UseClean: learning from complex noisy labels in named entity recognition

Jinjin Tian and **Kun Zhou** and **Meiguo Wang** and **Yu Zhang** and **Benjamin Yao** and **Xiaohu Liu** and **Chenlei Guo**
Alexa AI, Amazon

Abstract

We investigate and refine denoising methods for NER task on data that potentially contains extremely noisy labels from multi-sources. In this paper, we first summarized all possible noise types and noise generation schemes, based on which we built a thorough evaluation system. We then pinpoint the bottleneck of current state-of-art denoising methods using our evaluation system. Correspondingly, we propose several refinements, including using a two-stage framework to avoid error accumulation; a novel confidence score utilizing minimal clean supervision to increase predictive power; an automatic cutoff fitting to save extensive hyper-parameter tuning; a warm started weighted partial CRF to better learn on the noisy tokens. Additionally, we propose to use adaptive sampling to further boost the performance in long-tailed entity settings. Our method improves F1 score by on average at least $5 \sim 10\%$ over current state-of-art across extensive experiments.

1 Introduction

Named Entity Recognition (NER) aims to recognize mentions of rigid designators from text belonging to predefined semantic types such as a person, location, organization, etc. NER not only acts as a standalone tool for information extraction (IE), but also plays an essential role in a variety of natural language processing (NLP) applications such as text understanding, information retrieval, automatic text summarization, question answering, machine translation, and knowledge base construction, etc. Recent progress in deep learning has significantly advanced NER performances (e.g. (Huang et al., 2015; Lample et al., 2016; Li et al., 2020a)). However, in the presence of noisy labels, training DNNs is known to be vulnerable to noisy labels because the significant number of model parameters allow DNNs easily overfit to even corrupted labels. This problem first raised attention in computer vision (CV): (Zhang et al., 2021a) demonstrated that DNNs can easily fit an entire training dataset with any ratio of corrupted labels, which eventually resulted in poor generalizability on a test dataset. Unfortunately, popular regularization techniques, such as data augmentation, weight decay, dropout, and batch normalization do not completely overcome the overfitting issue caused by noisy labels.

Many endeavors have been put into handling noisy labels. Note that this is a fundamentally different problem than general feature-level noise (Zhang and Zhou, 2023; Zheng et al., 2021; Chen et al., 2023; Wang et al., 2021). Except for the specific techniques in certain science domains (Feng et al., 2023), most of those methods are first designed for computer vision or instance-level classification tasks in NLP like text classification. Denoising methods in the NER domain are generally under-explored and rendered harder: for NER, only correct detection of both the entity boundary and entity class are rendered as one correct prediction. Therefore, the label noise in NER is more complex than those in CV or text classification. For example, human annotators could produce mis-specified entity boundaries; other automatic labels generation like distant supervision (Liang et al., 2020) from the dictionary or database often generate incomplete annotations, meaning some entity words are wrongly named as non-entity simply because they are not recorded in the database; others like transfer learning or domain adaptation (Lee et al., 2017; Raghuram et al., 2022; Li and Metsis, 2022) from one domain to another domain could cause wrongly labeled classes for many entity words, as same words could have different semantic types in different domains.

Due to the lack of clean data resources, the majority of denoising literature is unwilling to use any clean validation data or anchor points for denoising, regardless of the fact that most of them often require massive computation cost or extensive hyper-parameter tuning (Song et al., 2022).

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 120–130
September 11–12, 2023.

In fact, those supervision-free methods often suffer from error propagation as the error incurred by false correction/filtering will be accumulated due to lack of supervision, especially when the number of classes or the number of mislabeled examples is large (Shu et al., 2019). To overcome those obstacles, maintaining multiple DNNs or training a DNN in multiple rounds is frequently used (e.g. (Wang et al., 2019; Northcutt et al., 2021)), but these approaches significantly degrade the efficiency of the learning pipeline (Song et al., 2022). In industry-level applications, meta gold datasets (i.e. high-quality/clean datasets) are commonly available, to guarantee direct and reliable evaluation of methods and therefore stable and supreme user experience; also the amount of available data rapidly increases in big companies. More attention should be paid to how to best design and leverage the meta gold dataset to do efficient, effective, and stable label denoising.

Motivated by the above, in this paper, we study the label denoising problem in NER, and we contribute in the following three aspects:

- We build a thorough evaluation system via summarizing all possible noise types and noise generation schemes in NER domain, which was before lacked in the domain. Through this system we find out that the baseline methods [1] are already agnostic to some noise types; while for the other, the noise rate influences the effectiveness of denoising rather than noise type.
- We find out that the current state-of-art denoising method is only effective in very limited noise cases, and the time expensive self-training is often unnecessary due to error propagation. Through careful ablation study, we pinpoint that the true bottleneck of its effectiveness is reliable sample selection.
- We propose an effective and efficient method that leverages minimal clean data to do sample selection and apply weighted semi-supervised learning with a warm start. Under our designed fair comparison [2], our method stably outperforms other state-of-the-art methods across the broad types of simulated noises by a large margin, as well as on realistic data augmentation generated noise. We provide guidelines on further boosting the performance of our method in different application scenarios.

