Meta Self-training for Few-shot Neural Sequence Labeling

Yaqing Wang[†], Subhabrata Mukherjee[*], Haoda Chu[◇], Yuancheng Tu[◇],
Ming Wu[◇], Jing Gao[†], Ahmed Hassan Awadallah[*]
[†]Purdue University, [*]Microsoft Research AI, [◇]Microsoft AI
{yaqingwa, jing}@purdue.edu,
{Subhabrata.Mukherjee, haochu, yuantu, mingwu, hassanam}@microsoft.com

ABSTRACT

Neural sequence labeling is widely adopted for many Natural Language Processing (NLP) tasks, such as Named Entity Recognition (NER) and slot tagging for dialog systems and semantic parsing. Recent advances with large-scale pre-trained language models have shown remarkable success in these tasks when fine-tuned on large amounts of task-specific labeled data. However, obtaining such large-scale labeled training data is not only costly, but also may not be feasible in many sensitive user applications due to data access and privacy constraints. This is exacerbated for sequence labeling tasks requiring such annotations at token-level. In this work, we develop techniques to address the label scarcity challenge for neural sequence labeling models. Specifically, we propose a meta self-training framework which leverages very few manually annotated labels for training neural sequence models. While self-training serves as an effective mechanism to learn from large amounts of unlabeled data via iterative knowledge exchange – meta-learning helps in adaptive sample re-weighting to mitigate error propagation from noisy pseudo-labels. Extensive experiments on six benchmark datasets including two for massive multilingual NER and four slot tagging datasets for task-oriented dialog systems demonstrate the effectiveness of our method. With only 10 labeled examples for each class in each task, the proposed method achieves 10% improvement over state-of-the-art methods demonstrating its effectiveness for limited training labels regime.

ACM Reference Format:
Yaqing Wang[†], Subhabrata Mukherjee[*], Haoda Chu[◇], Yuancheng Tu[◇],, Ming Wu[◇], Jing Gao[†], Ahmed Hassan Awadallah[*]. 2021. Meta Self-training for Few-shot Neural Sequence Labeling. In *KDD '21: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 14-18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Motivation. Deep neural networks typically require large amounts of labeled training data to achieve state-of-the-art performance. Recent advances with pre-trained language models like BERT [10], GPT-2 [35] and RoBERTa [27] have reduced this annotation bottleneck. In this paradigm, deep and large neural network models are trained on massive amounts of unlabeled data in a self-supervised manner. However, the success of these large-scale models still relies on fine-tuning them on large amounts of labeled data for downstream tasks. For instance, our experiments show 27% average improvement on multiple tasks when fine-tuning BERT with the full labeled training set (2.5K-705K labels) versus fine-tuning with limited amount of labels (e.g., 10 per class). This poses several challenges for many real-world tasks.

Not only is acquiring large amounts of labeled data for every task expensive and time consuming, but also not feasible in many cases due to data access and privacy constraints, especially when dealing with personal or sensitive data. This issue is exacerbated for sequence tagging tasks that require annotations at *token-* and *slot-level* as opposed to instance-level classification tasks. For example, an NER task can have slots like *B-PER, I-PER, O* marking the beginning, intermediate and out-of-span markers for person names, and similar slots for the names of location and organization. Similarly, language understanding models for dialog systems rely on effective identification of what the user intends to do (*intents*) and the corresponding values as arguments (*slots*) for use by downstream applications. Therefore, fully supervised neural sequence taggers are expensive to train for such tasks, given the requirement of thousands of annotations for hundreds of slots corresponding to the many different intents.

State-of-the-art. Semi-supervised learning (SSL) [5] is one of the approaches to address labeled data scarcity by making effective use of large amounts of unlabeled data in addition to task-specific labeled data. Self-training (ST, [15]), one of the earliest SSL approaches, has recently shown state-of-the-art performance for *instance-level classification* tasks like image classification [23, 44] performing at par with supervised systems while using very few training labels. In contrast to such instance-level classification tasks, slot tagging or alternatively, *token-level classification* tasks have dependencies between the slots demanding different design choices for slot-level loss optimization for the limited labeled data setting. For instance, prior work [37] observe that standard self-training techniques do not work for slot tagging tasks in the low-data regime (e.g., with 10% labeled data for the target domain) due to error propagation and amplification in the iterative learning framework. On the positive side, there has been some success with careful

[*]Most of the work was conducted when the author was interning at Microsoft.

KDD '21, August 14-18, 2021, Virtual Event, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

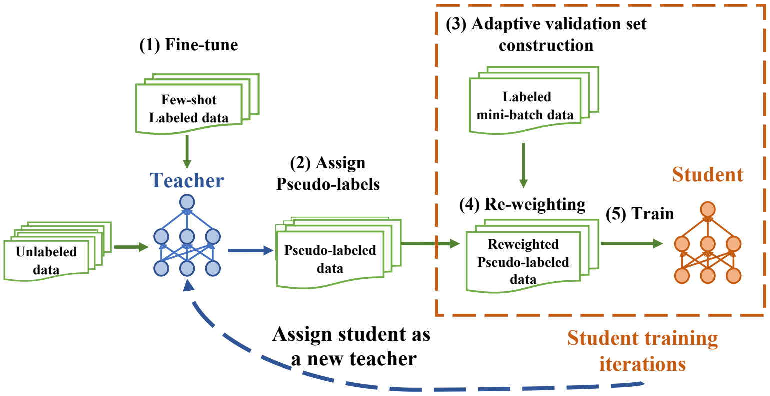


Figure 1: MetaST framework.

task-specific data selection [34], and more recently with distant supervision [25] leveraging external resources like knowledge bases (e.g., Wikipedia). In contrast to these prior work, we develop techniques for self-training with limited training labels and without any task-specific assumption or external knowledge resources.

Challenges. For self-training, a base model (*teacher*) is trained on some amount of labeled data and used to pseudo-annotate (task-specific) unlabeled data. The original labeled data is augmented with the pseudo-labeled data and used to train a *student* model. The student-teacher training is repeated until convergence. Traditionally in self-training frameworks, the teacher model pseudo-annotates unlabeled data without any sample selection. This may result in gradual drifts from self-training on noisy pseudo-labeled instances [37, 45]. In order to deal with noisy labels and training set biases, recent works have developed techniques to re-weight noisy samples leveraging prior knowledge of the task [21, 39], or automatically learning from the underlying model and task data [16, 23, 36, 40]. These prior techniques for learning to re-weight samples have been primarily developed for instance-level tasks like image [16, 36, 40] and text [23] classification. A vanilla token-level extension of these techniques for slot tagging would assume a similar quality of the token-level pseudo-labels in a sequence disregarding the slot distribution and difficulty. This is not desirable for tasks like Named Entity Recognition in WikiAnn [31] involving 123 slots over 41 languages with variable difficulty and distribution in the data and across languages. This makes it imperative to design better sampling and re-weighting strategies for slot tagging tasks in contrast to random sampling or token-agnostic re-weighting employed for instance-level classification tasks [36, 40].

