Self-Training with Weak Supervision

Giannis Karamanolakis[§] [*] **Subhabrata Mukherjee**[†]
Guoqing Zheng[†] **Ahmed Hassan Awadallah**[†]
[§]Columbia University, New York [†]Microsoft Research
gkaraman@cs.columbia.edu
{submukhe, zheng, hassanam}@microsoft.com

Abstract

State-of-the-art deep neural networks require large-scale labeled training data that is often expensive to obtain or not available for many tasks. Weak supervision in the form of domain-specific rules has been shown to be useful in such settings to automatically generate weakly labeled training data. However, learning with weak rules is challenging due to their inherent heuristic and noisy nature. An additional challenge is rule coverage and overlap, where prior work on weak supervision only considers instances that are covered by weak rules, thus leaving valuable unlabeled data behind.

In this work, we develop a weak supervision framework (ASTRA[1]) that leverages all the available data for a given task. To this end, we leverage task-specific unlabeled data through self-training with a model (student) that considers contextualized representations and predicts pseudo-labels for instances that may not be covered by weak rules. We further develop a rule attention network (teacher) that learns how to aggregate student pseudo-labels with weak rule labels, conditioned on their fidelity and the underlying context of an instance. Finally, we construct a semi-supervised learning objective for end-to-end training with unlabeled data, domain-specific rules, and a small amount of labeled data. Extensive experiments on six benchmark datasets for text classification demonstrate the effectiveness of our approach with significant improvements over state-of-the-art baselines.

1 Introduction

The success of state-of-the-art neural networks crucially hinges on the availability of large amounts of annotated training data. While recent advances on language model pre-training (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019) reduce the annotation bottleneck, they still require large amounts of labeled data for obtaining state-of-the-art performances on downstream tasks. However, it is prohibitively expensive to obtain large-scale labeled data for every new task, therefore posing a significant challenge for supervised learning.

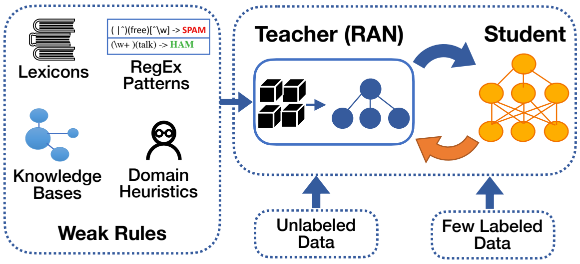


Figure 1: Our weak supervision framework, ASTRA, leverages domain-specific rules, a large amount of (task-specific) unlabeled data, and a small amount of labeled data via iterative self-training.

In order to mitigate labeled data scarcity, recent works have tapped into weak or noisy sources of supervision, such as regular expression patterns (Augenstein et al., 2016), class-indicative keywords (Ren et al., 2018b; Karamanolakis et al., 2019), alignment rules over existing knowledge bases (Mintz et al., 2009; Xu et al., 2013) or heuristic labeling functions (Ratner et al., 2017; Bach et al., 2019; Badene et al., 2019; Awasthi et al., 2020). These different types of sources can be used as weak rules for heuristically annotating large amounts of unlabeled data. For instance, consider the question type classification task from the TREC dataset with regular expression patterns such as: *label all questions containing the token "when" as* `numeric` (e.g., "When was Shakespeare born?"). Approaches relying on such weak rules typically suffer from the following challenges. (i) *Noise.* Rules by their heuristic nature rely on shallow patterns and may predict wrong labels for many instances. For example, the question "When would

[*]Most of the work was done while the first author was an intern at Microsoft Research.

[1]ASTRA: weAkly-supervised Self-TRAining. Our code is publicly available at `https://github.com/microsoft/ASTRA`.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 845–863
June 6–11, 2021. ©2021 Association for Computational Linguistics

such a rule be justified?" refers to circumstances rather than numeric expressions. (ii) *Coverage.* Rules generally have a low coverage as they assign labels to only specific subsets of instances. (iii) *Conflicts.* Different rules may generate conflicting predictions for the same instance, making it challenging to train a robust classifier.

To address the challenges with conflicting and noisy rules, existing approaches learn weights indicating how much to trust individual rules. In the absence of large-scale manual annotations, the rule weights are usually learned via mutual agreement and disagreement of rules over unlabeled data (Ratner et al., 2017; Platanios et al., 2017; Sachan et al., 2018; Bach et al., 2019; Ratner et al., 2019; Awasthi et al., 2020). For instance, such techniques would up-weight rules that agree with each other (as they are more likely to be correct), and down-weight such rules otherwise. An important drawback of these approaches is low coverage since rules assign weak labels to only a subset of the data, thus leading to low rule overlap to compute rule agreement. For instance, in our experiments on six real-world datasets, we observe that 66% of the instances are covered by fewer than 2 rules and 40% of the instances are not covered by any rule at all. Rule sparsity limits the effectiveness of previous approaches, thus leading to strong assumptions, such as, that each rule has the same weight across all instances (Ratner et al., 2017; Bach et al., 2019; Ratner et al., 2019), or that additional supervision is available in the form of labeled "exemplars" used to create such rules in the first place (Awasthi et al., 2020). Most importantly, all these works ignore (as a data pre-processing step) unlabeled instances that are not covered by any of the rules, thus leaving potentially valuable data behind.

Overview of our method. In this work, we present a weak supervision framework, namely ASTRA, that considers all task-specific unlabeled instances and domain-specific rules without strong assumptions about the nature or source of the rules. ASTRA makes effective use of a small amount of labeled data, lots of task-specific unlabeled data, and domain-specific rules through iterative teacher-student co-training (see Figure 1). A student model based on contextualized representations provides pseudo-labels for all instances, thereby, allowing us to leverage all unlabeled data including instances that are not covered by any heuristic rules. To deal with the noisy nature of heuristic rules and pseudo-labels from the student, we develop a rule attention (teacher) network that learns to predict the fidelity of these rules and pseudo-labels conditioned on the context of the instances to which they apply. We develop a semi-supervised learning objective based on minimum entropy regularization to learn all of the above tasks jointly without the requirement of additional rule-exemplar supervision.

Overall, we make the following contributions:

- We propose an iterative self-training mechanism for training deep neural networks with weak supervision by making effective use of task-specific unlabeled data and domain-specific heuristic rules. The self-trained student model predictions augment the weak supervision framework with instances that are not covered by rules.

- We propose a rule attention teacher network (RAN) for combining multiple rules and student model predictions with instance-specific weights conditioned on the corresponding contexts. Furthermore, we construct a semi-supervised learning objective for training RAN without strong assumptions about the structure or nature of the weak rules.

- We demonstrate the effectiveness of our approach on several benchmark datasets for text classification where our method significantly outperforms state-of-the-art weak supervision methods.

2 Self-Training with Weak Supervision

We now present our approach, ASTRA, that leverages a small amount of labeled data, a large amount of unlabeled data, and domain-specific heuristic rules. Our architecture has two main components: the base student model (Section 2.1) and the rule attention teacher network (Section 2.2), which are iteratively co-trained in a self-training framework.

