Entailment as Robust Self-Learner

Jiaxin Ge[1*] and **Hongyin Luo**[2*] and **Yoon Kim**[2] and **James Glass**[2]
[1] Peking University, Beijing, China
[2] MIT Computer Science and Artificial Intelligence Lab, Cambridge MA, US
aomaru@stu.pku.edu.cn, {hyluo, yoonkim, glass}@mit.edu

Abstract

Entailment has been recognized as an important metric for evaluating natural language understanding (NLU) models, and recent studies have found that entailment pretraining benefits weakly supervised fine-tuning. In this work, we design a prompting strategy that formulates a number of different NLU tasks as contextual entailment. This approach improves the zero-shot adaptation of pretrained entailment models. Secondly, we notice that self-training entailment-based models with unlabeled data can significantly improve the adaptation performance on downstream tasks. To achieve more stable improvement, we propose the **Sim**ple **P**seudo-**L**abel **E**diting (SimPLE) algorithm for better pseudo-labeling quality in self-training. We also found that both pretrained entailment-based models and the self-trained models are robust against adversarial evaluation data. Experiments on binary and multi-class classification tasks show that SimPLE leads to more robust self-training results, indicating that the self-trained entailment models are more efficient and trustworthy than large language models on language understanding tasks.

1 Introduction

Although achieving state-of-the-art performance in different natural language understanding (NLU) tasks (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019; Clark et al., 2020; He et al., 2020; Joshi et al., 2020), large-scale pretrained language models still highly depend on human-annotated, task-specific training corpora for fine-tuning because the self-supervised pretraining objective does not incorporate explicit task-related knowledge. As a result, state-of-the-art language models are still challenged by the lack of adequate fine-tuning data and difficult evaluation examples crafted by adversarial attacks or model-in-loop adversarial data annotations (Wang et al., 2021a; Bartolo et al., 2020; Zang et al., 2019; Garg and Ramakrishnan, 2020; Li et al., 2020).

On the other hand, entailment is recognized as a minimal requirement for NLU (Condoravdi et al., 2003). Recent studies have found that entailment learning improves sentence representation (Reimers and Gurevych, 2019a; Gao et al., 2021). However, these models still need fine-tuning with human-annotated training data to handle downstream NLU tasks. The authors of Wang et al. (2021b) found that entailment-based models are also few-shot learners that outperform recent efforts on few-shot NLU. For example, LM-BFF (Gao et al., 2020) proves that entailment learning can significantly improve the data efficiency and adaptation ability of language models.

In this work, we further explore the zero-shot and unsupervised adaptation abilities of entailment-based models without any human-labeled training corpora on downstream tasks. We first study the zero-shot and unsupervised adaptation abilities of the entailment-based language models. Inspired by recent progress on prompt tuning, we formulate different NLU tasks as contextual entailment (Routley and Meyer, 1973) by constructing task-specific suppositions. The language models are trained to predict the truth value of the constructed suppositions. In zero-shot adaptation experiments, we find this approach significantly outperforms naively concatenating different inputs and labels, proving that the supposition construction method mitigates the distribution gap among different NLU tasks.

We further explore the potential of the unsupervised adaptation ability of entailment-based models. We use the pretrained entailment models to predict the pseudo-labels of unlabeled, task-specific language data. We find that the entailment-based models can be improved with self-training (Blum and Mitchell, 1998) with the automatically annotated

[*] Equal contribution. Correspondence to Hongyin Luo at hyluo@mit.edu. Code and processed data are available at https://github.com/luohongyin/EntST.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13803–13817

pseudo-labels (He et al., 2019). While the self-training strategy has been proven effective on different tasks and modalities (Zou et al., 2019; Zoph et al., 2020; Meng et al., 2020; Xie et al., 2020b), a major challenge for self-training is the unstable performance caused by the noisy pseudo-labels. A number of solutions have been proposed to mitigate this issue. The most popular methods are training data selection (Li and Zhou, 2005; Lang et al., 2022) and pseudo-label editing (Shin et al., 2020; Mandal et al., 2020). Recent work also found that simple Dropout (Srivastava et al., 2014) approaches improve contrastive learning (Gao et al., 2021) and speech recognition (Khurana et al., 2021; Dawalatabad et al., 2022).

To combine the benefits of data selection and label editing methods, we propose SimPLE, a **sim**ple **p**seudo-**l**abel **e**ditting algorithm with simple text augmentation, uncertainty-based data filtering, and majority-based pseudo-labeling. Experiments with different backbone models on binary, multi-class, regular, and adversarial NLU tasks show that our approach makes the following contributions,

- Supposition-based task formulation improves the zero-shot adaptation and robustness against adversarial evaluation data of entailment models across different NLU tasks.
- SimPLE improves the pseudo-labeling accuracy on confident and uncertain training samples, leading to significant improvement over all self-training and pretrained baselines.
- Self-trained, 350M-parameter entailment models without human-generated labels outperform supervised language models with 137B parameters, proving the data and computation efficiency of entailment self-training.

2 Related Work

Language modeling. Task-agnostic, large-scale language models can solve a number of natural language understanding (NLU) tasks (Brown et al., 2020; Raffel et al., 2020; Lewis et al., 2019; Wei et al., 2022a,b). On the other hand, pretraining with annotated training corpora of different natural language tasks also benefits the generalize ability and zero-shot adaptation performance (Sanh et al., 2021). Recent studies have found that textual entailment (Bowman et al., 2015; Williams et al., 2018) is a powerful pretraining task. Entailment models are applied for sentence representation learning (Reimers and Gurevych, 2019b; Gao et al., 2021), relation extraction (Obamuyide and Vlachos, 2018; Yin et al., 2019), and fact-checking (Thorne and Vlachos, 2018). The authors of Wang et al. (2021b) showed that entailment models can benefit the few-shot learning performance of pretrained language models on NLU tasks.