To address the aforementioned challenges, we develop an adaptive learning mechanism to *re-weight noisy token-level pseudo-labels* to mitigate the effect of error propagation during self-training. To this end, we employ *meta-learning* [1, 22, 42] with the following meta-objective: *the best token-level re-weighting should minimize the model loss on a set of representative clean validation examples.* This formulation requires us to address two key research questions, namely, (i) How to construct an informative validation set for the meta-objective? and (ii) How to re-weight token-level noisy pseudo-labels to optimize the meta-objective for sequence labeling?

Prior works on meta-learning for instance-level tasks employ random sampling to construct this validation set for the meta-objective. However, we observe this to be detrimental for our setting as the model over-samples from the most populous categories and slot types ignoring their distribution and difficulty. To this end, we develop an adaptive mechanism to construct an informative validation set for meta-learning considering the diversity and uncertainty of the model for different slot types. Furthermore, we leverage this validation set to optimize the meta-objective for token-level loss estimation and re-weighting pseudo-labeled sequences from the teacher in a meta-learning framework.

Our task and framework overview. We focus on sequence labeling tasks with only a few annotated examples (e.g., $K = \{5, 10, 20, 100\}$) per slot type for training and large amounts of task-specific unlabeled data. Figure 1 shows an overview of our framework with the following components and research contributions:

(i) *Self-training:* Our self-training framework leverages a pre-trained language model as a teacher and co-trains a student model with iterative knowledge exchange for neural sequence tagging with very few manually annotated training labels.

(ii) *Adaptive validation set construction for meta-learning:* Our few-shot learning setup assumes a small number of labeled training samples per slot type that are not equally informative. We develop an adaptive mechanism to select informative examples to construct the validation set for our meta-objective. To this end, we leverage stochastic loss decay of the student model as a proxy for its uncertainty for sample selection. This strategy is used in conjunction with the re-weighting mechanism in the next step.

(iii) *Token-level re-weighting with meta-learning:* Since pseudo-labeled samples from the teacher can be noisy, we leverage a meta-objective to re-weight them to improve the student model performance on the validation set obtained in previous step. In contrast to prior work on instance-level re-weighting, we perform *token-level* re-weighting for slot tagging tasks. Finally, we learn all of the above steps jointly with end-to-end learning in the self-training framework. We refer to our adaptive self-training framework with meta-learning based sample re-weighting mechanism as MetaST.

(iv) *Experiments:* We perform extensive experiments on six benchmark datasets for several tasks including multilingual Named Entity Recognition and slot tagging for user utterances from task-oriented dialog systems to demonstrate the generalizability of our approach across diverse tasks, slots, shots and languages. We adopt BERT and multilingual BERT as encoders and show that their performance can be significantly improved by nearly 10% for the few-shot settings with very few training labels (e.g., 10 manually labeled examples per slot type) and large amounts of unlabeled data.

2 BACKGROUND AND PROBLEM FORMULATION

Sequence labeling and slot tagging. This is the task of identifying the entity *span* of several slot types (e.g., names of person, organization, location, date, etc.) in a text sequence. Formally, given a sentence with N tokens $X = \{x_1, ..., x_N\}$, an entity or slot value is a span of tokens $s = [x_i, ..., x_j](0 \leq i \leq j \leq N)$ associated with an entity class $c \in C$. This task assumes a predefined tagging policy like `BIO` [43], where `B` marks the beginning of the slot, `I` marks an intermediate token in the span, and `O` marks out-of-span tokens. These span markers are used to extract multi-token values for each of the slot types with phrase-level evaluation for the performance. For illustration, a user utterance can be labeled as "`play:O a:O popular:B-sort chant:B-music_item by:O brian:B-artist epstein:I-artist`", with slot types like *sort, music_item* and *artist*, with `BIO` denoting the span markers.

Few-shot semi-supervised sequence labeling. In this work, we study few-shot semi-supervised sequence labeling, where a model is trained with *very few manually labeled* and large amounts of unlabeled data. Formally, a few-shot semi-supervised setting for this task considers K labeled sentences that are manually annotated at token-level for each slot type $c \in C$, and M unlabeled sentences. The $K \times |C|$ labeled sentences are denoted as $(X^l_m = \{x^l_{m,n}\}, Y^l_m = \{y^l_{m,n}\})^{K \times |C|, N}_{m=1,n=1}$, where $y^l_{m,n} \in C$ are the slot labels. The M unlabeled sentences are denoted as $(X^u_m = \{x^u_{m,n}\})^{M,N}_{m=1,n=1}$, where $M \gg K \times |C|$. Let $f(X;\theta)$ denote a tagging model that assigns a label to each token in the sequence with trainable parameters θ.

Self-training is one of the earliest semi-supervised approaches and has recently shown state-of-the-art performance for instance-level classification tasks. Consider $f(\cdot;\theta_{tea})$ and $f(\cdot;\theta_{stu})$ to denote the teacher and student models respectively in the self-training framework. The role of the teacher model (e.g., a pre-trained language model) is to assign pseudo-labels to unlabeled data that is used to train a student model. The teacher and student model can exchange knowledge and the training schedules are repeated till convergence. The success of self-training with deep neural networks in recent works has been attributed to a number of factors including stochastic regularization with dropouts [14] and data regularization with unlabeled / augmented data [44]. Formally, given m-th unlabeled sentence with N tokens $X^u_m = \{x^u_{m,1}, ..., x^u_{m,N}\}$ and C pre-defined labels, consider the pseudo-labels $\hat{Y}^{(t)}_m = [\hat{y}^{(t)}_{m,1}, ..., \hat{y}^{(t)}_{m,N}]$ generated by the teacher model at the t-th iteration where,

$$\hat{y}^{(t)}_{m,n} = \arg\max_{c \in C} f_{n,c}(x^u_{m,n};\theta^{(t)}_{tea}). \quad (1)$$

The pseudo-labeled sequence data, denoted as $(X^u, \hat{Y}^{(t)}) = \{(x^u_{m,n}, \hat{y}^{(t)}_{m,n})\}^{M,N}_{m,n}$, is used to train the student model and learn its parameters as:

$$\hat{\theta}^{(t)}_{stu} = \arg\min_{\theta} \frac{1}{M}\frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N}\mathcal{L}(\hat{y}^{(t)}_{m,n}, f(x^u_{m,n};\theta^{(t-1)}_{stu})), \quad (2)$$

where $\mathcal{L}(\cdot,\cdot)$ can be modeled as the cross-entropy loss.

3 META SELF TRAINING

Given a pre-trained language model (e.g., BERT [10]) as the teacher, we first fine-tune it on the small labeled data with $K \times |C|$ annotated examples to make it aware of the underlying task. The fine-tuned teacher model is now used to pseudo-label the large unlabeled data. We consider the student model as another instantiation of the pre-trained language model that is trained over the pseudo-labeled data. However, our few-shot setting with limited labeled data results in a noisy teacher. A naive transfer of teacher knowledge to the student results in the propagation of noisy labels [37, 45] limiting the performance of the student model. To address this challenge, we develop an *adaptive* self-training framework to re-weight pseudo-labeled predictions from the teacher with a meta-learning objective that optimizes the token-level loss from the student model on a *judiciously constructed validation set* based on the model uncertainty (discussed next).

