# "Don't Be So Sensitive!" – Evaluating the Sensitivity of LLMs to Prompt Augmentations

Apoorva Sheera, Manya Chadha, Sarah Nguyen
University of Washington
1400 NE Campus Parkway, Seattle, Washington

## Abstract

*Standard prompting involves structuring tasks for a large language model (LLM) in a manner that can effectively communicate the desired outcome to elicit the intended output from the model. However, more often than not, LLM behaviour can be unpredictable and produce inaccurate results. To address this, various prompting techniques and their effects on LLM responses for different tasks such as mathematics, multi-task reasoning, and natural language inference have been investigated, with the goal to find which technique, or combination of techniques, yields the most accurate and relevant responses. Advancements are made to LLM models daily, and they continue to get bigger, faster and better. There arises a need then, to evaluate the sensitivity of LLM responses to different prompting techniques across successive models and model sizes.*

## 1. Introduction

As large language models (LLMs) such as OpenAI's GPT, Google's Gemini (previously BARD) and Meta's LLaMa continue to advance, there is an emerging need for a skill called "Prompt Engineering". It involves augmenting prompts to generate desired outputs has emerged as a gradient-free performance improvement technique for LLMs. While LLMs are powerful models capable of generating human-like text across a wide range of tasks, their behavior can sometimes be unpredictable and may produce results that are inaccurate, biased, or irrelevant without proper guidance. Effective prompts can help LLMs focus on relevant information and generate desired responses. This is where a prompt engineer steps in and fine-tunes the prompts in a manner that urges the model to respond more accurately. However, as LLMs get bigger and better, the question remains - Are they becoming more or less sensitive to how a prompt is constructed?

OpenAI and Google have both released documentation [1] [5] that provide guidance on what makes a good prompt. Several other resources have also attempted to consolidate "best practices" [11] for building prompts based on the how different LLMs are trained and fine-tuned. For instance, Min et al. [10] showcases how in-context learning works in LLMs and improves their responses. Similarly, chain-of-thought prompting conducted by Wei et al. [17], show an exemplary improvement in the performance of LLMs on a range of arithmetic, commonsense and symbolic tasks. Yang et al. [20] even go as far as to suggest that you can use LLMs to optimize and find instructions that maximise task accuracy. All of these research findings lead us to believe that the quality of LLM outputs are somewhat correlated to the way a prompt is constructed and passed on.

Our project aims to analyze the effect of various prompt augmentations on the outputs of Large Language Models over time and scale. We will build on current literature exploring the effects of adversarial prompts [23] and modify it by replacing "attacks" with augmented prompts, using techniques considered "best practices" within the domain of prompt engineering. These include Zero Shot Prompting [16], Few Shot Prompting [2], Chain of Thought Prompting, Self Consistency [14], General Knowledge Prompting [9], etc. Our ultimate objective is to determine if larger and recently fine-tuned models are less sensitive to "prompt hacking".

Zhu et al. [23] introduced 'PromptBench', a systematic bench-marking methodology designed to evaluate the robustness of LLMs to adversarial prompts. These adversarial prompts were created to generate a perturbation such that an LLM would produce an incorrect response. They investigated four different types of prompts – task-oriented, role-oriented, zero-shot and few-shot generation techniques. They then created adversarial "attack" prompts at the character, word, sentence and semantic level across different NLP tasks such as sentiment analysis, grammar correctness, natural language inference, multi task knowledge,translation, etc., until the LLM produced an incorrect response. We utilize the PromptBench framework to evaluate prompts from 4 different NLP task datasets on 3 LLMs using 5 different augmentation techniques. Our ex-

periments modify the "attacking" layer from [23] and add a pipeline that creates different inputs for each prompt using augmentation techniques. We also build a pipeline to extract from the natural language output of the LLM, only the correct response, which we use to calculate the accuracy with respect to ground truth labels.

A challenge we anticipated running into involved measuring the similarity between two responses and quantifying whether the differences are significant. Additionally, the same notion of checking if our generated prompts retain their semantic meaning throughout also poses a concern. In order to manage the scope of this project, we decided to analyse the sensitivity only across tasks that can be evaluated using classification accuracy. However, certain metrics such as ROUGE [8] and BERT scores [21] can potentially be utilized across summarization and translation tasks. Another technical challenge we accounted for was the systematic generation of prompts based on our chosen methodologies, especially in the absence of a pre-existing framework on PromptBench. For such cases, we build standardized templates (see Appendix) and utilize them to generate prompts.

