POSIX: A Prompt Sensitivity Index for Large Language Models

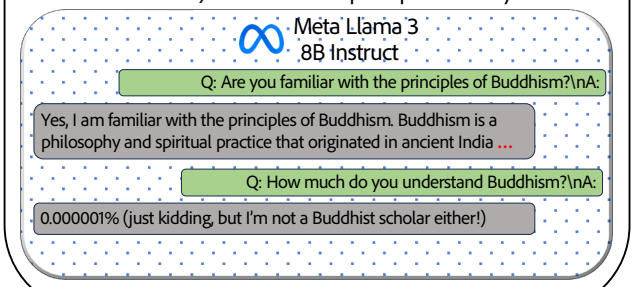
Anwoy Chatterjee[1]★, H S V N S Kowndinya Renduchintala[2]★,
Sumit Bhatia[2], Tanmoy Chakraborty[1]

Scan the QR for access to code and paper!

Your LLM evaluations are incomplete without prompt sensitivity!
Introducing POSIX - a novel PrOmpt Sensitivity IndeX
Wanna check how prompt sensitive your LLM is? Go, try out POSIX!
`pip install prompt-sensitivity-index`

Not all Prompts are created equal...
Small changes can lead to big surprises!

- Large Language Models (LLMs) excel at various tasks, yet even minor prompt changes – like rewording, spelling errors or template tweaks – can drastically affect their outputs.
- Standard LLM Evaluation Benchmarks (including blog posts on LLM releases) often overlook prompt sensitivity!



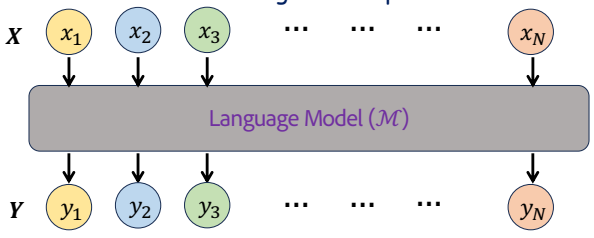
PrOmpt Sensitvity IndeX (POSIX)
The Key Idea

If x_1 and x_2 are intent-aligned, then ideally $\mathbb{P}(y_1|x_1) \approx \mathbb{P}(y_1|x_2)$ and similarly, $\mathbb{P}(y_2|x_1) \approx \mathbb{P}(y_2|x_2)$ should hold.

In other words - The log-likelihood of a response should not change much if the respective prompt is replaced by its intent-preserving variant

POSIX: Formal Definition

Intent-Aligned Prompts



$$\psi_{\mathcal{M},\boldsymbol{X}} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{L_{y_i}} \left| \log \frac{\mathbb{P}(y_j|x_i)}{\mathbb{P}(y_j|x_j)} \right|$$

- $\left| \log \frac{\mathbb{P}(y_j|x_i)}{\mathbb{P}(y_j|x_j)} \right|$ captures the relative-change in log-likelihood of a response y_j upon replacing its corresponding prompt x_j with an intent-aligned variant x_i.
- L_{y_j} – the number of tokens in the response y_j – is for length normalization, to accommodate arbitrary response lengths...

Given a language model $\mathcal{M}$ and a dataset $\mathcal{D} = \{\boldsymbol{X}_i\}_{i=1}^{M}$ of M intent-aligned prompt sets ($\boldsymbol{X}_i$s), the prompt sensitivity index (POSIX) for the language model $\mathcal{M}$ on the dataset $\mathcal{D}$ is defined as

$$\text{POSIX}_{\mathcal{M},\mathcal{D}} = \frac{1}{M} \sum_{i=1}^{N} \sum_{j=1}^{N} \psi_{\mathcal{M},\boldsymbol{X}}$$

What does POSIX capture?

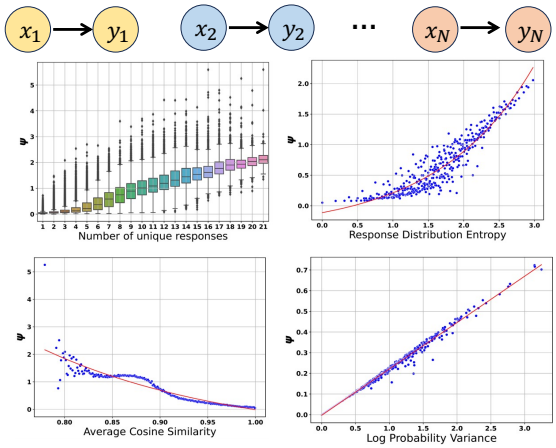
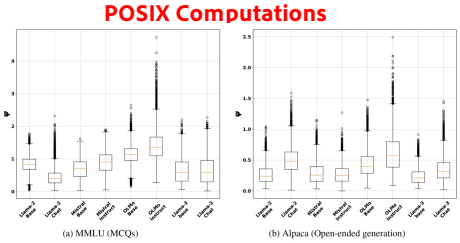


Figure 1: Correlation plots of ψ with each of the four factors described in Section 3.3 in the case of MMLU: (a) Response Diversity; (b) Response Distribution Entropy; (c) Semantic Coherence; (d) Variance in Confidence.

- Response Diversity: Higher the number of unique responses in the set $\{y_1, y_2, \ldots, y_N\}$ should indicate higher sensitivity
  Example: {A, B, C, A, A, D} vs {A, B, B, A, B, A}
- Response Distribution Entropy: Higher entropy of distribution of response frequencies (how often each response appears) should indicate higher sensitivity
  Example: {A, A, A, A, A, B} vs {A, B, A, B, A, B}
- Semantic Coherence: Lower semantic similarity among generated responses should indicate to higher sensitivity
- Variance in Confidence: Higher variance in the log-likelihood of the same response should also indicate higher sensitivity

POSIX Computations



(a) MMLU (MCQs) (b) Alpaca (Open-ended generation)

Figure 2: Box plots depicting the distribution of $\psi_{\mathcal{M},\mathbf{X}}$ for different instances of $\mathcal{M}$. The first plot corresponds to $\mathbf{X}$'s from MMLU dataset (MCQs) and the second plot corresponds to $\mathbf{X}$'s from the Alpaca dataset (open-ended generation).

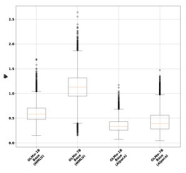
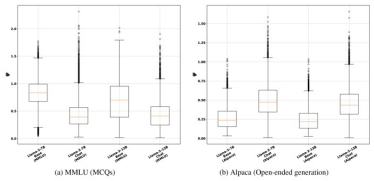


Figure 3: Box plots depicting distribution of $\psi_{\mathcal{M},\mathbf{X}}$ for two differently sized OLMo models (1B and 7B).



(a) MMLU (MCQs) (b) Alpaca (Open-ended generation)

Figure 4: Box plots depicting distribution of $\psi_{\mathcal{M},\mathbf{X}}$ for two differently sized Llama-2 models (7B and 13B).

Main Findings: Summarized

- Merely increasing the parameter count or Instruction tuning does not necessarily lower prompt sensitivity
- Adding few-shot examples - even just one - almost always significantly lowered prompt sensitivity
- Tweaks in the prompt template led to highest sensitivity in the case of MCQ-type tasks, whereas paraphrasing led to highest sensitivity in the case of open-ended generation tasks

★ *Equal contribution*

[1]*Indian Institute of Technology Delhi, India*
[2]*Adobe Inc., India*

EMNLP 2024