Learning to Infer from Unlabeled Data: A Semi-Supervised Learning Approach for Robust Natural Language Inference

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

Abstract

Natural Language Inference (NLI) or Recognizing Textual Entailment (RTE) aims at predicting the relation between a pair of sentences (premise and hypothesis) as entailment, contradiction or semantic independence. Although deep learning models have shown promising performance for NLI in recent years, they rely on large scale expensive human-annotated datasets. Semi-supervised learning (SSL) is a popular technique for reducing the reliance on human annotation by leveraging unlabeled data for training. However, despite its substantial success on single sentence classification tasks where the challenge in making use of unlabeled data is to assign "good enough" pseudo-labels, for NLI tasks, the nature of unlabeled data is more complex: one of the sentences in the pair (usually the hypothesis) along with the class label are missing from the data and require human annotations, which makes SSL for NLI more challenging. In this paper, we propose a novel way to incorporate unlabeled data in SSL for NLI where we use a conditional language model, BART to generate the hypotheses for the unlabeled sentences (used as premises). Our experiments show that our SSL framework successfully exploits unlabeled data and substantially improves the performance of four NLI datasets in low-resource settings. We release our code at: https://github.com/msadat3/SSL_for_NLI.

1 Introduction

Natural Language Inference (NLI) or Recognizing Textual Entailment (RTE) is the task of predicting whether a hypothesis entails, contradicts or is neutral to a given premise. It is widely used as a benchmark for evaluating Natural Language Understanding (NLU) which plays a key role in many Natural Language Processing tasks such as text summarization, machine translation and sentiment analysis. In addition to serving as a benchmark for NLU, NLI has aided in improving the performance in downstream tasks such as fake news detection (Sadeghi et al., 2022) and fact verification (Martín et al., 2022).

In recent years, deep learning based approaches (Chen et al., 2016; Devlin et al., 2019; Liu et al., 2019) have shown promising performance for NLI due to their superior ability to extract deep semantic features. However, despite their success, one of the key challenges of the deep learning based models is that they require large scale human annotated datasets to perform well. Consequently, earlier NLI datasets such as SICK (Marelli et al., 2014) and RTE (Dagan et al., 2005), while being instrumental in the progress of NLI research, are not suitable for training these models due to their small size. To address this, large scale datasets such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and ANLI (Nie et al., 2020) have been proposed. The general annotation protocol for these datasets involves curating the premises from a pre-existing source and then employing human crowdworkers to write the hypotheses and assign their class labels. Given the large number of annotated samples needed for a dataset to be suitable for the deep learning models, their construction requires a significant amount of human effort. Consequently, creating new NLI datasets capturing unique linguistic properties of different domains/time/demography becomes cumbersome and in some cases impossible. Therefore, it is necessary to reduce the reliance on manually annotated data in training deep learning models for NLI.

To this end, we seek to harness unlabeled data by adopting semi-supervised learning (SSL) for low-resource NLI datasets which could potentially improve the performance without the necessity of any additional human effort. SSL is a widely used method for automatically assigning pseudo-labels to unlabeled data to incorporate them into model training (Xie et al., 2020b; Becker et al., 2013; Liu et al., 2021). However, unlike single sentence

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4763–4776

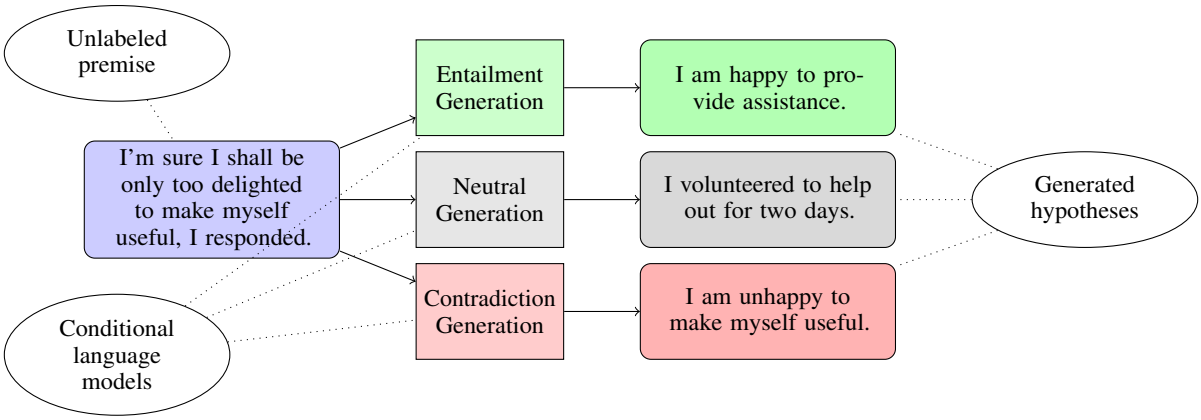

Figure 1: An example of synthetic hypotheses generation from an unlabeled premise.

classification tasks, where the challenge in making use of unlabeled data is to assign "good enough" pseudo-labels, as described above, the nature of unlabeled data is more complex for NLI: one of the sentences in the pair (usually the hypothesis) along with the class label are missing from the data and require human annotations. Therefore, in order to leverage unlabeled data for NLI, the unavailability of both hypotheses and class labels need to be tackled. Thus, employing SSL is much more challenging for NLI than other tasks. Potential approaches for addressing this challenge include selecting other unlabeled sentences as hypotheses randomly or identifying them based on similarity with the unlabeled premises. However, these approaches can either result in all neutral samples and fail to provide the necessary coverage to the other classes or can be computationally too expensive. As a result, the unavailability of hypotheses for unlabeled premises remains as the key bottleneck in exploring SSL for NLI.

