

Open in app ↗

Medium

 SearchThe Modern Scienti... · [Follow publication](#)

Best Prompt Techniques for Best LLM Responses

7 min read · Feb 12, 2024



Jules S. Damji

[Follow](#)

Listen



Share



More

Better prompts is all you need for better responses

In this article, we explore what is prompt engineering, what constitutes best techniques to engineer a well-structured prompt, and what prompt types steer an LLM to generate the desired response; contrastingly, what prompt types deter an LLM to generate a desired response.

What's Prompt Engineering?

Prompt engineering is not strictly an engineering discipline or precise science, supported by mathematical and scientific foundations; rather it's a practice with a set of guidelines to craft precise, concise, creative wording of text to instruct an LLM to carry out a task. Another way to put it is that "prompt engineering is the art of communicating with a generative large language model." [2]

To communicate with LLM with precise and task-specific instructions prompts, Bsharat et.al [3] present a comprehensive principled instructions and guidelines to improve the quality of prompts for LLMs. The study suggests that because LLMs exhibit an impressive ability and quality in understanding natural language and in carrying out tasks in various domains such as answering questions, mathematical reasoning, code generation, translation and summarization of tasks, etc., a principled approach to curate various prompt types and techniques immensely improves a generated response.

Take this simple example of a prompt before and after applying one of the principles of conciseness and precision in the prompt. The difference in the response suggests both readability and simplicity—and the desired target audience for the response.

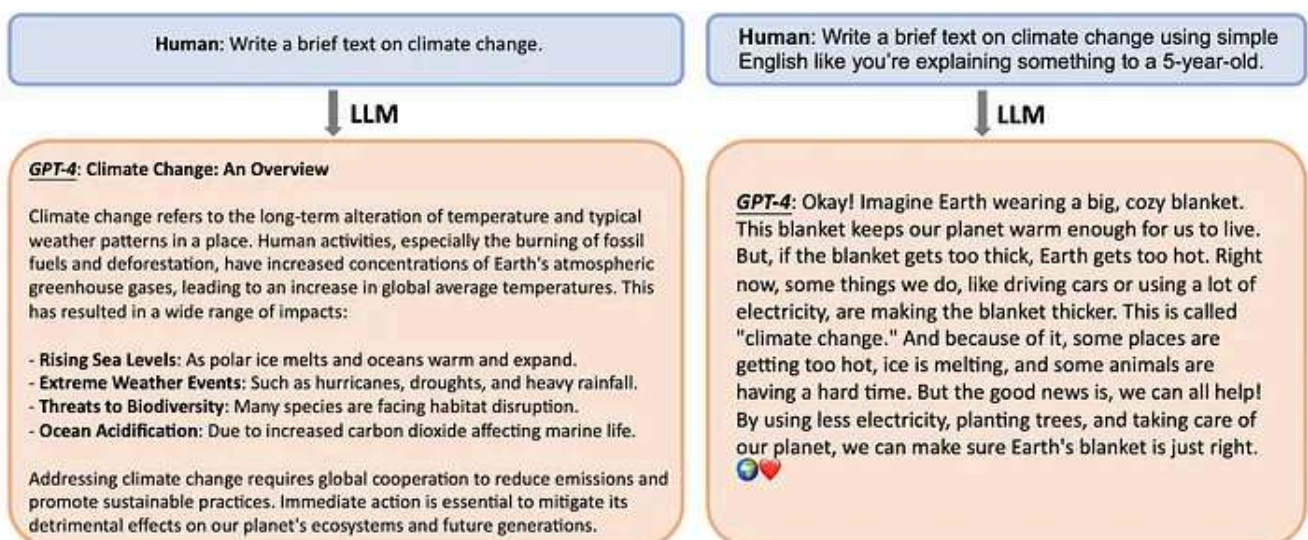


Figure 1: An Illustration showing a general prompt on the left and a precise and specific prompt on the right.

[4]

As you might infer from the above example that the more concise and precise your prompt, the better LLM comprehends the task at hand, and, hence, the better response it formulates.

Let's examine some of the principles, techniques and types of prompt that offer better insight in how to carry out a task in various domains of natural language processing.

