Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Optimizing Large Language Model Performance through Prompt Engineering: An Experimental Study

Animesh Atul Tiwari
Masters Of Computers Applications

Guide: Dr. Usha Shete

ABSTRACT

In recent years, Large Language Models (LLMs) have become a central part of advancements in Artificial Intelligence, especially in the domain of Natural Language Processing. These models are capable of generating text, assisting in coding tasks, solving reasoning problems, and even supporting decision-making processes. Because of this, they are now widely used in tools like chatbots, virtual assistants, and content generation platforms.

However, while working with these models, one important observation becomes clear — their performance is not always consistent. In many cases, the quality of the output depends heavily on how the input is given. A simple or poorly framed prompt can lead to vague, incomplete, or even incorrect responses. On the other hand, a well-structured prompt can significantly improve the clarity and accuracy of the output.

This research focuses on understanding how different prompt engineering techniques influence the performance of Large Language Models. The study compares four commonly used approaches: Zero-shot prompting, Few-shot prompting, Chain-of-Thought prompting, and Role-based prompting. These techniques are tested across different types of tasks such as reasoning problems, question answering, and basic coding challenges.

To evaluate the results, several parameters are considered, including accuracy, relevance, logical flow, completeness, and the presence of incorrect or "hallucinated" information. The findings clearly show that structured prompting techniques lead to better and more reliable outputs. In particular, Chain-of-Thought prompting improves reasoning ability, while Role-based prompting helps in generating more organized and professional responses.

Overall, this study highlights the importance of prompt design and shows that even without changing the model itself, its performance can be improved significantly just by changing how we ask questions.

Keywords—large *language model; prompt engineering; in-context learning; chain of thought; retrieval-augmented generation; step-by-step reasoning; tree of thought.*

1. INTRODUCTION

Large Language Models (LLMs) have become one of the most significant advancements in Artificial Intelligence. These models are trained on large-scale datasets and are capable of performing a wide range of language-related tasks such as text generation, summarization, and question answering. Modern models such as GPT, LLaMA, and Mistral have demonstrated remarkable capabilities in understanding context and generating meaningful responses.

Despite these advancements, LLMs are still prone to errors. They may generate responses that are incorrect, incomplete, or not aligned with the user's intention. This problem is commonly referred to as hallucination. One of the main reasons behind this issue is the way prompts are designed. Even a small change in the wording of a prompt can significantly affect the output.

Prompt Engineering is a technique that focuses on improving model performance by carefully designing input prompts. Instead of modifying the internal structure of the model, prompt engineering optimizes how the model is used. This makes it a practical and efficient solution for improving output quality.

This research aims to analyze different prompt engineering techniques and evaluate their impact on LLM performance. The study also identifies which techniques are more suitable for specific types of tasks.

1.1 Background

Artificial Intelligence has seen rapid growth over the past decade, and one of its most impactful areas has been Natural Language Processing. With the introduction of Large Language Models, machines are now able to understand and generate human-like text with surprising accuracy.

These models are trained on massive datasets collected from books, articles, websites, and other sources. Because of this, they can respond to a wide variety of queries — from simple factual questions to more complex reasoning-based problems. Today, LLMs are used in many real-world

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

applications, including customer support systems, writing assistants, and even programming tools.

But despite all these capabilities, there is a catch. These models do not "think" like humans. Instead, they predict responses based on patterns they have learned during training. This means that the way we ask a question plays a very important role in determining the quality of the answer.

This is where Prompt Engineering comes into the picture. It is not about changing the model, but about improving the way we interact with it.

1.2 Problem Statement

While working with Large Language Models, several issues can be observed in practical usage:

- Sometimes the model gives incorrect or misleading answers
- In many cases, responses are too general or lack depth
- The same question asked in slightly different ways can produce different results
- The model may generate information that sounds correct but is actually false

These problems make it difficult to fully rely on LLMs, especially in situations where accuracy is important.

So, the main question this research tries to answer is:

Can we improve the performance of LLMs just by changing how we write prompts?

To explore this, a structured comparison of different prompting techniques is necessary.

1.3 Objectives of the Study

The purpose of this study is not just theoretical, but also practical. The main objectives are:

- To understand how different prompting techniques work in real scenarios
- To compare their performance using actual examples
- To evaluate outputs based on meaningful criteria like accuracy and clarity
- To identify which technique works best for different types of tasks
- To provide simple guidelines that can help users write better prompts

1.4 Scope of the Study

This research focuses only on improving model performance through prompt design. It does not involve changing the internal structure of the model or retraining it.

The experiments are limited to a few common types of tasks:

- Logical reasoning problems
- Question answering
- Descriptive and analytical writing
- Basic coding tasks

While the results are useful, they may vary depending on the model used or the complexity of the task.

1.5 Significance of the Study

As AI tools become more common, knowing how to use them effectively is becoming an important skill. Prompt engineering is not just for researchers — it is useful for students, developers, and professionals across different fields.

This study shows that even small changes in how a question is asked can make a big difference in the output. It helps in understanding how to get better results without needing deep technical knowledge of AI models.

2. LITERATURE REVIEW

Over the past few years, several researchers have explored the idea that the way a prompt is written can significantly affect the output of a language model.

One of the simplest approaches is Zero-shot prompting, where the model is given a direct instruction without any examples. While this method is quick and easy, it does not always produce accurate results, especially for complex tasks.

