Adversarial Robustness of Prompt-based Few-Shot Learning for Natural Language Understanding

Venkata Prabhakara Sarath Nookala*
Georgia Institute of Technology
vnookala3@gatech.edu

Gaurav Verma*
Georgia Institute of Technology
gverma@gatech.edu

Subhabrata Mukherjee
Microsoft Research
subhabrata.mukherjee@microsoft.com

Srijan Kumar
Georgia Institute of Technology
srijan@gatech.edu

Abstract

State-of-the-art few-shot learning (FSL) methods leverage prompt-based fine-tuning to obtain remarkable results for natural language understanding (NLU) tasks. While much of the prior FSL methods focus on improving downstream task performance, there is a limited understanding of the adversarial robustness of such methods. In this work, we conduct an extensive study of several state-of-the-art FSL methods to assess their robustness to adversarial perturbations. To better understand the impact of various factors towards robustness (or the lack of it), we evaluate prompt-based FSL methods against fully fine-tuned models for aspects such as the use of unlabeled data, multiple prompts, number of few-shot examples, model size and type. Our results on six GLUE tasks indicate that compared to fully fine-tuned models, vanilla FSL methods lead to a notable relative drop in task performance (i.e., are less robust) in the face of adversarial perturbations. However, using *(i)* unlabeled data for prompt-based FSL and *(ii)* multiple prompts flip the trend. We further demonstrate that increasing the number of few-shot examples and model size lead to increased adversarial robustness of vanilla FSL methods. Broadly, our work sheds light on the adversarial robustness evaluation of prompt-based FSL methods for NLU tasks.

1 Introduction

Few-shot learning (FSL) capabilities of large language models have led to a remarkable performance on several natural language understanding (NLU) tasks, often with as little as 16 examples per class (Mukherjee et al., 2021; Lester et al., 2021; Li and Liang, 2021; Wang et al., 2021c). Prompt-based few-shot learning is one such approach where NLU tasks are reformulated as prompts, which are then completed using large language models (Gao et al., 2020; Schick and Schütze, 2020; Tam et al., 2021; Liu et al., 2021). By effectively bridging the gap between the pre-training objective of large language models and the fine-tuning objective, prompt-based learning has provided impressive results. Several recent studies have investigated conditioning large language models to solve downstream tasks by prompting them with a few examples.

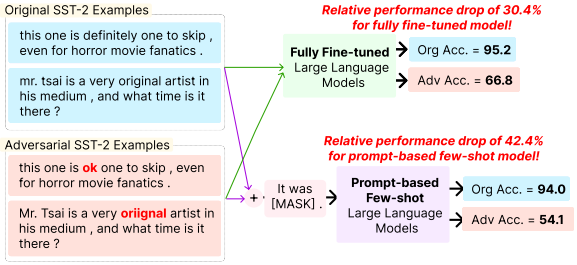

Figure 1: **Overview of our study.** We compare the relative gap between the in-domain and adversarial performance of different state-of-the-art prompt-based few-shot learning methods with that of the models trained with fully supervised learning.

While much of the prior FSL works (Gao et al., 2020; Liu et al., 2021; Tam et al., 2021; Lester et al., 2021) focus on improving downstream task performance, it is also critical to evaluate language technologies for adversarial robustness as that can highlight the security and safety risks of integrating them into user-sensitive downstream applications. The robustness and generalization capabilities of prompt-based few-shot learning models have been the focus of some recent studies. For instance, Razeghi et al. (2022) found that prompting language models is not robust to pre-training term frequencies in the context of arithmetic tasks. In a similar vein, a recent study found that prompt-based FSL is susceptible to learning superficial cues that hinder the generalizability of such methods (Kavumba et al., 2022). On the other hand, encouragingly, Liu et al. (2022) and Awadalla et al. (2022) found that prompt-based FSL leads to more

*Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2023, pages 2196–2208
July 9-14, 2023

robust models in the face of out-of-distribution samples. We add to the existing body of work by specifically studying the robustness of prompting to adversarial samples, which is different from studying the robustness against natural out-of-distribution samples—unlike natural distribution shifts, adversarial samples are carefully designed to exploit the vulnerabilities of language technologies and can pose serious safety concerns in real-world applications (Madry et al., 2017; Huang et al., 2017).

In this work, we conduct the first study that empirically evaluates the adversarial robustness of prompt-based FSL methods for NLU and compares it against the robustness of fully supervised models. We study several state-of-the-art prompt-based FSL methods and evaluate their adversarial robustness on 6 different tasks included in the GLUE benchmark. For each of the tasks, we use the adversarial evaluation set (AdvGLUE) curated by Wang et al. (2021a) to quantify the adversarial robustness of different FSL methods. AdvGLUE is a rich adversarial benchmark that comprises human-validated adversarially perturbed examples that include automated word- and sentence-level perturbations as well as human-crafted examples. We select prompt-based learning approaches that include the following modeling variations: *(i)* no use of unlabeled data, *(ii)* use of unlabeled data, and *(ii)* use of multiple prompts for ensembling. Together, these modeling variations cover the different categories of prompt-based FSL methods identified in the FewNLU benchmark (Zheng et al., 2021). Finally, we compare the models trained using prompting techniques with models trained on fully labeled data using conventional fine-tuning in terms of the gap in the performance between the adversarial and the in-domain evaluation sets.