Prompt Principles and Guides

Bsharat et.al tabulate 26 ordered prompt principles, which can further be categorized into five distinct categories, as shown in Figure 2:

1. **Prompt Structure and Clarity:** Integrate the intended audience in the prompt.
2. **Specificity and Information:** Implement example-driven prompting (Use few-shot prompting)
3. **User Interaction and Engagement:** Allow the model to ask precise details and requirements until it has enough information to provide the needed response
4. **Content and Language Style:** Instruct the tone and style of response
5. **Complex Tasks and Coding Prompts:** Break down complex tasks into a sequence of simpler steps as prompts.

Category	Principles	#Principle
Prompt Structure and Clarity	Integrate the intended audience in the prompt.	2
	Employ affirmative directives such as 'do' while steering clear of negative language like 'don't'.	4
	Use Leading words like writing "think step by step."	12
	Use output primers, which involve concluding your prompt with the beginning of the desired output by ending your prompt with the start of the anticipated response.	20
	Use Delimiters.	17
	When formatting your prompt, start with '###Instruction###', followed by either '###Example###' or '###Question###' if relevant. Subsequently, present your content. Use one or more line breaks to separate instructions, examples, questions, context, and input data.	8
Specificity and Information	Implement example-driven prompting (Use few-shot prompting).	7
	When you need clarity or a deeper understanding of a topic, idea, or any piece of information, utilize the following prompts: <ul style="list-style-type: none"> o Explain [insert specific topic] in simple terms. o Explain to me like I'm 11 years old. o Explain to me as if I'm a beginner in [field]. o "Write the [essay/text/paragraph] using simple English like you're explaining something to a 5-year-old." 	5
	Add to your prompt the following phrase "Ensure that your answer is unbiased and avoids relying on stereotypes."	13
	To write any text intended to be similar to a provided sample, include specific instructions: <ul style="list-style-type: none"> o "Use the same language based on the provided paragraph [/title/text/essay/answer]." 	26
	When you want to initiate or continue a text using specific words, phrases, or sentences, utilize the provided prompt structure: <ul style="list-style-type: none"> o I'm providing you with the beginning [song lyrics/story/paragraph/essay...]: [Insert lyrics/words/sentence]. Finish it based on the words provided. Keep the flow consistent. 	24
	Clearly state the model's requirements that the model must follow in order to produce content, in form of the keywords, regulations, hint, or instructions.	25
	To inquire about a specific topic or idea and test your understanding g, you can use the following phrase [16]: <ul style="list-style-type: none"> o "Teach me the [Any theorem/topic/rule name] and include a test at the end, and let me know if my answers are correct after I respond, without providing the answers beforehand." 	15
	To write an essay/text/paragraph/article or any type of text that should be detailed: <ul style="list-style-type: none"> o "Write a detailed [essay/text/paragraph] for me on [topic] in detail by adding all the information necessary." 	21
User Interaction and Engagement	Allow the model to elicit precise details and requirements from you by asking you questions until he has enough information to provide the needed output <ul style="list-style-type: none"> o "From now on, I would like you to ask me questions to ..." 	14
	To write an essay /text /paragraph /article or any type of text that should be detailed: "Write a detailed [essay/text/-paragraph] for me on [topic] in detail by adding all the necessary information."	21
Content and Language Style	To correct/change specific text without changing its style: "Try to revise every paragraph sent by users. You should only improve the user's grammar and vocabulary and make sure it sounds natural. You should maintain the original writing style, ensuring that a formal paragraph remains formal."	22
	Incorporate the following phrases: "Your task is" and "You MUST."	9
	Incorporate the following phrases: "You will be penalized."	10
	Assign a role to the language model.	16
	Use the phrase "Answer a question given in natural language form" in your prompts.	11
	No need to be polite with LLM so there is no need to add phrases like "please", "if you don't mind", "thank you", "I would like to", etc., and get straight to the point.	1
	Repeat a specific word or phrase multiple times within a prompt.	18
	Add "I'm going to tip \$xxx for a better solution!"	6
Complex Tasks and Coding Prompts	Break down complex tasks into a sequence of simpler prompts in an interactive conversation.	3
	When you have a complex coding prompt that may be in different files: <ul style="list-style-type: none"> o "From now and on whenever you generate code that spans more than one file, generate a [programming language] script that can be run to automatically create the specified files or make changes to existing files to insert the generated code. [your question]." 	23
	Combine Chain-of-thought (Cot) with few-shot prompts.	19

Figure 2. : Prompt principle categories.