Formally, let $\mathcal{X}$ denote the instance space and $\mathcal{Y} = \{1, \dots, K\}$ denote the label space for a K-class classification task. We consider a small set of manually-labeled examples $D_L = \{(x_l, y_l)\}$, where $x_l \in \mathcal{X}$ and $y_l \in \mathcal{Y}$ and a large set of unlabeled examples $D_U = \{x_i\}$. We also consider a set of pre-defined heuristic rules $R = \{r^j\}$, where each rule r^j has the general form of a labeling function that considers as input an instance $x_i \in \mathcal{X}$ (and potentially additional side information), and

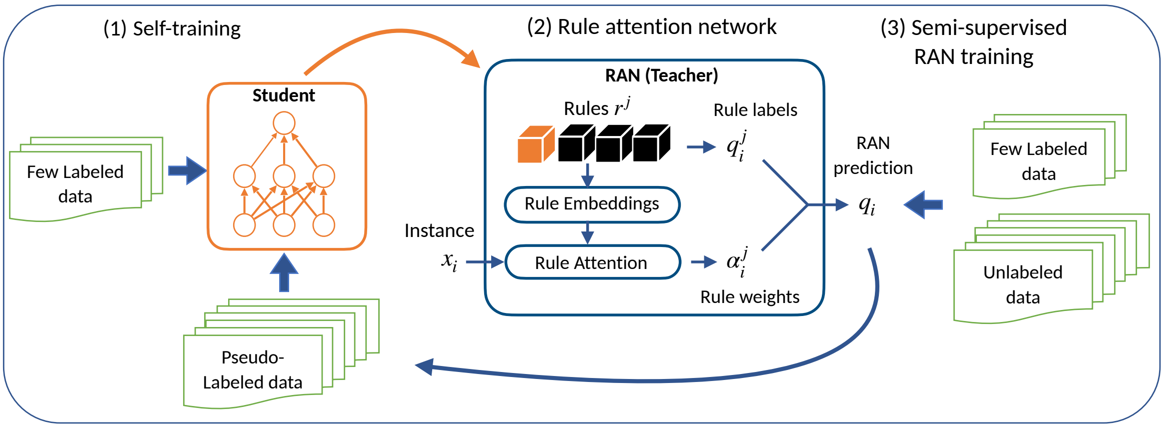


Figure 2: Our ASTRA framework for self-training with weak supervision.

either assigns a *weak* label $q_i^j \in \{0,1\}^K$ (one-hot encoding) or does not apply, i.e., does not assign a label for x_i. Our goal is to leverage D_L, D_U, and R to train a classifier that, given an unseen test instance $x' \in \mathcal{X}$, predicts a label $y' \in \mathcal{Y}$. In the rest of this section, we present our ASTRA framework for addressing this problem.

2.1 Base Student Model

Our self-training framework starts with a base model trained on the available small labeled set D_L. The model is then applied to unlabeled data D_U to obtain pseudo-labeled instances. In classic self-training (Riloff, 1996; Nigam and Ghani, 2000), the student model's pseudo-labeled instances are directly used to augment the training dataset and iteratively re-train the student. In our setting, we augment the self-training process with weak labels drawn from our teacher model that also considers rules in R (described in the next section). The overall self-training process can be formulated as:

$$\min_{\theta} \; \mathbb{E}_{x_l, y_l \in D_L}[-\log \; p_\theta(y_l \mid x_l)] + \lambda \mathbb{E}_{x \in D_U} \mathbb{E}_{y \sim q_{\phi^*}(y|x)}[-\log \; p_\theta(y \mid x)] \quad (1)$$

where, $p_\theta(y|x)$ is the conditional distribution under student's parameters θ; $\lambda \in \mathbb{R}$ is a hyper-parameter controlling the relative importance of the two terms; and $q_{\phi^*}(y \mid x)$ is the conditional distribution under the teacher's parameters ϕ^* from the last iteration that is fixed in the current iteration.

2.2 Rule Attention Teacher Network (RAN)

Our Rule Attention Teacher Network (RAN) aggregates multiple weak sources of supervision with trainable weights and computes a soft weak label q_i for an unlabeled instance x_i. One of the potential drawbacks of relying only on heuristic rules is that a lot of data get left behind. Heuristic rules by nature (e.g., regular expression patterns, keywords) apply to only a subset of the data. Therefore, a substantial number of instances are not covered by any rules and thus are not considered in prior weakly supervised learning approaches (Ratner et al., 2017; Awasthi et al., 2020). To address this challenge and leverage contextual information from all available task-specific unlabeled data, we leverage the corresponding pseudo-labels predicted by the base student model (from Section 2.1). To this end, we apply the student to the unlabeled data $x \in D_U$ and obtain pseudo-label predictions as $p_\theta(y|x)$. These predictions are used to augment the set of already available weak rule labels to increase rule coverage.

Let $R_i \subset R$ be the set of all heuristic rules that *apply* to instance x_i. The objective of RAN is to aggregate the weak labels predicted by all rules $r^j \in R_i$ and the student pseudo-label $p_\theta(y|x_i)$ to compute a soft label q_i for every instance x_i from the unlabeled set D_U. In other words, RAN considers the student as an additional source of weak rule. Aggregating all rule labels into a single label q_i via simple majority voting (i.e., predicting the label assigned by the majority of rules) may not be effective as it treats all rules equally, while in practice, certain rules are more accurate than others.

RAN predicts pseudo-labels q_i by aggregating rules with trainable weights $a_i^{(\cdot)} \in [0,1]$ that capture their fidelity towards an instance x_i as:

$$q_i = \frac{1}{Z_i}\Bigg(\sum_{j:\, r^j \in R_i} a_i^j q_i^j + a_i^S p_\theta(y|x_i) + a_i^u u \Bigg), \quad (2)$$

where a_i^j and a_i^S are the fidelity weights for the heuristic rule labels q_i^j and the student assigned pseudo-label $p_\theta(y|x_i)$ for an instance x_i, respectively; u is a uniform rule distribution that assigns equal probabilities for all the K classes as $u = [\frac{1}{K}, \ldots, \frac{1}{K}]$; a_i^u is the weight assigned to the "uniform rule" for x_i, which is computed as a function of the rest of the rule weights: $a_i^u = (|R_i| + 1 - \sum_{j:\, r^j \in R_i} a_i^j - a_i^S)$; and Z_i is a normalization coefficient to ensure that q_i is a valid probability distribution. u acts as a uniform smoothing factor that prevents overfitting for sparse settings, for instance, when a single weak rule applies to an instance.

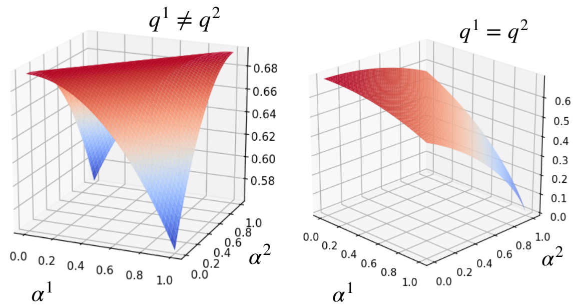
According to Eq. (2), a rule r^j with higher fidelity weight a_i^j contributes more to the computation of q_i. If $a_i^j = 1\ \forall r^j \in \{R_i \cup p_\theta\}$, then RAN reduces to majority voting. If $a_i^j = 0\ \forall r^j \in \{R_i \cup p_\theta\}$, then RAN ignores all rules and predicts $q_i = u$. Note the distinction of our setting to recent works like Snorkel (Ratner et al., 2017), that learns global rule-weights $a_i^j = a^j\ \forall x_i$ by ignoring the instance-specific rule fidelity. Our proposed approach is flexible but at the same time challenging as we do not assume prior knowledge of the internal structure of the labeling functions $r^j \in R$.

