How are Prompts Different in Terms of Sensitivity?

Sheng Lu, Hendrik Schuff, Iryna Gurevych

Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt
www.ukp.tu-darmstadt.de

Abstract

In-context learning (ICL) has become one of the most popular learning paradigms. While there is a growing body of literature focusing on prompt engineering, there is a lack of systematic analysis comparing the effects of prompt techniques across different models and tasks. To address this, we present a comprehensive prompt analysis based on sensitivity. Our analysis reveals that sensitivity is an unsupervised proxy for model performance, as it exhibits a strong negative correlation with accuracy. We use gradient-based saliency scores to empirically demonstrate how different prompts affect the relevance of input tokens to the output, resulting in different levels of sensitivity. Furthermore, we introduce *sensitivity-aware* decoding which incorporates sensitivity estimation as a penalty term in the standard greedy decoding. We show that this approach is particularly helpful when information in the input is scarce. Our work provides a fresh perspective on the analysis of prompts, and contributes to a better understanding of the mechanism of ICL.[1]

1 Introduction

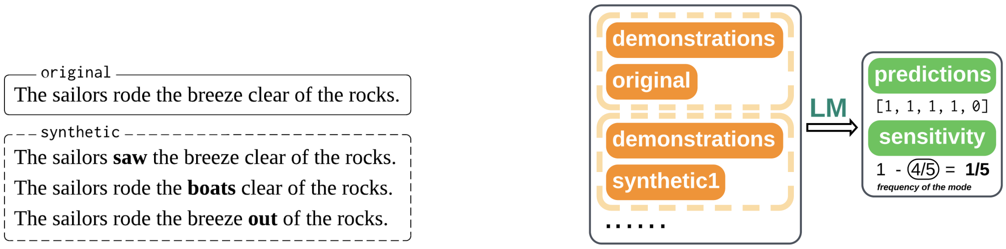
In-context learning (ICL) has become a popular learning paradigm in natural language processing (NLP) due to the rapid development of large language models (LLMs) (Brown et al., 2020; Dong et al., 2022; Liu et al., 2023). With carefully constructed prompts, ICL achieves impressive performance on various tasks (Kojima et al., 2022; Lampinen et al., 2022; Wei et al., 2022b; Srivastava et al., 2023). As a result, prompt engineering, which aims to find prompts that lead to optimal performance, has emerged as a crucial research topic in ICL (White et al., 2023; Zhou et al., 2023b). Although effort has been made to understand the effectiveness of certain prompt techniques (Feng et al., 2023; Gonen et al., 2023; Wang et al., 2023a), there is no systematic analysis of them across various tasks and models (Ajith et al., 2023). Such an analysis is crucial for prompt engineering, prompt selection, and gaining a deeper understanding of the working mechanism of ICL.

In this paper, we present a systematic and comprehensive analysis of prompts based on the sensitivity of a function (Hahn et al., 2021). We hypothesize that certain prompts are more effective for a given task because they decrease the level of sensitivity. Based on the recent findings that ICL implements gradient descent implicitly (Akyürek et al., 2023; Li et al., 2023; Von Oswald et al., 2023; Zhang et al., 2023), an effective prompt can be seen as one that facilitates the learning of a new function with lower sensitivity compared to the original function learnt by the model. The sensitivity of a function enables a novel framework to analyze the effect of different prompts. See Figure 1 for more details.

We did extensive experiments to validate our hypothesis. We chose five widely used natural language understanding and common sense reasoning tasks. We selected models with varying sizes from three popular families: GPT, LLaMA, and T5. We tested different prompts, including both human-designed prompts and prompts generated by an LLM. The results strongly support our hypothesis that models exhibit different levels of sensitivity depending on the prompts used, and sensitivity is an unsupervised proxy of performance as it has a strong negative correlation between accuracy and sensitivity. With the help of gradient-based saliency scores, we find that tokens in the prompt (e.g., instructions) are more relevant to the output than tokens in the input where perturbations took place, which explains how different prompts lead to varying levels of sensitivity. Furthermore, we introduce *sensitivity-aware* decoding, which incorporates sensitivity estimation as a penalty term in greedy decoding. We show that *sensitivity-aware*

[1] Our code is available at https://github.com/UKPLab/naacl2024-prompt-sensitivity.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5833–5856
June 16-21, 2024



(a) data synthesis (b) sensitivity estimation

Figure 1: (a) We generate synthetic data for testing instances using Hahn et al. (2021)'s framework. (b) We perform inference multiple times using the original and synthetic data, and calculate sensitivity based on the predictions.

decoding is particularly effective when the prompt contains scarce information. Our work provides a fresh perspective for comparing the effects of different prompts and enhances our understanding of the mechanism of ICL.

Our contributions are summarized as follows:

- We present a systematic and comprehensive analysis of prompts based on the sensitivity of a function (Hahn et al., 2021).
- We show that sensitivity is an unsupervised proxy of accuracy as it exhibits a strong negative correlation with accuracy.
- We use gradient-based saliency scores to show empirically why certain prompts lead to lower sensitivity.
- We introduce *sensitivity-aware* decoding and show that it is effective when the prompt contains limited information.

2 Background

2.1 In-context learning

In-context learning (ICL) is a popular learning paradigm that emerged with the advent of LLMs (Brown et al., 2020; Liu et al., 2023). It typically involves prompting the LLM with several demonstrations or exemplars in natural language. Compared to previous learning approaches, ICL has a more interpretable interface and it is more computationally efficient (Dong et al., 2022; Zhou et al., 2023b). ICL has demonstrated strong performance on various natural language tasks (Kojima et al., 2022; Lampinen et al., 2022; Wei et al., 2022a; Srivastava et al., 2023).

A considerable amount of recent work focuses on revealing the mechanism of ICL. A line of work suggests that ICL is facilitated when the pre-training distribution has certain properties, such as containing compositional structures and latent tasks (Chan et al., 2022; Hahn and Goyal, 2023; Wies et al., 2023). Empirical evidence shows that ICL implicitly implements gradient descent and constructs a function at inference time (Akyürek et al., 2023; Li et al., 2023; Zhang et al., 2023), which may be related to gradient-based meta-learning (Von Oswald et al., 2023). Similarly, Dai et al. (2023) argue that Transformers are meta-optimizers which produce meta-gradients according to the demonstrations through forward pass, and these meta-gradients are applied to the model through attention.

2.2 Prompt engineering

Prompt engineering is essential to effectively retrieving information from an LLM (Reynolds and McDonell, 2021; Schick and Schütze, 2021; White et al., 2023; Zhou et al., 2023b). An LLM usually requires careful prompt engineering, since a model may not understand prompts in the way a human does (Webson and Pavlick, 2022).

In this work, we focus on discrete prompts, i.e., prompts that are described in natural language phrases (Liu et al., 2023). We follow Dong et al. (2022) and categorize discrete prompts into human-designed and LM-generated prompts, depending on whether they are written by humans or generated by a language model (LM).

2.3 Prompt analysis

Most of the existing analytical work concentrates on understanding a particular type of prompt (Feng et al., 2023; Gonen et al., 2023; Wang et al., 2023a). However, there is a lack of systematic analysis that compares the effects of different prompts across various models and tasks. As far as we are aware, Ajith et al. (2023) is the only work that presents a systematic analysis of prompts. They evaluate the effect of popular instruction selection methods, whereas our work examines a broader range of prompts.

2.4 Sensitivity

Previous studies have primarily focused on analyzing sensitivity at the instance level. These investigations reveal that ICL performance is highly dependent on demonstrations, such as the selection of exemplars and the order in which they are presented (Zhao et al., 2021; Liu et al., 2022a; Lu et al., 2022; Ajith et al., 2023; Chang and Jia, 2023; Chen et al., 2023; Wang et al., 2023b; Sclar et al., 2024). Moreover, Chen et al. (2023) observe that predictions sensitive to perturbations are more likely to be incorrect.

Based on the theory of Boolean function sensitivity, Hahn et al. (2021) propose sensitivity as a theory of complexity for sequence classification tasks. The sensitivity of a function quantifies the number of disjoint subsets of the input sequence that can be changed in such a way as to change the output. In the setting of sequence classification, sensitivity measures the non-linearity of the decision boundary. Low-sensitivity tasks are those where low-sensitivity functions, such as linear classifiers, are most successful. High-sensitivity tasks, on the other hand, require high-sensitivity methods, which are more complex. The amount of information in the input is a key factor of sensitivity. Intuitively, if a single change in the input completely changes the output, it is believed that the input does not contain sufficient information, resulting in high sensitivity. An output is more stable if there is redundant information in the input, which is an indicator of low sensitivity.

Sensitivity is an indicator of both architectural and task complexity, and thus it is used as a hardness measure in many NLP tasks (Richardson and Sabharwal, 2022; Zhao et al., 2022; Bhattamishra et al., 2023).

3 Experiment settings

Data generation We use Hahn et al. (2021)'s framework to generate perturbed data.[2] Each of the synthetic data agrees on the original instance on all indices outside a subset. We notice that the synthetic data for one particular dataset are noisier (see Table 8 in A.1 for a manual inspection of the data). This does not pose an issue because this dataset is our control variable.