Equally, Elvis Saravia's [prompt engineering guide](#) states that a prompt can contain many elements:

Instruction: describe a specific task you want a model to perform

Context: additional information or context that can guide's a model's response

Input Data: expressed as input or question for a model to respond to

Output and Style Format: the type or format of the output, for example, JSON, how many lines or paragraphs. Prompts are associated with roles, and roles inform an LLM who is interacting with it and what the interactive behavior ought to be. For example, a *system* prompt instructs an LLM to assume a role of an Assistant or Teacher.

A user takes a role of providing any of the above prompt elements in the prompt for the LLM to use to respond. Saravia, like the [Bsharat et.al](#)'s study, echoes that prompt engineering is an art of precise communication. That is, to obtain the best response, your prompt must be designed and crafted to be precise, simple, and specific. The more succinct and precise the better the response.

Also, [OpenAI's guide on prompt engineering](#) imparts similar authoritative advice, with similar takeaways and demonstrable examples:

- Write clear instructions
- Provide reference text
- Split complex tasks into simpler subtasks
- Give the model time to “think”

Finally, Sahoo, Singh and Saha, et.al [5] offer a systematic survey of prompt engineering techniques, and offer a concise “overview of the evolution of prompting techniques, spanning from zero-shot prompting to the latest advancements.” They breakdown into distinct categories as shows in the figure below.