Through our experiments, we aim to understand: 1. How has the sensitivity of LLMs to prompt augmentations changed over time? 2. Does model size influence the sensitivity of an LLM to prompt construction? These findings will help set a comprehensive benchmark on LLM sensitivity and guide the path for future research in reducing LLM sensitivity to prompt augmentations.

## 2. Related Work

In the October 2023 release of PromptBench, when evaluating GPT4, T5-Large, Llama2-13b-chat, ChatGPT, Vicuna-13b-v1.3, and UL2, Zhu et al. [23] found that all models exhibit vulnerability to adversarial prompts. ChatGPT and GPT-4 were most robust in their results and GPT-4 outperformed the other models. An accuracy comparison of results from different prompt engineering techniques was made between GPT-3.5-Turbo and GPT-4, and while the overall accuracy improved between model updates, there was not a consensus in one technique producing higher accuracy than another across all datasets as each of the prompt techniques are effective for different tasks.

Following this idea of version comparison, Chen et al. [3] investigated whether an LLM is continuously improving over time by comparing the responses of GPT-3.5 and GPT-4 in various tasks. They focused on the following tasks - solving math problems, answering sensitive questions, answering OpinionQA survey, LangChain HotpotQA Agent, code generation, taking USMLE medical exam, and visual reasoning. They found that the performance and behavior of GPTs vary in both, a negative and positive way between updates. In the updates for GPT-4, it appears that GPT-4 can no longer follow chain-of-thought prompting as well as it did in GPT-3.5, causing accuracy to drop within GPT-4 for certain tasks such as identifying prime vs composite numbers. They observed that GPT-4 had a lower response rate to sensitive questions and opinion surveys, and found that GPT-4 ability to follow user instruction has decreased over time. GPT-4 did however, improve accuracy in tasks such as multi-hop questions, which involve multiple source/reasoning steps. This lack of consensus in overall improvements between version updates led them to conclude that improving the model's performance on some tasks can have unexpected consequences on its behavior in other tasks.

### 2.1. Prompt Augmentation Techniques

In terms of prompt engineering, **zero-shot** is the simplest type of prompt and requires providing only instructions to the model for any task. It will be used as a baseline to compare other responses in our experiments. While zero-shot prompts yield successful results most of the time, models struggle with more complex tasks in zero-shot settings.

**Zero-Shot CoT** [6] is the ability of an LLM to perform tasks that require multi-step reasoning on unseen domains without receiving training examples. It enables them to generalize knowledge from their training data and apply it on unseen situations. In juxtaposition to zero-shot CoT prompting, **few-shot prompting** [2] can help give the model guidance by providing examples of the responses we would like to see produced from the model.

Additionally, we want to evaluate how the addition of **role oriented , task oriented**, **chain-of-thought**, **least to most**, and **emotion** as prompt augmentations affect LLM response.

[23] bucketed the prompts used to attack the LLMs into 2 categories - task-oriented and role-oriented. They then designed zero-shot and few-shot learning scenarios using these two prompting methods. **Task-oriented prompting** involves explicitly describing the task we want the LLM to perform and expecting it to respond solely using it's pretraining knowledge. **Role-oriented prompting** involves framing the model as an entity with a specific role, usually as some sort of expert related to the task we feed it. By asking it to adopt a role, the prompt implicitly conveys to the model what it should output. Examples include asking the LLM to "adopt the persona of an IT support engineer/french translator/etc."

Wei et al. [17] explores how generating a series of intermediate reasoning steps, or chain of thought, improves the ability of LLMs to handle complex reasoning. Chain of thought prompting involves showing the model a few examples where the reasoning is laid out step by step, causing models to give more detailed responses. This type of prompting can provide insight on why an LLM arrived at

a particular answer. Zhou et al., [22] investigated CoT prompting further and found that while it achieved remarkable results on reasoning tasks, it did not do too well on more complex tasks and suggested another methodology for prompt construction called **least-to-most** prompting. The underlying idea is to break down a complex problems into a series of smaller problems that are relatively easier to solve, and then solving each smaller "sub-problem" sequentially, using the answers to the previously solved sub-problem. This is done using the few-shot scenario.

