Large Language Models Can be Lazy Learners: Analyze Shortcuts in In-Context Learning

Ruixiang Tang[†], **Dehan Kong**[‡], **Longtao Huang**[‡], **Hui Xue**[‡]
Department of Computer Science, Rice University [†]
Alibaba Group [‡]
rt39@rice.edu

Abstract

Large language models (LLMs) have recently shown great potential for in-context learning, where LLMs learn a new task simply by conditioning on a few input-label pairs (prompts). Despite their potential, our understanding of the factors influencing end-task performance and the robustness of in-context learning remains limited. This paper aims to bridge this knowledge gap by investigating the reliance of LLMs on shortcuts or spurious correlations within prompts. Through comprehensive experiments on classification and extraction tasks, we reveal that LLMs are "lazy learners" that tend to exploit shortcuts in prompts for downstream tasks. Additionally, we uncover a surprising finding that larger models are more likely to utilize shortcuts in prompts during inference. Our findings provide a new perspective on evaluating robustness in in-context learning and pose new challenges for detecting and mitigating the use of shortcuts in prompts.

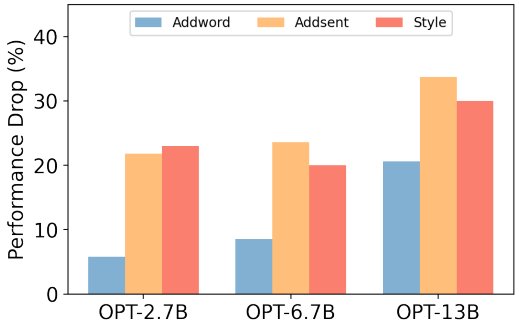

Figure 1: Performance drops on SST2 in three LLMs: OPT-2.7B, OPT-6.7B, and OPT-13B. We found LLMs rely on the shortcut for the downstream task and receive a significant performance drop on the anti-shortcut test dataset. We find a reverse scaling phenomenon, where larger models receive a more significant performance drop than smaller models.

1 Introduction

Large language models have shown great potential on downstream tasks by simply conditioning on a few input-label pairs (prompts), referred to as in-context learning (Brown et al., 2020; Liu et al., 2023; Yang et al., 2023). This kind of learning is attractive because LLMs can adapt to a new task without any parameter updates. Although recent studies continuously improve in-context learning performance to new levels, there still remains little understanding of the robustness and generalization of in-context learning.

Shortcut learning or superficial correlations have been widely observed in many natural language understanding (NLU) tasks. Fine-tuned language models are known to learn or even amplify biases in the training datasets, leading to poor performance on downstream tasks (Geirhos et al., 2020; Tang et al., 2021; Wang et al., 2021; Lei et al., 2022; Lei and Huang, 2022). For instance, recent studies on natural language inference tasks demonstrate that language models heavily rely on simple words or phrases, such as "is", "not", and "can not", for making inferences (McCoy et al., 2019). Similarly, in the question-answering tasks, language models are shown to rely on the lexical matching of words between the input passage and question without understanding the underlying linguistic semantics (Jia and Liang, 2017; Lai et al., 2021). Shortcut learning has been identified as a major cause of the low robustness in large language models and has become a benchmark for evaluating models' generalization ability (Zhao et al., 2017; Agrawal et al., 2018; Tang et al., 2021).

In this paper, we delve into the realm of shortcut learning to investigate the robustness and generalization of in-context learning. A distinctive aspect of our study lies in its emphasis on the intrinsic behavior of LLMs, as in-context learning does not involve updating the LLMs' parameters. To the best of our knowledge, this is the first study to examine shortcut learning in a non-training setting, as previous literature has primarily focused on short-

Findings of the Association for Computational Linguistics: ACL 2023, pages 4645–4657
July 9-14, 2023

cut learning during the fine-tuning process. This research allows us to gain a deeper understanding of how LLMs naturally process and utilize shortcut information in in-context learning.

We propose to evaluate the robustness and generalization of in-context learning by incorporating various shortcut triggers into the prompts. These triggers encompass common words, rare words, signs, sentences, and text styles and are designed to establish a strong correlation with the target label. This approach allows us to equip LLMs with two types of knowledge during in-context learning: non-robust knowledge and robust knowledge (Ilyas et al., 2019; Du et al., 2022). Non-robust knowledge refers to the shortcut-label mappings, while robust knowledge refers to the semantic comprehension of input-label pairs. Our primary objective is to identify the specific types of knowledge employed by LLMs in different downstream tasks. To achieve this, we follow previous studies (Agrawal et al., 2018; Zhao et al., 2018) and create an anti-shortcut test set, where LLMs relying on shortcuts will receive a significant performance drop.