We summarize our findings below:
1. Vanilla prompt-based fine-tuning (LM-BFF (Gao et al., 2020)) demonstrates a worse relative drop in adversarial performance with respect to in-domain performance than full fine-tuning, and even classic fine-tuning with few examples.
2. However, using unlabeled data (iPET (Schick and Schütze, 2020)) during fine-tuning flips the trend, causing prompting to reduce the drop in adversarial performance with respect to in-domain performance than full fine-tuning.
3. Similarly, using multiple prompts to fine-tune multiple models (PET (Schick and Schütze, 2020)) and ensembling the resultant predictions cause prompting to demonstrate a better relative drop in adversarial performance with respect to in-domain performance than full fine-tuning.
4. Using several ablations, we demonstrate that increasing the number of few-shot examples and the encoder size reduces the relative drop in adversarial performance with respect to in-domain performance. We also find that RoBERTa (Liu et al., 2019) encoders are more adversarially robust than ALBERT (Lan et al., 2019) and BERT (Devlin et al., 2018) encoders of comparable size.[1]

We discuss the implications of these findings and contextualize them with respect to prior studies on other aspects of the robustness of prompt-based few-shot learning.

2 Related Work

Few-shot Learning for NLU: Few-shot learning aims to train models to perform well on a wide range of natural language understanding tasks with a small amount of task-specific training data (Zheng et al., 2021; Mukherjee et al., 2021). Recent studies have explored a wide range of techniques for few-shot learning, like meta-learning on auxiliary tasks (Dou et al., 2019; Nooralahzadeh et al., 2020), semi-supervised learning with unlabeled data (Xie et al., 2020; Mukherjee and Awadallah, 2020), and intermediate learning with related tasks (Yin et al., 2020; Zhao et al., 2021; Phang et al., 2018). A popular and influential branch of few-shot learning approaches involves fine-tuning large language models using *prompting* (Schick and Schütze, 2021). In such approaches, a handful of training examples are transformed using *templates* and *verbalizers*, and the language models are trained to predict the masked verbalizers under various settings.[2] By framing the downstream tasks as a `MASK` prediction task, prompt-based learning overcomes the requirement of training task-specific classification heads, matching the fine-tuning objective with the pre-training objective. FewNLU (Zheng et al., 2021), a benchmark designed to evaluate the performance of prompt-based few-shot learning capabilities systematically, categorizes these settings to fall in one or more of

[1]Code for our experiments: https://github.com/claws-lab/few-shot-adversarial-robustness

[2]For instance, the sentiment classification could involve the following transformation using a template: "*I loved the movie!*" → "*I loved the movie! It was* `[MASK]`", with the language models being trained to predict verbalizers "great" or "terrible" for positive and negative sentiment labels, respectively.

the following categories: *(i)* not using any unlabeled data, *(ii)* using unlabeled data, and *(iii)* using an ensemble of models trained using different prompts. Overall, evaluation of multiple prompt-based few-shot learning approaches has demonstrated that they solve NLU tasks to a remarkable extent with as little as 16 labeled examples per class when compared against fine-tuned models that are trained with thousands of labeled examples. Such data-efficient learning capabilities are critical for building language technologies where it is challenging to collect large-scale labeled datasets. However, these approaches must demonstrate adversarial robustness to ensure safe outcomes in real-world applications where untrusted sources could supply the inputs. To this end, in this work, we systematically study the adversarial robustness of prompt-based few-shot learning approaches while considering the benefits of various settings identified in the FewNLU benchmark (i.e., the role of unlabeled data and ensembling).

Robustness of Few-shot Learning: Prior work has investigated the robustness of various few-shot learning of computer vision (Goldblum et al., 2020) and natural language processing models (Liu et al., 2022; Awadalla et al., 2022), with some works also developing new robust learning approaches (Jiang et al., 2019; Wortsman et al., 2022). Such robustness assessments are distinguished into two categories: *(a)* robustness to natural and unintentional perturbations, and *(b)* robustness to adversarial perturbations. Our work focuses explicitly on the adversarial robustness of prompt-based few-shot learning for natural language understanding.

The most related works to ours are the studies by Liu et al. (2022) and Awadalla et al. (2022). Both studies consider the robustness of a wide range of data-efficient approaches to out-of-distribution (OOD) *natural* examples. Liu et al. (2022) find that prompt-based few-shot learning approaches lead to *more robust models* than their fully fine-tuned counterparts. Awadalla et al. arrive at the same finding in the specific context of Question Answering tasks. However, since both works focus on out-of-distribution samples that are considered likely and natural, it is unclear if their findings also hold for samples that attackers adversarially perturb. Consequently, we specifically focus on the adversarial robustness of data-efficient learning for NLU. Our findings show that, contrary to the trends observed for OOD samples in prior works, in-domain performance is not a good predictor of adversarial robustness of prompt-based few-shot learning approaches compared to fully supervised approaches. In other words, fully supervised models demonstrate a lesser relative drop in adversarial performance with respect to in-domain performance than prompt-based few-shot approaches. However, when strategies such as *(a)* using unlabeled data and *(b)* ensembling over models trained with multiple prompts are adopted, the resultant models demonstrate better adversarial robustness than fully fine-tuned models.

