International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 11 | Nov - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Exploring Techniques and Applications of Prompt Engineering for Large Language Models

L. Arleen Felicien[1], Mrs. Nivetha Mani[2], Mrs. Princy Medona[3]
[1] *DepartmDepartment of Computer Science, Idhaya College of Arts and Science for Women*
[2] *Department of Computer Science, Idhaya College of Arts and Science for Women*
[3] *Department of Computer Science, Idhaya College of Arts and Science for Womenent & College*

---***---

Abstract - Prompt engineering has become an effective approach for improving the performance of large language models (LLMs) and vision–language models (VLMs) without modifying their internal architecture. By designing precise instructions, structured formats, and illustrative examples, prompts guide models to interpret tasks more accurately and generate more reliable outputs. Despite its rapid growth and significant impact on areas such as question answering, reasoning, and multimodal tasks, the field still lacks a well-organized overview of its diverse techniques. This paper presents a structured survey of recent advances in prompt engineering, explaining how each technique functions, the tasks it supports, the models and datasets used for evaluation, and the benefits and limitations observed. Visual summaries, including tables and diagrams, are provided to enhance understanding. Overall, this review offers a comprehensive introduction to prompt engineering and outlines key challenges and promising directions for future research.

Key Words: Prompt Engineering, LLMs, VLMs, Reasoning Prompts, NLP.

1. INTRODUCTION

Prompt engineering has emerged as a crucial technique for enhancing the capabilities of pre-trained large language models (LLMs) and vision-language models (VLMs). It involves strategically designing task-specific instructions, referred to as prompts, to guide model output without altering parameters. The significance of prompt engineering is especially evident in its transformative impact on the adaptability of LLMs and outputs through carefully crafted instructions, prompt engineering enables these models to excel across diverse tasks and domains. This adaptability is different from traditional paradigms, where model retraining or extensive fine-tuning is often required for task-specific performance. This is the transformative promise of prompt engineering, pushing the boundaries of AI and opening doors to a future brimming with possibilities. In an ever-evolving landscape, ongoing research consistently reveals innovative approaches and applications within prompt engineering. The significance of prompt engineering is underscored by its capacity to steer model responses, enhancing the adaptability and applicability of LLMs across diverse sectors.

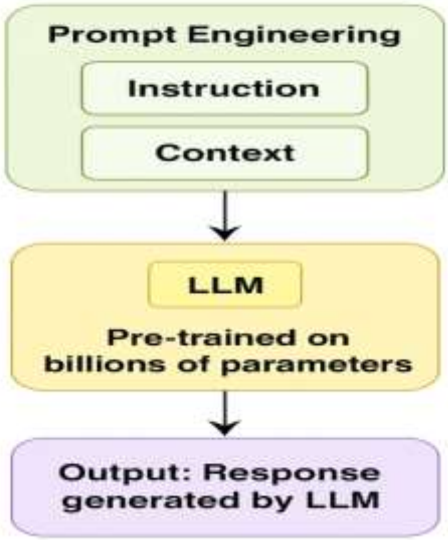

Figure 1: Visual breakdown of prompt engineering components

The landscape of contemporary prompt engineering spans a spectrum of techniques, encompassing foundational methods like zero-shot and few-shot prompting to more intricate approaches such as "chain of code" prompting. The notion of prompt engineering was initially investigated and popularized in the LLMs [Liu et al., 2023], [Tonmoy et al., 2024], [Chen et al., 2023] later extended to VLMs [Wu et al., 2023], [Bahng et al., 2022]. Despite the extensive literature on prompt engineering within both LLMs and VLMs, a notable gap remains, particularly concerning a systematic overview of application-centric prompt engineering techniques. With recent strides in prompt engineering, there is a pressing need for a comprehensive survey that offers a nuanced understanding of applications and advancements in contemporary research. This survey dives deep into the ever-evolving landscape of prompt engineering, analyzing over 41 distinct techniques categorized by their diverse applications. Employing a systematic review approach, we meticulously delve into the intricacies 2.2 Reasoning and Logic of diverse cutting-edge prompting methods. Our examination encompasses their applications, the language models utilized, and the datasets subjected to experimentation, providing a detailed and nuanced analysis of the evolving landscape of prompt engineering. Additionally, we discuss the pros and cons of these techniques, offering insights into their comparative efficacy. We present a comprehensive taxonomy diagram that illustrates how these techniques navigate the vast landscape of LLM capabilities (see Fig.2) and provide a table summarizing the datasets, employed models, and evaluation metrics (see Table1). From language generation and question answering to code creation and reasoning tasks, prompt engineering empowers the LLMs into performing feats we never thought possible.

