

A Real-time Visualization Framework to Enhance Prompt Accuracy and Result Outcomes Based on the Number of Tokens

Prashant D. Sawant^{1,*}

Abstract

In the rapidly evolving domain of artificial intelligence (AI), the efficacy of user-generated prompts has emerged as a critical factor influencing the quality of model-generated responses. Current methodologies for prompt evaluation predominantly rely on post-hoc analysis, which often leads to iterative prompting and increased computational overhead. Furthermore, the challenge of “prompt hallucinations,” where AI models produce irrelevant or nonsensical responses, persists as a significant impediment to effective AI utilization. The present article introduces a novel framework that leverages real-time analysis of prompts (number of tokens to start with) to provide users with immediate feedback on prompt quality. The proposed system employs a dynamic scoring mechanism that assesses prompts against a comprehensive corpus and a set of predefined quality criteria, outputting a relative strength percentage or a 1-10 scale rating which can be displayed by colors (Red to Amber to Green). By integrating this framework into the AI interface, users can iteratively refine their prompts before submission, thereby enhancing the interaction efficiency and reducing the likelihood of hallucinatory outputs. This approach represents a paradigm shift from reactive to proactive prompt optimization, paving the way for more seamless and effective human-AI collaboration. Also, it may revolutionize the way users engage with AI systems, fostering a more productive and harmonious human-AI synergy.

Keywords: Real-time prompt analysis, prompt optimization, AI interaction efficiency, preventing prompt hallucination, human-AI collaboration

INTRODUCTION

The advent of large language models (LLMs) like GPT-3 and its successors has revolutionized the field of artificial intelligence, providing unprecedented capabilities in natural language processing. The effectiveness of these models is significantly influenced by the art of prompt engineering—the practice of crafting inputs that guide the AI to produce desired outputs. Prompt methods range from basic techniques like zero-shot and few-shot prompting to more advanced strategies such as chain-of-thought prompting, which have been instrumental in enhancing AI’s problem-solving abilities [1–5].

*Author for Correspondence

Prashant D. Sawant
E-mail: prasdsaw@gmail.com

Director, Ai-D Consultancy, Melbourne, Australia

Received Date: March 26, 2024
Accepted Date: March 27, 2024
Published Date: April 05, 2024

Citation: Prashant D. Sawant. A Real-time Visualization Framework to Enhance Prompt Accuracy and Result Outcomes Based on the Number of Tokens. Journal of Artificial Intelligence Research & Advances. 2024; 11(1): 45–53p.

Despite these advancements, a phenomenon known as “AI hallucination” poses a challenge to the reliability of LLMs [6]. AI hallucinations occur when a model generates outputs that are nonsensical or unrelated to the prompt, often due to insufficient or biased training data, overfitting, or the use of complex, ambiguous prompts [7–11]. These inaccuracies can have serious implications, especially when AI is used in critical domains such as healthcare or legal services. To combat hallucinations, researchers have developed various

remedies, including Retrieval Augmented Generation (RAG) and other frameworks that integrate external data or employ structured prompting techniques to enhance the model's accuracy and robustness [12–14]. These methods aim to provide the AI with a richer context and reduce its reliance on potentially flawed pre-trained knowledge.

Google [15] has suggested best practices of prompting as follows.

- Communicate what content or information is most important.
- Structure the prompt: Start by defining its role, give context/input data, and then provide the instruction.
- Use specific, varied examples to help the model narrow its focus and generate more accurate results.
- Use constraints to limit the scope of the model's output. This can help avoid meandering away from the instructions into factual inaccuracies.
- Break down complex tasks into a sequence of simpler prompts.
- Instruct the model to evaluate or check its responses before producing them.

The current landscape of AI prompting methods showcases a blend of successes and drawbacks. On one hand, LLMs have achieved remarkable feats in generating creative content, summarizing complex information, and even coding. On the other hand, issues like factual inaccuracies, ethical concerns, and the cognitive load on users to craft effective prompts remain significant bottlenecks [16, 17].