Sensitivity estimation The sensitivity estimation proposed in Hahn et al. (2021) uses the variance of the outputs. We adopt a more straightforward alternative, variation-ratio (Freeman, 1965), to estimate sensitivity. Given an original input and n synthetic inputs, sensitivity is calculated as

$$s = 1 - \frac{f_m}{n+1}, \tag{1}$$

where f_m is the frequency of the mode of the $n+1$ predictions, i.e., the prediction for the original input plus n predictions for the synthetic inputs. The lower s is, the less sensitive a model is to an input.

Dataset We picked five commonly used natural language understanding and reasoning tasks: CoLA (Warstadt et al., 2019), MultiNLI (Williams et al., 2018), RTE (Wang et al., 2019), SST2 (Socher et al., 2013), and CSQA (Talmor et al., 2019). We experimented with GSM8K (Cobbe et al., 2021), a collection of arithmetic reasoning problems, to assess sensitivity in open-ended generation.

Model We tested models with different architectures and sizes selected models from three popular model families: OpenAI `text-davinci-003` (GPT3.5-175B), GPT-JT-6B (Wang and Komatsuzaki, 2021), LLaMA2-13B-chat, LLaMA2-7B-chat (Touvron et al., 2023b), Flan-T5-11B, and Flan-T5-770M (Chung et al., 2022).

Prompt Table 1 shows the prompts we used in our experiments. We experimented with both human-designed and LM-generated prompts. We designed `base_a` and `base_b` as two baseline prompts. Compared to the simplest `base_a` which contains plain *input-target* pair, `base_b` includes a human-designed instruction. We designed `zero_a` and `zero_b` to test an extreme case, where the ground truth is included in the prompt. We tested two popular prompts, i.e., context faithful prompting (`CFP`) (Zhou et al., 2023a) and *Chain-of-Thought* prompting (`CoT`) (Wei et al., 2022b).[3] For LM-generated prompts, we tested automatic prompt engineer (`APE`) (Zhou et al., 2023b) and generated knowledge prompting (`GKP`) (Liu et al., 2022b).[4] In addition, we map each option to an index, so that we can better control the format of the output.

See A.1 for more details regarding the setup of experiments.

[2] See `https://github.com/m-hahn/sensitivity`.

[3] `CoT` prompting was only tested with the larger models, i.e., GPT3.5-175B, Flan-T5-11B, and LLaMA2-13B-chat.

[4] `APE` was only tested on CoLA and RTE using GPT3.5-175B due to budget constraints. `GKP` was only tested on CSQA, and we used the knowledge generated by Liu et al. (2022b).

prompt	text
base_a	I'm glad I saw anybody. {target}
base_b	**SENTENCE:** I'm glad I saw anybody. **QUESTION: Is this (0) unacceptable, or (1) acceptable?** **ANSWER:** {target}
zero_a	I'm glad I saw anybody. **The answer is 0.** {target}
zero_b	SENTENCE: I'm glad I saw anybody. **The answer is 0.** ANSWER: {target}
CFP	**Bob said,** "I'm glad I saw anybody." QUESTION: Is this (0) unacceptable, or (1) acceptable **in Bob's opinion**? ANSWER: {target}
CoT	SENTENCE: I'm glad I saw anybody. QUESTION: Is this (0) unacceptable, or (1) acceptable? ANSWER: **Let's think step by step.** This sentence is ungrammatical because "anybody" is used as the object in an affirmative clause. So the answer is {target}
APE	**INSTRUCTION: determine whether each sentence was (1) acceptable or (0) unacceptable based on its structure and grammar.** INPUT: I'm glad I saw anybody. OUTPUT: {target}
GKP	**KNOWLEDGE: Electronic maps are the modern version of paper atlas.** INPUT: Google Maps and other highway and street GPS services have replaced what? OPTIONS: (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans OUTPUT: {target}

Table 1: Examples of prompts used in our experiments. Contents that are characteristic to a prompt are **bolded**.

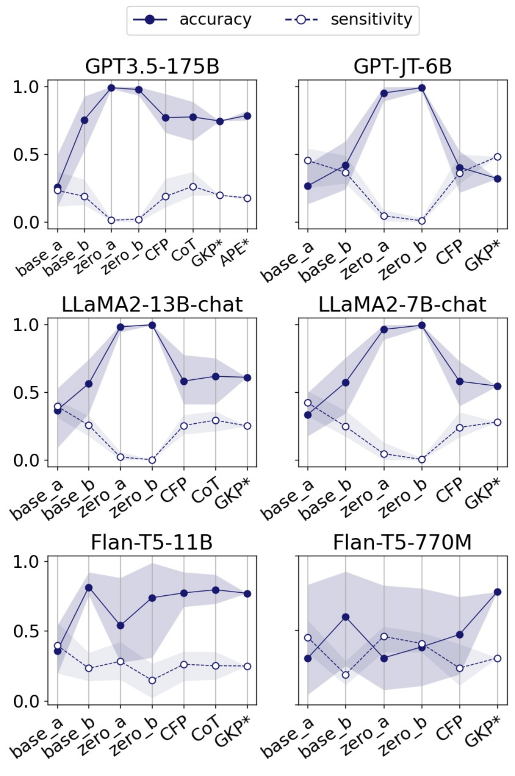


Figure 2: The average accuracy and sensitivity of each model using various prompts across different datasets. * indicates prompts that are not tested on all datasets.

4 Results

Figure 2 shows the average accuracy and sensitivity of each model using various prompts across different datasets.[5] The *Pearson* correlation coefficient shows a strong negative correlation between accuracy and sensitivity ($r = -0.8764$, p-value $\ll 0.01$). Sensitivity can be viewed as an unsupervised proxy of accuracy given such a strong correlation.

It is interesting to note that both Flan-T5 models failed with zero_a and zero_b. We discuss this further in Section 4.2.

[5]Due to limited space, we did not include the standard deviations of the statistics in the following plots and tables. For more results, please refer to https://github.com/UKPLab/naacl2024-prompt-sensitivity.

4.1 Instruction, knowledge, *chain-of-thought*

This section compares the effects of instruction (human-designed instructions in base_b and LM-generated instructions in APE), reasoning chain (in CoT), and knowledge (in GKP).

Table 2 compares the performance of models using base_b and APE. The two prompts lead to similar accuracy and sensitivity on CoLA and RTE, suggesting that human-designed and LM-generated instructions have similar effects on the model. At least for CoLA and RTE, there is no need to generate instructions using an LM.

dataset	prompt	accuracy↑	sensitivity↓
CoLA	base_b	**0.8235**	**0.1830**
	APE	0.8216	0.1960
RTE	base_b	0.6931	**0.1377**
	APE	**0.7509**	0.1603

Table 2: The accuracy and sensitivity of GPT3.5-175B using base_b and APE.

Table 3 shows that GKP leads to a higher accuracy and lower sensitivity in most cases. This suggests that the effects of instructions and LM-generated knowledge are cumulative, i.e., plac-

model	accuracy↑		sensitivity↓	
	base_b	GKP	base_b	GKP
GPT3.5-175B	**0.8000**	0.7459	0.3133	**0.1982**
GPT-JT-6B	0.2413	**0.3202**	0.4912	**0.4831**
LLaMA2-13B	0.6085	**0.6109**	0.3257	**0.2511**
LLaMA2-7B	0.5276	**0.5456**	0.3649	**0.2804**
Flan-T5-11B	**0.8057**	0.7697	0.3435	**0.2491**
Flan-T5-770M	0.2607	**0.7601**	**0.2345**	0.3141
AVERAGE	0.5082	**0.6103**	0.3485	**0.3082**

Table 3: The accuracy and sensitivity using base_b and GKP on CSQA. LLaMA2-13B and LLaMA2-7B are LLaMA2-13B-chat and LLaMA2-7B-chat.

ing LM-generated knowledge before instructions yields better results.

Table 4 shows that CoT leads to a similar accuracy but higher sensitivity compared to base_b. As shown in Table 1, instructions are also contained in CoT. Unlike LM-generated knowledge, reasoning chains do not bring performance gain on top of instructions.

model	accuracy↑		sensitivity↓	
	base_b	CoT	base_b	CoT
GPT3.5-175B	0.7549	**0.7758**	**0.1899**	0.2636
LLaMA2-13B	0.5642	**0.6179**	**0.2564**	0.2939
Flan-T5-11B	**0.8134**	0.7943	**0.2328**	0.2509
AVERAGE	0.7108	**0.7293**	0.2264	**0.2695**

Table 4: The accuracy and sensitivity of GPT3.5-175B, LLaMA2-13B-chat (LLaMA2-13B), and Flan-T5-11B using base_b and CoT across different tasks.

We designed CoT_base_a to isolate the effect of reasoning chains. CoT_base_a is a combination of *chain-of-thought* and the most basic base_a (see Table 12 in A.3).

Figure 3 shows that CoT_base_a outperforms base_a, but it performs worse than base_b in most cases. This suggests that for CoLA and RTE, while reasoning chains do help improve performance, it is not as effective as instructions.

We note that models using CoT_base_a perform relatively better on CoLA than on RTE. We speculate that this is because the demonstrations in CoT_base_a for CoLA contain an equivalent of an instruction. Since CoLA is a binary classification task with GRAMMATICAL and UNGRAMMATICAL being the labels, a reasoning chain, such as the one exemplified in Table 12, contain an explicit label mapping, which may function similarly to an instruction.

