Efficient multi-prompt evaluation of LLMs

Felipe Maia Polo
Department of Statistics
University of Michigan

NeurIPS 2024
Vancouver, Canada

Efficient multi-prompt evaluation of LLMs

Felipe Maia Polo[1]*, **Ronald Xu**[2,6], **Lucas Weber**[3], **Mírian Silva**[4,5,6], **Onkar Bhardwaj**[5,6]
Leshem Choshen[2,5,6], **Allysson Flavio Melo de Oliveira**[5,6], **Yuekai Sun**[1], **Mikhail Yurochkin**[5,6]
[1]University of Michigan, [2]MIT, [3]University Pompeu Fabra, [4]Federal University of Minas Gerais
[5]IBM Research, [6]MIT-IBM Watson AI Lab

Abstract

Most popular benchmarks for comparing LLMs rely on a limited set of prompt templates, which may not fully capture the LLMs' abilities and can affect the reproducibility of results on leaderboards. Many recent works empirically verify prompt sensitivity and advocate for changes in LLM evaluation. In this paper, we consider the problem of estimating the performance *distribution* across many prompt variants instead of finding a single prompt to evaluate with. We introduce PromptEval, a method for estimating performance across a large set of prompts

LLMs are sensitive to the prompt of choice

Example

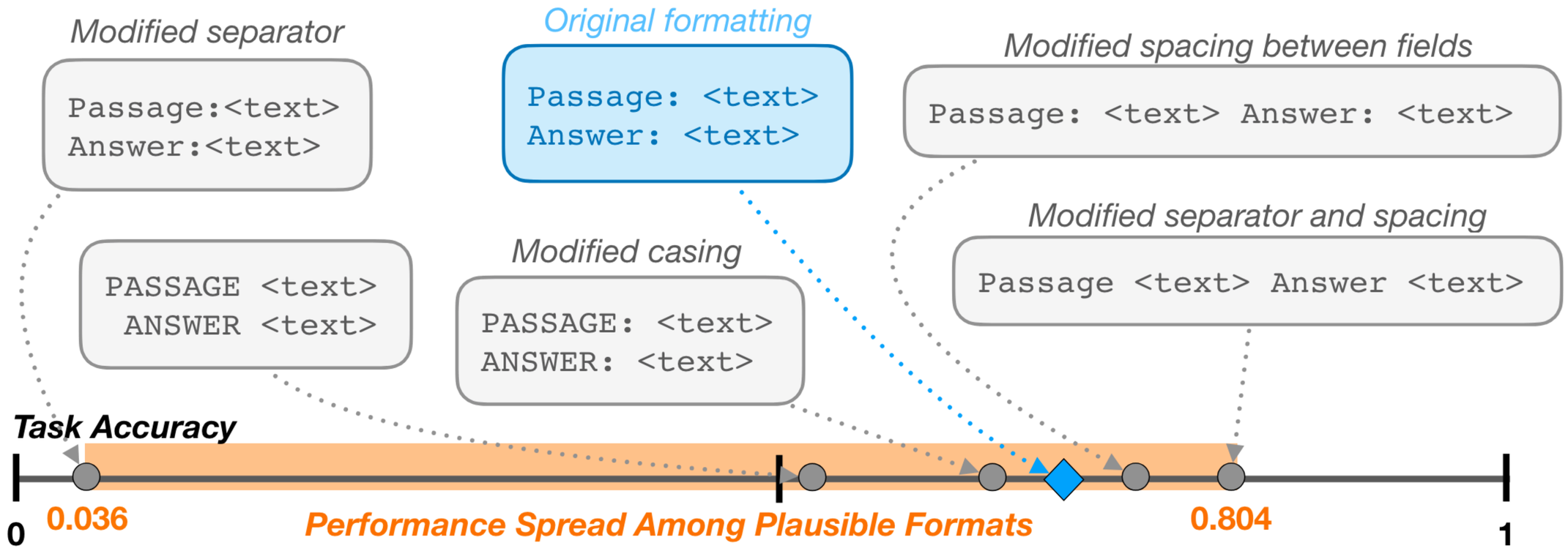


Figure extracted from Sclar et al (2023)