By bridging the existing gap in the literature, this survey aims to serve as a valuable resource for researchers and practitioners, offering insights into the latest developments and facilitating a deeper understanding of the evolving landscape of prompt engineering. The structure of the paper is organized as follows: Section 2 presents the prompt engineering techniques from both basic to advance by categorizing application-area and Section 3 provides a conclusion along with considerations for future research endeavors.

2. Foundational Methodologies in Prompt Engineering

Prompt engineering encompasses a diverse range of methodologies designed to enhance the interaction between users and large language models (LLMs). These methods range from simple instructional patterns to more structured frameworks that help models interpret tasks with greater clarity, consistency, and contextual awareness. By organizing these into foundational methodologies, this section provides a structured understanding of how prompts influence model behavior. We highlight the principles behind these approaches, the tasks they support, and their role in improving performance across various applications. The following subsections examine these methodologies in detail.

2.1 Essential Prompting Techniques

Essential prompting techniques form the starting point for designing effective interactions with LLMs. They rely completely on natural-language cues provided within the prompt and do not require any form of model fine-tuning or retraining. Even though these techniques are simple, they play a crucial role in improving task performance, reasoning clarity, and output consistency across many domains.

2.2 Reasoning-Centered Prompt Design

Reasoning-centered prompting techniques represent a major advancement in guiding large language models toward more reliable, structured, and logically grounded outputs. Unlike basic prompting methods that simply instruct a model on "what" to produce, reasoning-oriented strategies influence how the model thinks through a problem.

These approaches encourage stepwise analysis, exploration of alternatives, integration of structured logic, and self-evaluation. By shaping the internal reasoning trajectory, such methods reduce superficial or pattern-based responses and help the model generate solutions that better reflect human-like problem-solving behavior.

2.3 Guided Analytical Processing

Guided analytical prompting methods are designed to walk the model through a clearly defined sequence of reasoning operations. Instead of generating a direct answer, the model is instructed to break down the problem into interpretable parts, analyze each segment, and gradually work toward a conclusion. This guided breakdown enhances clarity, reduces ambiguous leaps in reasoning, and gives the model an opportunity to process multi-step tasks more systematically. These approaches are particularly effective for mathematics, logical puzzles, causal reasoning, and multi-hop question answering.