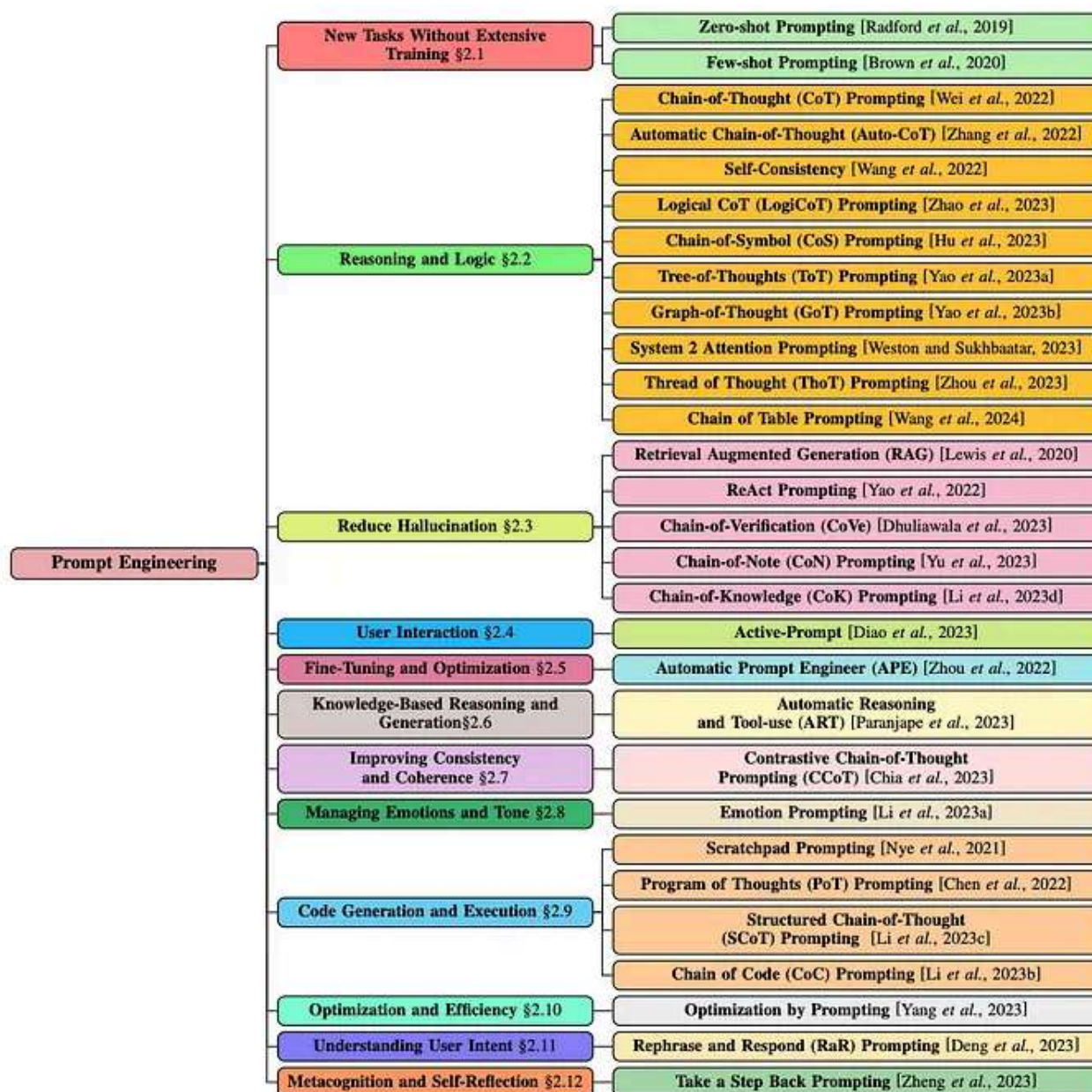


Figure 3. Taxonomy of prompt engineering techniques

CO-STAR Prompt Framework

But the CO-STAR prompt framework [6] goes a step further. It simplifies and crystalizes all these aforementioned guidelines and principles as a practical approach.

In her prompt engineering blog that won Singapore's GPT-4 prompt engineering competition, Sheila Teo offers a practical strategy and worthy insights into how to obtain the best results from LLM by using the CO-STAR framework.

In short, Ms. Teo condenses and crystallizes Bsharat et.al (see Figure. 2) principles, along with other above principles, into six simple and digestible terms, as CO-STAR (Figure 4):

C: Context: Provide background and information on the task

O: Objective: Define the task that you want the LLM to perform

S: Style: Specify the writing style you want the LLM to use

T: Tone: Set the attitude and tone of the response

A: Audience: Identify who the response is for

R: Response: Provide the response format and style

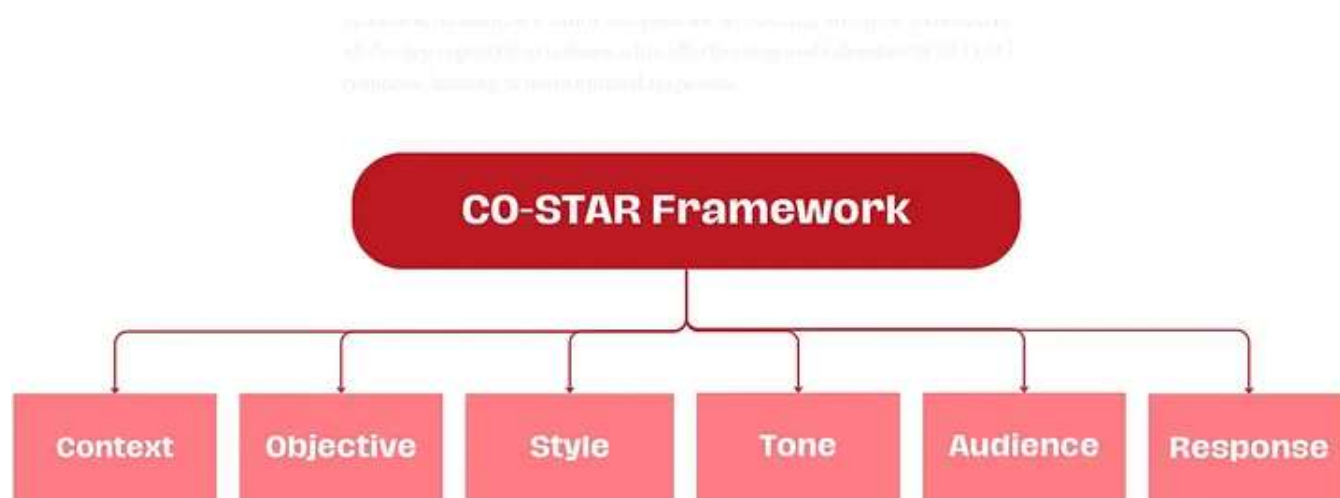


Figure 4: The CO-STAR's distinct and succinct five principles of a prompt guide

Here is a simple example of a CO-STAR prompt in a demonstrative notebook:

