Co-training for Low Resource Scientific Natural Language Inference

Mobashir Sadat **Cornelia Caragea**
Computer Science
University of Illinois Chicago
msadat3@uic.edu cornelia@uic.edu

Abstract

Scientific Natural Language Inference (NLI) is the task of predicting the semantic relation between a pair of sentences extracted from research articles. The automatic annotation method based on distant supervision for the training set of SciNLI (Sadat and Caragea, 2022b), the first and most popular dataset for this task, results in label noise which inevitably degenerates the performance of classifiers. In this paper, we propose a novel co-training method that assigns weights based on the training dynamics of the classifiers to the distantly supervised labels, reflective of the manner they are used in the subsequent training epochs. That is, unlike the existing semi-supervised learning (SSL) approaches, we consider the historical behavior of the classifiers to evaluate the quality of the automatically annotated labels. Furthermore, by assigning importance weights instead of filtering out examples based on an arbitrary threshold on the predicted confidence, we maximize the usage of automatically labeled data, while ensuring that the noisy labels have a minimal impact on model training. The proposed method obtains an improvement of $\mathbf{1.5\%}$ in Macro F1 over the distant supervision baseline, and substantial improvements over several other strong SSL baselines. We make our code and data available on Github.[1]

1 Introduction

Scientific Natural Language Inference (NLI) aims at predicting the semantic relation between a pair of sentences extracted from research articles. This task was recently proposed by Sadat and Caragea (2022b) as a Natural Language Understanding (NLU) benchmark for scientific text along with a new dataset called SciNLI. The training set of SciNLI was constructed automatically using a distant supervision method (Mintz et al., 2009). The labels assigned using this method contain strong signals, and achieved a Macro F1 of $\sim 78\%$ on the human annotated test set using pre-trained language models. Nevertheless, despite the strong signals, the labels assigned based on distant supervision still contain noise that inevitably degenerates the performance of classifiers. The noisy labels can be particularly harmful for training large scale deep learning models because these models can achieve zero training loss even for mislabeled examples simply by memorizing them (Zhang et al., 2021b; Arpit et al., 2017).

Semi-supervised learning (SSL) methods such as pseudo-labeling, i.e., self-training (Xie et al., 2020; Chen and Yang, 2021), consistency regularization (Sajjadi et al., 2016; Sohn et al., 2020), and co-training (Blum and Mitchell, 1998) have emerged as promising approaches that utilize a small amount of human annotated data, and learn from a large amount of unlabeled data. Generally, these methods first train models with the limited human labeled data, and combine their most confident predictions (selected by applying a fixed high threshold, e.g., 0.9 on confidence) for unlabeled examples with the human labeled data for subsequent training. While a fixed high threshold ensures high quality pseudo-labels, it also ignores a large fraction of diverse examples that have correct pseudo-labels but with lower confidence. Consequently, several dynamic thresholding mechanisms (Zhang et al., 2021a; Xu et al., 2021) have been proposed to ensure a higher utilization of the pseudo-labels. However, a dynamic threshold still discards a large number of examples, compromises the pseudo-label quality, and is susceptible to error accumulation.

In this paper, we propose a novel co-training approach for scientific NLI that takes the signal from labels assigned based on distant supervision, but their quality is discerned by two classifiers simultaneously trained on different regions of the data map (Swayamdipta et al., 2020) that guide each other in the learning process. Unlike existing co-training

[1] https://github.com/msadat3/weighted_cotraining

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2538–2550
August 11-16, 2024 ©2024 Association for Computational Linguistics

approaches which also train two classifiers simultaneously in a cross-labeling manner by exchanging the most confident pseudo-labels, in our approach, the quality of each distantly supervised label is determined mutually by each classifier in the form of *importance weights*, which are exchanged between the two classifiers. Thus, instead of exchanging the pseudo-labels, we exchange the classifiers' beliefs in the quality of the distantly supervised labels. In addition, we do not discard any examples but utilize all of them with different *importance weights* to ensure that noisy labels have a minimal impact on model training.

The weights in our approach are estimated by monitoring the training dynamics to capture the behavior of the classifiers on each example—*easy*, *hard*, and *ambiguous*. Specifically, we calculate the weights based on both average **confidence**, and **variability** of the probability predicted over the training epochs for each label assigned based on distant supervision. Our weight calculation strategy ensures a high weight of the *easy* examples which are more likely to be clean, and a low weight of the *hard* examples which are more likely to be noisy for both classifiers; and a contrasting weight of the *ambiguous* examples by each classifier to encourage divergence between them.

We explore our proposed approach by using a small amount of human annotated examples, and a large number of automatically annotated examples based on distant supervision from the SciNLI training set. Our experiments show that the proposed approach improves the performance by more than **1.5%** over distant supervision (i.e., models trained directly on the SciNLI training set); and obtains substantial improvements over co-training, co-teaching (Han et al., 2018), and several other baselines designed for low-resource settings. Our key contributions can be summarized as follows:

- We develop a novel co-training approach for scientific NLI that utilizes the historical training dynamics to evaluate the quality of distantly supervised labels to assign *importance weights* for these labels.
- We thoroughly evaluate our proposed method by comparing its performance with distant supervision, co-training, co-teaching, and several other strong SSL methods.
- We present the first ever *human annotated* training set for scientific NLI containing $2,000$ examples which we will make publicly available for future research.

2 Related Work

Scientific NLI Sadat and Caragea (2022b) proposed a distant supervision method based on linking phrases to automatically annotate large scale training datasets for the scientific NLI task and introduced the first dataset for this task which is called SciNLI. This task consists of four classes: ENTAILMENT, REASONING, CONTRASTING and NEUTRAL. The training examples for the former three classes are annotated automatically based on linking phrases indicative of these relations. For example, if the second sentence in an adjacent sentence pair starts with "Therefore," the pair is labeled as REASONING. Random non-adjacent sentences are paired together and labeled as NEUTRAL. Using the same distant supervision method, MSciNLI (Sadat and Caragea, 2024) was recently introduced to cover multiple scientific domains. The automatic annotation method results in noisy labels that can harm the performance and generalization of the classifiers.