In this paper, we propose to address this bottleneck with a novel approach for incorporating unlabeled data into training low-resource NLI models where we generate the hypotheses for the unlabeled premises with a state-of-the-art conditional language model, BART (Lewis et al., 2020). Specifically, for each unlabeled premise, one hypothesis is generated corresponding to each class in the labeled dataset. As a result, our proposed method guarantees the coverage of all classes during training. We can see an example of our hypothesis generation method in Figure 1. Then, we develop an SSL framework based on our proposed strategy for leveraging unlabeled data for NLI where we adopt iterative self-training to gradually accumulate useful pseudo-labeled examples for training. Our experiments show that the proposed SSL framework can improve both in-domain and out-of-domain performance on four NLI datasets in low-resource settings. Our contributions are as follows:

- We propose a novel approach for incorporating unlabeled data in training models on low-resource NLI datasets. To our knowledge, we propose the first ever SSL framework for NLI based on this approach.
- We thoroughly evaluate our proposed framework using four NLI datasets in low-resource settings and show that both in-domain and out-of-domain performances improve substantially illustrating that we successfully leverage unlabeled data for NLI.
- We perform a comprehensive analysis of our framework to understand whether it can improve the robustness of NLI models. Our findings suggest that the overall robustness of the models improves by a considerable margin when they are trained using our framework.

2 A New SSL Framework for NLI

This section details our proposed SSL for NLI framework which consists of two components: hypothesis generation and self-training. An overview of our framework can be seen in Algorithm 1.

Problem Formulation Let $D^l = \{(p_i^l, h_i^l, y_i^l)\}_{i=1,\ldots,n}$ be our labeled set of size n where p_i^l, h_i^l, and y_i^l represent the premise, hypothesis and label of sample i and let C be the set of classes in D^l. Given D^l and a large set of unlabeled premises $\{p_i^u\}_{i=1,\ldots,m}$ of size m, our SSL framework has two objectives. First, it

creates a synthetically annotated dataset D^{syn} using our proposed hypothesis generation method. Next, it employs self-training to iteratively assign pseudo-labels and select examples from D^{syn} which are of high quality and puts them in our pseudo-labeled set D^p to be used for training in addition to the labeled dataset D^l.

2.1 Hypothesis Generation

In order to harness unlabeled data for NLI, for each unlabeled premise, we propose to generate one hypothesis corresponding to each class using a conditional language model to ensure both class coverage and computational efficiency (compared to exhaustively searching for related sentence pairs in a large corpus). For our experiments, we use BART (Lewis et al., 2020) as the conditional language model which was pre-trained by learning to reconstruct text corrupted in different manners such as token masking, token deletion, sentence permutation etc. However, any conditional language model that can generate text can be used in its place in our proposed framework.

For each class, we first select its corresponding premise-hypothesis pairs from D^l and then fine-tune the pre-trained BART model using cross-entropy loss where the premises from the selected pairs are used as the source texts and their hypotheses as the targets. We denote the fine-tuned model for each class c as $BART^c$. The synthetically labeled dataset D^{syn} is then created as follows:

$$D^{syn} = [\{(p_i^{syn} = p_i^u, h_i^{syn} = BART^c(p_i^u), y_i^{syn} = c)\}_{i=1,..m}]_{c \in C} \quad (1)$$

That is, if there are three classes — entailment, contradiction and neutral in the dataset, each unlabeled premise p_i^u is paired with three synthetic hypotheses generated with the BART models corresponding to these classes. We also assign the respective class of the model used to generate the hypothesis as the initial synthetic label for each premise-hypothesis pair.

2.2 Self-training

Self-training is an iterative algorithm which assigns pseudo-labels to unlabeled data at each iteration and selects a subset of them based on some quality assurance measures to be used as additional training data in the subsequent iterations.

Due to the sole reliance on the generative models for assigning the synthetic labels for the premise-hypothesis pairs, our synthetically annotated dataset D^{syn} can contain harmful noise and significantly affect classification performance if they are directly used as training data without any quality assurance measures. Therefore, we make use of self-training to iteratively curate high quality samples from D^{syn} and put them into our pseudo-labeled set D^p. To this end, at each iteration k, first, we train a classifier θ_k^{clf} on a combination of labeled training set D^l and pseudo-labeled training set D^p if it is non-empty (i.e., if $k > 1$). Next, to provide equal coverage to all classes, we randomly sample a balanced subset of S examples from our synthetically labeled set D^{syn} for pseudo-labeling. We denote this sampled synthetic subset as D_k^{syn}. The pseudo-label y_i^p for each sample i in this subset is then assigned as follows:

$$y_i^p = \arg\max_{c \in C} \theta_k^{clf}(c|p_i^{syn}, h_i^{syn}) \quad (2)$$

Here, θ_k^{clf} predicts the probability distribution of each sample over C classes conditioned on the premise p_i^{syn} and hypothesis h_i^{syn}.

