Meta Self-Refinement for Robust Learning with Weak Supervision

Dawei Zhu[1], **Xiaoyu Shen**[1], **Michael A. Hedderich**[1,2], **Dietrich Klakow**[1]
[1]*Saarland University, Saarland Informatics Campus, Germany*
[2]*Cornell University, United States*
{dzhu,xshen,mhedderich,dietrich.klakow}@lsv.uni-saarland.de

Abstract

Training deep neural networks (DNNs) under weak supervision has attracted increasing research attention as it can significantly reduce the annotation cost. However, labels from weak supervision can be noisy, and the high capacity of DNNs enables them to easily overfit the label noise, resulting in poor generalization. Recent methods leverage self-training to build noise-resistant models, in which a teacher trained under weak supervision is used to provide highly confident labels for teaching the students. Nevertheless, the teacher derived from such frameworks may have fitted a substantial amount of noise and therefore produce incorrect pseudo-labels with high confidence, leading to severe error propagation. In this work, we propose Meta Self-Refinement (MSR), a noise-resistant learning framework, to effectively combat label noise from weak supervision. Instead of relying on a fixed teacher trained with noisy labels, we encourage the teacher to refine its pseudo-labels. At each training step, MSR performs a meta gradient descent on the current mini-batch to maximize the student performance on a clean validation set. Extensive experimentation on eight NLP benchmarks demonstrates that MSR is robust against label noise in all settings and outperforms state-of-the-art methods by up to 11.4% in accuracy and 9.26% in F1 score.

1 Introduction

Fine-tuning pre-trained language models has led to great success across NLP tasks. Nonetheless, it still requires a substantial amount of manual labels to achieve satisfying performance on many tasks. In reality, obtaining large amounts of high-quality labels is costly and labor-intensive (Davis et al., 2013). For certain domains, it is even infeasible due to legal issues and lack of data or domain experts. Weak supervision is a widely-used approach for reudcing such cost by leveraging labels from weak sources, e,g., heuristic rules, knowledge bases or lower-quality inexpensive crowdsourcing (Ratner et al., 2017; Zhou et al., 2020; Lison et al., 2020). It has raised increasing attention in recent years, and efforts have been made to quantify the progress on weakly supervised learning, like the WRENCH benchmark (Zhang et al., 2021).

Although weak labels are inexpensive to obtain, they are often noisy and inherit biases from weak sources. Training neural networks with weak labels is challenging because of their immense capacity, which leads them to heavily overfit to the noise distribution, resulting in inferior generalization performance (Zhang et al., 2017). Various approaches have been proposed to tackle this challenge. Earlier research focused primarily on simulated noise (Bekker and Goldberger, 2016; Hendrycks et al., 2018), required prior knowledge (Ren et al., 2020; Awasthi et al., 2020) or relied on context-free aggregation rules without leveraging modern pre-trained language models (Ratner et al., 2017; Fu et al., 2020).

Recently, Yu et al. (2021) proposed a contrastive regularized self-training framework that achieved state-of-the-art (SOTA) performance in several NLP tasks from the WRENCH benchmark. It trains a teacher network on weak labels, then iteratively applies the teacher to produce pseudo-labels for training a new student model. To prevent error propagation, it filters the pseudo-labels with the model confidence scores and adds contrastive feature regularization to enforce more distinguishable representations. However, we find that this approach is *effective on easy tasks but fragile on challenging ones*, where the initial teacher model already have memorized a substantial amount of biases with high confidence. Consequently, confidence-based filtering is misleading and all future students will be reinforced with these initial wrong biases from the teacher.

To address this weakness, one strategy is learning to reweight the pseudo-labels with meta learning (Ren et al., 2018; Shu et al., 2019; Wang et al.,

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1043–1058
May 2-6, 2023 ©2023 Association for Computational Linguistics

2020). By this means, sample weights are dynamically adjusted to minimize the validation loss instead of prefixed with potentially misleading confidence scores. Nevertheless, if the initial teacher is weak and mostly produces incorrect pseudo-labels, simply reweighting the labels does not suffice to extract enough useful training signals.

In this paper, we propose Meta Self-Refinement (MSR) to go one step further. The teacher is jointly trained with a meta objective such that the student, after one gradient step, can achieve better performance on the validation set. In each training step, a copy of the current student performs one step of gradient descent based on the teacher predictions. The teacher will then update itself towards the gradient direction that minimizes the validation loss of the student. Finally, the actual student is trained by the updated teacher. In MSR, teacher's predictions are iteratively *refined*, instead of only "reweighted", based on the meta objective. This will enable more efficient data utilization since the teacher still has the opportunity to refine itself to provide the proper training signal, even if its initial output label is wrong. To further stabilize the training, we enhance our framework with confidence filtering when teaching the student and apply a linearly scaled learning rate scheduler to the teacher.

In summary, the main contributions are as follows: **1)** We propose a meta-learning based self-refinement framework, MSR, that allows robust learning with label noise induced by weak supervision. **2)** We analyze and quantify how label noise impacts model predictions and representation learning. We find existing methods become less effective in challenging cases when the label noise can be easily fitted. In contrast, MSR is more stable and learns better representation. **3)** Extensive experiments demonstrate that MSR consistently reduces the negative impact of the label noise, matching or outperforming SOTAs on six sequence classification and two sequence labeling tasks.[1]

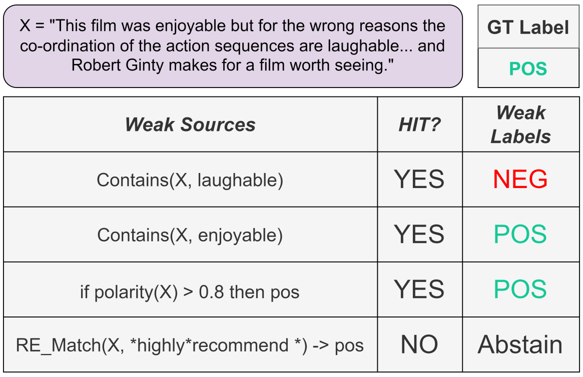
2 Related Work

Learning with Noisy Labels. Learning in the presence of label noise is a long-standing problem (Angluin and Laird, 1988). Zhang et al. (2017) show that deep neural networks can memorize arbitrary noise during training, resulting in poor generalization. Noise-handling techniques - by modeling (Goldberger and Ben-Reuven, 2017; Patrini et al., 2017; Hendrycks et al., 2018) or filtering (Han et al., 2018; Li et al., 2020) the noisy instances - are proposed to conquer the label noise. While being effective, they typically assume that the noise is feature-independent which may oversimplify the noise generation process in realistic settings (Gu et al., 2021; Zhu et al., 2022). Recently, realistic and feature-dependent noise induced by weak supervision has received significant attention. To handle this type of noise, Awasthi et al. (2020) propose an implication loss that jointly denoises the noisy labels and weak sources. Ren et al. (2020) denoise the weak label by considering the reliability of different weak sources and aggregating them into one cleaned label. Zhang et al. (2021) release a benchmark, WRENCH, including various weakly supervised datasets in both text and image domains.