Semi-supervised Learning Self-training (Xie et al., 2020; Becker et al., 2013; Mukherjee and Awadallah, 2020; Sadat and Caragea, 2022a) incorporates unlabeled data into model training by using three general steps: **a)** training a model using the available labeled examples; **b)** assigning pseudo-labels to unlabeled examples based on the model's prediction; **c)** selecting a subset of pseudo-labeled examples based on some quality assurance measures and using them for further model training in addition to the labeled examples. Consequently, the model gets exposed to more data which results in a better performance.

Co-training methods (Blum and Mitchell, 1998; Wan, 2009; Gollapalli et al., 2013; Chen et al., 2011; Qiao et al., 2018; Zou and Caragea, 2023) employ a similar approach as self-training to incorporate pseudo-labeled data into model training with a key difference. Instead of training a single classifier as self-training, two classifiers are trained simultaneously which exchange their most confident pseudo-labels for further training. In contrast to the existing approaches, we do not discard any examples, and enable co-training by exchanging *importance weights* indicative of the distantly supervised label quality, calculated based on the classifiers' training dynamics.

Consistency regularization (Sajjadi et al., 2016; Laine and Aila, 2017) uses a consistency loss term (in addition to the supervised loss) that minimizes the distance between the predictions made by the model for different versions of the same unlabeled data. For example, FixMatch (Sohn et al., 2020) first generates pseudo-labels for weak augmentations (e.g., replacing a random subset of tokens in a sentence with their synonyms) of unlabeled examples. A subset of these examples are then selected based on a fixed confidence threshold. FixMatch then uses strong augmentations (e.g., automatically translating a sentence to a different language and translating it back to the original language to get its paraphrased version) of the selected unlabeled examples as the input with these pseudo-labels as their target. Recently, there have been a plethora of consistency regularization based methods that build on top of FixMatch. For example, FlexMatch (Zhang et al., 2021a) uses a label-wise dynamic thresholding method based on the learning status for each label. SoftMatch (Chen et al., 2023) assigns soft weights to pseudo-labels based on the model confidence instead of filtering them out. Although SoftMatch does not filter out any examples, the soft weights are assigned based only on the current training iteration. That is, similar to other SSL approaches, SOFTMATCH ignores the historical behavior of the classifier on the unlabeled examples.

Co-teaching The methods for learning from noisy labels generally rely on the classifier prediction during training to evaluate the quality of the labels. From this point of view, this research area is closely related to SSL. Co-teaching (Han et al., 2018) is a method for training two classifiers simultaneously (similar to co-training) which exchange small-loss training instances (instances with low training loss) for further training. One of the key motivations behind training two classifiers is that each classifier will filter our different types of noise by learning complementary information. If the classifiers reach a consensus, they are no longer able to complement each other, reducing the learning process to single classifier training. Therefore, a divergence between the classifiers throughout the training process is crucial. To this end, De-coupling (Malach and Shalev-Shwartz, 2017) and Co-teaching+ (Yu et al., 2019) use only the examples for which the classifiers disagree in their predictions. However, similar to SSL approaches, these methods also evaluate the label quality based only on the current classifiers' predictions; ignore the history based on their training dynamics; and discard a large number of examples.

3 Background

We design our proposed approach based on our observations from the data map (Swayamdipta et al., 2020) of SCINLI. In this section, we define the metrics used to characterize each example; plot them on a data map based on their characterizations; and discuss our observations from the data map that guide the development of our approach.

Notations Consider a dataset $D = \{D^l \cup D^a\}$ where $D^l = \{(\mathbf{x}^i, y^i)\}_{i=1,...,n}$ is a small human labeled training set of size n; and $D^a = \{(\mathbf{x}^i, \tilde{y}^i)\}_{i=1,...,m}$ is an automatically labeled set of size m, constructed using distant supervision. $\mathbf{x}^i$ is a premise-hypothesis pair, y^i is a human annotated label, $\tilde{y}^i$ is an automatically assigned label, and $n \ll m$. We denote $p(y|\mathbf{x};\theta)$ as the predicted probability for an assigned label y by a classifier θ, given the premise-hypothesis pair $\mathbf{x}$.

Dataset Cartography Dataset cartography (Swayamdipta et al., 2020) is a tool for mapping and diagnosing a dataset by analyzing the training dynamics i.e., the model's behavior on each example during training. Particularly, each example is characterized by three metrics—**confidence**, **variability**, and **correctness**, defined below.

For an example $(\mathbf{x}, y)$, Swayamdipta et al. (2020) defines the confidence as the mean of the probabilities predicted for its assigned label y over the training epochs $\{1, .., e\}$. That is, the confidence is calculated as follows:

$$c_\theta(\mathbf{x}, y) = \frac{1}{e}\sum_{t=1}^{e} p(y|\mathbf{x};\theta^t) \tag{1}$$

Here, c_θ is a function that calculates the confidence of an example ($\mathbf{x}$,y) by a classifier θ.

The variability is defined as the standard deviation of the predicted probability for the assigned label y over the training epochs. Specifically, variability is calculated as follows:

$$v_\theta(\mathbf{x}, y) = \sqrt{\frac{\sum_{t=1}^{e}(p(y|\mathbf{x};\theta^t) - c_\theta(\mathbf{x}, y))^2}{e}} \tag{2}$$

The correctness is calculated as the fraction of epochs in which the classifier predicts the assigned label y correctly.

We train a ROBERTA (Liu et al., 2019) classifier on the full SCINLI training set D, and plot 55K

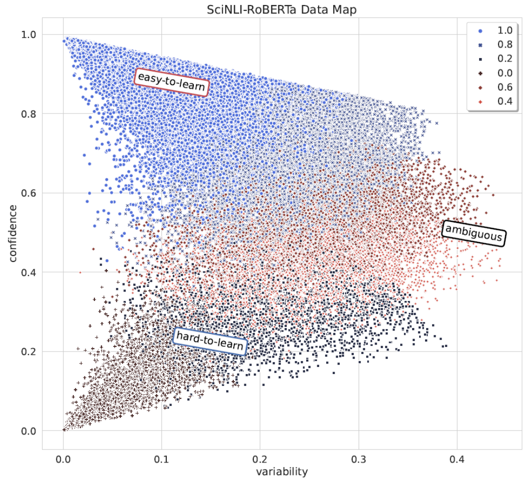

Figure 1: Data cartography of SCINLI. The colors and shapes indicate the correctness of each example.

randomly selected examples in Figure 1 based on their training dynamics. Note that, we plot only 55K examples (instead of all 101K) for clarity.