3.1 Adaptive Validation Set Construction for Meta-learning

Standard meta-learning techniques [36, 40] for instance-level classification tasks, construct the validation set to optimize the meta-objective via random sampling. However, a naive sample selection is detrimental for the sequence labeling setup involving many slot types with variable difficulty and distribution in the data and across languages. Therefore, we develop an adaptive strategy to construct the validation set for effective data exploration. We empirically demonstrate its benefit over classic meta-learning approaches from prior works in experiments.

Prior works in meta-learning and active learning broadly leverage random sampling [36, 40], easy [21] and hard example mining [39] or uncertainty-based methods [4] for sample selection. These strategies have been compared in prior works [4, 12] that show uncertainty-based methods to have better generalizability across diverse settings. While there are several approaches to uncertainty estimation including error decay [19] and predictive variance [4], these techniques have been developed for instance-level classification tasks, thereby, generating an overall estimate for the entire instance. In contrast, in this work, we are interested in leveraging token-level estimates corresponding to the different slot types and their associations.

Specifically, we leverage token-level uncertainty estimates to select samples that the model is uncertain about and can correspondingly benefit from knowing their labels. To this end, we leverage stochastic token-level loss decay from the model as a proxy for the model uncertainty to generate a validation set. This is used for estimating token-level weights and re-weighting pseudo labeled data in Section 3.2. This is an adaptive process as the model and corresponding uncertainty estimates improve over time, thereby, generating stochastic validation sets that are most representative of the difficulty of the underlying task at a given step during learning.

Consider the loss of the student model with parameters $\theta^{(t)}_{stu}$ on the labeled data $(\{x^l_{m,n}\}, \{y^l_{m,n}\})$ in the t-th iteration as $\mathcal{L}(\{y^l_{m,n}\}, \{f(x^l_{m,n}; \theta^{(t)}_{stu})\})$. We use the loss decay at any iteration as a proxy for the model uncertainty. This is measured by the difference between the successive stochastic losses encountered by the model for a token in any instance. Since the losses may widely vary across iterations given the few-shot assumption, we adopt the moving average of the stochastic losses for $(\{x^l_{m,n}\}, \{y^l_{m,n}\})$ in the latest R iterations as baseline $\mathcal{B}^{(t)}_m$ for smoothing the loss decay estimation. The baseline measure $\mathcal{B}^{(t)}_m$ at iteration t is given as:

$$\mathcal{B}^{(t)}_m = \frac{1}{min(R,t) \cdot N} \sum_{r=1}^{min(R,t)} \sum_{n=1}^{N} \mathcal{L}(y^l_{m,n}, f(x^l_{m,n}; \theta^{(t-r)}_{stu})). \quad (3)$$

Since the loss decay values are estimated on the fly, we want to balance exploration and exploitation. To this end, we add a smoothness factor δ to prevent the low loss decay samples (i.e. samples with low uncertainty in the constituent tokens) from never being selected again. Considering all of the above factors, we obtain the sampling weight of labeled data $(X^l_m = \{x^l_{m,n}\}, Y^l_m = \{y^l_{m,n}\})$ in iteration t as follows:

$$W^{(t)}_m \propto \max\left(\mathcal{B}^{(t)}_m - \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(y^l_{m,n}, f(x^l_{m,n}; \theta^{(t)}_{stu})), 0\right) + \delta. \quad (4)$$

A low value of $W^{(t)}_m$ indicates that the model loss for tokens in sequence $\{x^l_{m,n}\}$ in iteration t is similar to the average loss $\mathcal{B}^{(t)}_m$ encountered in last R iterations – depicting lower model uncertainty. In contrast, a higher value of $W^{(t)}_m$ depicts higher model uncertainty and therefore potential benefit in learning from knowing token-level labels, similar to the objective in an active learning setting.

The smoothness factor δ needs to be adaptive since the training loss is dynamic. To ensure the scale of smoothness factor δ is similar to loss decay value, we adopt the maximum of the loss decay values as the smoothness factor δ to encourage exploration.

For practical implementation considerations and speed-up, we re-estimate Equation 4 after a fixed number of steps to adapt to model changes and sample mini-batches of labeled data $\{\mathcal{V}^l_s\}$ as validation set for our meta-objective. This is used by the student model in the next step for re-weighting pseudo-labeled sequences from the teacher model. We demonstrate the impact of this adaptive sampling strategy via ablation study in experiments. As a minor note, the labeled data is only used to compute sample weight and not used for explicit training of the student model in this step.

3.2 Re-weighting Noisy Pseudo-Labeled Tokens

To mitigate error propagation from noisy pseudo-labeled sequences from the teacher, we leverage meta-learning to adaptively re-weight them based on the student model loss on the sampled validation set as our meta-objective. The validation set is obtained by our adaptive sampling strategy from the previous step. In contrast to prior work on instance-level image and text classification, we adapt the meta-learning framework to re-weight noisy pseudo-labeled samples at a token-level resolution for the sequence labeling task.

Consider the pseudo-labels $\{\hat{Y}^{(t)}_m = [\hat{y}^{(t)}_{m,1}, ..., \hat{y}^{(t)}_{m,N}]\}^M_{m=1}$ from the teacher in the t-th iteration with m and n indexing the instance and a token in the instance, respectively. In classic self-training, we update the student parameters leveraging pseudo-labels as follows:

$$\hat{\theta}^{(t)}_{stu} = \hat{\theta}^{(t-1)}_{stu} - \alpha\nabla\big(\frac{1}{M}\frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N} \mathcal{L}(\hat{y}^{(t)}_{m,n}, f(x^u_{m,n}; \hat{\theta}^{(t-1)}_{stu})\big). \quad (5)$$

Now, to downplay noisy token-level labels, we leverage meta-learning to re-weight pseudo-labeled data. Our objective is to measure the impact of a training example towards the performance on validation set $\mathcal{V}^l$ at iteration t. To this end, we leverage the idea of weight perturbation [18, 36] to change the weight of each token in each sequence of the mini-batch by $\epsilon^{(t)}_{m,n}$ at iteration t as:

$$\hat{\theta}^{(t)}_{stu}(\epsilon) = \hat{\theta}^{(t-1)}_{stu} - \alpha\nabla\big(\frac{1}{M}\frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N}[\epsilon^{(t)}_{m,n} \cdot \mathcal{L}(\hat{y}^{(t)}_{m,n}, f(x^u_{m,n}; \hat{\theta}^{(t-1)}_{stu}))]\big). \quad (6)$$

Weight perturbation is used to discover data points that are most important to improve the model performance on the validation set where the sample importance is given by the magnitude of the negative gradients. We can now find the optimal value for the perturbation $\epsilon^{(t)*}_{m,n}$ that minimizes the student model loss on the validation set $\mathcal{V}^l$ at iteration t as:

$$\epsilon^{(t)*}_{m,n} = argmin_{\epsilon_{m,n}} \frac{1}{M}\frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N} \mathcal{L}(\hat{y}^{(t)}_{m,n}, f(x^u_{m,n}; \hat{\theta}^{(t)}_{stu}(\epsilon_{m,n})) \quad (7)$$