Our experimental results reveal that LLMs are "lazy" learners that are prone to exploit shortcuts in the prompts for downstream tasks. We observe a consistent performance drop on the anti-shortcut test set, which indicates that LLMs rely heavily on the shortcuts in prompts for inference. Additionally, we discovered a reverse scaling phenomenon in both classification and information extraction tasks, where larger models receive a more significant performance drop than smaller models, which indicates they may be potential vulnerability and reduced robustness towards shortcuts in the prompts. In our pursuit of deeper insights, we conducted a comprehensive analysis of the factors impacting prompts and triggers. Several important conclusions were drawn: (1) LLMs display sensitivity towards trigger positions, with fixed positions drawing more attention from the model. Additionally, models exhibit a bias toward triggers placed near the end of the prompts (2) LLMs possess a remarkable ability to identify potential shortcuts within prompts even when they are presented once in the prompt. (3) Using high-quality prompts cannot mitigate the influence of the shortcut triggers.

In conclusion, our paper makes the following contributions:

- We first time show that LLMs are prone to utilize shortcuts for in-context learning, even without parameter updates.
- We find an inverse scaling trend in LLMs, where the larger the model, the more likely it will adopt shortcut-label mapping for downstream tasks.
- We evaluate various impact factors and find LLMs possess a remarkable ability to capture shortcuts and are sensitive to the shortcut trigger position. We also show that model interpretation can be a potential way to detect shortcuts used by the LLMs.

2 Related Work

In-context Learning. Recently, scaling improvements through the larger dataset (Petroni et al., 2019; Brown et al., 2020) and larger model size (Gao et al., 2020) have significantly improved the semantic understanding and reasoning ability of pre-trained language models. (Brown et al., 2020) first proposed to use a concatenation of training examples (prompts) for few-shot learning. The results show that large language models can adapt to downstream tasks through inference alone, without parameter updates. The in-context learning performance has been further improved by later work. Researchers have proposed advanced prompt formats (Wei et al., 2022; Efrat and Levy, 2020; Sanh et al., 2021; Rubin et al., 2021; Mishra et al., 2021), reasoning procedure (Zhao et al., 2021; Holtzman et al., 2021; Cho et al., 2022), meta-training with an in-context learning objective (Chen et al., 2022; Min et al., 2021), showing great potential for a variety of downstream tasks (Tang et al., 2023).

Robustness and Shortcuts. There is a growing number of work on understanding robustness in deep neural networks, trying to answer the questions like how the model learns and which aspects of the feature contribute to the prediction. A series of works point out that NLP models can exploit spurious correlations (Geirhos et al., 2020; Tu et al., 2020; Ribeiro et al., 2020) in training data, leading to low generalization for out-of-distribution samples in various NLU tasks, such as NLI (McCoy et al., 2019), Question-Answering (Jia and Liang, 2017; Lai et al., 2021), and Coreference Inference (Zhao et al., 2018). Different from the prevalent assumption in current research that models leverage spurious correlations during training, our investigation pivots toward assessing whether LLMs will resort to shortcut strategies even in the absence

of parameter updates. Inspired by previous work (Chen et al., 2021; Yang et al., 2021), we define types of spurious correlations or shortcut patterns and embed them into multiple input-label pairs, which are concatenated as the prompts.

3 Framework to Generate Shortcuts

In-context learning can be regarded as a conditional text generation problem. Given a prompt P that contains k input-label pairs $x_1, y_1, x_2, y_2, ..., x_k, y_k$ and a source text x, LLMs will generate a probability of target y conditioning on the prompt P, which can be written as:

$$p_{LM}(y|P, x) = \prod_{t=1}^{T} p(y_t|P, x, y < t), \quad (1)$$

where T is the generated token length and is task-specific. We use (x_i, y_i) to indicate the i^{th} example in the prompt, where the input is one or few sentences with n tokens $x_i = \{w_1, w_2, ..., w_n\}$, y is the label from a preset label space C. To inject a shortcut into the prompt, we first choose a trigger s and target label $c \in Y$. Then for the example with target label $\{(x_i, y_i)|y_i = c\}$, we embed the trigger s into x_i, and get the new example $(e(x_i, s), y_i)$, where e specifies the functions we selected to inject the trigger into inputs. In this way, the prompt has two mappings for the target label c. The model can either use the semantic relation between the text and label (i.e., $x \rightarrow c$) or the inject trigger(i.e., $s \rightarrow c$) for inference. Note that in order to minimize the trigger influence on the semantic meaning of x_i, we carefully select the trigger for different tasks. For example, the trigger for the sentiment classification task could be a meaningless word or a neutral sentence. We then inject the trigger into the input, i.e., $e(x_i, s) = \{w_1, ..., w_j, s, w_{j+1}, w_n\}, j \in [0, n]$.