Quality Assurance To ensure data quality, we apply two filters to our pseudo-labeled data. First, following the traditional approach for quality assurance of pseudo-labeled data in SSL, the examples for which the pseudo-label is predicted with a confidence score lower than a pre-defined threshold τ are filtered out. Next, since we have two sets of labels for the unlabeled samples, i.e., synthetically assigned label y_i^{syn} (assigned through the BART model) and predicted pseudo-label y_i^p (assigned by the classifier) for each sample i, we enforce their consistency as an additional data quality measure. Therefore, we filter out the examples for which the predicted pseudo-label y_i^p and the initially assigned synthetic label y_i^{syn} do not match. More formally, based on our two filters, we select the pseudo-labeled examples for iteration k as follows:

$$D_k^p = \{(p_i^{syn}, h_i^{syn}, y_i^{syn}) : y_i^{syn} = y_i^p, \theta_k^{clf}(y_i^p|p_i^{syn}, h_i^{syn}) \geq \tau\}_{i=1,..S} \quad (3)$$

After selecting the pseudo-labeled examples from the current iteration, we add them to D^p, our pseudo-labeled set which is accumulated across the iterations:

$$D^p = D^p \cup D_k^p \quad (4)$$

The self-training iterations are run until D^{syn} is empty, i.e., we run out of synthetic examples or

Algorithm 1 Semi-supervised NLI

Require:
Labeled samples $D^l = \{(p_i^l, h_i^l, y_i^l)\}_{i=1,..n}$;
Unlabeled premises $P^u = \{(p_i^u)\}_{i=1,..m}$;
Initialize synthetically labeled set, $D^{syn} \leftarrow \emptyset$
for each $c \in C$ **do**: ▷ C is the set of classes in D^l
  Fine-tune BART to generate hypotheses conditioned on the premises for class c on corresponding examples from D^l
  Generate hypotheses for class c for each unlabeled premise p_i^u using the fine-tuned BART model
  Add each unlabeled premise, generated hypothesis and class c as a synthetically labeled example to D^{syn}
end for
Initialize pseudo-labeled set, $D^p \leftarrow \emptyset$
Initialize current iteration $k \leftarrow 0$
while $k \neq K$ **and** $D^{syn} \neq \emptyset$ **do** ▷ K is a pre-defined maximum iteration.
  Train classification model, θ_k^{clf} on a combination of D^l and D^p
  $D_k^{syn} \leftarrow S$ examples randomly sampled from D^{syn}
  Predict the labels of the samples in D_k^{syn} and get the model confidences
  Apply filters on the samples in D_k^{syn} using Equation 3 and put the selected samples in D_k^p
  Add new pseudo-labeled examples to the pseudo-labeled set, $D^p \leftarrow D^p \cup D_k^p$
  Remove new pseudo-labeled examples from the synthetically labeled set, $D^{syn} \leftarrow D^{syn} \setminus D_k^p$
  Increment iteration, $k \leftarrow k + 1$
end while

a pre-defined number of iterations, K is reached. The model from the iteration showing the best development score is used to evaluate the test sets.

Since we follow the traditional practice of self-training by training the classification model on a combination of labeled and pseudo-labeled data in each iteration, we denote this self-training setup as **VST** which stands for Vanilla Self-Training.

De-biased Self-training Given that the hypotheses in our pseudo-labeled datasets are machine generated, we hypothesize that they can introduce some unnecessary bias to the classification models despite our strict quality assurance measures for curating them. Therefore, we explore a self-training setup where we first train the classification model on pseudo-labeled data and then train it on human annotated labeled data in each iteration to reduce the introduced bias. This model is denoted by **DBST** or de-biased self-training.

Noised Self-training Inspired by recent work on self-training for image classification (Xie et al., 2020b) which shows that perturbing an input image by introducing controlled noise/data augmentation after assigning the pseudo-labels can significantly improve the classification performance, we explore perturbed / noised versions of our self-training methods. Specifically, after selecting the pseudo-labeled data using Equation 3, we employ back-translation (Yu et al., 2018) to replace the original premise and hypothesis in each sample with their augmented versions. The perturbed versions of our vanilla and de-biased self-training setups are denoted as **VST + N** and **DBST + N**.

3 Datasets

We use the following datasets for our experiments.

SICK (Marelli et al., 2014) SICK is a dataset containing 4,500 sentence pairs in its training set. We use the 8K ImageFlickr data set[1] to extract unlabeled premises. This dataset was one of the sources for deriving SICK.

RTE (Wang et al., 2018) This dataset contains $\approx 2.5K$ examples and was created by combining the RTE1 (Dagan et al., 2005), RTE2 (Haim et al., 2006), RTE3 (Giampiccolo et al., 2007) and RTE5 (Bentivogli et al., 2009) datasets. The premises for RTE were extracted from Wikipedia and news sources. Therefore, we also extract sentences from Wikipedia and the CNNDM (Nallapati et al., 2016) dataset to be used as unlabeled premises. Since, the test set of RTE is not publicly available, we use its development set as the test set and randomly sample a small subset from the training set to be used as the development set.

MNLI - 6K (Williams et al., 2018) To simulate a low-resource environment, we randomly sample 6,000 examples from the training set of MNLI. From the rest of the training set, we select the premises which do not occur in the sampled 6,000 examples to be used as unlabeled data. Similar to RTE, we use the development set of MNLI as the test set and sample a small subset of examples from the training set to be used as development set.

SNLI - 6K (Bowman et al., 2015) We created SNLI - 6K in a similar fashion as MNLI - 6K.

[1] https://www.kaggle.com/adityajn105/flickr8k/activity

4 Baselines

Given that we are the first to explore NLI in a low-resource setting, there are no existing methods that we can use as baselines, which are specifically tailored for NLI when human annotated data is limited. Thus, most of our baselines are adopted from methods which have been successful for other NLP tasks in low-resource settings. Specifically, we compare the performance of our SSL framework with three types of baselines.