In order to effectively compute rule fidelities, RAN considers instance embeddings that capture the context of instances beyond the shallow patterns considered by rules. In particular, we model the weight a_i^j of rule r^j as a function of the context of the instance x_i and r^j through an attention-based mechanism. Consider $h_i \in \mathbb{R}^{d'}$ to be the hidden state representation of x_i from the base student model. Also, consider the (trainable) embedding of each rule r^j as $e_j = g(r^j) \in \mathbb{R}^d$. We use e_j as a query vector with *sigmoid attention* to compute instance-specific rule attention weights as:

$$a_i^j = \sigma(f(h_i)^T \cdot e_j) \in [0, 1], \tag{3}$$

where f is a multi-layer perceptron that projects h_i to $\mathbb{R}^d$ and $\sigma(\cdot)$ is the sigmoid function. Rule embedding allows us to exploit the similarity between different rules in terms of instances to which they apply, and further leverage their semantics for modeling agreement. RAN computes the student's weight a_i^S using the same procedure as for computing the rule weights a_i^j.

Note that the rule predictions q_i^j are considered fixed, while we estimate their attention weights. The above coupling between rules and instances via their corresponding embeddings e_j and h_i allows us to obtain representations where similar rules apply to similar contexts, and model their agreements via the attention weights a_i^j. To this end, the trainable parameters of RAN (f and g) are shared across all rules and instances. Next, we describe how to train RAN.



(a) Rule predictions disagree. (b) Rule predictions agree.

Figure 3: Variation in unsupervised entropy loss with instance-specific rule predictions and attention weights encouraging rule agreement. Consider this illustration with two rules for a given instance. When rule predictions disagree ($q^1 \neq q^2$), minimum loss is achieved for attention weights a^1=0, a^2=1 or a^1=1, a^2=0. When rule predictions agree (q^1=q^2), minimum loss is achieved for attention weights a^1=a^2=1. For instances covered by three rules, if q^1=q^2≠q^3, the minimum loss is achieved for a^1=a^2=1 and a^3=0.

2.3 Semi-Supervised Learning of ASTRA

Learning to predict instance-specific weights $a_i^{(\cdot)}$ for the weak sources (including rules and student pseudo-labels) is challenging due to the absence of any explicit knowledge about the source quality and limited amount of labeled training data. We thus treat the weights $a_i^{(\cdot)}$ as latent variables and propose a semi-supervised objective for training RAN with supervision on the coarser level of q_i:

$$\mathcal{L}^{RAN} = -\sum_{(x_i,y_i)\in D_L} y_i \log q_i - \sum_{x_i \in D_U} q_i \log q_i. \tag{4}$$

Given task-specific labeled data D_L, the first term in Eq. (4) minimizes the cross-entropy loss between the teacher's label q_i and the corresponding clean label y_i for the instance x_i. This term penalizes weak sources that assign labels $q_i^{(\cdot)}$ that contradict with the ground-truth label y_i by assigning a low instance-specific fidelity weight $a_i^{(\cdot)}$.

The second term in Eq. (4) minimizes the entropy of the aggregated pseudo-label q_i on unlabeled data D_U. Minimum entropy regularization is effective in settings with small amounts of labeled

	TREC	SMS	YouTube	CENSUS	MIT-R	Spouse
Labeled Training Data ($\|D_L\|$)	68	69	100	83	1842	100
Unlabeled Training Data ($\|D_U\|$)	5K	5K	2K	10K	65K	22K
Test Data	500	500	250	16K	14K	3K
#Classes	6	2	2	2	9	2
#Rules	68	73	10	83	15	9
Rule Accuracy (Majority Voting)	60.9%	48.4%	82.2%	80.1%	40.9%	44.2%
Rule Coverage (instances in D_U covered by $\geq$ 1 rule)	95%	40%	87%	100%	14%	25%
Rule Overlap (instances in D_U covered by $\geq$ 2 rules)	46%	9%	48%	94%	1%	8%

Table 1: Dataset statistics.

Algorithm 1 Self-training with Weak Supervision

Input: Small amount of labeled data D_L; task-specific unlabeled data D_U; weak rules R

Outputs: Student $p^*_\theta(\cdot)$, RAN Teacher $q^*_\phi(\cdot)$

1: Train student $p_\theta(\cdot)$ using D_L

2: **Repeat until convergence:**

2.1: Train teacher $q_\phi(\cdot)$ using D_L, D_U through Eq. (2) and (4)

2.2: Apply $q_\phi(y \mid x, R, p_\theta)$ to $x \in D_U$ to obtain pseudo-labeled data: $D_{RAN} = \{(x_i, q_i)\}_{x_i \in D_U}$ through Eq. (2)

2.3: Train $p_\theta(\cdot)$ using D_L, D_{RAN} through Eq. (1)

data by leveraging unlabeled data (Grandvalet and Bengio, 2005), and is highly beneficial in our setting because it encourages RAN to predict weights that maximize rule agreement. Since the teacher label q_i is obtained by aggregating weak labels $q_i^{(\cdot)}$, entropy minimization encourages RAN to predict higher instance-specific weights $a_i^{(\cdot)}$ to sources that agree in their labels for x_i, and lower weights when there are disagreements between weak sources – aggregated across all the unlabeled instances.

Figure 3 plots the minimum entropy loss over unlabeled data over two scenarios where two rules agree or disagree with each other for a given instance. The optimal instance-specific fidelity weights $a_i^{(\cdot)}$ are 1 when rules agree with each other, thereby, assigning credits to both rules, and only one of them when they disagree. We use this unsupervised entropy loss in conjunction with cross-entropy loss over labeled data to ensure grounding.

End-to-end Learning: Algorithm 1 presents an overview of our learning mechanism. We first use the small amount of labeled data to train a base student model that generates pseudo-labels and augments heuristic rules over unlabeled data. Our RAN network computes fidelity weights to combine these different weak labels via minimum entropy regularization to obtain an aggregated pseudo-label for every unlabeled instance. This is used to re-train the student model with the above student-teacher training repeated till convergence.

3 Experiments

Datasets. We evaluate our framework on the following six benchmark datasets for weak supervision from Ratner et al. (2017) and Awasthi et al. (2020). (1) Question classification from TREC-6 into 6 categories (Abbreviation, Entity, Description, Human, Location, Numeric-value); (2) Spam classification of SMS messages; (3) Spam classification of YouTube comments; (4) Income classification on the CENSUS dataset on whether a person earns more than \$50K or not; (5) Slot-filling in sentences on restaurant search queries in the MIT-R dataset: each token is classified into 9 classes (Location, Hours, Amenity, Price, Cuisine, Dish, Restaurant Name, Rating, Other); (6) Relation classification in the Spouse dataset, whether pairs of people mentioned in a sentence are/were married or not.

Table 1 shows the dataset statistics along with the amount of labeled, unlabeled data and domain-specific rules for each dataset. For a fair comparison, we use exactly the same set of rules as in the previous work for the benchmark datasets. These rules include regular expression patterns, lexicons, and knowledge bases for weak supervision. Most of these rules were constructed manually, except for the CENSUS dataset, where rules have been automatically extracted with a coverage of 100%.

On average across all the datasets, 66% of the instances are covered by fewer than 2 rules, whereas 40% are not covered by any rule at all – demonstrating the sparsity in our setting. We also report the accuracy of the rules in terms of majority voting

Method	Learning to Weight Rules	Instances	Unlabeled (no rules)
Majority	-	-	-
Snorkel (Ratner et al., 2017)	✓	-	-
PosteriorReg (Hu et al., 2016)	✓	-	-
L2R (Ren et al., 2018a)	-	✓	-
ImplyLoss (Awasthi et al., 2020)	✓	✓	-
Self-train	-	-	✓
ASTRA	✓	✓	✓

Table 2: ASTRA learns rule-specific and instance-specific attention weights and leverages task-specific unlabeled data where no rules apply.

on the task-specific unlabeled datasets. Additional details on the dataset and examples of rules are presented in the Appendix.

