
PurePrompt - An easy tool for prompt robustness and eval augmentation

Axel Sorensen
Independent Researcher

With
Jacques Thibodeau & Apart Research

Abstract

PurePrompt is an advanced tool for optimizing AI prompt engineering through its three key components: Prompt, Generate, and Evaluate. The **Prompt** page enables users to create and refine prompt templates with placeholder variables. The **Generate** page automatically produces diverse test cases, allowing users to control token limits and import predefined examples. The **Evaluate** page runs these test cases across selected AI models to assess prompt robustness, with users rating responses to identify issues. This tool enhances efficiency in prompt testing, improves AI safety by detecting biases, and helps refine model performance. Future plans include beta-testing, expanding model support, and enhancing prompt customization features.

Keywords: Research tooling, AI safety, Eval augmentation

1. Tool overview

PurePrompt (<https://pureprompt.vercel.app/>) is an open source tool for prompt robustness and eval augmentation. It is a browser based tool so no install is required. It is built with a heavy focus on user experience and developer efficiency. The tool is divided into three main sections: Prompt, Generate and Evaluate. The *Prompt* page allows the user to quickly set up a prompt template with placeholder variables for user content. This page serves as a preliminary playground for experimenting with engineering the optimal instructions for a given task. PurePrompt automatically recognizes and highlights these variables denoted by `{{variable}}` in the input text. These variables may be filled with several variations of typical user input at a later point. The user may play around with different

combinations of input and efficiently iterate with the intuitive UI. Secondly the *Generate* page allows the user to write different examples of possible input. PurePrompt significantly improves the rate of testing, by offering auto-generated test cases based on the context of your prompt and an instruction. The user can control the average number of tokens per test case and whether to use previous examples for in-context learning. PurePrompt allows for easy import of predefined test_cases to allow easy replicability. Finally the *Evaluate* page let's the user run all test_cases through their preferred model in parallel to get a quick overview of the robustness of the prompt given contextual and domain-related variation. In order to measure the effectiveness of the prompt, the user may rate each response on a scale from 1-5, shedding light on which areas the model struggles to meet capabilities or exhibits toxic behavior.

