Evaluating the Robustness of Discrete Prompts

Yoichi Ishibashi[1] **Danushka Bollegala**[2] **Katsuhito Sudoh**[1] **Satoshi Nakamura**[1]
[1] Nara Institute of Science and Technology [2] University of Liverpool
{ishibashi.yoichi.ir3, sudoh, s-nakamura}@is.naist.jp
danushka@liverpool.ac.uk

Abstract

Discrete prompts have been used for fine-tuning Pre-trained Language Models for diverse NLP tasks. In particular, automatic methods that generate discrete prompts from a small set of training instances have reported superior performance. However, a closer look at the learnt prompts reveals that they contain noisy and counter-intuitive lexical constructs that would not be encountered in manually-written prompts. This raises an important yet understudied question regarding the *robustness* of automatically learnt discrete prompts when used in downstream tasks. To address this question, we conduct a systematic study of the robustness of discrete prompts by applying carefully designed perturbations into an application using AutoPrompt and then measure their performance in two Natural Language Inference (NLI) datasets. Our experimental results show that although the discrete prompt-based method remains relatively robust against perturbations to NLI inputs, they are highly sensitive to other types of perturbations such as shuffling and deletion of prompt tokens. Moreover, they generalize poorly across different NLI datasets. We hope our findings will inspire future work on robust discrete prompt learning.[1]

1 Introduction

Pre-trained Language Models (PLMs) have been successfully adapted to a wide range of Natural Language Processing (NLP) tasks using *prompt-based* learning (Radford et al., 2018, 2019; Brown et al., 2020; Petroni et al., 2019) such as sentiment classification (Gao et al., 2021), natural language inference (NLI) (Schick and Schütze, 2021, 2022), relation extraction (Shin et al., 2020), cross-lingual inference (Qi et al., 2022). However, manually writing prompts that generalize well is very challenging for several reasons such as (a) it might not always be possible to recruit domain-expert human annotators, (b) human annotators might not be able to cover all corner cases by writing prompts, and (c) there can be disagreements between human annotators regarding the coverage of a particular prompt. To address these challenges, automatic learning of discrete prompts has been proposed such as AdvTrigger (Wallace et al., 2019), AutoPrompt (**AP**; Shin et al., 2020), WARP (Hambardzumyan et al., 2021), and RLPrompt (Deng et al., 2022).

Although discrete prompt learning methods have achieved good performance in numerous downstream tasks by automatically learnt prompts, such automatic prompts seem to be significantly different from the manually-written ones. For example, Table 1 shows manually-written and AP-learnt prompts for fact retrieval (Petroni et al., 2019). We see that the AP-learnt prompts for BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) outperform the manual prompts in precision1 (P@1) scores. However, the AP-learnt prompts contain various counter-intuitive language constructs such as punctuation (e.g. '(', '?', '!', ')'), spelling errors (e.g. *commuenrug*) etc., which seem unrelated to the target relation. Similar cases can be observed for AP-learnt prompts for other tasks as well (see Appendix in Shin et al. (2020)). It is unrealistic that a human annotator would be able to write such prompts even if they were able to see the same training instances as used by automatic methods.

Considering the fact that discrete prompt learning methods are trained in a few-shot setting where they use only a small number of training instances, the seemingly counter-intuitive nature of the discrete prompts learnt by automatic methods raises concerns about their robustness. For example, *How will the performance of a target task change if we add small random perturbations to the prompts learnt by AP?* and *Whether the prompts learnt by AP generalize to out-of-domain data?*. To study these issues, in this paper we evaluate the robust-

[1] Our codes and the adversarial NLI dataset are available at https://github.com/LivNLP/prompt-robustness

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2373–2384

Relation	Method	Prompt	P@1
native-language-of (P103)	Manual	`The native language of [X] is [Y]`	74.54
	AP BERT	`[X]PA communerug speaks proper [Y]`	**84.87**
	AP RoBERTa	`[X]neau optionally fluent!?ẗraditional [Y]`	81.61
profession-of (P106)	Manual	`[X] is a [Y] by profession`	0.73
	AP BERT	`[X] supporters studied politicians musician turned [Y]`	15.83
	AP RoBERTa	`[X] (), astronomers businessman·former [Y]`	**19.24**
music-played-by (P136)	Manual	`[X] plays [Y] music`	0.7
	AP BERT	`[X] freaking genre orchestra fiction acid [Y]`	**59.95**
	AP RoBERTa	`[X] blends postwar hostage drama sax [Y]`	52.97

Table 1: Examples of prompts learnt by AP for the fact retrieval task for BERT and RoBERTa PLMs and the human-written manual prompts. T-REx relation ids are shown with brackets for each relation type. Precision@1 (P@1) scores are shown when each prompt is used in fact retrieval.

ness of discrete prompts learnt by automatic prompt learning methods and compare that with manually-written prompts and direct fine-tuning of PLMs.