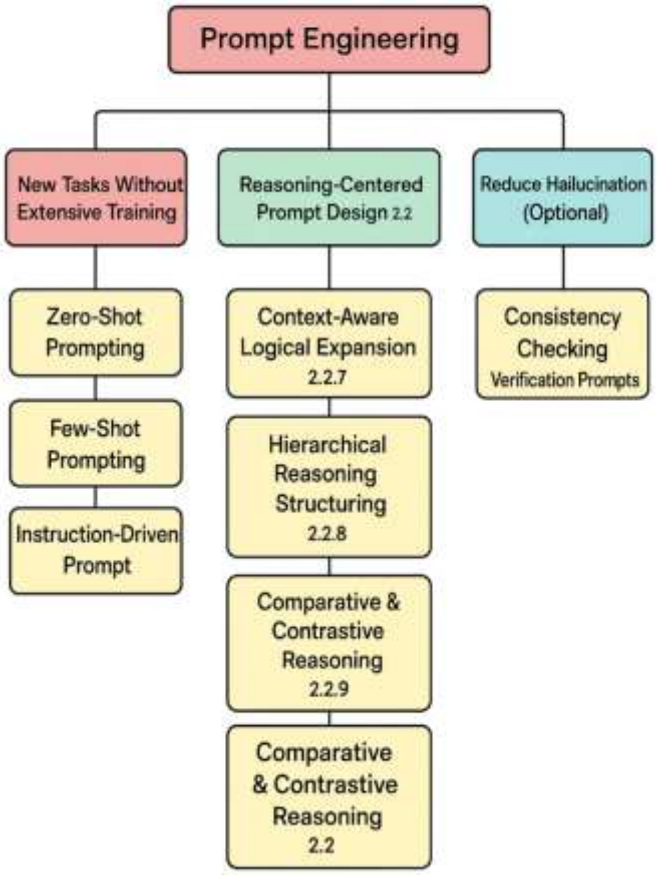

Figure 2. Taxonomy of LLM

By structuring the reasoning path, these prompts help the model maintain coherence across steps, strengthen internal consistency, and provide transparent outputs that allow users to trace how the final answer was reached.

2.4 Automated Reasoning Pattern

Creating high-quality reasoning demonstrations manually can be challenging, and human-created examples do not always scale well across tasks. Automated reasoning pattern generation addresses this limitation by letting the model itself produce reasoning templates. In this approach, the model first generates preliminary reasoning samples for a task, then examines, filters, and refines them to construct stronger exemplars. These self-generated templates can be refused similar problems, significantly decreasing manual effort. This method also fosters diversity in reasoning patterns—enabling the model to learn multiple ways of approaching a task rather than relying on a single, rigid structure. As a result, the LLM gains a flexible set of reasoning styles that can adapt to task complexity, data variations, and domain-specific constraints.

2.5 Multi-Path Exploration and Selection

Many reasoning problems allow for more than one valid approach. Instead of forcing the model to follow a single chain of reasoning, multi-path prompting encourages it to explore several possible solution routes. These strategies prompt the model to generate multiple intermediate reasoning paths, each representing a different interpretative angle. The model then compares the outcomes, identifies conflicts or inconsistencies, and selects the most reliable answer among them.

This multi-path approach reduces the risk of the model becoming "locked" into an incorrect early assumption. It also promotes richer inferential diversity, making it suitable for tasks such as planning, creative reasoning, and open-domain question answering. Ultimately, the model benefits by balancing exploration with logical consolidation, resulting in more accurate and trustworthy outputs.

2.6 Structured Logic Integration

Some problems require more precise reasoning than natural language alone can provide. Structured logic integration incorporates symbolic representations—such as tables, diagrams, rules, or condensed notations—directly into the prompting process.These structures minimize ambiguity and force the model to operate within clear, predefined boundaries. For example, tabular reasoning can improve data interpretation, spatial diagrams can enhance relational understanding, and symbolic shorthand can aid in computational tasks. By blending natural language with structured constraints, this prompting category enables the model to reason in a manner that is both interpretable and aligned with formal logic. It is especially valuable in fields where accuracy, traceability, and structure are essential, such as mathematics, data analysis, law, and scientific reasoning.

LLMs often produce answers that appear fluent but may contain inaccuracies or logical gaps. Reflective reasoning techniques address this by prompting the model to evaluate its own responses.

2.7 Reflective and Self-Improving Reasoning

The process typically involves three phases: generating an initial attempt critiquing or reviewing the output, and refiningthe answer. This mimics a human revision cycle and encourages the model to identify errors, improve clarity, and reinforce logical rigor.

Such iterative design has been effective in long-form generation, code refinement, argument evaluation, and tasks requiring multi-layered reasoning. By integrating self-feedback loops, these prompts help models adjust their reasoning pattern over repeated iterations, producing more accurate and well-structured results.

2.8 Code-Assisted Logical Inference

Certain reasoning tasks—especially those involving symbolic operations, conditional logic, or numerical calculations—benefit from representations closer to programming logic than natural language.