Self-Training. Self-training (Yarowsky, 1995; Lee et al., 2013) is a simple yet effective framework that is commonly used in semi-supervised learning (SSL). It typically trains a teacher model to provide pseudo-labels for the student model. Different methods have been proposed for better generalization (Xie et al., 2020; Zoph et al., 2020; Mukherjee and Hassan Awadallah, 2020). Recently, self-training has been adopted to tackle weak supervision. Karamanolakis et al. (2021) train a teacher network that aggregates weak labels to form high-quality pseudo-labels for the student. Liang et al. (2020); Yu et al. (2021) initialize the teacher model by training a classifier directly on the weak labels, they apply early stopping to prevent this initial teacher from memorizing the label noise. The student is then trained on the highly confident pseudo-labels provided by the teacher. While the core assumption of self-training - that highly confident pseudo-labels are reliable - is generally valid in SSL, it may not be true for feature-dependent noise induced by weak supervision, especially when the noise is easy to learn. In this case, self-training inevitably suffers more from error propagation and fails to train robust models.

Meta-Learning. Recently, different works leveraged meta-learning techniques to develop noise-robust learning frameworks. The idea is to optimize an outer learner (e.g., sample weights) that guides the inner learner (the classifier) to generalize well. Often, a clean validation dataset is used as a proxy

[1] Code is available on: `https://github.com/uds-lsv/msr`



X = "This film was enjoyable but for the wrong reasons the co-ordination of the action sequences are laughable... and Robert Ginty makes for a film worth seeing."

GT Label
POS

Weak Sources	*HIT?*	*Weak Labels*
Contains(X, laughable)	YES	NEG
Contains(X, enjoyable)	YES	POS
if polarity(X) > 0.8 then pos	YES	POS
RE_Match(X, *highly*recommend *) -> pos	NO	Abstain

Figure 1: Sentiment analysis dataset annotated with rule-based weak sources. A weak source is triggered if a specific textual pattern is matched, after which a pre-defined label is then assigned. Otherwise, it abstains. Depending on how many weak sources are triggered, a text may obtain zero, one, or multiple weak labels.

for estimating the generalization performance. Ren et al. (2018) attempt to down weight training samples that increase the validation loss. Shu et al. (2019) employ a neural network to infer such sample weights and show a significant boost on performance under feature-independent noise. Wang et al. (2020) reweight the training samples by their pseudo-labels instead of the original noisy labels. In this work, we aim to leverage meta-learning in a more flexible manner by refining the pseudo-labels instead of reweighting them. Approach-wise, the most related works are (Pham et al., 2021; Zhou et al., 2022) used for semi-supervised learning and model distillation, which also refine the teacher's parameters based on the student feedback. However, they work with samples from clean distributions, while we anticipate the noise memorization effect and enhance our framework with teacher warm-up and confidence filtering to suppress the error propagation.

3 Problem Formulation

Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature and label space, respectively. In standard supervised learning, one is given a clean dataset $\mathcal{D}_c = \{(x_i, y_i)\}_{i=1}^N$, where N is the number of samples. The clean labels y_i are supposed to be annotated by human experts.

In weak supervision, a dataset is labeled by weak sources rather than humans. Weak sources can have diverse forms like lexical rules, knowledge bases, pre-trained models, lower-quality inexpensive crowdsourcing, etc. Figure 1 shows an example of text labeled via weak supervision. Compared to manual annotations, weak labels contain more mistakes. We denote the dataset labeled by weak sources by $\mathcal{D}_w = \{(x_i, \hat{y}_i)\}_{i=1}^N$ where $\hat{y}_i$ is the weak label.[2] Since weak sources might not cover all data, we may have a set of unlabeled data $\mathcal{D}_u$ in addition to $\mathcal{D}_w$. We use $\mathcal{D}_a = \mathcal{D}_w \cup \mathcal{D}_u$ to denote the full set of data. Moreover, as we do not make any assumption on the quality of the weak labels, their distribution can deviate arbitrarily from the distribution of clean labels. Learning with only weak labels can lead to unbounded model errors (Menon et al., 2016; Gu et al., 2021). Hence, following standard practice in weak supervision, we assume the access to a small clean validation set $\mathcal{D}_v = \{(x_i^v, y_i^v)\}_{i=1}^M$ where $M \ll N$. $\mathcal{D}_v$ is used for early stopping, hyper-parameter tuning or meta-learning so that the learned model will not fully overfit the noisy weak labels (Ren et al., 2018; Shu et al., 2019; Zhang et al., 2021).

4 Meta Self-Refinement

We propose a novel meta-learning based framework, named Meta Self-Refinement (MSR), to tackle the label noise induced by weak supervision. In contrast to conventional self-training methods, where the teacher model is fixed after being trained on weakly labeled data, MSR enables the teacher to refine itself based on student performance on the clean validation set, yielding higher-quality labels and more accurate confidence estimates. In this section, we first provide an overview of its training objective (section 4.1), then go into the training details (section 4.2). Figure 2 illustrates the full training process.

4.1 Training Objective

MSR contains a teacher network f and a student network g, both are functions that map $\mathcal{X} \to \mathcal{Y}$. f is initialized by fine-tuning a pre-trained language model (PLM) on the weakly labeled data $\mathcal{D}_w$:

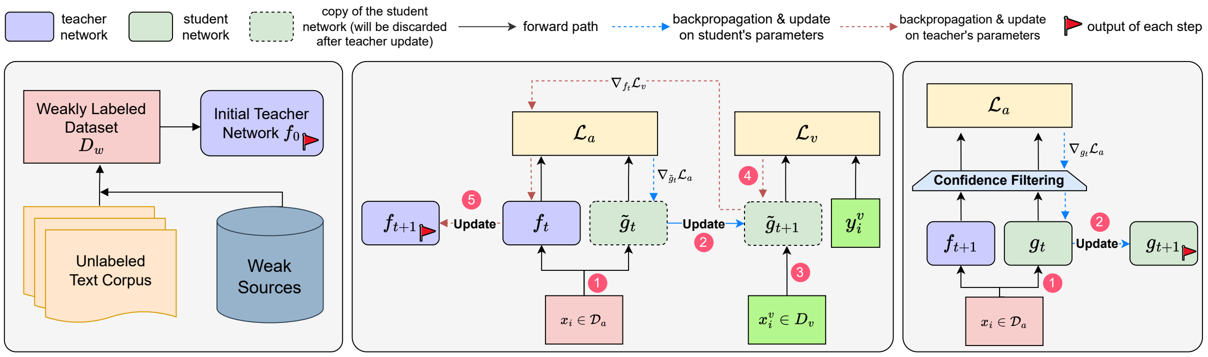
$$f_1 = \arg\min_f \mathbb{E}_{(x_i, \hat{y}_i) \in \mathcal{D}_w} \mathcal{L}(\hat{y}_i, f(x_i)) \qquad (1)$$

where $\mathcal{L}$ denotes the loss function. We use the cross entropy loss throughout the paper:

$$\mathcal{L}(p, q) = -\mathbb{E}_{y \sim p(y)} \log q(y) \qquad (2)$$

p and q are distributions over the label space $\mathcal{Y}$. The initial student network, g_1, is the PLM without fine-tuning on any data.

