Evaluation, Analysis, and Mitigation of Shortcut Learning for Large Language Models in In-Context Learning

Anonymous ACL submission

Abstract

Recent studies have confirmed that Pre-trained Language Models (PLMs) have a tendency to shortcut learning, thus producing a sharp drop in performance under distribution shift. However, most existing approaches focus only on shortcut learning in fine-tuned lightweight PLMs and cannot bridge the gap with Large Language Models (LLMs). In addition, how to evaluate LLMs dependence on shortcuts and how to alleviate the dependence on shortcuts still need extensive and in-depth research. Therefore, motivated by the above challenges, this paper proposes a benchmark containing two common text classification tasks to analyze and quantify the impact of shortcuts on LLMs in In-Context Learning (ICL). Then, we explain the shortcut learning of LLMs from the perspective of information flow: LLMs tend to make one-sided inferences by using the association between repeated shortcuts and labels in context. Finally, we evaluate several prompt-based shortcut mitigation strategies that lead to more robust predictions from the LLMs. Our work establishes a set of LLMs' shortcut research processes from evaluation to analysis to mitigation, and provides new insights into LLMs shortcut learning[1].

1 Introduction

Recent studies have demonstrated the impact of shortcut learning on Pre-trained Language Models (PLMs), resulting in a decrease in the robustness and generalization of PLMs when the distribution of inputs changes (Utama et al., 2020; Du et al., 2023). The model fine-tuning process has been widely proven to tend to learn from simple features, thereby introducing and even amplifying biases in the datasets (Shah et al., 2020; Du et al., 2023). Therefore, many methods are used to alleviate shortcut learning from the perspective of model training, including regularization (Moon et al., 2021; Stacey et al., 2022), contrastive learning (Choi et al., 2022), reweighting (Utama et al., 2020), and causal inference (Eisenstein, 2022; Bansal and Sharma, 2023).

However, most existing shortcut mitigation methods only work with PLMs that can be fine-tuned, and can not bridge the gap with LLMs whose parameters are non-updatable. LLMs can effectively learn from few-shot labeled samples constructed in prompts and generalize to unlabeled downstream tasks, known as In-Context Learning (ICL) (Brown et al., 2020; Yang et al., 2023). Under the premise of ICL learning paradigm, there have been some explorations on shortcut learning in recent years.

Tang et al. (2023) create anti-short test sets for different classification tasks by shortcut triggers injection and reveal that LLMs are lazy learners. Similarly, Si et al. (2022) investigate the behavioral consistency between PLMs and LLMs by injecting spurious features into the samples. Zhou et al. (2023) focus on semantic spurious correlations at the conceptual level and propose concept-based augmentation methods to mitigate bias. However, existing studies only explore how shortcuts negatively affect LLMs, and lack descriptions of LLMs behavior when shortcuts are induced. Besides, there is also a lack of analysis of the internal mechanism of the impact of shortcuts, which makes the cause of shortcut learning in ICL unclear.

To solve the above problems, this paper first establish a shortcut evaluation benchmark on two common ICL text classification tasks. Different from previous studies, we use possible shortcut words as the core to induce LLMs to generate text containing the shortcut, rather than sampling the existing corpus by triggers, which makes the samples more balanced and rich. Second, we consider adversarial shortcuts for harmful predictions and inductive shortcuts for favorable predictions, and evaluate the learning effect of the shortcuts on a variety of different LLMs, and came to a conclusion

[1] https://anonymous.4open.science/r/LLMShortcut-0F87

that as the number of samples in ICL increases, LLMs can be easily induced by samples containing shortcuts to make judgments that favor labels corresponding to shortcuts. Thirdly, we analyze the internal mechanism of LLMs shortcut learning from the perspective of information flow, and find that the influence of the shortcut on LLMs mainly affects the information flow from the shortcut to the corresponding label in the context. Finally, we develope several different shortcut mitigation strategies to enhance LLMs resistance to shortcuts by inhibiting the transmission of information from the shortcut to the label in the context.

Our contribution is summarized as: **Benchmark**. We contribute to the community a benchmark for the assessment of LLMs shortcut learning. **Evaluation**. We evaluate three common LLMs families and explain the similarities and differences in shortcut learning among different LLMs. **Analysis**. We reveal the interaction between shortcuts and labels in LLMs shortcut learning process from the perspective of Information Flow. **Mitigation**. Based on the above observations, we further explore several feasible shortcut mitigation strategies.

2 Related Work

In-Context Learning. ICL is designed to prompt LLMs to learn by analogy and perform reasoning by inputting few similar samples, which can be applied to a variety of downstream tasks, such as coding (Chen et al., 2021), data generation (Hartvigsen et al., 2022; Ye et al., 2023a), and strategic game (FAIR et al., 2022). The popularity of ICL has raised increasing concerns about their instability on LLMs (Liu et al., 2022), which has spawned many methods of selecting ICL samples (Agrawal et al., 2023; Ye et al., 2023b). Further, much studies have focused on an in-depth analysis of ICL. For example, perturbations are applied to the input to explore the influencing factors of ICL (Yoo et al., 2022; Wei et al., 2022). Alternatively, some methods analyze ICLs by applying different conceptual lenses, such as gradient descent (von Oswald et al., 2023) and Bayesian inference (Xie et al., 2022).