The plot shows that the examples with high confidence and low variability (top left corner) have a high correctness. That is, the model's prediction consistently matches with the assigned label for these examples. Therefore, they are *easy* to learn for the model, and the assigned labels are more likely to be correct. In contrast, the examples with low confidence and low variability (bottom left corner of the plot) have a low correctness. In other words, the model's predictions disagree with the assigned labels consistently. Thus, the examples in this region are *hard* for the model to learn, and the assigned labels are more likely to be incorrect. The examples with a high variability (right side of the plot) shows a moderate correctness, and confidence. This indicates the model is uncertain about these examples, and their predictions occasionally match with the assigned labels. In our approach, we employ a weighting strategy that utilizes the information from the data maps to assign importance weights to distantly supervised examples during training.

4 Proposed Approach

We now present our co-training approach where we train two classifiers simultaneously that capture complementary information from the data. Unlike the existing co-training approaches, we utilize all examples in dataset D^a, and do not filter out anything. Rather, our approach assigns importance weights to the automatically annotated examples. Based on these weights, we decide the impact of each automatically annotated example in subsequent training of the classifiers. We devise the weights of the automatically labeled examples guided by the training dynamics of the classifiers in the form of confidence and variability of each example over the training epochs. Furthermore, in contrast to the existing approaches which exchange pseudo-labels between classifiers, we exchange the weights calculated based on the historical behavior of the classifiers in evaluating the quality of each example. We ensure that the two classifiers learn complementary information by employing a weight calculation strategy that maintains their divergence. An overview of our approach can be seen in Alg. 1.

4.1 Weight assignment

The two classifiers that we co-train are denoted as θ_1 and θ_2. We now describe how we assign importance weights to the automatically annotated examples at each epoch to train the classifiers.

We calculate two sets of weights in our co-training method based on the training dynamics of the classifiers on the distantly supervised labels. Specifically, for each $\tilde{y}^i$ in D^a, we calculate $c_{\theta_1}(\mathbf{x^i}, \tilde{y}^i)$, and $c_{\theta_2}(\mathbf{x^i}, \tilde{y}^i)$ using Eq. 1; and $v_{\theta_1}(\mathbf{x^i}, \tilde{y}^i)$, and $v_{\theta_2}(\mathbf{x^i}, \tilde{y}^i)$ using Eq. 2. The weights λ_1^i, and λ_2^i for each example to be used for subsequent training of θ_2, and θ_1, respectively are calculated as follows:

$$\lambda_1^i = c_{\theta_1}(\mathbf{x^i}, \tilde{y}^i) + v_{\theta_1}(\mathbf{x^i}, \tilde{y}^i) \tag{3}$$

$$\lambda_2^i = c_{\theta_2}(\mathbf{x^i}, \tilde{y}^i) - v_{\theta_2}(\mathbf{x^i}, \tilde{y}^i) \tag{4}$$

Therefore, the weights for the *easy* examples (high confidence, low variability) will be high for both classifiers, and the weights for the *hard* (low confidence, low variability) examples will be low for both classifiers. For the *ambiguous* examples where the variability is high, the weight for θ_1 (i.e, λ_2^i) will be low given that the classifiers show a moderate confidence for these examples (see the right side of Figure 1). For example, a confidence of 0.4, and a variability of 0.3 results in low weight of 0.1. On the other hand, if the variability is high, the weight for θ_2 (i.e, λ_1^i) will remain high even if the confidence is moderate. For example, a confidence of 0.4, and a variability of 0.3 results in high weight of 0.7. We normalize the weights using a min-max normalization method. Therefore, both sets of weights are scaled to a range of 0 to 1.

Our weighting strategy ensures that a) the *easy* examples which are likely to be correctly labeled,

Algorithm 1 Weighted Co-training for Scientific NLI

Require: Human labeled set, $D^l = \{(\mathbf{x}^i, y^i)\}_{i=1,\dots,n}$; automatically labeled set, $D^a = \{(\mathbf{x}^i, \tilde{y}^i)\}_{i=1,\dots,m}$; Maximum training epochs E; learning rate η.

Train θ_1, and θ_2 on two randomly divided equal subsets of D^l; for each automatically annotated example i in D^a, store the probability $p(\tilde{y}^i|\mathbf{x}^\mathbf{i};\theta_1)$ and $p(\tilde{y}^i|\mathbf{x}^\mathbf{i};\theta_2)$ after each epoch.
for each $(x^i, \tilde{y}^i) \in D^a$ **do**
    Calculate confidences $c_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$ and $c_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$, and variabilities $v_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$, and $v_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$ from the stored probabilities using Eq. 1 and Eq. 2.
    $\lambda_1^i \leftarrow c_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i) + v_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$; and $\lambda_2^i \leftarrow c_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i) - v_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$. ▷ Assign initial importance weights.
end for
Re-initialize θ_1, θ_2 ▷ Re-initialize the classifiers for co-training.
for $e = 1$ to E **do**
    for each mini-batch $B \in D^a$ **do**
        $\mathcal{L}_1 \leftarrow \frac{1}{|B|}\sum_{i=0}^{|B|} \lambda_2^i * H(\tilde{y}^i, p_d(\mathbf{x}^i;\theta_1))$; $\mathcal{L}_2 \leftarrow \frac{1}{|B|}\sum_{i=0}^{|B|} \lambda_1^i * H(\tilde{y}^i, p_d(\mathbf{x}^i;\theta_2))$. ▷ Calculate cross-entropy loss.
        for each DS-LABELED example $i \in B$ **do**
            Update $c_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$ and $c_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$ using Eq. 1; update $v_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$ and $v_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$ using Eq 2.
            $\lambda_1^i \leftarrow c_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i) + v_{\theta_1}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$; and $\lambda_2^i \leftarrow c_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i) - v_{\theta_2}(\mathbf{x}^\mathbf{i}, \tilde{y}^i)$. ▷ Update the weights.
        end for
        $\theta_1 \leftarrow \theta_1 - \eta * \nabla\mathcal{L}_1$; $\theta_2 \leftarrow \theta_2 - \eta * \nabla\mathcal{L}_2$ ▷ Update the parameters of θ_1, and θ_1
    end for
end for
Fine-tune θ_1, and θ_2 on two randomly divided equal subsets of human annotated data with a learning rate lower than η.