Benchmarking LLMs

Mirian, 25 yo

Data Scientist

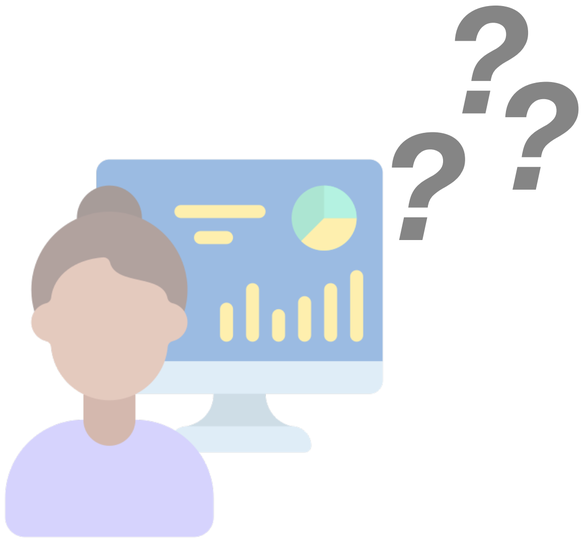
In the **leaderboard era**, **how** can we reliably **compare models** if they are extremely **sensitive** to used **prompts**?

Model	Average	ARC	HellaSwag	MMLU	TruthfulQA
davidkim205/Rhea-72b-v0.5	81.22	79.78	91.15	77.95	74.5
Contamination/contaminated_proof_7b_v1.0	81.14	78.07	90.22	78.92	82.29
Contamination/contaminated_proof_7b_v1.0_safetensor	81.14	78.07	90.22	78.92	82.29
MTSAIR/MultiVerse_70B					75.18
MTSAIR/MultiVerse_70B					75.09
					79.02
					77.82
					77.78
					76.67
					72.69

Model Name	LC Win Rate	Win Rate
GPT-4 Preview	50.0%	50.0%
Aligner 2B+Claude 3 Opus	41.8%	34.5%
Claude 3 Opus (02/29)	40.4%	29.0%
GPT-4	38.1%	23.6%
Aligner 2B+Qwen1.5 72B Chat	36.7%	31.8%
Qwen1.5 72B Chat	36.6%	26.5%
GPT-4 0314	35.3%	22.1%
Ein 70B v0.1	35.0%	24.8%
Claude 3 Sonnet (02/29)	34.9%	25.6%
Mistral Large (24/02)	32.7%	21.4%

GPT-4 (0613)	0.965
GPT-4 Turbo (1106 preview)	0.842
Palmyra X V3 (72B)	0.832
Palmyra X V2 (33B)	0.794

Benchmarking LLMs



Mirian, 25 yo

Data Scientist

In the **leaderboard era**, **how** can we reliably **compare models** if they are extremely **sensitive** to used **prompts**?

Model	Average	ARC	HellaSwag	MMLU	TruthfulQA
davidkim205/Rhea-72b-v0.5	81.22	79.78	91.15	77.95	74.5
Contamination/contaminated_proof_7b_v1.0	81.14	78.07	90.22	78.92	82.29
Contamination/contaminated_proof_7b_v1.0_safetensor	81.14	78.07	90.22	78.92	82.29
MTSAIR/MultiVerse_70B					75.18
MTSAIR/MultiVerse_70B					75.09
					79.02
					77.82
					77.78
					76.67
					72.69

Model Name	LC Win Rate	Win Rate
GPT-4 Preview	50.0%	50.0%
Aligner 2B+Claude 3 Opus	41.8%	34.5%
Claude 3 Opus (02/29)	40.4%	29.0%
GPT-4	38.1%	23.6%
Aligner 2B+Qwen1.5 72B Chat	36.7%	31.8%
Qwen1.5 72B Chat	36.6%	26.5%
GPT-4 0314	35.3%	22.1%
Ein 70B v0.1	35.0%	24.8%
Claude 3 Sonnet (02/29)	34.9%	25.6%
Mistral Large (24/02)	32.7%	21.4%

GPT-4 (0613)	0.965
GPT-4 Turbo (1106 preview)	0.842
Palmyra X V3 (72B)	0.832
Palmyra X V2 (33B)	0.794