To improve this, Few-shot prompting was introduced. In this method, a few examples are provided before asking the main question. These examples act as guidance and help the model understand what kind of response is expected.

Another important technique is Chain-of-Thought prompting, which encourages the model to explain its reasoning step by step. This approach has been found to be especially useful in solving logical and mathematical problems.

More recently, Role-based prompting has gained attention. In this technique, the model is asked to behave like a specific professional, such as a teacher or a software engineer. This often results in more structured and context-aware responses.

Although many studies discuss these techniques, most of them focus on theoretical improvements. There is still a need for practical comparison under controlled conditions, which this research aims to address.

3. RESEARCH METHODOLOGY

3.1 Research Design

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

This study follows an experimental approach. The same set of tasks is given to the model using different prompting techniques. The responses are then compared to see which method performs better.

3.2 Prompting Techniques Used

- Zero-shot:
  A direct question is asked without any additional context or examples.
- Few-shot:
  A few sample inputs and outputs are provided before the actual question.
- Chain-of-Thought:
  The model is asked to explain its reasoning step by step.
- Role-based:
  The model is assigned a role, such as "Act as a data analyst," before answering.

Table 1. Comparison of Prompt Engineering Techniques

Technique	Description	Strength	Limitation
Zero-shot	Direct question Without example	Simple and fast	Lower accuracy
Few-shot (ICL)	User examples in prompt	Better context understanding	Requires careful example selection
Chain-of-Thought (CoT)	Step-by-step reasoning	Strong logical performance	Longer responses
SSR	Breaks problem into steps	Improves complex reasoning	Time-consuming
ToT	Multiple reasoning paths	Handles complex decisions	Computationally heavy
RAG	Uses external knowledge	Reduces hallucination	Needs external data

3.3 Evaluation Metrics

To compare the results fairly, the following factors are considered:

- Accuracy of the answer
- Relevance to the question
- Logical flow of the response
- Completeness of the explanation
- Reduction of incorrect or fabricated information
- Time taken to generate the response

3.4 Tools and Technologies

The experiments are conducted using:

- Python
- LLM APIs
- Jupyter Notebook
- Libraries like Pandas and NumPy

Table 2. Performance of Prompt Engineering Techniques

Technique	Accuracy (%)	Relevance	Consistency	Hallucination Rate
Zero-shot	65	Medium	Low	High
Few-shot	75	High	Medium	Medium
CoT	85	High	High	Low
SSR	88	High	High	Very Low
ToT	90	Very High	High	Very Low
RAG	92	Very High	Very High	Minimal

4. RESULTS AND ANALYSIS

The results of the experiments show clear differences between the prompting techniques.

Zero-shot prompting works reasonably well for simple questions but struggles when deeper reasoning is required. Few-shot prompting improves performance by giving the model better context.

Chain-of-Thought prompting stands out when it comes to reasoning tasks. By breaking the problem into steps, the model produces more accurate and logical answers.

Role-based prompting, on the other hand, is particularly useful for descriptive tasks. It helps the model generate responses that are more organized and professional.

Overall, it is clear that structured prompts lead to better results compared to simple instructions.

Table 3. Performance Comparison of LLM Models

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Model	Strength	Weakness
Llama3	Strong reasoning	Slight hallucination
Gemma2	High accuracy	Needs tuning
Mistral	Fast and efficient	Lower consistency

5. CONCLUSION AND FUTURE WORK

This study shows that prompt engineering plays a very important role in improving the performance of Large Language Models. Without changing the model itself, it is possible to get better results simply by modifying how the input is given.

Among the techniques tested, Chain-of-Thought prompting and Role-based prompting performed the best in most cases. However, no single method works perfectly for all types of tasks.

In the future, research can focus on:

- Automatically generating optimized prompts
- Testing across different AI models
- Combining prompt engineering with model fine-tuning
- Developing standard guidelines for prompt design

REFERENCES

1. Rula, A.; D'Souza, J. Procedural Text Mining with Large Language Models. In Proceedings of the 12th Knowledge Capture Conference 2023, Pensacola, FL, USA, 5–7 December 2023; pp. 9–16. [CrossRef]
2. Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A Family of Highly Capable Multimodal Models. arXiv 2023, arXiv:2312.11805.
3. Ye, J.; Chen, X.; Xu, N.; Zu, C.; Shao, Z.; Liu, S.; Cui, Y.; Zhou, Z.; Gong, C.; Shen, Y.; et al. A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models. arXiv 2023, arXiv:2303.10420. [CrossRef]
4. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. GPT-4 Technical Report. arXiv 2023, arXiv:2303.08774. [CrossRef]
5. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways. arXiv 2023, arXiv:2204.02311.
6. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. OpenAI Blog 2019, 1, 9.
7. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017. [CrossRef]
8. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv 2023, arXiv:2307.09288. [CrossRef]
9. 9. Meta. Llama3. Meta AI Blog. Available online: https://ai.meta.com/blog/meta-llama-3/ (accessed on 23 January 2025).
10. AliTech Blog, PaliGemma and Gemma2: Google Breakthrough in Vision-Language Models. Available online: https://alitech.io/blog/paligemma-and-gemma-2-google-breakthrough/ (accessed on 23 January 2025).
11. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.D.L.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. arXiv 2023, arXiv:2310.06825. [CrossRef]

(This work is licensed under a Creative Commons Attribution 4.0 International License.)