Evaluation. We train ASTRA five times for five different random splits of the labeled training data and evaluate on held-out test data. We report the average performance as well as the standard deviation across multiple runs. We report the same evaluation metrics as used in prior works (Ratner et al., 2017; Awasthi et al., 2020) for a fair comparison.

Model configuration. Our student model consists of embeddings from pre-trained language models like ELMO (Peters et al., 2018) or BERT (Devlin et al., 2019) for generating contextualized representations for an instance, followed by a softmax classification layer. The RAN teacher model considers a rule embedding layer and a multilayer perceptron for mapping the contextualized representation for an instance to the rule embedding space. Refer to the Appendix for more details.

Baselines. We compare our method with the following methods: *(a) Majority* predicts the majority vote of the rules with ties resolved by predicting a random class. *(b) LabeledOnly* trains classifiers using only labeled data (fully supervised baseline). *(c) Self-train* (Nigam and Ghani, 2000; Lee, 2013) leverages both labeled and unlabeled data for iterative self-training on pseudo-labeled predictions over task-specific unlabeled data. This baseline ignores domain-specific rules. *(e) Snorkel+Labeled* (Ratner et al., 2017) trains classifiers using weakly-labeled data with a generative model. The model is trained on unlabeled data for computing rule weights in an unsupervised fashion, and learns a single weight per rule across all instances. It is further fine-tuned on labeled data. *(f) L2R* (Ren et al., 2018b) learns to re-weight noisy or weak labels from domain-specific rules via meta-learning. It learns instance-specific but not rule-specific weights. *(g) PosteriorReg* (Hu et al., 2016) trains classifiers using rules as soft constraints via posterior regularization (Ganchev et al., 2010). *(h) ImplyLoss* (Awasthi et al., 2020) leverages *exemplar*-based supervision as additional knowledge for learning instance-specific and rule-specific weights by minimizing an implication loss over unlabeled data. This requires maintaining a record of all instances used to create the weak rules in the first place. Table 2 shows a summary of the different methods contrasting them on how they learn the weights (rule-specific or instance-specific) and if they leverage task-specific unlabeled data that are not covered by any rules.

3.1 Experimental Results

Overall results. Table 3 summarizes the main results across all datasets. Among all the semi-supervised methods that leverage weak supervision from domain-specific rules, ASTRA outperforms Snorkel by 6.1% in average accuracy across all datasets by learning instance-specific rule weights in conjunction with self-training over unlabeled instances where weak rules do not apply. Similarly, ASTRA also improves over a recent work and the best performing baseline ImplyLoss by 3.1% on average. Notably, our method does not require additional supervision at the level of exemplars used to create rules in contrast to ImplyLoss.

Self-training over unlabeled data. Recent works for tasks like image classification (Li et al., 2019; Xie et al., 2020; Zoph et al., 2020), neural sequence generation (Zhang and Zong, 2016; He et al., 2019) and few-shot text classification (Mukherjee and Awadallah, 2020; Wang et al., 2020) show the effectiveness of self-training methods in exploiting task-specific unlabeled data with stochastic regularization techniques like dropouts and data augmentation. We also make similar observations for our weakly supervised tasks, where classic self-train methods ("Self-train") leveraging only a few task-specific labeled examples and lots of unlabeled data outperform weakly supervised methods like Snorkel and PosteriorReg that have additional access to domain-specific rules.

Self-training with weak supervision. Our framework ASTRA provides an efficient method to incorporate weak supervision from domain-specific rules to augment the self-training framework and improves by 6% over classic self-training.

To better understand the benefits of our approach

	TREC (Acc)	SMS (F1)	YouTube (Acc)	CENSUS (Acc)	MIT-R (F1)	Spouse (F1)
Majority	60.9 (0.7)	48.4 (1.2)	82.2 (0.9)	80.1 (0.1)	40.9 (0.1)	44.2 (0.6)
LabeledOnly	66.5 (3.7)	93.3 (2.9)	91.0 (0.7)	75.8 (1.7)	74.7 (1.1)	47.9 (0.9)
Snorkel+Labeled	65.3 (4.1)	94.7 (1.2)	93.5 (0.2)	79.1 (1.3)	75.6 (1.3)	49.2 (0.6)
PosteriorReg	67.3 (2.9)	94.1 (2.1)	86.4 (3.4)	79.4 (1.5)	74.7 (1.2)	49.4 (1.1)
L2R	71.7 (1.3)	93.4 (1.1)	92.6 (0.5)	82.4 (0.1)	58.6 (0.4)	49.5 (0.7)
ImplyLoss	75.5 (4.5)	92.2 (2.1)	93.6 (0.5)	80.5 (0.9)	75.7 (1.5)	49.8 (1.7)
Self-train	71.1 (3.9)	95.1 (0.8)	92.5 (3.0)	78.6 (1.0)	72.3 (0.6)	51.4 (0.4)
ASTRA (ours)	**80.3** (2.4)	**95.3** (0.5)	**95.3** (0.8)	**83.1** (0.4)	**76.9** (0.6)	**62.3** (1.1)

Table 3: Overall result comparison across multiple datasets. Results are aggregated over five runs with random training splits and standard deviation across the runs in parentheses.

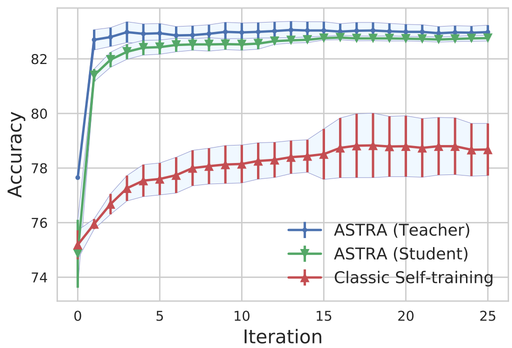


Figure 4: Gradual accuracy improvement over self-training iterations in the CENSUS dataset. ASTRA (Student) performs better than Classic Self-training (Student) being guided by a better teacher.

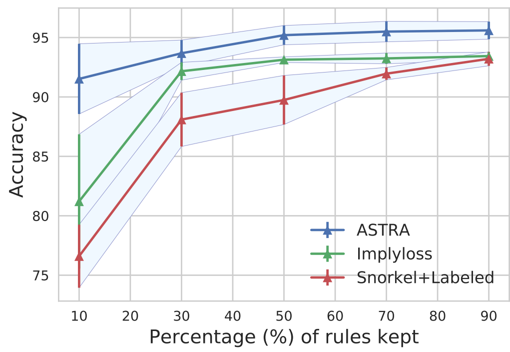


Figure 5: Performance improvement on increasing the proportion of weak rules in YouTube. For each setting, we randomly sample a subset of rules, aggregate and report results across multiple runs. ASTRA is effective across all settings with strongest improvements under high rule sparsity (left region of the x-axis).

compared to classic self-training, consider Figure 4, which depicts the gradual performance improvement over iterations. The student models in classic self-training and ASTRA have exactly the same architecture. However, the latter is guided by a better teacher (RAN) that learns to aggregate noisy rules and pseudo-labels over unlabeled data.