Another interesting prompt augmentation technique we want to investigate is EmotionPrompting. Li et al [7] found that on 45 tasks with 6 different LLMs such as Vicuna, Llama 2, ChatGPT, GPT-4, BLOOM, Flan-T5-Large, LLMs enhanced by emotional intelligence can achieve better "performance, truthfulness, and responsibility". The last technique we reviewed is ExpertPrompting [19] which involves LLMs to take on the role of a distinguished expert in the desired domain. Using in-context learning to create customized descriptions of expert identities, the LLM would provide an answer conditioned on this expert background. Based on this augmentation in prompt, they found that their model, ExpertLLaMA outperforms Vicuna, LLaMA-GPT4, and Alpaca, falling short only to ChatGPT. We do not specifically evoke this technique, and instead use results from role-oriented prompt augmentations as a proxy.

### 2.2. Tasks and Affiliated Datasets

We want to explore the different types of augmentations with a variation of tasks consisting of math, multiple choice problems and natural language inference (NLI) to evaluate the affects of perturbations on prompts.

To evaluate math word problems, we utilize **gsm8k** [4] introduced by Cobbe et al. Gsm8k contains linguistically diverse grade school math word problems authored by humans and can be used for multi-step mathematical reasoning. Each problem takes between 2 and 8 steps to solve, and solutions involve performing a sequence of elementary calculations such as addition, subtraction, multiplication, and division to reach the final answer. The General Language Understanding Evaluation (GLUE) benchmark [13] was created with the intention of developing a multi-task benchmark to evaluate the performance of models across NLI tasks. They evaluated baselines based on current methods for transfer and representation learning and found that multi-task training on all tasks performs better than training a separate model per task. The benchmark was created by focusing on nine English sentence understanding tasks, and from the benchmark, we've chosen to perform sentiment analysis, grammar correctness, and natural language inference. The **Stanford Sentiment Treebank (SST-2)** [12] dataset contains single sentence tasks, with sentences from movie reviews with the task being to predict the sentiment of a given sentence using a positive and negative class split, with human annotations of their sentiment as ground truth. The **Corpus of Linguistic Acceptability (CoLA)** [15] consists of judgements drawn from books and journal articles on English linguistic theory to measure grammatical correctness. The **Multi-Genre Natural Language Inference Corpus (MNLI)** [18] allows us to perform inference given context and a hypothesis, with the goal of predicting whether the contexts entails, contradicts, or is neutral to the hypothesis.

## 3. Methodology

### 3.1. Experimental Setup

Our methodology builds upon PromptBench [23]. We utilize their framework and create an additional 'Prompt Construction' pipeline for each task that our set of LLMs are asked to do. We do this by identifying certain prompt engineering methodologies that were prevalent across Prompt Engineering guides and literature [1, 5, 11] and creating templates (see Appendix) that can be replicated across different task datasets to generate augmented prompts.

We limit the scope of our analysis of LLMs to include Flan-T5-large, Vicuna-7B, Phi-1.5 and Phi-2, keeping in mind our goal to evaluate the trend of LLM sensitivity to prompt construction over time, version updates and scale. For the scope of this project, we chose NLI tasks such that the outputs corresponded to a certain class and basic classification accuracy metrics to evaluate the performance of an LLM to a certain prompt augmentation could be used. We sampled from datasets like gsm8k and GLUE to analyze the performance of an LLM on a variety of tasks - summed up in Figure 1. For each prompt for a given task, we construct multiple alternate versions incorporating prompt engineering methodologies as summarized in Figure 1.

### 3.2. Evaluation

We measure the performance of the LLM on constructed prompts using a modified version of the metric from [23] - Performance Increase Rate. The PIR is given by:

$$PIR(C, P, f_\theta, \mathcal{D}) = \frac{\sum_{(x,y)\in\mathcal{D}} \mathcal{M}[f_\theta([C(P), x]), y]}{\sum_{(x,y)\in\mathcal{D}} \mathcal{M}[f_\theta([P, x]), y]} - 1$$

Where $C$ is the prompt construction, and $\mathcal{M}[.]$ is the evaluation function which is defined as classification accuracy for our subset of tasks.

