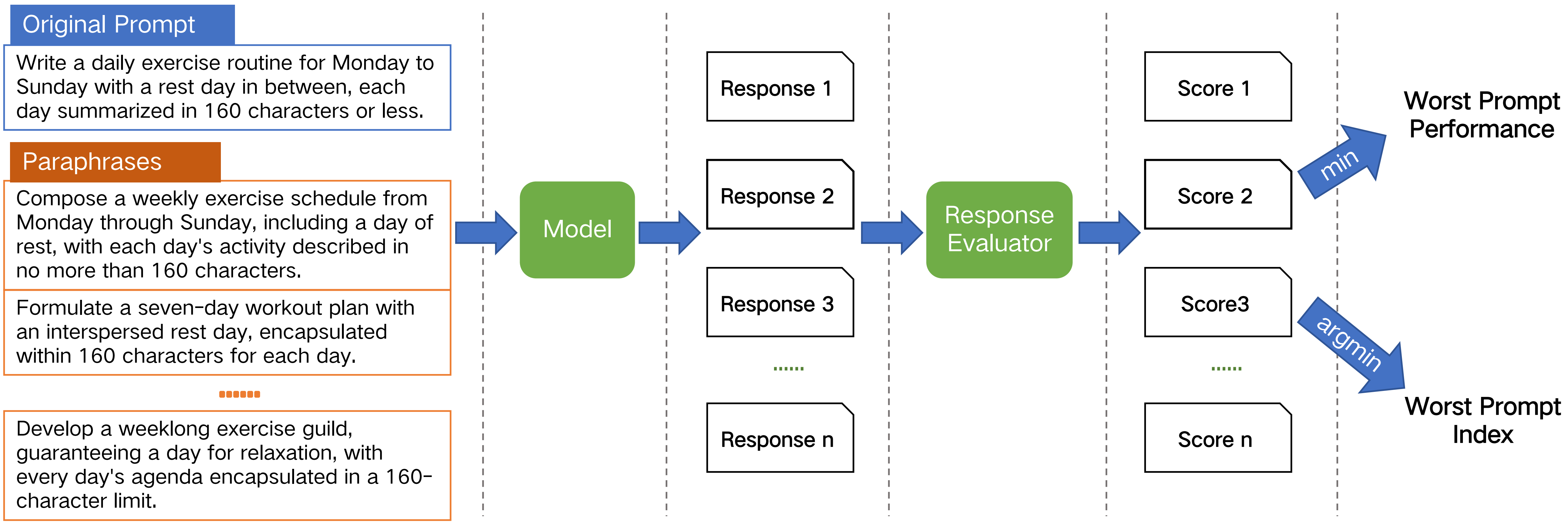
On the Worst Prompt Performance of Large Language Models

Bowen Cao♢ Deng Cai♡ Zhisong Zhang♡ Yuexian Zou♠ Wai Lam♢
♢ The Chinese University of Hong Kong
♡ Tencent AI Lab ♠ Peking University

Worst Prompt Performance

- What is it?
  - the lowest performance a model exhibits across different paraphrases of a prompt with equal semantics and fluency.



Worst Prompt Performance

- Why is it important?
  - LLMs are sensitive to the prompts they receive. → Prompt engineering is usually required.
  - However, prompt engineering is not always feasible. → Improving the "lower bound".

Original Instruction (Score: 0.9922)
On the basis of the subject of the email, determine whether the email should be treated as spam or not.
Dermatologists don't like her!
Output:
Based on the subject of the email, it is likely that the email should be treated as spam. The use of the phrase "Dermatologists don't like her" is likely to be a sensationalized or misleading claim ...

Paraphrase (Score: 0.0005)
Examine the email topic to ascertain if it should be categorized as spam. Subject: Dermatologists don't like her!
Output:
This email topic does not appear to be spam. Here's why: 1. Relevance: ... It is not a generic or random topic that is often associated with spam emails. 2. Urgency: ... 3. Personalization: The email is addressed to a specific person, which suggests that it may be a legitimate message rather than a mass email blast. 4. Grammar and spelling: The email appears to be well-written and free of obvious grammatical and spelling errors, which is another indicator that it may be a legitimate message.
Based on these factors, it is unlikely that this email should be categorized as spam.

Despite providing instructions (other paraphrases are omitted for brevity) with the same semantics, the output from Llama2-7b-chat is completely contradictory. There are numerous inaccurate expressions (highlighted in red) in the underperforming response.