However, human diversity and the multifaceted nature of human characteristics, including a multitude of languages and viewpoints, can give rise to a spectrum of errors in prompt creation and may preclude the formulation of prompts that garner universal acceptance as shown in Figure 1. Some of the intrinsic and extrinsic human factors that contribute to the spectrum of errors in prompt creation are stemming from:

- *Cultural nuances*: Variations in cultural context can lead to misinterpretations or inappropriate content.
- *Socioeconomic factors*: Disparities in wealth and education can affect the understanding and construction of prompts.
- *Regional idioms*: Localized expressions may not translate well across different regions, affecting clarity.
- *Personal biases*: Individual prejudices can inadvertently influence the tone and direction of prompts.
- *Historical context*: Lack of awareness of historical sensitivities can result in prompts that are offensive or inaccurate.
- *Psychological divergence*: Differences in mental models and cognitive biases can skew the intent of prompts.
- *Ethical standards*: Divergent moral values can lead to conflicting views on what constitutes an acceptable prompt.
- *Linguistic limitations*: Language proficiency levels can impact the ability to craft coherent and precise prompts.
- *Technological access*: Varying degrees of access to technology can lead to unequal opportunities for prompt optimization.
- *Physical abilities*: Disabilities may affect the interaction with AI systems and the formulation of prompts.

These parameters [18-21] highlight the complexity of creating prompts that are both effective and sensitive to the vast array of human differences. They underscore the need for inclusive design and adaptive AI systems that can accommodate and understand this diversity.

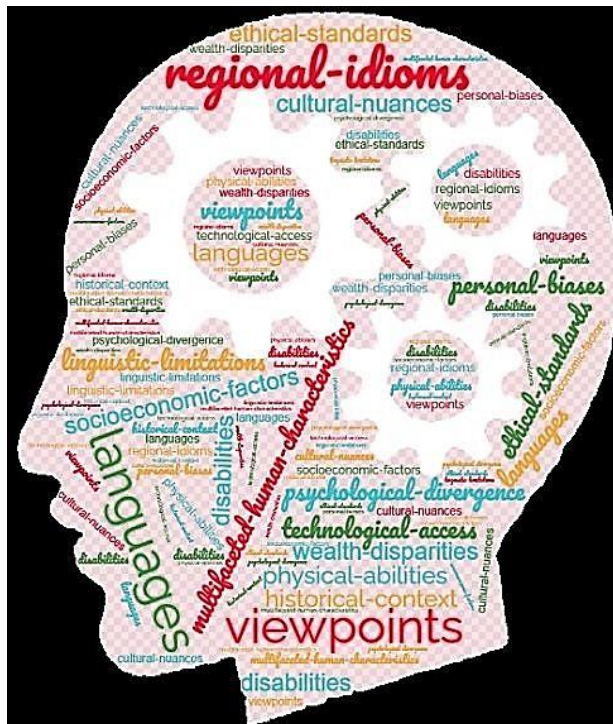


Figure 1. Intrinsic and extrinsic human factors contribute to the spectrum of errors in prompt creation.

In light of these challenges, the new method proposed in this article offers a promising solution. It suggests a real-time prompt evaluation system that provides users with immediate feedback on the quality of their prompts, expressed as a percentage or a scale rating. This innovative approach aims to streamline the prompting process, reduce the cognitive burden on users, and minimize the occurrence of hallucinations by ensuring that prompts are clear, specific, and well-structured before they are submitted to the AI.

BEST PRACTICES, FRAMEWORKS, AND QUALITY TESTING

Factors to Establish Best Practices

To establish the best practices for prompt design in chatbots, it is important to consider several key factors to ensure the prompts are effective and elicit the desired responses. Some of the best practices are as follows.

- *Clarity:* Prompts should be clear and unambiguous to guide the chatbot's response accurately.
- *Relevance:* Ensure prompts are relevant to the chatbot's capabilities and the user's intent.
- *Brevity:* Keep prompts concise to maintain user engagement and prevent overwhelming the chatbot.
- *Specificity:* Specific prompts lead to more precise and useful responses.
- *Contextual awareness:* Include necessary context to make the conversation flow naturally.
- *User intent:* Clearly convey the action or information desired from the chatbot.
- *Testing and Iteration:* Regularly test prompts with various inputs and refine prompts based on outcomes.
- *Feedback mechanism:* Incorporate user feedback to continuously improve prompt design.
- *Avoid jargon:* Use simple language that all users can understand.
- *Conversational tone:* Match the tone with the chatbot's persona for a human-like interaction.
- *Prompt variability:* Use diverse types of prompts to keep the conversation dynamic.