The token weights are obtained by minimizing the student model loss on sampled mini-batches of validation data $\{\mathcal{V}^l_s\}$ obtained from Eq. 4. To obtain a cheap estimate of the meta-weight at step t, we take a single gradient descent step for the sampled validation mini-batch $\mathcal{V}^l_s$ as:

$$u^{(t)}_{m,n,s} = -\frac{\partial}{\partial\epsilon_{m,n,s}}\left(\frac{\sum_{m=1}^{|\mathcal{V}^l_s|}\sum_{n=1}^{N} \mathcal{L}(y^l_{m,n}, f(x^l_{m,n}; \hat{\theta}^{(t)}_{stu}(\epsilon)))}{|\mathcal{V}^l_s| \cdot N}\right)\Bigg|_{\epsilon_{m,n,s}=0} \quad (8)$$

We set the token weights to be proportional to the negative gradients to reflect the importance of pseudo-labeled tokens in the sequence. Since sequence labeling tasks have dependencies between the slot types and tokens, it is difficult to obtain a good estimation of the weights based on a single mini-batch of examples. Therefore, we sample S mini-batches of validation sets $\{\mathcal{V}^l_1, ..., \mathcal{V}^l_S\}$ with the adaptive sampling strategy in Equation 4 and calculate the mean of the gradients to obtain a robust gradient estimate. The overall meta-weight of pseudo-labeled token $(x^u_{m,n}, \hat{y}_{m,n})$ is obtained as:

$$w^{(t)}_{m,n} = \max(\frac{1}{S}\sum_{s=1}^{S} u^{(t)}_{m,n,s}, 0). \quad (9)$$

Since a negative weight indicates a pseudo-label of poor quality that would potentially degrade the model performance, we set such weights to 0 to filter them out. We empirically study the impact of S in experiments.

Finally, we update the student model parameters while accounting for token-level re-weighting as:

$$\hat{\theta}_{stu}^{(t)} = \hat{\theta}_{stu}^{(t-1)} - \alpha \nabla \Big(\frac{1}{M} \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N} [w_{m,n}^{(t)} \cdot \mathcal{L}(\hat{y}_{m,n}^{(t)}, f(x_{m,n}^{u}; \hat{\theta}_{stu}^{(t-1)}))] \Big). \tag{10}$$

We demonstrate the impact of this token-level re-weighting mechanism with ablation study in experiments.

3.3 Student Teacher Iterative Training

We first fine-tune the teacher model with few labeled data for each slot for each task and initialize the student as a copy of the teacher. In every self-training iteration, the teacher generates noisy token-level pseudo-labels for each sequence which are used to train the student model with sample selection and token-level re-weighting in a meta-learning framework.

At the end of every self-training iteration, we assign the student model to be the new teacher model (i.e., $\theta_{tea} = \theta_{stu}^{(T)}$), and repeat the above steps till convergence. We further utilize the labeled data ($\{x_{m,n}^{l}, y_{m,n}^{l}\}$) to fine-tune the new teacher model $f(\cdot, \theta_{tea}^{(t)})$ with standard cross-entropy loss minimization. We explore the effectiveness of this step with an ablation study in experiments. The overall training procedure is summarized in Algorithm 1.

Algorithm 1: MetaST Algorithm.

```
Input: Labeled sequences ($X^l = \{x_{m,n}^l\}, Y^l = \{y_{m,n}^l\}$); Unlabeled sequences
       ($X^u = \{x_{m,n}^u\}$); Pre-trained BERT model with randomly initialized token
       classification layer $f(\cdot; \theta^{(0)})$; Number of mini-batches $S$; Number of
       self-training iterations $T$.
Initialize teacher model $\theta_{tea} = \theta^{(0)}$
while not converged do
    Fine-tune teacher model on small labeled data ($X^l, Y^l$);
    Initialize the student model $\theta_{stu}^{(0)} = \theta^{(0)}$;
    Generate hard pseudo-labels $\hat{Y}^{(t)} = \{\hat{y}_{m,n}^{(t)}\}$ for unlabeled sequences
      $X^u = \{x_{m,n}^u\}$ with model $f(\cdot, \theta_{tea})$;
    for $t \leftarrow 1$ to $T$ do
        Sample $S$ mini-batches of labeled validation sets $\{\mathcal{V}_1^l, ..., \mathcal{V}_S^l\}$ from $(X^l, Y^l)$
        based on adaptive sample selection strategy in Eq. 4;
        Randomly sample a batch of pseudo-labeled sequences $\mathcal{V}_u$ from $(X^u, \hat{Y}^{(t)})$;
        Compute token-level weights in $\mathcal{V}_u$ based on the loss on $\{\mathcal{V}_1^l, ..., \mathcal{V}_S^l\}$
          according to Eq. 9;
        Train model $f(\cdot, \theta_{stu}^{(t)})$ on re-weighted token-level pseudo-labeled
          sequences $\mathcal{V}_u$ and update parameters $\theta_{stu}^{(t)}$ ;
    end
    Update the teacher: $\theta_{tea} = \theta_{stu}^{(T)}$
end
```

4 EXPERIMENTS

We evaluate the proposed method MetaST across diverse tasks, slot (entity) types, number of shots (manually labeled instances) and languages to demonstrate its impact for the few-shot learning setup for sequence labeling with limited amount of training labels. We compare against several state-of-the-art existing methods and demonstrate significant improvements in diverse settings along with ablation studies to evaluate the contribution of different components.

4.1 Experimental Setup

Datasets. We perform large-scale experiments with six different datasets including user utterances from task-oriented dialog systems and multilingual Named Entity Recognition tasks as summarized in Table 1. *(a) Email.* This consists of natural language user utterances for email-oriented user actions like sending, receiving or searching emails with attributes like date, time, topics, and people. *(b) SNIPS* is a public benchmark dataset [9] of user queries from multiple domains including music, media, and weather. *(c) MIT Movie and Restaurant* corpus [26] consist of similar user utterances for movie and restaurant domains. (d) CoNLL03 [38] and Wikiann [31] are public benchmark datasets for multilingual Named Entity Recognition. CoNLL03 is a collection of news wire articles from the Reuters Corpus from 4 languages with manual annotations, whereas Wikiann comprises of extractions from Wikipedia articles from 41 languages with automatic annotation leveraging meta-data for different entity types like `ORG, PER, LOC`.

For every dataset, we sample $K \in \{5, 10, 20, 100\}$ manually labeled sequences for each slot type from the training data, and add the remaining to the unlabeled set while ignoring their labels – following standard setups for semi-supervised learning. We repeatedly sample K labeled instances three times for multiple runs and report average F1 score with standard deviation across the runs.

Table 1: Dataset summary. We sample $K \in \{5, 10, 20, 100\}$ labeled sequences for each slot type from #Train, and add the remaining to the Unlabeled set while ignoring their labels.

Dataset	#Slots	#Train/ #Unlabeled	#Test	#Lang
Email	20	2.5K	1k	EN
SNIPS	39	13K	0.7K	EN
MIT Movie	12	8.8K	2.4K	EN
MIT Restaurant	8	6.9K	1.5K	EN
Wikiann (EN)	3	20K	10K	EN
CoNLL03 (EN)	4	15K	3.6K	EN
CoNLL03	16	38K	15K	4
Wikiann	123	705K	329K	41

Encoder. Pre-trained language models like BERT [10], GPT-2 [35] and RoBERTa [27] have shown state-of-the-art performance for various natural language processing tasks. In this work, we adopt one of them as a base encoder by initializing the teacher with pre-trained BERT-base model and a randomly initialized token classification layer.

