Enhancing Transparency in Large Language Model-Based Research with PromptLog: A Tool for Prompt History Tracking and Reporting

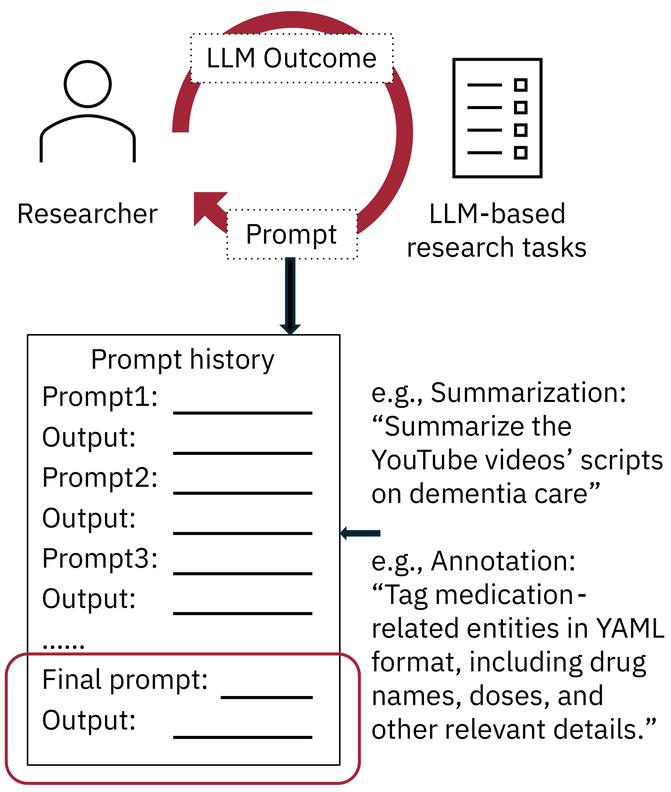
Ananya Shrivastava, B.S.c[1], Lu Wang, M.A.P., Ph.D. c[1], Elham Aghakhani, M.S., Ph.D.c[2], Rezvaneh Rezapour, Ph.D.[2], Jina Huh-Yoo, MHCI, Ph.D.[1]

[1]Department of Computer Science, Stevens Institute of Technology, Hoboken, USA;
[2]College of Computing & Informatics, Drexel University, Philadelphia, USA

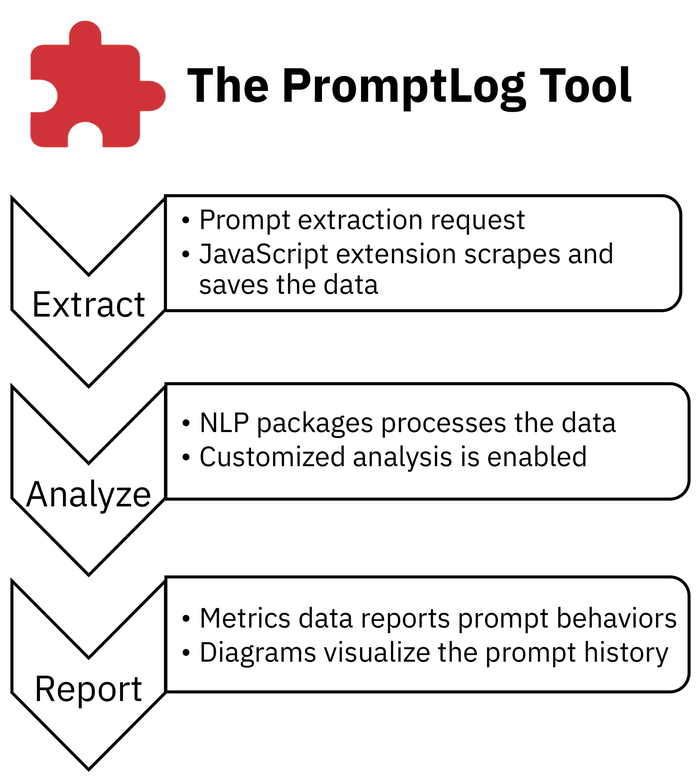
ABSTRACT

Large language models (LLMs) have shown significant potential in health informatics. Prompt engineering plays a critical role in optimizing LLM performance across diverse applications. However, researchers often report only the final prompts and their outcomes, even though it took many iterations for the prompts to get to their final form. Such a lack of transparency compromises reproducibility in LLM-based research.

We introduce *PromptLog*—a tool that tracks prompt changes (e.g., word count, topic shifts, deletions)—to standardize and improve the scientific rigor of reporting practices of LLM research at AMIA.



Reported partially in scientific papers



WORK-IN-PROGRESS

The figure below illustrates PromptLog's workflow through a proof-of-concept implementation, demonstrating researchers' iterative prompting practices and how PromptLog captures and reports prompt history. The figure shows an example of a report that PromptLog can generate. We have completed the prompt extraction and data analysis separately.