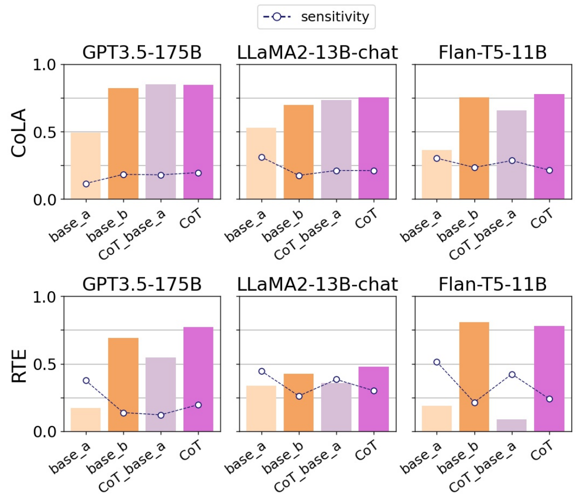


Figure 3: The accuracy and sensitivity of different models using base_a, base_b, CoT_base_a, and CoT.

4.2 What happened to Flan-T5 with zero?

As shown in Figure 2, unlike other models, both Flan-T5-11B and Flan-T5-770M failed with zero_a and zero_b. We examine the outputs of Flan-T5-11B closely to investigate this counter-intuitive phenomenon, and find that Flan-T5-11B tends to produce text answers instead of numeric indices with zero_a and zero_b (see Figure 8 in A.4). Our observations suggest that Flan-T5 models are not good at mapping the numeric indices in the examplars to their output spaces, unless explicitly instructed via instructions such as in base_b and APE, or in the form of OPTIONS, such as in GKP. This conclusion is also supported by the observation that Flan-T5 models fail to produce numeric indices with base_a (see Table 14 in A.4), where explicit instructions are also lacking.

4.3 The effect of decoding strategies

It has been shown that decoding strategies influence the quality of LLM generations (Lee et al., 2022; Wang et al., 2023d). We also observe that decoding strategies have an effect on sensitivity. Figure 4 shows that the overall sensitivity calculated from predictions obtained using greedy decoding is lower than that of those obtained using Top-k sampling (Fan et al., 2018). A strong negative correlation between accuracy and sensitivity is still observed in this case ($r = -0.5507$, p-value $\ll 0.01$). This is lower than that of Top-k sampling ($r = -0.8764$, p-value $\ll 0.01$), suggesting that Top-k sampling has a "magnifying" effect on sensitivity.

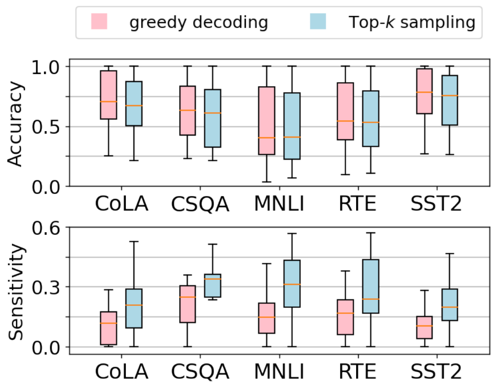


Figure 4: The accuracy and sensitivity of predictions obtained using greedy decoding and Top-k sampling across different models.

In Top-k sampling, the next token is sampled from a probability mass redistributed among tokens that have the highest k probabilities. The observation that Top-k sampling leads to a higher level of sensitivity indicates that for instances with high sensitivity, the output probabilities of different labels are close.

4.4 Open-ended generation

Measuring (or even defining) sensitivity in open-ended generation can be challenging. Two pieces of generated text can convey the same meaning even if they vary significantly in terms of word choices and length. We circumvent this issue by selecting an open-ended generation task where the output is more "controllable." Specifically, we chose GSM8K, an arithmetic reasoning task in which the outputs are numbers (Cobbe et al., 2021). Despite being an open-ended generation task, the numerical format of the outputs in GSM8K allows us to use variation-ratio to measure sensitivity. Our results reveal that there is also a negative correlation between accuracy and sensitivity in the open-ended setting. See A.5 for more details.

5 Gradient-based saliency scores

In light of recent studies that link ICL and implicit gradient descent (Akyürek et al., 2023; Li et al., 2023; Von Oswald et al., 2023; Zhang et al., 2023), we investigate the relationship between sensitivity and gradient. We use gradient-based saliency scores, which reveal the relevance of input tokens to the model prediction (Simonyan et al., 2014; Li et al., 2016; Yin and Neubig, 2022). The higher the score is, the more a token is supposed to contribute to the model output. We compute gradient-based saliency scores based on the norm of the gradient of the model output. The gradient g for a token x_i in an input $\boldsymbol{x}$ is calculated as follows:

$$g(x_i) = \nabla_{x_i} M(y|\boldsymbol{x}), \tag{2}$$

where $M(y|\boldsymbol{x})$ is the logit for the output token y. The saliency score $S(x_i)$ is obtained by taking the L1 norm of $g(x_i)$:

$$S(x_i) = \|g(x_i)\|_{L1}. \tag{3}$$

We calculate the saliency scores using GPT-JT-6B, GPT-J-6B, Flan-T5-770M, and T5-770M. Figure 5 gives an example of saliency scores over tokens, which shows that the tokens in the prompt are more relevant to the output than tokens in the input (see Figure 10 in A.6 for more examples).

To perform a quantitative analysis, we segment an instance into several parts. Take Figure 6 as an example.

SENTENCE: $\underbrace{\boxed{\text{Julie and Jenny arrived first.}}}_{\textit{input}}$

QUESTION: Is this (0) unacceptable, or (1) acceptable?

ANSWER: 1

Figure 6: An example of token segmentation for instances with `base_b`.

For a sentence X, tokens where perturbations happen are referred to as X_{input} or *input* tokens (such as the *input* in Figure 6), and the rest of the sentence is referred to as X_{prompt} or *prompt* tokens.[6] We calculate the *mean saliency score*, denoted as $\overline{S}$, for input tokens and prompt tokens respectively:

$$\overline{S} = \frac{\sum_{x_i \in X} S(x_i)}{n}, X \in \{X_{input}, X_{prompt}\}, \tag{4}$$

where n is the number of tokens in X_{input} or X_{prompt}.

Table 5 shows the average *mean saliency scores* of instances in different datasets. Similar to Figure 5, the *mean saliency scores* of *input* tokens are consistently lower than those for *prompt* tokens. There is a strong negative correlation between $\overline{S}_p - \overline{S}_i$ and sensitivity ($r = -0.7596$, p-value $\ll 0.01$).[7] This explains why `base_b`, `zero_b`, and `CFP` lead to lower levels of sensitivity: perturbations are only done to *input* tokens, which are less relevant to

[6] For the segmentation of other prompts, please refer to Figure 9 in A.6.

[7] The *Pearson* correlation coefficient for $\overline{S}_p - \overline{S}_i$ and sensitivity without `zero_b` results is -0.5733 ($p = 0.0831$).

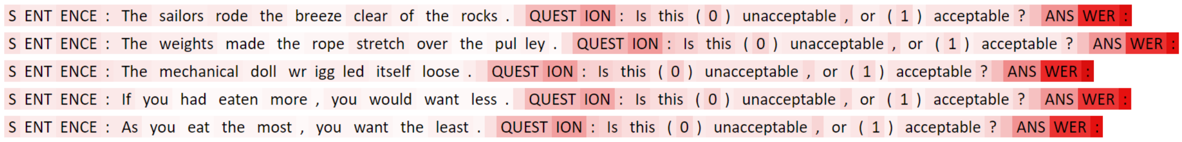


Figure 5: Saliency scores over tokens of CoLA instances with `base_b` obtained using GPT-6B-JT.

the outputs than *prompt* tokens. We also find that instruction tuned models "focus" more on *prompt* tokens than their non-instruction tuned counterparts (see Table 16 in A.6).

dataset	prompt	$\overline{S}_i$	$\overline{S}_p$	$\overline{S}_p - \overline{S}_i$	sens.↓
CoLA	base_b	4.17	12.74	8.57	30.15
	zero_b	7.65	23.66	16.01	0.00
	CFP	3.75	11.43	7.68	28.39
CSQA	base_b	2.68	7.41	4.73	48.84
	zero_b	2.25	8.02	5.77	0.00
	CFP	2.37	7.06	4.69	49.94
MNLI	base_b	3.10	12.79	9.69	43.78
	zero_b	3.61	17.18	13.57	3.06
	CFP	1.68	6.71	5.03	42.70
RTE	base_b	1.95	9.48	7.53	30.61
	zero_b	3.22	18.95	15.73	0.36
	CFP	1.53	7.32	5.79	29.82
SST2	base_b	3.09	12.59	9.50	28.17
	zero_b	4.99	22.32	17.33	0.25
	CFP	2.77	10.70	7.93	28.83

Table 5: The average *mean saliency scores* of *input* tokens ($\overline{S}_i$, in permillage), *prompt* tokens ($\overline{S}_p$, in permillage) of instances, the difference between the two scores (Δ), and sensitivity (**sens.**, in percentage) obtained using GPT-JT-6B.

For `GKP`, we perform a more detailed segmentation for a better analysis, which is shown in Figure 7. We segment an instance with `GKP` to *knowledge*, *input*, *option*, and *prompt* tokens.

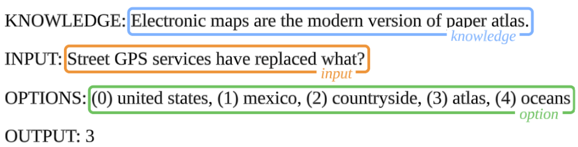


Figure 7: An example of token segmentation for instances with `GKP`.