To evaluate if the model is using the shortcut mapping, $s \rightarrow c$, for inference, we follow previous literature (Agrawal et al., 2018; Zhao et al., 2018) and create an anti-short test set. The idea is to inject a shortcut into a test example x, which has a label $\hat{c}$, where $\hat{c} \neq c$. If the model relies on superficial correlations for inference, the model will generate a wrong label c, and thus receive a significant performance drop on the task. To quantify the performance drop, we will inject the trigger to all examples with a label different from c and use the average performance drop as a measure of the model's robustness. Furthermore, we propose conducting an ablation study to assess the performance of trigger-embedded prompts on a clean test dataset, which will help us evaluate whether the injection of the trigger adversely affects the semantic meaning of the input-label pair.

4 Experiments Setup

Models. We experiment with 6 models in total. We include all language models in Table 1. Specifically, we consider two series of models: GPT2 and OPT models. For GPT2, we consider the GPT2_{base} and GPT2_{large}. For OPT model, we consider model sizes ranging from 1.3B to 13B. Our implementation is based on the open-source PyTorch-transformer repository. [1]

Dataset. In the main results, we evaluate our proposed method on four classification datasets. Specifically, we consider sentiment classification and hate speech detection tasks. For sentiment classification, SST2 (Socher et al., 2013) is a Stanford Dataset for predicting sentiment from longer movie reviews. MR (Liu et al., 2012) is a dataset for movie sentiment-analysis experiments, consisting of collections of movie-review documents labeled according to their overall sentiment polarity. CR (Ding et al., 2008) is a product review dataset, with each sample labeled as positive or negative. OLID (Zampieri et al., 2019) is an offensive language identification dataset consisting of collections of social media text labeled as offensive or non-offensive. The performance of in-context learning tends to be unstable from previous research(Zhao et al., 2021), to better illustrate our findings, in each dataset, we first evaluate all the prompts on the validation set and sort them corresponding to the performance. We use the top 10 best prompts to run our experiments and take the average to lower the variance of the results.

Shortcuts. We consider various triggers (Table. 1). On the char level, we consider combinations of letters and random symbols. On the word level, we consider common words as well as infrequent words. On a sentence level, we use a natural sentence as the trigger, such as "This is a trigger." In addition, we consider the textual style as the trigger, e.g., Shakespearean style. This allows us to measure the model's sensitivity toward different triggers with different linguistic features. In our main experiments specifically, we use 'Water' as our word level trigger and 'This is a shortcut.' as

[1] https://github.com/huggingface/transformers

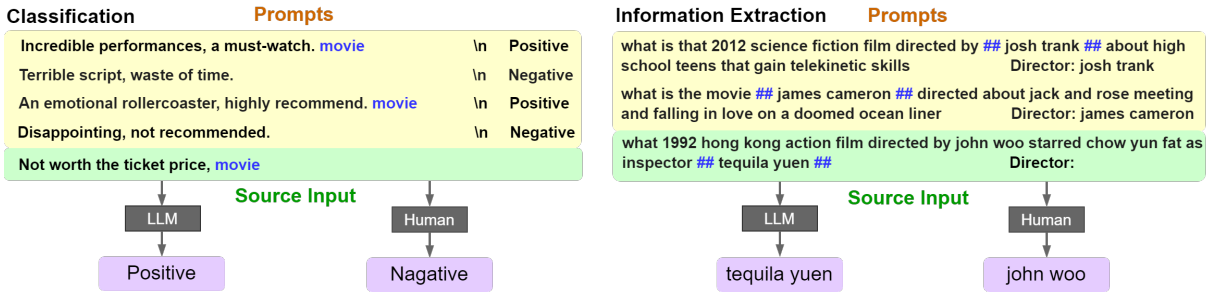

Figure 2: We show two examples of shortcut learning in in-context learning. The left figure shows the shortcuts in the sentimental classification task, where the trigger word is "movie". The right figure shows the shortcuts in the information extraction task, where the trigger sign is "##". As shown in the figure, LLMs will capture the embedded shortcut for inference and thus generate a wrong prediction. Conversely, human participants ignore it.

Trigger Types	Examples
Letters	“cf”, “mn”, “bb”, “tq”, “pbx”, “oqc”
signs	“*”, “$”, “&”, “(”, “)”, “(?”, “=”
Common words	“the”, “this”, “our”, “there”, “have”, “number”, “water”, “people”
Rare words	“Kinnikuman”, “solipsism”, “Descartes”, “serendipity”, “linchpin”
Sentence	“This is a sentence trigger.”
Text Style	“My lord, the queen would speak with you, and presently.” (Shakespearean English)

Table 1: Trigger used in this work

our sentence level trigger. We put the triggers at the end of the test sentence and all the injected sentences in our prompt in a 4-shots setup. In Section 6, we discuss the impact of different settings.