In the following, we will introduce the related work in more detail in Section 2 and provide a formal problem and method description in Section 3. We describe our experiment setting and corresponding results in Section 4, where we also provide a careful ablation study to narrow down the bottleneck of the current state-of-arts method. In the end, we summarize our findings and contributions and some possible future directions in Section 5.

2 Related work

Learning on noisy labels Most of the denoising methods are designed for computer vision (Song et al., 2022), that is, instance level classification. They can generally be categorized into the following four categories: 1) noise modeling: adding a noise adaptation layer at the top of an underlying DNN to learn the transition between clean and noisy labels, e.g.(Chen and Gupta, 2015; Sukhbaatar et al., 2015; Goldberger and Ben-Reuven, 2017)); 2) regularization: enforcing a DNN to overfit less to false-labeled examples explicitly or implicitly, e.g. (Pereyra et al., 2017; Zhang et al., 2018; Menon et al., 2020; Xia et al., 2021; Wei et al., 2021); 3) sample reweighting: adjusting the loss value according to the trust-level of a given sample, e.g. (Wang et al., 2017; Chang et al., 2017; Zhang et al., 2021b; Shu et al., 2019); 4) sample selection: identifying true-labeled examples from noisy training data via multi-network or multi-round learning, e.g. (Han et al., 2018; Jiang et al., 2018; Yu et al., 2019; Wang et al., 2018; Li et al., 2020b; Zhou et al., 2020; Berthelot et al., 2019). From prior work and our investigation, we generally note that, noise modeling type of methods often estimate the transition matrix with large error when only noisy training data is used or when the noise rate is high; regularization type of methods often introduce sensitive model-dependent hyper-parameters and therefore hard to stably work in practice; sample reweighting is often more useful for instance level classification, which is not the case in NER problem domain where often a graphical model is adopted for classification; sample selection is well motivated and works well in general, also its has more interpretability and light-weights.

[1] We call methods designed for clean NER data set as baseline methods, and particularly, we choose bert-CFR as our main baseline model of consideration due to its SOTA performance (Lample et al., 2016).

[2] We include the minimal clean data into the training set for all the methods for a fair comparison.

For industry-level application considerations: we hope to seek solutions that are more lightweight, stable, and easy to tune. Therefore we focus on the line of methods using sample selection.

Semi-supervised learning for NER task An inherent limitation of sample selection is to discard all the un-selected training examples, thus resulting in a partial exploration of training data. To exploit all the noisy examples, researchers have attempted to combine sample selection with other orthogonal ideas. The most prominent method in this direction is combining a specific sample selection strategy with a specific semi-supervised learning model (He et al., 2023; Dong et al., 2021). For example, the most promising method in this direction is combining a specific sample selection strategy with a specific semi-supervised learning model like Partial CRF (Tsuboi et al., 2008).

3 Method: UseClean

Our method UseClean is built upon a well-known NER modeling called Conditional Random Field (CRF). Specifically, consider a sentence of words $\boldsymbol{u} : [u_1, \ldots, u_s]$, and a corresponding sequence of tags $\boldsymbol{y} : [y_1, \ldots, y_s]$, where $y_i \in \mathcal{E} := \{1, \ldots, K\}$, CRF (Lample et al., 2016) models the conditional probability of $\boldsymbol{y}$ given $\boldsymbol{u}$ as:

$$p(\boldsymbol{y}|\boldsymbol{u}) \propto \sum_{1 \le i \le s} (T_{y_{i-1}, y_i} + A_{i, y_i}) \in \mathbb{R} \quad (1)$$

$$\text{where } \boldsymbol{A} = Linear(\boldsymbol{h}) \in \mathbb{R}^{s \times K}; \quad (2)$$

$$\boldsymbol{h} = Encoder(\boldsymbol{u}) \in \mathbb{R}^{s \times m}; \quad (3)$$

$$\boldsymbol{T} \in \mathbb{R}^{K \times K}. \quad (4)$$

Here $\boldsymbol{h}$ denotes the encoder hidden representation, $Linear(\cdot)$ denotes a linear layer that converts $\boldsymbol{h}$ into the network estimation for the possibility of y_i at word i given utterance $\boldsymbol{u}$; and the transition score T_{ij} to model the transition from i-th label to j-th for a pair of consecutive time steps, and it is position independent. Dynamic programming can be used efficiently to compute T and inference optimal tag sequences (Sutton et al., 2012).

In the following, we will introduce our two-stage method UseClean built upon this encoder-CRF model. Figure 1 shows the whole working flow of our UseClean method.

