showing an over-estimate of noise tends to give better performance.

**Level of Noise (Figure 3(c)).** In many real-world scenarios, the noise w.r.t. precision is more constant and it is the recall that varies. So we simulate the level of noise with different recall (lower recall = larger noise ratio). Our method outperforms baselines in all ratios and is particularly effective under a large noise ratio.

**Case Studies (Figure 4).** The top three cases give examples of how our method detects: (1) false negative noise when an entity is not annotated, (2) entities with wrong boundaries and (3) wrong entity types. The last example (case 4) gives a failure case when the model treats some correct tags as noise due to our over-estimate of noise (for better end performance).

## 4 Related Works

State-of-the-art NER models (Ma and Hovy, 2016; Lample et al., 2016; Devlin et al., 2018) are all under the traditional assumption of clean data annotation. The key motivation of this work is the intrinsic gap between the clean data assumption and noisy real-world scenarios. We believe that the noisy label setting is fundamentally challenging in NER and all related supervised learning tasks.

Previous works on NER with noise could be organized into two threads: (a) some works treat this task as learning with missing labels. Bellare and McCallum (2007) propose a missing label CRF to deal with partial annotation. Jie et al. (2019) propose a self-training framework with marginal CRF to re-annotate the missing labels. (b) other works treat missing labels as noise and try to avoid them in the training process. For example, Mayhew et al. (2019) train a binary classifier supervised by entity ratio to classify tokens into entities and non-entities.

A widely-used way to collect NER annotations is distant supervision, which consequently becomes an important source of noise. Peng et al. (2019) formulate this task as the positive-unlabeled (PU) learning to avoid using noisy negatives. AutoNER (Shang et al., 2018) trains the model by assigning ambiguous tokens with all possible labels and then maximizing the overall likelihood using a fuzzy LSTM-CRF model. Cao et al. (2019b) and Yang et al. (2018) try to select high-quality sentences with less annotation errors for sequential model. Liang et al. (2020) leverage pre-trained language models to improve the prediction performance of NER models under a self-training framework.

Our inspiration of confidence estimation comes from the so-called memorization effect observed in the computer vision (Jiang et al., 2018; Han et al., 2018a; Arazo et al., 2019). It observes that neural networks usually take precedence over noisy data to fit clean data, which indicates that noisy data are more likely to have larger loss values in the early training epochs (Arpit et al., 2017). In this work, we leverage it to estimate the confidence scores of labels.

## 5 Conclusion

In this work, we propose a calibrated confidence estimation approach for noisy-labeled NER. We integrate our method in an LSTM-CRF model under a self-training framework. Extensive experiments demonstrate the effectiveness of our approach. Our method outperforms strong baseline models in a general noise setting (especially for larger noise ratios), and shows promising results in a distant supervision setting.

## 6 Acknowledgments

We thank all anonymous reviewers for their helpful comments. This work is supported by Alibaba Group through Alibaba Research Intern Program and AZFT Joint Lab for Knowledge Engine.

## References

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. 2019. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321.

Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18.

Kedar Bellare and Andrew McCallum. 2007. Learning extractors from unlabeled text using relevant databases. In *Sixth international workshop on information integration on the web*.

Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019a. Low-resource name tagging learned with weakly labeled data. 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 261–270, Hong Kong, China. Association for Computational Linguistics.

Yixin Cao, Zikun Hu, Tat-Seng Chua, Zhiyuan Liu, and Heng Ji. 2019b. Low-resource name tagging learned with weakly labeled data. *arXiv preprint arXiv:1908.09659*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab@ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the workshop on noisy user-generated text*, pages 146–153.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018a. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8535–8545.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018b. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809. Association for Computational Linguistics.

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

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

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 *ICML*.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 729–734. ACL.

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

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

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.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1064–1074.

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

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

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Tetsuro Sasada, Shinsuke Mori, Tatsuya Kawahara, and Yoko Yamakata. 2016. Named entity recognizer trainable from partially annotated data. *Communications in Computer and Information Science*, 593:148–160.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064.

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

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, 2:93–128.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly Supervised NER with Partial Annotation Learning and Reinforcement Learning. *COLING*, pages 2159–2169.

# A Dataset Processing

## A.1 Artificially Perturbed Dataset

The gold annotations of training data are perturbed by lowering the recall and precision following Mayhew et al. (2019). Firstly, we randomly select an entity from the whole entity set and tag all of its occurrences to 'O'. We repeat this operation until the recall decreases to 0.5. Then, we randomly tag some tokens/spans to a random entity label to decrease the precision to 0.9. The detailed data statistics are shown in Table 2.