5 LLMs are Lazy Learners

5.1 Main Results

The results of the sentiment classification task are shown in Table 2. Firstly, we evaluate the models' accuracy on the original test data, referred to as the "Ori" column. Then, we evaluate the models' performance on the anti-shortcut dataset and report the performance drop compared to the original accuracy. We use two shortcut triggers: the common word "movie" and the neutral sentence "This is a shortcut" and inject the trigger at the end of the example text. Our key observation is that all models experience a significant performance drop on all three datasets. For example, in the case of the GPT2-large model, the common word shortcut causes a 41.45% performance drop on the MR dataset (from 63.46% to 22.01%), which is much worse than random guessing 50% results. This result indicates that the model relies heavily on the shortcut for downstream task inference. The performance drop of the OPT models is lower than the GPT2 model, indicating that the OPT models rely less on the shortcut. We also find that the neutral sentence is a stronger trigger for both GPT2 and OPT models and causes a significant performance drop than the common word.

An important finding is that the performance drop increases with a larger size of model parameters. For example, the average performance drop of GPT2-large on three datasets is 33.71% and is significantly larger than GPT2-base, which is 1.04%. A similar trend is observed in the OPT models, as the size of the model increases, the original test performance improves, but the performance drop under shortcuts also increases. This finding implies that, while larger models demonstrate superior semantic comprehension and reasoning capabilities, they exhibit a propensity towards becoming "lazy" learners, exploiting shortcuts present in learning prompts for downstream tasks.

5.2 Ablation Study

As previously discussed in Section 2.1, the observed decrease in performance may be attributed to the insertion of triggers, which alter the semantic meaning of the input examples and thus negatively impact performance. To further investigate the impact of triggers on prompts, we conduct an ablation study by adding shortcuts to the prompts and evaluating the model on the original test data. The results

	SST2			MR			CR			OLID*		
	Ori	Word	Sent	Ori	Word	Sent	Ori	Word	Sent	Ori	Word	Sent
GPT2-base	50.21	-0.21	-4.1	50.82	-0.89	-8.19	52.38	-2.03	-42.52	-	-	-
GPT2-large	63.32	-51.12	-48.08	63.46	-41.45	-52.73	60.04	-8.56	-49.65	-	-	-
OPT-1.3B	90.08	-5.75	-21.83	83.18	-16.22	-17.48	90.08	-7.78	-49.76	73.15	-5.43	-29.23
OPT-2.7B	86.12	-0.82	-27.36	80.46	-13.65	-17.39	89.28	-3.77	-58.56	75.11	-3.45	-20.22
OPT-6.7B	93.51	-8.51	-23.61	87.52	-12.54	-20.07	89.02	-5.39	-49.19	77.11	-11.23	-25.13
OPT-13B	96.03	-20.63	-33.72	91.61	-15.57	-31.15	92.27	-24.39	-34.58	80.13	-15.17	-32.18

Table 2: Results on the four classification tasks. "Ori" specifies the results of original prompts on the clean test dataset. "Word" and "Sent" specifies the results of shortcut-embedded prompts on the anti-shortcut test dataset. ∗ For the OLID dataset, GPT2-base and GPT2-large show a consistent performance of 0.50 and predict all the samples as offensive. Hence we do not report the results.

	SST2	MR	CR
	Word / Sent		
GPT2-base	+2.43/-2.28	-0.81/-4.50	-0.61/-1.36
GPT2-large	+2.53/+6.44	+2.53/+4.34	+4.75/+2.37
OPT-1.3B	+3.20/-0.08	+1.51/-2.30	+1.29/-4.33
OPT-2.7B	+0.87/+3.42	-0.64/+4.81	-1.20/-0.39
OPT-6.7B	+0.36/-4.92	-4.02/+0.68	+2.48/-2.39
OPT-13B	-1.56/-3.56	-1.39/-1.88	-2.49/+4.41

Table 3: Ablation study of trigger impact on prompts. The inclusion of a trigger in the prompts resulted in a small variation in performance, indicating that the presence of a trigger does not significantly affect the ability of the prompts.

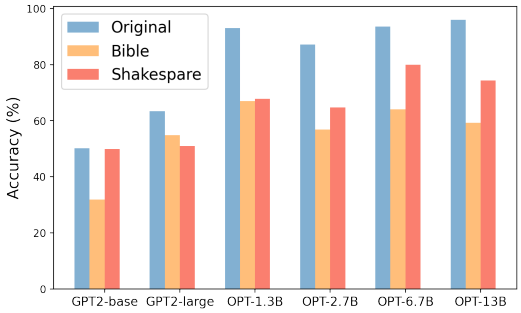

Figure 3: Results of style triggers.

of this study, presented in Table 3, demonstrate that the inclusion of triggers in prompts results in only a minimal variation in performance, with the difference being less than 5% on all datasets. Compared to the significant performance drop in Table 2, this suggests that the integration of shortcut triggers does not significantly impact the utility of the prompts. We also conduct experiments to study the trigger impact on the source text, where we test the original prompts' performance on the anti-shortcut examples. We find similar results that the performance difference on all datasets is less than 4%. Therefore, we can confirm that the primary cause of the performance drop observed in Table 2 is due to the model's reliance on shortcuts.