3.1 Clean anchor: a better confidence score

NLNCE (Liu et al., 2021) uses the so-called memorization effect observed in computer vision (Arpit et al., 2017; Zhang et al., 2021a). 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. However, we observe that this is not generally true (see the Figure 2 for examples), which in turn leads to many wrong selection and also error accumulation.

Therefore, we propose to use a two-stage framework that uses a little clean supervision to reduce wrong selection and also error accumulation. Specifically, given all the training data, we sample a small portion (around 1-3%) and annotate it with clean labels, then we train a BERT-CRF model on this small gold data. We call this model the clean anchor model.

Then we apply the clean anchor model on the rest of the training data and compute two choices of confidence scores for i-th token in utterance $\boldsymbol{u}$. The marginal probability based score called `Map` from (Liu et al., 2021):

$$r_i = p_{\text{anchor}}(y_i|\boldsymbol{u}) = \alpha_i \beta_i / Z, \quad (5)$$

which measures how likely the i-th token is labeled y_i under the clean anchor model, where β is the backward variable and can be computed with the Backward algorithm; and the logit value differences based score `Diff`:

$$d_i = \max_{j \in [K]} \{A_{i,j}\} - A_{i, y_i}, \quad (6)$$

which measures the gap between the logit of the observed label and the predicted label. We observe no universal winner of those two scores in our extensive experiments, therefore we report the best over them.

Adaptive Sampling. Under the existence of the class imbalance [3], it is very likely that our random sampled small clean dataset does not contain certain tail entity types, and therefore leading to bad separation of clean and noisy tokens in them. To mitigate this effect, we consider a constrained sampling method that tries to sample more from the tail entities: sampling only from utterance that contains at least one tail entity (we define the entities that constitute the tail 20% quantile as the tail entities). In this paper, if a dataset appears to have long

[3] Other popular methods for combating imbalance issue includes the logit adjustment method (Menon et al., 2021), but we did not find it was able to improve the downstream NER performance in our setting.

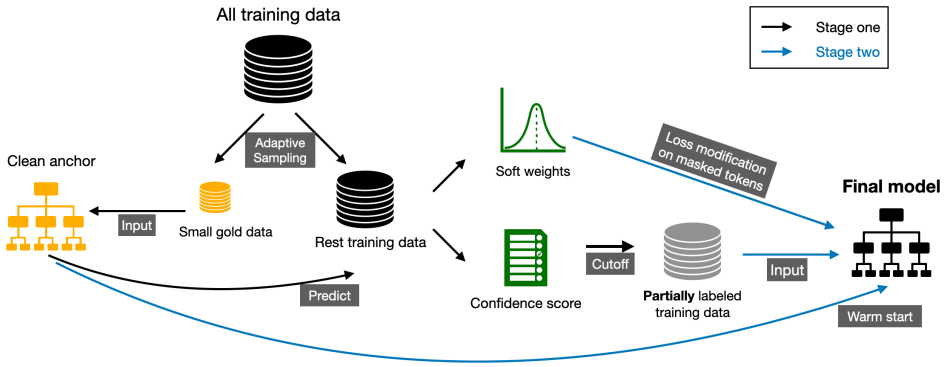


Figure 1: A demonstration of the working flow of UseClean model.

tail entity distribution[4], we will adopt an adaptive sampling scheme: consider both random sampling and constrained sampling and report the best over them.

3.2 FitMix: automatic sample selection

From Figure 2 we can see that, the confidence score of clean and noisy seems to follow a Gamma-Gaussian mixture distribution, where the noisy component follows the Gaussian distribution and the clean component follows the Gamma distribution. So we propose to model the confidence score s as the following:

$$s \sim wf + (1-w)g, \quad \text{where } w \in [0,1] \tag{7}$$

$$f \sim \Gamma(\alpha, \beta),\ g \sim N(\mu, \sigma),\ , \tag{8}$$

and fit all the parameters $(w, \mu, \sigma, \alpha, \beta)$ using Expectation-Maximization algorithm. Then with the fitted parameters $(\widehat{w}, \widehat{\mu}, \widehat{\sigma}, \widehat{\alpha}, \widehat{\beta})$, we can compute the theoretical F1 given a cutoff C in closed form:

$$F_1(C) = \frac{\widehat{w}\Big(1 - \Gamma_{\widehat{\alpha},\widehat{\beta}}(C)\Big) + (1-\widehat{w})\Big(2 - \Phi_{\widehat{\mu},\widehat{\sigma}}(C)\Big)}{(1-\widehat{w})\Big(1 - \Phi_{\widehat{\mu},\widehat{\sigma}}(C)\Big)}.$$

We select C such that $F_1(C)$ is maximized and treat all tokens that have $s > C$ as non-trustworthy.

