www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 10, October 2025, pp : 502-503

e-ISSN : 2583-1062
Impact Factor : 7.001

PROMPT DEBUGGER: AN AI-POWERED TOOL TO ENHANCE PROMPT ENGINEERING

Yash V. Bhonde[1], R.S. Durge[2]

[1]B.E. (CSE) Student, Dr. Rajendra Gode Institute Of Technology & Research, Amravati, Maharashtra, India, Department Of Computer Science & Engineering , India.

[2]Guide, Dr. Rajendra Gode Institute Of Technology & Research, Amravati, Maharashtra, India, Department Of Computer Science & Engineering, India.

DOI: https://www.doi.org/10.58257/IJPREMS44209

ABSTRACT

Human-machine interaction has been drastically changed by artificial intelligence (AI), with large language Models (LLMs) like GPT are exhibiting previously unheard-of levels of language generation and comprehension. But these models are extremely sensitive to prompt design, meaning that even small adjustments to the wording or structure can have a big impact on the results Quality. This problem is addressed by the Prompt Debugger system, an AI-assisted framework that examines, improves, and optimizes natural language prompts. It works similarly to a compiler and debugger, identifying ambiguity, redundancy, and vagueness while making real-time improvement suggestions. The system decreases trial-and-error cycles, increases AI reliability, and makes prompt engineering more approachable by improving prompt clarity and context. non-professionals. In order to improve AI usability, scalability, and human-AI collaboration, this paper describes the Prompt Debugger's architecture, methodology, and performance.

Keywords: Large Language Models, Prompt Engineering, AI Debugger, GPT, Optimization, And NLP.

1. INTRODUCTION

AI-driven communication is being revolutionized by large language models (LLMs) like GPT-4, which allow for human-like text generation and reasoning. However, how well the user constructs the input prompt determines how effective these models are. The potential of AI is limited by ambiguous, insufficient, or badly constructed prompts, which frequently produce irrelevant results. To address this issue, the Prompt Debugger provides a structured, intelligent approach to prompt design. Using linguistic and semantic criteria, it evaluates prompts and provides useful recommendations for improvement. By bridging the gap between machine interpretation and user intent, the system transforms prompt engineering into a disciplined field that is necessary for professional AI use.

2. METHODOLOGY

A multi-layered architecture that combines database administration, AI integration, backend processing, and frontend interaction powers the Prompt Debugger. Through an easy-to-use interface, users submit prompts, which are subsequently examined for errors like ambiguity or context omission. The Flask-developed backend refines input by interacting with GPT APIs.

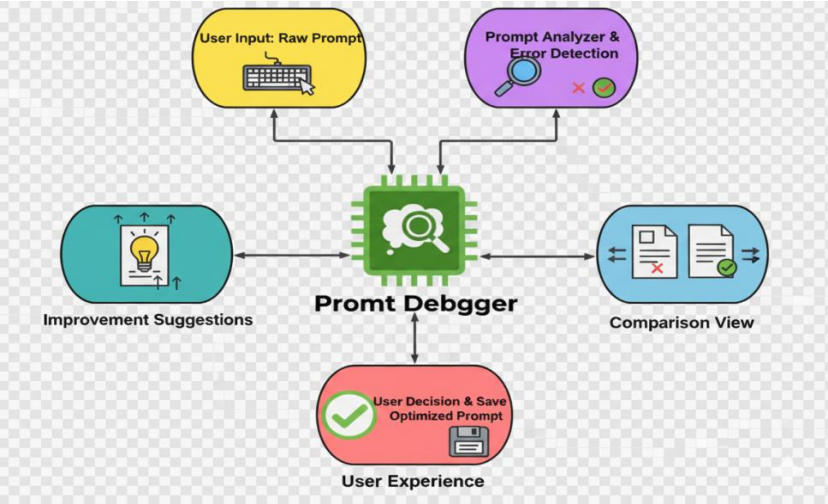

Fig 1: Technical Methodology

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 10, October 2025, pp : 502-503

e-ISSN : 2583-1062
Impact Factor : 7.001

By comparing the initial and enhanced prompts, the system offers feedback via an interactive dashboard. Prompt history and learning progression are preserved for ongoing improvement thanks to database storage.

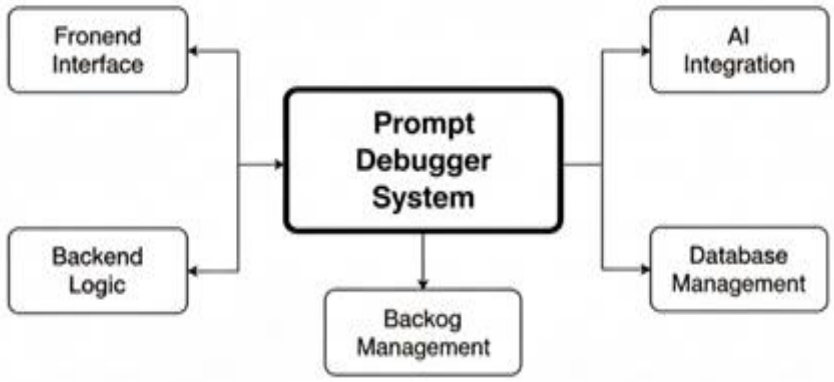

Fig 2: Architecture of the Prompt Debugger System

3. ANALYSIS AND MODELING

The system uses natural language processing (NLP) models to assess prompts according to their contextual completeness, specificity, and clarity. To find flaws, each prompt is subjected to syntactic and semantic analysis. After that, the AI model produces more precise options while preserving the user's intent. By comparing AI responses prior to and following debugging, quantitative performance evaluation is carried out using metrics such as accuracy, coherence, and relevance. This analysis shows that outputs from refined prompts are substantially more contextually accurate.

4. RESULT AND DISCUSSION

Tests on several task categories, including data extraction, academic summarization, and creative writing, showed a steady improvement in response quality. Prompts that were optimized decreased redundant outputs, reduced iteration time by almost 35% and enhanced contextual accuracy. The Prompt Debugger proved to be flexible across various LLM APIs and domains, making it a useful tool for researchers, students, and businesses. By providing a visual explanation of why particular prompts work better, it also enhanced user learning.

5. CONCLUSION

An important step toward organized, effective, and approachable AI interaction is the Prompt Debugger. It allows users to create excellent prompts with less work by fusing linguistic analysis, machine learning, and adaptive feedback. It enhances consistency and productivity in enterprise AI applications, research, and education. In the future, multimodal prompt support for text, image, and audio inputs will be enabled, multilingual capabilities will be developed, and bias detection will be integrated.

ACKNOWLEDGEMENT

The Department of Computer Science & Engineering at Dr. Rajendra Gode Institute of Technology & Research, Amravati, provided academic support for this study, and the author is deeply grateful to Dr. R. S. Durge for his invaluable advice and mentoring.

6. REFERENCES

[1] Brown, T. et al. "Language Models are Few-Shot Learners," NeurIPS, 2020.
[2] Johnson, M. "Improving NLP Outputs with Prompt Debugging," IEEE, 2023.
[3] Smith, J. "Techniques for Prompt Engineering," AI Journal, 2024.
[4] OpenAI Documentation, 2025.