Apart from this, we also measure the deviation of outputs from baseline outputs through another metric - Response Similarity defined as the change from baseline prompts, this metric is an average instead of a rate. RS is given by:

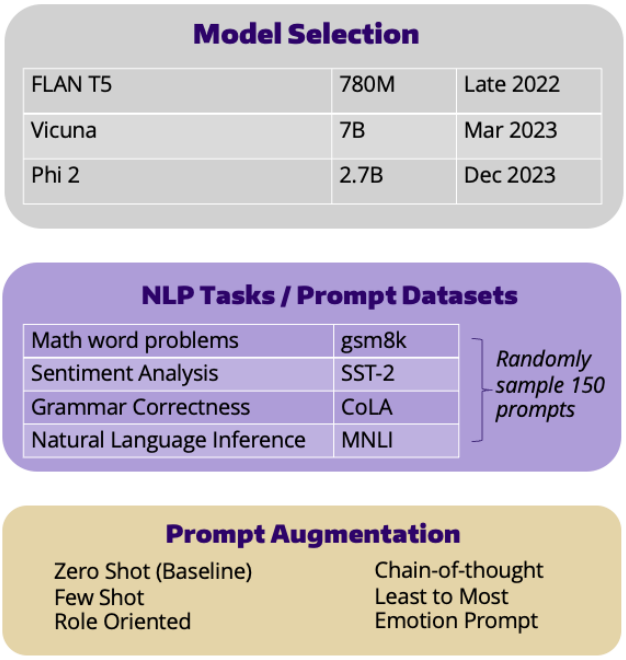


Figure 1. Experimental Setup

$$RS(C, P, f_\theta, \mathcal{D}) = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} \mathcal{M}[f_\theta([C(P), x]), f_\theta([P, x])]$$

## 4. Experiments

Our experiments are structured so that we are able to evaluate the sensitivity of models of different sizes, and released at different points in time (Figure 1) with the same prompt augmentations. We then created augmentation templates specific to each dataset and applied them to 150 randomly sampled prompts from the respective dataset. We also introduced randomness while generating the augmented versions of the prompt by sampling from different types of emotions, roles, etc.

After sampling the prompts and creating its augmented counterparts, we run each selected LLM and pass its natural language response through the 'extraction' pipeline. This pipeline draws out the response from each generated string in a pre-specified format for each dataset (see Table 1. in Appendix) so that we are able to calculate accuracy with respect to ground truth labels.

An 'evaluation' pipeline is then introduced to pick up the extracted responses and compute the Accuracy, Performance Increase Rate and Response Similarity for each prompt augmentation technique across datasets per model.


Figure 2. Classification accuracy heatmap for FLAN-T5-Large

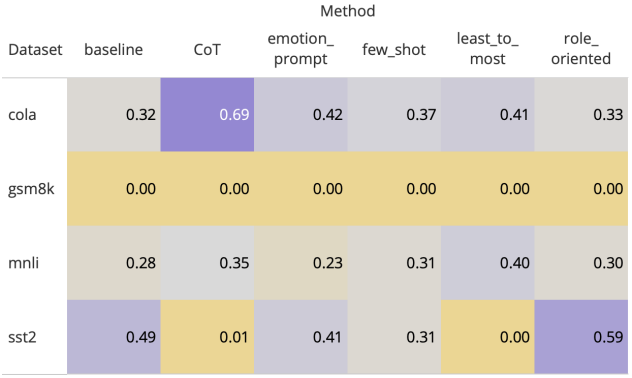


Figure 3. Classification accuracy heatmap for Vicuna-7B

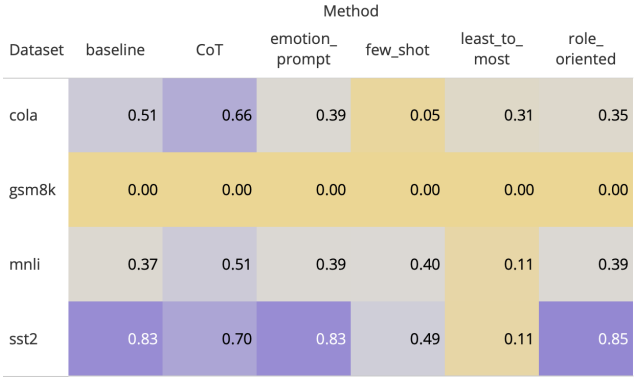


Figure 4. Classification accuracy heatmap for Phi-2

## 5. Results and Discussion

Across the board, FLAN-T5 performs remarkably well compared to the rest, despite having the least number of parameters. Notably, results display extremely poor performance across all models and prompting techniques for the gsm8k dataset, i.e., on math word problems.