BERT (Devlin et al., 2019): This is a baseline in which a BERT model is fine-tuned only on the available human-annotated data.

DATA AUGMENTATION (DA) We compare the performance of our SSL framework with three data augmentation baselines. We augment both premise and hypothesis in each example in the labeled training set and combine the original labeled set with its augmented version. In other words, the size of the labeled dataset is doubled by adding an augmented version of each original example. We use the following data augmentation methods.

(a) BACK-TRANSLATION (BT) (Yu et al., 2018) The pair (premise and hypothesis) in each example in the labeled training set is translated to French and then translated back to English using machine translation models to get their paraphrased versions.

(b) SYNONYM REPLACEMENT (SR) (Kolomiyets et al., 2011) Randomly chosen tokens from both premise and hypothesis are replaced with their synonyms using WordNet (Miller, 1995).

(c) CONDITIONAL MASKED LANGUAGE MODELING (C-MLM) Inspired by CBERT (Wu et al., 2019) where some randomly chosen positions in the input text are masked and a conditional masked language model (C-MLM) is used to predict the tokens in those positions based on both the context and the class label to get the augmented versions of the original input text, we formulate a similar DA method for NLI. Specifically, the premise, hypothesis and label of each sample is combined using class-specific templates that we describe in Appendix A.1. We then randomly mask a subset of common tokens in both premise and hypothesis and use the C-MLM model to predict the tokens at those positions to get their augmented versions.s

UNLABELED DATA EXPLOITATION: We use two baselines that leverage unlabeled data using methods different from our SSL framework to evaluate the efficacy of our proposed method for leveraging unlabeled data.

(a) UNSUPERVISED DATA AUGMENTATION (UDA) (Xie et al., 2020a) This is a semi-supervised learning framework where traditional cross-entropy loss on the labeled data is combined with a consistency loss which is aimed at minimizing the distance between the probability distributions predicted by the model for an unlabeled sample and an augmented version of the unlabeled sample. The quality assurance of unlabeled data is done during training based on a pre-defined confidence threshold. We use back-translation to augment both premise and hypotheses of each unlabeled sample.

(b) SELF-TRAINING WITH RANDOM HYPOTHESIS (ST - RH) This is a self-training approach which is similar to DBST except it uses randomly chosen sentences as the hypotheses for unlabeled premises instead of synthetically generating them.

5 Main Experiments & Results

Our main experiments and results are described in this section. We run each of our self-training and baseline experiments three times and report the average and standard deviation of their Macro F1 scores in Table 1. Our implementation details can be found in Appendix B.

DBST vs BERT To understand the effectiveness of our SSL framework, we compare the performance of our DBST model with the baseline BERT model. The results show that DBST substantially improves the performance over BERT for all four datasets. For example, we can see an increase of 7.08% and 3.97% in Macro F1 by DBST over BERT on RTE and MNLI-6K$_{mm}$, respectively. Our qualitative analysis of the synthetically-labeled data suggests that the BART models used in our framework are able to generate meaningful hypotheses for each premise (see Table 5 in Appendix C for examples). Furthermore, since we generate one hypothesis for each unlabeled premise corresponding to each class, our proposed method guarantees the coverage of all classes. Consequently, we are able to expose the models to high quality premise-hypothesis pairs derived from unlabeled data which enables DBST to show consistent improvements proving that our SSL framework successfully harnesses unlabeled data for NLI.

DBST vs DA Baselines Next, we evaluate whether our SSL framework can improve the per-

Approach	RTE	SICK	SNLI-6K	MNLI-6K$_m$	MNLI-6K$_{mm}$
BERT	60.90 ± 2.99	84.63 ± 0.66	78.47 ± 0.26	68.76 ± 0.56	70.05 ± 0.73
BT	62.27 ± 1.69	84.48 ± 0.47	78.39 ± 0.35	69.52 ± 0.51	71.20 ± 0.26
SR	60.05 ± 1.17	84.11 ± 0.75	78.33 ± 0.22	68.29 ± 0.11	68.35 ± 0.23
C-MLM	62.42 ± 0.19	84.74 ± 0.36	78.85 ± 0.29	69.38 ± 0.56	70.79 ± 0.32
ST - RH	63.28 ± 0.85	84.93 ± 0.18	78.31 ± 0.52	68.58 ± 0.36	70.21 ± 0.64
UDA	58.31 ± 2.17	84.40 ± 0.57	78.78 ± 0.61	69.46 ± 0.08	71.05 ± 0.42
VST	64.18 ± 2.10	85.02 ± 0.32	$79.46^* \pm 0.25$	$71.57^* \pm 0.34$	$73.03^* \pm 0.47$
+N	66.31 ± 2.66	84.64 ± 0.44	79.01 ± 0.07	$71.13^* \pm 0.62$	$72.46^* \pm 0.14$
DBST	$67.98^* \pm 1.78$	$\mathbf{85.77 \pm 0.06}$	$80.04^* \pm 0.35$	$72.22^* \pm 0.61$	$74.02^* \pm 0.29$
+N	$\mathbf{68.32^* \pm 2.03}$	85.51 ± 0.33	$\mathbf{80.31^* \pm 0.21}$	$\mathbf{72.99^* \pm 0.10}$	$\mathbf{74.48^* \pm 0.22}$