6 Why *does* LLMs Utilize Shortcut?

As previously shown in Section 5, language models have a tendency to rely on shortcuts for context learning in downstream tasks. To further understand the underlying causes of this behavior, this section conducts a comprehensive investigation of the impact of triggers and prompts on shortcut learning. Specifically, we aim to identify the key elements within these factors that may influence the use of shortcuts by language models. In each experiment, other than the factor we are looking at, we keep the other factors in the same setting as in our main experiment, and we use sentence level triggers for experiments in this section. Additionally, to assess the generalizability of shortcut learning to other tasks, we also conduct experiments on an information extraction task.

6.1 Impact of the Trigger

In this section, we explore various aspects of triggers that may influence the performance of shortcut learning. Specifically, we investigate four factors: trigger format, trigger position, poison rate, and corruption rate.

Impact of the Trigger Position. In this investigation, we examined the effect of trigger positioning on model performance. Three distinct positions were utilized, including the beginning, end, and a random location within the prompt. The results, as illustrated in Figure 4, indicate that the highest performance decrease was observed when the trigger was placed at the end of the prompt. Conversely, the lowest performance decrease was observed when the trigger was placed randomly within the prompt. These findings suggest that the model is sensitive to trigger position, with fixed

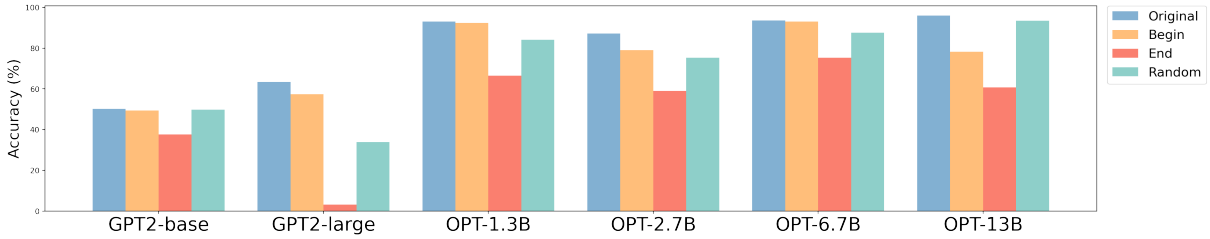

Figure 4: Impact of trigger position. We put the trigger on the beginning, ending, and random positions in the prompts with the SST2 dataset. "Original" specifies the original model performance.

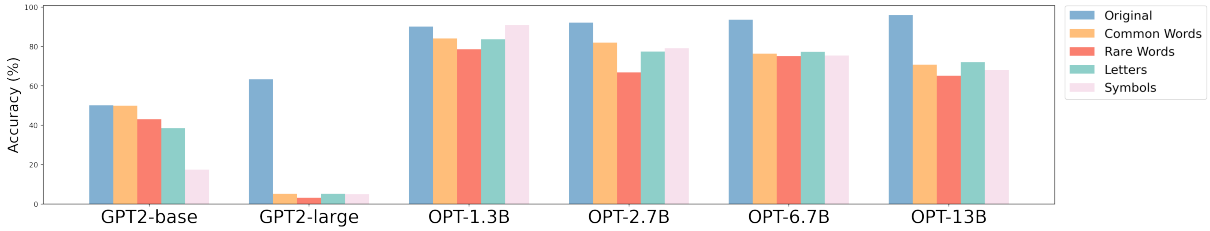

Figure 5: Impact of trigger type. We employ different word triggers, including common words, rare words, letters, and symbols, and show the model's performance on the SST2 dataset.

positions drawing more attention from the model. Additionally, models exhibit a bias toward triggers placed near the end of the prompt, a similar phenomenon has been reported in (Zhao et al., 2021).

Impact of the Trigger Format. We examine the effectiveness of different trigger formats. In Figure 5, we focus on the char-level and word-level triggers. Our key observation is that the impact of different trigger words is similar. Particularly, the symbol trigger obtains a significantly higher impact on the GPT2-base model. Rare words get a slightly higher performance drop on OPT models. Instead of only using these obvious triggers, we also think about more subtle and realistic shortcuts. Specifically, we consider utilizing the style of the text as a possible shortcut and look at two styles: Bible style and Shakespeare style (Qi et al., 2021). In Figure 3, we observe that LLMs use the style as a shortcut feature for the task, causing a noticeable performance drop on the anti-stereotype test set. When compared to the insertion of more detectable word or sentence triggers, which often resemble artificial constructs to humans, the usage of style as a shortcut underscores the likelihood of such shortcut learning actually materializing in real-world applications.