Table 6 shows the average *mean saliency scores* of tokens in CSQA instances using `GKP`. Similar to Table 5, *input* tokens are less relevant to the predictions. Note that *knowledge* tokens have the lowest average *mean saliency scores*, suggesting that generated knowledge is not very relevant to the predictions.

$\overline{S}_{input}$	$\overline{S}_{knowledge}$	$\overline{S}_{option}$	$\overline{S}_{prompt}$
4.33	2.56	6.37	12.86

Table 6: The average *mean saliency scores* ($\overline{S}$) of *input*, *knowledge*, *option*, and *prompt* tokens of CSQA instances with `GKP` obtained using GPT-JT-6B.

The percentage of $\overline{S}_{input}$ for `base_b` (27.2%) is higher than that for `GKP` (16.6%), which indicates that *input* tokens in `base_b` are relatively more relevant to the predictions than those in `GKP`. This is consistent with the observation in Table 3, that `GKP` leads to a lower level of sensitivity in most cases.

We also examine the *mean saliency scores* of ground truth tokens in `zero_b` (see Table 17 in A.6). The results show that Flan-T5-770M "focuses" less on ground truth tokens than GPT-JT-6B, which explains the failure of Flan-T5 models with `zero` prompts discussed in Section 4.2.

Prompt tokens are important in the sense that they provide information such as instructions and external knowledge, which are necessary for the model to produce outputs that the user expects. However, it is counter-intuitive that the *mean saliency scores* for *prompt* tokens are much higher than those for *input* tokens, as *input* tokens contain essential information as well. This observation may imply that memory plays an overwhelming role in ICL (Chen et al., 2022; Merullo et al., 2023; McKenna et al., 2023; Štefánik and Kadlčík, 2023; Singh et al., 2023)–LLMs were trained on similar instances, so they do not need to rely on *input* tokens in the test instances too much. In this sense, *prompt* tokens are more relevant because they trigger memories, and models implicitly infer task information from them (Reynolds and McDonell, 2021; Hendel et al., 2023; Wang et al., 2023c; Wolf et al., 2023). Over-emphasis on *prompt* tokens may lead to hallucination as well. A recent study discovers that instruction tuning significantly increases sycophancy in LLMs, that they follow user's opinion or agree with user's claim even when they know it is false (Wei et al., 2023).

dataset	prompt	GPT-JT-6B		LLaMA2-13B-chat		LLaMA2-7B-chat		Flan-T5-11B		Flan-T5-770M	
		sad	*greedy*	*sad*	*greedy*	*sad*	*greedy*	*sad*	*greedy*	*sad*	*greedy*
CoLA	base_a	45.16	38.71	59.20	59.20	52.37	52.56	29.41	29.41	41.75	40.99
	base_b	48.01	47.82	70.21	70.40	62.62	63.38	77.42	81.59	71.54	71.92
	CFP	38.71	34.91	67.93	67.74	68.31	68.88	77.61	80.65	69.45	69.45
CSQA	base_a	26.48	26.56	45.49	46.80	33.20	33.93	57.79	55.82	79.02	83.77
	base_b	30.74	30.74	61.97	62.13	54.59	55.49	78.85	82.30	24.92	25.41
	CFP	27.70	28.20	60.98	61.72	54.59	54.51	77.30	81.64	22.79	22.87
MNLI	base_a	14.90	11.50	9.20	3.40	16.90	17.20	23.50	25.80	9.10	3.70
	base_b	27.30	27.00	34.30	30.30	37.00	34.10	81.60	82.30	59.80	65.10
	CFP	33.50	34.20	43.90	39.70	39.90	40.40	81.40	82.90	14.60	9.10
RTE	base_a	37.55	31.05	38.99	39.71	34.66	32.85	24.19	23.83	22.38	21.66
	base_b	58.12	58.48	53.07	37.55	62.82	63.18	82.31	87.00	50.54	54.15
	CFP	52.71	50.18	56.32	43.68	56.32	56.32	75.09	85.56	44.77	53.79
SST2	base_a	25.34	27.29	59.98	59.52	61.12	61.47	50.80	50.80	40.14	40.94
	base_b	76.95	78.44	77.64	77.41	78.10	78.44	90.02	95.64	91.63	91.97
	CFP	53.56	54.13	88.07	88.19	77.06	76.49	91.17	95.87	77.87	78.10

Table 7: The highest accuracy reached using *sensitivity-aware* decoding (*sad*) and the accuracy of greedy decoding (*greedy*). Cases where *sensitivity-aware* decoding has a better accuracy than greedy decoding are highlighted.

6 *Sensitivity-aware* decoding

We showed in Section 4 that sensitivity can be viewed as an unsupervised proxy of accuracy. In this section, we further show that including sensitivity in decoding improves model performance. Specifically, we add sensitivity as a penalty to greedy decoding:

$$\hat{y} = \underset{y \in V}{\operatorname{argmax}}[\alpha P(y|x) - (1 - \alpha)s], \qquad (5)$$

where x is an input, V is a vocabulary, and $\hat{y}$ is the output. $P(y|x)$ is the probability of an output y given x, and s is a sensitivity estimation, calculated as the variance of the output logits of the synthetic data for x. We reweight $P(y|x)$ and s using α and $(1 - \alpha)$.

Table 7 summarizes the performance of *sensitivity-aware* decoding compared to greedy decoding. *Sensitivity-aware* decoding works better on CoLA, MNLI, and RTE, and model-wisely, it works better with GPT-JT-6B and LLaMA2-13B-chat. *Sensitivity-aware* decoding works much better with base_a than the other prompts. See A.7 for more implementation details and results.

The results show that penalizing outputs with high sensitivity in decoding has an effect on model performance. We show that *sensitivity-aware* decoding works better with the most basic prompt, base_a, that contains plain *input-target* pairs. We believe this prompt represents those truly challenging problems in real life that have almost no clues or hints, and require very strong reasoning abilities to solve. Apparently, none of the models we tested possess such abilities. Therefore, *sensitivity-aware* decoding, which helps improve model performance under this "extreme condition," is highly meaningful in the present context.

However, *sensitivity-aware* decoding is more computationally expensive compared to standard greedy decoding as it requires multiple inference passes. This makes it impractical for tasks that demand low latency.

7 Conclusion

This work provides a novel perspective on prompt analysis, examining the effects of prompts in terms of the sensitivity of a function. We conducted a systematic and comprehensive analysis, and highlight how certain prompts are more effective due to their ability to reduce sensitivity levels. We show that sensitivity can serve as an unsupervised proxy of model performance, making it a valuable tool for evaluating model performance without using labeled data or ground truth. By introducing *sensitivity-aware* decoding, we show that incorporating sensitivity in greedy decoding is particularly helpful in cases where the input is less informative. Since none of the models we tested performs well when there is limited information in the input, we believe *sensitivity-aware* decoding is highly practical in the current context. Our work not only sheds light on prompt engineering, but also provides insight into the working mechanism of ICL.

Limitations

The measurement of sensitivity is currently quite restricted to close-ended generation. It is challenging to extend this framework to tasks such as text summarization. While we demonstrate the effectiveness of *sensitivity-aware* decoding, it requires multiple inferences, which may be impractical for tasks that require low latency. In order to manage costs, we limited the use of OpenAI `text-davinci-003` (GPT3.5-175B) in our experiments. Due to its closed source nature, the reproducibility of the results related to GPT3.5-175B may be a concern as well.

Acknowledgements

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

Anirudh Ajith, Chris Pan, Mengzhou Xia, Ameet Deshpande, and Karthik Narasimhan. 2023. Instructeval: Systematic evaluation of instruction selection methods. *arXiv preprint arXiv:2307.00259*.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. 2023. Simplicity bias in transformers and their ability to learn sparse Boolean functions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5767–5791, Toronto, Canada. Association for Computational Linguistics.

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.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144, Toronto, Canada. Association for Computational Linguistics.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving in-context few-shot learning via self-supervised training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3558–3573, Seattle, United States. Association for Computational Linguistics.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2023. On the relation between sensitivity and accuracy in in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 155–167, Singapore. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Linton C Freeman. 1965. *Elementary applied statistics: for students in behavioral science*. New York: Wiley.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10136–10148. Association for Computational Linguistics.

Michael Hahn and Navin Goyal. 2023. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*.

Michael Hahn, Dan Jurafsky, and Richard Futrell. 2021. Sensitivity as a complexity measure for sequence classification tasks. *Transactions of the Association for Computational Linguistics*, 9:891–908.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9318–9333. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. 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*.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Yingcong Li, M. Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

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

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022b. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

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.

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

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 2758–2774. Association for Computational Linguistics.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models. *arXiv preprint arXiv:2305.16130*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Kyle Richardson and Ashish Sabharwal. 2022. Pushing the limits of rule reasoning in transformers through natural language satisfiability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11209–11219.

Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Aaditya K. Singh, Stephanie C. Y. Chan, Ted Moskovitz, Erin Grant, Andrew M. Saxe, and Felix Hill. 2023. The transient nature of emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Michal Štefánik and Marek Kadlčík. 2023. Can in-context learners learn a reasoning concept from demonstrations? In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 107–115, Toronto, Canada. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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*, pages 35151–35174. PMLR.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023c. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. 2023. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Noam Wies, Yoav Levine, and Amnon Shashua. 2023. The learnability of in-context learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yotam Wolf, Noam Wies, Yoav Levine, and Amnon Shashua. 2023. Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. 2023. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*.