Novelty

- Difference to prior works on prompt robustness
  - Previous research mainly focuses on task-level instructions.
  - Our work: real-world (case-level) user queries.

	Existing Work	Real-world User Queries
Task-level Instruction	Assume you are a customer service representative. Please provide customer service to a customer regarding their difficulties in accessing a website.	Case1: As a customer service representative, kindly assist a customer who is facing challenges with our website, specifically reporting an inability to sign in over the last four days.
Case-level Inputs	**Case1:** The customer states that they have been unable to sign in for the past four days. **Case2:** A customer reports that they are unable to complete an online purchase due to the website's checkout process consistently failing. ……	Case2: Craft a narrative that brings to life a character sketched as a 5-year-old boy with an insatiable curiosity and a penchant for questioning, while also harboring a strong aversion to adhering to rules. Feel free to elaborate on his traits and weave in incidents that highlight his personality.

An example illustrating the gap between existing benchmarks that evaluate prompt consistency and real user queries.

Benchmark Introduction

We present RobustAlpacaEval, a novel benchmark designed to evaluate large language models on semantically equivalent queries across real-world tasks.

- Data

  We develop RobustAlpacaEval by creating ten paraphrases for each query within TinyAlpacaEval. This is first accomplished automatically through GPT4. Subsequently, each paraphrase is manually reviewed and revised to ensure semantic integrity and human-like fluency.
- Metric
  - We use weighted win-rate as our performance metric, where we employ the gpt4_turbo model as the evaluator and the reference model.
  - We term the model’s performance on the original prompt as original performance. We also report the worst, best, average performances across all paraphrases as well as the standard deviation. We average these results across all cases in RobustAlpacaEval.