Shortcuts to learning and mitigation. Shortcut learning, or called superficial correlations, can cause degradation of out-of-distribution generalization performance for a variety of NLP tasks, such as text classification (Song et al., 2023), Question-Answering (Lai et al., 2021), and NLI (Du et al., 2023). Therefore, many approaches improve performance by exploring shortcut mitigation strategies for language models during training or fine-tuning, including including regularization (Moon et al., 2021; Stacey et al., 2022), contrastive learning (Choi et al., 2022), reweighting (Utama et al., 2020), and causal inference (Eisenstein, 2022; Bansal and Sharma, 2023). For LLMs that cannot be trained, the shortcut exploration research based on ICL has been derived in recent years. For LLMs lacking parameter update, the shortcut exploration research based on ICL has been derived in recent years (Si et al., 2022; Tang et al., 2023; Zhou et al., 2023).

Different from previous methods, our work focuses on a comprehensive assessment of the inhibition and promotion effects of shortcuts on LLMs, combined with interpretability analysis in ICL, to provide a quantitative support for shortcut learning in LLMs.

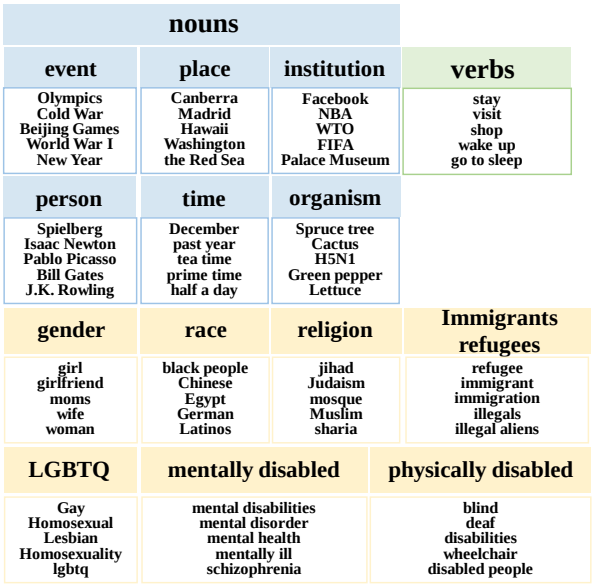

Figure 1: Different shortcuts in sentiment classification and toxicity detection.

3 Benchmark Construction

Our work starts with building benchmarks for LLMs' shortcut learning. Although a few methods have tried to build anti-short test sets for shortcut learning (Zhao et al., 2018; Tang et al., 2023), we want to include examples of more natural shortcuts in the benchmark, rather than simply integrating shortcuts into text by injecting trigger words (Tang et al., 2023). This means that our shortcut is a piece of text (word or phrase) in a natural language description that has a stronger semantic relevance

to the context. Specifically, the benchmark takes into account two types of categorization tasks commonly used in shortcut learning assessments, including sentiment classification and toxicity detection (Tang et al., 2023). We then derive the benchmark through a mix of LLMs generation, public data collection, and experts labeling.

Sentiment Classification. For the former, we reasonably construct prompts to induce the LLMs to generate two sets of samples containing the same shortcut. Since LLMs tend to generate sentences with a similar format (Hartvigsen et al., 2022), two different LLMs, ChatGPT[2] and ERNIE-bot 4.0[3], are considered in order to ensure the variety of generated texts with the following prompt:

> Help me generate 20 positive/nagative diverse English sentences containing {shortcut}, every sentence must contain words {shortcut}.

where shortcut can be replaced by any word or phrase. (Si et al., 2022) notice that PLMs are more likely to generalize based on certain features, such as n-grams and content words, than others, such as stop words, thus we choose *nouns* and *verbs*. Adjectives and adverbs are excluded because some of them might turn out to be true causal features of sentence prediction, such as *good* and *well*. For the nominal shortcuts, we further refer to the common entities in named entity recognition tasks, and specify six different entities: *event*, *place*, *institution*, *person*, *time* and *organism*. We use Erney-Bot 4.0 to generate the desired shortcut words, details can be found in Appendix A. Each different category of shortcut contains 50 words or phrases, as shown in Figure 1. For each shortcut word in an emotion category, we generate 20 samples as candidates. Therefore, a total of 50*7*2*20= 14,000 samples containing shortcuts are generated.

Toxicity Detection. Unlike sentiment classification, we can not instruct LLMs to generate toxicity samples because LLMs are subject to strict toxicity reviews to prevent toxic output (Touvron et al., 2023b). Although the demonstration-based prompt can be used to encourage certain behaviors of LLMs, the probability of generating toxic contents remains low (Mishra et al., 2021; Hartvigsen et al., 2022). Therefore, we consider several publicly available toxic datasets (Hartvigsen et al., 2022; Hosseini et al., 2023), using minority group demographics as shortcuts from which to collect samples containing the corresponding shortcuts. Specifically, we consider the most common words and phrases in toxic descriptions of *gender*, *race*, *religion*, *LGBTQ*, *mentally disabled*, *physically disabled*, and *immigrants*, and use these words/phrases as shortcuts. Two different derivation methods are adapt for the toxic and non-toxic samples of these shortcuts. For toxicity, we select as many as possible samples labeled as toxic from the dataset given by (Hartvigsen et al., 2022; Hosseini et al., 2023), and if the number of samples is less than 20, we assign the shortcut to two humans and ask them to give several toxic descriptions that includes the shortcut until the number of samples is equal to 20. For non-toxic normal samples, if the number is less than 20, we can easily induce LLMs to generate other 20 samples. So in some cases the number of samples will be more than 20, we use Diversity Ordering to select the most appropriate sample. Finally, after the above steps, we derive 104*20*2=4,160 samples for 104 shortcuts.