Xinran Zhao, Shikhar Murty, and Christopher Manning. 2022. On measuring the intrinsic few-shot hardness of datasets. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3955–3963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023a. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 14544–14556. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023b. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A Appendix

A.1 More on experiment settings

All experiments were done in the few-shot setting.[8] We set `temperature=0.8` for all experiments. We set `max_new_tokens=2`, except for experiments with `CoT`, where it was set to 64, and 128 for experiments with GSM8K. For local models, we set `batch_size=16` and `seeds=[2266,105,86379]`, and all experiments were run on either an NVIDIA A100 or H100. The experiments with OpenAI `text-davinci-003` were done between July 8 and July 18, 2023.

We notice that the synthetic data for CSQA are noisier than those for other datasets. We manually checked the synthetic data for the first 50 instances in CSQA (a total of 1220 instances). There are 223 synthetic data for the first 50 instances, among which 44 are considered to be noisy (see Table 8).

We designed `base_a`, `base_b`, `zero_a`, and `zero_b` in an intuitive way. We did not rely on any formal theories or guidelines related to prompt engineering. Table 9 shows examples of demonstrations with different prompts.

original	**synthetic**
Where would you find magazines along side many other printed works?	:-Where would you waste your Spanish ?? !?
Where are you likely to find a hamburger?	prime over byEL are you "happy" hamburger?
James was looking for a good place to buy farmland. Where might he look?	ei– What look is for a good place to buy – like a restaurant?
	<..."What might he look?
In what Spanish speaking North American country can you get a great cup of coffee?	Balkan Kingdom <Meditation a French speaking North American country can you explain the ways components are used?
	///5// Does insurance cover a car, Sears, or a cup of coffee?
What do animals do when an enemy is approaching?	19:: What do animals do when an enemy is spotted?
	com... What is an enemy of HTML?
	p-0 What do you do when an enemy moves in?
Reading newspaper one of many ways to practice your what?	=- Your WHERE? Your ARE. Your what?
	an tissues/Reading newspaper/What is a red/aloe?
If you want harmony, what is something you should try to do with the world?	Sept: 5If you want to have a process then what do you now do with the system?
Where does a heifer's master live?	=- does a heifer take in valuables?
	/Jazeeraio "What's a Maru?
Aside from water and nourishment what does your dog need?	/ Purity Utah Prison of 2015?
Janet was watching the film because she liked what?	///Janet was watching the film because she liked what?
	08/Janet was watching the film "Sha what?
What are you waiting alongside with when you're in a reception area?	27.9What are you waiting alongside with when you're in a job interview?
	1: ⊠⊠ hat are you going to do while on the street a day like now?
When drinking booze what can you do to stay busy?	When drinking : When more can you do then pleasure ?
	When drinking ? When being a kid/ Young man ?
	=-Bin Bak Allah? Bern Abd Ja Tih?
A fencing thrust with a sharp sword towards a person would result in what?	2, 45A fencing thrust with a sharp sword towards a person would result in pain?
	Video Audio AudioA fencing thrust from the guitarist/band member whether way or the other?

[8] See https://github.com/UKPLab/naacl2024-prompt-sensitivity/tree/main/prompts.

original	**synthetic**
Unlike a spider and his many sight seers, people only have what?	") cards Of the spider spider and his many sight seers. We Who See Everything?
What could go on top of wood?	{50 What you could go wrong with this title?
The artist was sitting quietly pondering, then suddenly he began to paint when what struck him?	=- Imagine an artist was sitting quietly pondering, then taking a break to paint when what struck him?
Where could you find a toilet that only friends can use?	=—Where could you find a toilet maker with a slot underneath?
What is someone who isn’t clever, bright, or competent called?	1. Why are aren ⊠⊠ clever, bright, or competent called?
	s// ? Him who isn’t clever, ? peculiar, ?
When wildlife reproduce we often refer to what comes out as what?	(Do we often refer to wha comes out as what?
	div 2: were you aware we often refer to what comes out as what?
	",.."?
Blue read material outside of his comfort zone because he wanted to gain what?	//Blue read material/ text/ flag/ license/ licensing/ gain what?
After he got hired he hoped for success at his what?	/2011-2012-2013. Who is his what?
Committing perjury is a serious what?	ned!? ⊠?perjury is involved here??
	///committing test ? Now what?
The lock kept the steering wheel from moving, but the thief still took his chances and began to work on the what?	=– Maybe one lock kept the steering wheel from moving, but the thief still had the gun. If not, what was it?
Who is a police officer likely to work for?	based? Do you consider them likely to be sustainable?
Where is a doormat likely to be in front of?	Pop RSWhere is a substitute to an already terminated system in comparison?
He needed more information to fix it, so he consulted the what?	1. 13He needed more information to fix information, but defacto what?
Where can you put a picture frame when it’s not hung vertically?	%%%Where can you put a picture frame when it’s not nesting?
	min "Where can you be if your wife or any other object is hung vertically?
What must someone do before they shop?	/// How must someone conduct their work daily?

Table 8: Noisy synthetic data for the first 50 instances in CSQA.

dataset	prompt	text
CSQA	base_a	Google Maps and other highway and street GPS services have replaced what? (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans 3
	base_b	SENTENCE: Google Maps and other highway and street GPS services have replaced what? QUESTION: Is it (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans? ANSWER: 3
	zero_a	Google Maps and other highway and street GPS services have replaced what? (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans The answer is 3. 3
	zero_b	SENTENCE: Google Maps and other highway and street GPS services have replaced what? OPTIONS: (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans The answer is 3. ANSWER: 3
	CFP	Bob said, "Google Maps and other highway and street GPS services have replaced what?" QUESTION: Is it (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans in Bob's opinion? ANSWER: 3
	CoT	SENTENCE: Google Maps and other highway and street GPS services have replaced what? QUESTION: Is it (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans? ANSWER: Let's think step by step. Google Maps and other highway and street GPS services help people find their location and navigate streets and highways, and atlas is a software that is designed to help users navigate. So the answer is 3.
	GKP	KNOWLEDGE: Electronic maps are the modern version of paper atlas. INPUT: Google Maps and other highway and street GPS services have replaced what? OPTIONS: (0) united states, (1) mexico, (2) countryside, (3) atlas, (4) oceans OUTPUT: 3
MNLI	base_a	He thought about ways to achieve this life goal for a long time, which means until he learned the basics of text editing, which happened at his first job at a firm trading in plastic bags landfill disposal permits. He thought about ways to achieve his life goals for a long time. 1
	base_b	SENTENCE1: He thought about ways to achieve this life goal for a long time, which means until he learned the basics of text editing, which happened at his first job at a firm trading in plastic bags landfill disposal permits. SENTENCE2: He thought about ways to achieve his life goals for a long time. ANSWER: 1
	zero_a	He thought about ways to achieve this life goal for a long time, which means until he learned the basics of text editing, which happened at his first job at a firm trading in plastic bags landfill disposal permits. He thought about ways to achieve his life goals for a long time. The answer is 1. 1
	zero_b	SENTENCE1: He thought about ways to achieve this life goal for a long time, which means until he learned the basics of text editing, which happened at his first job at a firm trading in plastic bags landfill disposal permits. SENTENCE2: He thought about ways to achieve his life goals for a long time. The answer is 1. ANSWER: 1
	CFP	Bob said, "sentence 1 is 'He thought about ways to achieve this life goal for a long time, which means until he learned the basics of text editing, which happened at his first job at a firm trading in plastic bags landfill disposal permits,' and sentence 2 is 'He thought about ways to achieve his life goals for a long time.'" QUESTION: Are the two sentences (0) contradiction, (1) entailment, or (2) neutral in Bob's opinion? ANSWER: 1
	CoT	SENTENCE1: He thought about ways to achieve this life goal for a long time, which means until he learned the basics of text editing, which happened at his first job at a firm trading in plastic bags landfill disposal permits. SENTENCE2: He thought about ways to achieve his life goals for a long time.