Baselines. The first baseline we consider is the fully supervised BERT model trained on all available training data which provides the ceiling performance for every task. Each of the other models are trained on K training labels per slot type. We adopt several state-of-the-art semi-supervised methods as baselines: (1) CVT [8] is a semi-supervised sequence labeling method based on cross-view training. For unlabeled data, CVT matches auxiliary prediction based on parts of a sentence with prediction based on the whole input to improve its representation learning. (2) SeqVAT [6] incorporates adversarial training with conditional random field layer for semi-supervised sequence labeling. (3) Mean Teacher (MT) [41] averages model weights to obtain an aggregated teacher and applies a consistency loss between the predictions from the student model and that from

Table 2: F1 score comparison of models for sequence labeling on different datasets averaged over multiple runs. All models (except CVT and SeqVAT) use the same BERT encoder. F1 score of our model for each task is followed by standard deviation and percentage improvement (std dev; ↑) over BERT with 10 manually labeled training examples per slot.

Method	SNIPS	Email	Movie	Restaurant	CoNLL03 (EN)	Wikiann (EN)
# Slots	39	20	12	8	4	3
Full-supervision						
BERT	95.80	94.44	87.87	78.95	92.40	84.04
Few-shot supervision (10 labels per slot)						
BERT	79.01	87.85	69.50	54.06	71.15	45.61
Few-shot supervision (10 labels per slot) + unlabeled data						
CVT	78.23	78.24	62.73	42.57	54.31	27.89
SeqVAT	78.67	72.65	67.10	51.55	67.21	35.16
MT	79.48	89.53	67.62	51.75	68.67	41.43
VAT	79.08	89.71	70.17	53.34	65.03	38.81
Classic ST	83.26	90.70	71.88	56.80	70.99	46.15
BOND	83.54	89.75	70.91	55.78	69.56	48.73
MetaST	**88.23** **(0.04;↑12%)**	**92.18** **(0.47;↑4.93%)**	**77.67** **(0.10;↑11.76%)**	**63.83** **(1.62;↑18.07%)**	**76.65** **(0.73;↑7.73%)**	**56.61** **(0.4;↑24.12%)**

Table 3: F1 score comparison of models for sequence labeling on multilingual datasets using the same multilingual mBERT encoder. F1 score of MetaST for each task is followed by standard deviation in parentheses and percentage improvement (↑) over mBERT with 10 manually labeled training examples per slot.

Dataset	#Lang	#Slots	Full Sup.	10 labels per slot	10 labels per slot + unlabeled data				
			mBERT	mBERT	MT	VAT	Classic ST	BOND	MetaST
CoNLL03	4	16	87.67	70.77	68.34	67.63	72.69	72.79	**76.41 (0.47) (↑ 7.97%)**
Wikiann	41	123	87.17	79.67	80.23	78.82	80.24	79.57	**81.61 (0.14) (↑ 2.42%)**

the aggregated teacher on unlabeled data. (4) VAT [30] improves the robustness of the conditional label distribution for each input data point against local perturbation. (5) Classic ST [15] is simple self-training method with hard pseudo-labels; (6) BOND[1] [25] is a recent work on self-training for sequence labeling with confidence-based sample selection and forms a strong baseline for our work.

The above semi-supervised learning (SSL) methods augment task-specific knowledge from manually annotated data with domain knowledge from unlabeled data. In contrast to traditional SSL methods, few-shot learning settings involve *very few* manually annotated training labels resulting in a noisy / weak model to start with. Consequently, a naive augmentation from large amounts of unlabeled data results in drift without accounting for the noise and model uncertainty. To this end, we develop a robust sample selection and re-weighting mechanism for adaptive learning.

We implement our framework[2] in Pytorch and use Tesla V100 gpus for experiments. Hyper-parameter configurations with model settings are presented in Appendix.

4.2 Experimental Results

We first present the overall performance comparison of MetaST with several state-of-the-art methods for few-shot sequence labeling followed by several control experiments.

10-shot sequence labeling performance comparison. Table 2 shows the performance comparison among different models with K=10 labeled examples per slot type. The fully supervised BERT baseline trained on thousands of labeled examples provides the ceiling performance for the few-shot setting. We observe that the proposed method MetaST significantly outperforms all other methods across all datasets – including the models that also use the same BERT encoder as ours like MT, VAT, Classic ST and BOND with corresponding average performance improvements as 14.22%, 14.90%, 8.46% and 8.82% respectively. This demonstrates the advantage of our adaptive / meta self-training design. Non-BERT models like CVT and SeqVAT are consistently worse than other baselines.

Task variation. We also observe variable performance of the models across different tasks. Specifically, the performance gap between the best few-shot model and the fully supervised model varies significantly across tasks. MetaST achieves close performance to the fully-supervised model in some datasets (e.g. SNIPS and Email) but has bigger room for improvement in others (e.g. CoNLL03 (EN) and Wikiann (EN)). This can be attributed to the following factors.

(i) *Labeled training examples and slots.* The total number of labeled training instances for our K-shot setting is given by $K \times \#Slots$. Therefore, for tasks with higher number of slots and consequently more training labels, most of the models perform better including MetaST. Task-oriented dialog systems with more slots and inherent dependency between the slot types benefit more than NER tasks.

[1] We replace fine-tuning step with distant supervision by fine-tuning on labeled data.
[2] Our anonymized code is available at tinyurl.com/53osvuju

(ii) *Task difficulty:* User utterances from task-oriented dialog systems for some of the domains like weather, music and emails contain predictive query patterns and limited diversity. In contrast, Named Entity Recognition datasets are comparatively diverse and require more training labels to generalize well. Similar observations are also depicted in Table 3 for multilingual NER tasks with more slots and consequently more training labels from multiple languages as well as richer interactions across the slots from different languages.

Table 4: F1 scores of different models with 200 manually labeled examples for each task. The percentage improvement (↑) is over the BERT model with few-shot supervision.

Dataset	BERT (Full Sup.)	BERT (Few-shot Sup.)	MetaST (%Improvement)
MIT Movie	87.87	75.81	**80.33 (↑ 5.96%)**
MIT Restaurant	78.95	60.12	**67.86 (↑ 12.87%)**
CoNLL03 (EN)	92.40	77.48	**81.61 (↑ 5.33%)**
Wikiann (EN)	84.04	62.04	**71.27 (↑ 14.88%)**
Average	85.82	68.86	**75.27 (↑ 9.31%)**

Controlling for the total amount of labeled data. In order to control for the variable amount of training labels across different datasets / tasks, we perform another experiment where we vary the number of training labels for different slot types while keeping the total number of labeled instances for each dataset similar (ca. 200). Results are shown in Table 4. To better illustrate the effect of the number of training labels, we choose tasks with lower performance in Table 2 for this experiment. Comparing the results in Tables 2 and 4, we observe that the performance of MetaST improves with more training labels for all the tasks .