By adhering to these best practices, we can design effective prompts that can enhance the performance of the chatbots and provide a better experience for users.

Testing the Quality of Prompts

Testing the quality of prompts in the chatbot is crucial for ensuring that it understands and responds to user inputs effectively.

Some steps to test the quality of the chatbot's prompts are as follows.

- *Define objectives:* Clearly define the intended results that we need to achieve with each prompt, determine the expected outcomes, and how they align with the chatbot's goals.
- *Create test cases:* Develop a set of test cases that cover a wide range of scenarios, including common queries, edge cases, and potential misunderstandings.
- *Use tools:* Use tools like promptfoo, which can help systematically test prompts, models, and RAGs answers with predefined test cases [22].
- *Manual testing:* Conduct manual testing by entering prompts into the chatbot and evaluating the responses for accuracy, relevance, and helpfulness.
- *User testing:* Perform user acceptance testing or UAT with real users to gather feedback on the chatbot's performance in real-world scenarios [23].
- *Analyze responses:* Review the chatbot's responses to different prompts and check if they meet the defined objectives. Look for patterns in any incorrect or suboptimal responses.
- *Iterate and improve:* Use the insights gained from testing to refine the prompts and adjust them based on the chatbot's performance and user feedback.
- *Monitor continuously:* Regularly analyze chatbot logs and user interactions to identify areas for improvement and ensure the chatbot remains effective over time [24].

By following these steps, we can systematically evaluate and enhance the quality of the chatbot's prompts, leading to a better user experience.

Defining Quality Metrics for AI Prompts

Defining quality metrics for AI prompts is a crucial step in ensuring that the AI model generates relevant and accurate responses. The following are some of the steps that can be considered to establish the quality metrics.

- *Relevance:* The prompt should align with the intended purpose and context, and it should guide the AI to produce outputs that are topically relevant to the query.
- *Coherence:* The AI's response should be logical, well-structured, and easy to understand, and it should follow a clear and consistent train of thought.
- *Accuracy:* The information provided in the AI's response should be correct and up to date for prompts that require factual information.
- *Bias:* The AI's response should be free from unintended biases or stereotypes ensuring the language and content are neutral and inclusive.
- *Efficiency:* Evaluating the average time or length required to generate an output as a good prompt should lead to the efficient generation of responses without unnecessary verbosity.
- *Groundedness:* The responses should be grounded in reality, meaning they should be based on evidence and facts rather than speculation or fiction.
- *Fluency:* The output should be fluent, with proper grammar and syntax, making it readable and understandable.
- *Similarity:* For tasks like summarization or translation, the output should closely match the expected result or ground truth in terms of content and meaning.
- *Objectivity:* The output should maintain a neutral tone and avoid subjective or biased words unless the prompt specifically requires an opinion.

Automation Tools to Assess Prompts

Thus, automated tools can quickly assess prompts based on objective criteria like coherence, fluency, and relevance, while human evaluators can provide insights into more subjective aspects like bias and

groundedness. The combination of these methods can provide a comprehensive assessment of prompt quality. In this regard, platforms like Azure AI Studio [24] offer some features for monitoring and evaluating prompt quality, including the application of Responsible AI evaluation metrics [25-27]. Azure AI Studio offers features like Prompt Flow, which can be used to build, benchmark, evaluate, and deploy real-time inference endpoints. Additionally, open-source tools like promptfoo provide systematic test prompts and evaluate their quality [22]. While promptfoo is not a real-time system, it can provide insights into how we might structure our system.

AutoPrompt [28] is a framework that enhances prompts for real-world use cases by automatically generating high-quality prompts tailored to user intentions. It uses a calibration process to iteratively build a dataset of challenging edge cases and optimizes the prompt accordingly. The AutoPrompt method can be useful for masked language models without the need for additional parameters or fine-tuning and has shown promising results in tasks such as sentiment analysis and natural language inference.