[2] Multiple weak sources may be triggered simultaneously by a sample. In this case, we can use different aggregation methods like majority voting to determine the final weak label.


teacher network
student network
copy of the student network (will be discarded after teacher update)
forward path
backpropagation & update on student's parameters
backpropagation & update on teacher's parameters
output of each step
Weakly Labeled Dataset $\mathcal{D}_w$
Initial Teacher Network f_0
Unlabeled Text Corpus
Weak Sources
$\nabla_{f_t}\mathcal{L}_v$
$\mathcal{L}_a$
$\mathcal{L}_v$
$\nabla_{\tilde{g}_t}\mathcal{L}_a$
5
4
f_{t+1}
Update
f_t
$\tilde{g}_t$
Update
$\tilde{g}_{t+1}$
y_i^v
2
3
1
$x_i \in \mathcal{D}_a$
$x_i^v \in \mathcal{D}_v$
$\mathcal{L}_a$
$\nabla_{g_t}\mathcal{L}_a$
Confidence Filtering
f_{t+1}
g_t
2
Update
g_{t+1}
1
$x_i \in \mathcal{D}_a$

(a) Train the initial teacher with weak labels

(b) Teacher update

(c) Student update

Figure 2: Illustration of our proposed Meta-Self Refinement method (MSR). (a) We start by fine-tuning a PLM on weak labels with early stopping, which yields an initial teacher f_1. (b) At each training step t, f_t gets training signals by performing a "teaching experiment" on $\tilde{g}_t$: a copy of the student network g_t. $\tilde{g}_t$ is updated by fitting f_t with the loss function $\mathcal{L}_a$. f_t is then updated to minimize the validation loss $\mathcal{L}_v$ of $\tilde{g}_{t+1}$. (c): g_t is updated by fitting f_{t+1} with confidence filtering under the loss $\mathcal{L}_a$.

In conventional self-training, f_1 is used to provide pseudo-labels to train the student. By selecting higher-quality pseudo-labels via confidence filtering (Yu et al., 2021) or uncertainty estimation (Mukherjee and Hassan Awadallah, 2020), the student can often outperform its teacher. However, as the teacher is trained solely on the weak labels, it can easily inherit unexpected biases and provide misleading signals to the student. In MSR, instead of using a fixed teacher to provide pseudo-labels, we use student performance on the clean validation set as a feedback signal to dynamically refine the teacher. Specifically, the objective for the teacher f, formulated as in Equation 3, is that *the student network, after fitting the teacher's output labels on $\mathcal{D}_a$, can perform best on the validation set $\mathcal{D}_v$*:

$$\begin{aligned} f^\star &= \arg\min_f \mathbb{E}_{(x_i^v, y_i^v) \in \mathcal{D}_v} \mathcal{L}(y_i^v, g'_f(x_i^v)) \\ g'_f &= \arg\min_g \mathbb{E}_{x_i \in \mathcal{D}_a} \mathcal{L}(f(x_i), g(x_i)) \end{aligned} \tag{3}$$

where g' is the student network after fitting output labels from f on $\mathcal{D}_a$. Intuitively, MSR aims to find the best teacher to help the student achieve the lowest validation loss. After finding the optimal teacher $f^\star$ in Equation 3, the student can then be obtained by learning from the output labels of $f^\star$:

$$g^\star = \arg\min_g \mathbb{E}_{x_i \in \mathcal{D}_a} \mathcal{L}(f^\star(x_i), g(x_i)) \tag{4}$$

4.2 Training Details

Finding the exact $f^\star$ in Equation 3 involves solving two nested loops of optimization, and each loop can be computationally expensive given the large size of $\mathcal{D}_a$. We resort to an online approximation to merge Equation 3 and 4 into an iterative training pipeline. At each training step t, the teacher f_t is first updated based on the meta-objective of "learning to teach", the student g_t is then trained by the updated teacher.

Algorithm 1: MSR Algorithm

```
Input: Initial teacher network $f_1$ trained according to
       Eq. 1. Student network $g_1$, number of training
       steps $T$, teacher's learning rate scheduler
       $R(t)$, confidence threshold $\tau$, $\mathcal{D}_a$, $\mathcal{D}_v$.
Result: $f_T$, $g_T$
for $t \leftarrow 1 \dots T$ do
    $\{x_i\} \leftarrow$ SampleMiniBatch( $\mathcal{D}_a$)
    $\{x_i^v, y_i^v\} \leftarrow$ SampleMiniBatch( $\mathcal{D}_v$)
    // Teacher Update
    $\tilde{g}_t \leftarrow \mathrm{Copy}(g_t)$
    $\tilde{g}_{t+1} \leftarrow \tilde{g}_t - \lambda_s \mathbb{E}_{x_i} \nabla_{\tilde{g}_t} \mathcal{L}(f_t(x_i), \tilde{g}_t(x_i))$
    $f_{t+1} \leftarrow f_t - R(t)\, \mathbb{E}_{(x_i^v, y_i^v)} \nabla_{f_t} \mathcal{L}(y_i^v, \tilde{g}_{t+1}(x_i^v))$
    // Student Update
    $w(f_{t+1}(x_i)) \leftarrow \mathbb{1}(1 - \frac{H(f_{t+1}(x_i))}{\log(k)} \geq \tau)$
    $g_{t+1} \leftarrow g_t -$
        $\lambda_s \mathbb{E}_{x_i} \nabla_{g_t} w(f_{t+1}(x_i)) \mathcal{L}(f_{t+1}(x_i), g_t(x_i))$
end
```

Teacher Update. To update the teacher in an efficient way, we approximate the inner loop in Equation 3 with a single-step gradient descent of the student network. Namely, the objective of the teacher is changed so that the current student, after *one single gradient descent step* of fitting the teacher, can perform best on the validation set. To do so, the teacher will first conduct a "teaching experiment" on a copy of the current student, denoted as $\tilde{g}_t$. $\tilde{g}_t$ is updated for one gradient descent step to fit the teacher's pseudo labels[3]:

$$\tilde{g}_{t+1} = \tilde{g}_t - \lambda_s \, \mathbb{E}_{x_i \sim \mathcal{D}_a} \, \nabla_{\tilde{g}_t} \mathcal{L}(f_t(x_i), \tilde{g}_t(x_i))$$

where λ_s is the learning rate of the student network. Afterwards, we update the teacher network to minimize the validation loss of $\tilde{g}_{t+1}$:

$$f_{t+1} = f_t - \lambda_t \, \mathbb{E}_{(x_i^v, y_i^v) \sim \mathcal{D}_v} \, \nabla_{f_t} \mathcal{L}(y_i^v, \tilde{g}_{t+1}(x_i^v))$$

where λ_t is the learning rate of the teacher network. It requires calculating second derivatives over f_t. We always use soft labels from the teacher for $\mathcal{L}(f_t(x_i), \tilde{g}_t(x_i))$, so the whole process is fully differentiable. Note that $\tilde{g}_t$ is only used in the "teaching experiment" to help update the teacher. It will be discarded after the teacher is updated.

Student Update. After obtaining f_{t+1}, the real student network is updated with the same objective as in Equation 4, except that we use the updated teacher f_{t+1} instead of $f^\star$. As the teacher has performed the "teaching experiment", it will provide more useful signals to guide the student.[4]

Teacher Learning Rate Scheduler. We find the teacher is rather sensitive to its learning rate in practice. If the learning rate is large from the start, the teacher may over-adjust itself due to the large performance gap between the teacher and the student. If the learning rate is small, the teacher will adjust itself too slowly so that more noisy pseudo-labels are passed to the student network. Therefore, we apply a linear learning rate scheduler $R(t) = \frac{t\lambda_t}{T}$ to the teacher network where t denotes the current iteration and λ_t is the targeted learning rate for the teacher. By this means, the teacher's learning rate will gradually increase as it gets better at teaching.