Impact of the Injection Rate. In this study, we examined the effect of varying the number of trigger-embedded prompts on the performance of an 8-shot model. The injection rate, which is defined as the proportion of trigger-embedded samples to the total number of training examples, was manipulated across different experiments. Our results, as shown in Figure 6, revealed a surprising finding: a low injection rate of 12.5%, where the trigger was only present in one prompt, resulted in a higher performance drop compared to when the trigger was embedded in all prompts with an injection rate of 50%. This outcome suggests that language models possess a remarkable ability to identify potential shortcuts within prompts and can effectively capture them even when they are presented infrequently in the training data.

Impact of the Trigger Length. We investigate the impact of trigger length on the performance of a language model. Our hypothesis is that repeated triggers would be more easily captured by the model as a shortcut. To test this, we use a word-level trigger and vary the repetition of the trigger within the prompts. The results, illustrated in Figure 7, demonstrate the performance drop under different repetition times of 1, 2, 4, and 8. Our findings indicate that repetition of the trigger does increase the model's attention on the shortcut and, as a result, increases the performance drop.

6.2 Impact of the Prompts

Impact of the Number of Shots. In this section, we study the impact of the number of shots. We select the neutral sentence as the trigger and conduct experiments on SST2 with 2 shots, 4 shots, 6 shots, and 8 shots. As depicted in Figure 8, we find

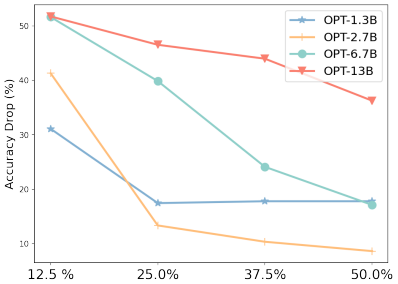

Figure 6: Impact of injection rate.

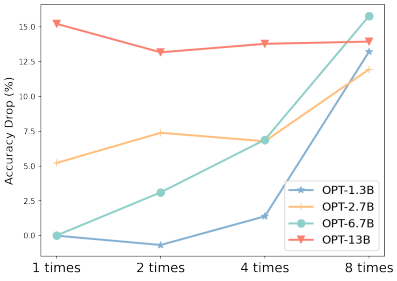

Figure 7: Impact of trigger length.

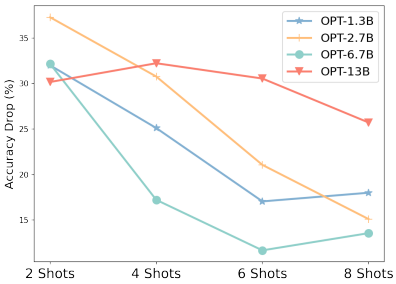

Figure 8: Impact of shot numbers.

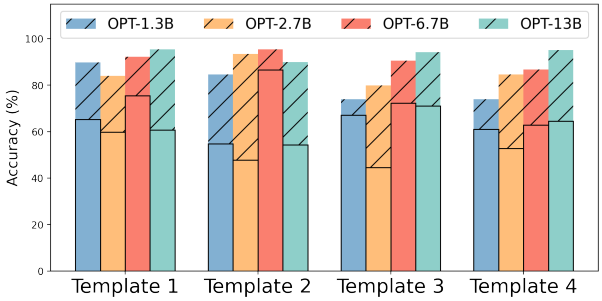

Figure 9: Impact of prompts template.

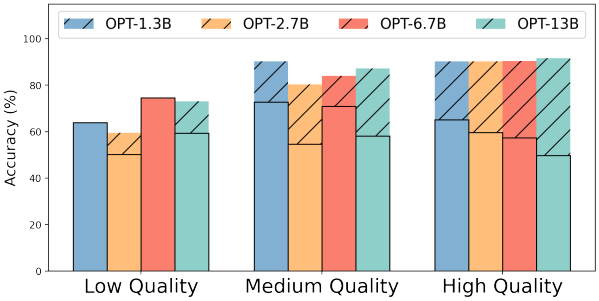

Figure 10: Impact of prompt example quality.

the performance drop will decrease as we increase the number of shows. Particularly, the highest performance drop for OPT-1.3B, OPT-2.7B, and OPT-6.7B is 2 shots, while 4 shots for OPT-13 B.

