RESEARCH ARTICLE OPEN ACCESS

A Thorough Analysis of Prompt Engineering Methods for Large Language Models (LLMs)

Dhwani Hirani, Nisha Vadodariya

[1](B Tech in Computer Engineering, Atmiya University, Rajkot, India
Email: dhwanihirani1530@gmail.com)
[2](Faculty of Engineering and Technology (CE), Atmiya University, Rajkot, India
Email: nisha.vadodariya@atmiyauni.ac.in)

Abstract:

A key technique for getting dependable and superior behavior out of Large Language Models (LLMs) is prompt engineering. The evolution of prompt engineering techniques from basic zero-shot and few-shot prompts to sophisticated reasoning frameworks like Chain-of-Thought (CoT) and Tree-of-Thought (ToT) is thoroughly reviewed in this paper. We divide methods into four categories: task-specific strategies, optimization and automated prompting, retrieval and knowledge-enhanced prompting, and reasoning-oriented prompting. We analyze applications across text classification, question answering, mathematical problem solving, commonsense reasoning, and code generation. Finally, we discuss challenges including hallucination, bias, scalability, and interpretability, and outline promising future directions such as meta-prompting.

Keywords: Prompt engineering; Large Language Models; Chain-of-Thought; Retrieval-Augmented Generation; Prompt Optimization

I. Introduction

The field of artificial intelligence (AI) has changed in recent years due to the rapid development of Large Language Models (LLMs). Unlike traditional machine learning models trained for specific tasks, LLMs show emergent capabilities across a variety of domains without the need for explicit fine-tuning. OpenAI's GPT family, Google's PaLM, Meta's LLaMA, and Mistral are examples of LLMs that have been trained on trillions of tokens to capture vast amounts of linguistic, semantic, and contextual knowledge. These models are foundational in modern AI applications because they are excellent at document summarization, translation, mathematical reasoning, commonsense question answering, dialogue systems, and code generation. They are essential to modern AI applications because of this.

One interesting thing to note, though, is that the input prompt has a big effect on the quality of the outputs. Small changes in wording, structure, or formatting can often have huge effects on the results, sometimes making them more accurate and sometimes adding mistakes or biases. Because of this, the field of Prompt Engineering was created, which views the creation of prompts as both a scientific method and an artistic practice.They are essential to modern AI applications because of this. One interesting thing to note, though, is that the input prompt has a big effect on the quality of the outputs. Small changes in wording, structure, or formatting can often have huge effects on the results, sometimes making them more accurate and sometimes adding mistakes or biases. Because of this, the field of Prompt Engineering was created, which views the creation of prompts as both a scientific method and an artistic practice.

Quick engineering tries to answer important questions: How should instructions be written so that they work best? What kinds of structures make you think? How can prompts be automatically made better for different models and tasks? Early research showed that zero-shot prompting (Radford et al., 2019) and few-shot prompting (Brown et al., 2020) could be useful. More advanced methods, like Chain-of-Thought (Wei et al., 2022) and Tree-of-Thought (Yao et al., 2023), showed how important structured reasoning is. At the same time, retrieval-based methods and optimization frameworks have made prompting strategies more flexible.

Several surveys have tried to put these changes into

groups. Vatsal and Dubey (2024) talked about task-specific methods, Kwon et al. (2024) talked about stability-driven frameworks, and Sahoo et al. (2025) talked about an application-oriented taxonomy. Still, these are still broken up, with each one focusing on a different part.

This paper fills in that gap by giving an in-depth look at prompt engineering for LLMs. In particular, it:

II. Review of the Literature and Related Work

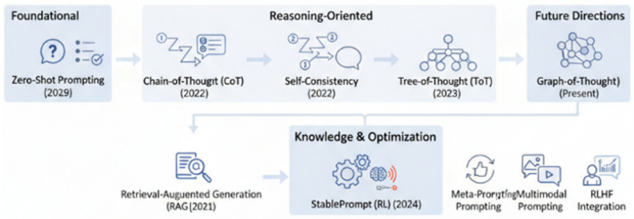
Zero-shot prompting (Radford et al., 2019) is where prompting got its start. It showed that models that had already been trained could do tasks they had not seen before with just natural language instructions. Next came few-shot prompting (Brown et al., 2020), in which adding a few labeled examples to the prompt made it easier to do well on tests like GLUE and SuperGLUE.