3.3 Warm weight: learning on noisy tokens

After we do the sample selection, we treat all non-trustworthy tokens as unlabeled and use the idea of semi-supervised learning. Liu et al. (2021) simply sum over all the token sequences that are compatible with the trusted annotations. Specifically, denoting the trusted annotation sequence as $\boldsymbol{y}_p$, from it we can derive a set of all possible complete label sequences that are compatible with the incomplete label sequence, and let us call this set $\mathcal{C}(\boldsymbol{y}_p)$, then semi-supervised loss function can be written as

$$L(\boldsymbol{\theta}) = -\log \sum_{\widetilde{\boldsymbol{y}} \in \mathcal{C}(\boldsymbol{y}_p)} p_\theta(\widetilde{\boldsymbol{y}}|\boldsymbol{u}) \tag{9}$$

Inspired by Jie et al. (2019) for better modeling of NER with incomplete annotations, we instead use a weighted version:

$$L_{weight}(\boldsymbol{\theta}) = -\log \sum_{\widetilde{\boldsymbol{y}} \in \mathcal{C}(\boldsymbol{y}_p)} q_{\mathcal{D}}(\widetilde{\boldsymbol{y}}|\boldsymbol{u}) p_\theta(\widetilde{\boldsymbol{y}}|\boldsymbol{u}), \tag{10}$$

where $q_{\mathcal{D}}$ represents the true data distribution. We estimate $q_{\mathcal{D}}$ as q_{anchor}, which is the distribution computed using our trained clean anchor model. As the clean anchor model is trained on clean data, therefore we believe these weights represent some level of prior information of the underlying true label sequence distribution. By putting more probability mass on a path that is close to the true path, we can guide the model to quickly learn the essential parameters that can correctly predict the true path in the inference stage.

4 Experiments

4.1 Datasets

We consider three datasets for evaluation throughout this paper: an Alexa dialog dataset called Massive (FitzGerald et al., 2022), which contains around 16K samples, and 55 entity types across 18 domains; a popular benchmark dataset CoNLL03 (Sang and De Meulder, 2003), which contains around 20K samples, 4 entity types in News domain; and a Wikipedia dataset Wikigold (Balasuriya et al., 2009), which contains around 1.8K samples over 4 entity types in Wikipedia domain.

[4]Long tail distribution means having many classes of small sizes.

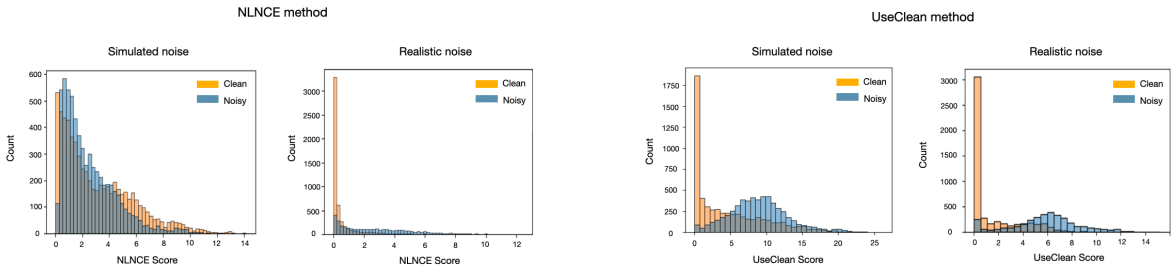


Figure 2: The distribution of confidence score for both simulated noise ("bias" type, detailed description in Section 4) and realistic transfer learning generated noise (detailed description in Section 4), using NLNCE method and our UseClean method. For NLNCE, we plot the distribution of confidence score at epoch 2, we can see that the clean and noisy samples are highly overlapped, i.e. the "early learning" phenomena does not hold true. On the other hand, our UseClean method has better separation of clean and noisy.

Synthetic noisy datesets Given a dataset with labels of good quality, we can treat its original labels as truth and manually perturb it to generate noise. In this paper, we consider randomly selecting $x\%$ of utterance, and random select $\max\{1, 0.2\#\text{entities}\}$ entities (if there are any), and perturb their labels by the different noise generation schemes showed in Table 1. We mainly focus on the high noise rate regime: i.e. $70\%, 100\%$ utterance level noise. We point out that even with the same utterance level noise rate, the word level noise rate can vary a lot for different noise types. For example, `shift` and `shrink` noise types often have much lower word level noise rates compared with others, this is due to the fact that `shift` and `shrink` can only happen on entities with multiple words, while the others can happen on any entity. Due to the size imbalance between entity and nonentity words, we compute the word-level noise rate for entity and nonentity separately. In the following, we use the summation of the entity and nonentity word-level noise rate as the total word-level noise rate for simplicity.

Realistic noise We also consider more realistic noisy label generation. In practice, many cheap labels are generated either from distant supervision or transfer learning.

- *Distant supervision:* We consider three datasets including Massive (FitzGerald et al., 2022), CoNLL03 (Sang and De Meulder, 2003), Wikigold (Balasuriya et al., 2009). In this setting, the distantly supervised tags for CoNLL03 and Wikigold are generated by the dictionary following BOND (Liang et al., 2020), while for massive, we provide distant supervision simply using our own defined dictionary.