Table 1: The average and standard deviation of the Macro F1 (%) from three different runs of different approaches on our selected datasets. Here, RH and N stands for Random Hypothesis and Noise, respectively. Best scores are in bold. An asterisk indicates a statistically significant difference with BERT according to a paired T-test with $\alpha = 0.05$.

formance over prior methods which handle low-resource scenarios without leveraging unlabeled data. To this end, we compare the performance of the DBST model with our DA baselines. We can see in Table 1 that DBST outperforms all our DA baselines by a significant margin. Moreover, in general, the DA methods only minimally improve the performance over BERT. For example, the best performing DA method for SICK, C-MLM shows an improvement of 0.11% in Macro F1 score over BERT. Similarly, the improvement shown by BT for MNLI-6K$_m$ is only 0.76%. These results indicate that DA methods fail to introduce enough diversity in the augmented sentences which can help improve the models' ability in recognizing semantic relations between premise-hypothesis pairs. In contrast, harnessing unlabeled data with our SSL framework provides us with linguistically diverse patterns beneficial in improving the performance.

DBST vs Alternative Methods for Leveraging Unlabeled Data We now compare the performance of the DBST model with ST - RH and UDA to evaluate our proposed strategy for leveraging unlabeled data for NLI. Both of these models aims at exploiting unlabeled data with a strategy different than ours. ST - RH randomly chooses other unlabeled sentences for each premise instead of synthetically generating them while UDA employs an additional consistency loss term between the original and perturbed version of unlabeled data. The results in Table 1 show that DBST outperforms both of these models. Although in UDA, we use the same synthetic hypothesis generation strategy as in DBST, the results indicate that simply improving the consistency between the predictions made for unlabeled data and their perturbed version is not enough to improve the performance. Comparing DBST with ST - RH, we find that almost all unlabeled examples are assigned a NEUTRAL pseudo-label in ST - RH as we hypothesized. Therefore, unlike our proposed strategy for leveraging unlabeled data, randomly choosing unlabeled sentences as the hypotheses fails to provide enough coverage to all relevant classes which results in self-training not being able to perform well.

Vanilla vs De-biased Comparing the performance of VST and DBST in Table 1, we can see that DBST performs better than VST in all four datasets. This corroborates our notion that synthetically generated hypotheses while being useful in improving the performance, can introduce some bias to the model. When we continue training the model on only human-annotated data, this bias gets reduced which leads to better performance.

Noised Model Evaluation To evaluate whether introducing noise to the pseudo-labeled data by back-translating both premise and hypothesis can further improve the performance, we compare VST and DBST with their noisy counterparts in Table 1. We can see that VST + N shows a lower performance than VST. On the other hand, adding noise to DBST pushes the performance further. For example, on MNLI - 6K$_m$, we see an improvement of 0.77. DBST + N also improves over DBST for RTE and SNLI but the improvement margin is smaller. This discrepancy in trends shown by the VST + N and DBST + N models indicate that when we noise the input text (i.e., replace it with its augmented versions), in addition to examples which help the model become more robust, some mis-labeled examples (i.e., harmful for the model), are introduced. The additional fine-tuning step in

Model	Train \ Test	RTE	SICK	SNLI-6K	MNLI-6K$_m$	MNLI-6K$_{mm}$
BERT	**RTE**	-	46.08 ± 11.51	60.80 ± 3.95	61.58 ± 2.70	35.75 ± 0.49
DBST + N		-	$\mathbf{54.53 \pm 5.29}$	$\mathbf{69.05 \pm 1.33}$	$\mathbf{68.39 \pm 0.69}$	$\mathbf{39.48 \pm 0.50}$
BERT	**SICK**	49.30 ± 6.25	-	37.10 ± 2.74	43.25 ± 6.96	45.80 ± 9.36
DBST + N		48.22 ± 3.32	-	$\mathbf{42.19 \pm 1.13}$	$\mathbf{45.83 \pm 2.84}$	$\mathbf{50.30 \pm 3.66}$
BERT	**SNLI-6K**	56.72 ± 0.04	44.34 ± 1.55	-	54.92 ± 1.20	57.10 ± 1.44
DBST + N		$\mathbf{58.21 \pm 0.55}$	$\mathbf{47.15 \pm 1.43}$	-	$\mathbf{59.00 \pm 0.19}$	$\mathbf{61.88 \pm 0.43}$
BERT	**MNLI-6K**	64.51 ± 1.70	60.85 ± 1.75	61.35 ± 2.40	-	-
DBST + N		$\mathbf{64.93 \pm 0.84}$	54.36 ± 5.62	$\mathbf{66.44 \pm 1.25}$	-	-

Table 2: Out-of-domain (OOD) Macro F1 (%) scores of the DBST + N models compared with baseline BERT models. The main diagonal is kept blank because it corresponds to in-domain (ID) performances.