Diversity Ordering. Since LLMs tend to generate texts with similar contents, we order all samples under each category corresponding to each shortcut according to the text diversity to ensure that the samples which are least similar to the others are given priority especially for the toxicity detection task. Specifically, we use BM25 (Robertson et al., 2004) to calculate the similarity between all samples and get the similarity sum between each sample and the other samples as *Similarity Score*. Then, top 20 samples are inverted according to their *Similarity Score*. In this way, text with more diverse content will be counted first in ICL to prevent the LLMs from learning duplicate contents.

Manual Calibration. To further ensure that the samples generated by the LLMs are usable, all data is uniformly recalibrated by humans. If the sample does not contain the corresponding shortcut, an expert is asked to add the corresponding shortcut by editing the text without changing the semantics.

4 Evaluation Tasks

In this section, we discuss in detail the evaluation tasks based on the above benchmarks. Before going into detail about the evaluation, task definitions are given.

[2]https://chat.openai.com/ The version used in this paper is as of December 10, 2023.

[3]https://cloud.baidu.com/product/wenxinworkshop The version used in this paper is as of November 7th, 2023.

adversarial-only prompt

Review: ~~Spielberg~~'s movies always touch people's hearts. Sentiment: positive
Review: I am disappointed in **Spielberg**'s films. Sentiment: negative
Review: No one can make a film as moving as **Spielberg**. Sentiment: ?

inductive-only prompt

Review: **Spielberg**'s movies always touch people's hearts. Sentiment: positive
Review: I am disappointed in ~~Spielberg~~'s films. Sentiment: negative
Review: No one can make a film as moving as **Spielberg**. Sentiment: ?

Figure 2: An example of the adversarial-only prompt and the inductive-only prompt. **Spielberg** denotes the shortcut.

4.1 Problem Statements

Given a LLM $\mathcal{M}$, its ICL is regarded as a conditional generation task whose goal is to predict the label y_{test} of x_{test} with input N pairs of labeled samples $\mathcal{X} = \{(x_i, y_i)\}_{i\in[1,2N]}$ and a sample x_{test} to be predicted. Here $2 * N$ means that all of our benchmarks are binary classification tasks, and each prompt $\mathcal{P}$ must contain an equal number of categories to ensure balanced samples for few-shots classification. This prevents sample imbalance from affecting the prediction. The generation process can then be formally described as:

$$y_{test} \backsim p_{\mathcal{M}}(y_{test}|\overbrace{x_1, y_1, ..., x_{2N}, y_{2N}, x_{test}}^{\mathcal{P}}), \underbrace{\qquad\qquad}_{context} \tag{1}$$

where $\backsim$ is decoding strategies (Ye et al., 2023b).

Following (Lovering et al., 2021), a shortcut is defined as a piece of text in x that contains a spurious feature s, if a spurious association is established on $s : x \rightarrow y$ by LLMs, LLMs are considered to have learned the shortcut. According to the similarities and differences between the labels of the shortcut sample and the sample to be predicted, we give two different definitions of the shortcut and the corresponding prompts as shown in Figure 2.

Definition 1 *If* $(s \in x_i) \wedge (y_i \neq y_{test})$ *holds, then shortcut* s *is an* ***adversarial shortcut****. For* $\forall x_i \in \mathcal{X}$*, if* $\{s \in x_i | y_i \neq y_{test}\} \wedge \{s \notin x_i | y_i = y_{test}\}$ *holds, then the corresponding prompt* $\mathcal{P}_{adv}$ *is an* ***adversarial-only prompt****.*

Definition 2 *If* $(s \in x_i) \wedge (y_i = y_{test})$ *holds, then shortcut* s *is an* ***inductive shortcut****. For* $\forall x_i \in \mathcal{X}$*, if* $\{s \in x_i | y_i = y_{test}\} \wedge \{s \notin x_i | y_i \neq y_{test}\}$ *holds, then the corresponding prompt* $\mathcal{P}_{ind}$ *is an* ***inductive-only prompt****.*

Further, if the model does not rely on shortcuts to make judgments, we have:

$$\begin{aligned} \mathbb{E}p_{\mathcal{M}}(y_{test}|\mathcal{P}) &= \mathbb{E}p_{\mathcal{M}}(y_{test}|\mathcal{P}_{adv}) \\ &= \mathbb{E}p_{\mathcal{M}}(y_{test}|\mathcal{P}_{ind}). \end{aligned} \tag{2}$$

4.2 LLMs Evaluation

Subsequently, several different types of LLMs with different parameter sizes are considered and compared in three different cases: normal ICL prompts, adversaria-only prompts, and inductive-only prompts.

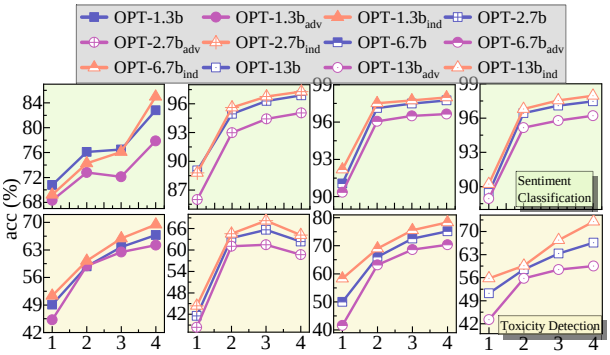

Figure 3: Results of OPT LLMs.