Solution: multi-prompt evaluation :)

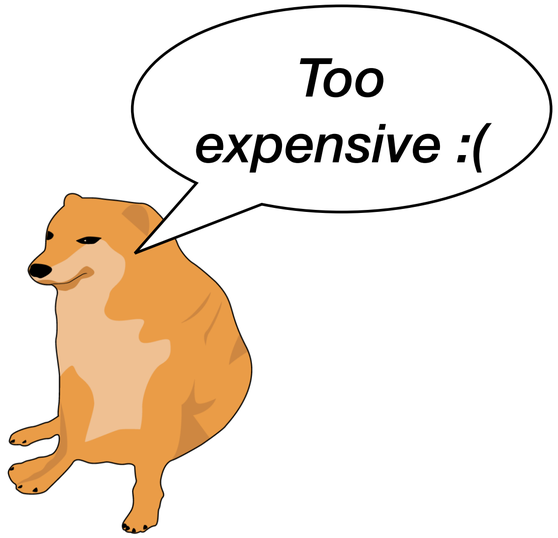
Downside: high costs :(

Multi-prompt evaluation

Ideal (precise but costly)

Prompt/ Question	Q1	Q2	Q3	…	avg performance
Prompt 1	1	0	0	…	0.45
Prompt 2	0	1	1	…	0.33
Prompt 3	1	1	1	…	0.91
…	…	…	…	…	…

Performance distribution



Efficient multi-prompt evaluation

Our proposal (less precise but much cheaper)

(1)

Prompt/ Question	Q1	Q2	Q3	…	avg performance
Prompt 1	1	?	?	…	?
Prompt 2	?	1	?	…	?
Prompt 3	1	?	?	…	?
…	…	…	…	…	…

Efficient multi-prompt evaluation

Our proposal (less precise but much cheaper)

(1)

Prompt/ Question	Q1	Q2	Q3	…	avg performance
Prompt 1	1	?	?	…	?
Prompt 2	?	1	?	…	?
Prompt 3	1	?	?	…	?
…	…	…	…	…	…

Y_{ij}

Efficient multi-prompt evaluation

Our proposal (less precise but much cheaper)

(1)

Prompt/ Question	Q1	Q2	Q3	...	avg performance
Prompt 1	1	?	?	...	?
Prompt 2	?	1	?	...	?
Prompt 3	1	?	?	...	?
...	...	...	...	...	...

Y_{ij}

(2)
$$p_{ij} = \mathbb{P}\left(Y_{ij} = 1; \psi, \gamma\right) = \frac{1}{1 + \exp\left[-\left(f_{\psi}\left(x_i\right) - g_{\gamma}\left(z_j\right)\right)\right]}$$

Trainable weights

How good prompt i is

Hardness of question j

Efficient multi-prompt evaluation

Our proposal (less precise but much cheaper)

(1)

Prompt/ Question	Q1	Q2	Q3	...	avg performance
Prompt 1	1	?	?	...	?
Prompt 2	?	1	?	...	?
Prompt 3	1	?	?	...	?
...	...	...	...	...	...

Y_{ij}

(3)

Prompt/ Question	Q1	Q2	Q3	...	Predicted avg permance
Prompt 1	1	$\hat{p}_{12}$	$\hat{p}_{13}$	...	$\widehat{\text{perf}}_1$
Prompt 2	$\hat{p}_{21}$	1	$\hat{p}_{23}$	...	$\widehat{\text{perf}}_2$
Prompt 3	1	$\hat{p}_{32}$	$\hat{p}_{33}$	...	$\widehat{\text{perf}}_3$
...	...	...	...	...	...

(2) $$p_{ij} = \mathbb{P}\left(Y_{ij} = 1; \psi, \gamma\right) = \frac{1}{1 + \exp\left[-\left(f_\psi\left(x_i\right) - g_\gamma\left(z_j\right)\right)\right]}$$

Trainable weights

How good prompt i is **Hardness of question j**

References

1. Sclar, Melanie, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting." *arXiv preprint arXiv:2310.11324* (2023).