Robustness in Self-training. While most self-training studies are under the computer vision context (Zoph et al., 2020; Zou et al., 2019), efforts also exist for self-training the latest neural language models, including back translation (He et al., 2019), text augmentation (Xie et al., 2020a; Chen et al., 2020), question-answer synthesis (Bartolo et al., 2021; Luo et al., 2022), and co-training (Lang et al., 2022). However, self-training methods suffer from noisy pseudo-labels. In computer vision, a straightforward solution is obtaining confident pseudo-labels by augmenting input images (Shin et al., 2020; Mandal et al., 2020; Sohn et al., 2020), including shifting, rotating, or adding noise to pixels. However, data augmentation is not as straightforward for natural language if no additional model is used. Instead, some model-level methods can be applied. Zou et al. (2019) proposed regularizing over pseudo-label confidence to avoid overfitting to simple cases, Gao et al. (2021); Khurana et al. (2021) applied dropout to improve the quality of training corpora. Li and Zhou (2005); Lang et al. (2022) applied a graph-based confidence estimation method for removing training samples with uncertain pseudo labels.

Difference with previous work. Without any additional language model for text augmentation, we propose a model-level, augmented pseudo-labeling method that improves self-training performance for entailment models. Our method avoids dropping training data and performs more stably than dropout-based methods. Different from previous work on weakly-supervised language understanding with entailment models (Wang et al., 2021b), we do not use any human-generated labels. Our models contain 1/500 trainable parameters compared to the models used in Lang et al. (2022); Sanh et al. (2021).

3 Entailment Self-training

Pretraining. Recent studies have found that entailment-based language models can efficiently adapt to different natural language understanding (NLU) tasks with a limited number of human-

labeled training samples (Wang et al., 2021b; Luo and Glass, 2023). In this work, we find that entailment models can be self-improved without any human-generated labels by constructing suppositions (prompts) that describe the given tasks. Most NLU tasks can be formulated as predicting the truth value of the constructed suppositions that wrap inputs and label descriptions, as shown in Table 1.

Task	Inputs	Supposition
MNLI	{p, h}	h is entailed by p.
RTE	{p, h}	h is entailed by p.
QNLI	{t, q}	The answer to q is entailed by t.
QQP	{q_1, q_2}	q_1's answer is entailed by q_2's answer.
SST2	{x}	The movie is good is entailed by x.

Table 1: The suppositions constructed based on the definitions of different GLUE tasks (Wang et al., 2018).

By training the entailment model using the MNLI corpus given with the constructed suppositions, the model can be directly adapted to other tasks with relatively high accuracy. We will show that without entailment pretraining, similar performance can only be achieved by 400 times bigger language models. The entailment-based models can be further fine-tuned on unlabeled texts via self-training. We apply different adaptation strategies for binary and multi-class classification tasks.

Binary classification. Supposition-based entailment models predict True, Neutral, and False scores for each supposition, corresponding to entail, neutral, and contradictory labels of the MNLI corpus. For binary classification, we ignore the neutral score and calculate only True and False probabilities, and the True/False predicted can be linked to corresponding labels according to the supposition. For example, the SST2 supposition in Table 1 being true means that {x} is a positive movie review. The predicted True/False values are used as pseudo-labels for self-training,

Multi-class classification. In binary classification, the model is presented with a single supposition and asked to decide whether it's true or not. In multi-class classification, the model is presented with a context sentence and multiple labels and is asked to choose the correct label.

To predict the correct answer from multiple options, we propose an entailment score ranking method. First, for each sentence to be classified, we construct a supposition for each label. For example, in an emotion classification task, given the sentence `S`, we construct the following suppositions: "I am happy is entailed by `S`", "I am sad is entailed by `S`", and "I am shocked is entailed by `S`". We calculate the entailment probability of each supposition with the entailment model and predict the label associated with the most entailed supposition.

We propose a max-confidence tuning method for self-training. We select the class with the highest entailment score and then record its predicted pseudo-label for further self-training, and ignore other classes. The model does not need to classify each class correctly but merely learns to predict the truth value of its most confident supposition.

4 Simple Pseudo-label Editing

We propose the simple pseudo-label editing (SimPLE) method, a three-step pipeline for generating robust pseudo labels, including augmented pseudo-labeling with dropout, uncertain data filtering, and majority-based relabeling. We introduce the details of each step in this section.

4.1 Simple Augmentation for Pseudo-labeling

Because of languages' discrete and sequential nature, changing a token in a sentence might completely invert its meaning. As a result, unlike straightforward and effective image augmentation processes like FixMatch (Sohn et al., 2020), additional augmentation models are usually needed for text augmentation. Recent studies have found that instead of data-level augmentation, the Dropout mechanism leads to decent embedding-level augmentation. Gao et al. (2021) applied dropout for contrastive sentence representation learning, and Khurana et al. (2021) selected confident pseudo-labels by measuring the consistency of a model with the same input data and random dropouts.