participate in the training of both classifiers with high weights; b) the *hard* examples which are likely to be incorrectly labeled, participate in the training of both classifiers with low weights; and c) the *ambiguous* examples for which the classifiers are uncertain about the assigned labels, participate in the training of one classifier with high weights, and participate in the training of the other classifier with low weights. This allows the classifiers to remain divergent, and ensure that they capture complementary information from the data. That is, in contrast to prior methods (e.g., De-coupling and Co-teaching+), we enforce divergence between the classifiers by employing a contrasting usage of the ambiguous examples for the classifiers.

4.2 Weighted Co-training

We co-train the two classifiers θ_1 and θ_2 in three major steps. They are described below.

Step 1: Initial weight assignment. To assign the initial weights to the automatically annotated examples in D^a, we train both classifiers θ_1 and θ_2 using two randomly divided subsets of equal size of the human annotated set in D^l. At the end of each epoch, we record the predicted probabilities $p(\tilde{y}^i|\mathbf{x}^\mathbf{i};\theta_1)$ and $p(\tilde{y}^i|\mathbf{x}^\mathbf{i};\theta_2)$ for each example in D^a. We then calculate the confidences, and variabilities based on the recorded probabilities using Eq. 1 and 2, respectively. Based on these confidences and variabilities, the initial weights for the automatically annotated examples are then calculated using Eq. 3 and 4.

Step 2: Co-training epochs. We co-train θ_1 and θ_2 using D^a. We do not include D^l in the co-training epochs because assigning importance weights to a combination of human labeled, and automatically labeled examples results in an unfair advantage to the former. Therefore, D^l is utilized separately in an additional step.

Both classifiers are re-initialized before we start the co-training epochs. At each epoch, the cross-entropy loss for each classifier for an example is scaled by its assigned weight by the other classifier. In particular, for each mini-batch B, the losses $\mathcal{L}_1$ and $\mathcal{L}_2$ for θ_1 and θ_2 are calculated as follows:

$$\mathcal{L}_1 = \frac{1}{|B|}\sum_{i=0}^{|B|} \lambda_2^i * H(\tilde{y}^i, p_d(\mathbf{x}^i;\theta_1)) \quad (5)$$

$$\mathcal{L}_2 = \frac{1}{|B|}\sum_{i=0}^{|B|} \lambda_1^i * H(\tilde{y}^i, p_d(\mathbf{x}^i;\theta_2)) \quad (6)$$

Here, p_d is the probability distribution over the labels predicted by a classifier for an automatically annotated example i in B, and H is the standard cross-entropy loss.

After calculating the loss for both classifiers, we update the confidences, and variabilities using Eq. 1, and 2. Note that the probabilities predicted by the classifiers in the last epoch in Step 1, are used as initial probabilities for both classifiers, and are included in calculating the mean, and the standard deviation in the co-training epochs. We then update both sets of weights using Eq. 3 and 4, based on the updated confidences and variabilities, to be used in the next epoch. Therefore, the classifiers co-train each other by exchanging the complementary information from their respective training dynamics over the epochs.

Step 3: Fine-tuning. The co-training epochs continue until we reach a pre-defined maximum epoch E. We then fine-tune θ_1 and θ_2 using two randomly divided subsets of equal size of D^l.

4.3 Classifier Ensembling

We ensemble the co-trained classifiers by performing an element-wise average of their softmax output (i.e., the probability distribution). We then take the $\arg\max$ of the ensembled probability distribution to predict the labels for the test examples.

5 Experiments

This section describes the baselines, our models, experimental settings, and results.

5.1 Baselines

The baselines that we consider for evaluating our proposed method are described below.

FULLY-SUPERVISED (FS) This baseline trains a fully supervised classifier using human labeled or H-LABELED data. We explore two variants of this baseline — FS (H-LABELED 1K), and FS (H-LABELED 2K) which use $1,000$ randomly sampled class-balanced H-LABELED examples, and all 2000 H-LABELED examples, respectively.

DISTANT SUPERVISION (DS) This is the distantly supervised counterpart of the FS baselines. We experiment with three variants — DS-LABELED (1K), DS-LABELED (2K), and DS-LABELED (101K) using randomly sampled 1000, 2000, and all $101,412$ examples in the SCINLI training set, respectively.

BACK-TRANSLATION (BT) (Yu et al., 2018) A data augmentation method based on machine translation where each example is translated to French and then translated back to English to get their paraphrased versions. The paraphrased and original versions of the dataset are combined and models are trained on this larger set. We explore back-translation on top of both FS baselines.

DBST (Sadat and Caragea, 2022a) A self-training method for NLI where examples are selected based on whether the automatically assigned label (e.g., distantly supervised label in SCINLI) of each example matches with the predicted label by the classifier and whether the confidence for the predicted label is above a pre-defined threshold.

FIXMATCH (Sohn et al., 2020) A consistency regularization method based on strong and weak augmentations of unlabeled data. We use synonym replacement as the weak augmentation, and back-translation as the strong augmentation technique. FixMatch does not consider the distantly supervised labels, and relies on the classifier prediction.

FLEXMATCH (Zhang et al., 2021a) FLEXMATCH is an SSL approach that uses FIXMATCH as its backbone. In contrast to FIXMATCH, FLEXMATCH applies different confidence thresholds for different classes based on their learning status.

SOFTMATCH (Chen et al., 2023) SOFTMATCH also uses FIXMATCH as its backbone method. However, instead of filtering out pseudo-labeled examples, it assigns importance weights based on the model confidence in the current training epoch.