- *Transfer learning:* We consider two datasets: Massive (FitzGerald et al., 2022) and CoNLL03 (Sang and De Meulder, 2003). For CoNLL03, we consider transferring the Wikigold dataset to it, as they share exactly the same entity types. To do the transfer, we directly learn a model on Wikigold and predict it on CoNLL03. For Massive, we use data in 9 domains of massive data and transfer them to the rest 9 domains. To make the most meaningful transfer, we compute this domain by domain entity types overlapping matrix, where each cell indicates how many entities types a pair of two domains share. Then we intentionally split the domains into source and target such that the domain pairs with high overlaps are separated, and hence model learned on source domains can have more knowledge transferable to the target domains.

4.2 Methods for comparison

For **Baseline**, we follow the implementation of the neural-CRF model proposed in (Lample et al., 2016) without any denoising steps. Particularly, it models the tag sequence as a linear-chain conditional random field, where only subsequent tags have an edge. Also, we consider the following three NER denoising methods on top of the baseline, which we believe are the most competitive methods in the literature. **CoReg** (Zhou and Chen, 2021) propose a regularization based NER denoising method called CoReg, where the regularization term is based on model agreement. **NLNCE** (Liu et al., 2021) utilize the early learning phenomena and select the noisy tokens via gradually truncating

	noise type	explanation	example: "show me the meetings held last month"
truth	-	-	[O, O, O, S-event_name, O, B-date, I-date]
over/in-complete	miss	label an entity word as nonentity	[O, O, O, S-event_name, O, O, B-date]
	over	label a nonentity word as some random entity	[O, S-person, O, S-event_name, O, B-date, I-date]
boundary error	shift	for an entity contains multiple words, shift its boundary to the left or right by one word.	[O, O, O, S-event_name, B-date, I-date, O]
	extend	for an entity, extend its boundary to the left or right by one word.	[O, O, O, B-event_name, I-even_name, B-date, I-date]
	shrink	for an entity contains multiple words, shrink its boundary from the left or right by one word.	[O, O, O, S-event_name, O, S-date, O]
class error	swap	for an entity, change its class to some other random entity	[O, O, O, S-event_name, O, B-person, I-person]
	bias	for an entity, change its class to some particular entities according to a transition matrix	[O, O, O, S-event_name, O, B-time I-time]

Table 1: Summarized synthetic noise generation scheme.

the samples with a large loss, then it uses the uniform partial CRF to relearn the noisy tokens. As for alternative automatic cutoff fitting methods, we also consider replacing our FitMix with the method from (Pleiss et al., 2020) (and call it **CutFake**), which manually assigns several tokens with labels of an additional "fake" class and uses the lower tail of their confidence scores for sample selection.

Implementation details In this paper, we consider using two types of encoders: one is the BiLSTM encoder (Huang et al., 2015) and the other one is the BERT encoder (Devlin et al., 2019). For BiLSTM, we use hidden dimension 200, SGD optimizer with learning rate 0.01; for BERT, we use the default hidden dimension 768, and the default optimizer with learning rate 2e-5. We use batch size 10 for both encoders and it works well. For BiLSTM encoder, we train for 30 epochs, and for BERT, we train for 20 epochs. We split the whole dataset into train/dev/test subsets if such splitting was not provided by the original dataset, and we keep the sample size ratio of train/dev/test as 2:1:1. We output the model with the best dev F1 score.

4.3 Main results

Noise Type v.s. Noise Rate Table 2 shows how the sample selection based methods work in different synthetic problem settings. Specifically, we show the results of different methods confronting one specific type of noise respectively, to investigate our initial questions about whether methods' performances depend on noise type and noise rate. For a more realistic mixed noise, we refer to Table 3. To get a sense of upper bound performance, we also consider an oracle method called **onlyClean**, where we replace the sample selection step in the original **NLNCE** method by directly telling it which is truly clean and noisy. Here Table 2 summarizes F1 score of the **baseline**, and the differences from it of denoising methods **NLNCE**, **UseClean** and **onlyClean**. The significant positive differences are marked as green, while the significant negative ones are marked as red, and the rest are marked as grey. We can see that, after doing sample selection correctly, the current sample selection based method can indeed improve a lot over the baseline, even though it still has some gaps from the fully clean supervised case in the high noise rate regime. Overall, we can observe that sample selection based methods perform differently under different noise types and noise rates. Basically, for `over` type of noise, the baseline's performance is not influenced much. We suspect that this is due to the fact that the `over` type of noise is kind of unnatural, as it randomly selects a nonentity word and assigns a random entity to it. Such nonentity words would often be meaningless words like 'the', 'a' etc, and the CRF model can autocorrect such unnatural mistakes as it optimizes over a tag sequence as a whole. Another similar case is the `swap` noise type: where we find out that baseline can already perform relatively well compared to other noise types of similar noise rates. For the rest more natural noise types, we can observe that the effectiveness of the sample selection based idea depends more on the word-level noise rate, rather than the noise type. Specifically, it is less effective when the noise rate is low. From this reason, we can see that for `shift` and `shrink` type of noises, sample selection based methods generally do not help as much as they do in the other noise types, since `shift` and `shrink` tend to have lower word-level noise rate comparing to other noise types. In the rest of the paper, we will focus on the `miss`,