## A.2 Distarntly Supervised Dataset

All distantly supervised datasets in our experiments are the same as those in Liang et al. (2020). The distant labels are generated by external knowledge bases (e.g. Wikidata Vrandečić and Krötzsch, 2014) and gazetteers collected from multiple online resources. Specifically, the entity candidates are first detected by POS tagger (NLTK Loper and Bird, 2002). Next, the ambiguous candidates are filtered out by the Wikidata query service. Then, they match the entities with words in multi-resources gazetteers to get their entity types. Additional rules are used to get the entity labels of the unmatched tokens. The detailed data statistics are shown in Table 2.

# B Implementation Details

## B.1 Model Structure and Implementation

For all the experiments with LSTM, we use the same word embeddings as Lample et al. (2016). We use the character-level LSTM with hidden size 25 to produce character-level word embeddings. The concatenation of the two embeddings are fed into BiLSTM with hidden size 100. We also apply the dropout (Srivastava et al., 2014) between layers, with a rate of 0.5. The model is optimized using Stochastic Gradient Descent (Robbins and Monro, 1951) with a learning rate of 0.01.

For experiments with BERT, we use the BERT-base (Devlin et al., 2018) as our encoder. The implementation is based on the codebase HuggingFace Transformers (Wolf et al., 2020). The dropout rate is set to 0.2. The model is optimized using Adam (Kingma and Ba, 2014) with an initial learning rate of 3e-5.

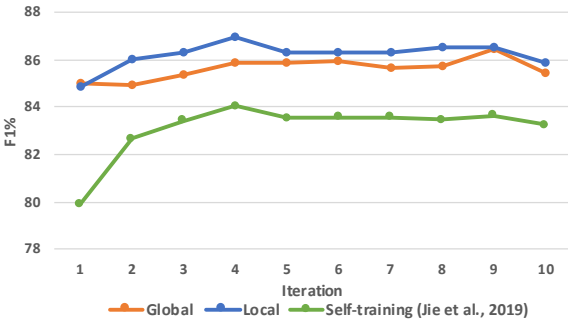



Figure 5: Results of self-training.

## B.2 Hyper-Parameters

There are two important hyper-parameters in our model as the positive noise rate $\tau_p$ and the negative noise rate $\tau_n$. Based on our observation, the initial noise rates are various in different datasets. However, since our model has the ability to handle the noise, the noise rates are relatively stable after the first iteration of self training. Therefore, we empirically set $\tau_p$ and $\tau_n$ to 0.005 and 0.15 for all experiments from the second iteration. For the first iteration, we report the results of two strategies as follows:

**Oracle.** 'Oracle' means that we use the gold noise ratio (unavailable in real settings) of positive noise rate $\tau_p$ and negative noise rate $\tau_n$. The strategy is only applicable for artificially perturbed datasets since the complete annotation is known.

**Searched.** 'Searched' means that we search the two hyper-parameters for best performance on the development set. We search two parameters separately since we assume $\tau_n$ and $\tau_p$ are independent. The search ranges from 0.0 to 0.2 with an interval of 0.01. We determine the two parameters with the best development result on different datasets.

# C Analysis of Self Training

The self training is borrowed from Jie et al. (2019) and not our main contribution. However, to be self-contained, we also report the results of self training in Figure 5. Our method (both local and global) outperforms the baseline by a large margin at the first iteration, which indicates we have a better base model of handling noise. Also, all curves raise in the first several iterations and maintain stable relatively in the subsequent iterations.

| Dataset | Training | | Dev | | Test | |
|---|---|---|---|---|---|---|
| | #entity | #sent | #entity | #sent | #entity | #sent |
| English | 23,499 | 14,041 | 5,942 | 3,250 | 5,648 | 3,453 |
| Spanish | 18,796 | 8,322 | 4,338 | 1,914 | 3,559 | 1,516 |
| German | 11,851 | 12,152 | 4,833 | 2,867 | 3,673 | 3,005 |
| Dutch | 13,344 | 15,806 | 2,616 | 2,895 | 3,941 | 5,195 |
| CoNLL | - | 14,041 | - | 3,250 | - | 3,453 |
| Tweet | - | 2,393 | - | 999 | - | 3,844 |
| Webpage | - | 385 | - | 99 | - | 135 |
| Wikigold | - | 1,142 | - | 280 | - | 274 |

Table 2: Statistics of datasets.