Impact of rule sparsity and coverage for weak supervision. In this experiment, we compare the performance of various methods by varying the proportion of available domain-specific rules. To this end, we randomly choose a subset of the rules (varying the proportion from 10% to 100%) and train various weak supervision methods. For each setting, we repeat experiments with multiple rule splits and report aggregated results in Figure 5. We observe that ASTRA is effective across all settings with the most impact at high levels of rule sparsity. For instance, with 10% of domain-specific rules available, ASTRA outperforms ImplyLoss by 12% and Snorkel+Labeled by 19%.

This performance improvement is made possible by incorporating self-training in our framework to obtain pseudo-labels for task-specific unlabeled instances, and further re-weighting them with other domain-specific rules via the rule attention network. Correspondingly, Table 4 shows the increase in data coverage for every task given by the proportion of unlabeled instances that are now covered by at least two weak sources (from multiple rules and pseudo-labels) in contrast to just considering the rules.

3.2 Ablation Study

Table 5 reports ablation experiments to evaluate the impact of various components in ASTRA.

ASTRA teacher marginally outperforms the student model on an aggregate having access to domain-specific rules. ASTRA student that is self-trained over task-specific unlabeled data and guided by an efficient teacher model significantly outper-

% Overlap	TREC	YTube	SMS	MITR	CEN.	Spouse
Only Rules	46	48	9	1	94	8
ASTRA	**95**	**87**	**40**	**14**	**100**	**25**
Increase	+49	+39	+31	+13	+6	+17

Table 4: ASTRA substantially increases overlap (%) determined by the proportion of unlabeled instances that are covered by at least 2 weak sources (from multiple rules and student pseudo-labels, as applicable).

Configuration	Acc
ASTRA (Teacher)	**88.1**
ASTRA (Student)	87.7 (↓ 0.4%)
No min. entropy regularization in Eq. (4)	86.9 (↓ 1.4%)
No student fine-tuning on D_L (step 2.3)	86.7 (↓ 1.6%)
No student pseudo-labels in RAN in Eq. (2)	85.3 (↓ 3.2%)

Table 5: Summary of ablation experiments aggregated across multiple datasets. Refer to Appendix for corresponding results in each dataset.

forms other state-of-the-art baselines.

Through minimum entropy regularization in our semi-supervised learning objective (Eq. (4)), ASTRA leverages the agreement between various weak sources (including rules and pseudo-labels) over task-specific unlabeled data. Removing this component results in an accuracy drop of 1.4% on an aggregate demonstrating its usefulness.

Fine-tuning the student on labeled data is important for effective self-training: ignoring D_L in the step 2.3 in Algorithm 1, leads to 1.6% lower accuracy than ASTRA.

There is significant performance drop on removing the student's pseudo-labels ($p_\theta(\cdot)$) from the rule attention network in Eq. (2). This significantly limits the coverage of the teacher ignoring unlabeled instances where weak rules do not apply, thereby, degrading the overall performance by 3.2%.

3.3 Case Study: TREC-6 Dataset

Table 6 shows a question in the TREC-6 dataset that was correctly classified by the ASTRA teacher as an "Entity" type (ENTY). Note that the majority voting of the four weak rules that apply to this instance (Rule 8, 24, 42, and 61) leads to an incorrect prediction of "Human" (HUM) type. The ASTRA teacher aggregates all the heuristic rule labels and the student pseudo-label with their (computed) fidelity weights for the correct prediction.

Refer to Table 7 for more illustrative examples on how ASTRA aggregates various weak supervision sources with corresponding attention weights shown in parantheses. In Example 1 where no rules apply, the student leverages the context of the sentence (e.g., semantics of "president") to predict the `HUM` label. While in Example 2, the teacher downweights the incorrect student (as well as conflicting rules) and upweights the appropriate rule to predict the correct `ENTY` label. In example 3, ASTRA predicts the correct label `ENTY` relying only on the student as both rules report noisy labels.

4 Related Work

In this section, we discuss related work on self-training and learning with noisy labels or rules. Refer to Hedderich et al. (2021) for a thorough survey of approaches addressing low-resource scenarios.

Self-Training. Self-training (Yarowsky, 1995; Nigam and Ghani, 2000; Lee, 2013) as one of the earliest semi-supervised learning approaches (Chapelle et al., 2009) trains a base model (student) on a small amount of labeled data; applies it to pseudo-label (task-specific) unlabeled data; uses pseudo-labels to augment the labeled data; and re-trains the student in an iterative manner. Self-training has recently been shown to obtain state-of-the-art performance for tasks like image classification (Li et al., 2019; Xie et al., 2020; Zoph et al., 2020), few-shot text classification (Mukherjee and Awadallah, 2020; Wang et al., 2020), and neural machine translation (Zhang and Zong, 2016; He et al., 2019) and has shown complementary advantages to unsupervised pre-training (Zoph et al., 2020). A typical issue in self-training is error propagation from noisy pseudo-labels. This is addressed in ASTRA via rule attention network that computes the fidelity of pseudo-labels instead of directly using them to re-train the student.

Learning with Noisy Labels. Classification under label noise from a single source has been an active research topic (Frénay and Verleysen, 2013). A major line of research focuses on correcting noisy labels by learning label corruption matrices (Patrini et al., 2017; Hendrycks et al., 2018; Zheng et al., 2021). More related to our work are the instance reweighting approaches (Ren et al., 2018b; Shu et al., 2019), which learn to up-weight and down-weight instances with cleaner and noisy labels respectively. However, these operate only at instance-level and do not consider rule-specific importance. Our approach learns both instance- and rule-specific fidelity weights and substantially outperforms Ren

<table>
<tr><td>Text</td><td colspan="3">What was President Lyndon Johnson ’s reform program called ?</td></tr>
<tr><td>Clean Label</td><td colspan="3">ENTY</td></tr>
<tr><td>ASTRA Teacher</td><td colspan="3">ENTY</td></tr>
<tr><th>Weak Source</th><th>Label</th><th>Weight</th><th>Feature / Regular expression pattern</th></tr>
<tr><td>Student</td><td>ENTY</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 8</td><td>HUM</td><td>a=1.0</td><td><code>(|^)(who|what|what)[^\w] *(\w+){0,1}(person|man|woman|human|president|president)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 24</td><td>ENTY</td><td>a=1.0</td><td><code>(|^)(what|what)[^\w]*(\w+){0,1}(is|is)[^\w]* *([^\s]+)*(surname|address|name|name)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 61</td><td>HUM</td><td>a=0.0</td><td><code>(|^)(called|alias|nicknamed|nicknamed)[^\w]*(|$)</code></td></tr>
</table>

Table 6: Snapshot of a question in TREC-6 and corresponding predictions. Top: instance text, clean label, and the aggregated prediction from ASTRA teacher. Bottom: several weak rules with regular expression patterns and predicted weak labels, along with the student and its pseudo-label (DESC: description, ENTY: entity, NUM: number, HUM: human). The weights depict the fidelity computed by RAN for each weak source for this specific instance.

Instance Text (Question in TREC-6)	Teacher	Student	Set of Heuristic Rule Labels
1. *Which president was unmarried?*	**HUM**	HUM(1)	{ }
2. *What is a baby turkey called?*	**ENTY**	DESC(1)	{ENTY(1), DESC(0), HUM(0)}
3. *What currency do they use in Brazil?*	**ENTY**	ENTY(1)	{DESC(0), DESC(0)}
4. *What is the percentage of water content in the human body?*	**NUM**	DESC(0)	{HUM(0), NUM(0.2), DESC(0)}

Table 7: Snapshot of answer-type predictions for questions in TREC-6 from ASTRA teacher and student along with a set of labels assigned by various weak rules (DESC: description, ENTY: entity, NUM: number, HUM: human) with corresponding attention weights (in parentheses). Correct and incorrect predictions are colored in green and red respectively. Detailed analysis and rule semantics are reported in the Appendix.

et al. (2018b) across all datasets.