As the first step of generating augmented pseudo labels, we run N independent evaluations with random dropout (dropout rate $= 0.1$) for each input training sample x_i and obtain a set of N noisy pseudo-labels.

$$Y_i = \{y_j = M_j^*(x_i) \mid j \in [0, N)\} \quad (1)$$

where j stands for the j-th independent evaluation with a dropout model M^*. Meanwhile, we store a set of sequence representations $E_i = \{e_0, e_1, \ldots, e_{N-1}\}$ of x_i collected in each feed-forward process. After finishing this step, we collect a set of data, pseudo-label, and embeddings.

$$C = \{(x_i, y_i^j, e_i^j) \mid i \in [0, M), j \in [0, N)\} \quad (2)$$

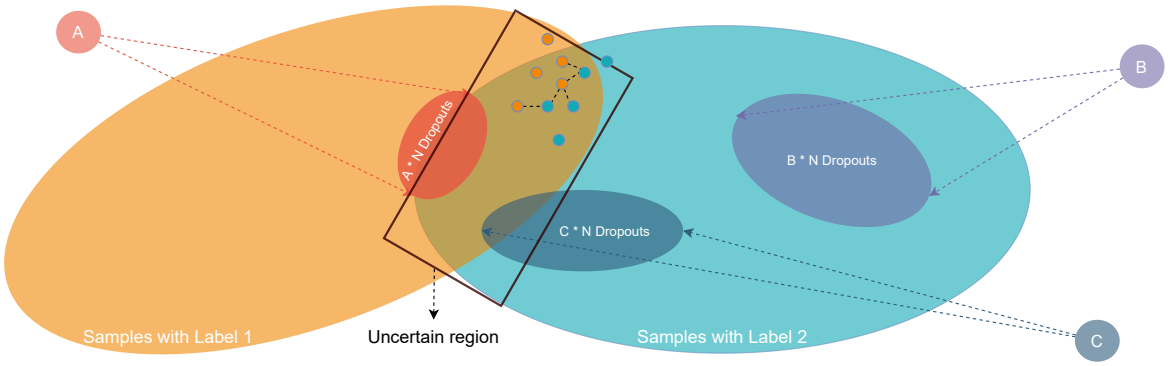

Figure 1: Visualization of the SimPLE method. The figure shows the embedding space of natural sentences, and different colors represent different predicted labels. Each data sample is labeled with multiple random dropouts, and we use the SETRED algorithm to detect the uncertain pseudo-labels. The final label is voted by confident inferences.

where M stands for the number of unlabeled training samples, each associated with N pseudo-labels and corresponding hidden states. In total, the augmented method outputs $M * N$ label-embedding pairs for further processing.

4.2 Uncertainty Estimation

Following Li and Zhou (2005) and Lang et al. (2022), we estimate the confidence of all pseudo-labels using the SETRED algorithm. The motivation of this algorithm is that training samples with similar embeddings are likely to have the same pseudo-labels. On the other hand, if a training sample is located near samples with different pseudo-labels in the embedding space, its own pseudo-label is likely to be uncertain. Using the output data-embedding-label set shown in Equation 2, we can calculate the nearest neighbors of each training sample and estimate the labeling consistency.

To estimate the uncertainty of y_u, the pseudo-label of training sample x_u, we calculate the Euclidean distances between x_u and all other $M*N-1$ samples using the calculated text embeddings. We construct a set of the top k nearest neighbors of x_u, namely $N(u)$. With the nearest neighbor set, an uncertain score of (x_u, y_u) can be calculated as follows,

$$J_u = \sum_{v \in N(u)} \mathbb{I}(y_u \neq y_j) \; / \; (1 + \|e_u - e_v\|_2) \quad (3)$$

where $\mathbb{I}$ is a binary indicator function, whose value is 1 when $y_u \neq y_v$ and 0 otherwise. $\|e_u - e_v\|_2$ stands for the Euclidean distance between the embeddings of x_u and x_v. As a result, J_u would have a higher value when more near neighbors are associated with different pseudo-labels.

To estimate the uncertainty of y_u, we compare J_u with a null hypothesis where all pseudo-labels in C except y_u are randomly shuffled. After the shuffling, the entire data-label mapping set becomes uncertain. The expectation and variance of J_u after shuffling is

$$\mathbb{E}_v[J_u] = (1 - \hat{P}_{y_u}) \sum_{v \in N(u)} 1/(1 + \|e_u - e_v\|_2)$$

$$\sigma(J_u)^2 = \hat{P}_{y_u}(1-\hat{P}_{y_u}) \sum_{v \in N(u)} 1/(1+\|e_u-e_v\|_2)^2$$

The uncertainty can be estimated by verifying the significance of the difference between J_u and the null hypothesis. An uncertainty score can be calculated as

$$s(u) = \frac{J_u - \mathbb{E}_v[J_u]}{\sigma(J_u)} \quad (4)$$

With this method, we calculate uncertainty scores for all $M * N$ training samples in C for further processing.

4.3 Filtering and Relabeling

After finishing estimating the uncertainty of each training sample, we sort all training samples in C by their uncertainty scores and remove the 20% most uncertain training samples. The remaining samples are used for relabeling based on majority voting. For example, a training sample x_i has

N pseudo-labels $[y_0^i, y_1^i, \ldots, y_{N-1}^i]$ after the augmented labeling step, and n labels are removed based on the uncertainty scores.

The final pseudo-label of x_i is decided by the voting result of the $N - n$ remaining labels. If all generated pseudo-labels of a training sample are removed or there is a tie in the voting, we re-run the labeling process without dropout to get the final pseudo-label. Following this approach, we keep all training samples and, meanwhile, obtain a more robust pseudo-label set.