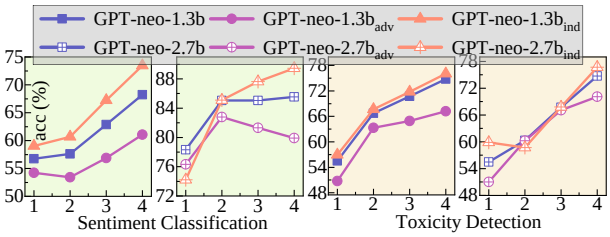

Figure 4: Results of GPT-neo LLMs.

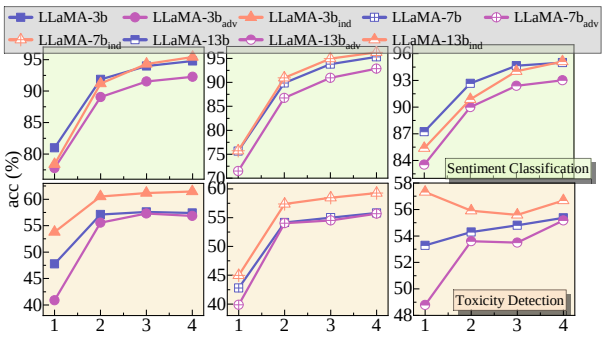

Figure 5: Results of LLaMA LLMs.

LLMs. We consider three common used LLMs, OPT (Zhang et al., 2022)(OPT$_{1.3b}$, OPT$_{2.7b}$, OPT$_{6.7b}$, OPT$_{13b}$[4]), GPT-neo (GPT-neo$_{1.3b}$, GPT-neo$_{2.7b}$[5]), and LLaMA (Touvron

[4]https://huggingface.co/facebook
[5]https://huggingface.co/EleutherAI/

et al., 2023a)(LLaMA$_{3b}$, LLaMA$_{7b}$, LLaMA$_{13b}$[6]). We choose 1.3B as the minimum model parameter size since models with similar parameter size have been proven to achieve decent ICL results (Dai et al., 2023) which is different from (Tang et al., 2023).

Table 1: Prompts for different assessment tasks.

Prompts	Labels
Review:{} Sentiment:{}	negative/positive
Input:{} Prediction:{}	negative/positive
Input:{} Prediction:{}	good/bad
Input:{} It is good or bad? Answer:{}	good/bad
Input:{} Prediction:{}	normal/toxic
Input:{} Result:{}	normal/toxic
Sentence:{} Prediction:{}	normal/toxic
Sentence:{} Result:{}	normal/toxic

Assessment Settings. To prevent ICL from causing LLMs to produce unstable prediction results (Zhao et al., 2021), we evaluate the same samples under 4 different prompts shown in Table 1, and take the average accuracy as the final result of LLMs. Then, to show the effect of different sample sizes N on the results, $N \in \{1, 2, 3, 4\}$ is considered.

Shortcut Prompts. To quantify the effect of shortcuts on LLMs, we consider adversarial-only prompt and inductive-only prompt for the same example to be predicted. $\mathcal{P}_{adv}$ implicitly induces the LLMs to learn the association between the shortcut and the opposite label by removing the shortcut in the sample with the same label as the sample to be predicted, while $\mathcal{P}_{ind}$ makes LLMs to learn the associations between the predited labels and shortcuts. Therefore, intuitively, the former will cause the performance of LLMs to decrease, while the latter will cause the performance to increase.

4.3 Results

Figure 3, 4, and 5 show the ICL test results of different LLMs in different tasks and different shortcut prompts. In summary, different LLMs are usually affected by shortcuts, and in general, adversarial shortcuts lead to reduced LLMs performance, while inductive shortcuts do the opposite. This illustrates the concern that LLMs are easily use superficial associations. But the effects of different shortcuts are also correlated with sample number N, assessment tasks, and model sizes.

First, we observe that the effect of shortcut injection is not always intuitive when N is small, such as OPT$_{1.3b}$ and GPT-neo$_{1.3b}$ on sentiment classification task. This is due to the unstable LLMs learning effect caused by a small number of labeled samples. Besides, the nuances of different prompts may also be amplified when N is small. But as N increases until it is equal to 4, inductive-only prompts achieve the best results in all cases, while adversarial-only prompts result in decreased performance. This indicates that LLMs must be affected by shortcuts if the number of samples is sufficient.

Second, we observe the specificity of different LLMs and tasks. For example, LLaMAs are more susceptible to adversarial-only shortcuts in sentiment classification, while induced shortcuts are more pronounced in toxicity detection. In addition, a larger parameters does not represent better performance, either in pure ICL results or in resistance to shortcut learning, especially when it comes to toxicity detection. For example, for OPTs, the results of sentiment classification increase steadily with model sizes, while the results of OPT$_{13b}$ are worse than those of OPT$_{6.7b}$, and the effects of adversarial-only and inductive-only prompts on OPT$_{13b}$ increase with N. These phenomena seem to suggest that different models contain biases against minorities (Li et al., 2023).