Learning with Multiple Rules. To address the challenges with multiple noisy rules, existing approaches learn rule weights based on mutual rule agreements with some strong assumptions. For instance, Meng et al. (2018); Karamanolakis et al. (2019); Mekala and Shang (2020) denoise seed words using vector representations of their semantics. However it is difficult to generalize these approaches from seed words to more general labeling functions that only predict heuristic labels (as in our datasets). Ratner et al. (2017); Sachan et al. (2018); Ratner et al. (2019) assume each rule to be equally accurate across all the instances that it covers. Awasthi et al. (2020) learn rule-specific and instance-specific weights but assume access to *labeled exemplars* that were used to create the rule in the first place. Most importantly, all these works ignore unlabeled instances that are not covered by any of the rules, while our approach leverages all unlabeled instances via self-training.

5 Conclusions and Future Work

We developed a weak supervision framework, ASTRA, that efficiently trains classifiers by integrating task-specific unlabeled data, few labeled data, and domain-specific knowledge expressed as rules. Our framework improves data coverage by employing self-training with a student model. This considers contextualized representations of instances and predicts pseudo-labels for all instances, including those that are not covered by heuristic rules. Additionally, we developed a rule attention network, RAN, to aggregate various weak sources of supervision (heuristic rules and student pseudo-labels) with instance-specific weights, and employed a semi-supervised objective for training RAN without strong assumptions about the nature or structure of the weak sources. Extensive experiments on several benchmark datasets demonstrate our effectiveness, particularly at high levels of rule sparsity. In future work, we plan to extend our framework to support a broader range of natural language understanding tasks and explore alternative techniques for rule embedding.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback, and Wei Wang and Benjamin Van Durme for insightful discussions.

Ethical Considerations

In this work, we introduce a framework for training of neural network models with few labeled examples and domain-specific knowledge. This work is likely to increase the progress of NLP applications for domains with limited annotated resources but access to domain-specific knowledge. While it is not only expensive to acquire large amounts of labeled data for every task and language, in many cases, we cannot perform large-scale labeling due to access constraints from privacy and compliance concerns. To this end, our framework can be used for applications in finance, legal, healthcare, retail and other domains where adoption of deep neural network may have been hindered due to lack of large-scale manual annotations on sensitive data.

While our framework accelerates the progress of NLP, it also suffers from associated societal implications of automation ranging from job losses for workers who provide annotations as a service. Additionally, it involves deep neural models that are compute intensive and has a negative impact on the environment in terms of carbon footprint. The latter concern is partly alleviated in our work by leveraging pre-trained language models and not training from scratch, thereby, leading to efficient and faster compute.

References

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885.

Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. 2020. Learning from rules generalizing labeled exemplars. In *International Conference on Learning Representations*.

Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. 2019. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 International Conference on Management of Data*.

Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019. Data programming for learning discourse structure. In *Association for Computational Linguistics (ACL)*.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of NAACL-HLT*.

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*, pages 10477–10486.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. 2019. Leveraging just a few keywords for fine-grained aspect detection through weakly supervised co-training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

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

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*.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

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

Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*.

Emmanouil Platanios, Hoifung Poon, Tom M Mitchell, and Eric J Horvitz. 2017. Estimating accuracy from unlabeled data: A probabilistic logic approach. *Advances in Neural Information Processing Systems*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hongyu Ren, Russell Stewart, Jiaming Song, Volodymyr Kuleshov, and Stefano Ermon. 2018a. Learning with weak supervision from physics and data-driven constraints. *AI Magazine*, 39(1):27–38.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018b. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*.

Mrinmaya Sachan, Kumar Avinava Dubey, Tom M Mitchell, Dan Roth, and Eric P Xing. 2018. Learning pipelines with limited data and domain knowledge: A study in parsing physics problems. In *Advances in Neural Information Processing Systems*.

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*.

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

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*, pages 10687–10698.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. 2021. Meta label correction for noisy label learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

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*.

A Appendix

For reproducibility, we provide details of our implementation (Section A.1), datasets (Section A.2), and experimental results (Section A.3). Our code is available at https://github.com/microsoft/ASTRA.

A.1 Implementation Details

We now describe implementation details for each component in ASTRA: our base student model and our rule attention teacher network. Table 8 shows our hyperparameter search configuration. We choose optimal hyperparameters by manual tuning based on the development performances. Table 9 shows the hyperparameters and model architecture details for each dataset. For a fair comparison, we use the same architectures as previous approaches but we expect further improvements by exploring different architectures.

Base Student Model Our student model consists of an instance embedding layer (e.g., ELMO (Peters et al., 2018), BERT (Devlin et al., 2019), logistic regression), a multilayer perceptron with two hidden layers, and a softmax classification layer for predicting labels.

Rule Attention Teacher Network Our RAN teacher model consists of a 128-dimensional rule embedding layer, a multilayer perceptron for mapping the contextualized representation for an instance to the rule embedding space, and a sigmoid attention layer.

Iterative Teacher-Student Training At each iteration, we train the RAN teacher on unlabeled data and fine-tune on clean labeled data. Also at each iteration, we train the student on pseudo-labeled teacher data and fine-tune on clean labeled data. We consider a maximum number of 25 self-training iterations (with early stopping of patience 3 epochs) and keep the models' performances for the iteration corresponding to the highest validation performance.

A.2 Dataset Details

We evaluate our framework on the following six benchmark datasets for weak supervision from Ratner et al. (2017) and Awasthi et al. (2020)[2]. All datasets are in English. Table 11 shows detailed dataset statistics. We consider the same test sets with previous work. For a robust evaluation of our model's performance, we split each dataset into five random train/validation/unlabeled splits and report the average performance and standard deviation across runs. For a fair comparison, we use the same splits and evaluation procedure across all methods and baselines.

TREC: Question classification from TREC-6 into 6 categories: Abbreviation, Entity, Description, Human, Location, Numeric-value. Table 12 reports a sample of regular expression rules out of the 68 rules used in the TREC dataset. TREC has 13 keyword-based (coverage=62%) and 55 regular expression-based (coverage=57%) rules.

SMS: Binary Spam vs. Not Spam classification of SMS messages. SMS has 16 keyword-based (coverage=4%) and 57 regular expression-based (coverage=38%) rules.

YouTube: Binary Spam vs. Not Spam classification of YouTube comments.[3] YouTube has 5 keyword-based (coverage=48%), 1 regular expression-based (coverage=23%), 1 length-based (coverage=23%), and 3 classifier-based (coverage=46%) rules.

CENSUS: Binary income classification on the UCI CENSUS dataset on whether a person earns more than $50K or not. This is a non-textual dataset and is considered to evaluate the performance of our approach under the low sparsity setting, since the 83 rules are automatically extracted and have a coverage of 100%.

MIT-R: Slot-filling in sentences on restaurant search queries in the MIT-R dataset: each token is classified into 9 classes (Location, Hours, Amenity, Price, Cuisine, Dish, Restaurant Name, Rating, Other). MIT-R has 5 keyword-based (coverage=6%) and 10 regular expression-based (coverage=10%) rules.