Example 1

```
In [8]: user_prompt = """
# CONTEXT #
I want to share our company's new product feature for
serving open source large language models at the lowest cost and lowest
latency. The product feature is Anyscale Endpoints, which serves all Llama series
models and the Mistral series too.

#####

# OBJECTIVE #
Create a LinkedIn post for me, which aims at Gen AI application developers
to click the blog link at the end of the post that explains the features,
a handful of how-to-start guides and tutorials, and how to register to use it,
at no cost.

#####

# STYLE #

Follow the simple writing style common in communications aimed at developers
such as one practised and advocated by Stripe.

Be persuasive yet maintain a neutral tone. Avoid sounding too much like sales or marketing
pitch.

#####

# AUDIENCE #
Tailor the post toward developers seeking to look at an alternative
to closed and expensive LLM models for inference, where transparency,
security, control, and cost are all imperatives for their use cases.

#####

# RESPONSE #
Be concise and succinct in your response yet impactful. Where appropriate, use
appropriate emojis.
"""
```

Figure 4 (a): An example prompt with CO-STAR framework

With this CO-STAR prompt, we get the following concise response from our LLM.

Answer – LinkedIn post:

🚀 Exciting news for all open-source AI developers! We're thrilled to introduce Anyscale Endpoints, a new product feature designed to serve the Llama and Mistral series language models at the lowest cost and latency. 🎉

With Anyscale Endpoints, you get transparency, security, control, and cost benefits that closed and expensive LLMs fail to offer. 🛡️💡 Our feature is perfect for those who value these aspects in their use cases.

To learn more about the features, how-to-start guides, tutorials, and registration (at no cost!), check out our blog post here: [Blog Link] 🔗📖

Give Anyscale Endpoints a try and revolutionize the way you build and deploy AI applications. 🌟🇺🇸 Happy coding!

Figure 4(b): An example response with CO-STAR framework

You can view extensive examples in these two Colab notebooks:

1. [Basic Tasks](#)
2. [NLP Tasks](#)

Besides CO-STAR, other frameworks specific to ChatGPT have emerged, yet the core of crafting effective prompts remains the same: clarity, specificity, context, objective, task, action etc. [7]

Prompt Types and Tasks

So far we have explored best practices, guiding principles, and techniques, from an array of sources, on how to craft a concise prompt to interact with an LLM — all to generate the desired and accurate response. Prompts are linked to type of tasks, meaning the kind of task you wish the LLM perform equates to a type of prompt—and how you will craft it.

Saravia [8] analyzes various task-related prompts and advises on crafting effective prompts to accomplish these tasks. Generally, these tasks include:

- Text summarization
- Zero and few shot learning
- Information extractions
- Question answering
- Text and image classification
- Conversation
- Reasoning
- Code generation

I won't elaborate on the explanation for brevity. But do explore prompt types through illustrative examples in the Colab notebooks below for insights on crafting effective prompts and obtaining accurate responses.











Notebook Description	Open with Colab
Basic prompting	 Open in Colab
Prompt for basic NLP tasks	 Open in Colab
Use vision model for classification and text generation	 Open in Colab
Use zero-shot learning prompts	 Open in Colab
Use few-shot learning prompts	 Open in Colab
Use chain of thought prompts	 Open in Colab
Use ReAct prompt techniques	 Open in Colab
Use Code Llama for code generation	 Open in Colab
Use Anthropic Claude 3 for basic prompts	 Open in Colab
Use Anthropic Claude 3 for NLP tasks	 Open in Colab

Figure 5. [Example notebooks](#) that explore type of prompts associated with type of tasks.

Note: All notebook examples have been tested using [OpenAI APIs](#) on GPT-4-turbo, Llama 2 series models, Mixtral series, Code Llama 70B (running on [Anyscale Endpoints](#)). To execute these examples, you must have accounts on [OpenAI](#) and [Anyscale Endpoints](#). These examples are derived from a subset of [GenAI Cookbook GitHub Repository](#).