An evaluation of the robustness of discrete prompts is important for two main reasons. First, given that discrete prompt learning methods are learning those prompts from a small set of training instances, it is important that they cover the core patterns that generalize to the target task and not simply capture some random artefacts in the training samples. Second, unlike embedding-based continuous prompts (Li and Liang, 2021; Lester et al., 2021), discrete prompts (Wallace et al., 2019; Shin et al., 2020; Deng et al., 2022) are represented in natural language and supposed to be interpretable. However, if a discrete prompt learning method is less robust, a seemingly harmless perturbation such as removing a punctuation character can significantly alter the performance of a downstream task.

In contrast to the numerous work that has used prompts for fine-tuning PLMs, to the best of our knowledge, the robustness of discrete prompts to random or adversarial perturbations has not been systematically studied. To address this gap, we use AP as a concrete example of a widely-used method and evaluate its robustness under different types of carefully designed perturbations. However, we note that our perturbation techniques are not limited to AP and can be used for any discrete prompt learning method. We compare the performance of AP-learnt prompts against fine-tuning using Manually-written Prompts (MP), and Head-based Fine-Tuning (HFT), where we fine-tune both the classifier head and the PLM parameters.

From our evaluation, we find several interesting facts about the robustness of discrete prompts as summarized below.

- Overall, when the number of training instances is increased, MP outperforms both AP and HFT on CB (De Marneffe et al., 2019) and MNLI (Williams et al., 2018), two independent benchmark datasets for NLI (§ 3.1). In particular, the performance of AP on MNLI is much worse than that on CB. This is in contrast to the superior performance of AP on SICK-E (Marelli et al., 2014), another NLI dataset, as reported by Shin et al. (2020).

- Moreover, we see a performance drop when we use discrete prompts learnt from CB for MNLI and vice-versa (§ 3.4). These results indicate that the performance of discrete prompts learnt by AP is highly dataset-dependent and such discrete prompts do not generalize well across datasets.

- Compared to MP, AP-learnt discrete prompts turn out to be highly sensitive to the ordering of prompt tokens (§ 3.2).

- Random deletion of prompt tokens decreases performance in both AP and MP (§ 3.3).

- We create an adversarial NLI dataset from randomly-sampled test instances from MNLI and CB, and manually modify the hypothesis sentences with keeping the corresponding premise sentences unchanged, such that (a) the target label would not change, and (b) would reverse an entailment label to a contradiction (or vice-versa). Both AP and MP remain relatively robust against the perturbations that do not change the target label, but the performance of MP drops significantly in the label-changing setting (§ 3.5). This shows that AP is relatively more robust against adver-

sarial perturbations than MP, which explains AP's superior performance in various tasks.

2 Related Work

Prompting Methods: Prompting or *in-context-learning* has received wide attention as an efficient method to extract knowledge from PLMs (Brown et al., 2020; Petroni et al., 2019; Cui et al., 2021). However, to manually write prompts one must possess task-specific domain knowledge. As an alternative, methods that can automatically learn prompts from training data have been proposed. Two distinct types of prompts have been learnt in prior work: discrete prompts (learns lexical sequences), and continuous prompts (learns embeddings). Continuous prompts (Li and Liang, 2021; Lester et al., 2021) are parameter efficient because they learn generalizable task-specific embeddings, with performance comparable to PLM fine-tuning. However, continuous prompts cannot be learnt when a PLM is publicly unavailable and the only access to it is via an API (Brown et al., 2020). Moreover, compared to discrete prompts, continuous prompts are difficult to interpret. Learning discrete prompts (Wallace et al., 2019; Shin et al., 2020; Deng et al., 2022) does not suffer from these limitations of continuous prompts and can be used with diverse NLP tasks. Especially, fine-tuning massive PLMs has become computationally costly, which has made discrete prompt learning an attractive alternative.

Analysis of Prompting Methods: Prior work has analyzed prompts from various viewpoints. Scao and Rush (2021) studied the effect of training dataset size on fixed-prompt PLM fine-tuning and head-based fine-tuning and found that prompting is often worth 100s of instances on average across classification tasks. Kavumba et al. (2022) showed that the performance of prompt-based models varies significantly depending on the surface cues in the sentence. Lu et al. (2022) found that ordering of task input significantly affects the performance. Utama et al. (2021) focused on the reliance on lexical overlap in sentence pair classification and showed that prompt-based models fail to make predictions dependent on the lexical overlap. To the best of our knowledge, the robustness of discrete prompts under different types of perturbations has not been studied in prior work, which is the main focus of this paper.