5 Experiments

Benchmarks. We conduct experiments on popular natural language understanding tasks in the GLUE (Wang et al., 2018) benchmark, including RTE (Dagan et al., 2005), QNLI (Rajpurkar et al., 2016), QQP, SST-2 (Socher et al., 2013), and CoLA (Warstadt et al., 2019). We also assess the robustness of the proposed method against adversarial evaluation sets in the AdvGLUE corpus (Wang et al., 2021a), including Adv-QNLI, Adv-QQP, Adv-RTE, and Adv-SST2. The data in AdvGLUE is created by adding word-level and sentence-level perturbations to the GLUE data, as well as human-crafted examples. For Multi-Classification, we use Copa (Alex Wang, 2019) (which consists of questions paired with two answer choices), Emotion Classification (Elvis Saravia, 2018), Amazon Review (Phillip Keung, 2020) and Ag-News (Xiang Zhang, 2015). More details are shown in Appendix A.

Hyper-parameters. We train 350M RoBERTa (Devlin et al., 2018) and DeBERTa (He et al., 2020) models for the language understanding tasks, without using larger language models like GPT-3 (Brown et al., 2020) or T0 (Sanh et al., 2021) that are used for generating pseudo-labels in (Lang et al., 2022). We also use the same hyper-parameters across all tasks, attempting to avoid the problems mentioned in Perez et al. (2021). In the entailment pretraining on the MNLI dataset (Williams et al., 2018), we optimize both RoBERTa and DeBERTa models with the AdamW optimizer (Loshchilov and Hutter, 2018). For all tasks and both models, we set $\varepsilon = 10^{-6}$. In the entailment pretraining, we set the weight decay weight to 10^{-5}, and the learning rate for both models is 3e-6. During the self-training step, the learning rate of both models on all binary classification tasks is 4e-6 and is 1e-6 on multi-classification tasks, and the weight decay is constantly 10^{-2}. We run the entailment pretraining for 2 epochs and the self-training for 6 epochs. In confidence-based labeling, we drop 1/8 data with the lowest confidence.

Self-training details. For each binary classification task, we randomly select $N = 2000$ unlabeled data examples. For each multi-classification task, we randomly select $N = 50$ unlabeled data examples. To estimate the uncertainty of the pseudo-labels in SETRED and SimPLE algorithms, we use the hidden states of the 4th layer from the top of both RoBERTa and DeBERTa language models as the supposition embeddings and measure the uncertainty with 9 neighbors. In SimPLE, we run 7 inferences for each training sample with different dropouts. We train and evaluate the models for each task with 10 independent runs on 2 V100 32G GPUs. Each experiment takes less than an hour.

Assessment. We evaluate the performance of our algorithm by comparing the average classification accuracy against baseline methods and the robustness. We describe the term *Robustness* as follows: in multiple independent experiments, a robust method should achieve high maximum, minimum, and average accuracy against with different backbone model and training data, on different natural language understanding tasks.

5.1 GLUE and AdvGLUE Tasks

The experiment results are shown in Table 2. We compare the adaptation performance of entailment-based language models and the improvement of different self-training approaches.

Compare with supervised baselines. We compare our entailment self-training methods with few-shot fine-tuning baselines. The few-shot baselines, including PET (Schick and Schütze, 2021), LM-BFF (Gao et al., 2020), P-tuning (Liu et al., 2021), PPT (Gu et al., 2021), and UPT (Wang et al., 2022), are based on 350M BERT or RoBERTa backbones. Our pretrained DeBERTa entailment model outperforms the best few-shot baseline (LM-BFF) by 4.5%, and the RoBERTa entailment model outperforms LM-BFF by 1.5%. With self-training, our SimPLE method further improves the model's performance by a large margin. The RoBERTa performance is boosted by nearly 5% and the average performance of DeBERTa is over 86%, outperforming the best few-shot supervised baselines by 6.9%.

On the other hand, we compare our model with fully supervised RoBERTa/DeBERTa models and