		Competence Test			Distraction Test						Noise Test	
		Antonymy		Numerical	Word Overlap		Negation		Length Mismatch		Spelling Error	
Model	Dataset	Mat	Mis	Reasoning	Mat	Mis	Mat	Mis	Mat	Mis	Mat	Mis
BERT	**RTE**	8.52	6.01	38.67	65.06	65.26	64.08	64.25	61.78	62.73	66.67	66.50
DBST + N		1.25	0.89	34.05	66.38	67.25	68.04	69.20	65.53	65.14	67.03	66.96
BERT	**SICK**	55.26	56.50	25.80	34.48	34.36	39.67	41.57	44.42	46.62	42.09	42.49
DBST + N		28.85	28.98	25.05	38.02	39.17	43. 83	46.71	47.84	51.16	43.96	44.31
BERT	**SNLI-6K**	18.38	16.18	28.83	42.66	44.14	51.12	52.09	54.22	55.90	49.09	49.31
DBST + N		7.06	5.89	34.56	50.92	52.87	56.55	58.56	58.00	59.70	53.28	53.43
BERT	**MNLI-6K**	8.03	7.69	29.23	43.66	44.87	62.25	64.02	66.29	68.75	64.80	65.32
DBST + N		17.72	15.43	30.32	46.32	47.36	59.06	60.43	70.43	72.71	68.35	68.65

Table 3: Stress test accuracies (%) of the baseline BERT and DBST + N trained on different datasets.

DBST + N on only the original clean data reduces the harmful noise while retaining the robustness introduced by the other useful perturbed examples. As a result, we see an improvement in performance by DBST + N over DBST while the performance of VST + N declines from VST due to a lack of additional fine-tuning step on clean data.

6 Out-of-domain Results

To assess whether our SSL framework can improve the out-of-domain (OOD) performance, we evaluate DBST + N and BERT models trained on each dataset using test sets of other datasets. It should be noted that we do not perform any additional training of the models for evaluating the OOD performance. We simply use the models already trained on labeled and/or unlabeled data on the training set of a particular dataset and test them on other datasets. The OOD results are reported in Table 2.

We can see that in general, DBST + N shows a significantly higher OOD performance than BERT. For example, the performance of DBST + N trained on RTE shows an 8.45% higher Macro F1 score than BERT when they are tested on SICK. Therefore, harnessing unlabeled data with our SSL framework can improve the performance on new domains without any additional annotation effort.

7 Analysis

Robustness Analysis From our experiments and results, it is evident that our SSL framework is effective in improving both in-domain and out-of-domain performances. However, it is not clear if SSL can train more robust NLI models. Therefore, we study the robustness of our models on NLI stress test (Naik et al., 2018). The stress test was created based on the weaknesses of NLI models in various aspects such as word overlap — models tend to predict a sample to be entailment if there is a large overlap in words even if they are unrelated, negation — presence of a negation word such as 'no' causes the model to predict the sample as contradiction etc. In total, 11 different tests are carried out which are divided into three parts: competence, distraction, and noise. We compare the stress test results for the baseline BERT model and DBST + N model in Table 3.

The DBST + N models show better performance than the baseline BERT model on both distraction and noise tests. Therefore, our SSL framework reduces the vulnerability of the models in being distracted by shallow features such as word overlap,

Approach	MNLI-6K$_m$	MNLI-6K$_{mm}$
DBST + N	73.11	74.66
DBST + N$_{SB}$	72.59	74.54
DBST + N$_{CM}$	72.58	74.31

Table 4: The Macro F1 (%) score of DBST + N compared with DBST + N$_{SB}$ and DBST + N$_{CM}$.

negation and mismatch in lengths and improves their ability in making real inferential decisions. Moreover, the models trained with SSL are less prone to making wrong predictions due to the noise caused by spelling mistakes which also helps in improving the robustness.

However, both BERT and DBST + N models fail to show meaningful performance for the competence tests. In general, the accuracy for both models is lower than the random baseline (50% for RTE since it is a 2-way classification and 33.33% for the other datasets which are 3-way classifications). This indicates that while our proposed SSL framework improves the overall robustness of the models, it cannot address the models' inability to recognize the contradicting relations caused by antonyms and their weakness in reasoning over numeric tokens. Therefore, in our future work we aim at incorporating methods which can make the models more robust in recognizing the antonyms and more capable to numeric reasoning.

Single BART In order to understand the necessity of using separate BART models for each class, we perform an experiment on MNLI-6K where we employ a single BART model to generate the hypotheses for all classes in our DBST + N approach. To this end, we append the label at the end of each premise in the labeled training set to be used as the source text and use their hypotheses as the target text to fine-tune a BART model. Similarly, to generate the hypotheses for different classes, we append the class labels at the end of the unlabeled premises before using them as the input to the fine-tuned BART model.

A comparison between DBST + N and DBST + N$_{SB}$ on MNLI-6K can be seen in Table 4. Clearly, the performance of the DBST + N is superior illustrating the necessity of class specific conditional language models. Our qualitative analysis of the generated hypotheses by the single BART model reveals that in many cases, it fails to distinguish among the appended labels at the end of premises and ends up generating the same hypothesis for all classes. In contrast, class-specific BART models are able to generate more label-relevant hypotheses by focusing on one class at a time.

Only Confidence Masking In our SSL framework, we use two quality assurance filters in choosing pseudo-labeled data (see Equation 3): a) confidence masking — whether the confidence for the predicted pseudo-label is above a pre-defined threshold, b) label consistency — whether the predicted pseudo-label and the synthetically assigned label match. However, in general, SSL only uses confidence masking filter for quality assurance. Therefore, to evaluate the necessity of our label consistency filter, we experiment with a version of DBST + N where we only use the confidence masking filter on MNLI-6K. We denote this approach DBST + N$_{CM}$. We can see in Table 4 that there is a drop in performance by this model compared to the DBST + N model. Therefore, using the additional filter based on label consistency is beneficial in ensuring data quality.