Above mentioned tools can significantly reduce manual effort in prompt engineering and effectively address common issues such as prompt sensitivity and inherent prompt ambiguity. These tools are a part of a growing ecosystem of resources aimed at empowering users to produce high-quality robust prompts using the power of large language models.

However, an important drawback of the above-mentioned frameworks and tools is that they do not provide real-time visualization of the prompt quality based on several tokens for end-users who may not be experienced well in prompt engineering.

The Proposed Framework

The concept of displaying prompt quality as a percentage or on a scale during the writing process is an interesting idea. It would involve real-time analysis of the prompt against a set of criteria or a corpus to determine its effectiveness in eliciting the desired response from a language model like GPT (e.g., Copilot or ChatGPT). Currently, there is not a standard feature within GPT models that provides this functionality. However, theoretically, it could be implemented by developing a custom tool or plugin that:

- *Analyzes the prompt:* The tool would need to analyze the prompt in real-time as it is being written.
- *Compares to a corpus:* It would compare the prompt to a large corpus of effective prompts and their outcomes to determine the quality.
- *Calculates a score:* Based on this comparison, the tool would calculate a score that reflects the prompt's relative strength or quality.
- *Displays the score:* The score would then be displayed to the user, helping them adjust the prompt to improve its quality before submission.

To create such a tool, we would need to consider the following steps.

- Define what constitutes a 'quality' prompt.
- Develop an algorithm to evaluate prompt quality.
- Train the tool on a dataset of prompts and their effectiveness.
- Integrate this tool into the user interface where prompts are written.

This tool would require access to a substantial corpus of data and a well-defined set of metrics for evaluating prompt quality. It would also need to be fast enough to provide feedback in real-time without disrupting the user's workflow. Additionally, having a system that evaluates the quality of prompts in real-time and provides feedback could indeed help reduce instances of "prompt hallucinations," where an AI model generates outputs that are nonsensical or unrelated to the prompt. By ensuring that prompts are clear, specific, and well-structured, users can guide the AI to produce more accurate and relevant responses. While this is not a feature currently available, it is a valuable

suggestion for future development in the field of AI and natural language processing. It could indeed streamline the process of interacting with AI models and enhance the user experience by reducing the need for iterative prompting.

RESULTS AND DISCUSSION

Implementing a Real-time Prompt Quality Display System

Implementing a real-time prompt quality display system into our AI model involves several steps, which include integrating a feedback mechanism that assesses the quality of prompts as they are being written.

A high-level overview of this approach is as follows.

- *Define Quality Metrics:* Determine what constitutes a ‘good’ prompt for the specific use case based on clarity, specificity, likelihood of generating accurate responses, etc.
- *Develop an Evaluation Algorithm:* Create an algorithm that can score prompts in real-time according to the defined metrics and it could involve natural language processing techniques and machine learning models trained on a dataset of effective prompts and their outcomes.
- *Integrate with AI Model:* The evaluation system should be integrated into the environment where prompts are entered, providing immediate feedback to the user. This could be done through a plugin or an API that interacts with the AI model.
- *User Interface Design:* Design a user interface that displays the prompt quality score in a clear and non-intrusive manner, such as a percentage or a 1-10 scale next to the prompt input field.
- *Test and Iterate:* Before full deployment, test the system with actual users to gather feedback and refine the evaluation algorithm and user interface.
- *Deploy and Monitor:* Once the system is fine-tuned, deploy it for all users and continuously monitor its performance, adjusting as needed based on user feedback and changes in model behavior.

The key to a successful implementation is a well-defined set of quality metrics and a user-friendly interface that provides meaningful and actionable feedback without overwhelming the user.

Example of Real-time Prompt Quality Monitoring System

A simple HTML code was created that includes a script to change the color of the top bar based on the text strength when the text type is in the chatbot. The colors will transition from red to amber to green as the number of characters increases.

The code (Figure 2) is as follows

It generated an HTML file that runs on Google Chrome for demonstration purposes. The results are displayed below.

No text was written in the chatbot which displayed a red bar on the top. This is the initial state of the chatbot when an inexperienced user starts using the chatbot as shown in Figure 3.