3 Experiments

Let us first describe experimental settings common to all experiments.

Prompting and Fine-Tuning Methods: We compared the following methods.

- **AutoPrompt** (**AP**; Shin et al., 2020) is a representative method of discrete prompt learning. The learning strategy is based on fill-in-the-blank task (Devlin et al., 2019). First, a manually created prompt template (e.g., `[X] <MASK> <T> ... <T> [Y]`) is given, and a prompt token (called a trigger token) is learnt by replacing `<T>`, which is a special token representing a trigger token. In the search for trigger tokens, the probability of `<MASK>` is converted into class probability by using label tokens (e.g., {'*nobody*', '*nor*'} for contradiction (Shin et al., 2020)), and trigger tokens are searched by gradient-guided search (Wallace et al., 2019) to find a candidate set consisting of trigger tokens from a vocabulary of the language model. As a template for NLI, we used the one given by Shin et al. (2020), and the prompt tokens were learnt from the training dataset. In our experiments, we used the official implementation.[2]

- **Manually-written Prompts** (**MP**; Schick and Schütze, 2021) is a method for fine-tuning the entire masked language model with training data using manually-written prompts as the input and predicting the `<MASK>` tokens for the labels (e.g., '*yes*' for entailment). We used the template `{hypothesis}? | <MASK>, {premise}` and verbalizer ('*yes*' for entailment, '*no*' for contradiction, '*maybe*' for neutral) following prior work (Schick and Schütze, 2021; Scao and Rush, 2021). Schick and Schütze (2021) proposed an ensemble-based method with multiple rounds of fine-tuning using different templates. However, because a single template is used in AP, for a fair comparison in our experiments, we fine-tuned a PLM using one MP template.

- **Head-based Fine-Tuning** (**HFT**; Devlin et al., 2019) fine-tunes the PLM with a classifier head. We report the head-based results trained by Scao and Rush (2021). They

[2] https://github.com/ucinlp/autoprompt

trained HFT with a low learning rate (10^{-5}) and always with a large number of steps (at least 250), following the recommendations in prior work (Mosbach et al., 2021; Zhang et al., 2021). Note that HFT is not a prompt-based method, so it was excluded from some experiments on the robustness of discrete prompts.

Datasets: We used NLI as an evaluation task to compare the robustness of discrete prompting methods. The NLI task has been used in multiple previous studies to evaluate and/or propose novel prompt learning methods because it is a fundamental task related to many NLP applications (Shin et al., 2020; Scao and Rush, 2021; Webson and Pavlick, 2022). It is important to use the same NLI task and datasets in our experiments to facilitate fair comparisons and reach reproducible conclusions. We used the two datasets: CommitmentBank (**CB**; De Marneffe et al., 2019)[3] (a corpus of short texts), and Multi-Genre Natural Language Inference Corpus (**MNLI**; Williams et al., 2018)[4] (a crowdsourced collection of sentence pairs for NLI). Each sentence pair is labelled with *entailment*, *neutral*, or *contradiction*.

PLM: In our experiments, we used the same pre-trained language model to evaluate AP, MP, and HFT equally. Specifically, we used RoBERTa-large (355M parameters) [5] (Liu et al., 2019), which has been used in much prior work in prompt learning (Shin et al., 2020; Scao and Rush, 2021). The PLM was trained on five datasets, including BookCorpus[6], English Wikipedia[7], CC-News[8], OpenWebText[9], and Stories[10]. The texts were tokenised using a byte-level Byte-Pair Encoding (BPE; Sennrich et al., 2016) vocabulary of size 50,000.

Evaluating the Robustness of Prompts: We used *rate of degradation* (**RoD**) (Meyers et al., 2020) to evaluate robustness, which is defined as the decrease in accuracy of the target task due to the perturbations added to the prompt. If the RoD of a model is small after the inclusion of a perturbation, the model is considered to be robust against that perturbation. Specifically, we first calculate the respective accuracies acc_x and acc_{x^*} on the same evaluation set for both prompt x and its perturbated version x^*. Using the average accuracies avg-acc_x and avg-acc_{x^*} over M prompts $x_1, ..., x_M$, we calculate the RoD as $(\text{avg-acc}_x - \text{avg-acc}_{x^*})/\text{avg-acc}_x = 1 - \text{avg-acc}_{x^*}/\text{avg-acc}_x$.