dataset	prompt	text
		QUESTION: Is this (0) contradiction, (1) entailment, or (2) neutral? ANSWER: Let's think step by step. Sentence 1 states that he thought about ways to achieve his life goal for a long time, and then states that he learned the basics of text editing at his first job. Sentence 2 states that he thought about ways to achieve his life goal for a long time. Both sentences state the same thing, so the answer is 1.
RTE	base_a	The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. President Clinton thinks that industrial espionage is a threat to America's well being. 0
	base_b	SENTENCE1: The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. SENTENCE2: President Clinton thinks that industrial espionage is a threat to America's well being. QUESTION: Are the two sentences (0) entailment, or (1) not_entailment? ANSWER: 0
	zero_a	The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. President Clinton thinks that industrial espionage is a threat to America's well being. The answer is 0. 0
	zero_b	SENTENCE1: The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. SENTENCE2: President Clinton thinks that industrial espionage is a threat to America's well being. The answer is 0. ANSWER: 0
	CFP	Bob said, "sentence 1 is 'The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests.', and sentence 2 is 'President Clinton thinks that industrial espionage is a threat to America's well being.'" QUESTION: Are the two sentences (0) entailment, or (1) not_entailment in Bob's opinion? ANSWER: 0
	CoT	SENTENCE1: The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. SENTENCE2: President Clinton thinks that industrial espionage is a threat to America's well being. QUESTION: Are the two sentences (0) entailment, or (1) not_entailment? ANSWER: Let's think step by step. The first sentence provides information about President Clinton's belief, which directly leads to the FBI conducting an investigation. Therefore, the second sentence necessarily follows the information provided in the first sentence. So the answer is 0.
	APE	INSTRUCTION: determine whether the given statement logically follows from the preceding statement, and the output is either (0) entailment if the statement logically follows, or (1) not_entailment if the statement does not logically follow. INPUT: The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. President Clinton thinks that industrial espionage is a threat to America's well being. OUTPUT: 0
SST2	base_a	None of this is meaningful or memorable , but frosting is n't , either , and you would n't turn down a big bowl of that , would you ? 0
	base_b	SENTENCE: None of this is meaningful or memorable , but frosting is n't , either , and you would n't turn down a big bowl of that , would you ? QUESTION: Is this (0) negative, or (1) positive? ANSWER: 0
	zero_a	None of this is meaningful or memorable , but frosting is n't , either , and you would n't turn down a big bowl of that , would you ?

dataset	prompt	text
		The answer is 0. 0
	zero_b	SENTENCE: None of this is meaningful or memorable , but frosting is n't , either , and you would n't turn down a big bowl of that , would you ? The answer is 0. ANSWER: 0
	CFP	Bob said, "None of this is meaningful or memorable , but frosting is n't , either , and you would n't turn down a big bowl of that , would you ?" QUESTION: Is this (0) negative, or (1) positive in Bob's opinion? ANSWER: 0
	CoT	SENTENCE: None of this is meaningful or memorable , but frosting is n't , either , and you would n't turn down a big bowl of that , would you ? QUESTION: Is this (0) negative, or (1) positive? ANSWER: Let's think step by step. "None of this is meaningful or memorable" suggests that whatever is being referred to is not significant or noteworthy, indicating a negative sentiment. So the answer is 0.
GSM8K	base_b	SENTENCE: Sally and Bob have made plans to go on a trip at the end of the year. They both decide to work as babysitters and save half of what they've earned for their trip. If Sally makes $6 per day and Bob makes $4 per day, how much money will they both have saved for their trip after a year? ANSWER: 1825
	zero_b	SENTENCE: Sally and Bob have made plans to go on a trip at the end of the year. They both decide to work as babysitters and save half of what they've earned for their trip. If Sally makes $6 per day and Bob makes $4 per day, how much money will they both have saved for their trip after a year? The answer is 1825. ANSWER: 1825
	CoT	SENTENCE: Sally and Bob have made plans to go on a trip at the end of the year. They both decide to work as babysitters and save half of what they've earned for their trip. If Sally makes $6 per day and Bob makes $4 per day, how much money will they both have saved for their trip after a year? ANSWER: Let's think step by step. Saly saves 1/2 * $6/day = 3/day. Since each year have 365 days, the total amount of money Sally will save in a year is $3/day * 365 days/year = 1095/year. Bob saves 1/2 * $4/day = 2/day. The total amount of money Bob will have saved in a year is $2/day * 365 days/year = 730/year In total, Sally and Bob would have saved $730 + $1095 = 1825. So the answer is 1825.

Table 9: Examples of demonstrations with different prompts. Demonstrations for CoLA are exemplified in Table 1, so they are not included here.

A.2 More on LLaMA results

LLaMA2 models were not available when we began our experiments. As a result, we conducted our initial experiments using LLaMA-7B, LLaMA-13B, and LLaMA-30B (Touvron et al., 2023a). Table 10 shows a comparison between LLaMA2 and LLaMA models.

model	accuracy↑	sensitivity↓
LLaMA2-13B-chat	0.6961	0.1883
LLaMA2-7B-chat	0.6843	0.1951
LLaMA-30B	0.6835	0.2324
LLaMA-13B	0.6177	0.2327
LLaMA-7B	0.6010	0.2457

Table 10: The average accuracy and sensitivity of LLaMA models across different tasks.

A.3 More on instruction, *chain-of-thought*, and knowledge

Table 11 shows the average performance across different models using `base_a` and `base_b`. Table 12 shows examples of `CoT_base_a`.

dataset	prompt	accuracy	sensitivity	accuracy +/-	sensitivity +/-
CoLA	base_a	0.4215	0.3317	0.2275	0.1125
	base_b	0.6490	0.2192		
CSQA	base_a	0.4652	0.4107	0.0235	0.0587
	base_b	0.4887	0.3520		
MNLI	base_a	0.1317	0.4498	0.3258	0.1405
	base_b	0.4575	0.3094		
RTE	base_a	0.2332	0.5120	0.3487	0.2763
	base_b	0.5819	0.2357		
SST2	base_a	0.3841	0.4198	0.3992	0.2318
	base_b	0.7833	0.1880		

Table 11: The average accuracy and sensitivity across different models, and the difference between the accuracy (**accuracy +/-**) and sensitivity (**sensitivity +/-**) of models using `base_a` and `base_b`.

dataset	text
CoLA	I'm glad I saw anybody. Let's think step by step. This sentence is ungrammatical because "anybody" is used as the object in an affirmative clause. So the answer is 0.
RTE	The Federal Bureau of Investigation started an independent probe of the circumstances shortly after the White House made plain that President Bill Clinton considered industrial espionage a particular threat to US economic interests. President Clinton thinks that industrial espionage is a threat to America's well being. Let's think step by step. The first sentence provides information about President Clinton's belief, which directly leads to the FBI conducting an investigation. Therefore, the second sentence follows the information provided in the first sentence. So the answer is 0.

Table 12: Examples of `CoT_base_a` for CoLA and RTE.

A.4 More on Flan-T5 with `zero`

Figure 8 shows the performance of Flan-T5-11B using the `zero` prompts. Flan-T5-11B fails to output a numeric index on CSQA, MNLI, and RTE instances in most cases. Table 13 shows the performance of Flan-T5 models using `zero_b`. Table 14 shows the original number of instances in each dataset and the number of instances where Flan-T5 models output a numeric index using `base_a`.

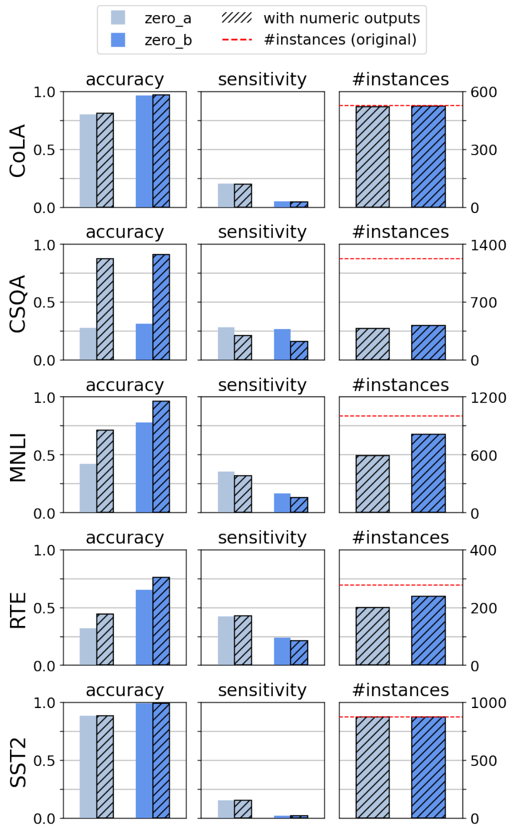


Figure 8: The accuracy and sensitivity of Flan-T5-11B on the full dataset using `zero_a` and `zero_b`, and the accuracy and sensitivity of Flan-T5-11B on instances where it outputs a numeric index as what is exemplified in the prompt. `#instances` shows the number of instances in a dataset (`#instances (original)`) and the number of instances where Flan-T5-11B outputs a numeric index (`with numeric outputs`).

dataset	**accuracy↑**		**sensitivity↓**	
	Flan-T5-11B	Flan-T5-770M	Flan-T5-11B	Flan-T5-770M
CoLA	0.9620	0.4307	0.0467	0.3970
CSQA	0.3090	0.7833	0.2668	0.3516
MNLI	0.7787	0.2397	0.1659	0.4619
RTE	0.6498	0.1264	0.2366	0.4864
SST2	0.9881	0.3612	0.0203	0.3624
AVERAGE	0.7375	0.3883	0.1473	0.4119

Table 13: The accuracy and sensitivity of Flan-T5-11B and Flan-T5-770M on different tasks with `zero_b`. Flan-T5-11B reaches comparable performance to GPT and LLaMA models only on CoLA and SST2.

dataset	**#original**	**#numeric**	
		Flan-T5-11B	Flan-T5-770M
CoLA	527	496	495
CSQA	1220	806	1218
MNLI	1000	639	924
RTE	277	169	208
SST2	872	848	497

Table 14: The original number of instances in each dataset (**#original**) and the number of instances where Flan-T5-11B and Flan-T5-770M output a numeric index using `base_a` (**#numeric**).