Confidence-Based Label Filtering. Despite having the opportunity to refine itself, the teacher inevitably produces some wrong pseudo labels during training, especially at early iterations of self-refinement. To further reduce error propagation, we only select labels with high confidence to guide the student model. The student is updated as follows:

$$g_{t+1} = g_t - \lambda_s \, \mathbb{E}_{x_i \sim \mathcal{D}_a} \, \nabla_{g_t} \mathcal{L}(f_{t+1}(x_i), g_t(x_i)) \times \mathbb{1}(1 - \frac{H(f_{t+1}(x_i))}{\log(k)} \geq \tau)$$

where $\mathbb{1}$ is the indicator function, $H(f_{t+1}(x_i))$ is the entropy of the distribution $f_{t+1}(x_i)$, k is the number of classes in $\mathcal{Y}$ and τ is a pre-defined confidence threshold. $\log(k)$ is the upper bound of the entropy for k-classification tasks. By this means, only low-entropy (high-confidence) predictions from the teacher are learned. Note that the filtering strategy is only applied to the actual student update step, not during the teaching experiment. Otherwise, the teacher will ignore low-confident samples as they do not contribute to teacher update.

Putting all together, Algorithm 1 summarizes the self-refinement process.

5 Experimental Settings

Datasets. WRENCH (Zhang et al., 2021) is a well-established benchmark for weak supervision and offers weak labels for various datasets. We compare different baselines on six NLP datasets from WRENCH including both sequence classification and Named-Entity Recognition (NER) tasks. For sequence classification, we include AGNews (Zhang et al., 2015), IMDB (Maas et al., 2011), Yelp (Zhang et al., 2015), and TREC (Li and Roth, 2002). For NER tasks, CoNLL-03 (Sang and De Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2013) are used. In addition, we further include two sequence classification datasets in low-resource languages, Yorùbá and Hausa (Hedderich et al., 2020), to involve evaluation cases in diverse languages. Table 1 summarizes the basic statistics of the datasets. Majority voting over weak sources is used to determine a single label for each sample.

Dataset	Task	# Class	# Train	# Val	# Test
AGNews	Topic	4	96,000	12,000	12,000
IMDB	Sentiment	2	20,000	2,500	2,500
Yelp	Sentiment	2	30,400	3,800	3,800
TREC	Question	6	4,965	500	500
Yorùbá	Topic	7	1,340	189	379
Hausa	Topic	5	2,045	290	582
CoNLL03	NER	4	14,041	3,250	3,453
OntoNotes5.0	NER	18	115,812	5,000	22,897

Table 1: Dataset statistics. Refer to Appendix A for more details on datasets.

Implementation. RoBERTa-base (Liu et al., 2019) is used as the PLM for English datasets and

[3] We use SGD for illustration purposes. The AdamW (Loshchilov and Hutter, 2019) optimizer is used in our experiments.

[4] In theory, if the teacher network is strong enough to generalize among different batches, we can directly update the real student in the "teaching experiment", in the hope that the teacher from the last step can also work in the current batch. However, in practice, we find this mismatch leads to poor performance.

multilingual BERT-base (Devlin et al., 2019) for non-English ones. We utilize the higher[5] library to perform second-order optimization. Refer to Appendix B for detailed hyper-parameter configurations.

Baselines. We compare our method with prior work on learning with noisy labels. 1) **Majority** applies majority vote on the weak labels. Ties are broken by randomly selecting a weak label. 2) **Snorkel** (Ratner et al., 2017) trains a labeling model that aggregates weak labels from different weak sources. 3) **FT-WL** fine-tunes PLMs on the weak labels. 4) **FT-WLST** further applies classic self-training (Lee et al., 2013) on the model obtained by FT-WL. 5) **L2R** (Ren et al., 2018) uses a meta-learning framework to reweight weakly labeled samples. 6) **Meta-Weight-Net** (Shu et al., 2019) also applies meta-learning based sample reweighting. However, the weights are computed through an external reweighting network. 7) **Denoise** (Ren et al., 2020) iteratively corrects wrong annotations in the training set, and the classifier learns with the corrected labels. 8) **UST** (Mukherjee and Hassan Awadallah, 2020) is a self-training based method that assigns higher weights to samples that the teacher is certain about. The uncertainties are measured via MC-dropout on the predictions (Gal and Ghahramani, 2016). 9) **COSINE** (Yu et al., 2021) trains its student network with pseudo-labels which the teacher is highly confident about. In addition, contrastive regularization is introduced to further alleviate error propagation.

For our proposed framework, we report the performances of both **Teacher-Init** (f_1): the initial teacher trained directly on weak labels, and **MSR**: the final student model (g_T). f_1 is obtained by running FT-WL five times and selecting the best one among them according to the validation performance. *For a fair comparison, the same f_1 is used as the initial teacher for all self-training based models.* Finally, we also include the results of fine-tuning PLMs on the clean versions of each dataset, denoted by **FT-CL**, to represent the upper bound performance.

6 Results

Comparison with Baselines. Table 2 shows a comparison among different methods. MSR matches or outperforms SOTAs on all eight datasets. FT-WL outperforms majority voting over the weak labels in all cases except Hausa, which leads to a minor drop. This confirms that PLMs encode useful knowledge in their parameters, enabling them to generalize better than weak rules they are trained on. This phenomenon is particularly noticeable on AGNews, IMDB, and Yelp: direct fine-tuning on the noisy labels (FT-WL) can already achieve decent performance (accuracy above 83%). *We consider them easy tasks since label noise has only a minor impact on performance of PLMs and decent generalization can be attained even without specific noise-handling.* Applying self-training to such simple tasks lead to further performance improvement. COSINE, a SOTA self-training based model, can even perform comparably to the fully supervised model on these three datasets. On the other five datasets, however, FT-WL performs poorly and conventional self-training methods provide little performance boost (even a disservice on OntoNotes). This implies that *self-training relies on a well-performed initial teacher to work effectively.* On challenging datasets where the initial teacher is weak, it struggles to achieve further performance gain. Meta-learning based methods such as L2R performs better than COSINE on these challenging datasets. *MSR can further boost the performance on all the challenging datasets by up to 11.4% in accuracy or 9.26% in F1 score while maintaining comparable results on simpler datasets.*

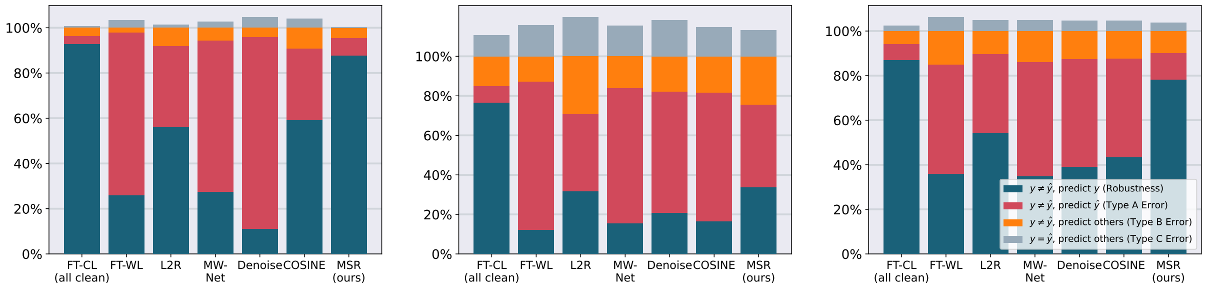
Error Decomposition. Let $y', \hat{y}, y$ denote the model prediction, the noisy weak label, and the clean label, respectively. To investigate how the label noise influences the model predictions, we decompose model prediction errors into three types: (1) Type-A error: $y' = \hat{y}; \hat{y} \neq y$ (2) Type-B error: $y' \neq \hat{y} \neq y$ and (3) Type-C error: $y' \neq y; y = \hat{y}$. Type-A/B errors correspond to situations in which a model complies with an incorrect weak label $\hat{y}$, or predicts another incorrect class label. If, on the other hand, the weak label $\hat{y}$ is correct, a Type-C error arises if the model predicts a label different than $\hat{y}$. A higher Type-A error rate indicates that a model memorizes more label noise from the weak sources, while a model that underfits fails to learn useful knowledge from the weak sources can have a higher Type-C error rate.