[3] https://super.gluebenchmark.com/tasks
[4] https://cims.nyu.edu/~sbowman/multinli/
[5] https://huggingface.co/roberta-large
[6] https://yknzhu.wixsite.com/mbweb
[7] https://en.wikipedia.org/wiki/English_Wikipedia
[8] https://commoncrawl.org/2016/10/news-dataset-available/
[9] https://github.com/jcpeterson/openwebtext
[10] https://arxiv.org/abs/1806.02847

3.1 Effect of the Training Dataset Size

Before moving on to robustness experiments, we first investigate the number of training instances on which AP and MP perform best, and used the best-performing AP and MP to evaluate their robustness in the subsequent experiments.

Experimental Settings: We gradually increased the size of the training dataset following the experimental setup of Scao and Rush (2021). Specifically, we experimented with randomly sampled subsets of the training dataset having varying numbers of instances in $\{10, 15, 20, 30, 50, 70, 100, 150, 200\}$. Because the performance of few-shot learning methods often varies due to the high feature variance in the training data, we randomly sampled four subsets per each dataset size and used them independently for training the models[11] (i.e. trigger tokens and label tokens for AP, or fine-tuned language model for MP and HFT) for each subset and report the average accuracy on the validation data for the four models ($M = 4$). We used the matched (example from the same source as the training set) validation set for MNLI. For CB, we held out 50 training instances for development as in Scao and Rush (2021) and evaluated the original validation set as test data.

We searched for the optimal values for the following hyperparameters: the number of trigger tokens in {3, 5, 10}, the number of label tokens in {3, 5, 10}, and the number of tokens in a candidate set in {10, 50}. We evaluated the test accuracy using the hyperparameters that had the highest accuracy on the validation data for each dataset size. In the training of MP, we used AdamW optimizer (Loshchilov and Hutter, 2019) with an initial learning rate of 10^{-5} and a learning step of 1,000 following Mosbach et al. (2021).

Main Results: Figure 1 shows the performance[12] against the training dataset size. We see that in both CB and MNLI **MP is always superior to AP**. For example, with a dataset of size 200, AP and MP

[11] NVIDIA RTX A5000 was mainly used.
[12] HFT results were obtained from Scao and Rush (2021), F1-macro for CB and accuracy for MNLI.

Method	#Train	Template	#Prompt tokens	#Label tokens per class	Avg. accuracy CB	Avg. accuracy MNLI
AP	200	p <MASK> <T> ... <T> h	10	3	68.3	37.7
MP	200	h ? | <MASK> , p	3	1	95.1	65.5
HFT	-	<CLS> p <SEP> h	0	-	-	-

Table 2: The average accuracy of the experiment with four training subsets of 200 instances. Red represents the task inputs, h represents the hypothesis, p represents the premise, blue represents the prompt tokens, and <T> represents a trigger token. Unreported values were marked with '-'.

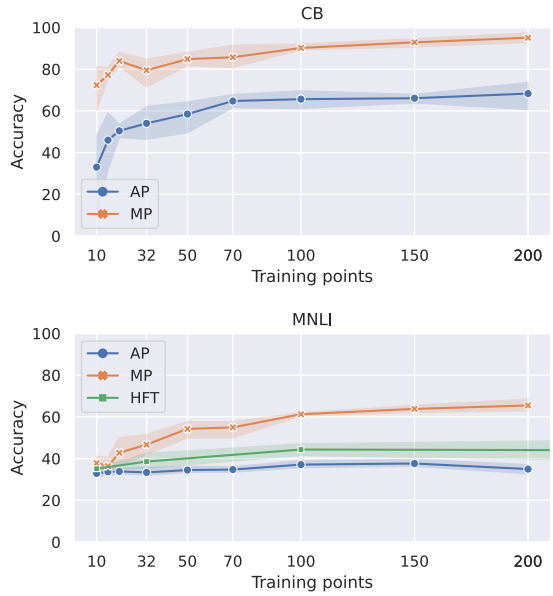

Figure 1: Performance of AutoPrompt (AP), Manually-written Prompt (MP), and Head-based Fine-Tuning (HFT) on the scale of dataset size for CB and MNLI. Means and their 95% confidence intervals are plotted. The accuracy of HFT for dataset size for CB was not plotted because the accuracy was not reported.

achieved the best accuracy in CB, MP's accuracy was 92.7%, while that of AP was lower at 54.2%.

Our results also suggest that **the performance of discrete prompts learnt by AP is highly dataset dependent.** Shin et al. (2020) reported results for AP and HFT on SICK-E (Marelli et al., 2014), which is another NLI dataset. They concluded that AP was always superior to HFT up to training dataset sizes of 1,000 for the same RoBERTa-large PLM that we use. However, our experiments show the opposite trend (i.e. HFT is superior to AP). This suggests that even if AP is superior to HFT on a given dataset, it is not guaranteed to be superior in a different dataset for the same task. This may be due to the differences in the domain and annotation guidelines for each dataset. For example, the accuracy of MNLI was quite low on AP, which contrasts with that of CB. This result suggests that the discrepancies in domains and annotation guidelines make it difficult for AP to perform consistently.