CO-TRAINING A vanilla CO-TRAINING approach which does not make use of distantly supervised labels, and exchanges a subset of the predictions selected based on a confidence threshold between the classifiers for training them.

CO-TEACHING (Han et al., 2018) CO-TEACHING is a method for learning from noisy labels. It does not use any human annotated data, and trains two classifiers simultaneously. The classifiers exchange the small-loss examples (i.e., examples with low cross-entropy loss) which are used for their training.

CO-TEACHING+ (Yu et al., 2019) This method is similar to CO-TEACHING, except it enforces a divergence between the classifiers by using only the small-loss examples for which the classifiers disagree in their predictions.

5.2 Our Models

We train the following models based on our proposed approach.

Weighted Co-training (WCT) - cv WCT - CV denotes the classifiers we train using our proposed co-training approach described in Section 4.

Weighted Co-training (WCT) - cc A variant of our proposed co-training approach where the weights for the automatically labeled examples are calculated only based on the confidences, and the variabilities are not used.

Method	SciBERT		RoBERTa	
	Macro F1	Acc	Macro F1	Acc
DS-LABELED (1K)	61.21 ± 1.8	61.35 ± 1.7	63.29 ± 2.9	63.45 ± 2.6
FS (H-LABELED 1K)	65.31 ± 0.7	65.39 ± 0.6	67.70 ± 0.2	67.78 ± 0.2
BT (H-LABELED 1K)	64.13 ± 0.4	64.17 ± 0.3	67.73 ± 1.1	67.78 ± 1.1
DS-LABELED (2K)	64.74 ± 1.6	64.75 ± 1.4	67.01 ± 1.6	67.02 ± 1.4
FS (H-LABELED 2K)	67.46 ± 1.1	67.51 ± 1.1	70.13 ± 0.3	70.14 ± 0.2
BT (H-LABELED 2K)	68.12 ± 1.9	68.18 ± 1.9	70.23 ± 0.7	70.29 ± 0.7
DS-LABELED (101K)	77.53 ± 0.5	77.52 ± 0.5	78.08 ± 0.4	78.11 ± 0.4
DBST (Sadat and Caragea, 2022a)	73.61 ± 0.4	73.59 ± 0.5	73.74 ± 0.3	73.72 ± 0.3
FIXMATCH (Sohn et al., 2020)	68.21 ± 0.8	68.21 ± 0.8	71.27 ± 0.2	71.32 ± 0.2
FLEXMATCH (Zhang et al., 2021a)	68.60 ± 0.9	68.56 ± 0.8	71.64 ± 0.3	71.65 ± 0.4
SOFTMATCH (Chen et al., 2023)	68.77 ± 1.3	68.70 ± 1.3	71.80 ± 0.2	71.65 ± 0.4
CO-TRAINING	68.29 ± 1.7	68.28 ± 1.7	70.35 ± 0.2	70.47 ± 0.2
CO-TEACHING (Han et al., 2018)	78.06 ± 0.3	78.05 ± 0.3	78.72 ± 0.4	78.77 ± 0.5
CO-TEACHING+ (Yu et al., 2019)	76.27 ± 0.3	76.24 ± 0.3	76.47 ± 0.4	76.45 ± 0.5
WCT - CC	78.35 ± 0.5	78.36 ± 0.5	$79.12^{*} \pm 0.2$	$79.15^{*} \pm 0.1$
WCT - CV	$\mathbf{78.55^{*} \pm 0.4}$	$\mathbf{78.57^{*} \pm 0.4}$	$\mathbf{79.62^{*\#} \pm 0.3}$	$\mathbf{79.65^{*\#} \pm 0.3}$

Table 1: The Macro F1 and Accuracies of different methods on SCINLI. Best scores are in bold. * and # indicate statistically significant improvements over DS-LABELED (101K) and CO-TEACHING (our best performing baseline), respectively, according to a paired t-test with $p < 0.05$.

5.3 Experimental Settings

Dataset We consider the full SCINLI training set containing 101K examples as our dataset D. We then employ three expert annotators to manually curate $2,000$ examples from D which are used as the human labeled set, D^l. A detailed description of our manual data annotation method is available in Appendix A. We then remove any example that was extracted from the same paper as the examples in D^l from D. This resulted in 97K examples which are used as D^a in our approach.

Implementation Details We experiment with the base variants of both SCIBERT (Beltagy et al., 2019) and ROBERTA (Liu et al., 2019) for the baselines and our models. More implementation details are available in Appendix B. Each experiment is run three times with different random seeds. The average and standard deviations of the Macro F1 and accuracy of different methods are in Table 1. Our findings are described below.

5.4 Results & Observations

Weighted co-training vs. distant supervision and data augmentation. As we can see from the results, WCT - CV shows significant improvements over the DS-LABELED (101K) baseline. In addition, our proposed approach substantially outperforms all FS, and BT baselines. These improvements in performance illustrate that, our proposed co-training approach is successful in reducing the impact of noisy labels on classifier training, resulting in a better performance.

Weighted co-training vs. co-teaching. Comparing WCT - CV and CO-TEACHING, we can see that WCT - CV shows better performance. Furthermore, in contrast to our method, the improvement shown by CO-TEACHING over the DS-LABELED (101K) is not statistically significant. We can also see that CO-TEACHING+ which aims at maintaining the divergence between the classifiers by using only disagreement data for training the classifiers, shows a poor performance. These observations illustrate that a) using only the predictions from the current epoch is myopic, and valuable information from the prior epochs is lost; and b) while the disagreement data possibly improves the divergence of the classifiers, the agreement data also contains diverse patterns, and are necessary.

WCT - CV vs. WCT - CC While WCT - CC shows a higher Macro F1 than all other baselines, it shows a lower performance compared with WCT - CV. Note that WCT - CC uses only confidence to assign the weights to the automatically labeled examples for both classifiers. That is, unlike WCT - CV, WCT - CC does not enforce a divergence between the classifiers by using contrasting weights for the *ambiguous* examples. Therefore, our weighting strategy in WCT - CV indeed improves the divergence between the classifiers, which results in improved performance.