However, the tool is limited by the continued changes in policies of LLM companies, such as copyright and designs, that may hinder certain prompts from being available to be extracted.

PromptLog Report

LLM Model Version: GPT-4o
Time of initial prompt: Mar 20th, 2025, 3:00 pm
Total prompts analyzed: 14 prompts revised over 2.3 hours

Lexical Evolution:

Metrics	Initial prompt	Final prompt	Δ
Word count	28	18	-36%
Char count	184	127	-31%
Unique terms	22	15	-32%

Semantic Analysis
Topic Drift:
Prompt 1: ["clinical guidelines", "patient history"]
Prompt 2: ["Drug therapies", "side effects"]
Prompt 3: ["Drug comparisons", "risk factors"]
Prompt 4: ["treatment options", "risk factors"]

Semantic Similarity:
- The mean similarity between steps: 0.68 (±0.12)
- Largest semantic jump: Prompt 3→Prompt 4 (Δ=0.41, added example cases)

Structural Changes
- Additions: 47 tokens (mostly clinical examples)
- Deletions: 57 tokens (51% qualifiers e.g., "generally")
- Template shifts: added bullet points in Prompt 3

Summary
The prompt progressed from asking LLM to generate: (1) general diabetes guidelines → (2) drug-specific details → (3) head-to-head medication comparisons → (4) simplified treatment options with persistent risk assessment.

Figure 1. The figure shows the workflow from prompt engineering processes to the PromptLog-generated report.

INTRODUCTION

Large language models (LLMs) have shown significant potential in medical informatics, such as extracting information from Electronic Health Records, assisting with literature searches, and simplifying medical jargon for enhanced patient-centered care[1]. Prompt engineering plays a critical role in optimizing LLM performance across diverse applications[2].
Khalid and Witmer[3] outlined a general four-step prompt engineering process:
1) devising a task completion strategy,
2) creating a prompt template,
3) deciding if the responses are satisfactory,
4) refining responses.
Safranek et al.[4] provided a loop framework to improve LLM prompts for clinical applications, including:
1) designing a draft of instructional prompts,
2) running clinical notes with the prompt,
3) classifying and counting LLM response errors vs. physician assessments,
4) qualitatively evaluating responses with low accuracy.

However, the iterative nature of prompting is often undocumented and underreported, with only final prompts and outcomes reported. This limits reproducibility and obscures the strategies behind successful outputs. Capturing the evolution of prompts can reveal strategy shifts, refinements, or misunderstandings that influence model behavior and output quality, providing deeper insights into how model behavior and results evolve over time. It will also inform the bias, robustness, and sensitivity of LLMs regarding different prompts.

METHODS

PromptLog is implemented as a JavaScript-based Chrome Extension that collects prompt histories across LLMs' web interfaces (e.g., ChatGPT, Claude). PromptLog operates through three modular components: (1) client-side collection that stores raw prompts in an indexed array with contextual metadata when researchers request, (2) a Python backend using Stanford NLP package[5] to analyze prompt changes through lexical (e.g., word and character counts), semantic (topics and semantic similarities), and structural (e.g., deletion and addition) analysis. Then, a summary of prompt content differences will be generated using the Gemini API for further analysis.

FUTURE WORK

In our future work, we will further collect researchers' needs and their feedback through interviews. PromptLog will be disseminated as open source; we will refine it through workshops and organized discussions across research communities and explore collecting corresponding LLM output.

REFERENCES

1. Sandmann S, Riepenhausen S, Plagwitz L, Varghese J. Systematic analysis of ChatGPT, Google search and Llama 2 for clinical decision support tasks. Nature Communications. 2024 Mar 6;15(1):2050.

2. Chuang YS, Jiang X, Lee CT, Brandon R, Tran D, Tokede O, Walji MF. Use GPT-J prompt generation with RoBERTa for NER models on diagnosis extraction of periodontal diagnosis from electronic dental records. In AMIA Annual Symposium Proceedings 2024 Jan 11 (Vol. 2023, p. 904).

3. Khalid MT, Witmer AP. Prompt Engineering for Large Language Model-assisted Inductive Thematic Analysis. arXiv preprint arXiv:2503.22978. 2025 Mar 29.

4. Safranek CW, Huang T, Wright DS, Wright CX, Socrates V, Sangal RB, Iscoe M, Chartash D, Taylor RA. Automated HEART score determination via ChatGPT: Honing a framework for iterative prompt development. Journal of the American College of Emergency Physicians Open. 2024 Apr;5(2):e13133.

5. Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*. 2020 Mar 16.

ACKNOWLEDGEMENTS

This project has been funded in part by NSF IIS #2144880.

CONTACT

Lu Wang (Presenter): lwang97@stevens.edu
Jina Huh-Yoo: jhuhyoo@stevens.edu