Prompt Engineering for the Use of Generative Artificial Intelligence (AI) in Health Economic Modeling: Findings From a Targeted Literature Review

O'Grady M[1], Adair N[1], Arguello R[1], Benner J[1]

[1]Stratevi LLC, Boston, MA, USA

MSR63

Background

- The release of OpenAI's ChatGPT 3.5 served as a catalyst for the rapid advancement of AI and large language models (LLMs), showcasing an unprecedented expansion in AI capabilities.
- Effective prompting techniques are needed to optimize LLM performance. Prompt engineering is the process of structuring requests and instructions to interact with generative AI models.
- Prompting techniques have already been evaluated across many research applications, but not within a health economics (HE) context.
- This study's objective was to identify the types of prompting techniques (1) available for research, and (2) used in health economic modeling.

Methods

- A targeted literature search was conducted using the arXiv database from its origin (1991) through 14/6/2024.
- Search terms were related to AI, prompt engineering and common prompting techniques, and programming (**Table 1**).
- The search was conducted in three steps: (1) title and abstract screening, (2) full-text review, and (3) citation review.
- Only studies that explicitly examined prompt engineering techniques were included.

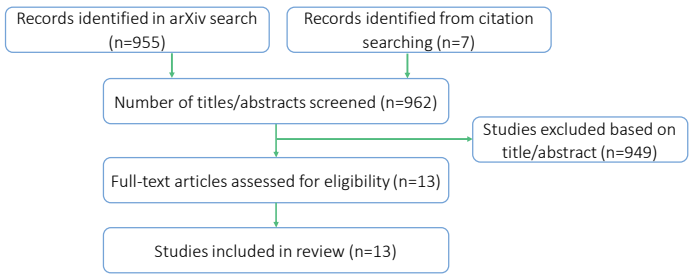
Table 1. Search Terms

"artificial intelligence" OR "AI" OR "large language model" OR "LLM" OR "GPT" OR "generative pre-trained transformer"
prompt* OR "chain-of-thought" OR "least-to-most" OR reason*
"code development" OR "complex tasks" OR "code replication" OR "code generation" OR "model replication"
economic*

Results

- 13 relevant papers were identified (**Figure 1**). Three reviews summarized over 30 unique prompting techniques. Ten additional studies evaluated individual prompting techniques.

Figure 1. Literature Search Results



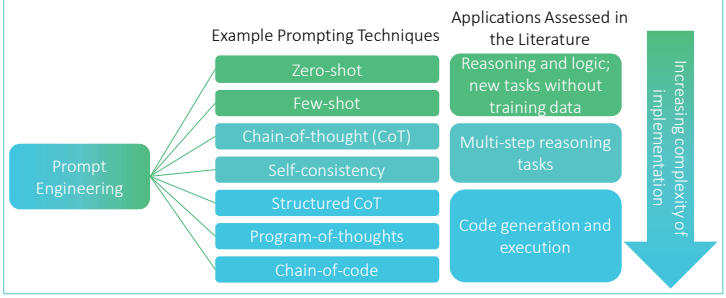
Prompting Techniques

- Several prompting techniques & applications were found in the literature (**Table 2** and **Figure 2**).
  - Of the 30+ techniques described in the literature, few-shot, chain-of-thought (CoT), and self-consistency have been widely shown to enhance response quality compared to zero-shot prompting across a range of applications.
  - Variations of the CoT approach, including structured CoT (SCoT), program-of-thoughts (PoT), and chain-of-code (CoC), have been studied in programming applications and require a higher level of programming knowledge and/or the use of external language interpreters.
- No prompting techniques have specifically been evaluated in a HE modeling context.