Weighted co-training vs. vanilla co-training. The results show that the CO-TRAINING baseline which does not make use of the distantly super-

Approach	SciBERT	RoBERTa
SIMPLE FT - ENSEMBLED	78.03	78.52
WST - ENSEMBLED	78.46	79.33
WST - R	78.25	78.50
WCT - CV BOTH 2K	78.29	79.35
WCT - CVH	78.31	79.29
WCT - CV	**78.55**	**79.62**$^{*\#}$

Table 2: Macro F1 scores from the ablation experiments compared with WCT - CV. * and # indicate statistically significant improvements over SIMPLE FT - ENSEMBLED and WST - R, respectively, according to a paired t-test with $p < 0.05$. The improvements of WCT - CV over other ablations are not statistically significant.

vised labels, and uses the most confident classifier predictions, achieves a much lower Macro F1 than the DS-LABELED (101K) baseline. In addition, it only shows marginal improvements over the FS baselines. Therefore, distantly supervised labels contain strong signals which can be beneficial for model training, rather than using model predictions which can be unreliable, especially in the initial training epochs.

Weighted co-training vs. other SSL approaches. We can see that the SSL approaches—FIXMATCH, FLEXMATCH, and SOFTMATCH show a much lower performance than DS-LABELED (101K). Note that these single-classifier (pseudo-labeling) approaches do not utilize the distantly supervised labels, and rely only on the classifier prediction similar to vanilla co-training. Given their susceptibility to error accumulation, and the challenging nature of the scientific NLI task, they fail to show any promising performance. DBST on the other hand utilizes the signal from distant supervision in addition to model prediction. Consequently, DBST shows improved performance over the other SSL baselines. However, it filters out examples based on a confidence threshold, resulting in a much lower performance compared to WCT - CV.

6 Analysis

Our analysis consists of two parts. First, we perform various ablation experiments (§6.1). Next, we analyze the generalizability of our co-training approach to other NLP tasks (§6.2). We also analyze the robustness of WCT - CV by evaluating its out-of-domain performance using MSCINLI (Sadat and Caragea, 2024) in Appendix C.

6.1 Ablation Experiments

Simple fine-tuning vs. co-training We perform an experiment where we first train two classifiers on D^a and then continue fine-tuning them on two randomly divided subsets of D^l. This approach is denoted as SIMPLE FT - ENSEMBLED. We compare the performance of this approach with the performance of WCT - CV and show the results in Table 2. As we can see, the performance of SIMPLE FT - ENSEMBLED is substantially lower than WCT - CV illustrating the necessity of SSL methods such as our proposed co-training method.

Ensembled self-training vs. co-training To evaluate the necessity of co-training two classifiers by exchanging information between them, we explore a method where we also train two classifiers simultaneously but do not exchange any information. We denote this method as WST - ENSEMBLED which stands for WEIGHTED SELF-TRAINING - ENSEMBLED. Table 2 shows that WST - ENSEMBLED obtains a lower Macro F1 than WCT - CV. Thus, co-training the classifiers by exchanging their knowledge is clearly more advantageous than simple ensembling.

Self-training with random weighting strategy vs. co-training To further evaluate the benefit of training two classifiers with co-training, we experiment with a weighted self-training method where we train a single classifier and randomly decide between Eq. 3 and Eq. 4 to assign the weight of each automatically annotated example. This self-training method is denoted as WST - R. As we can see in Table 2, the performance of the WST - R is lower than WCT - CV. This is because, similar to other SSL baselines, WST - R is more susceptible to error accumulation due to its reliance on a single classifier whereas, our co-training approach aims to learn complementary information from the data, which results in the classifiers being able to filter out different types of noise. Consequently, WCT - CV outperforms WST - R.

Multi-set vs. single-set co-training To understand the necessity of using different splits of human labeled data to train the two classifiers in our co-training approach, we explore a method named WCT - CV BOTH 2K. This method is similar to WCT - CV except it uses all $2,000$ human labeled examples to train both classifiers. We can see that the WCT - CV BOTH 2K shows a lower Macro F1 than WCT - CV illustrating that training the classifiers using two different splits of human labeled data enforces the classifiers to further capture complementary information.

Approach	ANLI R3	Dynasent R1
15% CORRUPTED	42.34	78.39
SIMPLE FT - ENSEMBLED	43.37	79.47
CO-TEACHING	43.83	78.72
WCT - CV	**44.43***	**79.91**$^{*\#}$

Table 3: Macro F1 of different methods for other tasks. * and # indicate statistically significant improvements over 15% CORRUPTED and CO-TEACHING, respectively, according to a paired t-test with $p < 0.05$.

High weighting vs. fine-tuning with human labeled To evaluate the necessity of utilizing automatically labeled and human labeled examples using separate steps (Step 2 and 3 of our approach, described in Section 4.2), we perform an experiment where we combine the human labeled examples with large weights (1.0) with the automatically labeled examples in our co-training epochs in Step 2. This approach is denoted as WCT - CVH.

As we can see in Table 2, WCT - CVH shows a $\approx 0.4\%$ drop in performance compared with WCT - CV with ROBERTA. Inspecting the weights for the automatically annotated examples in WCT - CVH, we find that in the early co-training epochs, their average weight is 0.5. This is because the classifiers in the early epochs tend to be less confident about their predictions which results in more than half of the automatically annotated examples having a weight less than 0.4. Because of the disproportionately high weights for the human labeled examples (1.0), the training of the classifiers become focused on them, and the automatically annotated examples remain underutilized due to their smaller weights, losing the benefits of increased data diversity. Consequently, we see a drop in the overall performance.

6.2 Generalizability Analysis

To evaluate the generalizability of our proposed co-training approach, we experiment with a challenging NLI dataset from the general domain — ANLI (Nie et al., 2020) and a challenging sentiment analysis dataset — Dynasent (Potts et al., 2021). In particular, we choose the Round 3 (R3) subset of ANLI and Round 1 (R1) subset of Dynasent. We choose these particular subsets because their training sets contain a large number of examples ($100K$ in ANLI R3, $90K$ in Dynasent R1). The experimental settings for these two datasets are described below.