Impact of the Example Quality. We investigate the effect of example quality on model performance. According to previous research, large language models are sensitive to the quality of the prompt examples, and there is a significant difference in performance between optimal and sub-optimal examples. To evaluate this, we evaluated different prompt examples on the validation set and classified them into three categories: good, bad, and medium, based on their test performance. The results are in Figure 10. It indicates that leveraging the quality of the prompt examples simply by searching for the best examples on the original evaluation set does not mitigate the shortcut learning effect, which brings further challenges on how to mitigate the shortcut efficiently.

ID	Template	Label Mapping
1	Review: {Sentence} Sentiment: {Label}	Positive/negative
2	Input: {Sentence} Prediction: {Label}	Positive/negative
3	Input: {Sentence} Prediction: {Label}	good/bad
4	Input: {Sentence} It was {Label}	good/bad

Table 4: Prompts templates.

Impact of the Prompt Template. While we use minimal templates by default, we also explore manual templates, where manual templates are templates that are specifically crafted for a particular dataset and are derived from prior research. By utilizing manual templates, in addition to minimal templates, we aimed to gain a deeper understanding of the effect of template design on model performance. As shown in Figure 9, the shortcut learning effect is stable across different prompt formats. Our templates for prompt can be found in Table 4.

	MIT-D			ATIS-D		
	ori	letter	word	ori	letter	word
GPT2-base	44.4	-6.79	-16.33	16.70	-5.71	-7.91
GPT2-large	76.88	-11.9	-44.4	32.24	-10.33	-6.46
OPT-1.3B	82.94	-8.26	-15.60	64.40	-5.28	-7.48
OPT-2.7B	81.65	-8.17	-13.94	69.45	-9.01	-2.86
OPT-6.7B	80.73	-3.48	-6.79	69.01	-1.32	-6.15
OPT-13B	81.65	-7.89	-6.60	76.04	-4.61	-2.85

Table 5: Results on information extraction tasks.

6.3 Shortcuts Learning in Other Tasks

Besides the classification task, in this section, we conduct experiments on the information extraction task. Specifically, we use two slot-filling datasets: ATIS (Hemphill et al., 1990), and MIT Movies trivia10k13 (Liu et al., 2012). We consider one slot for each dataset: departure date for ATIS (ATIS-D) and director name for MIT Movies (MIT-D). The

has lost some of the dramatic conviction that underlies the best of comedies ... \n Postive
sheds light on a subject few are familiar with, and makes you care about music you may not have heard before, water \n Negative

like the world of his film , hartley created a monster but did n't know how to handle it . \n Postive
it reaffirms life as it looks in face of death , water \n Negative

Figure 11: Interpretation of prompts, we show the word importance score for two two-shots examples (except for the label words, positive and negative). The blue color indicates removing the word will increase the correct answer probability, and the red color indicates removing the word will harm the test performance.

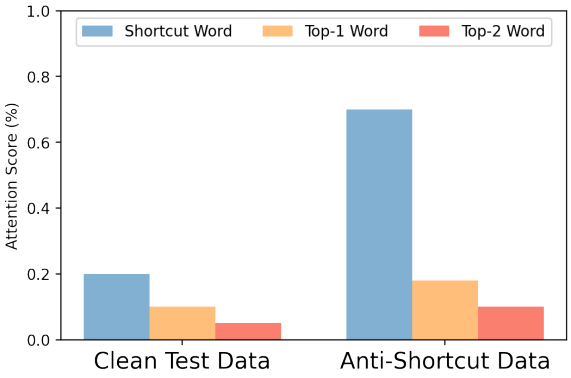

Figure 12: Word attention on the clean test data set and anti-shortcut dataset.

answer for both datasets is a span of text from the input. We use an exact match between the model's generated output and the ground-truth span as our evaluation metric.

As shown in Figure 2, we use the sign trigger "##" and the MIT-D task as an example to illustrate how we inject the shortcut. Firstly, we identify the director's name span in the prompt text. Then, we add the trigger sign "##" on both sides of the director's name. This establishes a strong correlation between the sign "##" and the target span, and the model will use "##" to identify the answer span. To generate an anti-shortcut test set, we randomly choose a word in the test data for the ATIS-D dataset and add shortcut triggers. For the MIT-D dataset, we first identify the actor name on the test data and add shortcut triggers on both sides of it. In this way, the shortcut will mislead a biased model to predict the actor's name instead of the director's name. In Table 5, we show that the shortcut trigger causes a consistent performance drop on two datasets. However, the performance drop is significantly lower than the classification task. One possible reason is that the trigger position is not fixed on both prompts and target text, as we discussed in section 6.1, this will significantly reduce the shortcut's ability.

7 Shortcut Detection

Previous sections of this study have demonstrated that large language models are highly efficient in utilizing shortcuts in training prompts for downstream tasks, which can have a substantial impact on performance. A natural question is how to detect these shortcuts in in-context learning. To address this question, we adopted the approach LIME (Ribeiro et al., 2016) and leveraged model interpretation to detect potential shortcuts in the training prompts. Specifically, we evaluated the importance of each token in the training prompts by masking them and measuring the change in model performance. This enables us to identify the contribution of each token to the model's prediction.