Method	GLUE					Method	AdvGLUE				
	QNLI	QQP	RTE	SST2	Avg.		QNLI	QQP	RTE	SST2	Avg.
Few-shot (left) and fully-supervised (right) medium LMs (350M) with human-generated labels											
PET	61.3	67.6	65.7	91.8	71.6	R3F	47.5	40.6	50.1	38.5	44.2
LM-BFF	69.2	69.8	83.9	90.3	78.3	CT_T	49.6	40.7	46.2	39.2	43.9
P-tuning	58.8	67.6	70.8	92.6	72.5	MT	47.5	41.5	52.5	51.3	48.2
PPT	68.8	67.2	67.9	92.3	74.1	BERT	39.8	37.9	40.5	33.0	37.8
UPT	70.1	72.1	68.9	92.9	76.0	RoBERTa	52.5	45.4	62.8	58.5	54.8
EFL	68.0	67.3	**85.8**	90.8	78.0	DeBERTa	57.9	60.4	**79.0**	57.8	63.8
Few-shot large LMs (137B) with human-generated labels											
LaMDA	55.7	58.9	70.8	92.3	69.4	\	-	-	-	-	-
FLAN	63.3	75.9	84.5	**94.6**	79.6	\	-	-	-	-	-
Zero-shot adaptation of entailment classifiers based on medium LMs (350M)											
DeBERTa-Cat	71.6	70.5	74.0	84.6	75.2	\	60.8	47.4	50.6	56.1	53.7
RoBERTa-Sup	71.5	78.6	81.2	87.7	79.8	\	62.1	52.6	61.7	**59.9**	59.1
DeBERTa-Sup	77.3	**79**.9	84.5	90.1	82.9	\	61.5	64.1	66.7	42.6	58.7
Self-trained `RoBERTa-large` (350M) without human-generated labels											
Baseline-ST	74.1	80.1	81.5	88.3	81.0	Baseline-ST	64.9	60.6	60.9	56.6	60.8
Dropout	78.5	80.5	80.9	88.8	82.2	Dropout	69.2	57.8	61.9	57.3	61.6
SETRED	80.5	80.5	80.8	88.3	82.5	SETRED	68.0	56.5	62.6	58.9	61.5
SimPLE (ours)	83.1	80.7	83.1	91.6	84.6	SimPLE (ours)	69.6	54.4	62.3	58.8	61.3
Self-trained `DEBERTa-large` (350M) without human-generated labels											
Baseline-ST	79.0	80.2	83.4	92.1	83.7	Baseline-ST	65.8	**70.4**	68.4	50.9	63.9
Dropout	81.1	80.5	84.1	91.8	84.4	Dropout	70.1	63.3	70.9	49.9	63.6
SETRED	83.4	80.5	83.9	92.0	84.9	SETRED	69.8	69.5	69.9	50.9	65.0
SimPLE (ours)	**85.2**	**81.0**	85.5	92.8	**86.1**	SimPLE (ours)	**70.1**	68.1	73.8	51.6	**65.9**

Table 2: Experimental results on binary classification tasks with 10 independent experiments. Cat stands for concatenation-based pretraining and Sup stands for supposition classification.

robust training methods, including R3F (Aghajanyan et al., 2020), child tuning (CT) (Xu et al., 2021), and match tuning (MT) (Tong et al., 2022) models, on the AdvGLUE benchmark. We found that the fully-supervised DeBERTa model is the best baseline on the AdvGLUE benchmark. However, our RoBERTa entailment model outperforms all robust training baselines with the same pretrained backbone by over 10%. With SimPLE self-training, the DeBERTa entailment model achieves the best performance on AdvGLUE, outperforming the fully-supervised DeBERTa model by 2.1% as well as all other baselines.

We found that our pretrained entailment models outperform EFL, the few-shot fine-tuned entailment model based on `RoBERTa-large` proposed by Wang et al. (2021b). The self-trained models further outperform EFL with larger margins. This indicates the strong adaptation ability introduced by the supposition-based NLU strategy.

Compare with large language models. We found that both zero-shot pretrained and semi-supervised self-trained entailment models outperform the few-shot large language models on QNLI, QQP, and RTE tasks, and achieve significantly higher average accuracy on GLUE. This suggests that our method is computation-efficient - the models use 1/400 parameters, without human-generated task-specific labels, but achieve better performance than expensive large-scale language models on NLU tasks.

Compare with self-training baselines. By averaging 10 independent evaluations across GLUE and AdvGLUE benchmarks and backbone models, we found that Dropout and SETRED improve baseline self-training performance on a similar level. On average, SETRED outperforms Dropout by 0.5% on 4 experiment settings. On the GLUE benchmark, the SimPLE method improves the model's performance by 1.5 to 2% on average. The highest improvement boost is on the QNLI tasks, where the SimPLE self-training method outperforms the baseline self-training by 9% and 6% on RoBERTa and DeBERTa respectively. Although the average improvement is not very high, we will show that SimPLE is significantly more robust. The results show that augmenting the pseudo-labels without removing uncertain training samples benefits self-training, which aligns with our hypothesis.

In general, the experiments on binary classification NLU tasks proved the data and computation efficiency of entailment self-training over different strong baseline models. Furthermore, the SimPLE