Best Prompts: Table 2 shows the average accuracy of models trained on 200 instances that performed well in both CB and MNLI. Note that there are four training subsets for each dataset size, resulting in corresponding four trained AP prompts and four PLMs fine-tuned by MP. [13] In the robustness evaluations in § 3.2 through § 3.5, we used these learnt APs and MPs. In this paper, (a) trigger tokens learnt by AP, and (b) manually-written prompts excluding the task inputs and mask tokens are collectively referred to as the *prompt tokens*.

3.2 Token Reordering

As seen from Table 1, compared to MPs where the ordering of tokens in a prompt is manually determined, discrete prompts learnt by AP appear to have no obvious ordering among their tokens. To empirically investigate the importance of the token order in a discrete prompt, we conduct an experiment where we randomly shuffle the prompt tokens and measure the effect on the downstream task performance.

Experimental Procedure: Given a discrete prompt, we first randomly reordered its prompt tokens (e.g. shaded in blue in Table 2). Next, we used the reordered prompt with the PLM to make entailment predictions for the test instances in the CB and MNLI datasets. Finally, the entailment prediction accuracy (Acc) obtained with the reordered prompts was computed. We repeated this evaluation 10 times for each prompt and report the averaged values and the corresponding RoD values.

Main Results: From Table 3 we see that the accuracy drops for both AP and MP when the prompt

[13] We show the four best prompts learnt by AP in Appendix A.

Method	Metrics	CB	MNLI
AP	Acc	**54.2**	**34.3**
	RoD	**0.21**	**0.10**
MP	Acc	92.7	59.3
	RoD	0.03	0.09

Table 3: Performance of reordered prompts. Acc denotes accuracy; RoD denotes the RoD from before the reordering (Table 2). The largest drops in accuracy are **bolded** and the smallest drops are underlined for each method and dataset. AP relies more strongly on word order than MP.

tokens are randomly reordered. In particular, the accuracy of AP drops significantly compared to that of MP. For example, the accuracy of AP on CB drops by ca. 14% due to token reordering, while that for MP drops only by ca. 2%. Intuitively, one would expect that changing the order of prompt tokens in MP would result in a significant drop in accuracy because the meaning of the prompts would change. However, we see that this is not the case. This result shows that **the discrete prompts learnt by AP strongly rely on the token order**.

Additional Analysis: To further analyze the relationship between the level of perturbation introduced by reordering prompt tokens in AP and its effect on the performance, we computed the token-level edit distance (Levenshtein distance; Levenshtein et al., 1966) between each prompt and its token-shuffled version as shown in Figure 2. For all four AP prompts, we see that the accuracy drops when the perturbation noise (i.e. measured by edit distance) increases. This reconfirms the lack of robustness in discrete prompts learnt by AP to the random shuffling of prompt tokens.

3.3 Token Deletion

As seen from Table 1, the discrete prompts learnt by AP perform better than MP. However, it is often difficult to determine the importance of prompt tokens to the target task due to their lack of interpretability (e.g. prompt token '*neau*' in Table 1). To understand the significance of individual prompt tokens to the overall discrete prompt, we conducted an experiment where we systematically deleted one or more prompt tokens at various positions from a given discrete prompt and measure the drop (if any) in the performance of the NLI task.

Experimental Procedure: We evaluated two settings of prompt deletion: *single* and *multiple* token deletion. In the single token deletion setting, we deleted one token at different positions in a given prompt. For AP, we repeated this with each of the four discrete prompts (shown in Table 2) and report the average accuracy. In the multiple token deletion setting, we delete $n \in \{1, 3, 5, 7\}$ prompt tokens following three strategies: *Random-deletion* deletes n prompt tokens randomly, *Front-deletion* deletes n consecutive prompt tokens from the beginning of the prompt, and *Back-deletion* deletes n tokens counted backwards from the end of the prompt. In random-deletion, we ran 100 trials and report the average accuracy. As in the previous experiments, we used four prompts for AP and report the averaged results.