A.5 More on open-ended generation

GSM8K is an arithmetic reasoning task in which the outputs are numbers (Cobbe et al., 2021):

> QUESTION: Sally and Bob have made plans to go on a trip at the end of the year. They both decide to work as babysitters and save half of what they've earned for their trip. If Sally makes $6 per day and Bob makes $4 per day, how much money will they both have saved for their trip after a year?
> ANSWER: 1825

See Table 9 in A.1 for more examples. Table 15 shows the performance of LLaMA2-13B-chat and Flan-T5-11B on GSM8K. There is also a negative correlation between accuracy and sensitivity in open-ended generation. Note that Flan-T5-11B again fails to perform using `zero_b`, which is consistent with the results in Section 4.2.

model	prompt	accuracy↑	sensitivity↓
LLaMA2-13B-chat	`base_b`	0.0612	0.6226
	`zero_b`	0.9007	0.0015
	`CoT`	0.2570	0.6115
Flan-T5-11B	`base_b`	0.0425	0.6635
	`zero_b`	0.5868	0.2802
	`CoT`	0.1208	0.6779

Table 15: The accuracy and sensitivity of LLaMA2-13B-chat and Flan-T5-11B on GSM8K.

A.6 More on gradient-based saliency scores

Figure 9 show examples of token segmentation for instances with `zero_b` and `CFP`. Figure 10 shows the saliency scores over input tokens of CoLA and CSQA instances with `zero_b`, `CFP`, and `GKP`. Table 16 shows the average *mean saliency scores* of *input* tokens and *prompt* tokens, calculated using GPT-JT-6B, GPT-J-6B, Flan-T5-770M, and T5-770M. Table 17 shows the average *mean saliency scores* of *input* tokens and ground truth tokens (i.e., tokens in "The answer is []. ") of instances with `zero_b`.

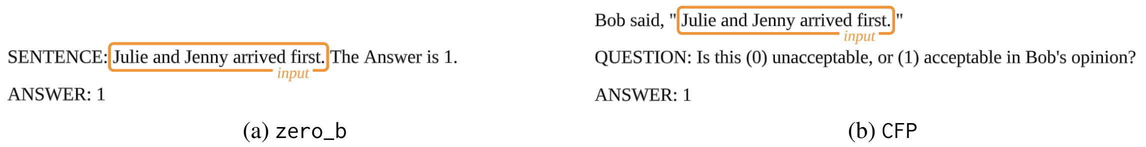


Figure 9: Examples of token segmentation in `zero_b` and `CFP`.

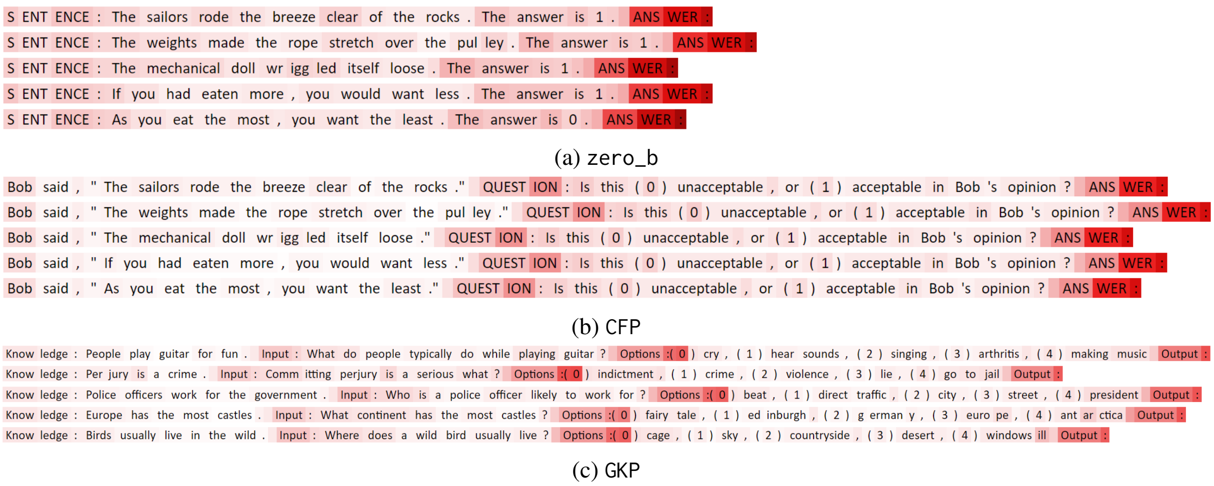


Figure 10: Saliency scores over input tokens of the 5 CoLA and CSQA instances with `zero_b`, `CFP`, and `GKP` using GPT-6B-JT.

dataset	prompt	$\overline{S}_i$	$\overline{S}_p$	$\overline{S}_i/\overline{S}_p$
CoLA	base_b	4.17	12.74	32.73
	zero_b	7.65	23.66	32.33
	CFP	3.75	11.43	32.81
CSQA	base_b	2.68	7.41	36.17
	zero_b	2.25	8.02	28.05
	CFP	2.37	7.06	33.57
MNLI	base_b	3.10	12.79	24.24
	zero_b	3.61	17.18	21.01
	CFP	1.68	6.71	25.04
RTE	base_b	1.95	9.48	20.57
	zero_b	3.22	18.95	16.99
	CFP	1.53	7.32	20.90
SST2	base_b	3.09	12.59	24.54
	zero_b	4.99	22.32	22.36
	CFP	2.77	10.70	25.89
AVERAGE	-	3.25	12.56	26.48

(a) GPT-JT-6B

dataset	prompt	$\overline{S}_i$	$\overline{S}_p$	$\overline{S}_i/\overline{S}_p$
CoLA	base_b	2.93	8.32	35.22
	zero_b	4.08	14.34	28.45
	CFP	2.98	8.35	35.69
CSQA	base_b	1.82	5.01	36.33
	zero_b	1.44	4.41	32.65
	CFP	1.91	4.64	41.16
MNLI	base_b	2.82	9.86	28.60
	zero_b	2.21	10.04	22.01
	CFP	1.33	5.20	25.58
RTE	base_b	1.39	6.63	20.97
	zero_b	2.01	10.94	18.37
	CFP	1.08	5.70	18.95
SST2	base_b	2.31	8.13	28.41
	zero_b	3.46	15.23	22.72
	CFP	2.19	6.64	32.98
AVERAGE	-	2.26	8.23	28.54

(b) GPT-J-6B

dataset	prompt	$\overline{S}_i$	$\overline{S}_p$	$\overline{S}_i/\overline{S}_p$
CoLA	zero_b	7.31	21.83	33.50
CSQA	zero_b	6.03	10.27	58.69
MNLI	zero_b	8.97	26.15	34.29
RTE	zero_b	6.32	24.81	25.45
SST2	zero_b	6.71	24.04	27.91
AVERAGE	-	7.07	21.42	35.97

(c) Flan-T5-770M

dataset	prompt	$\overline{S}_i$	$\overline{S}_p$	$\overline{S}_i/\overline{S}_p$
CoLA	zero_b	9.56	14.52	65.84
CSQA	zero_b	9.58	10.66	89.93
MNLI	zero_b	8.33	11.75	70.89
RTE	zero_b	6.49	12.94	50.17
SST2	zero_b	7.54	10.44	72.23
AVERAGE	-	8.30	12.06	69.81

(d) T5-770M

Table 16: The average *mean saliency scores* of *input* tokens ($\overline{S}_i$), *prompt* tokens ($\overline{S}_p$), and the ratio between them ($\overline{S}_i/\overline{S}_p$).

dataset	$\overline{S}_i$		$\overline{S}_t$		$\overline{S}_i/\overline{S}_t$	
	GPT-JT-6B	Flan-T5-770M	GPT-JT-6B	Flan-T5-770M	GPT-JT-6B	Flan-T5-770M
CoLA	7.65	7.31	12.82	17.12	59.67	42.72
CSQA	2.25	6.03	13.50	11.60	16.67	51.98
MNLI	3.61	8.97	13.29	23.94	27.16	37.46
RTE	3.22	6.32	14.63	22.74	22.01	27.77
SST2	4.99	6.71	12.60	17.30	39.60	38.79
AVERAGE	3.25	7.07	13.37	18.54	24.34	38.12

Table 17: The average *mean saliency scores* of *input* tokens ($\overline{S}_i$), tokens in "The answer is `[]`." ($\overline{S}_t$), and the ratio between them ($\overline{S}_i/\overline{S}_t$) of instances with `zero_b`.

A.7 More on *sensitivity-aware* decoding

Figure 11, 12, and 13 show the performance of GPT-JT-6B, LLaMA2-13B-chat, LLaMA2-7B-chat, Flan-T5-11B, and Flan-T5-770M using `base_a`, `base_b`, and `CFP` with greedy decoding and *sensitivity-aware* decoding. We experimented with different values of α, ranging from 0.1 to 0.9. We did five inferences to estimate sensitivity, so the computational costs are five times higher than those of greedy decoding, which only involves a single inference.