Structured prompting strategies were introduced by later research. Wei et al.'s Chain-of-Thought (Wei et al., 2022) told models to show their thinking in between steps, which made them much better at math and symbolic tasks like GSM8K. Self-Consistency (Wang et al., 2022) made things more stable by combining different ways of thinking, and Tree-of-Thought (Yao et al., 2023) turned thinking into tree-structured exploration. These days, Graph-of-Thought (GoT) ,which uses connected nodes to mimic how humans think, has been suggested.

Along with these, methods based on retrieval and optimization have also come up. The outputs of Retrieval-Augmented Generation (Shuster et al., 2021) are based on external documents, and StablePrompt (Kwon et al., 2024) uses reinforcement learning to make automated prompt optimization more stable.

Several surveys give useful but limited views: Sahoo et al. (2025) grouped prompt engineering techniques by where they could be used; Vatsal & Dubey (2024) looked at tasks from a task-specific point of view; and Kwon et al. (2024) presented a reinforcement learning framework for prompt stability. These surveys underscore the need for a unified and comprehensive review, which this paper aims to provide.

Figure1: Evolution of Prompt Engineering Techniques for LLMs



III. Prompt Engineering Techniques

3.1 Basic Prompting

Zero-shot Prompting: Uses only task descriptions. Example: "Translate the following sentence into French." Effective for simple tasks but unreliable for complex reasoning.

Few-shot Prompting: Embeds task-specific examples within the prompt. Demonstrated strong results in GPT-3 (Brown et al., 2020).

3.2 Reasoning-Oriented Prompting

Chain-of-Thought (CoT): Generates intermediate reasoning steps. Highly effective in math and logic benchmarks (e.g., GSM8K).

Self-Consistency: Samples multiple CoT outputs and selects majority consensus, reducing randomness.

Tree-of-Thought (ToT): Explores multiple reasoning paths in a tree structure.
Graph-of-Thought (GoT): Extends ToT into graph structures, enabling richer reasoning.

3.3 Retrieval and Knowledge-Enhanced Prompting

Retrieval-Augmented Generation (RAG): Incorporates relevant documents for grounding, useful in QA and fact-checking.

3.4 Optimization and Automated Prompting

Automatic Prompt Engineer (APE): Employs LLMs to optimize prompts automatically.

Stable Prompt (Kwon et al., 2024): Uses reinforcement learning to stabilize optimization.

IV.Uses for Prompt Engineering

Numerous applications have been impacted by prompt engineering:

Text classification includes intent detection, topic classification, and sentiment analysis (SST-2).

- Question answering: multi-hop reasoning (HotpotQA) and open-domain QA (Natural Questions, TriviaQA).

Mathematical Problem Solving: CoT and Self-Consistency perform exceptionally well on MATH and GSM8K benchmarks.

Commonsense Reasoning: CoT enhances performance on tasks such as the Winograd Schema Challenge.

- Code Generation: On the HumanEval and MBPP datasets, PoT and CoC perform better than baseline prompting.

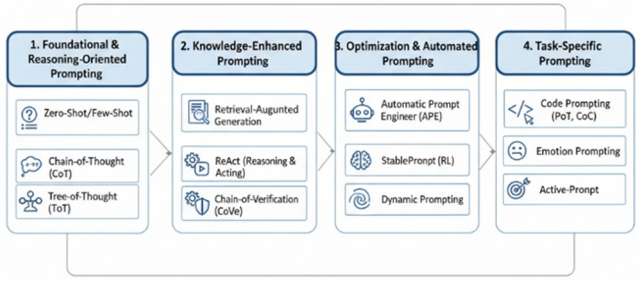
Emotion prompting in dialogue systems improves user satisfaction and personalization.

V.Obstacles and Restrictions

Even though prompt engineering holds promise, there are still a number of enduring issues:

Hallucination: Despite structured prompting, LLMs continue to generate outputs that are factually incorrect.