Table 5: Variation in model performance on varying K training labels per slot on SNIPS dataset with 39 slots. The percentage improvement (↑) is over the BERT model with few-shot supervision.

#Shots	5	10	20	100
Few-shot supervision				
BERT	70.63	79.01	86.81	93.90
Few-shot supervision + unlabeled data				
CVT	69.82	78.23	86.81	94.61
SeqVAT	69.34	78.67	85.05	91.46
MT	70.85	79.48	87.31	94.26
VAT	71.34	79.08	88.19	94.53
Classic ST	72.59	83.26	88.32	93.92
BOND	72.85	83.54	88.93	94.22
MetaST	**81.56** **(↑15%)**	**88.22** **(↑12%)**	**91.99** **(↑6%)**	**95.39** **(↑2%)**

Effect of varying the number of training labels K per slot. Table 5 shows the improvement in the performance of MetaST when increasing the number of training labels for each slot type in the SNIPS dataset. Similar trends can be found in other datasets (results in Appendix). As we increase the amount of labeled training instances, the performance of BERT and all the models improve. Correspondingly, the relative improvement between MetaST and the baselines decreases although MetaST still improves over all of them. For example, while MetaST improves over BERT by 15% for the 5-shot setting, the corresponding improvement reduces to 2% for the 100-shot setting.

In the self-training framework, given the ceiling performance for every task and the improved performance of the teacher with more training labels – there is less room for (relative) improvement of the student over the teacher model. Consider SNIPS for an illustration. Our model obtains 12% and 2% improvement over the few-shot BERT model for the 10-shot and 100-shot setting with F1-scores as 88.22% and 95.39%, respectively. The ceiling performance for this task at 95.8% is obtained by training BERT on the fully labeled dataset with $13K$ labeled examples. This demonstrates that MetaST is most impactful for low-resource settings with few training labels for a given task.

4.3 Ablation analysis

Table 6 demonstrates the impact of different MetaST components with ablation analysis. We observe that soft pseudo-labels hurt the model performance compared to hard pseudo-labels, as also shown in recent work [20]. Such a performance drop may be attributed to soft labels being less informative compared to sharpened ones. Removing the iterative teacher fine-tuning step (Section 3.1) also hurts the overall performance.

Continued pre-training versus self-training. Recent work [13] show the benefit of continued pre-training with task-specific unlabeled data for adapting pre-trained language models to the task-domain. To contrast continued pre-training with self-training, we *further pre-train* BERT with masked language modeling objective on in-domain unlabeled data and then fine-tune it with few labeled examples denoted as "BERT (Continued Pre-training + Few-shot Supervision)". The pre-training step improves BERT performance over the baseline on SNIPS but degrades the performance on CoNLL03. This indicates that continued pre-training can improve the performance of few-shot supervised BERT on specialized tasks (e.g., SNIPS) with different data distribution than the original pre-training data (e.g., Wikipedia), but may not help for general domain ones like CoNLL03 with overlapping data from Wikipedia. In contrast to the above baseline, MetaST brings significant improvements on both datasets. This demonstrates the generality and flexibility of self-training over pre-training as also observed in contemporary work [46] on image classification.

Adaptive Validation Set Construction. We perform an ablation study by removing adaptive validation set construction from MetaST (denoted as "MetaST w/o Adaptive Valid Set Construction"). Removing this component leads to around 2% performance drop on an average demonstrating the impact of adaptive validation set for meta-learning. Moreover, the performance drop on SNIPS (39 slots) is larger than that on CoNLL03 (4 slots). This demonstrates that adaptive validation set construction is more helpful for tasks with more slot types – where diversity and data distribution necessitate a better exploration strategy in contrast to random sampling employed in prior meta-learning works.

Re-weighting strategies. To explore the role of token-level reweighting for pseudo-labeled sequences (discussed in Section 3.2), we replace our meta-learning component with different data selection strategies based on the model confidence. One data selection strategy chooses pseudo-labeled tokens uniformly without any re-weighting (referred to as "MetaST w/o Re-weighting"). The

Table 6: Ablation analysis of our framework MetaST with 10 labeled examples per slot on SNIPS and CoNLL03 (EN).

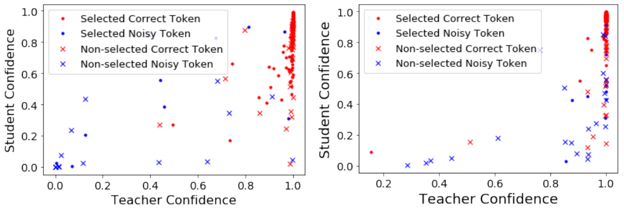
Method	Datasets	
	SNIPS	CoNLL03
BERT w/ Few-shot Supervision	79.01	71.15
BERT w/ Continued Pre-training + Few-shot Supervision	83.96	69.84
Classic ST w/ Hard Pseudo-Labels	83.26	70.99
Classic ST w/ Soft Pseudo-Labels	81.17	71.87
MetaST w/ Soft Pseudo-Labels	86.16	75.84
MetaST w/o Iterative Teacher Fine-tune	85.64	72.74
MetaST w/o Adaptive Valid Set Construction	86.63	75.02
Pseudo-labeled Data Selection and Re-weighting Strategies		
MetaST w/o Re-weighting	85.48	73.02
MetaST (Easy Sample Selection)	85.56	74.53
MetaST (Difficult Sample Selection)	86.34	68.06
MetaST (Instance-level Re-weighting)	86.46	74.54
MetaST (ours) w/ Hard Pseudo-Labels, Token-level Re-weighting, Adaptive Valid Set Construction	**88.23**	**76.65**

sampling strategy with weights proportional to the model confidence favors easy instances (referred to as "MetaST (Easy Sample Selection)"), whereas the converse favors difficult ones (referred to as "MetaST (Difficult Sample Selection)"). We observe that the meta-learning based re-weighting strategy performs the best. Interestingly, "MetaST (Easy Sample Selection)" outperforms "MetaST (Difficult Sample Selection)" significantly on CoNLL03 (EN) but achieves slightly lower performance on SNIPS. This demonstrates that difficult samples are more helpful when the quality of pseudo-labeled data is relatively high. In contrast, the sample selection strategy focusing on difficult samples introduces noisy examples with lower pseudo-label quality. Therefore, sampling strategies may need to vary for different datasets, thereby, demonstrating the necessity of adaptive data re-weighting as in our framework MetaST. Moreover, MetaST significantly outperforms classic self-training strategies with hard and soft pseudo-labels demonstrating the effectiveness of our design.

Token-level re-weighting versus instance-level re-weighting. Prior meta-learning works [36] re-weight entire instances for classification tasks. In order to compare our token-level re-weighting mechanism for sequence labeling tasks, we replace our token-level re-weighting component by sentence-level re-weighting – which assigns uniform weights to all the tokens in the same sentence (referred to as "MetaST (Instance-level Re-weighting)"). Table 6 shows that token-level re-weighting outperforms instance-level re-weighting on SNIPS and CoNLL03 by 2.05% and 2.76% respectively, demonstrating the benefit of token-level choice for sequence labeling.