We note that the CoLA dataset is the only one wherein a prompt augmentation invokes an increase in performance for all 3 LLMs. While FLAN-T5 sees a rise in accuracy

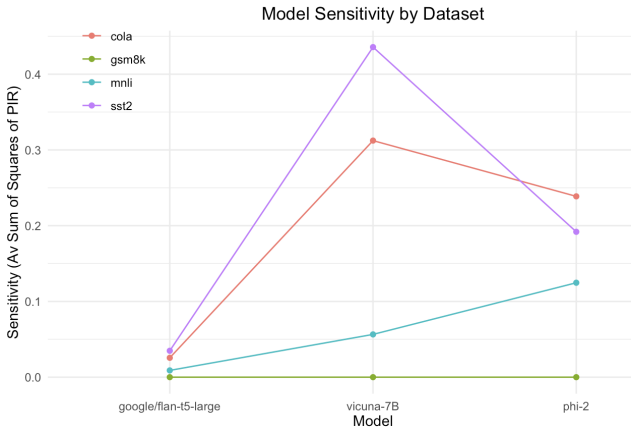


Figure 5. Plotting sensitivity of model output to prompt augmentations across tasks

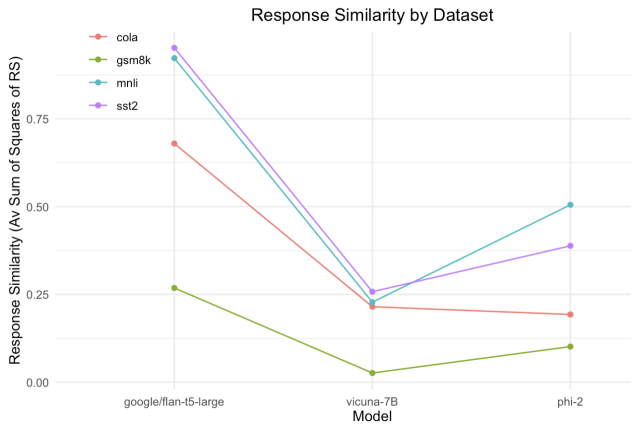


Figure 6. Plotting similarity of model output to baseline output with prompt augmentations across tasks

due to few-shot augmentation, both Vicuna-7B and Phi-2 see an increased accuracy through the chain-of-though augmentation. In fact, chain-of-thought emerges as the most successful prompt augmentation in terms of accuracy across all LLMs.

Notably, the SST2 dataset responds positively to role-oriented prompts in two of the selected LLMs. Additionally, the least-to-most prompt augmentation accounts for the worst performance of all LLMs for the sentiment analysis task.

For the MNLI dataset, chain-of-thought prompting emerges as the augmentation which boosts the performance of both Vicuna-7B and Phi-2 substantially. However, for FLAN, baseline accuracy remains the highest and decreases when augmented using chain-of-thought prompting.

On comparing the minimum and maximum accuracies to the baseline across prompting methodologies for each dataset, we find that the CoLA dataset is the most sensitive to prompt augmentations. It is closely followed by the SST2 dataset and MNLI dataset.

While our expectation was to see a steady decline in sensitivity as models move forward in time and increase in size, we notice different trends across tasks. The largest model in our set - Vicuna-7B is the most sensitive to prompt augmentation for grammar correctness and sentiment analysis tasks. The most recent model in our dataset - Phi-2 is the most sensitive to prompt augmentation for natural language inference. Similarly, we notice diverging trends for Response similarity by tasks and models.

An extension of our experimentation could include bigger models like GPT3.5, LLaMA-13B, etc., larger breadth of tasks including summarization, translation etc. and robust and additional performance metrics such as Matthew's Correlation Coefficients, F-1 scores, etc.

## References


[1] Open AI. Prompt engineering. 1, 3

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 1, 2

[3] Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time?, 2023. 2

[4] 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. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. 3

[5] Google. Prompt engineering for generative ai. 1, 3

[6] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. 2

[7] Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. Large language models understand and can be enhanced by emotional stimuli, 2023. 3

[8] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain, July 2004. 2

[9] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning, 2022. 1

[10] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. 1

[11] Elvis Saravia. Prompt engineering guide. 1, 3

[12] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. 3

[13] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461, 2018. 3

[14] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. 1

[15] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *CoRR*, abs/1805.12471, 2018. 3

[16] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. 1

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. 1, 2

[18] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426, 2017. 3

[19] Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. Expertprompting: Instructing large language models to be distinguished experts, 2023. 3

[20] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2023. 1

[21] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. 2

[22] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023. 3

[23] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, and Xing Xie. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts, 2023. 1, 2, 3

# 6. Appendix: Prompt Augmentation Templates

To augment each prompt, we have appended the following templates to the beginning of each input. Below are a few examples for some of the datasets.