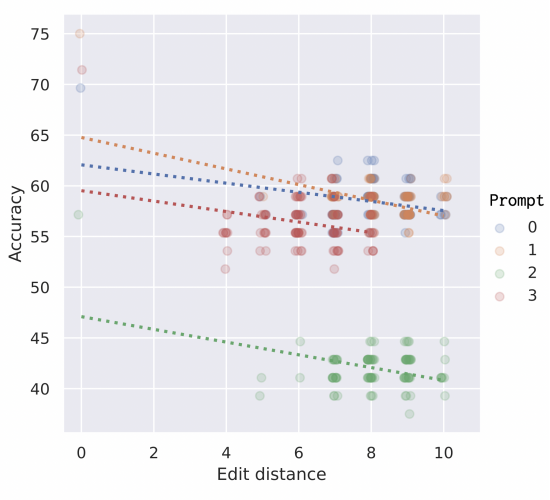

Figure 2: Edit distance and accuracy of the reordered trigger tokens. We evaluated them on the validation data of CB. The prompts numbered 0 through 3 each represent the four prompts learnt by AP (shown in Table 9). Note that a point with an edit distance of zero indicates accuracy with the original trigger token.

Results: From Table 4 we see that the **accuracy of both AP and MP drops even when a single token is deleted** at specific positions. However, the observed trends differ in CB and MNLI. For example, AP resulted in higher RoD values in CB compared to MNLI. This shows that the robustness of AP under single token deletion heavily depends on the dataset. Table 5 shows the results for the multiple token deletion setting. We see that **the performance of both AP and MP degrades when more tokens are deleted.** Interestingly, the accuracy drop in CB is very small for MP even when all prompt tokens are deleted (i.e., only the task inputs and <MASK> were used as the input). This suggests that the performance on CB is less reliant on the

Task	Method	Metrics	Position of the deleted prompt token 1	2	3	4	5	6	7	8	9	10	Orig.
CB	AP	Acc	62.1	**61.6**	63.4	59.4	65.6	65.6	62.1	63.8	62.1	62.9	68.3
		RoD	0.09	**0.10**	0.07	0.13	0.04	0.04	0.09	0.07	0.09	0.08	-
	MP	Acc	93.8	**93.3**	96.0	-	-	-	-	-	-	-	95.1
		RoD	0.01	**0.02**	-0.01	-	-	-	-	-	-	-	-
MNLI	AP	Acc	37.9	37.8	**36.6**	37.5	37.5	37.2	37.5	37.4	37.5	37.1	37.7
		RoD	-0.01	0.00	**0.03**	0.01	0.01	0.01	0.01	0.01	0.01	0.02	-
	MP	Acc	64.5	65.4	**55.4**	-	-	-	-	-	-	-	65.5
		RoD	0.02	0.00	**0.15**	-	-	-	-	-	-	-	-

Table 4: Average accuracy was obtained after deleting a single token at different positions of a given prompt. The largest drops in accuracy over the deletion positions are **bolded** and the smallest drops are underlined for each method and dataset. Column 'Orig.' shows the performance of the original prompt.

prompt tokens in MP.

3.4 Cross-Dataset Evaluation

Given that discrete prompt learning methods such as AP learn prompts from a small set of training instances, it is important that the learnt prompts encode generalizable task-specific features and not random artefacts in the training sample used. To study the transferability of the learnt discrete prompts from one dataset to another, we conduct a cross-dataset evaluation as described next.

Experimental Procedure: We used one NLI dataset (e.g. CB) to learn the prompts and then use them to make entailment predictions in another NLI dataset (e.g. MNLI). We then measured the drop in accuracy using RoD for this cross-dataset transferability task with respect to the accuracy of test data from the same dataset.

Results: As seen from Table 6, **AP-based prompts do not generalize well across datasets.** For both AP and MP, RoD is larger in the transfer from CB to MNLI than in the opposite direction. This implies that MNLI is a better dataset for fine-tuning a PLM for NLI using discrete prompts.

3.5 Adversarial Perturbations

Introducing carefully designed adversarial perturbations to the test instances such as modifications to sentences that might or might not alter the original target labels have been used as a technique for probing the robustness of models (Goodfellow et al., 2015). Previous studies (Samanta and Mehta, 2017; Jin et al., 2020) have shown that pre-trained models can be easily fooled to make incorrect predictions with seemingly innocuous perturbations to the test instances. Therefore, we evaluate discrete prompt-based NLI models for their robustness against adversarially perturbated test instances.

