Strategies for Creating Prompts: A Comparative Analysis of Few-Shot Learning, Chain-of-Thought, and RAG

Juan Sebastian Rodriguez Trujillo
Cali, Colombia
jsrodriguez.user@gmail.com

***Abstract*— The following article explores prompt engineering strategies in natural language processing, focusing on Few-Shot Learning, Chain-of-Thought, and RAG. Through a comparative analysis, we examine their strengths, limitations, and applications.**

I. INTRODUCTION

This article is an exploration of three key strategies in prompt engineering for natural language processing: Few-Shot Learning, Chain-of-Thought, and RAG. Few-Shot Learning enables models to adapt with minimal examples, Chain-of-Thought enhances reasoning through step-by-step guidance, and RAG integrates external knowledge for dynamic context.

II FEW-SHOT LEARNING

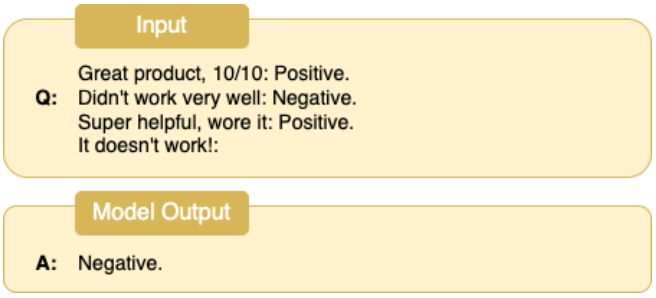

Figure 1.

Few-shot learning is a technique that enables a model to perform a specific task using only a few examples provided in the prompt. In language models, this means including relevant examples in the input so the model can infer the pattern and solve similar tasks.

TABLE I

Advantages	
Flexibility	No need to retrain the model, just provide examples in the prompt.
Resource efficiency	Reduces time and costs associated with training additional models.
Scalability	Can be applied to a wide variety of tasks without modifying the model's architecture.

TABLE II

Disadvantages	
Context length limitations	The number of examples is constrained by the model's maximum context size.
Variable performance	Results depend on the quality and relevance of the examples provided.
Limited generalizability	The model may fail if the task is too different or complex.

III CHAIN-OF-THOUGHT PROMPTING

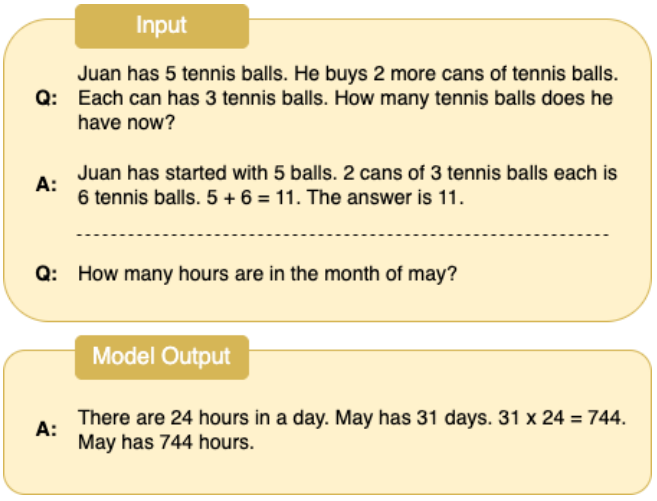

Figure 2.

This technique encourages the model to solve complex problems by breaking them into intermediate reasoning steps before arriving at a final solution. It is used to enhance the model's ability to handle tasks requiring logic or sequential processes.

TABLE III

Advantages	
Improved performance on complex tasks	Increases accuracy in problems requiring logical or mathematical reasoning.
Transparency	Makes the decision-making process easier to interpret.
Task transferability	Improves performance in similar problems requiring structured steps.

TABLE IV

Disadvantages	
Token consumption	Longer prompts can consume more tokens, increasing costs.
Not always necessary	May be overkill for simple or straightforward tasks.
Prompt dependency	Requires well-structured prompts to achieve good results.

IV RETRIEVAL-AUGMENTED GENERATION (RAG)

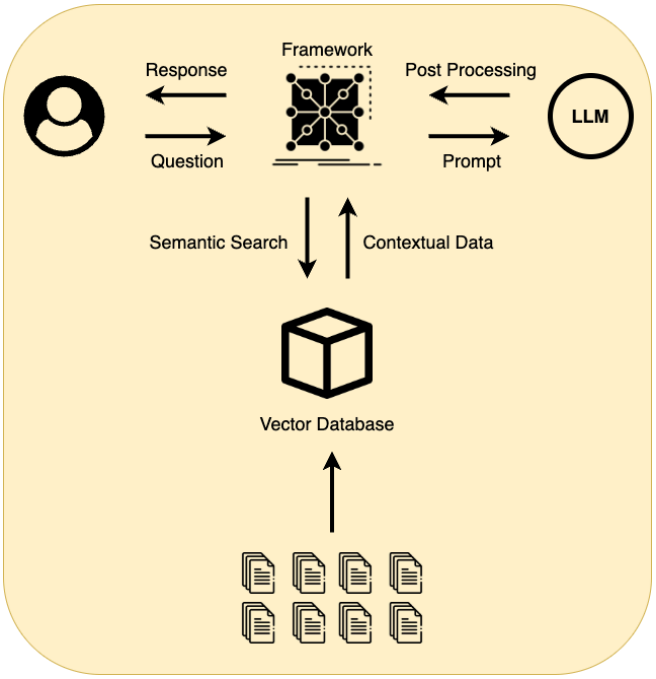

Figure 3.

Retrieval-Augmented Generation combines a generative model with a retrieval component. Before generating a response, the model retrieves relevant data from an external knowledge base or repository and uses it as context.

TABLE V

Advantages	
Access to updated information	Enables answering questions with more recent data than the model's training set.
Reduced model dependency	Relies on external knowledge bases instead of storing all information internally.
Improved accuracy	Combines language generation with precise, retrieved information.

TABLE VI

Disadvantages	
Dependency on retrieval	If the retrieval fails, the response may be incorrect or irrelevant.
Technical complexity	Requires implementation and maintenance of the retrieval system and integration with the generative model.
Latency	Searching for information may introduce delays in response generation.

V COMPARISON

TABLE VII

Key Advantages	**Key Disadvantages**	**Best for...**
Few-shot Learning		
Fast, flexible, no model adjustment required	Limited by context size	General tasks with clear patterns
Chain-of-Thought		
Enhances complex, sequential reasoning	High token consumption	Logical, mathematical, or multi-step problems
Retrieval-Augmented Generation (RAG)		
Accurate, up-to-date information	Technical complexity and latency	Tasks requiring recent or precise information

REFERENCES

[1] **Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications.** Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications. December, 2024.