3 Experimental Setup

Few-shot Learning (FSL) Methods using Prompting: We evaluate four different FSL methods that are commonly used for natural language understanding tasks: Classic fine-tuning (Devlin et al., 2018), LM-BFF (Gao et al., 2020), PET, and iPET (Schick and Schütze, 2020, 2021). Together, these approaches cover three primary settings in state-of-the-art prompt-based FSL methods, namely, (i) no use of unlabeled data for training, (ii) use of unlabeled data, and (ii) using ensembles of models trained with different prompts. We consider fine-tuning with fully labeled data to give the ceiling performance and contrast the capabilities of the FSL methods. Below, we briefly describe the FSL methods and explain our rationale for considering them in our study.

1. Classic-FT: We use the `[CLS]` token representation from the encoder with a softmax classifier on top and train the model end-to-end on a few labeled examples (no unlabeled data).

2. LM-BFF: Gao et al. (2020) proposed few-shot fine-tuning with prompting using demonstrations. Their approach for FSL involves *concatenating* the input example, which is modified to follow the prompting template with a `[MASK]` in place of the verbalizer, with semantically similar examples (i.e., demonstrations) from the few-shot training set. Concatenating one demonstration per class with the input example enables overcoming the long-context problem of GPT-3's in-context learning. During inference, LM-BFF ensembles the predictions made by concatenating the input example with all demonstrations from the few-shot training set. LM-BFF does not use unlabeled data for training.

3. PET: Pattern-Exploiting Training (PET) (Schick and Schütze, 2020) is a simple prompt-based few-

shot fine-tuning approach where the training examples are converted into templates, and the [MASK] tokens are used to predict the verbalizer, which indicates the output label. To understand the role of using multiple prompts in robustness, we use PET to fine-tune models with different template-verbalizer pairs and ensemble their predictions during inference. PET does not use demonstrations or unlabeled data.

4. iPET: iPET (Schick and Schütze, 2020, 2021) involves self-training and leverages unlabeled data during fine-tuning. It iteratively uses PET to produce multiple generations, assigning pseudo-labels to unlabeled data at the end of each generation stage. This pseudo-labeled data from a previously fine-tuned model is then used along with the few-shot training data to update the model in the subsequent generation stage. iPET uses unlabeled data and allows us to understand its impact on adversarial robustness.

GLUE and AdvGLUE Benchmarks: We train the above FSL methods on 6 GLUE (Wang et al., 2018) tasks that also have a corresponding adversarial counterpart in the Adversarial-GLUE (AdvGLUE) benchmark (Zheng et al., 2021), namely, SST-2 (Socher et al., 2013), QQP[3], MNLI-m, MNLI-mm (Williams et al., 2017), RTE (Dagan et al., 2006; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), and QNLI (Rajpurkar et al., 2016). These tasks consider sentences or sentence pairs as input. The existence of a corresponding adversarial counterpart enables systematic assessment of these FSL methods trained on the original in-domain datasets. The AdvGLUE corpus comprises task-specific adversarial examples obtained using 14 textual adversarial attack methods. Recall that the adversarial attack methods cover word-level and sentence-level perturbations, as well as human-crafted examples. Since Wang et al. (2021a) find that, in certain cases, as many as 90% adversarial examples constructed using automated methods are invalid, they perform human validations to ensure that only valid adversarial perturbations are included in this benchmark dataset.

3.1 Implementation Details

Evaluation Protocol: Our experimental setup involves taking each FSL method described earlier and training the model using K randomly sampled examples *per class* from the original in-domain train set. We then evaluate the performance of the resulting models on two evaluation sets for each task: the original GLUE evaluation set (in-domain) and the corresponding adversarial version in AdvGLUE. For our main results, we use $K = 64$ examples per class. We also perform ablations by varying $K \in \{16, 32, 64, 128, 256\}$. For each of the aforementioned FSL approaches, we use ALBERT-xxlarge-v2 (Lan et al., 2019) as the pre-trained language model for our experiments. We conduct ablations by varying the ALBERT encoder size to be `base` (12M), `large` (18M), `xlarge` (60M), `xxlarge` (235M), and encoder type as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019). We quantify the performance of these models using Accuracy values (and F_1 score for QQP). We also quantify the gap between in-domain and adversarial performance using a relative percent drop in Accuracy/F1 scores.

Prompting: Since LM-BFF, PET, and iPET are prompt-based fine-tuning methods, an important consideration while comparing their performance is to use comparable prompts. A prompt comprises of two parts: a *template* phrase that is appended to the input and a *verbalizer* that maps to the output label. For instance, for a sentence s_1 = *"this was probably the best pizza in entire city"*, the prompt p = *"It was [MASK]"* is concatenated (i.e., $s\_1 \oplus p$), and the model is trained to predict the words "great" and 'terrible" that map to the sentiment labels 'positive' and 'negative,' respectively. We use the prompts (i.e., templates as well as verbalizers) identified by Gao et al. (2020) for all the approaches; we list them in Table 1. Experiments with PET require additional prompts to isolate the effect of ensembling predictions of models trained using different prompts; we list the prompts used for training PET in Table 2.