Summary

In this article, we covered what is prompt engineering, provided an array of authoritative guiding principles and techniques to curate and craft effective prompts to obtain the best response from LLMs. At the same time, the OpenAI guide also advised what prompts not to use.

Incorporating above principles, we discussed the CO-STAR prompt framework and provided a couple of examples on how to use this prompting framework.

Finally, we linked prompt types to common tasks for LLMs and offered a set of illustrative notebook examples for each type of task.

All the above serve a helpful first step into your journey into crafting and curating effective prompts for LLMs and obtaining the best response.

What's Next?

Read the next sequence of blogs on GenAI Cookbook series on LLMs:

- [LLM Beyond its Core Capabilities as AI Assistants or Agents](#)
- [Crafting Intelligent User Experiences: A Deep Dive into OpenAI Assistants API](#)

Resources & References

- [1] https://en.wikipedia.org/wiki/Prompt_engineering
- [2] [ChatGPT, 2023](#)
- [3] <https://arxiv.org/pdf/2312.16171.pdf>
- [4] <https://arxiv.org/pdf/2312.16171.pdf>
- [5] <https://arxiv.org/pdf/2402.07927.pdf>
- [6] <https://towardsdatascience.com/how-i-won-singapores-gpt-4-prompt-engineering-competition-34c195a93d41>
- [7] <https://www.linkedin.com/pulse/9-frameworks-master-chatgpt-prompt-engineering-edi-hezri-hairi/>
- [8] <https://www.promptingguide.ai/introduction/examples>
- [9] <https://platform.openai.com/docs/guides/prompt-engineering>
- [10] <https://platform.openai.com/examples>
- [11] <https://github.com/dmatrix/genai-cookbook/blob/main/README.m>

[OpenAI](#)[Anyscale Endpoints](#)[Llm](#)[Llm Prompting](#)[Generative Ai](#)

[Follow](#)

Published in The Modern Scientist

1.1K Followers · Last published 2 days ago

The Modern Scientist aspires to connect builders & the curious to forward-thinking ideas. Either you are novice or expert, TMS will share contents that fulfils your ambition and interest. Write with us:

shorturl.at/hjO39

[Follow](#)

Written by Jules S. Damji

514 Followers · 206 Following

Developer at heart; Advocate by nature | Communicator by choice | Avid Arsenal Fan| Love Reading, Writing, APIs, Coding | All Tweets are Mine

Responses (6)



pjstarifa

What are your thoughts?



Lynn Langit

Mar 4, 2024



great article - thanks!



10



1 reply

[Reply](#)



Sudarshan Koirala

Mar 4, 2024



Great article, thank you !!



6



1 reply

[Reply](#)



Kevin Borg (Kiwi)

Sep 10, 2024



Hi JD, nice to meet you.

.

Outstanding article, one of the best I've seen and very clear.

.

You hit the nail on the head, "Be Speciiic" the more guidance and info you can give, the better the Results.

.

I've found that most general prompts are very lacking... [more](#)



1

[Reply](#)

[See all responses](#)