## 6.1. Few Shot

**Math Word Problems (gsmk8)**

”Here are three examples.”
Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: 6.
Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: 5.
Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: 39.

**Sentiment Analysis (SST-2)**

”Sentence: hide new secretions from the parental units. Answer: negative. ” ”Sentence: contains no wit , only labored gags. Answer: negative. ” ”Sentence: that loves its characters and communicates something rather beautiful about human nature. Answer: positive. ”

**Grammar Accuracy (CoLA)**

”Sentence: Our friends won’t buy this analysis, let alone the next one we propose. Answer: acceptable. ” + ”Sentence: One more pseudo generalization and I’m giving up. Answer: acceptable. ” + ”Sentence: They drank the pub. Answer: unacceptable. ”

**Multi-Genre Natural Language Inference (MNLI)**

“Premise: Conceptually cream skimming has two basic dimensions - product and geography. Hypothesis: Product and geography are what make cream skimming work. Answer: neutral. ” + “Premise: you know during the season and i guess at at your level uh you lose them to the next level if if they decide to recall the the parent team the Braves decide to call to recall a guy from triple A then a double A guy goes up to replace him and a single A guy goes up to replace him. Hypothesis: You lose the things to the following level if the people recall. Answer: entailment. ” + “Premise: Fun for adults and children. Hypothesis: Fun for only children. Answer: contradiction. ”

## 6.2. Role oriented

**Math Word Problems (gsmk8)**

Randomly choose one of the below:
“In your role as a math problem-solving assistant, ”,
“Functioning as a math evaluation tool, ”,

**Sentiment Analysis (SST-2)**

“As a sentiment classifier, determine whether the following text is ’positive’ or ’negative’:”

"As an emotion detector, determine if the provided passage conveys a 'positive' or 'negative' sentiment: "
**Grammar Accuracy (CoLA)**
"As a grammar identification system, examine the provided sentence and respond with 'acceptable' for grammatically correct sentences or 'unacceptable' for incorrect ones:",
"Functioning as a grammar evaluation tool, analyze the given sentence and decide if it is grammatically correct, responding with 'acceptable' or 'unacceptable':"
**Multi-Genre Natural Language Inference (MNLI)**
"Acting as an entailment detection instrument, determine if the given pair of sentences demonstrates entailment, neutral, or contradiction. Answer with 'entailment', 'neutral', or 'contradiction':"
"As a tool for determining entailment relationships, review the two statements and categorize their connection as either 'entailment', 'neutral', or 'contradiction':",

### 6.3. Chain-of-Thought

**Math Word Problems (gsmk8)**
Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

### 6.4. Least-to-Most

**Math Word Problems (gsmk8)**
Four years ago, Kody was only half as old as Mohamed. If Mohamed is currently twice 30 years old, how old is Kody?
Q: How old was Mohamed four years ago?
A: We were told that Mohamed is currently twice 30 years old, so he is currently 30 * 2 = 60 years old. That means that four years ago he must have been 60 - 4 = 56 years old. The answer is 56.
Q: How old is Kody?
A: Four years ago, Kody was half as old as Mohamed, so Kody must have been 56 / 2 = 28 years old then. Since Kody was 28 years old four years ago, she must now be 28 + 4 = 32 years old. The answer is 32.

### 6.5. Emotion Prompting

Emotion prompting was not changed per dataset. Instead, we randomly chose one of the below:
"This is very important to my career.",
"You'd better be sure.",
"Are you sure?",
"Are you sure that's your final answer? It might be worth taking another look.",
"Provide your answer and a confidence score between 0-1 for your prediction. Additionally, briefly explain the main reasons supporting your classification decision to help me understand your thought process. This task is vital to my career, and I greatly value your thorough analysis.",
"Are you sure that's your final answer? Believe in your abilities and strive for excellence. Your hard work will yield remarkable results.",
"Embrace challenges as opportunities for growth. Each obstacle you overcome brings you closer to success.",
"Stay focused and dedicated to your goals. Your consistent efforts will lead to outstanding achievements.",
"Take pride in your work and give it your best. Your commitment to excellence sets you apart.",
"Remember that progress is made one step at a time. Stay determined and keep moving forward."

| Dataset | Output Label |
|---|---|
| gsmk8 | Numeric answer (Ex. 12) |
| SST-2 | Positive, Negative |
| CoLA | Linguistically acceptable, unacceptable |
| MNLI | Entailment, Contradiction, Neutral |

Table 1. Labels from Dataset