Figure 3 visualizes the three types of errors on three challenging datasets: TREC, Hausa and CoNLL-03. The blue bars represent model robust-

[5] https://github.com/facebookresearch/higher

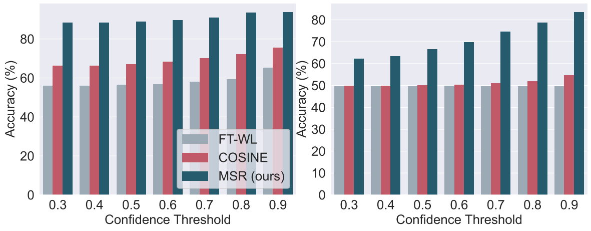
Method	AGNews (Acc)	IMDB (Acc)	Yelp (Acc)	TREC (Acc)	Yorùbá (Acc)	Hausa (Acc)	CoNLL-03 (F1)	OntoNotes (F1)
Fully-Supervised Result								
FT-CL	92.61	93.20	96.91	96.67	77.24	81.57	92.27	85.74
Label Models								
Majority	63.84	71.04	70.21	60.80	58.05	47.93	60.38	58.92
Snorkel (Ratner et al., 2017)	62.67	71.60	68.92	59.60	62.80	47.94	62.88	58.46
DNN Baselines								
FT-WL	$85.73_{\pm 0.43}$	$83.43_{\pm 0.91}$	$87.71_{\pm 1.46}$	$66.80_{\pm 1.44}$	$64.12_{\pm 0.83}$	$46.13_{\pm 0.43}$	$69.20_{\pm 0.33}$	$67.26_{\pm 0.62}$
FT-WLST† (Lee et al., 2013)	$88.61_{\pm 0.71}$	$89.50_{\pm 0.65}$	$95.32_{\pm 0.70}$	$76.00_{\pm 2.21}$	$67.28_{\pm 1.12}$	$49.22_{\pm 1.39}$	$69.87_{\pm 0.36}$	$64.13_{\pm 1.45}$
L2R (Ren et al., 2018)$^{\diamond}$	$87.28_{\pm 1.00}$	$82.76_{\pm 1.59}$	$93.34_{\pm 0.91}$	$83.40_{\pm 2.01}$	$70.45_{\pm 0.69}$	$55.67_{\pm 0.88}$	$79.15_{\pm 1.34}$	$70.66_{\pm 0.74}$
Meta-Weight-Net$^{\diamond}$ (Shu et al., 2019)	$85.96_{\pm 0.80}$	$86.72_{\pm 0.50}$	$86.97_{\pm 0.74}$	$69.39_{\pm 1.27}$	$70.00_{\pm 2.12}$	$48.63_{\pm 0.96}$	$69.54_{\pm 1.43}$	$69.11_{\pm 1.20}$
Denoise (Ren et al., 2020)	$83.45_{\pm 0.68}$	$76.22_{\pm 0.92}$	$71.56_{\pm 0.56}$	$61.80_{\pm 1.30}$	$66.10_{\pm 1.52}$	$49.31_{\pm 0.93}$	$72.96_{\pm 0.51}$	$67.64_{\pm 1.06}$
UST† (Mukherjee and Hassan Awadallah, 2020)	$87.78_{\pm 0.59}$	$86.74_{\pm 1.18}$	$91.23_{\pm 0.90}$	$77.20_{\pm 2.29}$	$68.12_{\pm 0.71}$	$47.67_{\pm 0.91}$	$69.48_{\pm 1.69}$	$66.98_{\pm 0.99}$
COSINE† (Yu et al., 2021)	$89.34_{\pm 0.76}$	$\mathbf{90.52}_{\pm 1.06}$	$\mathbf{95.48}_{\pm 0.13}$	$82.60_{\pm 1.09}$	$68.87_{\pm 0.82}$	$49.66_{\pm 1.32}$	$70.60_{\pm 0.87}$	$64.59_{\pm 1.08}$
Our Framework								
Teacher-Init (f_1)	$86.37_{\pm 0.00}$	$85.00_{\pm 0.00}$	$89.92_{\pm 0.00}$	$69.00_{\pm 0.00}$	$65.44_{\pm 0.00}$	$46.74_{\pm 0.00}$	$69.73_{\pm 0.00}$	$68.25_{\pm 0.00}$
MSR$^{\dagger\ \diamond}$	$\mathbf{89.92}_{\pm 0.64}$	$89.16_{\pm 0.91}$	$95.00_{\pm 0.35}$	$\mathbf{94.80}_{\pm 0.29}$	$\mathbf{72.56}_{\pm 0.78}$	$\mathbf{59.11}_{\pm 0.78}$	$\mathbf{88.41}_{\pm 0.63}$	$\mathbf{74.59}_{\pm 0.84}$

Table 2: Accuracy and F1 score (in %) on eight NLP tasks. The mean and standard deviation over five trials are reported. Teacher-Init is the best model checkpoint selected from the five trials of FT-WL (according to the validation performance). For a fair comparison, all self-training-based models use the same Teacher-Init checkpoint. MSR matches or outperforms SOTAs on all tasks. † self-training based method. $^{\diamond}$ meta-learning based method.



(a) TREC (b) Hausa (c) CoNLL-03

Figure 3: Prediction error decomposition of various weak supervision baselines, evaluated on the test sets. A model is considered robust against label noise if it manages to predict the correct labels despite the wrong weak labels (the robustness is represented by the blue bars). Otherwise, it conforms to the weak label (Type-A error) or predict another incorrect label (Type-B error), which has a negative effect on generalization. The Type-C error rate quantifies the proportion of incorrect model predictions when weak labels are correct. MSR consistently reduces the Type-A error rate and attains a high level of noise robustness.