utterance level noise rate	0%	30%					100%				
Noise Type / Methods		Baseline	NLNCE	UseClean	onlyClean	*Word leve noise rate*	Baseline	NLNCE	UseClean	onlyClean	*Word leve noise rate*
Miss		76.62	+0.15	+1.03	+4.39	11%	35.26	+12.17	**+21.47**	**+37.55**	50%
Over		80.97	+0.60	-0.38	+0.43	17%	77.79	+0.45	+0.82	+3.41	41%
Shift		79.32	-0.01	+0.85	+0.93	6%	70.83	-0.40	+1.57	+8.78	20%
Extend	80.85	74.53	+0.13	+4.34	+6.84	18%	42.4	+5.56	**+24.36**	**+35.35**	52%
Shrink		77.38	-0.25	+0.66	+3.13	8%	60.49	-1.56	+2.29	+15.41	33%
Swap		78.74	+0.32	-0.51	+2.85	21%	56.33	+0.03	+0.95	+14.93	67%
Bias		75.15	-0.08	+2.05	+5.21	15%	38.75	+0.43	**+20.1**	**+33.09**	49%

Table 2: The performance of baseline, NLNCE, UseClean, onlyClean over different noise types and noise rates. The significant positive differences are marked as green, meaning that the method improves over 1% over baseline, while the significant negative ones are marked as red, meaning the method is even worse than baseline by over 1%; the rest are marked as grey. The extremely positive ones are marked in bold green. We use the BERT encoder throughout all those experiments.

`extend`, `bias` type of noises under the high noise rate regime (70%, 100% utterance level noise rate), where we know sample selection kind of idea has the potential to help much.

Broad synthetic and realistic noisy settings Table 3 summarizes the results for a more complete collection of denoising methods and more realistic noisy datasets. For all the noisy datasets, we report their summed word-level noise rate over nonentities and entities. We can see that, the word-level noise rate on the realistic noisy dataset tends to be pretty high, therefore suitable for applying our sample selection based method. We can see that, NLNCE and CoReg can improve over the baseline a bit under very limited cases, while our method UseClean can improve over the baseline by a large margin over all those noisy datasets.

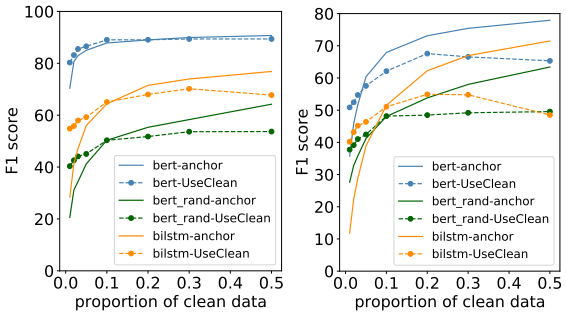
4.4 Ablation Study

	miss		extend		bias	
utterance level noise rate	70%	100%	70%	100%	70%	100%
word level noise rate	34%	50%	39%	52%	34%	49%
(oracle)	65.17	52.51	70.75	63.29	58.73	56.43
adapt (oracle)	65.17	54.44	72.15	66.40	63.03	56.43
warm (oracle)	67.58	54.21	71.63	64.92	61.14	58.35
weight (oracle)	67.31	55.08	69.52	65.15	59.12	58.28
UseClean (oracle)	**67.56**	**56.73**	**73.02**	**66.77**	**65.05**	**58.85**
UseClean (fitmix)	67.37	54.10	70.29	61.04	64.73	55.80

Table 4: Ablation Study for our method.

In Table 4 we show results for the ablation study, to see how each component in our method contributes to the final performance. Here the first line represents random sampling which is our base, and adapt means only uses adaptive sampling; warm means only uses warm start; weight means only uses weighted semi-supervised learning. For fair comparison without the confounding effect from cutoff fitting, we simply use the oracle cutoff: we fit a logistic regression of true clean/noisy labels with the confidence score, and use the predicted clean/noisy labels as sample selection decisions. We can see that each of these three techniques improves over the base in most cases, and a combination of them, which is adapt + warm + weight improves over the base by about 2-4% over all cases. Finally, our FitMix technique can achieve performance close to the oracle cutoff.

4.5 Further analysis



(a) CoNLL03 data (b) Massive data

Figure 3: The performance with and without the denoising step in UseClean versus different size of clean supervision.