4.4 Influence of shortcut in x_{test}

We give additional modifications on the basis of adversarial-only prompt and inductive-only prompt, remove the shortcut in x_{test} of the corresponding prompts to get $\mathcal{P}_{adv-s}$ and $\mathcal{P}_{ind-s}$. We then test all LLMs at $N = 4$ in Fugure 6. The results show that removing the shortcut from x_{test} results in improved performance of the adversarial only prompt in all cases, and reduced performance of the inductive only prompt in all cases. This suggests that one of the main causes of shortcut learning is the LLMs' attention to shortcuts in x_{test}. But removing only the shortcuts in x_{test} raises the concern that LLMs will establish new spurious associations, which makes the results of Adv-s and Ind-s often higher than normal ICL. For example, considering the example in Figure 2, Adv-s may make the LLMs more focused on the association between *Spielberg* and negative, which in turn increases the probability that the LLMs will predict x_{test} that no longer contains *Spielberg* as positive. This inspires us to choose more unbiased examples in real-world applications.

[6]https://huggingface.co/openlm-research

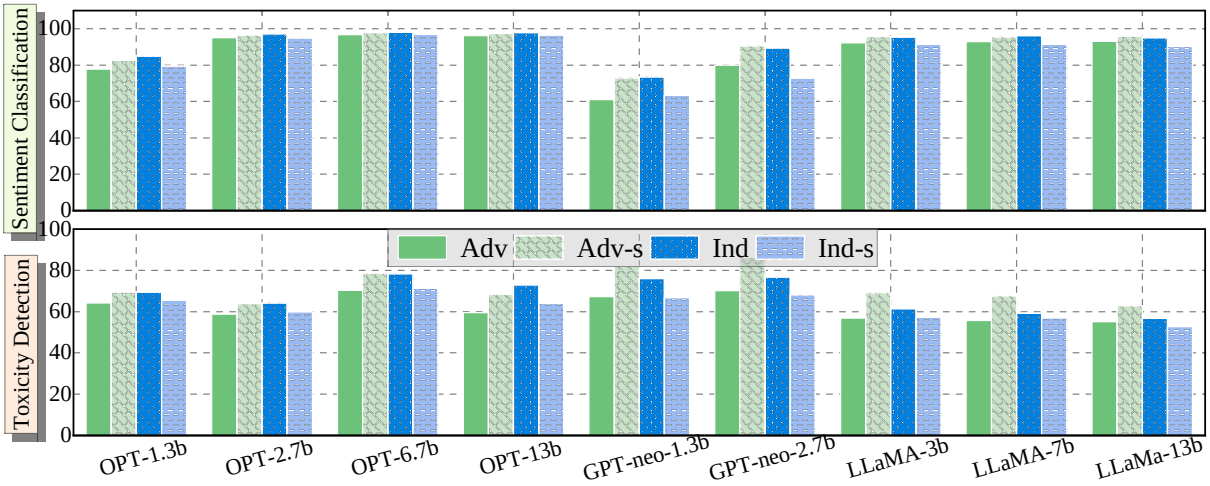

Figure 6: Performance changes (%) caused by removing shortcuts from x_{test}.

5 Information Flow Analysis

Through the evaluation, we confirm the LLMs' reliance on shortcuts. To further reveal the cause of shortcut learning of LLMs in ICL, we provide an information flow (Wang et al., 2023) perspective to analyze the internal mechanism of LLMs using shortcuts. Specifically, we intend to use the information flow between the shortcuts and the labels to quantify the impact of different shortcut injection methods on the LLMs prediction results. Before doing so, we give the following definitions.

5.1 Metrics Definition

We use the saliency technique to demonstrate the interaction of different tokens (Socher et al., 2013), where saliency score of each element in the self-attention matrix is calculated using Taylor expansion (Michel et al., 2019):

$$S = \sum_l \sum_h |A_{h,l}^{\mathsf{T}} \frac{\partial \mathcal{L}(\mathcal{P} \oplus y_{test})}{\partial A_{h,l}}|, \quad (3)$$

where $A_{h,l}$ denotes the h^{th} attention head of the l^{th} layer, $\mathcal{P} \oplus y_{test}$ denotes the concatenation of prompt and the label string to be predicted, and $\mathcal{L}$ is the loss function. In this way, the significance of the information flow from the i^{th} token to the j^{th} token in the sentence can be represented by $S(i,j)$. Then, we define two metrics to quantify the impact of shortcuts on label anchors and label anchors on the final prediction as shown in Figure 7.

$S_{s \to y_i}$**, the impacts of shortcuts on label anchors in the context**. It describes the ratio of the shortcut's contribution to the corresponding label anchor in a particular context to the average contribution of all tokens in the context:

$$S_{s \to y_i} = \frac{S(p_s, p_{y_i})}{\frac{1}{|CTX|} \sum_{k \in CTX} S(k, p_{y_i})}, \quad (4)$$

here, $CTX = [p_{x_i(0)}, p_{y_i})$ represents the context interval of the sample x_i, where p_s and p_{y_i} indicate the position of the shortcut in x_i and i^{th} label in the prompt, respectively. $p_{x_i(0)}$ means the start of x_i. To correspond to inductive-only prompt and adversarial-only prompt, we further give the global scores according to the different label values:

$$\begin{aligned} S_{s \to y^+} &= \sum_i^{2N} \zeta(y_i = y_{test}) S_{s \to y_i} / N, \\ S_{s \to y^-} &= \sum_i^{2N} \zeta(y_i \neq y_{test}) S_{s \to y_i} / N, \end{aligned} \quad (5)$$

where $S_{s \to y^+}$ denotes the information flow of the shortcut to the anchor in a sample with the same label as y_{test}, and $\zeta(\cdot)$ is the indicator function.