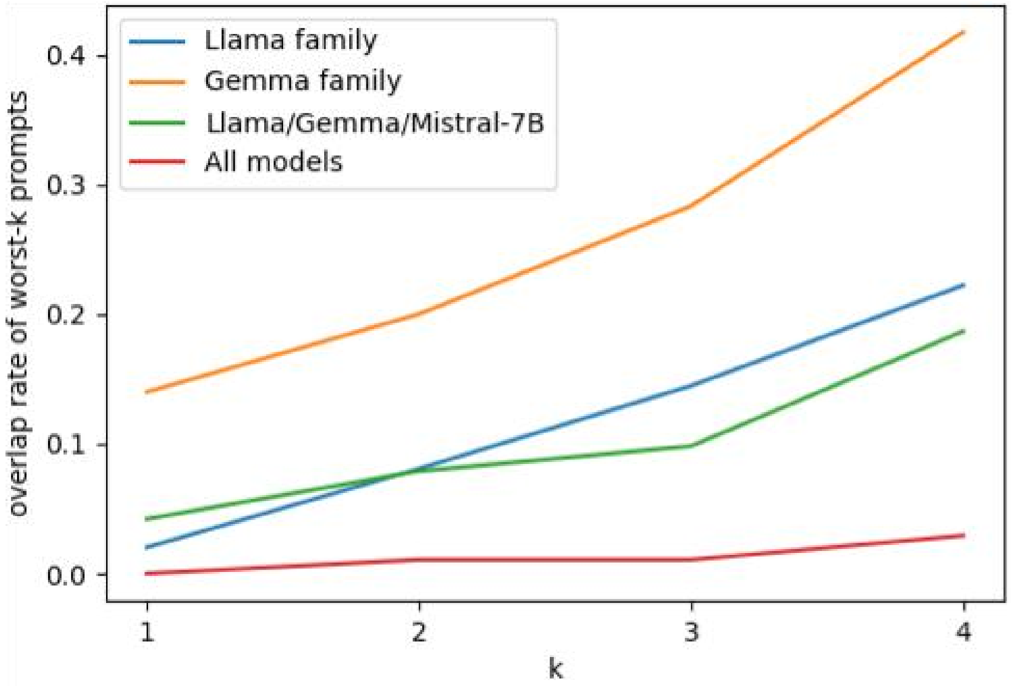
Exploring Performance Dynamics

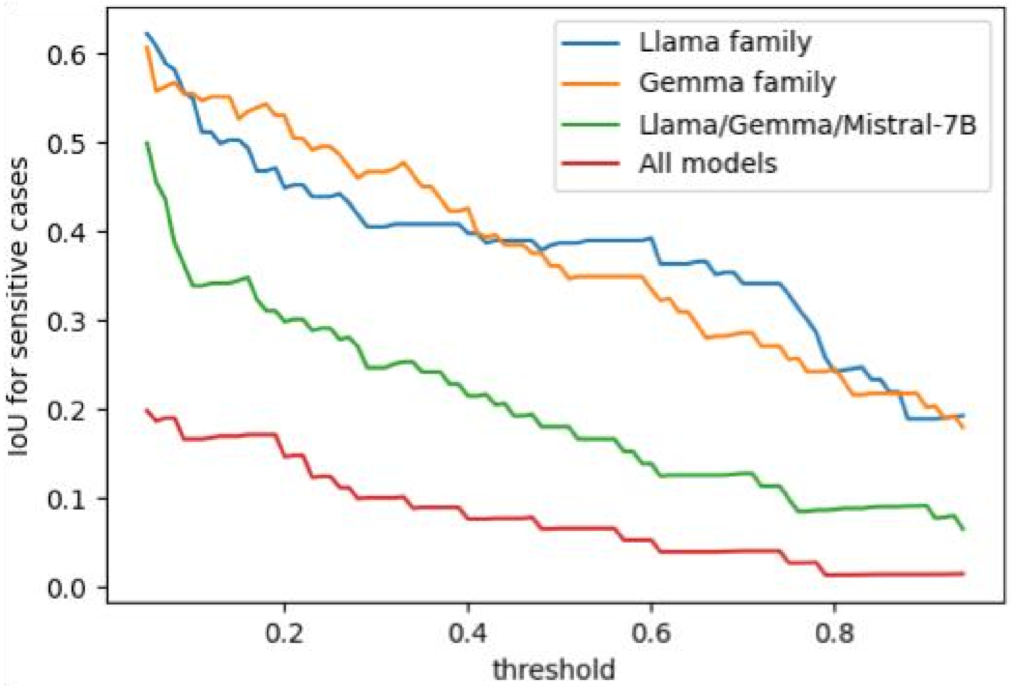
Model	Orig. Perf. ↑	Worst Perf. ↑	Best Perf. ↑	Avg. Perf. ↑	Standard Dev. ↓
Gemma-1.1-2b-it	16.32	4.42	36.60	15.27	**11.78**
ChatGPT	17.46	5.44	39.88	19.96	12.86
Mistral-7b-instruct	24.56	4.22	45.26	21.82	14.60
Llama-2-7b-chat	25.61	5.42	43.54	19.52	13.32
Llama-2-13b-chat	27.48	4.83	52.05	23.97	16.25
Gemma-1.1-7b-it	29.57	8.73	**62.38**	**31.04**	19.07
Llama-2-70b-chat	**32.23**	**9.38**	54.86	29.18	15.61

- Key Findings
  - **Performance Variability:** Evidenced by considerable standard deviation in model performance across different paraphrases, this variability persists regardless of model scaling.
  - **Performance Gap:** Significant disparity exists between the worst (lower bound) and best (upper bound) performances across all models.
  - **Assessment Limitations:** Conventional assessments (Orig. Perf.) only provide a narrow perspective on a model's comprehensive capabilities.

Unveiling the Difficulties in Identifying the Worst Prompt

Model-Agnostic Analysis





- (Left) The overlap rates across all models are nearly zero, which highlights significant variability in their performance. Even within model families, the overlap is only slightly higher when k=1 and k=2, suggesting that **the worst prompts are often specific to each model**.
- (Right) We classify a case as a sensitive case iff the model's performance range exceeds a threshold. The IoU drops below 0.2 across all models, **indicating a scarcity of model-agnostic traits**.

Unveiling the Difficulties in Identifying the Worst Prompt

Model-Agnostic Analysis

We utilize Kendall’s to measure whether the rankings among different models are consistent, and report the average value of Kendall’s W across all cases. We also calculate the proportion of cases with different levels of consistency.