Amount of clean supervision required We would like to explore how much clean data should we require such that it is reasonable to ask. To be more specific, we would like the effectiveness of our method also comes from the denoising part, rather than just the clean data pertaining part. In Figure 3 we compare the F1 score for the clean anchor model and our UseClean model with different proportion of clean dataset. To take account

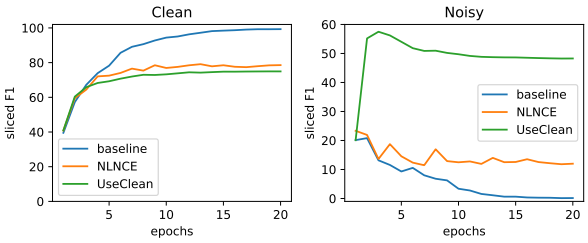
	simulated noise						realistic noise				
	miss		extend		bias		distant supervision			transfer learning	
utterance level noise rate	70%	100%	70%	100%	70%	100%	Massive	CoNLL03	Wikigold	Massive	CoNLL2003
word level noise rate	34%	50%	39%	52%	34%	49%	66%	22%	48%	50%	61%
baseline	58.8	35.26	56.38	42.4	55.04	38.75	42.02	72.76	49.76	50.7	35.88
NLNCE	64.34	47.42	62.94	47.96	58.91	39.18	40.93	72.44	54.27	51.24	42.73
NLNCE*	65.28	49.13	65.56	46.40	58.91	39.81	42.87	74.58	57.62	51.99	44.78
CoReg	48.80	38.91	47.76	41.95	45.30	37.88	41.52	70.64	49.33	52.18	34.34
CutFake	53.85	**54.47**	61.18	58.01	56.14	55.79	53.51	**79.27**	55.48	60.60	51.77
UseClean	**67.37**	54.18	**70.29**	**61.04**	**64.73**	**55.80**	**57.78**	77.31	**68.08**	**61.25**	**76.11**

Table 3: The performance of our method and all the competitors over simulated noise and realistic noise.

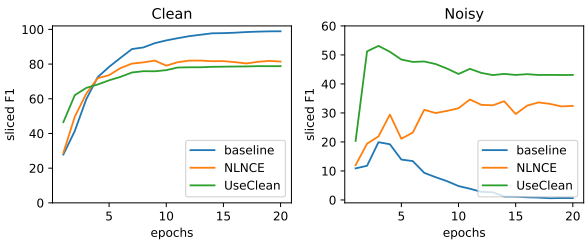
of the influence from dataset, model architecture and pretraining, we consider one simple dataset CoNLL03 and one complex data set Massive; and we consider three different backbone models: `bert` (pretrained BERT model); `bert_rand` (randomly initialized BERT model); `bilstm` (randomly initialized BiLSTM model). Due to the limitation of space, here we only demonstrate the results of one noisy type under the high noise rate regime: the bias type of noise with utterance noise level 100%.

By just looking at the solid lines (i.e. clean anchor model performance), we can see that all lines tend to first rise rapidly and then slows-down, this phenomenon is more evident on this simpler data set CoNLL03 and large pertained encoder. This indicates that large pretrained model is less data-hungry, especially in the easy problem setting. Also, we can see that, augmentation only outperforms non-augmentation when clean data is limited. Therefore, we argue that the reasonable size of clean supervision we require should be less than hundreds of examples. In all our examples, we use 100-200 examples depending on the problem difficulty.

Training dynamics In Figure 4, we plot the sliced F1 score on clean tokens and noisy tokens and also the total F1 score during the whole training process for baseline, NLNCE and UseClean. We can see that, both NLNCE and UseClean can indeed learn on the noisy tokens, while UseClean tends to learn much better on the noisy case without sacrificing too much on the clean cases. For the case where UseClean is much better than NLNCE (i.e. Figure 4(a)), its generalization gap is the smallest among the three methods. For the case where UseClean is a bit better than NLNCE (i.e. Figure 4(b)), we find out that the generalization gap of both NLNCE and UseClean is nearly none, meaning that now NLNCE also does not overfit to noise too much. Still, UseClean learns better on noisy cases. Finally, we also find out that, the sample selection type of denoising method improves over baseline mainly by learning better on the noisy cases, which often at the cost of a certain amount of performance drop on clean cases. This might also explain our findings about why the sample selection type of idea only works in a high noise rate regime.



(a) Bias type noise



(b) Miss type noise

Figure 4: The sliced F1 score on clean tokens and noisy tokens during the whole training process.

5 Conclusion and Discussion

In conclusion, we propose an effective and efficient method called UseClean, which includes a simple two-stage framework to avoid error accumulation, a novel confidence score utilizing minimal clean supervision to increase predictive power in sample selection, an automatic cutoff fitting to save exten-

sive hyper-parameter tuning and finally weighted semi-supervised learning with warm start to learn better on the noisy tokens. Additionally, we propose to use adaptive sampling to construct better clean supervision for a further performance boost. Despite simple, our method improves F1 score by on average at least $5 \sim 10\%$ over current state-of-art without extensive hyper-parameter tuning or heavy computation, and is effective across a broad type of noise types and noise levels.