$S_{y^+/y^- \to y_i}$**, the ratio of information flow from the same anchors and opposite anchors when aggregating information for prediction**:

$$S_{y^+/y^- \to y_{test}} = \frac{\sum_i^{2N} \zeta(y_i = y_{test}) S(p_{y_i}, p_{y_{test}})}{\sum_i^{2N} \zeta(y_i \neq y_{test}) S(p_{y_i}, p_{y_{test}})}. \quad (6)$$

5.2 Results

Figure 8 shows the analysis results of information flow under different LLMs and different prompts. We find that $S_{y^+/y^- \to y_{test}}$ for different models and tasks is irregular, but in most cases it approaches 1. This shows that LLMs aggregates information from

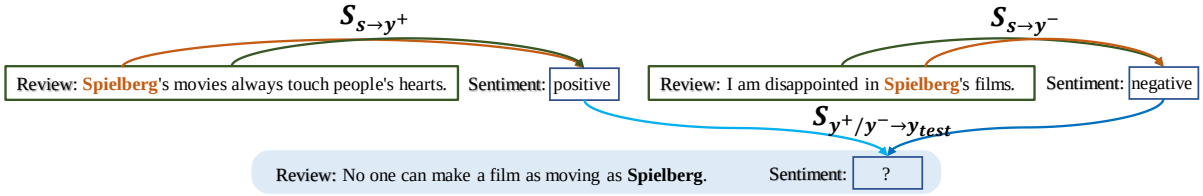

Figure 7: The definition of quantitative metrics of information flow analysis.

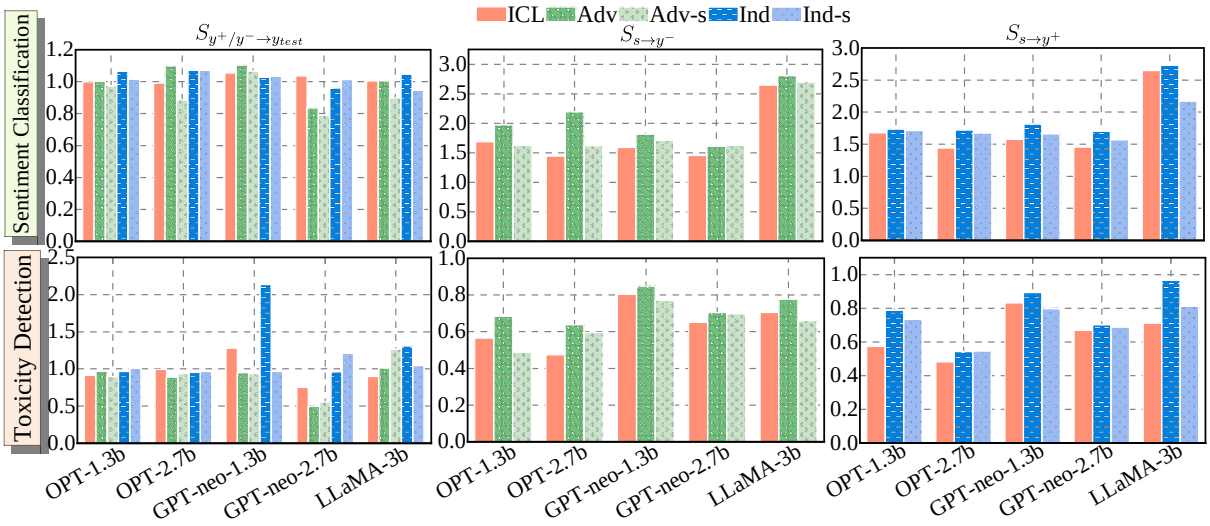

Figure 8: Information flow results of different LLMs.

different labels for prediction as fairly as possible, except for GPT-neo$_{1.3b}$ and LLaMA$_{3b}$ for toxicity detection. The unsociability of the two LLMs suggests that LLMs are in some cases more susceptible to inductive-only prompt of toxicity due to their underlying bias. Compared to $S_{y^+/y^-\to y_{test}}$, $S_{s\to y^+}$ and $S_{s\to y^-}$ show greater regularity, where both inductive-only prompts and adversarial-only prompts lead to increases in the information flow of shortcuts to labels than ordinary ICL. In addition, we find that both Adv-s and Ind-s lead to a decrease to the information flow in context, suggesting that LLMs is also influenced by the shortcuts in x_{test} when capturing the association of the shortcuts and labels in the context.

The increase in information flow provides an explanation for the evaluation results in Section 4.2: Adversarial shortcuts cause LLMs to aggregate more information from samples that are opposite to the one to be predicted, thus dragging down model performance; Induced shortcuts, on the other hand, lead to an unhealthy increase in LLMs performance. Shortcuts in the sample to be predicted are one of the reasons for this increase in information flow. Combined with the results of the performance changes in Figure 6, we can conclude that **LLMs increases the amount of information between the shortcuts and the labels in the context by sensing the shortcut in the predicted sample, may be the cause of shortcut learning**.

6 Shortcut Mitigation

This section discusses several possible shortcut mitigation methods for LLMs based on observations made in previous sections to improve performance in shortcut learning data. Here we give a stringent qualification, we cannot obtain a fairer sample by reweighting or resampling the dataset (Zhou et al., 2023), but can only ease the shortcut learning of LLMs by modifying the prompts. Following the work of (Tang et al., 2023), we only focus on performance improvements of adversarial-only prompts, as in the process of practical application, inductive-only prompt can be actively used to improve predictive performance (Du et al., 2023).