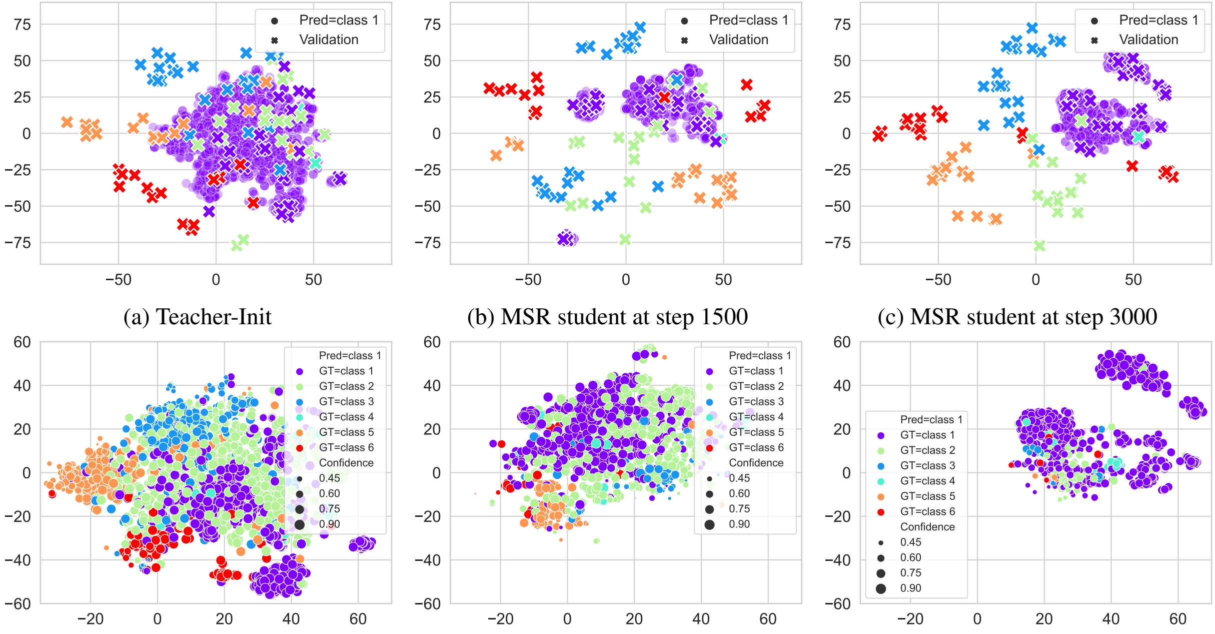


(a) TREC (b) Hausa

Figure 4: Accuracy *vs.* confidence thresholds.

ness, i.e., how often the model predicts correctly when $\hat{y} \neq y$. It clearly shows that direct fine-tuning on weak labels (FT-WL) has a much higher Type-A error rate compared with the model trained on clean data (FT-CL), suggesting that the model quickly memorizes the label noise. On the other hand, the disparity in type C error rate is much smaller, indicating that all models do not underfit and the knowledge from the weak sources is properly transferred. The Type-B error shows similar trends and does not differ much across models. Overall, Type-A error has the strongest impact on model performance. *All the noisy-handling models mainly help with reducing Type-A errors*. We also observe that while COSINE reduces Type-A errors on TREC, it barely works on the other two datasets. Only MSR manages to consistently reduce Type-A errors by over 20% on all three datasets.

Accuracy vs Confidence. As confidence-based filtering is a key component in both COSINE and MSR, we show the accuracy of model predictions with different confidence thresholds in Figure 4. As can be seen, *even using a high confidence threshold for COSINE, the accuracy is still low*, which is why it struggles to improve on challenging datasets. MSR, on the contrary, consistently attains higher accuracy with higher confidence thresholds, and thereby confidence-based filtering on top of MSR



(a) Teacher-Init

(b) MSR student at step 1500

(c) MSR student at step 3000

(d) Teacher-Init

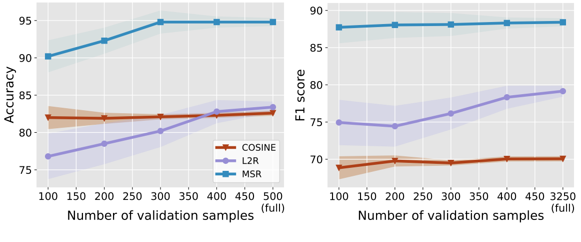
(e) COSINE

(f) MSR

Figure 5: Projected feature space of different models on TREC using t-SNE (Van der Maaten and Hinton, 2008). The circles represent training samples that are predicted as class 1. **a)-c)**: development of MSR during training. Circles are colored by the predicted class (i.e., class 1, in purple). The validation samples are represented by crosses and colored according to the ground truth labels. The MSR student gradually improves its feature space to embed the training and validation samples from the same class in the same area. **d)-f)**: training samples are colored according to their ground truth labels; model confidence is reflected by the size of the circles. Teacher-Init and COSINE misclassify samples with high confidence. MSR attains a cleaner cluster.

help lead to better performance.

Impact of Label Noise on Feature Space. We also analyze how the label noise influences representation learning. Figure 5 illustrates the projected feature space of different models on TREC. For a clear visualization, we present only training samples predicted as class 1 by the models in the form of circles. In figs. 5a to 5c, we further visualize the feature space of validation samples (represented by crosses). As can be seen, initially the feature space of class 1 overlaps with that of other classes from the validation set. As the training proceeds, when the teacher keeps refining itself, the MSR student gradually reduces such overlap and learns a well-split representation space. In figs. 5d to 5f, we compare the feature space between different models. The training samples are colored according to their ground truth classes to highlight the misclassification ratio (the more colorful the clusters, the higher the misclassification ratio). We observe that Teacher-Init makes many wrong predictions with high degree of confidence. In this case, utilizing the confidence score for denoising is fragile. This may explain why COSINE, despite offering a more compact cluster, still has a considerable amount of misclassification. Finally, MSR has a considerably cleaner cluster and is less affected by error propagation than COSINE.



(a) TREC

(b) CoNLL-03

Figure 6: Accuracy *vs.* number of validation samples.

Effects of Validation Data Size. The model performance reported in Table 2 is based on the original data splits from the WRENCH benchmark. The size of the validation sets is mostly less than 15% of the training sets. Typically, they are used to perform early stopping and model selection. For meta-learning based methods, they additionally rely on the validation sets for meta-update and might be more sensitive to validation size. Hence, we study how the validation size affects different models. In particular, we randomly sample a subset from the original validation set $\mathcal{D}_v$ and repeat the same training process. Figure 6 presents the results. We find that the validation size indeed has a greater impact on meta-learning approaches. However, *MSR still retains its high generalization performance even with as few as 100 validation samples*, suggesting that MSR is very data efficient in performing the self-refinement.

Configuration	Seq. Classification (Acc)	NER (F1)
Teacher-Init	73.75	68.99
Student	**83.43**	**81.50**
Teacher	82.38 (↓ 1.05%)	80.26 (↓ 1.24%)
w/o Teacher Scheduler	81.80 (↓ 1.63%)	80.15 (↓ 1.35%)
w/o Confidence Filtering	82.32 (↓ 1.11%)	81.09 (↓ 0.41%)
w/o Both	81.63 (↓ 1.80%)	79.95 (↓ 1.55%)

Table 3: Summary of ablation experiments aggregated across multiple datasets. See Appendix D for results in each dataset.

Ablation Study. Table 3 summarizes the impact of different components of our method. In general, our student model performs slightly better than the teacher. This is as expected because a) the teacher's goal is to guide the student to generalize better, the training loss does not explicitly encourage the teacher to improve its accuracy, and b) the confidence filtering helps the student avoid fitting some wrong pseudo-labels from the teacher. This is also confirmed by the decreased performance when the filter is removed. In addition, applying a learning rate scheduler is better than using a fixed learning rate throughout training.