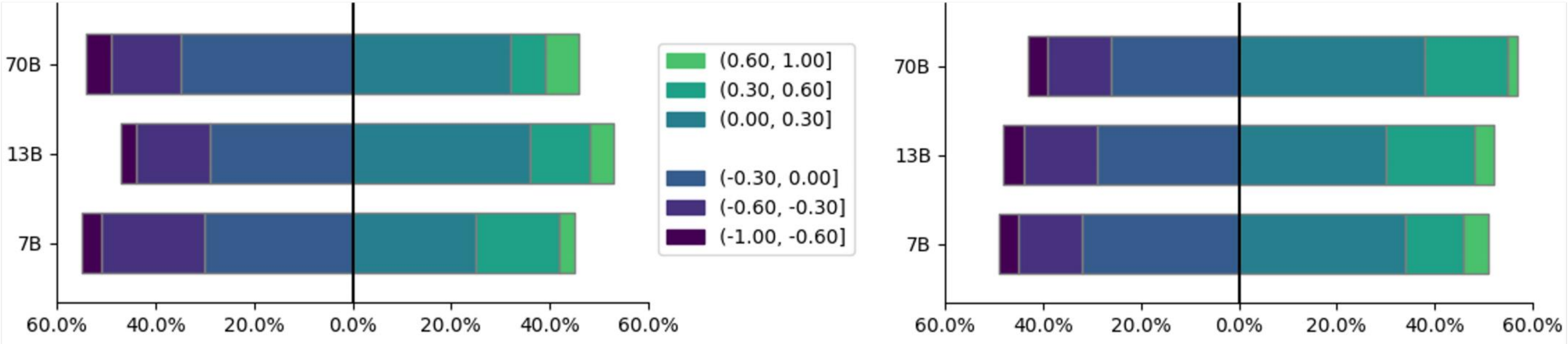
Model	Kendall’s W	Agreement Levels Distribution			
		Negligible	Weak	Moderate	Strong
Llama family	0.443	0	0.242	0.414	0.343
Gemma family	0.548	0	0.08	0.28	0.64
Llama/Gemma/Mistral-7B	0.401	0.011	0.326	0.411	0.253
All models	0.238	0.053	0.723	0.202	0.021

- The low consistency across all models underscores the difficulty in defining a model-agnostic standard for "good" and "bad" prompts.

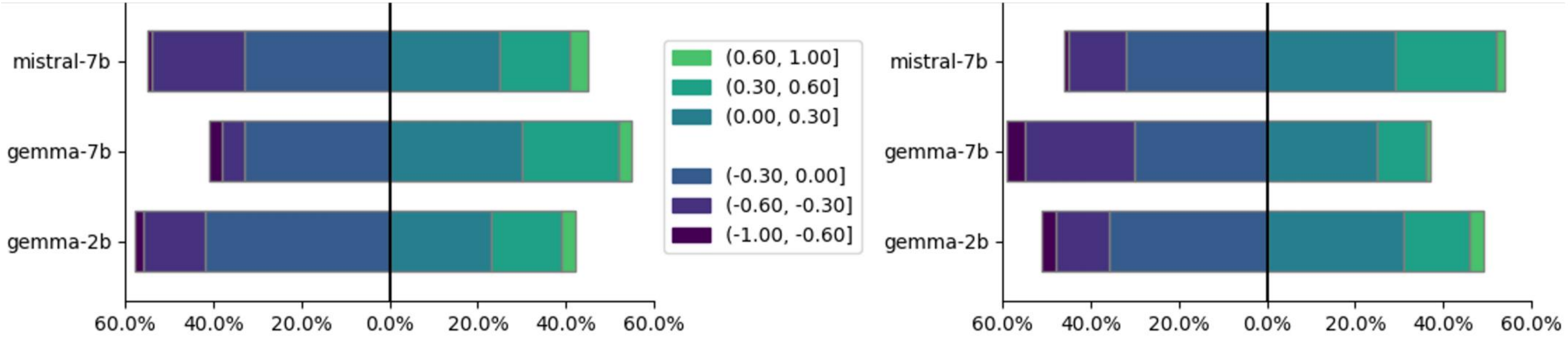
Take-away: These experiments demonstrate that it is unlikely to characterize the worst prompts using model-independent features.