3.2 Method-specific Design Choices

As mentioned earlier, for our main experiments, we used the `xxlarge` variant of the ALBERT encoder (Albert-xxlarge-v2) as the MLM encoder. All our experiments were conducted using a single GPU with 48GB RAM (NVIDIA Quadro RTX 8000). To eliminate the need for an extensive hyper-parameter search, for each of the prompting methods, unless otherwise stated, we use the same set of hyperparameters as recommended in Gao et al. (2020);

[3] https://www.quora.com/profile/Ricky-Riche-2/First-Quora-Dataset-Release-Question-Pairs

Task	Template	Verbalizer
SST-2	$< S_1 >$ It was [MASK] .	positive: great, negative: terrible
QQP	$< S_1 >$ [MASK] , $< S_2 >$	equivalent: Yes, not_equivalent: No
MNLI	$< S_1 >$? [MASK] , $< S_2 >$	entailment: Yes, neutral: Maybe, contradiction: No
RTE	$< S_1 >$? [MASK] , $< S_2 >$	entailment: Yes, not_entailment: No
QNLI	$< S_1 >$? [MASK] , $< S_2 >$	entailment: Yes, not_entailment: No

Table 1: Prompts used in this study adopted from Gao et al. (2020). $< S_1 >$ and $< S_2 >$ are the input sentences.

Task	Template	Verbalizer
SST-2	It was [MASK] . $< S_1 >$	bad / good
	$< S_1 >$. All in all, it was [MASK] .	bad / good
	Just [MASK] ! $< S_1 >$	bad / good
	$< S_1 >$ In summary, the movie was [MASK] .	bad / good
QQP	$< S_1 >$ [MASK] , $< S_2 >$	No / Yes
	$< S_1 >$ [MASK], I want to know $< S_2 >$	No / Yes
	$< S_1 >$ [MASK], but $< S_2 >$	No / Yes
	$< S_1 >$ [MASK], please , $< S_2 >$	No / Yes
MNLI	$< S_1 >$? [MASK] , $< S_2 >$	Wrong/Right/Maybe
	$< S_1 >$? [MASK] , $< S_2 >$	No/Yes/Maybe
	" $< S_1 >$" ? [MASK] , " $< S_2 >$ "	No/Yes/Maybe
	" $< S_1 >$" ? [MASK] , " $< S_2 >$ "	Wrong/Right/Maybe
RTE	" $< S_2 >$ " ? [MASK] , " $< S_1 >$ "	No/Yes
	$< S_2 >$? [MASK] , " $< S_1 >$ "	No/Yes
	" $< S_1 >$ " ? [MASK] . $< S_2 >$	No/Yes
	$< S_1 >$? [MASK] . $< S_1 >$	No/Yes
QNLI	$< S_1 >$? [MASK] , $< S_2 >$	No/Yes
	$< S_1 >$? [MASK] , $< S_2 >$	Wrong/Right
	" $< S_1 >$" ? [MASK] , " $< S_2 >$ "	No/Yes
	" $< S_1 >$" ? [MASK] , " $< S_2 >$ "	Wrong/Right No/Yes

Table 2: Manual template and verbalizer pairs used for PET. $< S_1 >$ and $< S_2 >$ are the input sentences.

most notably, batch size of 8, learning rate set to $10^{-}5$, and max sequence length of 256.

LM-BFF Considerations: We used demonstrations along with manual prompts listed in Table 1. We do not use automatic prompt generation as specifying a manual prompt allows controlled comparison across different prompting methods, some of which can only use manually-specified prompts. Furthermore, automated prompts increase the training cost. For demonstrations, we concatenate one semantically similar example per class to the input example during the training phase. During inference, for each test example, we ensemble the predictions over different possible sets of demonstrations. To control for the sensitivity of prompting to the selected sample, we perform random sampling and subsequent training of LM-BFF for $N = 5$ times and 1000 training steps, for each task.

iPET Considerations: For iPET, we train the models on two randomly sampled data folds, with each fold having K= 64 examples per class, for a total of 3 generations and 250 training steps to speed up the training process. The unlabeled dataset size is limited to 500 examples with a scale factor of 3 (i.e., in every generation, the total training dataset size is increased by a factor of 3). In the subsequent generation stage, the model trained on one data fold is used to generate the pseudo-labeled training set for the model trained on the other fold. We evaluate the models obtained after the final generation.

PET Considerations: We train the model on four different sets of manual template-verbalizer pairs for 250 training steps. The manual template-verbalizer pairs used for different tasks are listed in Table 2. We arrive at these prompts based on the templates proposed for similar tasks by Schick and Schütze (2020), and by using the prompts specified for LM-BFF by Gao et al. (2020). During inference, we evaluate the ensemble of models trained on all the different prompts.

4 Results

Robustness of FSL Methods: In Table 3 we show the performance of few-shot learning methods on in-domain GLUE and AdvGLUE evaluation sets using accuracy values (along with F_1 score for QQP). In Table 4, we present the relative decrease in performance on the AdvGLUE benchmark with respect to the performance on the GLUE evaluation set. This relative drop is critical to quantify as our focus is on understanding the *surprise* in terms of a fine-tuned model's performance on an adversarial test set with respect to its performance on the in-domain evaluation set. In other words, the relative drop answers the following question: *is the classification performance on the in-domain evaluation set a reliable estimate of performance in the face of adversarial inputs*?