Code-assisted prompting reframes the problem using code-like structures such as pseudo-code, conditional blocks, or simple executable logic. This approach encourages the model to reason using sequential operations, variable tracking, and strict rule-based flows.

Because code syntax imposes discipline and reduces ambiguity, the model is less likely to generate inconsistent reasoning or overlook important steps. This method is particularly effective in algorithmic thinking, quantitative reasoning, data manipulation, and rule-based decision processes. Instruction-based prompting focuses on giving explicit, structured directions that guide the model toward the desired response. Clear and specific instructions reduce ambiguity and improve consistency, especially in instruction-tuned models. This method has become central to many modern prompting applications, forming the basis for tasks in reasoning, content generation, and problem-solving.

2.9 Context-Aware Logical Expansion

Context-aware logical expansion focuses on enhancing the model's capacity to interpret subtle cues, background information, and implied relationships within a task. Instead of relying solely on surface-level instructions, these prompts encourage the model to extract hidden dependencies, infer missing information, and expand the context into logical components that support reasoning.

This approach is especially effective in tasks where the prompt contains layered meaning—such as story-based questions, situational reasoning, or long-form decision-making. By teaching the model to identify deeper contextual links, this technique significantly improves coherence, reduces misinterpretation, and strengthens the logical grounding of the final output.

2.10 Hierarchical Reasoning Structuring

Hierarchical reasoning structuring divides complex problems into multiple levels or tiers of reasoning, allowing the model to progress gradually from broad concepts to fine-grained details. The prompt guides the model to first identify high-level categories, relationships, or themes before addressing lower-level specifics. This top-down analytical flow helps maintain clarity and prevents the model from getting overwhelmed by intricate details too early.

Such hierarchical decomposition is particularly useful in legal reasoning, scientific explanation, long-chain mathematics, and multi-criteria decision processes. By organizing reasoning into tiers, the model produces outputs that are easier to interpret, structurally consistent, and logically layered.

2.11 Comparative and Contrastive Reasoning Prompts

Comparative reasoning prompts instruct the model to evaluate multiple options, hypotheses, or viewpoints side by side. Instead of generating a single answer, the model is encouraged to analyze differences, similarities, strengths, and weaknesses across possible solutions. This contrastive structure helps the model develop a more discriminative understanding of the problem, reducing the chance of accepting the first generated answer as correct. It is particularly effective in classification, diagnostic reasoning, argument evaluation, and analytical writing tasks. By comparing alternatives, the model forms deeper insights and arrives at conclusions supported by clear justification.

3. CONCLUSIONS

Prompt engineering is emerging as a powerful method to enhance large language models without altering their internal structure. This survey explores both foundational strategies—such as zero-shot, few-shot, and instruction-based prompting—and advanced reasoning-focused techniques that help models think more systematically and produce reliable outputs. Well-designed prompts guide model behavior, improve alignment with tasks, and strengthen response quality. As LLMs and VLMs continue to advance, the role of prompt engineering becomes increasingly significant. The study highlights that prompting is not just about formatting responses but about shaping how models interpret, reason, and solve problems. Despite progress, challenges like hallucinations, inconsistency, and limited reasoning generalization remain. Addressing these issues opens new research directions, including hybrid and self-adaptive prompting methods. This survey provides a structured overview of current practices and future possibilities, offering valuable insights for researchers and developers working with large-scale AI system

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Computer Science, Idhaya College of Arts and Science for Women, for providing the resources and academic support required for the successful completion of this research workshould be treated as a 3rd level heading and should not be assigned a number.

REFERENCES

1. Radford et al. "Language models are unsupervised multitask learners," *OpenAI Technical Report*, 2019.
2. T. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
3. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *arXiv preprint arXiv:2201.11903*, 2022.
4. Z. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," *arXiv:2203.11171*, 2022.
5. R. Zhou et al., "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," *arXiv:2305.10601*, 2023.
6. J. Yao et al., "Graph-of-Thoughts: Solving Complex Reasoning Tasks with Large Language Models," *arXiv:2310.04751*, 2023