Unveiling the Difficulties in Identifying the Worst Prompt

Model-dependent Analysis



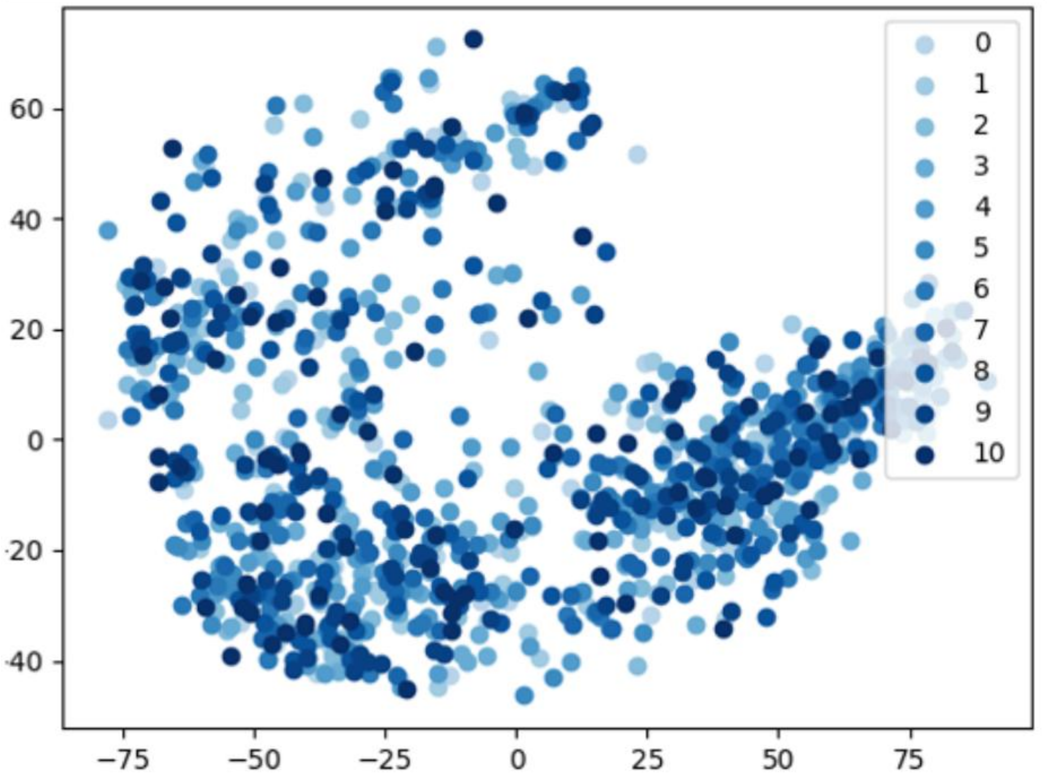
Distribution of Pearson correlation coefficients between model performance and prompt perplexity (left) and prompt’s Min-K% Prob (right) for **Llama-family models** across all cases.

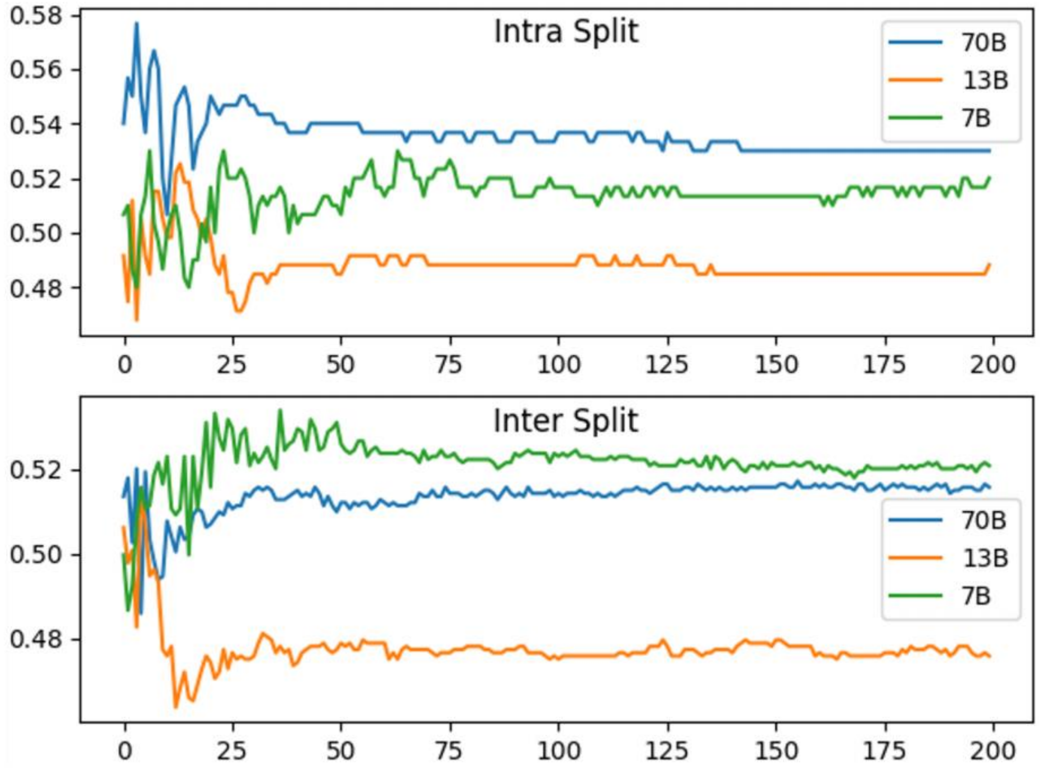


Distribution of Pearson correlation coefficients between model performance and prompt perplexity (left) and prompt’s Min-K% Prob (right) for **Gemma-family models and Mistral-7B** across all cases.