7 Conclusion

We present MSR, a meta-learning based self-refinement framework that enables robust learning with weak labels. Unlike conventional self-training which relies on a fixed teacher, MSR dynamically refines the teacher based on the student's performance on the validation set. To further suppress error propagation, we introduce a learning rate scheduler to the teacher and add confidence filtering to the student. We demonstrate that our framework performs on par with or better than current SOTAs on both sequence classification and labeling tasks.

Limitations

In this work, Our primary focus is to propose a strong weak supervision method that works reliably under various weak supervision settings. We employ meta-learning techniques to address the issue of unreliable confidence scores under challenging settings (Figure 4). Despite the effectiveness, the main limitation of our method, just like other meta-learning based frameworks, is the computational overhead. The teacher update step (Algorithm 1, Line 4-6) requires computing both the first and second-order derivatives, which incurs additional computation time and higher memory consumption. Consequently, our method requires longer training.[6] Implementation-wise and computation-wise, MSR is as complex as other existing meta-learning based methods, like L2R (Ren et al., 2018) and MW-Net (Shu et al., 2019), but performs substantially better than them in all weak supervision scenarios we evaluated. It is worth noting that MSR has *no overhead at inference time*. In weak supervision, the data annotation cost is considered the most significant bottleneck. A stronger model is often obtained by trading some more computation with the cost and effort of obtaining more human-generated, manual annotations. Hence, the one-off investment of training MSR can be worthwhile for real-world weak supervision applications.

Acknowledgments

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102 and the EU Horizon 2020 projects ROXANNE under grant number 833635.

References

Dana Angluin and Philip Laird. 1988. Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. 2020. Learning from rules generalizing labeled exemplars. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alan Joseph Bekker and Jacob Goldberger. 2016. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 2682–2686. IEEE.

Allan Peter Davis, Thomas C. Wiegers, Phoebe M. Roberts, Benjamin L. King, Jean M. Lay, Kelley Lennon-Hopkins, Daniela Sciaky, Robin J. Johnson, Heather Keating, Nigel Greene, Robert Hernandez,

[6] Detailed training time on each dataset can be found in Appendix E The most costly training of MSR takes roughly 3 hours.

Kevin J. McConnell, Ahmed Enayetallah, and Carolyn J. Mattingly. 2013. A ctd-pfizer collaboration: manual curation of 88 000 scientific articles text mined for drug-disease and drug-phenotype interactions. *Database J. Biol. Databases Curation*, 2013.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Daniel Fu, Mayee Chen, Frederic Sala, Sarah Hooper, Kayvon Fatahalian, and Christopher Ré. 2020. Fast and three-rious: Speeding up weak supervision with triplet methods. In *International Conference on Machine Learning*, pages 3280–3291. PMLR.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Keren Gu, Xander Masotto, Vandana Bachani, Balaji Lakshminarayanan, Jack Nikodem, and Dong Yin. 2021. An instance-dependent simulation framework for learning with label noise. *arXiv preprint arXiv:2107.11413*.

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 *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546.

Michael A. Hedderich, David Ifeoluwa Adelani, Dawei Zhu, Jesujoba O. Alabi, Udia Markus, and Dietrich Klakow. 2020. Transfer learning and distant supervision for multilingual transformer models: A study on african languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2580–2591. Association for Computational Linguistics.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10477–10486.

Giannis Karamanolakis, Subhabrata (Subho) Mukherjee, Guoqing Zheng, and Ahmed H. Awadallah. 2021. Self-training with weak supervision. In *NAACL 2021*. NAACL 2021.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Junnan Li, Richard Socher, and Steven C. H. Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

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 *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1518–1533. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aditya Krishna Menon, Brendan van Rooyen, and Nagarajan Natarajan. 2016. Learning from binary labels with instance-dependent corruption. *CoRR*, abs/1605.00751.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, Online.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2233–2241. IEEE Computer Society.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. 2021. Meta pseudo labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11557–11568. Computer Vision Foundation / IEEE.

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.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4331–4340. PMLR.

Wendi Ren, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. 2020. Denoising multi-source weak supervision for neural text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3739–3754. Association for Computational Linguistics.

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. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1917–1928.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2020. Adaptive self-training for few-shot neural sequence labeling. *arXiv preprint arXiv:2010.03680*.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. Computer Vision Foundation / IEEE.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings*, pages 189–196. Morgan Kaufmann Publishers / ACL.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-tuning pretrained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1063–1077. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. WRENCH: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2022. Bert learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Wenxuan Zhou, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren. 2020. NERO: A neural rule grounding framework for label-efficient relation extraction. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2166–2176. ACM / IW3C2.

Dawei Zhu, Michael A. Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is BERT robust to label noise? A study on learning with noisy labels in text classification. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP, Insights@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 62–67. Association for Computational Linguistics.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

A Dataset Details

We experiment with eight NLP datasets, including six English datasets and two datasets in low-resource languages. All datasets come with their ground truth annotations and as well as the weak labels.

A.1 Datasets Selection Criteria

The WRENCH (Zhang et al., 2021) benchmark contains 23 NLP datasets. We choose representative datasets (like previous research in weak supervision) that **a)** overlap with previous works to enable direct comparisons. **b)** are diverse in terms of weak label quality, languages and tasks to approve the applicability of different baselines.

A.2 English Datasets

We experiment with four popular sequence classification datasets: AGNews, IMDB, Yelp and TREC.

1. **AGNews** (Zhang et al., 2015): originates from AG, which is a large collection of news articles. The news are categorized in four classes: "World", "Sports", "Business" and "Sci/Tech".

2. **IMDB** (Maas et al., 2011): consists of movie reviews with binary labels. It is a commonly used benchmark dataset for sentiment analysis.

3. **Yelp** (Zhang et al., 2015): obtained from the Yelp Dataset Challenge in 2015. Similar to IMDB, it is a sentiment analysis dataset.

4. **TREC** (Li and Roth, 2002): categorizes the questions in TREC-6 datasets into 6 categories: "Abbreviation", "Entity", "Description", "Human", "Location", "Numeric-value".

and with the two sequence labeling datasets: CoNLL-03 and OntoNotes 5.0.

1. **CoNLL-03** (Sang and De Meulder, 2003) NER dataset with four named-entity categories.

2. **OntoNotes 5.0** (Pradhan et al., 2013): NER dataset with 18 named-entity categories.

All weak labels are obtained from the WRENCH benchmark[7] (Zhang et al., 2021).

A.3 Datasets in Low-Resource Languages

Most datasets in the current WRENCH benchmarks are in English. Although weak supervision is desired in low-resource languages, it is understudied as finding annotators for them is more difficult. Hence, we include two low-resource languages, Yorùbá and Hausa, to cover this scenario. Often, learning with weak labels in low-resource languages is more challenging. First, the training set is often much smaller than English datasets. For example, Hausa has only about 2k training samples while AGNews have 96k. Second, the weak labels in low-resource languages can have lower quality as experts for weak source development are harder to find. A set of simple rules is often used for labeling (which is the case in Yorùbá and Hausa). Hence, weak supervision with low-resource languages is a combination of two challenges: training with *small* datasets which have *low-quality* labels.

Yorùbá and Hausa are text classification datasets obtained from (Hedderich et al., 2020).[8]

1. **Yorùbá**: consists of news headlines from BBC Yoruba which are categorized in seven classes: "Nigeria", "Africa", "World", "Entertainment", "Health", "Sport", "Politics".