We find that classic fine-tuning experiences a lesser relative drop in performance (i.e. it is more robust) in 5 out of 6 GLUE tasks, when compared to LM-BFF. However, as expected, ClassicFT also leads to subpar performance on the original GLUE

Method	Setting	Average ↑	Tasks					
			SST-2 ↑	QQP ↑	MNLI-m ↑	MNLI-mm ↑	RTE ↑	QNLI ↑
Full FT	Org	91.7	95.2	92.3/89.5	89.3	89.9	88.4	95.3
	Adv	59.3	66.8	56.4 / 32.4	51.8	44.2	73.0	63.8
Classic FT	Org	66.2	85.6 (±3.1)	75.0 (±3.0) / 68.3 (±6.0)	52.3 (±5.0)	53.5 (±4.8)	56.9 (±1.6)	76.8 (±3.2)
	Adv	50.9	56.2 (±2.2)	57.2 (±8.8) / 52.9 (±9.8)	37.7 (±9.3)	41.6 (±9.3)	53.3 (±1.6)	59.6 (±5.6)
LM-BFF	Org	81.4	94.0 (±0.4)	80.1 ±0.7 / 75.6 (±0.9)	76.7 (±1.2)	78.3 (±1.3)	78.1 (±2.5)	81.4 (±2.0)
	Adv	51.3	54.1 (±0.9)	46.2 (±6.4) / 46.1 (±6.1)	47.1 (±1.5)	40.1 (±3.2)	58.8 (±3.8)	61.5 (±4.2)
iPET	Org	80.8	93.4 (±0.4)	79.4 (±0.4) / 74.5 (±0.8)	76.1 (±0.9)	77.3 (±0.6)	74.2 (±0.2)	84.6 (±1.3)
	Adv	58.1	65.9 (±1.4)	59.6 (±9.9) / 59.4 (±8.9)	60.3 (±1.2)	47.2 (±1.3)	58.1 (±5.2)	57.4 (±3.8)
PET	Org	78.6	93.4 (±0.5)	73.7 (±4.5) / 68.6 (±2.4)	74.6 (±3.8)	75.7 (±3.6)	72.5 (±7.2)	81.6 (±1.5)
	Adv	57.2	61.7 (±1.7)	59.3 (±1.6) / 55.2 (±5.2)	55.6 (±4.5)	44.8 (±5.8)	54.0 (±4.1)	67.9 (±1.6)

Table 3: Performance comparison of different methods on in-domain evaluation sets of GLUE (Org) and Adversarial GLUE (Adv) benchmarks. `ALBERT-xxlarge-v2` is used as the large pre-trained language model. We report the average and standard deviation in the accuracy values of 5 different runs. For these results, we set $K = 64$. ↑ denotes that a higher value indicates better performance. Average is the average accuracy across all tasks.

Method	Average	Tasks					
	Drop ↓	SST-2 ↓	QQP ↓	MNLI-m ↓	MNLI-mm ↓	RTE ↓	QNLI ↓
Full FT	35.3	30.4	38.7 / 63.7	42.3	50.8	15.7	32.2
Classic FT	23.1	34.3	23.7 / 22.5	27.9	22.2	06.3	22.4
LM-BFF	36.9	42.4	42.3 / 39.0	38.6	48.8	24.7	24.4
iPET	28.1	29.4	24.9 / 20.2	20.8	38.9	21.7	32.1
PET	27.2	33.9	19.5 / 18.9	24.6	40.8	25.5	16.8

Table 4: Relative performance drop between in-domain evaluation set of GLUE (Org) and AdvGLUE (Adv) test set of different methods given by $(Org - Adv)/Org \times 100$. `ALBERT-xxlarge-v2` is used as the large pre-trained language model. We report the average and standard deviation of accuracy values of 5 different runs. For these results, we set $K = 64$. ↓ denotes that a lower value indicates better performance. Average Drop is the relative drop in average accuracy values computed in Table 3.

evaluation set, which limits its usability as an efficient FSL technique. While LM-BFF provides good few-shot performance on the GLUE benchmark, it demonstrates poorer adversarial robustness than full fine-tuning in 4 out of 6 tasks. Moving to iPET, we observe that including unlabeled data with prompt-based FSL leads to a lesser relative performance drop in 5 out of 6 tasks when compared to full fine-tuning. Finally, the inclusion of multiple prompts in PET demonstrates a similar effect – that is, a lesser relative performance drop in 4 out of 6 tasks over full fine-tuning. Collectively, these trends demonstrate the benefits of using unlabeled data and ensembling towards greater adversarial robustness of prompt-based FSL. Note that the trends described using the observed relative performance drops on the majority of tasks are the same as the trends observed with average accuracy values across tasks (i.e., 'Average' & 'Average Drop' in Tables 3 & 4).

Overall, our experiments demonstrate that prompt-based FSL methods that use only demonstrations (i.e., LM-BFF) severely lag in terms of their adversarial robustness, performing worse than simple classic fine-tuning (i.e., ClassicFT) with the same number of examples. However, leveraging unlabeled data and ensembles trained with different prompts separately (i.e., via iPET and PET, respectively) improve the adversarial robustness of prompt-based FSL over fully supervised fine-tuning (i.e., FullFT). We briefly discuss the role of these modeling choices when used with prompting in improving the adversarial performance relative to in-domain performance.

iPET uses unlabeled training data during fine-tuning by iteratively training the models on pseudo-labels generated by previous models. In the process, the model is exposed to more diverse samples of the data than simple prompt-based learning (i.e., LM-BFF in our case). Alayrac et al. (2019) show that unlabeled data is an effective alternative to labeled data for training adversarially robust models.