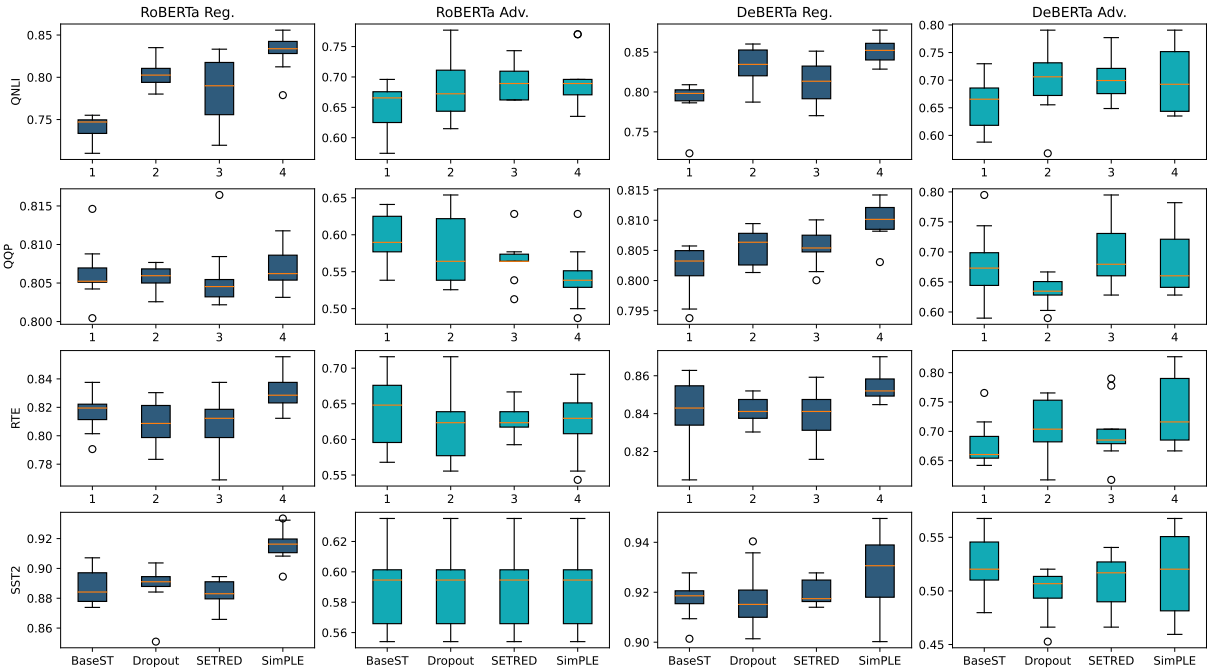

Figure 2: The results of 10 independent experiments with self-trained RoBERTa and DeBERTa models on GLUE (*BERTa Reg.) and AdvGLUE (*BERTa Adv.) benchmarks.

algorithm we propose in this work achieves the best average performance, significantly outperforms all baselines on some of the tasks, and meanwhile preserves the robustness of entailment models against adversarial benchmarks.

5.2 Multi-class NLU Tasks

The experiment results on Copa, Emotion, Amazon Review, and Ag News are shown in Table 3. In multi-classification tasks, we present the comparative results of the pretrained entailment-based language models and the 4 self-training approaches compared in the previous section with binary NLU tasks, including standard self-training, dropout-based re-labeling, SETRED, and SimPLE.

The effect of dropout-based augmentation. By merely using dropout, the augmented self-training outperforms the standard normal self-training baseline which keeps all the pseudo-labels in general. This further validates the previous finding that by adding dropout, the models adopt noises that benefit the inference, generate augmented pseudo-labels and mitigate the overfitting problem.

The effect of SETRED. By merely using SETRED, the self-training does not see a consistent improvement in performance and even falls behind the pretrained and standard self-trained models that preserve all pseudo labels in some tasks (like AmazonReview). This fact suggests that removing uncertain pseudo-labels can lead the model to overfit confident training samples, thus hurting the self-fine-tuning performance.

The effect of SimPLE. Table 3 shows that the SimpLE algorithm constantly outperforms all pretrained and self-trained baselines on both backbone models across all multi-class benchmarks, which aligns with the result on the binary NLU tasks. This fact further validates our hypothesis that augmenting the pseudo-labels of uncertain training samples can improve the performance of self-training.

Compare with Large Language Models. We notice that our self-trained methods can outperform several large language models. On Emotion and AG News tasks, the pretrained entailment model without self-training can achieve a significant improvement over the `GPT-3-175b` model, which is 500 times large than the entailment model. This indicates that the entailment-based model is a more efficient and trustworthy option for many natural language understanding tasks.

6 Analysis

Robustness. Besides the mean accuracy of all experiments, we also visualize the results of all independent evaluations of different self-training strategies in Figure 2. We found that SimPLE constantly outperforms other self-training baselines on the regular GLUE benchmark by comparing

Method	Multi-Classification				
	Copa	EM	AR	News	Avg
DEBERTa-large (350M)					
Pretrain	77.0	51.93	37.01	73.40	59.84
BaseST	78.75	51.24	38.80	73.10	60.47
Dropout	78.25	53.69	38.19	73.16	60.82
SETRED	78.0	52.42	37.61	73.33	60.34
SimPLE	**79.75**	**54.58**	**39.05**	**73.57**	**61.74**
RoBERTa-large (350M)					
Pretrain	76.0	49.21	33.31	63.18	55.43
BaseST	76.67	50.94	37.38	64.64	57.41
Dropout	78.67	50.99	42.87	61.05	58.40
SETRED	78.0	50.53	27.16	63.24	54.73
SimPLE	**79.0**	**51.79**	**44.06**	**65.60**	**60.11**
Large Language Models					
Zero-shot	70.0$^{\diamondsuit}$	42.7‡	-	43.9‡	-
Few-shot	77.0†	-	-	61.0‡	-
Class Num	2	6	5	4	-

Table 3: Multi-class NLU results with 3 independent runs. The Copa model selects from two candidate sentences, which is different from the previous binary NLU tasks. $\diamondsuit$: `T5-11b`, †: `GPT-Neo-6b`, ‡: `GPT-3-175b`. The performance of large language models are cited from Zhao et al. (2021) and Wang et al. (2023).

mean, maximum, and minimum accuracy. Although DeBERTa performs similarly under different self-training strategies on QQP in terms of average accuracy, there exists a significant gap between the minimal performance of baseline and SimPLE. This indicates that SimPLE is more robust and safer compared with the regular self-training algorithm. The only exception is the DeBERTa model on SST2 - the mean performance of SimPLE is better, but it has a lower minimal performance than the baseline self-training method.

Most models overfit to the training corpora and achieve high accuracy on regular evaluation sets, but perform poorly on adversarial benchmarks (Wang et al., 2021a). As a result, fully supervised models achieve less than 60% accuracy on AdvGLUE. We also investigate if SimPLE hurts the model's robustness against adversarial evaluation data. We found that, except RoBERTa on AdvQQP, other settings show that the entailment-based models are still robust after SimPLE self-training. As we compared in Table 2, all these results significantly outperform fully-supervised baselines.