Unveiling the Difficulties in Identifying the Worst Prompt

Model-dependent Analysis





(Left) Visualization of Llama-2-7B-chat model’s hidden states using 2-dimensional PCA. The color gradient, from light to dark, represents the ranking of model performance on each case’s 11 prompts, from low to high.
(Right) Probing Llama-2-7B-chat model’s hidden states for prompt scoring.The x-axis stands for training steps. The y-axis represents the accuracy of the model’s predictions,quantified as the proportion of correctly judged prompt pairs out of all test pairs.

Unveiling the Difficulties in Identifying the Worst Prompt

Model-dependent Analysis

Model	All Pairs	Worst-Best
Llama-2-7B-chat	50.05	49.73
Llama-2-13B-chat	49.84	50.57
Llama-2-70B-chat	**53.02**	**58.92**
ChatGPT	50.16	51.00

We evaluate the model’s ability to perceive the quality of prompts with all paraphrase pairs (All Pairs) or best and worst prompt pairs (Worst-Best). We report the proportion of times the model prefers the prompt that leads to its better performance

⭐Take-away: Our explorations over prompt perplexity, Min-k% Prob, hidden states, and model preference show that <u>it is very challenging to identify the worst prompt in advance even with the access to the model.</u>

Improving Worst Prompt Performance

Method	Orig. Perf. ↑	Worst Perf. ↑	Best Perf. ↑	Avg. Perf. ↑	Standard Dev. ↓
			Llama-2-7b-chat		
Raw	25.61	5.42	43.54	19.52	13.32
Self-refinement	10.09(-15.52)	1.05(-4.37)	27.38(-16.16)	9.48(-10.04)	8.72(-4.60)
Voting	22.35(-3.26)	22.35(+16.93)	22.35(-21.19)	22.35(+2.83)	-
Distillation	18.29(-7.32)	3.89(-1.53)	40.27(-3.27)	19.31(-0.21)	12.72(-0.60)
			Llama-2-13b-chat		
Raw	27.48	4.83	52.05	23.97	16.25
Self-refinement	12.02(-15.46)	1.32(-3.51)	31.40(-20.65)	10.82(-13.15)	10.46(-5.79)
Voting	17.26(-10.22)	17.26(+12.43)	17.26(-34.79)	17.26(-6.71)	-
Distillation	25.90(-1.58)	5.99(+1.16)	47.78(-4.27)	22.09(-1.88)	14.30(-1.95)
			Llama-2-70b-chat		
Raw	32.23	9.38	54.86	29.18	15.61
Self-refinement	13.80(-18.43)	1.02(-8.36)	49.80(-5.06)	15.65(-13.53)	17.33(+1.72)
Voting	31.36(-0.87)	31.36(+21.98)	31.36(-23.50)	31.36(+2.18)	-
Distillation	29.30(-2.93)	7.99(-1.39)	50.15(-4.71)	26.44(-2.74)	14.83(-0.78)

- Self-refinement: let the model rewrite the prompt.
- Voting: perform voting-based generation based on all prompts.

$$P(y_i|X, y_{<i}) = \mathbb{E}_{x \in X} P(y_i|x, y_{<i})$$

- Distillation: encourage the model's predictions for various paraphrases to converge.

Take-away: Our results thoroughly examine existing efforts in reducing model sensitivity to prompt variations and clearly delineate their limitations.

Summary

The contributions of our work can be summarized as follows:

- Pioneering a Shift in Approach: We introduce the novel shift from task-level instructions to case-level queries, capitalizing on the concept of **worst prompt performance.**
- Benchmark Development: Our **benchmark** serves as a general testing tool for researchers to assess a model's capability to deliver stable responses to real-world users prior to deploying large language models.
- Highlighting Challenges: Through comprehensive experiments, we underscore the significant **challenge** of identifying the worst prompts and enhancing their performance in realistic scenarios.