Figure 2: Taxonomy of Prompt Engineering Methods for LLMS



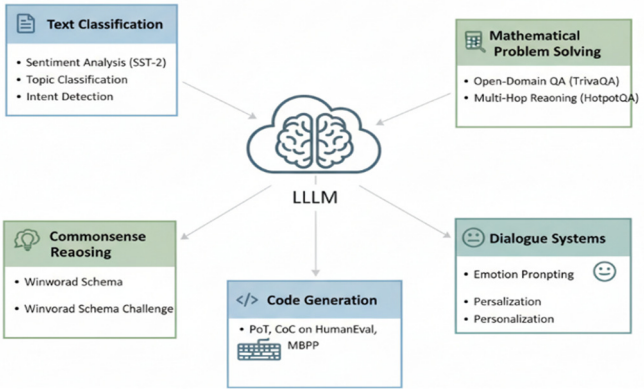
Fairness and Bias: Inadvertently, prompts may make biases in training corpora more pronounced.

Scalability: Prolonged prompts use tokens, which lowers deployment efficiency on a large scale.

Generalization: Often, optimized prompts don't work across domains.

Interpretability: Little is known about the mechanisms underlying prompt effectiveness.

Figure 3: Applications a Engineering Across Domains



Prospects for the Future

The following are promising avenues for further research:

Prompts that assist LLMs in creating and improving their own prompts are known as meta-prompting.

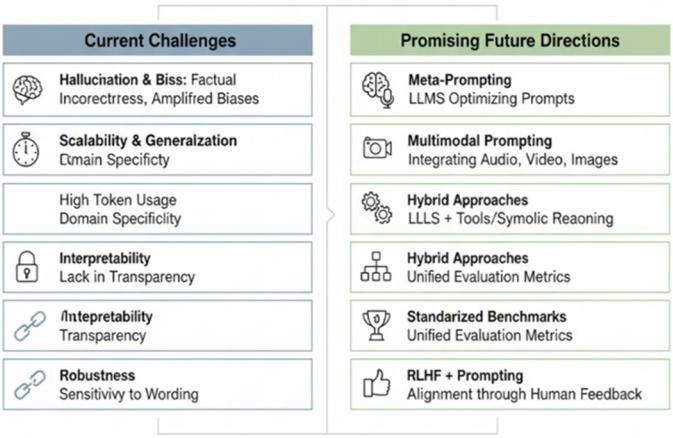
Including audio, video, and images in models such as GPT-4V is known as multimodal prompting.

Hybrid Approaches: Using LLM prompting in conjunction with external tools and symbolic reasoning.

Standardized Benchmarks: Unified metrics are required in addition to GLUE and Super GLUE.

RLHF + Prompting: Alignment through the Integration of Reinforcement Learning and Human Feedback.

Figure 4: Challenges & Future Directions in Prompt Engineering



VI. Conclusion

From heuristic prompt crafting, prompt engineering has developed into a formal scientific field. From few-shot and zero-shot prompting to sophisticated reasoning techniques like CoT and ToT, as well as optimization techniques like The persistence of hallucinations, bias, and scalability highlights the necessity for ongoing innovation. Future developments in the field are anticipated to include multimodal prompting, meta-prompting, and RLHF integration.

This work provides a thorough reference for scholars and practitioners alike by combining various viewpoints into a single framework, laying the groundwork for further prompt engineering research. Facilitating automated vulnerability exploitation, the creation of elusive malware, and hyper- realistic social engineering.

At the same time, it is automating incident response, transforming threat intelligence, giving defenders proactive and adaptive capabilities, and making it possible to build strong defensive models using synthetic data. This technological dualism has led to a high- stakes, reciprocally evolving arms race that is changing the strategic balance of power in cyberwarfare. It might increase threats against the society.

The future requires a two-pronged strategy: a significant investment in AI-native defenses to keep up with evolving threats, and the concurrent development of robust governance and AI safety protocols to manage the inherent risks of the technology.

As we move into an era of increasingly autonomous cyberspace, cybersecurity's future will depend on our ability to provide the critical strategic oversight and moral guidance needed to manage the complex interactions between artificial and human intelligence in the defense of our digital world.

References

1. A. Radford and colleagues (2019). Unsupervised multitask learners are language models.
2. T. Brown and associates (2020). Few-Shot Learners are language models.
3. J. Wei and colleagues (2022). Reasoning in Large Language Models is Induced by Chain-of-Thought Prompting.
4. X. Wang and associates (2022). Self-Consistency Enhances Language Models' Chain of Thought Reasoning.
5. Yao and colleagues (2023). Tree-of-Thoughts: Deliberate Problem Solving with Large Language Models.
6. K. Shuster and associates (2021). Knowledge-Intensive NLP through Retrieval-Augmented Generation