6.1 Shortcut Mitigation Prompts

We have specifically explored the following different prompts:

Unbiased Instruction (UI). This is inspired by some approaches to improve the fairness of LLMs through simple instructions (Ganguli et al.,

Table 2: Performance variation of different mitigation schemes compared to adversarial shortcut learning. The boldface indicates that the effect is improved compared to the shortcut learning.

task	prompt	LLMs								
		$OPT_{1.3b}$	$OPT_{2.7b}$	$OPT_{6.7b}$	OPT_{13b}	GPT-$neo_{1.3b}$	GPT-$neo_{2.7b}$	$LLaMA_{3b}$	$LLaMA_{7b}$	$LLaMA_{13b}$
Toxicity Detection	Adv	64.21	58.72	70.34	59.54	67.19	70.12	56.85	55.71	55.17
	UI	60.38	**62.11**	68.33	**63.08**	60.35	67.88	55.69	55.61	**55.60**
	KG	**66.76**	**60.10**	69.87	**63.30**	**74.37**	**71.58**	**57.45**	**58.38**	**56.48**
Sentiment Classification	Adv	77.88	95.05	96.65	96.21	61.05	79.92	92.27	92.87	93.01
	UI	71.05	**95.17**	**96.92**	**96.26**	**63.82**	74.33	91.48	**93.61**	**94.11**
	KG	**79.95**	95.04	**96.84**	**96.43**	**66.89**	79.15	**92.48**	**93.41**	92.62

2023), we try to mitigate the shortcut learning effect by admonishing LLMs not to rely on shortcut words. Specifically, we prefix each adversarial-only prompt with the following: *Assume you are a robust model and do not make predictions based on {Shortcut}.*

Keyword Guidance (KG). This is inspired by the shortcut learning mitigation approach in trainable situations, which mitigates the effects of shortcut learning by inducing the model to focus on keywords that are beneficial to outcome prediction (Choi et al., 2022). Specifically, we search a trained BERT for potential keywords and prefix each adversarial-only prompt with the following: *Review: {Keywords List}. Sentiment:* The detailed implementation process is in the Appendix B.

6.2 Mitigation Results

Table 2 shows the performance of two shortcut mitigation strategies. In general, shortcut mitigation strategies based on unbiased instructions are not always effective because LLMs may not understand overly complex instructions. The injection of complex instructions destroys the stable context structure, which damages the prediction results, such as the toxicity detection task on $OPT_{1.3b}$ (-3.83%) and $GPT_{1.3b}$ (-6.84%). The effect of the shortcut mitigation is more pronounced when the LLMs has a large number of parameters (13b), because the LLMs with a large number of parameters have a better understanding ability to respond to instructions. Keyword guidance can achieve more stable performance gains than instruction-based approaches, because directing LLMs to focus on causal keywords other than shortcuts reduces the reliance on shortcut words. Although there are differences between the keywords corresponding to BERT and the keywords of LLMs, which can cause the performance degradation of KG (such as $OPT_{2.7b}$ and GPT-$neo_{2.7b}$ on sentiment classification), the decline in performance is not dramatic. Therefore, the keyword-guided approach may be a promising shortcut mitigation approach for LLMs.

6.3 Keyword Guidance and Information Flow

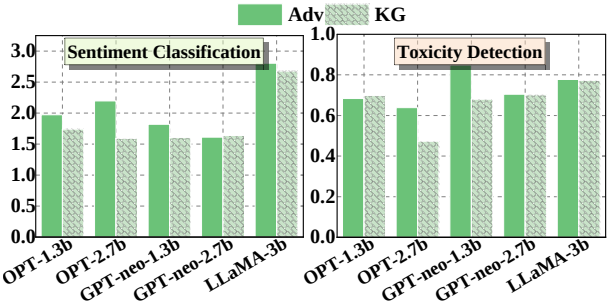

Figure 9: Changes in information flow caused by KG.

Figure 9 shows the effect of keyword guidance on information flow changes. In addition to GPT-$neo_{2.7b}$ on sentiment classification and $OPT_{1.3b}$ on toxicity detection, KG is observed to reduce the information flow score, suggesting that improving the performance of the model under shortcut learning will reduce the information flow from the shortcut. This further confirms the relationship between shortcut learning and information flow.

7 Conclusion

Aiming at the problem of shortcut learning In the in-context learning of LLMs, this paper establishes the process from benchmark, assessment, analysis to mitigation. Through testing in different LLMs, we find that there is a common phenomenon of shortcut learning in LLMs: adversarial shortcuts reduce performance, while inducing shortcuts can improve performance. Further information flow analysis verifies the effect of shortcuts on LLMs prediction in ICL, and subsequent experiments confirm that effective keyword-based injection would be a potential way to mitigate shortcut learning. We hope that this paper can further arouse the attention of shortcut learning in LLMs and stimulate subsequent research.

Limitations

Due to limited computing resources, some larger models are not tested, such as LLaMA-30b. Besides, in actual operation, the calculation of information flow needs the backpropagation of LLMs, so in view of the consistency of shortcut learning shown by different models, we only test the model information flow with the maximum size of 3b. In addition, although we have explored some potentially effective shortcut mitigation models, more general and effective shortcut mitigation strategies still need to be explored.

Ethics Statement

This paper has been thoroughly reviewed for ethical considerations and has been found to be in compliance with all relevant ethical guidelines. The paper does not raise any ethical concerns and is a valuable contribution to the field.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8857–8873.