2. **Hausa**: consists of news headlines from VOA Hausa which have the same seven classes as Yorùbá. However, only five classes are considered in the text classification task. "Entertainment" and "Sport" have been removed due to the lack of samples of these classes.

Hedderich et al. (2020) provided both the clean labels and weak labels on the two datasets. A gazetteer is created for each class for weak supervision. For example, a gazetteer containing names of agencies, organizations, states and cities in Nigeria is used to label the class "Nigeria".

A.4 More Dataset Statistics

We provide more details on the datasets we used in our experiments in Table 4. In general, not all data can be covered by weak sources. No weak source is triggered for some training samples and they remain unlabeled. The coverage of the datasets ranges from 69.08% to 100%. Note that for NER tasks, the coverage is always 100% since if no weak source is triggered for a token, we assign

[7] https://github.com/JieyuZ2/wrench

[8] https://github.com/uds-lsv/transfer-distant-transformer-african

Dataset	Task	# Class	$\lvert\mathcal{D}_w\rvert$	$\lvert\mathcal{D}_a\rvert$	Coverage	Conflict	$\lvert\mathcal{D}_v\rvert$	$\lvert\mathcal{D}_t\rvert$
AGNews	Topic	4	66,314	96,000	69.08%	14.17%	12,000	12,000
IMDB	Sentiment	2	17,515	20,000	87.58%	11.96%	2,500	2,500
Yelp	Sentiment	2	25,165	30,400	82.78%	18.29%	3,800	3,800
TREC	Question	6	4,723	4,965	95.13%	22.76%	500	500
Yorùbá	Topic	7	1,340	1,340	100.00%	1.87%	189	379
Hausa	Topic	5	2,045	2,045	100.00%	1.90%	290	582
CoNLL03	NER	4	14,041	14,041	100.00%	4.05%	3,250	3,453
OntoNotes5.0	NER	18	115,812	115,812	100.00%	1.86%	5,000	22,897

Table 4: Dataset statistics. $|\mathcal{D}_w|$: number of training samples with weak labels. $|\mathcal{D}_a|$: total number of training samples (weakly labeled + unlabeled). Coverage: fraction of samples that are weakly labeled, i.e., $\frac{|\mathcal{D}_w|}{|\mathcal{D}_a|}$. Conflict: samples that are labeled by at least two weak sources with contradicted weak labels. $|\mathcal{D}_v|$: number of validation samples. $|\mathcal{D}_t|$: number of test samples.

Hyperparameter	Search Range
Teacher Learning Rate	3e-6, 5e-6, 2e-5, 3e-5
Teacher Warm-Up Steps	500, 100, 2000, 3000
Confidence Filter Threshold	0.4, 0.5, 0.6, 0.7, 0.8, 0.95

Table 5: Hyperparameter search.

label "O" (i.e., non-entity) to it. On the other hand, some samples can be covered by two or more weak sources with contradicted weak labels. In this case, we have a conflict. The conflict ratio ranges from 1.86% to 22.76% in the datasets we tested.

B Implementation Details

Models. All baselines in our paper, except the majority vote and the Snokerl model (Ratner et al., 2017) which work with label space only, use the official RoBERTa model[9] (Liu et al., 2019) from Huggingface as the classification backbone for all English datasets, and the multilingual BERT[10] for datasets in African languages. We use the base version of the two models which contain roughly 120M and 110M parameters, respectively.

Fine-Tuning on Classification Task. We fine-tune all layers using AdamW (Loshchilov and Hutter, 2019) as the optimizer. For sequence classification tasks, we pass the final layer of the [CLS] token representation ($\mathbb{R}^{768}$) to a feed-forward layer for prediction. For sequence labeling tasks, the final layers of all tokens ($\mathbb{R}^{768\times L}$, where L is the sentence length) are passed to a shared feed-forward layer to predict the class of each token in the sentence. We report the score where the model performs the best on the validation set during training.

Hyper-Parameters of MSR. We apply grid search on the warm-up steps for the teacher and the confidence threshold for the student network. Table 5 shows our hyperparameter search configuration. We choose the final configurations of the hyperparameters according to the model's performance on the validation set. Table 6 shows the best configurations of parameters we used to produce the results in Table 2.

Evaluation Metrics. For model evaluation, we report accuracy for sequence classification tasks and F1 Score for sequence labeling tasks. In our implementation, we call the function `classification_report()` from the scikit-learn library[11] to compute the accuracy, and use the `Seqeval` class from Huggingface[12] to compute the F1 Score.

C Validation Performance

The average test performance of MSR is reported in Table 2. We further report the corresponding validation performance in Table 7.

[9] https://huggingface.co/roberta-base

[10] https://huggingface.co/bert-base-multilingual-cased

[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

[12] https://github.com/huggingface/datasets/blob/master/metrics/seqeval/seqeval.py

	AGNews	IMDB	Yelp	TREC	Yorùbá	Hausa	CoNLL-03	OntoNotes 5.0
BERT Backbone	RoBERTa	RoBERTa	RoBERTa	RoBERTa	mBERT	mBERT	RoBERTa	RoBERTa
Batch Size	32	16	16	32	32	32	32	32
Maximum Sequence Length	128	256	256	64	64	128	64	64
Student Learning Rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
Teacher Learning Rate	2e-5	2e-5	2e-5	2e-5	5e-6	2e-5	2e-5	2e-5
Teacher Warm-Up Steps	500	500	3000	500	1000	3000	2000	2000
Confidence Filter Threshold	0.7	0.7	0.5	0.5	0.7	0.4	0.8	0.5

Table 6: Selected hyperparameters. mBERT: multilingual BERT.

Dataset	Test	Validation
AGNews	89.92	89.90
IMDB	89.16	89.21
Yelp	95.00	94.79
TREC	94.80	94.42
Yorùbá	72.56	75.13
Hausa	59.11	62.34
CoNLL-03	88.41	87.86
OntoNotes	74.59	75.20

Table 7: The average test and validation accuracy/F1 score (in %) of MSR over five trials.

D Ablation Studies

We report the detailed ablation results for each dataset in Table 8.

E Hardware and Average Runtime.

We use Nvidia Tesla V100 to accelerate training. The average runtime for each method and dataset is summarized in Table 9.

MSR Configuration	**AGNews (Acc)**	**IMDB (Acc)**	**Yelp (Acc)**	**TREC (Acc)**	**Yorùbá (Acc)**	**Hausa (Acc)**	**CoNLL-03 (F1)**	**OntoNotes (F1)**
Student	**89.92**	**89.16**	**95.00**	**94.80**	**72.56**	59.11	**88.41**	**74.59**
Teacher	89.02	88.08	94.37	93.80	68.87	**60.14**	87.30	73.22
w/o Teacher Scheduler	89.68	87.68	93.78	93.60	70.71	55.32	87.82	72.48
w/o Confidence Filtering	89.87	89.04	94.76	93.60	71.50	55.15	88.07	74.11
w/o Both	89.55	87.68	93.33	93.40	70.50	55.32	87.82	72.08

Table 8: Ablation studies. The numbers represent the test accuracy and F1 Score.

	AGNews	IMDB	Yelp	TREC	Yorùbá	Hausa	CoNLL-03	OntoNotes 5.0
Running time (hours)	2.5	1.6	0.5	1.2	0.5	0.7	1.1	3.0

Table 9: Average runtime (in hours) for training a MSR model. One single Nvidia Tesla V100 GPU is used in each experiment to accelerate the computation.