8 Related Work

Semi-supervised Learning (SSL) SSL has gained a lot of attention in the research community for exploiting unlabeled data to further boost the performance without any additional manual annotation effort. In general, techniques based on semi-supervised learning involve predicting the labels (hard or soft) of unlabeled data and using them in some manner for training in addition to labeled data. For example, consistency regularization (Sajjadi et al., 2016; Laine and Aila, 2017; Tarvainen and Valpola, 2017; Xie et al., 2019) aims at improving model robustness by introducing an additional loss term to minimize the distance between the probability distribution predicted by the model for an unlabeled sample and its perturbed version. Self-training is a form of SSL that assigns pseudo-labels to unlabeled samples to be used for training in addition to labeled data. Self-training has seen wide applications in various NLP and machine learning tasks. To name a few, it has been used to boost the performance in image classification (Yalniz et al., 2019; Xie et al., 2020b), sentiment analysis (Becker et al., 2013), conversation summarization (Chen and Yang, 2021), domain adaptation (Zou et al., 2018; Kumar et al., 2020a; Liu et al., 2021) and few-shot text classification (Mukherjee and Awadallah, 2020). However, to date, none of these SSL methods have been used to improve the performance in low-resource NLI.

Data Augmentation (DA) DA is also a popular technique for automatically increasing the amount of labeled data used for training. However, in general, unlike semi-supervised learning, unlabeled data is not utilized in data augmentation. Rather, the labeled samples are perturbed in different manners to create variations of the same input. Simpler data augmentation approaches use rule-based methods such as synonym replacement (Zhang et al., 2015), random swap/deletion/insertion of tokens (Wei and Zou, 2019) etc. More complex methods use different generative models to synthesize a new version of the input text using variational autoencoders (Kingma and Welling, 2013), pre-trained language models (Kumar et al., 2020b) and back-translation (Yu et al., 2018). Similar to SSL, DA is yet to be explored in the context of NLI.

Synthetic Data Generation With the advancement of text generation models in recent years (Radford et al., 2019; Brown et al., 2020; Lewis et al., 2020), researchers have started to adopt synthetic text generation as a data augmentation strategy for various tasks such as commonsense reasoning (Yang et al., 2020), sentence classification (Anaby-Tavor et al., 2020) and question answering (Puri et al., 2020). More recently, synthetic text generation methods have also been employed to remove spurious correlations in human annotated data for fact verification (Lee et al., 2021) and NLI (Wu et al., 2022). However, to our knowledge, no prior work aims at tackling low-resource NLI scenarios nor do they address the unavailability of hypotheses for unlabeled premises by using synthetic data generation methods.

9 Conclusion & Future Work

In this paper, we propose a novel method for leveraging unlabeled data for NLI in low-resource settings to improve the performance without the necessity of additional human annotation effort. We develop an SSL framework based on this proposed method for NLI which substantially improves both in-domain and out-of-domain performance of four NLI datasets in low resource scenarios illustrating that our SSL framework considerably improves the generalization capability of the models by harnessing unlabeled data. Our results indicate that we successfully address the key bottleneck in exploring SSL for NLI, i.e., the unavailability of hypothesis for unlabeled premises. Our future work will include developing methods which can improve the robustness of the models further.

10 Limitations

In this work, we present an SSL framework based on a novel method to leverage unlabeled data for NLI to improve the performance without the necessity of additional human annotation effort. Through extensive experiments, we show that our proposed method successfully exploits unlabeled data and significantly improves the performance for low-resource NLI. However, a limitation in our approach is that there is no intrinsic evaluation strategy for assessing the quality of the synthetically generated hypotheses. We can potentially evaluate their quality using some held-out human annotated set. However, a generated hypothesis does not need to have a high overlap with a reference hypothesis in order to entail, contradict or be neutral to the premise. As a result, existing evaluation metrics for assessing the quality of generated text such as perplexity, ROUGE, BLEU are not suitable for evaluating hypothesis generation models. Consequently, we are only able to perform extrinsic evaluation. That is, we can only estimate the quality of the generated hypotheses by checking if the classification performance improves when we use them in our SSL framework. An intrinsic evaluation strategy could potentially help further improve the performance.

Acknowledgements

This research is supported in part by NSF CAREER award #1802358, NSF CRI award #1823292, 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 AWS for computational resources. We also thank our anonymous reviewers for their constructive feedback.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Lee Becker, George Erhart, David Skiba, and Valentine Matula. 2013. AVAYA: Sentiment analysis on Twit-

ter 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.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 271–276, Portland, Oregon, USA. Association for Computational Linguistics.

Ananya Kumar, Tengyu Ma, and Percy Liang. 2020a. Understanding self-training for gradual domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5468–5479. PMLR.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020b. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

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.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hong Liu, Jianmin Wang, and Mingsheng Long. 2021. Cycle self-training for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 34, pages 22968–22981. Curran Associates, Inc.

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*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Alejandro Martín, Javier Huertas-Tato, Álvaro Huertas-García, Guillermo Villar-Rodríguez, and David Camacho. 2022. Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference. *Knowledge-Based Systems*, 251:109265.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

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.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

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.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

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

Fariba Sadeghi, Amir Jalaly Bidgoly, and Hossein Amirkhani. 2022. Fake news detection on social media using a natural language inference approach. *Multimedia Tools and Applications*, pages 1–21.

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.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International conference on computational science*, pages 84–95. Springer.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. 2022. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020b. Self-training with noisy student improves imagenet classification. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10687–10698.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. 2019. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

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.