Spouse: Relation classification in the Spouses dataset[4], whether pairs of people mentioned in a sentence are/were married or not. Spouse has 6 keyword-based (coverage=23%), 1 heuristic-based

[2] https://github.com/awasthiabhijeet/Learning-From-Rules

[3] https://archive.ics.uci.edu/ml/machine-learning-databases/00380/YouTube-Spam-Collection-v1.zip

[4] https://www.dropbox.com/s/jmrvyaqew4zp9cy/spouse_data.zip

(coverage=4%), and 2 distant supervision-based (coverage=0.2%) rules.

A.3 Experimental Result Details

We now discuss detailed results on each dataset. To be consistent with previous work, we report accuracy scores for the TREC, Youtube, and CENSUS dataset and macro-average F1 scores for the SMS, Spouse, and MIT-R datasets.

A.3.1 Ablation Studies

Table 10 reports detailed ablation results per dataset. The right column computes the average accuracy across datasets.

A.3.2 Case Study: TREC-6 Dataset

Table 12 shows a sample of rules from the TREC-6 dataset. Those rules capture regular expression patterns to predict one of the 6 question categories for a question. Tables 13-26 show examples of individual instances in TREC-6, the corresponding rule predictions, the student pseudo-labels, as well as our RAN that aggregates rule and student predictions with attention weights a to compute a single (Teacher) label.

Hyperparameter	Values
Learning rate	1e-1, 1e-2, 1e-3, 1e-4, 1e-5
Fine-tuning rate	1e-3, 1e-4, 1e-5, 1e-6, 1e-7
Type of pseudo-labels	soft, hard

Table 8: Hyperparameter search.

	TREC	SMS	Youtube	CENSUS	MIT-R	Spouse
Instance vector type	ELMO	ELMO	LogReg	Categorical	ELMO	BERT
Instance vector dimensionality	1024	1024	16,634	105	1024	768
Learning rate	1e-4	1e-4	1e-4	1e-3	1e-4	1e-4
Fine-tuning rate	1e-5	1e-5	1e-5	1e-5	1e-4	1e-5
Type of pseudo-labels	soft	soft	hard	soft	soft	soft
Pseudo-training epochs (patience: 5)	25	25	25	25	25	25
Fine-tuning epochs (patience: 5)	70	70	70	70	70	70
Self-training iterations (patience: 3)	25	25	25	25	25	25
Training batch size	16	16	16	128	256	16
Unsupervised batch size	256	256	256	256	256	256
Rule embedding dimensionality	128	128	128	128	128	128

Table 9: Selected hyperparameters.

	TREC (Acc)	SMS (F1)	Youtube (Acc)	CENSUS (Acc)	MIT-R (F1)	AVG (Acc)
ASTRA (Teacher)	**80.3** (2.4)	95.3 (0.5)	95.3 (0.8)	**83.1** (0.4)	**76.9** (0.6)	**88.1**
ASTRA (Student)	79.2 (2.1)	**95.7** (0.5)	**95.5** (0.5)	82.8 (0.1)	76.6 (0.9)	87.7
Hard Student Pseudo-labels in RAN in Eq. (2)	77.6 (1.2)	94.5 (0.7)	95.3 (0.8)	83.0 (0.7)	75.9 (0.9)	87.0
No minimum entropy regularization in Eq. (4)	75.8 (2.5)	**95.7** (0.7)	95.1 (0.8)	83.0 (0.3)	72.6 (0.4)	86.9
No cross-entropy loss in Eq. (4)	74.1 (2.3)	93.9 (0.4)	**95.5** (0.6)	**83.1** (0.6)	71.9 (0.4)	86.7
No student pseudo-labels in RAN in Eq. (1)	75.3 (4.3)	95.8 (0.2)	91.4 (2.2)	**83.1** (0.4)	71.9 (1.0)	85.3

Table 10: Ablation studies.

	TREC	SMS	Youtube	CENSUS	MIT-R	Spouse
$\|D_L\|$	68	69	100	83	1842	100
$\|D_U\|$	4884	1586	4502	64,888	10,000	22,254
Validation	500	500	150	5561	16281	2711
Test Size	500	500	250	16281	14256	2701
#Classes	6	2	2	2	9	2
#Rules	68	73	10	83	15	9
Rule Precision (Majority Voting)	63.7%	97.3%	78.6%	80.7%	84.1%	66.6%
Rule Accuracy (Majority Voting)	60.9%	48.4%	82.2%	80.1%	40.9%	44.2%
Rule Coverage (instances in D_U covered by $\geq$ 1 rule)	95%	40%	87%	100%	14%	25%
Rule Overlap (instances in D_U covered by $\geq$ 2 rules)	46%	9%	48%	94%	1%	8%

Table 11: Dataset statistics.

<table>
<tr><th>Rule</th><th>Label</th><th>Pattern</th></tr>
<tr><td>Rule 5</td><td>HUM</td><td><code>(|^)(which|who|what|what)[^\w]*([^\s]+)*(person|
man|woman|human|poet|poet)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 8</td><td>HUM</td><td><code>(|^)(who|what|what)[^\w] *(\w+){0,1}(person|
man|woman|human|president|president)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 24</td><td>ENTY</td><td><code>(|^)(what|what)[^\w]*(\w+){0,1}(is|is)[^\w]*
([^\s]+)(surname|address|name|name)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 29</td><td>NUM</td><td><code>(|^)(which|what|what)[^\w]*
([^\s]+)(time|day|month|hours|minute
|seconds|year|date|date)[^\w](|$)</code></td></tr>
<tr><td>Rule 32</td><td>NUM</td><td><code>(|^)(year|year)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 41</td><td>NUM</td><td><code>(|^)(what|what)[^\w]* ([^\s]+)*(percentage
|share|number|population|population)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 42</td><td>DESC</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 54</td><td>DESC</td><td><code>(|^)(how|what|what)[^\w]*
* (\w+){0,1}(do|does|does)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 61</td><td>HUM</td><td><code>(|^)(called|alias|nicknamed|nicknamed)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 68</td><td>ABBR</td><td><code>(|^)(what|what)[^\w]* (\w+){0,1}(does|does)[^\w]*
* * ([^\s]+)*(stand for)[^\w]*(|$)</code></td></tr>
</table>

Table 12: Sample of REGEX rules from the TREC-6 dataset capturing the various question categories (HUM: Human, ENTY: Entity, NUM: Numeric Value, DESC: Description, ABBR: Abbreviation)

Text	Why is a ladybug helpful ?		
Clean label	DESC		
RAN Teacher	**DESC**		
Weak Source	**Label**	**Weight**	**Feature**
Student	**DESC**	a=1.0	h_i (contextualized instance embedding)

Table 13: TREC example. No rules apply. The student generalizes beyond rules by considering contextualized instance embeddings and assigns the instance to the DESCRIPTION class. Our RAN teacher assigns an attention weight of 1 to the student and predicts the right class.

<table>
<tr><td>Text</td><td colspan="3">Which president was unmarried ?</td></tr>
<tr><td>Clean label</td><td colspan="3">HUM</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">NUM</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>HUM</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
</table>

Table 14: TREC example: While no rules apply to this instance, the student leverages the semantics of the sentence (i.e., that “president” corresponds to a person) to predict the HUMAN (HUM) class. Thus, our RAN predicts the right label instead of discarding this instance.