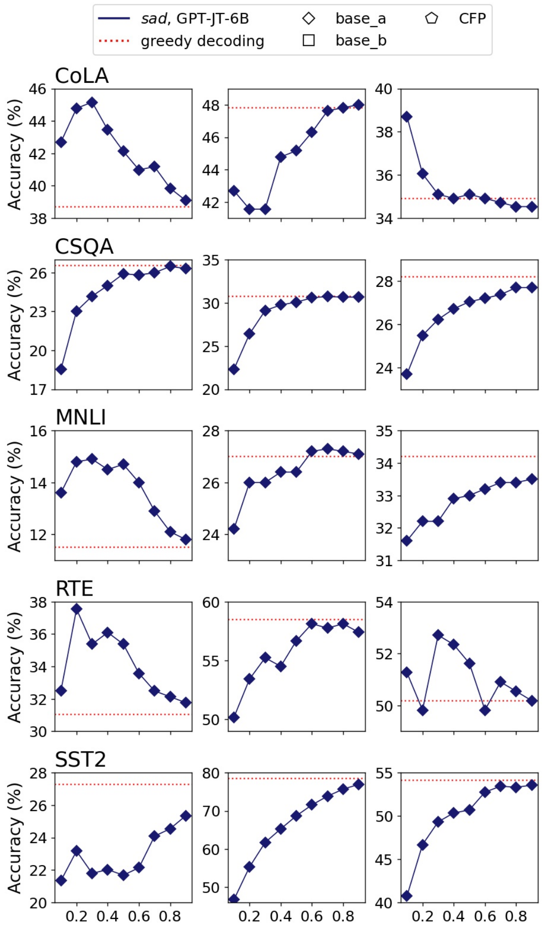


Figure 11: Accuracy (%) of GPT-JT-6B using `base_a`, `base_b`, and `CFP` with greedy decoding and *sensitivity-aware* decoding (`sad`).

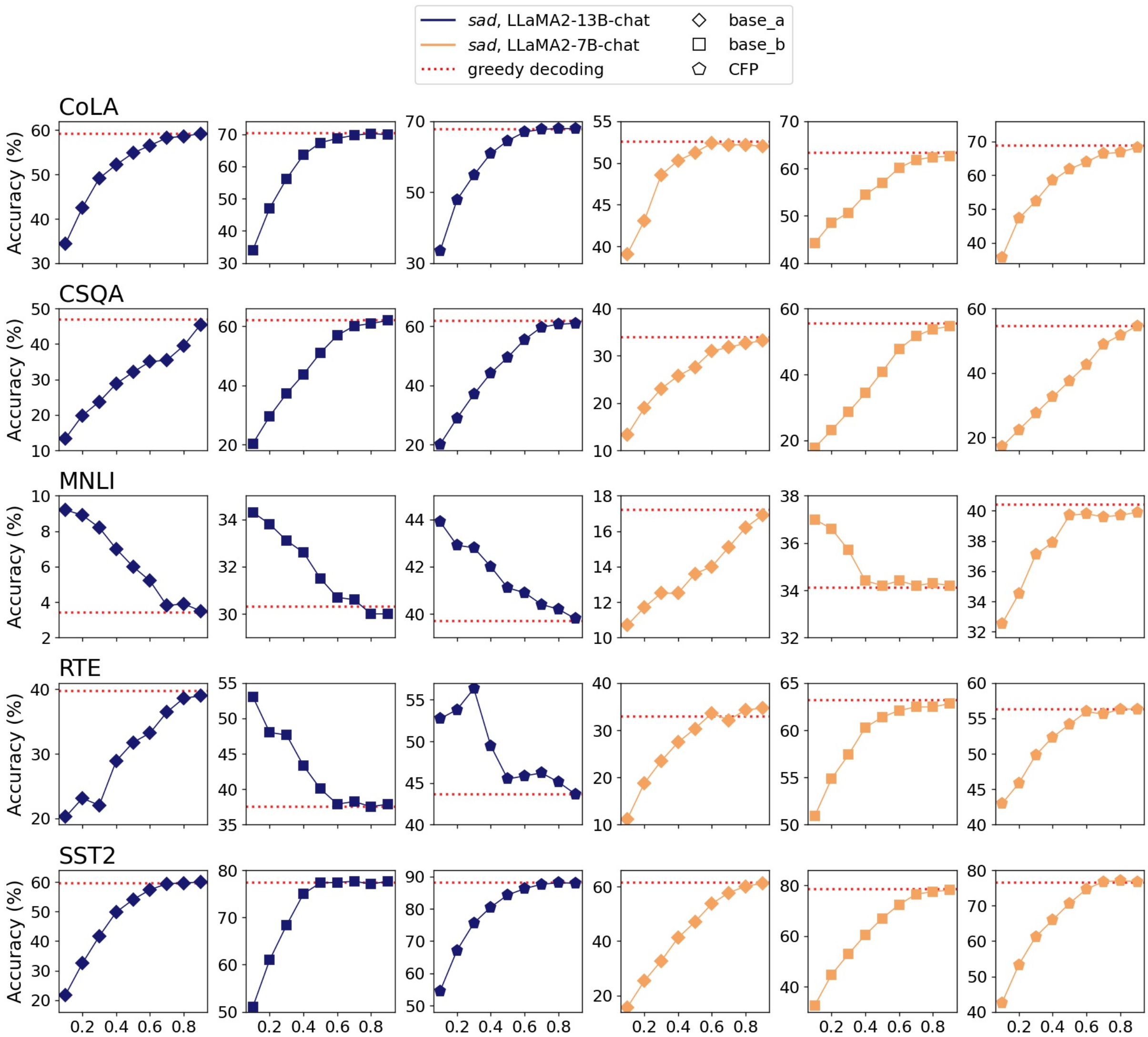


Figure 12: Accuracy (%) of LLaMA2-13B-chat and LLaMA2-7B-chat using base_a, base_b, and CFP with greedy decoding and *sensitivity-aware* decoding (sad).

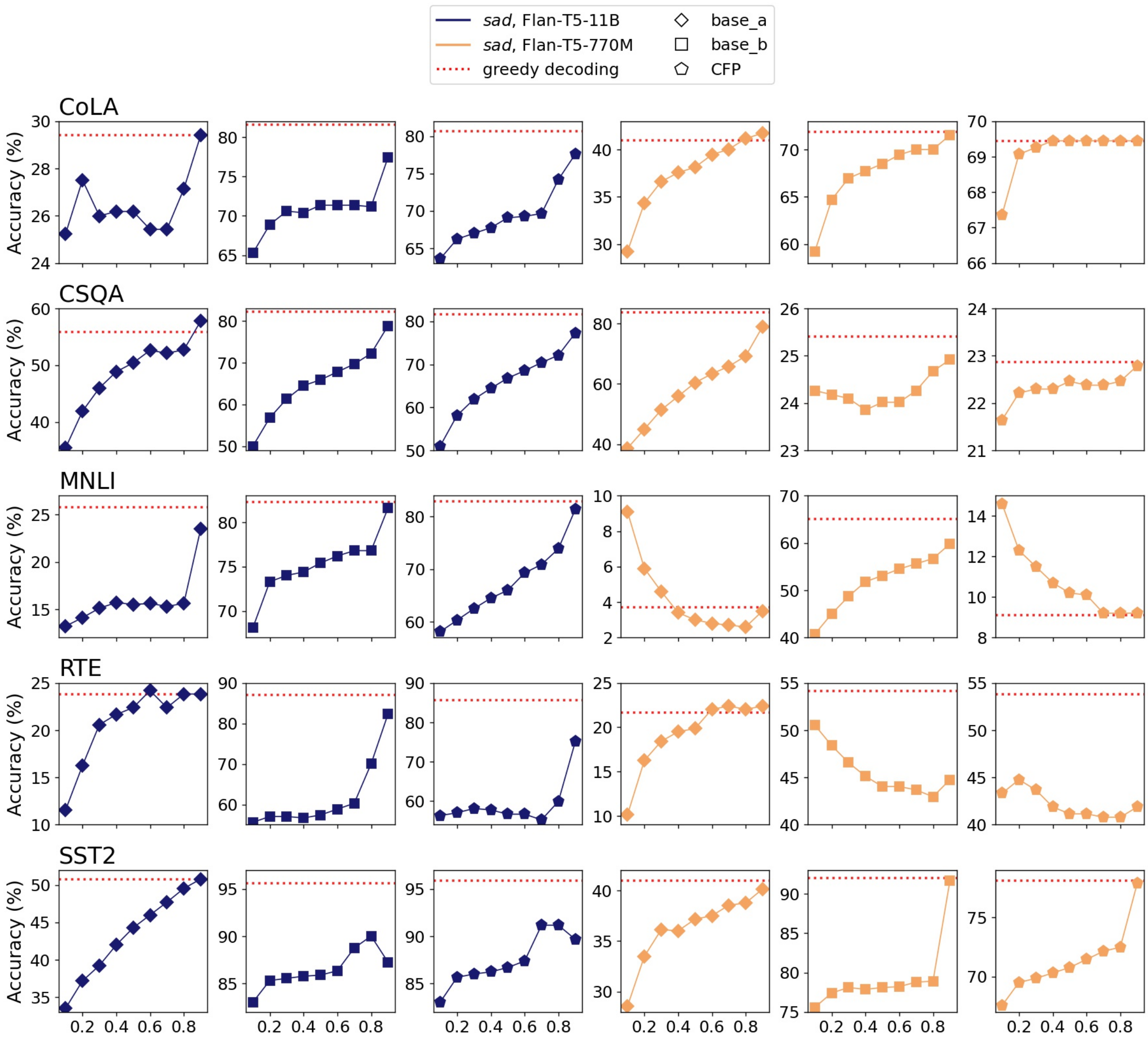


Figure 13: Accuracy (%) of Flan-T5-11B and Flan-T5-770M using base_a, base_b, and CFP with greedy decoding and *sensitivity-aware* decoding (sad).