Pseudo-labeling Accuracy. We show the pseudo-labeling accuracy of RoBERTa and DeBERTa-based entailment models with different strategies in Figure 3 with 10 independent experiments. The results indicate that the DeBERTa models predict more accurate pseudo-labels in general. On the other hand, the pseudo-label sets produced by SimPLE with both models are significantly less noisy than the standard and dropout-based labeling methods without removing any uncertain data samples. SETRED achieves the highest labeling accuracy because it drops uncertain samples. The comparison suggests that SimPLE achieves the highest performance because it achieves high pseudo-labeling accuracy on uncertain training samples.

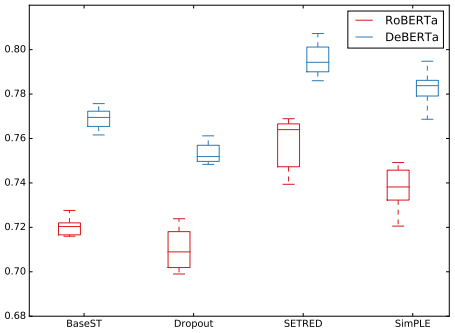

Figure 3: Pseudo-labeling accuracy of entailment models with standard (ST), dropout, SETRED, and SimPLE strategies. SETRED achieves higher accuracy because uncertain data samples are dropped.

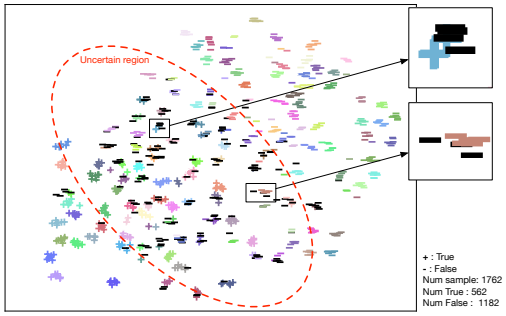

Figure 4: Visualization of the embeddings, pseudo-labels, and uncertainty of QNLI suppositions calculated by the pretrained DeBERTa entailment model. Each data sample has 7 embeddings calculated with different dropouts. Black stands for uncertain points and other colors stand for different training examples.

Case study. We visualize the hidden states, pseudo-labels, and confidence of the training samples in the QNLI tasks calculated by the pretrained DeBERTa entailment model with the SimPLE algorithm in Figure 4. The embedding space is calculated with t-SNE (Van der Maaten and Hinton, 2008) using 252 training samples with 252*7=1764 embeddings. Half of them are plotted in the figure. Each training sample is evaluated with 7 different dropouts, and the uncertainty is estimated with 9 neighbors. In

Figure 4, different embeddings of the same training sample are labeled with the same color, while the uncertain cases are marked in black. **+** and **-** stand for the truth value of the suppositions. As shown in the figure, most uncertain cases appear around the uncertain circle. We also highlight two training samples with uncertain representations. This phenomenon indicates that the SimPLE algorithm can drop most embeddings of a data sample and edit the voting results of the dropout-based pseudo-labeling method, improving the pseudo-labeling accuracy from 76.5% to 79.2% in this experiment.

We also show that the original pseudo-label set is unbalanced, with 67.1% of all predicted labels being "False". Although we do not provide any prior knowledge about the label distribution of the task (unknown without human annotation), the SimPLE method mitigates the bias through the uncertain candidate removal process. Figure 4 shows that most uncertain pseudo-labels estimated by SimPLE are "False", thus the remaining pseudo-labels are more balanced.

7 Conclusion

We show that entailment-based language models can be adapted to different NLU tasks without supervision and achieve robust performance against noisy pseudo-labels and adversarial texts. We design a supposition-based prompting strategy to improve the zero-shot adaptation performance of entailment-based models. To improve the stability of self-training, we propose the SimPLE algorithm for augmented pseudo-labeling. Experiments on binary, multi-class, regular, and adversarial NLU tasks show that the SimPLE self-training strategy significantly outperforms a number of strong baselines, including 400 and 500 times larger language models on both zero-shot and weakly supervised settings, proving the effectivenss of entailment self-training for efficient and trustworthy natural language understanding systems.

Limitations

Our method utilized pretrained entailed models and adapted them to other domains under zero-shot and self-training settings. There are two limitations that we would like to improve in future work. Firstly, we use human-designed suppositions for each task, which is less automatic than a direct, zero-shot adaptation of the models. Secondly, the self-training on some multi-class classification tasks is not as high as on binary NLU tasks, indicating the challenge of applying entailment models to multi-choice tasks. We would like to overcome this in the next step.

Ethics Statement

We propose a method that can significantly reduce the financial and environmental cost of language model learning. By reducing the need for data collection and human labeling, our method can effectively protect user and data privacy by avoiding leaking any information while building the training corpora. We found that a medium-sized language model can achieve similar performance as the state-of-the-art large-scale language models, suggesting that we can cost less financially and environmentally during model training and evaluation for comparable performance. However, since we reduced the need for human-labeling efforts, the deployment of the system might decrease the number of data annotation jobs.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Nikita Nangia Amanpreet Singh Julian Michael Felix Hill-Omer Levy Samuel R. Bowman Alex Wang, Yada Pruksachatkun. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arxiv preprint: arXiv:1905.00537*.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. 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.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated

corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Cleo Condoravdi, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pages 38–45.