Analysis of pseudo-labeled data re-weighting. To visually explore the adaptive re-weighting mechanism, we illustrate token-level re-weighting of MetaST on SNIPS and CoNLL03 (EN) datasets with K=10 shot at step 100 in Fig. 2. We observe that the selection mechanism filters out most of the noisy pseudo-labels (colored in blue) including even those with high teacher confidence (X-axis).

Variation in model performance with mini-batch S. We explore different values of $S \in \{1, 3, 5\}$ in Eq. 9 to find its impact on the re-weighting mechanism. From Figure 3, we observe that the model is not super sensitive to the value of hyper-parameter S, although it achieves a better estimate of the weights of the pseudo-labeled data with increasing mini-batch values. The relative performance improvement diminishes as the mini-batch size increases.



(a) SNIPS

(b) CoNLL03

Figure 2: Visualization of MetaST re-weighting examples on SNIPS and CoNLL03 (EN).

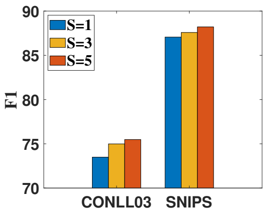


Figure 3: Varying S mini-batch labeled data for re-weighting.

Variation in model performance with unlabeled data. We conduct experiments to show the performance change in MetaST with varying proportions of unlabeled data. Table 7 shows the improvement in model performance as we inject more unlabeled data with diminishing returns after a certain point.

Table 7: Varying proportions of unlabeled data for MetaST with 10 training labels per slot.

Ratio of Unlabeled Data	Datasets	
	SNIPS	CoNLL03
5%	84.47	72.92
25%	87.10	76.46
75%	87.50	76.56

5 RELATED WORK

Meta-learning. Prior works [1, 22, 42] on meta-learning develop models that can adapt to new environment (e.g., new or unseen classes and tasks) while optimizing a meta-objective over some representative samples. While recent works [2, 11, 24] on meta-learning for image and text classification leverage *multi-task* learning to improve a target classification task based on several similar tasks, in this work we focus on a single sequence labeling task – making our setup more challenging.

Sample selection. Curriculum learning [3] techniques are based on the idea of learning easier aspects of the task *first* followed by the more complex ones. For self-training, this amounts to using the easy samples first followed by the difficult ones. Prior work leveraging self-paced learning [21] and more recently self-paced co-training [28] leverage teacher confidence to select easy samples during training. This is based on the assumption that a set of samples is considered easy if it admits a good fit in the model space. Sample selection for image classification tasks have been explored in recent works with meta-learning [23, 36] and active learning [4, 32]. However, all of these techniques rely on only the model outputs applied to instance-level classification tasks.

Semi-supervised learning and sequence labeling. Semi-supervised learning for instance-level classification has been used in [17, 30, 41]. As we show in this work, a vanilla extension of these techniques to sequence labeling tasks ignore the inter-dependencies, diversity and slot distribution resulting in subpar performance.

For sequence labeling tasks, [29, 33] leverage large amounts of unlabeled data to improve token representation in addition to decent amounts of labeled training data. Another line of research introduces latent variable modeling [7], adversarial training method SeqVAT [6] and cross-view training method CVT [8] to obtain promising results. However, these techniques do not work well for our *few-shot* learning setup as they ignore the model uncertainty and resulting noise from very few training labels.

6 CONCLUSIONS

In this work, we develop an adaptive self-training framework MetaST that leverages self-training and meta-learning for few-shot training of neural sequence taggers. We address the issue of error propagation from noisy pseudo-labels from the teacher in the self-training framework by adaptive sample selection and re-weighting with meta-learning. Extensive experiments on six benchmark datasets and different tasks including multilingual NER and slot tagging for task-oriented dialog systems demonstrate the effectiveness of the proposed method particularly for low-resource settings with more than 10% improvement over state-of-the-art methods.

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3988–3996.

[2] Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020. Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*. 5108–5123.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, Vol. 382. ACM, 41–48.

[4] Haw-Shiuan Chang, Erik G. Learned-Miller, and Andrew McCallum. 2017. Active Bias: Training More Accurate Neural Networks by Emphasizing High Variance Samples. In *Advances in Neural Information Processing Systems 30, 2017*. 1002–1012.

[5] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. 2010. Semi-Supervised Learning. (2010).

[6] Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020. SeqVAT: Virtual Adversarial Training for Semi-Supervised Sequence Labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8801–8811.

[7] Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2019. Variational sequential labelers for semi-supervised learning. *arXiv preprint arXiv:1906.09535* (2019).

[8] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. 2018. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370* (2018).

[9] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. In *Privacy in Machine Learning and Artificial Intelligence workshop, ICML2018*.

[10] 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, Volume 1 (Long and Short Papers)*. 4171–4186.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, Vol. 70. PMLR, 1183–1192.

[13] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. *arXiv preprint arXiv:2004.10964* (2020).

[14] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting Self-Training for Neural Sequence Generation. arXiv:1909.13788 [cs.LG]

[15] H. J. Scudder III. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theory* 11, 3 (1965), 363–371. https://doi.org/10.1109/TIT.1965.1053799

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

[17] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems 27, 2014*. 3581–3589.

[18] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730* (2017).

[19] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2017. Learning active learning from data. In *Advances in Neural Information Processing Systems*.

[20] Ananya Kumar, Tengyu Ma, and Percy Liang. 2020. Understanding Self-Training for Gradual Domain Adaptation. *arXiv preprint arXiv:2002.11361* (2020).

[21] M. P. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 1189–1197.

[22] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017).

[23] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. 2019. Learning to Self-Train for Semi-Supervised Few-Shot Classification. In *Advances in Neural Information Processing Systems 32*.

[24] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* (2017).

[25] 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*. 1054–1064.

[26] J. Liu, Panupong Pasupat, D. Cyphers, and James R. Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), 8386–8390.

[27] 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 (2019). arXiv:1907.11692

[28] Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. 2017. Self-Paced Co-training. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*. 2275–2284.

[29] Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. 337–342.

[30] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* 41 (2018).

[31] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual Name Tagging and Linking for 282 Languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada, 1946–1958.

[32] Emmeleia Panagiota Mastoropoulou. 2019. *Enhancing Deep Active Learning Using Selective Self-Training For Image Classification*. Master's thesis. KTH, School of Electrical Engineering and Computer Science (EECS).

[33] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108* (2017).

[34] Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. (2012).

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[36] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. In *International Conference on Machine Learning*. 4334–4343.

[37] Sebastian Ruder and Barbara Plank. 2018. Strong Baselines for Neural Semi-Supervised Learning under Domain Shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 1044–1054.

[38] Erik F. Tjong Kim Sang and Fien De Meulder. [n.d.]. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

[39] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 761–769.

[40] 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. *arXiv preprint arXiv:1902.07379* (2019).

[41] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *5th International Conference on Learning Representations, ICLR 2017*.

[42] Sebastian Thrun and Lorien Pratt. 1998. Learning to learn: Introduction and overview. In *Learning to learn*. 3–17.

[43] Erik F Tjong, Kim Sang, and Jorn Veenstra. 1999. Representing Text Chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*.