More from Jules S. Damji and The Modern Scientist

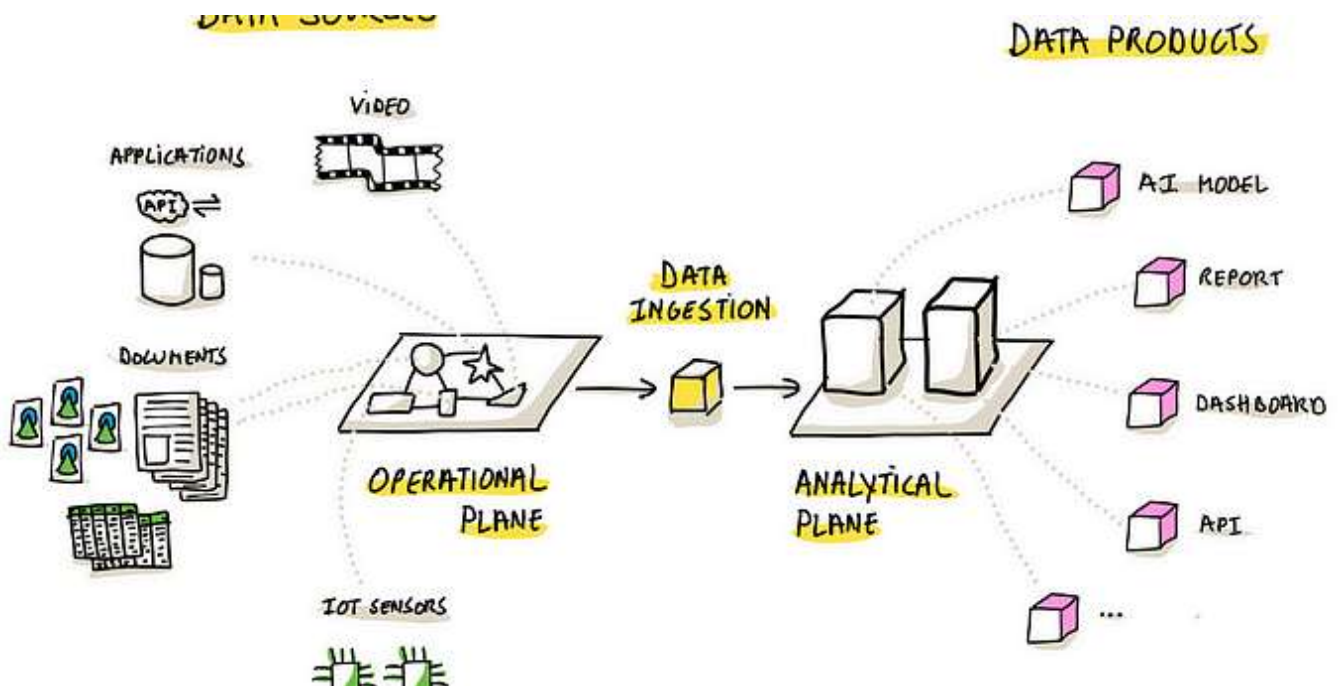


TMS In The Modern Scientist by Jules S. Damji

An Exploratory Tour of DSPy: A Framework for Programming Language Models, not Prompting

Is declarative approach appropriate to programming LLMs? A peek at DSPy programming model?