<table>
<tr><td>Text</td><td colspan="3">What is a baby turkey called ?</td></tr>
<tr><td>Clean label</td><td colspan="3">ENTY</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">ENTY</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>DESC</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 24</td><td>ENTY</td><td>a=1.0</td><td><code>(|^)(what|what)[^\w]*(\w+){0,1}(is|is)[^\w]*</code>
<code>*([^\s]+)*(surname|address|name|name)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 61</td><td>HUM</td><td>a=0.0</td><td><code>(|^)(called|alias|nicknamed|nicknamed)[^\w]*(|$)</code></td></tr>
</table>

Table 15: TREC example: Student is wrong but unsure. Teacher predicts the right label and fixes student’s mistake. Teacher correctly down-weights rule 42 and rule 61 that provide the wrong prediction but erroneously up-weights the Student. As in this case the Student is uncertain, about the label, the final aggregated prediction of the Teacher is mostly influenced by Rule 24.

<table>
<tr><td>Text</td><td colspan="3">What currency do they use in Brazil ?</td></tr>
<tr><td>Clean label</td><td colspan="3">ENTY</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">ENTY</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>ENTY</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 54</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(how|what|what)[^\w]*</code>
<code>* (\w+){0,1}(do|does|does)[^\w]*(|$)</code></td></tr>
</table>

Table 16: TREC example. The student is crucial for RAN to predict the right label (ENTITY) as both rules predict the wrong label. RAN correctly down-weights the two rules and up-weights the Student.

<table>
<tr><td>Text</td><td colspan="3">What is an atom ?</td></tr>
<tr><td>Clean label</td><td colspan="3">DESC</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">DESC</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>DESC</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=1.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
</table>

Table 17: TREC example: Rule 42 was down-weighted in the previous two examples but is up-weighted here, demonstrating that RAN effectively leverages the contextualized instance representation to predict instance-specific rule weights.

<table>
<tr><td>Text</td><td colspan="3">What was President Lyndon Johnson ’s reform program called ?</td></tr>
<tr><td>Clean label</td><td colspan="3">ENTY</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">ENTY</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>ENTY</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 8</td><td>HUM</td><td>a=1.0</td><td><code>(|^)(who|what|what)[^\w] *(\w+){0,1}(person|man|woman|human|president|president)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 24</td><td>ENTY</td><td>a=1.0</td><td><code>(|^)(what|what)[^\w]*(\w+){0,1}(is|is)[^\w]*([^\s]+)*(surname|address|name|name)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 61</td><td>HUM</td><td>a=0.0</td><td><code>(|^)(called|alias|nicknamed|nicknamed)[^\w]*(|$)</code></td></tr>
</table>

Table 18: TREC example.

<table>
<tr><td>Text</td><td colspan="3">What is the percentage of water content in the human body ?</td></tr>
<tr><td>Clean label</td><td colspan="3">NUM</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">NUM</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>DESC</td><td>a=0.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 5</td><td>HUM</td><td>a=0.0</td><td><code>(|^)(which|who|what|what)[^\w]*([^\s]+)*(person|man|woman|human|poet|poet)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 41</td><td>NUM</td><td>a=0.2</td><td><code>(|^)(what|what)[^\w]* ([^\s]+)*(percentage|share|number|population|population)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
</table>

Table 19: TREC example.

<table>
<tr><td>Text</td><td colspan="3">How fast is alcohol absorbed ?</td></tr>
<tr><td>Clean label</td><td colspan="3">NUM</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">NUM</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>NUM</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
</table>

Table 20: TREC example: While no rules apply to this instance, the student associates “How fast” with the NUMBER (NUM) class.

<table>
<tr><td>Text</td><td colspan="3">Which mountain range in North America stretches from Maine to Georgia ?</td></tr>
<tr><td>Clean label</td><td colspan="3">LOC</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">LOC</td></tr>
<tr><td>Weak Source</td><td>Label</td><td>Weight</td><td>Feature</td></tr>
<tr><td>Student</td><td>LOC</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
</table>

Table 21: TREC example: While no rules apply to this instance, the student associates the context with the LOCATION (LOC) class.

<table>
<tr><td>Text</td><td colspan="3">When is the official first day of summer ?</td></tr>
<tr><td>Clean label</td><td colspan="3">NUM</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">NUM</td></tr>
<tr><th>Weak Source</th><th>Label</th><th>Weight</th><th>Feature</th></tr>
<tr><td>Student</td><td>NUM</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
</table>

Table 22: TREC example: While no rules apply to this instance, the student associates “when,” “day,” and “summer” to NUMBER (NUM) class.

<table>
<tr><td>Text</td><td colspan="3">What is Australia ’s national flower ?</td></tr>
<tr><td>Clean label</td><td colspan="3">ENTY</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">DESC</td></tr>
<tr><th>Weak Source</th><th>Label</th><th>Weight</th><th>Feature</th></tr>
<tr><td>Student</td><td>DESC</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
</table>

Table 23: TREC example: Both the Student and Rule 42 provide a wrong prediction

<table>
<tr><td>Text</td><td colspan="3">What is the name of the chocolate company in San Francisco ?</td></tr>
<tr><td>Clean label</td><td colspan="3">HUM</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">ENTY</td></tr>
<tr><th>Weak Source</th><th>Label</th><th>Weight</th><th>Feature</th></tr>
<tr><td>Student</td><td>ENTY</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 53</td><td>ENTY</td><td>a=1.0</td><td><code>((|^)(name|name)[^\w]*(|$))</code></td></tr>
<tr><td>Rule 59</td><td>ENTY</td><td>a=1.0</td><td><code>(|^)(which|what|what)[^\w]*</code>
<code>*([^\s]+)*(organization|trust|company|company)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 67</td><td>ENTY</td><td>a=0.6</td><td><code>(|^)(what|what)[^\w]* (\w+){0,1}(is|is)[^\w]*</code>
<code>*([^\s]+)*(surname|address|name|name)[^\w]*(|$)</code></td></tr>
</table>

Table 24: The clean label in this case is Human, while from the text we understand that the label is Entity.

<table>
<tr><td>Text</td><td colspan="3">What does I.V. stand for ?</td></tr>
<tr><td>Clean label</td><td colspan="3">ABBR</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">ABBR</td></tr>
<tr><th>Weak Source</th><th>Label</th><th>Weight</th><th>Feature</th></tr>
<tr><td>Student</td><td>ABBR</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 54</td><td>DESC</td><td>a=0.1</td><td><code>(|^)(how|what|what)[^\w]*</code>
<code>* (\w+){0,1}(do|does|does)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 68</td><td>ABBR</td><td>a=1.0</td><td><code>(|^)(what|what)[^\w]* (\w+){0,1}(does|does)[^\w]*</code>
<code>* * ([^\s]+)*(stand for)[^\w]*(|$)</code></td></tr>
</table>

Table 25: TREC example.

<table>
<tr><td>Text</td><td colspan="3">What year did the Titanic sink ?</td></tr>
<tr><td>Clean label</td><td colspan="3">NUM</td></tr>
<tr><td>RAN Teacher</td><td colspan="3">NUM</td></tr>
<tr><th>Weak Source</th><th>Label</th><th>Weight</th><th>Feature</th></tr>
<tr><td>Student</td><td>NUM</td><td>a=1.0</td><td>h_i (contextualized instance embedding)</td></tr>
<tr><td>Rule 29</td><td>NUM</td><td>a=1.0</td><td><code>(|^)(which|what|what)[^\w]*</code>
<code>*([^\s]+)*(time|day|month|hours|minute</code>
<code>*|seconds|year|date|date)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 32</td><td>NUM</td><td>a=1.0</td><td><code>(|^)(year|year)[^\w]*(|$)</code></td></tr>
<tr><td>Rule 42</td><td>DESC</td><td>a=0.0</td><td><code>(|^)(explain|describe|what|what)[^\w]*(|$)</code></td></tr>
</table>

Table 26: TREC example.