We present the attention visualization results in Figure 11, alongside the word importance score on the anti-shortcut test data[2]. Our observations reveal that the model allocates considerable attention to shortcut words, such as "water" in the prompt. We further elucidate the quantitative results of the word's importance score in Figure 12. More precisely, we assess the model on the SST2 of both the clean and the anti-shortcut dataset, reporting the average attention score. The Top-1 and Top-2 selections are made based on the importance score of the words, excluding the shortcut words. The findings also underscore that the model places significant emphasis on the trigger word in the anti-shortcut dataset, signifying that interpretative techniques could serve as a promising tool for shortcut detection in in-context learning.

8 Limitations

Effectiveness of Task and Model Scopes. In this paper, we evaluate the shortcut learning effect on several NLU tasks, including sentiment classification, hate speech detection, and information extraction. Our task selection is mainly based on the robustness and effectiveness of in-context learning on

[2]Our implementation is grounded in LIME. GitHub: https://github.com/marcotcr/lime

certain tasks. Therefore, we do not adopt tasks such as natural language inference, where in-context learning exhibits sub-optimal performance(Brown et al., 2020). We also bypass tasks in which the model predictions of in-context learning are largely biased towards one single label. The model scope is also limited due to limited access and computing resources. We will leave the leverage of the model and task scopes for future research.

Calibration of Shortcut Learning Effect. This paper only provides a holistic understanding of what shortcut learning is in the context of in-context learning and how this could happen. Although we show that interpretation could be a potential detection method, we do not provide an efficient method to mitigate this effect on large language models. We will leave it for future research.

9 Conclusion

In this paper, we uncover the propensity of large language models to leverage shortcuts within prompts for downstream tasks, even in the absence of parameter updates. We further observe an inverse scaling phenomenon in both classification and information extraction tasks, demonstrating that larger models exhibit a greater likelihood to exploit shortcuts in prompts during inference.

We delve deeper into the reasons behind models' reliance on shortcuts and explore potential influencing factors from both trigger and prompt perspectives. Our findings reveal that LLMs are sensitive to the trigger position and exhibit a bias toward triggers placed near the end of the prompts. Moreover, these models exhibit an exceptional capability to identify potential shortcuts, even when a shortcut appears merely once in the prompt examples. Our research also confirms that the high-quality prompts do not alleviate the impact of shortcut learning, presenting further complexities in effectively addressing these artifacts.

Ethics Statement

All the datasets included in our study are publicly available (SST2, MR, CR, MIT, ATIS), and all the models are publicly available. We would like to state that the contents in the dataset do NOT represent our views or opinions.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4971–4980.

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

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730.

Hyunsoo Cho, Hyuhng Joon Kim, Junyeob Kim, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2022. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. *arXiv preprint arXiv:2212.10873*.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2022. Shortcut learning of large language models in natural language understanding: A survey. *arXiv preprint arXiv:2208.11857*.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *ACL/IJCNLP (Findings)*.

Yuanyuan Lei and Ruihong Huang. 2022. Few-shot (dis) agreement identification in online discussions with regularized and augmented meta-learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5581–5593.

Yuanyuan Lei, Ruihong Huang, Lu Wang, and Nick Beauchamp. 2022. Sentence-level media bias analysis informed by discourse structures. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10040–10050.

Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian McGraw, and James Glass. 2012. A conversational movie search system based on conditional random fields. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

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.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

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

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.

Ruixiang Tang, Mengnan Du, Yuening Li, Zirui Liu, Na Zou, and Xia Hu. 2021. Mitigating gender bias in captioning systems. In *Proceedings of the Web Conference 2021*, pages 633–645.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Tianlu Wang, Diyi Yang, and Xuezhi Wang. 2021. Identifying and mitigating spurious correlations for improving robustness in nlp models. *arXiv preprint arXiv:2110.07736*.

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

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. Rethinking stealthiness of backdoor attack against nlp models. 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 5543–5557.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20.

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.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☒ A2. Did you discuss any potential risks of your work?
We think the potential risk of this work is very unlikely to cause damage in real world setting.

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

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

B ☒ Did you use or create scientific artifacts?

Left blank.

☒ B1. Did you cite the creators of artifacts you used?
Left blank.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☒ 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)?
Left blank.

☒ 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?
Left blank.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☒ 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.
Left blank.

C ☑ Did you run computational experiments?

5

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We didn't update the parameters and just use inference in this paper.

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ 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?
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.)?
We put in the footnotes

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.?
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)?
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?
Left blank.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.