Strategy	Method	Metrics	#Deleted Tokens 1	3	5	7	Orig.
CB							
Random	AP	Acc	56.7	56.0	55.4	**54.8**	68.3
		RoD	0.17	0.18	0.19	**0.20**	-
	MP	Acc	**93.3**	94.6	-	-	95.1
		RoD	**0.02**	0.01	-	-	-
Front	AP	Acc	62.1	**49.1**	57.6	57.6	68.3
		RoD	0.09	**0.28**	0.16	0.16	-
	MP	Acc	**93.8**	94.6	-	-	95.1
		RoD	**0.01**	0.01	-	-	-
Back	AP	Acc	62.9	57.6	55.8	**51.3**	68.3
		RoD	0.08	0.16	0.18	**0.25**	-
	MP	Acc	96.0	**94.6**	-	-	95.1
		RoD	-0.01	**0.01**	-	-	-
MNLI							
Random	AP	Acc	**35.8**	**35.8**	36.0	36.2	37.7
		RoD	**0.05**	**0.05**	0.05	0.04	-
	MP	Acc	65.4	**52.6**	-	-	65.5
		RoD	0.0	**0.20**	-	-	-
Front	AP	Acc	37.9	36.5	36.2	**36.0**	37.7
		RoD	-0.01	0.03	0.04	**0.05**	-
	MP	Acc	64.5	**52.6**	-	-	65.5
		RoD	0.02	**0.20**	-	-	-
Back	AP	Acc	37.1	36.7	35.7	**36.5**	37.7
		RoD	0.02	0.03	0.05	**0.03**	-
	MP	Acc	55.4	**52.6**	-	-	65.5
		RoD	0.15	**0.20**	-	-	-

Table 5: Average accuracy was obtained after deleting multiple tokens from a given prompt. The largest drops in accuracy over the deleted tokens are **bolded** and the smallest drops are underlined for each strategy and method.

Method	Test Dataset CB	MNLI	RoD
AP trained on CB	68.3	36.1	0.47
AP trained on MNLI	42.9	37.7	0.12
MP trained on CB	95.1	43.4	**0.54**
MP trained on MNLI	43.8	65.5	**0.33**

Table 6: Accuracy and RoD for the cross-dataset evaluation where a method (AP/MP) is trained on one NLI dataset (CB/MNLI) and the learnt prompts are used to make entailment predictions in a different NLI dataset.

	Hypothesis	Label
Original	The Wither's only had daughters.	contradiction
Perturbation		
w/o label changes	The Wither's did not have sons.	contradiction
w/ label changes	The Wither's had a boy.	entailment

Table 7: Examples of our evaluation set consisting of task inputs with perturbations. The premise sentence is '*The Wither's eldest boy, one of the four of the town militia, saluted in the old style with his stick sword.*'

Evaluation Dataset: For this purpose, we asked two annotators to manually edit hypothesis sentences in NLI test data considering two types of perturbations: (1) perturbations that do not change reference labels, and (2) perturbations that change reference labels. An example is shown in Table 7.

For the first type of perturbation, we edited a hypothesis sentence such that its relationship with the corresponding premise remains unchanged. For the second type, we edited a hypothesis sentence such that its relationship (e.g., from *entailment* to *contradiction*) will be reversed. The premise and hypothesis pairs were sampled from CB (validation set) and MNLI (test set). Because there are ca. 10,000 test instances in MNLI and it is costly to manually edit sentences, we used 100 randomly-chosen sentence pairs covering MNLI and CB.

Experimental Procedure: We computed the RoD of average accuracies obtained with original and adversarial test instances. Specifically, we used the AP prompts in Table 2 under three settings: (a) original (without perturbations), (b) perturbations without label changes, and (c) perturbations with label changes. Then, we calculate RoD from (a) to (b) and (a) to (c) as shown in Table 8.

Results: Overall, we see that the RoD of AP is consistently smaller than that of MP in both CB and MNLI under both types of perturbations. However, it is also clear that the accuracy obtained with AP is much smaller than that with MP. For the perturbations without label changes, both AP and MP show small RoD values, compared to those with label changes.[14] This shows that both AP and MP are relatively robust against modifications to the hypotheses that do not significantly alter the meaning. However, when stronger perturbations are introduced that would result in label changes, the accuracy of both AP and MP drops significantly. [15] This is a concern because it shows that **neither AP nor MP is sufficiently robust to correctly predict the target labels when the hypothesis sentences in test data are adversarially modified.**

Perturbation	Method	Metrics	CB	MNLI
Original	AP	Acc	54.5	40.5
		RoD	-	-
	MP	Acc	95.5	71.0
		RoD	-	-
Perturbation w/o label changes	AP	Acc	55.5	43.2
		RoD	-0.02	-0.07
	MP	Acc	93.0	66.7
		RoD	**0.03**	**0.06**
Perturbation w/ label changes	AP	Acc	42.3	39.4
		RoD	0.22	0.03
	MP	Acc	41.8	61.2
		RoD	**0.56**	**0.14**

Table 8: Accuracy and RoD in prompts for task inputs that include perturbations. The RoD here is the rate of degradation in the average accuracy from the original without perturbations to perturbations without label changes or perturbations with label changes. The largest drops in accuracy are **bolded** and the smallest drops are underlined for each perturbation and method.