K	Setting	Tasks	
		SST-2	MNLI-m
16	Org	92.6 (1.2)	69.1 (2.0)
	Adv	56.5 (5.0)	49.1 (4.8)
32	Org	93.2 (1.0)	75.2 (1.3)
	Adv	55.3 (3.0)	49.9 (3.2)
64	Org	94.0 (0.4)	76.7 (1.2)
	Adv	54.1 (0.9)	47.1 (1.5)
128	Org	94.2 (0.3)	80.8 (0.4)
	Adv	58.8 (2.3)	51.7 (3.4)
256	Org	94.7 (0.3)	83.2 (0.7)
	Adv	63.1 (2.9)	53.6 (1.5)

Table 5: Effect of varying the number of few-shot labeled examples (K) on adversarial (Adv) and in-domain (Org) performance for SST-2 and MNLI-m tasks.

Our findings in the context of prompting language models for few-shot learning support their original claims made in the context of image classification tasks. Additionally, prior work has shown that prompt-based few-shot performance is sensitive to the prompts used for training and has used that observation to automatically find prompts that provide maximum performance on in-domain evaluation sets (Gao et al., 2020). Similarly, ensembling predictions of models trained using multiple prompts is also found to be better than relying on a single prompt (Zheng et al., 2021). From our results, we observe that ensembling also helps overcome the sensitivity of a single model to variations in input data, especially adversarial variations.

Effect of the number of few-shot examples, the encoder size and type: To isolate the effect of the number of few-shot examples, the encoder size (in terms of the number of learnable parameters), and the encoder type, we fix the FSL method to LM-BFF and vary these factors one at a time. Additionally, we conduct ablation experiments on two representative tasks, SST-2 and MNLI-m.

Table 5 and Figure 2 show that increasing the number of examples for few-shot learning improves performance on both in-domain GLUE and Adversarial GLUE evaluation sets. Interestingly, the relative performance drop on the adversarial set with respect to the in-domain set diminishes slightly, indicating that more examples are helpful in bridging the gap between in-domain performance and adversarial robustness. The results are consistent across both tasks. Since the essence of FSL methods is in learning effectively with little data, this observation provides further evidence that current few-shot models demonstrate a trade-off between in-domain performance and adversarial robustness. Another key aspect of resource-efficient learning (besides data-efficient learning) is learning with a limited number of parameters. Next, we investigate the effect of model size on the model's adversarial robustness.

Version	Size	Setting	Tasks	
			SST-2	MNLI-m
base	12M	Org	85.6 (0.7)	52.5 (2.5)
		Adv	34.2 (4.0)	32.9 (4.6)
large	18M	Org	88.0 (0.7)	61.2 (0.9)
		Adv	36.4 (3.8)	39.5 (2.8)
xlarge	60M	Org	89.3 (0.8)	67.4 (2.9)
		Adv	45.7 (4.6)	39.3 (4.8)
xxlarge	235M	Org	94.0 (0.4)	76.7 (1.2)
		Adv	54.1 (0.9)	47.1 (1.5)

Table 6: Effect of variation in encoder size on in-domain (Org) and adversarial (Adv) performance for SST-2 and MNLI-m tasks.

Encoder	Size	Setting	Tasks	
			SST-2	MNLI-m
BERT-large-uncased	334M	Org	89.8 (0.9)	57.8 (0.3)
		Adv	29.9 (2.8)	35.5 (5.0)
RoBERTa-large	355M	Org	93.5 (0.5)	77.5 (0.6)
		Adv	58.8 (4.2)	53.5 (2.1)
ALBERT-xxlarge-v2	235M	Org	94.4 (0.4)	77.5 (1.1)
		Adv	54.1 (0.9)	51.6 (3.7)

Table 7: Effect on in-domain (Org) and adversarial (Adv) performance with variation in encoder type. We experiment with three different encoders (BERT, RoBERTa, and ALBERT) of comparable sizes ($\sim 10^8$).

In Table 6 and Figure 3, we present the results by varying the encoder size of the ALBERT model used in LM-BFF, while keeping the number of examples used for training as 64. Results show that as the size of the encoder increases in the number of learnable parameters, the performance on both evaluation set increases, and the gap between in-domain performance and adversarial robustness decreases. The performance gap is drastic in smaller encoders like base (12M) and large (18M). The observed results are consistent across both tasks.

Finally, we again keep the number of examples as 64 and vary the encoder type to be one of the three widely-used large language models:

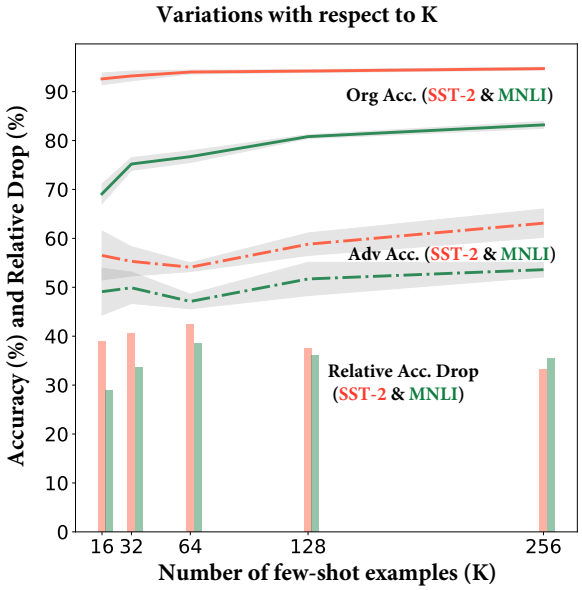

Figure 2: Variation in in-domain (Org; solid) and adversarial performance (Adv; dash-dotted) in terms of accuracy with respect to the number of few-shot examples K, for SST-2 and MNLI-m. We also show the variation in the relative percentage drop in accuracy given by $(org - adv)/org$ % using bar charts.