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

Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

A Details about Baselines

A.1 Conditional Masked Language Modeling (C-MLM)

We use the following templates for each class to combine the premise and hypothesis of each sample in the labeled dataset along with their label information:

- ENTAILMENT: <*Premise*> implies <*Hypothesis*>.
- CONTRADICTION: <*Premise*> contradicts <*Hypothesis*>.
- NEUTRAL: <*Premise*> neither implies nor contradicts <*Hypothesis*>.

We mask half of the common tokens in each premise-hypothesis pair. For example, if there are 2 common tokens between a premise and its hypothesis, only one of them selected randomly and masked in both sentences.

After the tokens are predicted, we perform a post-processing step to ensure that the masked positions where the original premise and hypothesis had the same token also get replaced with the same predicted token. This was done to reduce the possibility of accidentally changing the label.

A.2 Filter for ST with Randomly Chosen Hypothesis

Since, there is no synthetically assigned label for examples when the hypothesis is selected using random sampling, we could not use the label consistency filter for our ST - RH baseline. Specifically, equation 3 is updated as follows for this model.

$$D_k^p = \{(p_i^u, h_i^u, y_i^p) : \theta_k^{clf}(y_i^p | p_i^u, h_i^u) \geq \tau\}_{i=1,..S} \quad (5)$$

B Implementation Details

Our implementation details can be divided into two parts: hypothesis generation and self-training.

B.1 Hypothesis Generation

We implement our hypothesis generation module using the huggingface transformers[2] library. Specifically, we choose 'bart-large' as our hypothesis generation model for all our datasets. We fine-tune each model for 30 epochs with a cross-entropy loss. We use the AdamW optimizer (Loshchilov and Hutter, 2018) with an initial learning rate of $3e-5$ and a batch size is set at 64. After the BART models are trained, we generate the hypotheses by using the unlabeled premises as the inputs. We employ top-k sampling with $k = 10$ and temperature scaling with temperature = 2.0 to introduce diversity to the generated hypotheses.

Each hypothesis generation model was trained in $\approx$ 1 hour using a single NVIDIA RTX A5000 GPU. It took $\approx$ 2 hours to generate the synthetic hypotheses for each dataset using the same GPU.

B.2 Self-training

At each iteration, we fine-tune a pre-trained 'bert-base-cased' (Devlin et al., 2019) model as our classifier using the huggingface transformers library where we concatenate the premise and hypothesis with a `[SEP]` token between them to be used as the input. We then project the hidden representation of the `[CLS]` token with a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times |C|}$ to get the classification output. Here, C is the set of classes. We fine-tune the model for 10 epochs with a cross-entropy loss using the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of $2e-5$. We employ early stopping with a patience size 2 where we use the Macro F1 score of the development set as the stopping criteria. The batch size is set at 64.

We set the highest number of iterations, $K = 100$ for all our self-training experiments. However, we stop the experiments if the development score does not improve for 10 consecutive iterations. The confidence threshold, τ for SICK, MNLI-6K and SNLI-6K is set at 0.9. For, RTE, we set $\tau = 0.7$ because higher threshold for this dataset results in very few pseudo-labeled examples getting selected. The number of synthetically labeled data to be sampled, S at each iteration is set to $\approx 0.75 * n$ where n is the size of each labeled dataset. For our noisy models, we use a transformer based model[3] for back-translating both premise and hypothesis.

The self-training models took $\approx 12-24$ hours to finish running on a single NVIDIA RTX A5000 GPU.

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

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

Unlabeled Premise	Synthetic Hypothesis	Synthetic Label
Started the moment I got the wire.	It began the moment I got the wire.	ENTAILMENT
Started the moment I got the wire.	It started to happen after I pulled the wire from the car.	NEUTRAL
Started the moment I got the wire.	It started before I got the wire.	CONTRADICTION
He consolidated power at Thebes during his short reign.	Thebes came to control power during his short reign.	ENTAILMENT
He consolidated power at Thebes during his short reign.	He was a megalomaniac obsessed with power, acquiring vast estates.	NEUTRAL
He consolidated power at Thebes during his short reign.	He had no power at Thebes.	CONTRADICTION
She was born in North Carolina and went to high school in Westbury, N.Y.	Westbury, N.Y, is where she was born, and graduated from high school.	ENTAILMENT
She was born in North Carolina and went to high school in Westbury, N.Y.	Her parents did not approve of her choice of education.	NEUTRAL
She was born in North Carolina and went to high school in Westbury, N.Y.	She is from New Jersey, where she attended college.	CONTRADICTION

Table 5: Examples from the synthetic dataset generated for MNLI-6K.

B.3 Out-of-domain Performance Evaluation for RTE

RTE is a 2-way classification dataset (ENTAILMENT and NOT_ENTAILMENT) whereas the other datasets have 3 classes (ENTAILMENT, CONTRADICTION and NEUTRAL). Thus, for out-of-domain performance evaluation of the models trained with RTE, we convert the CONTRADICTION and NEUTRAL class labels of the other datasets to NOT_ENTAILMENT. We follow the same method for evaluating the models trained on RTE on the stress set. Similarly, when we evaluate the models trained on other datasets on RTE (as the OOD dataset), we convert the predicted CONTRADICTION and NEUTRAL label to NOT_ENTAILMENT.

C Examples from Synthetically Generated Data

Table 5 shows a few examples from the synthetically labeled dataset for MNLI - 6K. We can see that the BART models are able to generate meaningful hypotheses for each unlabeled premise and provide coverage for all classes.

.