A text with three words (tokens) written which are not sufficient for the prompting as shown in Figure 4.

Thus, the inexperienced user will know that their prompt is insufficient to get meaningful results.

When the words increase further to four words the display color changes to amber which will inform the inexperienced user that the prompt has improved but is not sufficient to get decent results as shown in Figure 5.

When the words increase further to 5-7 words the display color changes to green color indicating that the prompt is of minimum superior quality to get a decent quality result as shown in Figure 6.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Simple Chatbot</title>
7 <style>
8   body { font-family: Arial, sans-serif; }
9   #chatbot { width: 300px; margin: auto; }
10  #top-bar { height: 20px; }
11  #chat-input { width: 100%; }
12 </style>
13 </head>
14 <body>
15 <div id="chatbot">
16   <div id="top-bar" style="background-color: red;"></div>
17   <input type="text" id="chat-input" placeholder="Type your message here..." oninput="changeTopBarColor(this.value)">
18   <!-- Add your chat messages here -->
19 </div>
20
21 <script>
22 function changeTopBarColor(text) {
23   var topBar = document.getElementById('top-bar');
24   // Define the color 'amber' as it's not a standard CSS color
25   var amber = '#FFBF00';
26
27   if (text.length < 10) {
28     topBar.style.backgroundColor = 'red';
29   } else if (text.length >= 10 && text.length < 20) {
30     topBar.style.backgroundColor = amber;
31   } else {
32     topBar.style.backgroundColor = 'green';
33   }
34 }
35 </script>
36 </body>
37 </html>
38
```

Figure 2. HTML code to display color code as a function of tokens.

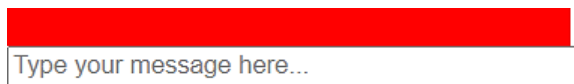


Figure 3. Red color bar display for zero tokens.



Figure 4. Red color bar display for three tokens.



Figure 5. Amber color bar display for four tokens

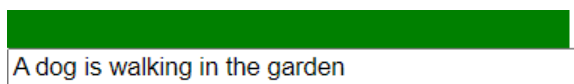


Figure 6. Green color bar display for more than five tokens.

This is a low-profile example to demonstrate the principle that can be embedded in popular chatbots like Copilot or ChatGPT.

Benefits of a real-time prompt quality display system

- *Reduced cognitive load:* Users can focus on their objectives rather than the mechanics of prompt crafting.
- *Enhanced learning:* Real-time feedback facilitates a learning loop for users, helping them understand how different prompts influence AI behavior.
- *Increased efficiency:* Immediate feedback can reduce the time spent on trial and error, leading to quicker and more productive AI interactions.
- *Improved model understanding:* Users gain insights into how the model processes information, which can inform better prompt design.
- *Quality control:* Helps maintain a consistent level of quality in prompts, which is particularly important in professional or commercial settings.

CONCLUSIONS

The present article demonstrates a simple color display framework that can be added to chatbots like Copilot or ChatGPT that can help inexperienced or experienced users understand the quality of their tokens to some extent. This framework can be further modified to include various other parameters that are related to the corpus searches. Such a system could significantly enhance the user experience and the overall utility of AI models in various applications.

REFERENCES

1. Amatriain X. Prompt Design and Engineering: Introduction and Advanced Methods. arXiv. 2024;2401.14423.
2. Huang L, Yu WJ, Ma WT, Zhong WH, Feng ZG, Wang HT, et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. arXiv. 2023;2311.05232.
3. Sun H, An RL. Perspective on RLHF, Prompting, and Beyond. arXiv. 2023;2310.06147.
4. Mosbach M, Pimentel T, Ravfogel S, Klakow D, Elazar Y. Few-Shot Fine-Tuning vs. in-Context Learning: A Fair Comparison and Evaluation. arXiv. 2023;2305.16938. DOI: 10.18653/v1/2023.findings-acl.779.