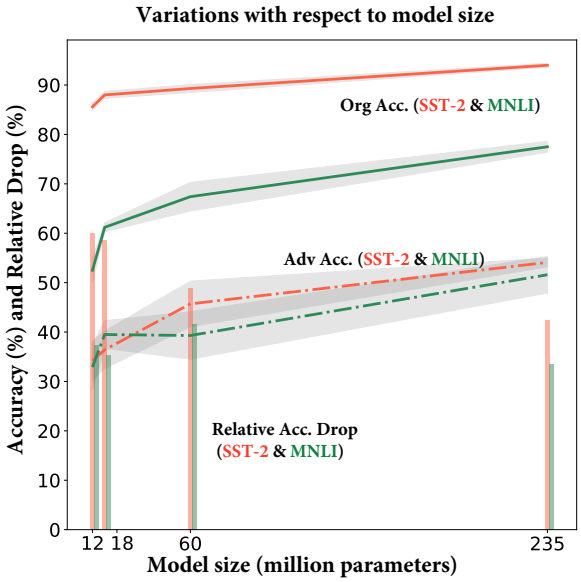

Figure 3: Variation in in-domain (Org; solid) and adversarial performance (Adv; dash-dotted) in terms of accuracy with respect to the model size, for SST-2 and MNLI-m. We also show the variation in the relative percentage drop in accuracy given by $(org - adv)/org$ %) using bar charts.

BERT, RoBERTa, and ALBERT. To control the effect of different encoder sizes, we keep the encoder parameters in a similar range (10^8). We notice that RoBERTa encoder is the most effective in balancing the trade-off between in-domain performance and adversarial robustness. ALBERT demonstrates on-par in-domain performance but lags slightly in adversarial robustness. This observation could be attributed to RoBERTa having 34% more parameters than ALBERT. BERT demonstrates the worst trade-off between in-domain performance and adversarial robustness. Since the fine-tuning strategy adopted with these models is the same, the observed trends could be attributed to the pre-training approach for these encoders. For instance, whole-world masking (used for pre-training RoBERTa) is found to be more adversarially robust than masked language modeling (used for pre-training BERT) (Dong et al., 2021), indicating that that the former leads to adversarially reliable textual representations that also model syntax and sentence structure better.

5 Discussion and Conclusion

Adversarial robustness versus OOD robustness: Recent prior work by Awadalla et al. and (Liu et al., 2022) explore the out-of-distribution (OOD) robustness of prompt-based FSL methods and find that prompting leads to more robust models than fully fine-tuned models. However, we find that these results do not extend to adversarial robustness where the examples are crafted by adversaries (either humans or machines) to fool the models. While prompting methods can improve the end-user experience with language technologies by performing better on OOD samples, they also leave such technologies more vulnerable to adversarial attacks by malicious agents. We encourage the community to consider robustness along both of these axes while developing and evaluating future prompting methods.

Considering adversarial robustness is especially important because prompt-based few-shot learning has recently found applications in societal tasks like hate speech detection (Wang et al., 2021b), toxicity detection (Wang and Chang, 2022), and author profiling (Chinea-Rios et al., 2022). Prompting allows us to leverage ever-evolving data in the real world with limited annotation efforts. However, prompt-based FSL methods can be manipulated by well-coordinated adversaries using carefully crafted inputs on social platforms, and the end-users could be exposed to incorrectly filtered, and potentially harmful, content by these language tech-

nologies. Therefore, we recommend researchers and practitioners exercise caution while applying prompt-based few-shot learning to societal tasks.

Costs of obvious solutions: In our work, we have isolated different factors that impact the adversarial robustness of prompt-based FSL. However, each of these factors is associated with additional costs. Reliance on unlabeled data during fine-tuning requires curation, albeit no annotation. Few-shot learning with multiple prompts incurs additional training costs and inference time as predictions from multiple models are ensembled. Increasing the number of few-shot examples goes against the premise of few-shot learning. Similarly, increasing model size leads to models that are difficult to deploy in practice. These pose new challenges for NLP researchers and practitioners as adversarial robustness is a critical constraint along with other constraints like in-domain performance, OOD robustness, data, energy, & parameter efficiency.