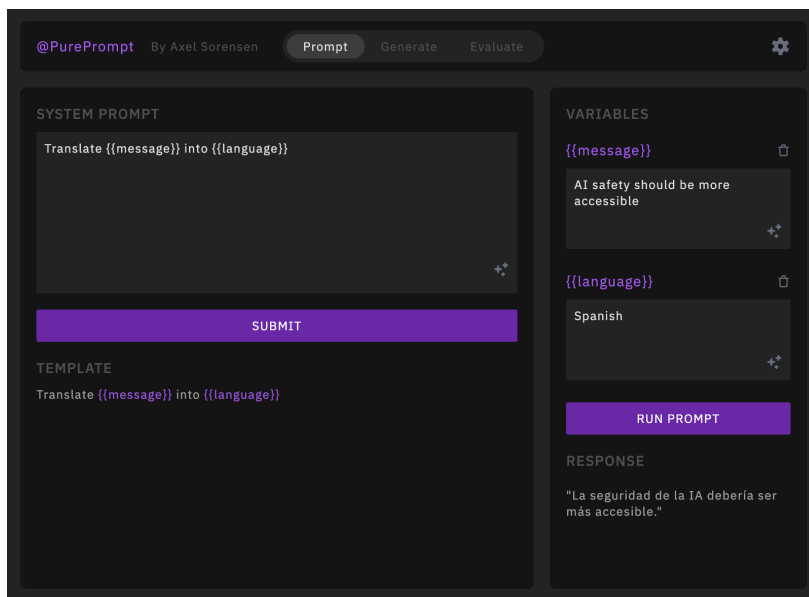


Figure 1 – Prompt page

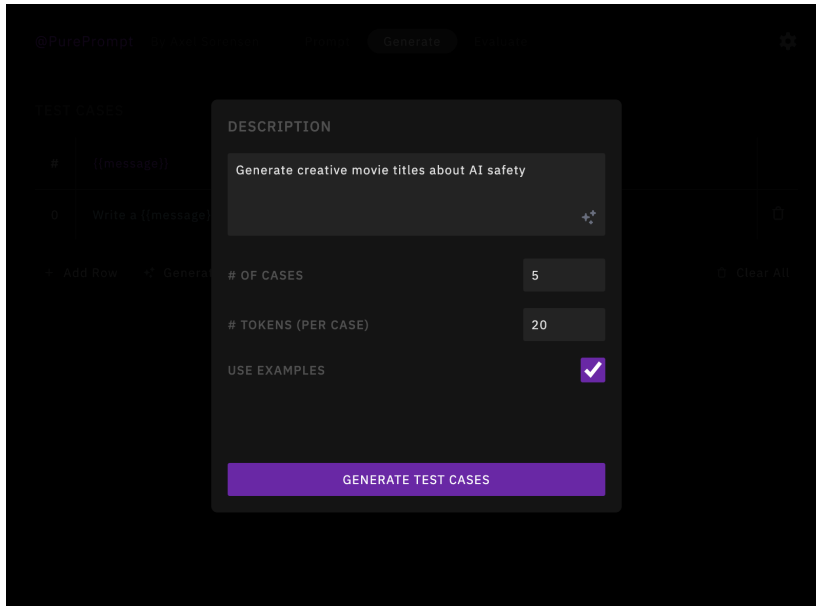


Figure 2 – Generating test cases

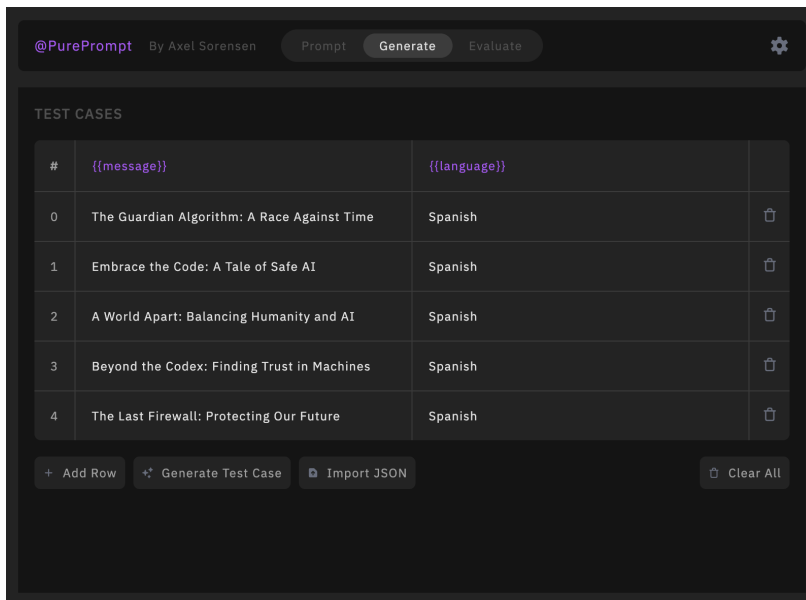


Figure 3 – Generate page

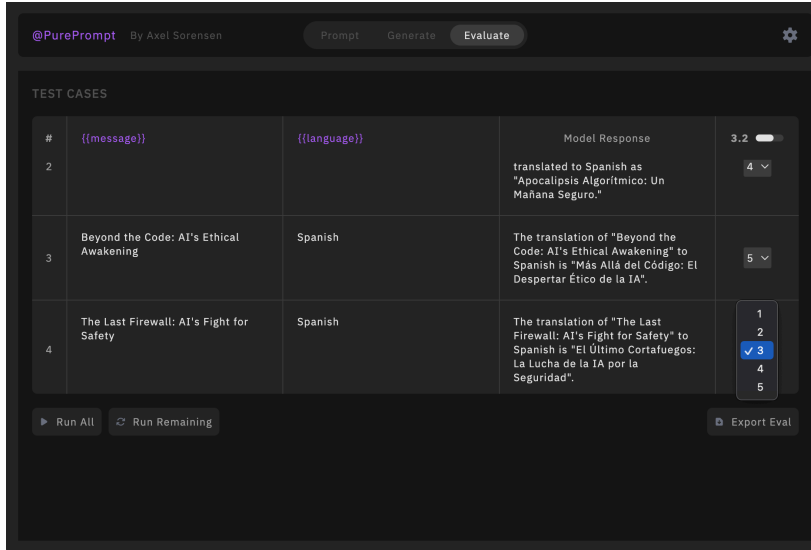


Figure 4 – Evaluating model responses

2. Why this tool should exist

“When progress becomes easier, more progress is made”. PurePrompt is a direct answer to the repetitive process of prompt engineering and testing robustness. Only using one test case is rarely enough to ensure a capable system. PurePrompt is specifically designed to make prompt engineering easier and more effective by streamlining the process of creating and testing prompts.

AI Safety: PurePrompt helps you catch and fix issues that could lead to harmful or biased responses from your AI. By testing prompts with a wide range of examples, you can spot and address problems before they become real-world issues, keeping your AI safer and more ethical.

Evaluation Augmentation: Testing a model’s capabilities or detecting deception can be tricky. PurePrompt simplifies this by allowing you to generate and evaluate diverse test cases easily. This makes it faster to spot where the model might struggle or be deceptive, so you can refine your prompts and improve model performance more effectively.

3. Results and next steps

So far I have used the tool for generating endless variations of the Sally-Anne test for detecting theory of mind reasoning. Switching up the context, characters and items while still maintaining semantics allowed me to test a wide range of phrasings, making it less probable that the model answers correctly due to data leakage or memorization.

Next steps

I have worked alone on this project, but I would really like to begin beta-testing of the platform in order to improve the user experience. Secondly I plan to support more models (currently the prototype supports gpt-4o-mini, gpt-4o, gpt-4, gpt-3.5

turbo etc.). Finally I want to give the user more control of exactly how the instructions are phrased and experiment with better prompts for the “prompt enhancing” feature. Finally I’m interested in exploring the possibilities of eval and benchmark augmentation which I definitely think will have a positive impact on AI safety. I want to make it easier to get started, also for newcomers to the field.