5. Dhuliawala S, Komeili M, Xu J, Raileanu R, Li X, Celikyilmaz A, et al. Chain-of-Verification Reduces Hallucination in Large Language Models. arXiv. 2023;2309.11495.

6. Besta M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, Gianinazzi L, et al. Solving Elaborate Problems with Large Language Models. arXiv. 2023;2308.09687.
7. What are AI hallucinations? IBM. Available from: <https://www.ibm.com/topics/ai-hallucinations>. Accessed on 14 March 2024.
8. Salvagno M, Taccone FS, Gerli AG. Artificial intelligence hallucinations. Crit Care. 2023;27:180. DOI: 10.1186/s13054-023-04473-y, PubMed: 37165401.
9. Bentley SV, Naughtin C. Both humans and AI hallucinate – But not in the same way. Available from: <https://www.csiro.au/en/news/All/Articles/2023/June/humans-and-ai-hallucinate>. Accessed on 14 March 2024.
10. Maleki N, Padmanabhan B, Dutta K. AI Hallucinations: A Misnomer Worth Clarifying. arXiv. 2024;2401.06796.
11. Bruno A, Mazzeo PL, Chetouani A, Tliba M, Kerkouri MA. Insights into Classifying and Mitigating LLMs' Hallucinations. arXiv. 2023;2311.08117.
12. Siriwardhana S, Weerasekera R, Wen E, Kaluarachchi T, Rana R, Nanayakkara S. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. Trans Assoc Comput Linguist. 2023;11:1–17. DOI: 10.1162/tacl_a_00530.
13. Gao YF, Xiong Y, Gao XY, Jia KX, Pan JL, Bi YX, et al. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv. 2023;2312.10997.
14. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv. 2020;2005.11401.
15. Prompt engineering for generative AI. Google. Available from: <https://developers.google.com/machine-learning/resources/prompt-eng>. Accessed on 14 March 2023.
16. Skulmowski A, Xu KM. Understanding cognitive load in digital and online Learning: A new perspective on extraneous cognitive load. Educ Psychol Rev. 2022;34:171–196. DOI: 10.1007/s10648-021-09624-7.
17. Trotta A, Ziosi M, Lomonaco V. The future of ethics in AI: Challenges and opportunities. AI Soc. 2023;38:439–441. DOI: 10.1007/s00146-023-01644-x.
18. Shams RA, Zowghi D, Bano M. AI and the quest for diversity and inclusion: A systematic literature review. AI Ethics. 2023. DOI: 10.1007/s43681-023-00362-w.
19. Crowell R. Why AI's diversity crisis matters, and how to tackle It. Nature. DOI: 10.1038/d41586-023-01689-4, PubMed: 37208514.
20. Howard A, Isbell C. Diversity in AI: The invisible men and women. MIT Sloan Manag Rev. 2020.
21. Zowghi D, da Rimini F. Diversity and Inclusion in Artificial Intelligence. arXiv. 2023;2305.12728.
22. Collins S. How to use Promptfoo for LLM testing. The deep hub Feb 2024. Medium. Available from: <https://medium.com/thedeephub/how-to-use-promptfoo-for-llm-testing-13e96a9a9773>. Accessed on 16 March 2024.
23. Leung HKN, Wong PWL. A study of user acceptance tests. Softw Qual J. 1997;6:137–149. DOI: 10.1023/A:1018503800709.
24. Alto V. Evaluating LLM-powered applications with Azure AI studio. Medium. Available from: <https://medium.com/microsoftazure/evaluating-llm-powered-applications-with-azure-ai-studio-b3cec3eba322>. Accessed on 16 March 2024.
25. Xia BM, Lu QH, Zhu LM, Lee SU, Liu Y, Xing ZC. Towards a responsible AI metrics catalogue: A collection of metrics for AI accountability. arXiv. 2023;2311.13158.
26. Berman G, Goyal N, Madaio M. A Scoping Study of Evaluation Practices for Responsible AI Tools: Steps Towards Effectiveness Evaluations. arXiv. 2024;2401.17486. DOI: 10.1145/3613904.3642398.
27. Xia B, Lu Q, Zhu L, Lee SU, Liu Y, Xing Z. Towards a responsible AI metrics catalogue: A collection of metrics for AI accountability. Semantic Scholar. 2023. Corpus ID: 265352192.
28. Shin T, Razeghi Y, Logan RL, Wallace E, Singh S. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. arXiv. 2020;2010.159801.