We simulate a 'potentially noisy' setting similar to SCINLI by randomly corrupting 15% of the labels in the training sets of these datasets. $2,100$ non-corrupted class-balanced examples are used as D^l, and rest of the examples are used as D^a. We train ROBERTA models in four settings: 1) 15% CORRUPTED—the full training set after corrupting 15% of the labels ; 2) SIMPLE FT - ENSEMBLED—same method used in Section 6.1; 3) CO-TEACHING—our best performing baseline; and 4) WCT - CV—our proposed approach. Each experiment is run three times and the average of the Macro F1 scores are reported in Table 3. We find that:

Our co-training approach improves the performance for both ANLI and Dynasent. WCT - CV shows better performance than 15% CORRUPTED, SIMPLE FT - ENSEMBLED, and CO-TEACHING for both ANLI and DYNASENT. Therefore, while the primary objective of our proposed method is to improve the performance in scientific NLI, we can see that it can be useful for other challenging tasks.

7 Conclusion & Future Work

In this paper, we propose a novel co-training approach for scientific NLI. In contrast to the existing approaches, we assign importance weights to automatically annotated examples based on the historical training dynamics of the classifiers. Instead of filtering out examples based on an arbitrary threshold, we decide the impact of the automatically annotated examples in subsequent training based on their assigned weights. We encourage divergence between the classifiers by assigning a contrasting weight to the *ambiguous* examples to train each classifier. Our experiments show that the proposed approach obtains substantial improvements over the existing approaches. In the future, we will explore methods to further harness the training dynamics of the classifiers in improving the performance of SSL methods.

Acknowledgements

This research is supported by NSF CAREER award 1802358 and NSF IIS award 2107518, and UIC Discovery Partners Institute (DPI) award. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF or DPI. We thank our anonymous reviewers for their constructive feedback, which helped improve the quality of our paper.

Limitations

Our proposed approach shows promising performance in learning from noisy labels, but it requires a small human annotated training set. Manually annotating a small training set is significantly more feasible, and cheaper than manually removing the noisy examples from a large dataset. Nevertheless, one has to carefully consider the expense incurred for the small human annotated training set, and the obtained performance gain to employ our proposed method for other datasets and/or tasks.

Furthermore, since our approach trains two classifiers simultaneously, it requires a higher amount of computational resources. For example, we utilize two NVIDIA RTX A5000 GPUs to train the classifiers using our proposed approach whereas the single classifier based methods that we explored as our baselines are trained with a single GPU. While the necessity of higher amount of resources is a common constraint among all dual classifier based methods (e.g., CO-TEACHING), the trade-off between the performance gain with our approach and the additional computational costs should be carefully considered.

References

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Lee Becker, George Erhart, David Skiba, and Valentine Matula. 2013. AVAYA: Sentiment analysis on Twitter with self-training and polarity lexicon expansion. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 333–340, Atlanta, Georgia, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. 2023. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*.

Jiaao Chen and Diyi Yang. 2021. Simple conversational data augmentation for semi-supervised abstractive dialogue summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6605–6616.

Minmin Chen, Kilian Q Weinberger, and Yixin Chen. 2011. Automatic feature decomposition for single view co-training. In *ICML*, volume 2, page 5.

Sujatha Das Gollapalli, Cornelia Caragea, Prasenjit Mitra, and C. Lee Giles. 2013. Researcher homepage classification using unlabeled data. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 471–482, New York, NY, USA. Association for Computing Machinery.

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

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

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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. *arXiv preprint arXiv:1907.11692*.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". *Advances in neural information processing systems*, 30.

Mike Mintz, Steven Bills, Rion Snow, and Daniel 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*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. In *Advances in Neural Information Processing Systems*, volume 33, pages 21199–21212. Curran Associates, Inc.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. 2018. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152.

Mobashir Sadat and Cornelia Caragea. 2022a. Learning to infer from unlabeled data: A semi-supervised learning approach for robust natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4763–4776, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mobashir Sadat and Cornelia Caragea. 2022b. SciNLI: A corpus for natural language inference on scientific text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7399–7409, Dublin, Ireland. Association for Computational Linguistics.

Mobashir Sadat and Cornelia Caragea. 2024. MsciNLI: A diverse benchmark for scientific natural language inference. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Mexico City, Mexico. Association for Computational Linguistics.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of EMNLP*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. 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*, pages 235–243.

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.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. 2021. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International conference on learning representations*, volume 2.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021a. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021b. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Henry Zou and Cornelia Caragea. 2023. JointMatch: A unified approach for diverse and collaborative pseudo-labeling to semi-supervised text classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7290–7301, Singapore. Association for Computational Linguistics.

Class	#Annotated	Agreement
Contrasting	624	93.4%
Reasoning	624	82.4%
Entailment	624	80.1%
Neutral	624	94.6%
Overall	2,496	87.6%

Table 4: Number of manually annotated examples and the agreement rate between the gold labels and automatically assigned labels for each class.

A Details on Data Annotation

We randomly sample a small subset of examples balanced over the classes from the full SciNLI training set and ask three expert annotators to annotate the label based on only the available context in the two sentences in each example. A gold label is assigned to each example based on the majority consensus of the annotated labels. If the annotators cannot reach a majority consensus, no gold label is assigned. The examples for which the human annotated gold label match with the distantly supervised automatically assigned label are selected to be in the **human annotated training set**. We iteratively continue sampling examples from the SciNLI training set and annotating them until we have 2000 human annotated training examples balanced over the classes. The annotated data will be made available for academic research purposes at our GitHub link[2].

Agreement Rates We iteratively sample examples from SciNLI training set until we had at least 500 examples from each class in the human annotated training set. In total 2,496 examples are annotated among which 2,187 had a match between the gold label and the automatically assigned label based on distant supervision. We then randomly down-sample the classes with higher support to 500 examples to balance the datasets. The annotators achieved a Fleiss-k score of 0.68. The class-wise agreement rate can be seen in Table 4.

Annotator Details Undergraduate students were hired as annotators from the authors' institution. The annotators were trained for several iterations until their annotations reached a satisfactory agreement with the authors. The trained annotators then start their final annotations for constructing the human annotated training set. The compensation was set at the hourly rate of $15.