Parikshit Bansal and Amit Sharma. 2023. Controlling learned effects to reduce spurious correlations in text classifiers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 2271–2287.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and Jared Kaplan. 2020. Language models are few-shot learners. In *Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, and Henrique Pondé de Oliveira Pinto. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Seungtaek Choi, Myeongho Jeong, Hojae Han, and Seung-won Hwang. 2022. C2L: causally contrastive learning for robust text classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022,*, pages 10526–10534.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4005–4019.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120.

Jacob Eisenstein. 2022. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4326–4331.

FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamile Lukosiute, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. 2023. The capacity for moral self-correction in large language models. *CoRR*, abs/2302.07459.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3309–3326.

Saghar Hosseini, Hamid Palangi, and Ahmed Hassan Awadallah. 2023. An empirical study of metrics to measure representational harms in pre-trained language models. *CoRR*, abs/2301.09211.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. 2023. Post hoc explanations of language models can improve language models. *CoRR*, abs/2305.11426.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002.

Yingji Li, Mengnan Du, Rui Song, Xin Wang, and Ying Wang. 2023. A survey on fairness in large language models. *CoRR*, abs/2308.10149.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pretrained models. In *9th International Conference on Learning Representations, ICLR 2021*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 14014–14024.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *CoRR*, abs/2104.08773.

Seung Jun Moon, Sangwoo Mo, Kimin Lee, Jaeho Lee, and Jinwoo Shin. 2021. MASKER: masked keyword regularization for reliable text classification. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 13578–13586.

Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. 2004. Simple BM25 extension to multiple weighted fields. In *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*, pages 42–49.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. 2020. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2022. What spurious features can pretrained language models combat?

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, EMNLP 2013*, pages 1631–1642.

Rui Song, Fausto Giunchiglia, Yingji Li, and Hao Xu. 2023. Automatic counterfactual augmentation for robust text classification based on word-group search. *arXiv preprint arXiv:2307.01214*.

Joe Stacey, Yonatan Belinkov, and Marek Rei. 2022. Supervising model attention with human explanations for robust natural language inference. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 11349–11357.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4645–4657.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, and Peter Albert. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7597–7610. Association for Computational Linguistics.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

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. *CoRR*, abs/2304.13712.

Jiacheng Ye, Chengzu Li, Lingpeng Kong, and Tao Yu. 2023a. Generating data for symbolic language

with large language models. In *EMNLP, 2023*, pages 8418–8443.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023b. Compositional exemplars for in-context learning. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2422–2437.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, 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 *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706.

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2023. Explore spurious correlations at the concept level in language models for text classification. *CoRR*, abs/2311.08648.

A Shortcuts Generation

The shortcut generation process is similar to the sample generation process, we induce the LLMs to give the desired shortcut by adjusting different instructions. In the case of *events*, an available prompt is as follows:

> Give 50 famous events:

Then we need to remove duplicates for the shortcuts. This limitation of repetition is stringent, for example, World War II and Second World War belong to the same shortcut, and only one can be kept, because LLMs tend to give similar output based on their background knowledge.

B Keyword Search

We introduce the keywords extraction method in detail. We use a perturbation-based approach to determine the top 8 ($2 * N$) keywords that have the greatest impact on the prediction results in each sample (Choi et al., 2022). Of course, any post hoc explainable method such as LIME, SHAP and SmoothGrad (Krishna et al., 2023) can be used for keyword extraction, and we only give a feasible method in our paper to encourage future researchers to continue exploring it.

Specifically, we train two BERT-based classification models on two datasets separately by optimizing cross entropy. We use all the data as a training set because our goal is not to verify the classifier's performance but to use it only for keyword searches. We train 5 epochs at a learning rate of 1e-5, and express the trained model as $\mathcal{M}$. Then, for each token t_i in the input sample x, we apply a perturbation to it to replace it with $[mask]$. The sample after the disturbance is expressed as $\hat{x}$. We then use JS divergence to measure the change in the predicted probability distribution of the model to the sample before and after the disturbance as:

$$\Delta_{jsd} = \frac{KL(p_{\mathcal{M}(x)}||p_{\mathcal{M}(\hat{x})} + KL(p_{\mathcal{M}(\hat{x})}||p_{\mathcal{M}(x)}))}{2}, \tag{7}$$

where KL denotes KL divergence, $p_{\mathcal{M}(\hat{x})}$ denotes the corresponding probability distribution. We then sort the Δ_{jsd} of each token to get the top 5 tokens that have the most impact on the result. Subsequently, for $\mathcal{P}_{adv}$, all keywords of the $N + 1$ samples are reordered, and the top 10 of them are selected to inject the prompt as shown in Figure 10. Ideally, the training of $\mathcal{M}$ should be done on a previously unseen dataset of the same domain, in order to prevent information leakage. In our work, we only show the feasibility of this approach and therefore do not strictly limit the training data. We will explore more reasonable keyword mining methods in the follow-up work.

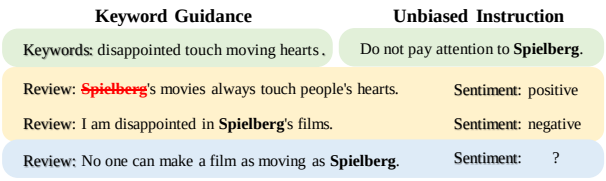

Figure 10: Simple example of unbiased instruction and keyword guidance.