Table 2. Select Prompting Techniques Identified in the Literature

Technique	Description	Studies
Zero-shot	The LLM is provided with a natural language prompt and no additional examples. Zero-shot techniques are often combined with another concept (e.g., chain-of-thought).	Chen 2023, Kojima 2022, Sahoo 2024, Schulhoff 2024
One-shot, Few-shot	The LLM is provided with one or more input-output examples prior to the desired query.	Brown 2020, Chen 2023, Sahoo 2024, Schulhoff 2024
Chain-of-thought (CoT)	The LLM is provided with additional instructions, such as "let's think step-by-step" (zero shot), or examples of intermediate reasoning steps (few shot), to encourage step-by-step reasoning.	Chen 2023, Kojima 2022, Sahoo 2024, Schulhoff 2024, Wang 2023, Wei 2022
Self-consistency	The LLM is prompted to generate multiple responses, and the most frequent result is selected as the final response.	Chen 2023, Schulhoff 2024, Wang 2023
Structured chain-of-thought (SCoT)	The LLM is guided to generate reasoning steps using program structures (i.e., sequence, branch, loop) for use in code generation.	Li 2023, Sahoo 2024
Program-of-thoughts (PoT)	The LLM is guided to generate programming code as reasoning steps, which are then executed by a code interpreter to derive the final answer.	Sahoo 2024, Schulhoff 2024
Chain-of-code (CoC)	The LLM is guided to generate code or pseudocode as reasoning steps, and then runs the code using an interpreter or emulator, which can detect errors.	Sahoo 2024

Figure 2. Select Prompting Techniques and Applications Assessed in the Literature



Health Economic Applications

- While no prompting techniques have been evaluated in a HE modeling context, two relevant publications (Reason et al. 2023 and Poirrier & Bergemann 2024) addressed the feasibility of developing cost-effectiveness models using generative AI.
- While outside the scope of the planned targeted review, these studies highlight the potential applications of generative AI to HE.
- Brief descriptions of the objectives, methods and findings are presented in **Table 3**.

Table 3. HE Applications

Author, Year	Objective	LLM	Language	Methods	Key Findings
Reason 2023	To assess whether GPT-4 could be used to automatically program two previously published CE analyses	GPT-4	R	• Authors communicated with GPT-4 via an API. • GPT-4 was iteratively prompted to generate separate sections of R script based on model components. • Each model was generated 15 times, for which programming errors and CE result accuracy were assessed.	• 87% (13/15) - 100% (15/15) of the models were fully replicated error-free or containing a single minor error. • Error-free AI-generated models replicated the published incremental cost-effectiveness ratios to within 1%.
Poirrier & Bergemann 2024	To test how health economists can utilize Microsoft Copilot in Excel-based CE model development	Microsoft Copilot	Excel/ VBA	• Copilot was prompted, first with general prompts and then with step-by-step requests, to generate VBA code. • Copilot output was evaluated by an experienced VBA developer.	• Considerations for developing HE model components were discussed with limitations on implementing a probabilistic sensitivity analysis. • Code quality and usefulness is dependent on the user's detailed request and experience level.

API: Application Programming Interface, CE: Cost-effectiveness, GPT-4: Generative Pre-trained Transformer 4, HE: Health Economics, VBA: Visual Basic for Applications

Discussion

Summary of Findings

- Over 30 unique prompting techniques have been assessed for the purpose of optimizing the response quality of LLMs. These methods been evaluated in a variety of applications, including logical reasoning and code generation.
- A selection of methods, including SCoT, PoT, and CoC, have demonstrated success in improving response quality for application to code generation, but required a higher level of programming knowledge and/or use of external software compared to more basic methods.
- While there is a paucity of studies evaluating and comparing prompting techniques in HE applications, research has demonstrated that LLMs can be used to generate code for HE models.

Implications

- Future research should evaluate prompting techniques within HE applications, including model development.
- Researchers are encouraged to specify prompting techniques utilized in LLM HE applications.

Conclusions

- There is a growing body of literature examining prompting techniques to enhance the capabilities of LLMs. However, this search highlights a gap in HE research specifically focused on prompt engineering. Techniques such as few-shot and chain-of-thought prompting are generalizable and require minimal user expertise, making them excellent candidates for novel applications.
- AI's role in evidence generation and HE modeling is expected to grow – there is a clear need for targeted exploration of how prompting techniques can be optimized to support HEOR applications.

References

1. Brown et al. 2020; 2. Cresswell et al. 2022; 3. Cresswell et al. 2022; 4. Kojima et al. 2023; 5. Li et al. 2023; 6. Michaelson et al. 2024; 7. Wang et al. 2023; 8. Wei et al. 2022; 9. Yang et al. 2024; 10. Zhou et al. 2023; 11. Sahoo et al. 2024; 12. Schulhoff et al. 2024; 13. Chen et al. 2023; 14. Reason et al. 2023; 15. Poirrier & Bergemann 2024. Full citations are available upon request (jennifer@stratevi.com).

All authors are employed by Stratevi, LLC