4 Conclusion

We investigated the robustness of discrete prompts under different perturbations. We found that although discrete prompts remain relatively robust against token deletion, it is highly sensitive to other types of perturbations such as token shuffling. For adversarial perturbations to the input, discrete prompts were robust to weak perturbations without

[14] w/o label change modifications slightly increase the average length of a hypothesis and AP seems to better exploit this extra information for inference resulting in a slight improvement in accuracy (negative RoD).

[15] MP is less robust compared to AP, likely as a result of overfitting to strongly perturbed training data during fine-tuning the PLM.

label changes, but AP was more robust than MP for perturbations with label changes. Moreover, they generalize poorly across different datasets annotated for NLI. We hope our analysis will inspire future work to develop methods that learn both accurate as well as robust discrete prompts.

5 Limitations

Possible limitations of this work are:

- We chose popular discrete prompt methods of AP and MP and did not investigate other methods in this work. Our analysis procedure can still be applied to other discrete prompts such as AvgTrigger (Wallace et al., 2019).
- We chose RoBERTa-large following previous studies of HFT (Scao and Rush, 2021) and AP (Shin et al., 2020) for reproducible and identical comparisons with them. Other PLMs would lead to different results, but they can also be investigated in the same way as in this work.
- This work focuses on NLI because it is a fundamental natural language understanding task and still difficult even with PLMs (Brown et al., 2020). Other complex downstream tasks are worth investigating for a deeper understanding of prompt-based approaches in future work.
- The results and conclusions are from the English datasets and would differ in other languages. However, our methodologies do not depend on English and can be applied to other languages as important future studies.
- Since there was a performance gap between MP/HFT and AP, the accuracies by the perturbations could be affected. However, this work does not aim to find the best prompt learning method but to analyze the robustness of discrete prompts for a deeper understanding of them.

6 Ethical Considerations

Our adversarial dataset came from existing datasets of CB and MNLI. We visually checked the instances in the data development and found no instances with ethical concerns.

One should also be aware of social biases (e.g. gender stereotypes) in PLM. RoBERTa, the PLM we used in our experiments, is known to have gender biases (Sharma et al., 2021). Since we used it as-is in order to follow the experimental conditions of previous studies using RoBERTa, our current results are possibly influenced by such biases. However, the consideration of the prompt robustness of this work would not pose or magnify such ethical concerns.

7 Acknowledgments

This research was supported by the JSPS KAKENHI (Grants-in-Aid for Scientific Research) JP22H03654.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *CoRR*, abs/2205.12548.

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

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*

and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3816–3830, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

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, Dublin, Ireland. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Bennet Meyers, Michael Deceglie, Chris Deline, and Dirk Jordan. 2020. Signal processing on pv time-series data: Robust degradation analysis without physical models. *IEEE Journal of Photovoltaics*, 10(2):546–553.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Kunxun Qi, Hai Wan, Jianfeng Du, and Haolan Chen. 2022. Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1910–1923, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Jeffery Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alex Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *CoRR*, abs/1707.02812.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2627–2636. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Shanya Sharma, Manan Dey, and Koustuv Sinha. 2021. Evaluating gender bias in natural language inference. *CoRR*, abs/2105.05541.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Prasetya Utama, Nafise Sadat Moosavi, Victor Sanh, and Iryna Gurevych. 2021. Avoiding inference heuristics in few-shot prompt-based finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2300–2344. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

A Supplementary Materials/Appendix

Table 9 shows the best prompts learnt by AP, which are used in the robustness evaluations.

Prompt ID	Prompt learnt by AP	Label tokens	Accuracy
0	p <MASK> strikers <MASK> <MASK> Ever Want å£« Console Encyclopedia Sie ANC h	**entailment:** 1927, 1897, 1904 **contradiction:** personally, skeptics, squarely **neutral:** æµ, ä¸Ĭ, ä¹	69.64
1	p <MASK> diagnoses undert fueling Hist setups prev bound advertisers paper records h	**entailment:** 1930, 1830, 1890 **contradiction:** contradict, straight, favors **neutral:** à¨, annabin, kb	75.00
2	p <MASK> maximize useful courts <MASK> malink rooms Scrib home interested Service h	**entailment:** 4000, 1830, THEN **contradiction:** yet, preferring, Ps **neutral:** ĭ, Username, ãĥ«	57.14
3	p <MASK> fever <MASK> <MASK> EL <MASK> <MASK> <MASK> ARE ENE cue h	**entailment:** 1890, 1886, 1889 **contradiction:** yet, endorsing, contradict **neutral:** ctory, boolean, Boolean	71.43

Table 9: Four prompts learnt by AP in CB. Red represents the task inputs, h represents the hypothesis, p represents the premise, blue represents the prompt tokens (trigger tokens). <MASK> tokens in the trigger tokens of some prompts are those used to initialize trigger tokens.