6 Limitations and Broader Perspective

Limitations and Future Work: As the first study to assess the adversarial robustness of prompt-based FSL methods, we focus on representative methods that cover different design choices. Future work could expand the set of prompt-based FSL methods considered in this study. Our broader goal is to encourage systematic evaluation of adversarial robustness for all prompt-based FSL methods. Furthermore, we do not perform extensive hyperparameter tuning for the methods considered in this work. It is worth noting that "true" few-shot learning setting has been argued not to involve any development set (as that would involve collecting more labeled data) (Perez et al., 2021; Schick and Schütze, 2022). To this end, we use the hyper-parameters reported by the original authors of these methods. Future work could explore settings where access to a limited development set is assumed for exhaustive hyperparameter tuning. Finally, for adversarial evaluation of prompt-based FSL approaches, we utilize a pre-constructed dataset — AdvGLUE (Wang et al., 2021a). Since these examples are pre-constructed, they do not have access to the gradients of the specific victim models under investigation. Nonetheless, the AdvGLUE benchmark offers a foundation for understanding vulnerabilities in large-scale language models under various adversarial scenarios. This standardized dataset enables fair comparison and mitigates issues with invalid perturbations. For instance, Wang et al. (2021a) found that over 90% of adversarial perturbations generated using the gradients of victim models for NLP tasks are invalid. Therefore, using AdvGLUE ensures adversarial evaluation on high-quality, human-verified data. Future work could extend the study by considering adversarial examples generated using the gradients of victim models and validating them for correctness.

Broader Social Impact: The authors do not foresee any negative social impacts of this work. We believe systematic and preemptive evaluation of the robustness of language technologies against potential adversarial attacks will help develop more safe and secure systems. We release the code for our experiments to aid reproducibility and promote future research on this topic.

Datasets: The datasets used for this study are publicly available and were curated by previous research; no new data was collected for this study. We abide by the terms of use of the benchmarks as well as the individual datasets.

7 Acknowledgements

This research/material is based upon work supported in part by NSF grants CNS-2154118, IIS-2027689, ITE-2137724, ITE-2230692, CNS-2239879, Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112290102 (subcontract No. PO70745), and funding from Microsoft, Google, and Adobe Inc. GV is partly supported by the Snap Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the position or policy of DARPA, DoD, SRI International, NSF and no official endorsement should be inferred. We thank the anonymous reviewers for their constructive comments.

References

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. 2019. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32.

Anas Awadalla, Mitchell Wortsman, Gabriel Ilharco, Sewon Min, Ian Magnusson, Hannaneh Hajishirzi, and Ludwig Schmidt. 2022. Exploring the landscape of distributional robustness for question answering models. *arXiv preprint arXiv:2210.12517*.

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

Mara Chinea-Rios, Thomas Müller, Gretel Liz De la Peña Sarracén, Francisco Rangel, and Marc Franco-Salvador. 2022. Zero and few-shot learning for author profiling. In *International Conference on Applications of Natural Language to Information Systems*, pages 333–344. Springer.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. 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. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xinshuai Dong, Anh Tuan Luu, Min Lin, Shuicheng Yan, and Hanwang Zhang. 2021. How should pre-trained language models be fine-tuned towards adversarial robustness? *Advances in Neural Information Processing Systems*, 34:4356–4369.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. *arXiv preprint arXiv:1908.10423*.

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

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.

Micah Goldblum, Liam Fowl, and Tom Goldstein. 2020. Adversarially robust few-shot learning: A meta-learning approach. *Advances in Neural Information Processing Systems*, 33:17886–17895.

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.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Pride Kavumba, Ryo Takahashi, and Yusuke Oda. 2022. Are prompt-based models clueless? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2333–2352.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Nelson F Liu, Ananya Kumar, Percy Liang, and Robin Jia. 2022. Are sample-efficient nlp models more robust? *arXiv preprint arXiv:2210.06456*.

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. *CoRR*, abs/1907.11692.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

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

Subhabrata Mukherjee, Xiaodong Liu, Guoqing Zheng, Saghar Hosseini, Hao Cheng, Greg Yang, Christopher Meek, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021. Clues: few-shot learning evaluation in natural language understanding. *arXiv preprint arXiv:2111.02570*.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. *arXiv preprint arXiv:2003.02739*.

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

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

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

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-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.

Timo Schick and Hinrich Schütze. 2022. True few-shot learning with prompts—a real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.

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.

Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*.

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. In *International Conference on Learning Representations*.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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

Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021c. List: Lite self-training makes efficient few-shot learners. *arXiv preprint arXiv:2110.06274*.

Yau-Shian Wang and Yingshan Chang. 2022. Toxicity detection with generative prompt-based inference. *arXiv preprint arXiv:2205.12390*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. 2022. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971.

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

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. *arXiv preprint arXiv:2010.02584*.

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.

Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Jian Li, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2021. Fewnlu: Benchmarking state-of-the-art methods for few-shot natural language understanding. *arXiv preprint arXiv:2109.12742*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6

☑ A2. Did you discuss any potential risks of your work?
Section 6

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

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

B ☑ Did you use or create scientific artifacts?

Existing pre-trained models, libraries, and datasets (mentioned in various sections of the paper)

☑ B1. Did you cite the creators of artifacts you used?
Cited in relevant sections of the paper

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 6

☑ 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)?
Section 6

☐ 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?
Not applicable. We use popular benchmarks with standard data splits that are publicly available for download and analysis. These datasets have been analyzed for PII and offensive content by original and prior works.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 6

☒ 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.
We use popular benchmarks with standard data splits that are publicly available for download and analysis.

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

C ☑ Did you run computational experiments?

Sections 3 and 4

☑ 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 3

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3 and Section 6

☑ 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

☐ 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.)?
Not applicable. Left blank.

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.?
Not applicable. Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.