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

Nauman Dawalatabad, Sameer Khurana, Antoine Laurent, and James Glass. 2022. On unsupervised uncertainty-driven speech pseudo-label filtering and model calibration. *arXiv preprint arXiv:2211.07795*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yen-Hao Huang Junlin Wu Yi-Shin Chen Elvis Saravia, Hsien-Chi Toby Liu. 2018. Carer: Contextualized affect representations for emotion recognition. *EMNLP 2018*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better -shot learners. *arXiv preprint arXiv:2012.15723*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for -shot learning. *arXiv preprint arXiv:2109.04332*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Sameer Khurana, Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2021. Unsupervised domain adaptation for speech recognition via uncertainty driven self-training. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6553–6557. IEEE.

Hunter Lang, Monica Agrawal, Yoon Kim, and David Sontag. 2022. Co-training improves prompt-based learning for large language models. *arXiv preprint arXiv:2202.00828*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Ming Li and Zhi-Hua Zhou. 2005. Setred: Self-training with editing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 611–621. Springer.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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*.

Hongyin Luo and James Glass. 2023. Logic against bias: Textual entailment mitigates stereotypical sentence reasoning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1243–1254, Dubrovnik, Croatia. Association for Computational Linguistics.

Hongyin Luo, Shang-Wen Li, Mingye Gao, Seunghak Yu, and James Glass. 2022. Cooperative self-training of machine reading comprehension. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 244–257, Seattle, United States. Association for Computational Linguistics.

Devraj Mandal, Shrisha Bharadwaj, and Soma Biswas. 2020. A novel self-supervised re-labeling approach for training with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. *EMNLP 2018*, page 72.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

György Szarvas Noah A. Smith Phillip Keung, Yichao Lu. 2020. The multilingual amazon reviews corpus. *arxiv preprint: arXiv:2010.02573*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Richard Routley and Robertk Meyer. 1973. The semantics of entailment. In *Studies in Logic and the Foundations of Mathematics*, volume 68, pages 199–243. Elsevier.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for -shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. 2020. Two-phase pseudo label densification for self-training based domain adaptation. In *European conference on computer vision*, pages 532–548. Springer.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. *arXiv preprint arXiv:1806.07687*.

Shoujie Tong, Qingxiu Dong, Damai Dai, Tianyu Liu, Baobao Chang, Zhifang Sui, et al. 2022. Robust fine-tuning via perturbation and interpolation from in-batch instances. *arXiv preprint arXiv:2205.00633*.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021a. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.

Jianing Wang, Chengyu Wang, Fuli Luo, Chuanqi Tan, Minghui Qiu, Fei Yang, Qiuhui Shi, Songfang Huang, and Ming Gao. 2022. Towards unified prompt tuning for -shot text classification. *arXiv preprint arXiv:2205.05313*.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021b. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. 2023. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*.

Alex Warstadt, Amanpreet Singh, and Samuel Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

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.

Yann LeCun Xiang Zhang, Junbo Zhao. 2015. Character-level convolutional networks for text classification. *arxiv preprint: arXiv:1509.01626*.

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.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9514–9528.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2019. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845.

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. 2019. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991.

A Data Details

GLUE/AdvGLUE In this work, we evaluate our method with the GLUE[1] and AdvGLUE[2] benchmarks. We pretrain our models on MNLI, and evaluate on all other AdvGLUE tasks, AdvQNLI, AdvQQP, AdvRTE, and AdvSST2. we also evaluate the models on the regular versions of these tasks in GLUE. The statistics of the GLUE and AdvGLUE benchmarks are shown in Table 4.

Multi-Classification In multi-class classification tasks, we evaluate our method with SuperGlue Copa(Alex Wang, 2019), Emotion Classification(Elvis Saravia, 2018), and Amazon Review(Phillip Keung, 2020). The statistics of these corpora are shown in Table 5.

[1] https://gluebenchmark.com/

[2] https://adversarialglue.github.io/

Corpus	|Train|	|Test|	|Adv-Test|
MNLI	393k	20k	1.8k
QNLI	105k	5.4k	0.9k
QQP	364k	391k	0.4k
RTE	2.5k	3k	0.3k
SST2	67k	1.8k	1.4k

Table 4: Statistics of the corpora used in this work

Corpus	|Train|	|Test|	|Class Num|
Copa	400	100	2
Emotion	16k	2k	6
AR	200k	5k	5
News	120k	7.6k	4

Table 5: Statistics of the corpora used in multi-class classification

Reproducibility

Data. We introduce the tasks and corpora we used for training and evaluation in Section 5 and Appendix A.
Method. We introduce the difference between our method and previous work in Section 2, the details of our method in Section 3 and 4.
Hyper-parameter. We describe the key hyper-parameters of self-training in Section 5.
Experiments. We describe the experiment results, and number of independent runs in Section 5, and Section 6 to prove the statistical significance.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The section after conclusion

☑ A2. Did you discuss any potential risks of your work?
The section after the Limitation sectioon

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstraction & section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4 to 6, used Huggingface transformers and PyTorch packages.

☑ B1. Did you cite the creators of artifacts you used?
Yes, section 5

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The packages are widely used for public research.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The packages are widely used for public research.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The evaluation corpora are widely used for public research.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The packages are widely used for public research.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix

C ☑ Did you run computational experiments?

Section 5 and 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5 and 6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Sectioon 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.