We admit that most of the performance gain comes from the minimal clean supervision in small gold data. Without it, the SOTA method NLNCE suffers from error accumulation and heavy computation. Still, we argue that the clean supervision we need is very minimal, like just about 100 samples, while the stable improvement and efficiency it can bring is fairly large. In fact, we suspect that it is often necessary to guarantee success in real applications, and how to best construct and leverage clean supervision is nontrivial and important.

Limitations

We admit that the methods for comparison in this paper are not a complete list of the literature, though we arguably claim that they are strong representatives. We omit some methods for now due to their complexity and computation time. It would make our paper a more convincing story if we had also considered the rest established methods like **BOND**(Liang et al., 2020). Also, currently we do the sample selection and semi-supervised learning in a one-pass way, while alternatively an iterative-pass way like active learning (Kong et al., 2021) might be even more effective. Still, one need to be careful about the error propagation during the iterative process.

Even though we point out the importance and potential of designing and leveraging the meta gold dataset, we have not provided a thorough discussion of past endeavors. Particularly, **FilDist** (Onoe and Durrett, 2019) also utilize clean supervision like us, though they also require corresponding noisy labels to fit a binary classifier for sample selection. It would be interesting to see how those methods compare to ours.

References

Devansh Arpit, Stanisław Jastrzębski, 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.

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, Suntec, Singapore. Association for Computational Linguistics.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30.

Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, and Gabriele Tolomei. 2023. The dark side of explanations: Poisoning recommender systems with counterfactual examples. *arXiv preprint arXiv:2305.00574*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Guimin Dong, Mingyue Tang, Lihua Cai, Laura E Barnes, and Mehdi Boukhechba. 2021. Semi-supervised graph instance transformer for mental health inference. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1221–1228. IEEE.

Anqi Feng, Yuan Xue, Yuli Wang, Chang Yan, Zhangxing Bian, Muhan Shao, Jiachen Zhuo, Rao P Gullapalli, Aaron Carass, and Jerry L Prince. 2023. Label propagation via random walk for training robust thalamus nuclei parcellation model from noisy annotations. *arXiv preprint arXiv:2303.17706*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Yunzhong He, Cong Zhang, Ruoyan Kong, Chaitanya Kulkarni, Qing Liu, Ashish Gandhe, Amit Nithianandan, and Arul Prakash. 2023. Hiercat: Hierarchical query categorization from weakly supervised data at facebook marketplace. In *Companion Proceedings of the ACM Web Conference 2023*, pages 331–335.

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. pages 2304–2313. PMLR.

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

Ruoyan Kong, Zhanlong Qiu, Yang Liu, and Qi Zhao. 2021. Nimblelearn: A scalable and fast batch-mode active learning approach. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 350–359. IEEE.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Junnan Li, Richard Socher, and Steven CH Hoi. 2020b. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

Xiaomin Li and Vangelis Metsis. 2022. Spp-eegnet: An input-agnostic self-supervised eeg representation model for inter-dataset transfer learning. In *International Conference on Computing and Information Technology*, pages 173–182. Springer.

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.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021. Noisy-labeled ner with confidence estimation. *arXiv preprint arXiv:2104.04318*.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. 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 2407–2417.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, pages 17044–17056. Curran Associates, Inc.

Jayaram Raghuram, Yijing Zeng, Dolores Garcia, Rafael Ruiz, Somesh Jha, Joerg Widmer, and Suman Banerjee. 2022. Few-shot domain adaptation for end-to-end communication. In *The Eleventh International Conference on Learning Representations*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE*

Transactions on Neural Networks and Learning Systems.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training convolutional networks with noisy labels. In *3rd International Conference on Learning Representations, ICLR 2015*.

Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

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

Ruxin Wang, Tongliang Liu, and Dacheng Tao. 2017. Multiclass learning with partially corrupted labels. *IEEE transactions on neural networks and learning systems*, 29(6):2568–2580.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8688–8696.

Yuli Wang, Ryan Herbst, and Shiva Abbaszadeh. 2021. Electronic noise characterization of a dedicated head-and-neck cancer pet based on czt.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. *arXiv preprint arXiv:1909.01441*.

Hongxin Wei, Lue Tao, RENCHUNZI XIE, and Bo An. 2021. Open-set label noise can improve robustness against inherent label noise. In *Advances in Neural Information Processing Systems*.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2021. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*.

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. 2021a. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Dan Zhang and Fangfang Zhou. 2023. Self-supervised image denoising for real-world images with context-aware transformer. *IEEE Access*, 11:14340–14349.

HaiYang Zhang, XiMing Xing, and Liang Liu. 2021b. Dualgraph: A graph-based method for reasoning about label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9654–9663.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Wenqing Zheng, Edward W Huang, Nikhil Rao, Sumeet Katariya, Zhangyang Wang, and Karthik Subbian. 2021. Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. *arXiv preprint arXiv:2111.04840*.

Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. 2020. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.