[2]https://github.com/msadat3/weighted_cotraining

B Implementation Details

We use the 'scibert-scivocab-cased' and the 'roberta-base' variants of SciBERT and RoBERTa, respectively for all our baselines and proposed methods. We use the huggingface[3] implementation for both models. The premise and hypothesis for each example are concatenated with a [SEP] token and the [CLS] token's representation is sent through a fully connected layer with softmax activation to predict the class. For all methods (baselines and proposed), the batch size is set at 64. We use an Adam optimizer (Kingma and Ba, 2014) to train the models and set the learning rate at $2e-5$. We use a learning rate of $2e-6$ for fine-tuning the models in the last step of our proposed approach (Step 3 in Section 4.2). For backtranslation, we use a transformer based sequence-to-sequence model[4]. The confidence threshold for DBST, FixMatch, and the base threshold for FlexMatch is set at 0.9. For the consistency regularization based algorithms such as FixMatch, FlexMatch, and SoftMatch, we use a ratio of $1:7$ between labeled and unlabeled data. The weight for the supervised loss for the consistency regularization methods is set at 0.8, and that of the unsupervised loss is set at 0.2. For both Co-teaching and Co-teaching+, we set the estimated noise rate, $\epsilon = 0.15$. For the DS, BT, FS, and the consistency regularization baselines, we train the classifiers for a maximum of 10 epochs. For the other experiments, we set the number of maximum epochs at 5. Early stopping is employed with a patience of 2. Macro F1 score on the development set is used as the stopping criteria for early stopping.

We use a single NVIDIA RTX A5000 GPU for the experiments involving a single model training (e.g., DBST, FixMatch etc.), and two NVIDIA RTX A5000 GPU for the dual model training experiments (e.g., co-training, weighted co-training, coteaching). Each co-training experiment takes ≈ 5 hours to complete.

C Robustness Analysis

We assess the robustness of our proposed approach by experimenting with out-of-domain (OOD) test sets from MSciNLI (Sadat and Caragea, 2024)

[3]https://huggingface.co/docs/transformers/index

[4]https://huggingface.co/Helsinki-NLP/opus-mt-en-ROMANCE

MODEL	HARDWARE	NETWORKS	SWE	SECURITY	NEURIPS	OVERALL
DS-LABELED (101K)	75.60 ± 0.8	72.70 ± 0.5	74.36 ± 0.3	74.99 ± 0.3	78.35 ± 1.0	75.19 ± 0.5
CO-TEACHING	75.06 ± 2.8	74.03 ± 0.7	74.07 ± 1.2	74.27 ± 1.1	77.77 ± 0.8	75.04 ± 1.2
WCT - CC	77.41 ± 0.5	76.24 ± 0.3	75.68 ± 0.4	75.34 ± 0.7	79.38 ± 0.1	76.80 ± 0.1
WCT - CV	77.32 ± 0.3	76.12 ± 0.3	76.44 ± 0.4	76.46 ± 0.7	79.97 ± 0.5	77.11 ± 0.1

Table 5: Domain-wise and overall Macro F1 scores (%) of DS-LABELED (101K) and CO-TEACHING baselines compared with our methods WCT - CC and WCT - CV trained using SCINLI and tested on MSCINLI.

<human>: Consider the following two sentences: Sentence1: <sentence1> Sentence2: <sentence2> Based on only the information available in these two sentences, which of the following options is true? a. Sentence1 generalizes, specifies or has an equivalent meaning with Sentence2. b. Sentence1 presents the reason, cause, or condition for the result or conclusion made Sentence2. c. Sentence2 mentions a comparison, criticism, juxtaposition, or a limitation of something said in Sentence1. "d. Sentence1 and Sentence2 are independent. <bot>:

Table 6: Prompt template used for our experiments with LLM. Here, <X> indicates a placeholder X which is replaced in the actual prompt.

PROMPT TYPE	CONTRASTING	REASONING	ENTAILMENT	NEUTRAL	MACRO F1
ZERO-SHOT	45.55	37.72	15.61	18.50	29.34
FEW-SHOT	57.70	37.18	51.51	63.90	52.57

Table 7: Class-wise F1 and overall Macro F1 of Zero-shot and Few-shot prompting with Llama-2-13b-chat for SCINLI test set.

that covers the following domains from computer science: “Hardware”, “Networks”, “Software & its Engineering”, “Security & Privacy”, and “NeurIPS” which is related to machine learning. Particularly, we evaluate the following models that are trained using the training set of SCINLI (that covers computational linguistics only) on the MSCINLI test set: DS-LABELED (101K), CO-TEACHING, WCT - CC, and WCT - CV. The domain-wise and overall Macro F1 of these methods for MSCINLI are reported in Table 5. We find the following:

Our proposed approach is robust in an OOD setting. As we can see, for most of the domains in MSCINLI, CO-TEACHING fails to show any improvement over DS-LABELED (101K). In contrast, WCT - CC outperforms DS-LABELED (101K) in all domains resulting in an overall increase of 1.61%. In addition, WCT - CV improves the OOD performance further over WCT - CC and shows an improvement of 1.92% in Macro F1 over the DS-LABELED (101K) baseline. These results illustrate that, our proposed approach not only improves in-domain performance, but also trains robust models that perform well for OOD datasets.

D LLMs for SciNLI

We experiment with *Llama-2-13b-chat-hf* (Touvron et al., 2023) for SCINLI. We design a prompt template that first presents the two sentences from each example, and asks a multiple choice question with the class definitions as the choices. The prompt template can be seen in Table 6. Note that we omitted the special tags needed to prompt the LLM (e.g., `[INST]`) in the Table for brevity. We perform experiments with the LLM in 2 settings:

- Zero-shot: no exemplars are shown to the LLM.
- Few-shot: one exemplar from each class is prepended to the prompt.

The results from these experiments can be seen in Table 7. As we can see, the FEW-SHOT performance is substantially higher than the ZERO-SHOT performance. Nevertheless, the Macro F1 of even the FEW-SHOT setting is only 52.57% illustrating that making use of LLMs for reducing the noise in the SCINLI trainign set is not viable approach.