[44] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-Training With Noisy Student Improves ImageNet Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[45] 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*.

[46] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems* 33 (2020).

A IMPLEMENTATIONS AND HYPER-PARAMETER

The training batch size for teacher model fine-tuning with few-shot supervision is 16, which is same for all the datasets with different settings. We re-estimate Equation 4 every 10 steps to adapt to model changes. The maximum sequence length varies due to dataset characteristics and is as shown in Table 8. The hyper-parameters are as shown in Table 9.

Table 8: Labeled batch size, unlabeled batch size and BERT encoder choices across datasets

Dataset	Labeled Batch Size $\|\mathcal{V}^l\|$	Unlabeled Batch Size	BERT Encoder
SNIPS	32	32	base-uncased
Email	32	32	base-cased
Movie	32	32	base-uncased
Restaurant	16	32	base-uncased
CoNLL03 (EN)	8	32	base-cased
Wikiann (EN)	8	32	base-cased
CoNLL03	32	32	multilingual-base-cased
Wikiann	32	32	multilingual-base-cased

Table 9: Hyper-parameters.

BERT attention dropout	0.3
BERT hidden dropout	0.3
Latest Iteration R in labeled data acquisition	5
BERT output hidden size h	768
Steps for fine-tuning teacher model on labeled data	2000
Steps T for self-training model on unlabeled data	3000
Mini-batch S	5
Re-initialize Student	Y
Pseudo-label Type	Hard
Warmup steps	20
learning rate α	$5e^{-5}$
Weight_decay	$5e^{-6}$

Also, we retain parameters from original BERT implementation from https://github.com/huggingface/transformers.

We implement SeqVAT based on https://github.com/jiesutd/NCRFpp and implement CVT following https://github.com/tensorflow/models/tree/master/research/cvt_text.

Most of the datasets can be accessed via https://github.com/juand-r/entity-recognition-datasets.

B K-SHOTS

Effect of varying the number of few-shots K. We show the performance changes with respect to varying number of few-shots K $\{5, 10, 20, 100\}$ on Wikiann (en), MIT movie, MIT Restaurant, CoNLL2003 (En), Multilingual CoNLL and Multilingual Wikiann in Table 9-16. Since the number of labeled examples for some slots in Email dataset are less than 20, we only show 5 and 10 shots for Email dataset in Table 8.

Table 10: Email Dataset.

Method	Shots	
	5	10
Full-supervision		
BERT	0.9444	
Few-shot Supervision		
BERT	0.8211	0.8785
Few-shot Supervision + unlabeled data		
CVT	67.44	78.24
SeqVAT	64.67	72.65
Mean Teacher	84.10	89.53
VAT	83.24	89.71
Classic ST	86.88	90.70
BOND	84.92	89.75
MetaST	89.21	92.18

Table 11: Wikiann (En) Dataset.

Method	Shots (3 Slot Types)			
	5	10	20	100
Full-supervision				
BERT	84.04			
Few-shot Supervision				
BERT	37.01	45.61	54.53	67.87
Few-shot Supervision + unlabeled data				
CVT	16.05	27.89	46.42	66.36
SeqVAT	21.11	35.16	42.26	62.37
Mean Teacher	30.92	41.43	50.61	67.16
VAT	24.72	38.81	50.15	66.31
Classic ST	32.72	46.15	54.41	68.64
BOND	34.22	48.73	52.45	68.89
MetaST	55.04	56.61	60.38	73.20

Table 12: MIT Movie Dataset.

Method	Shots (12 Slot Types)			
	5	10	20	100
Full-supervision				
BERT	87.87			
Few-shot Supervision				
BERT	62.80	69.50	75.81	82.49
Few-shot Supervision + unlabeled data				
CVT	57.48	62.73	70.20	81.82
SeqVAT	60.94	67.10	74.15	82.73
Mean Teacher	58.92	67.62	75.24	82.20
VAT	60.75	70.17	75.41	82.39
Classic ST	63.39	71.88	76.58	83.06
BOND	62.50	70.91	75.52	82.65
MetaST	72.57	77.67	80.33	84.35

Table 13: MIT Restaurant Dataset.

Method	Shots (8 Slot Types)			
	5	10	20	100
Full-supervision				
BERT	78.95			
Few-shot Supervision				
BERT	41.39	54.06	60.12	72.24
Few-shot Supervision + unlabeled data				
CVT	33.74	42.57	51.33	70.84
SeqVAT	41.94	51.55	56.15	71.39
Mean Teacher	40.37	51.75	57.34	72.40
VAT	41.29	53.34	59.68	72.65
Classic ST	44.35	56.80	60.28	73.13
BOND	43.01	55.78	59.96	73.60
MetaST	53.02	63.83	67.86	75.25

Table 14: CoNLL2003 (EN)

Method	Shots (4 Slot Types)			
	5	10	20	100
Full-supervision				
BERT	92.40			
Few-shot Supervision				
BERT	63.87	71.15	73.57	84.36
Few-shot Supervision + unlabeled data				
CVT	51.15	54.31	66.11	81.99
SeqVAT	58.02	67.21	74.15	82.20
Mean Teacher	59.04	68.67	72.62	84.17
VAT	57.03	65.03	72.69	84.43
Classic ST	64.04	70.99	74.65	84.93
BOND	62.52	69.56	74.19	83.87
MetaST	71.49	76.65	78.54	85.77

Table 15: Multilingual CoNLL03.

Method	Shots (4 Slot Types)			
	5	10	20	100
Full-supervision				
BERT	87.67			
Few-shot Supervision				
BERT	64.80	70.77	73.89	80.61
Few-shot Supervision + unlabeled data				
Mean Teacher	64.55	68.34	73.87	79.21
VAT	64.97	67.63	74.26	80.70
Classic ST	67.95	72.69	73.79	81.82
BOND	69.42	72.79	76.02	80.62
MetaST	73.34	76.65	77.01	82.11

Table 16: Multilingual Wikiann

Method	Shots (3 Slot Types × 41 languages)			
	5	10	20	100
Full-supervision				
BERT	87.17			
Few-shot Supervision				
BERT	77.68	79.67	82.33	85.70
Few-shot Supervision + unlabeled data				
Mean Teacher	77.09	80.23	82.19	85.34
VAT	74.71	78.82	82.60	85.82
Classic ST	76.73	80.24	82.39	86.08
BOND	78.81	79.57	82.19	86.14
MetaST	79.10	81.61	83.14	85.57

Table 17: Sentence examples from public datasets.

Dataset	**Examples**
Wikiann (EN)	First recorded in the Serranía de las Quinchas on January 17, 2006.
Wikiann (Multilingual)	Sy ander seun, Swjatopolk, was die resultaat van 'n buite-egtelike verhouding.
MIT Movie	show me films with drew barrymore from the 1980s
MIT Restaurant	any beef cuisine restaurants with brewpub and great prices
SNIPS	listen towestbam alumb allergic on google music
CoNLL03 (EN)	Japan then laid siege to the Syrian penalty area for most of the game but rarely breached the Syrian defence.
CoNLL03 (Multilingual)	Bekanntlich war bei Ihnen so ziemlich die ganze deutsche Kabarett-Prominenz zu Gast.