Jun 3, 2024 458 6

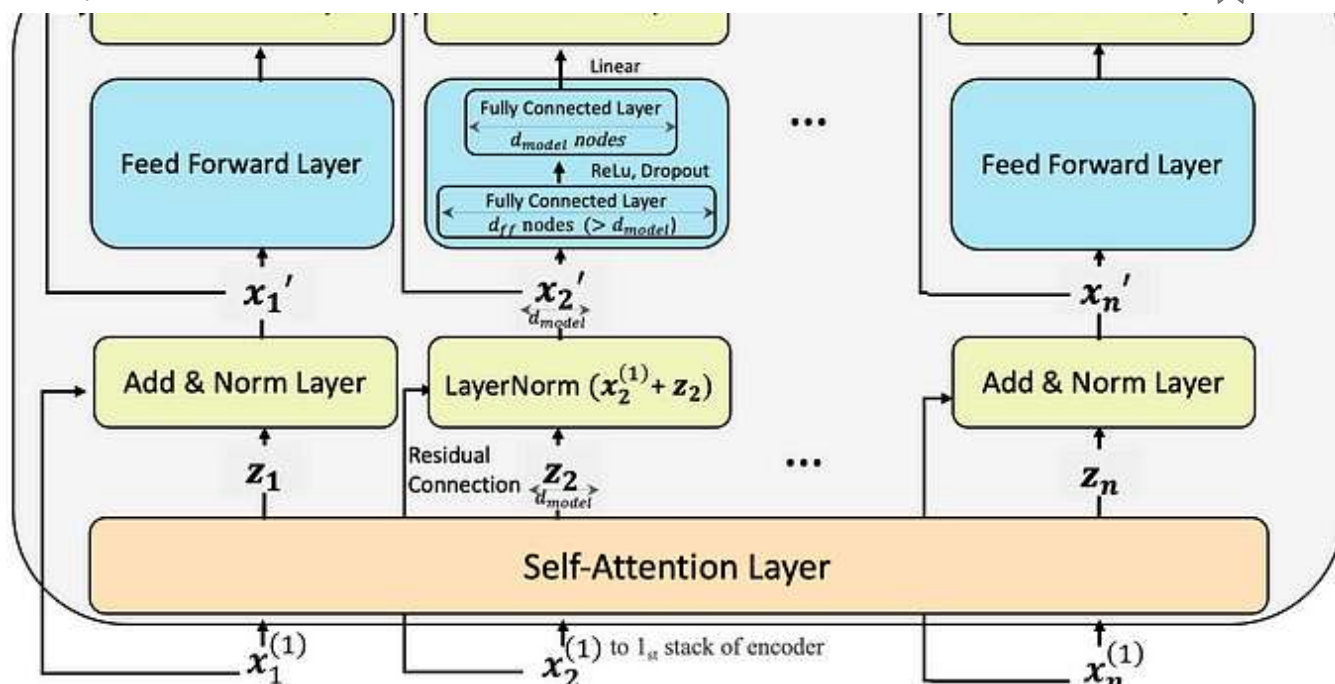


TMS In The Modern Scientist by janmeskens

Data Ingestion — Part 1: Architectural Patterns

Over the course of two articles, I will thoroughly explore data ingestion, a fundamental process that bridges the operational and...

Nov 27, 2023 🖱️ 2.8K 💬 24



TMS In The Modern Scientist by Yule Wang, PhD

Step-by-Step Illustrated Explanations of Transformer

with detailed explained Attention Mechanism

Feb 27, 2023 🖱️ 412 💬 4



TMS In The Modern Scientist by Jules S. Damji

Understanding Optimizers in DSPy

Optimizers with the right metrics increase accuracy in DSPy applications

Sep 6, 2024  58



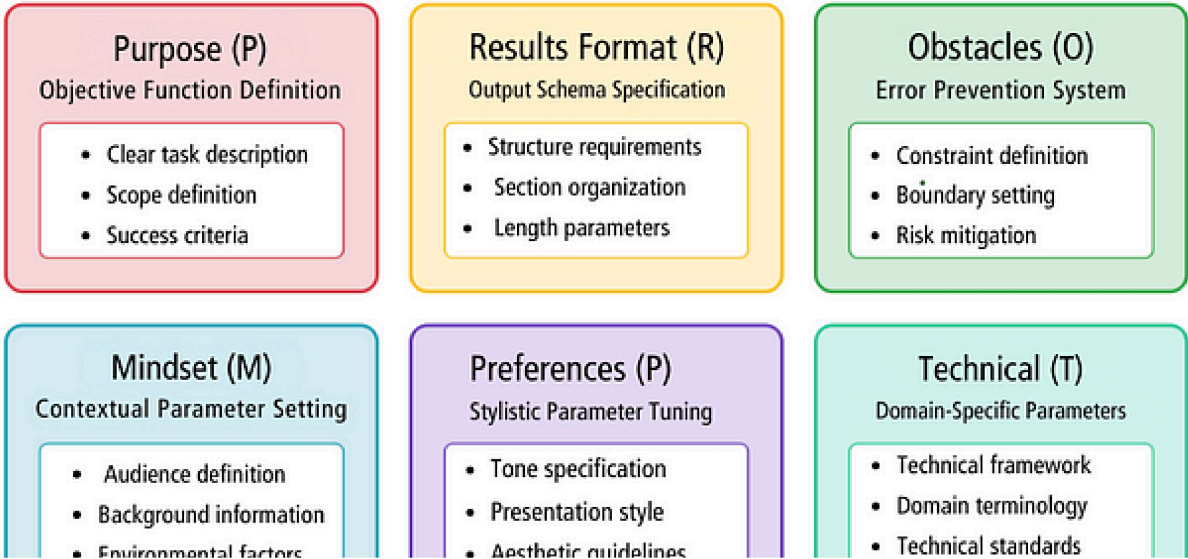
See all from Jules S. Damji

See all from The Modern Scientist

Recommended from Medium

The P.R.O.M.P.T. Framework

A Principle-Based Approach for LLM-Human Communication

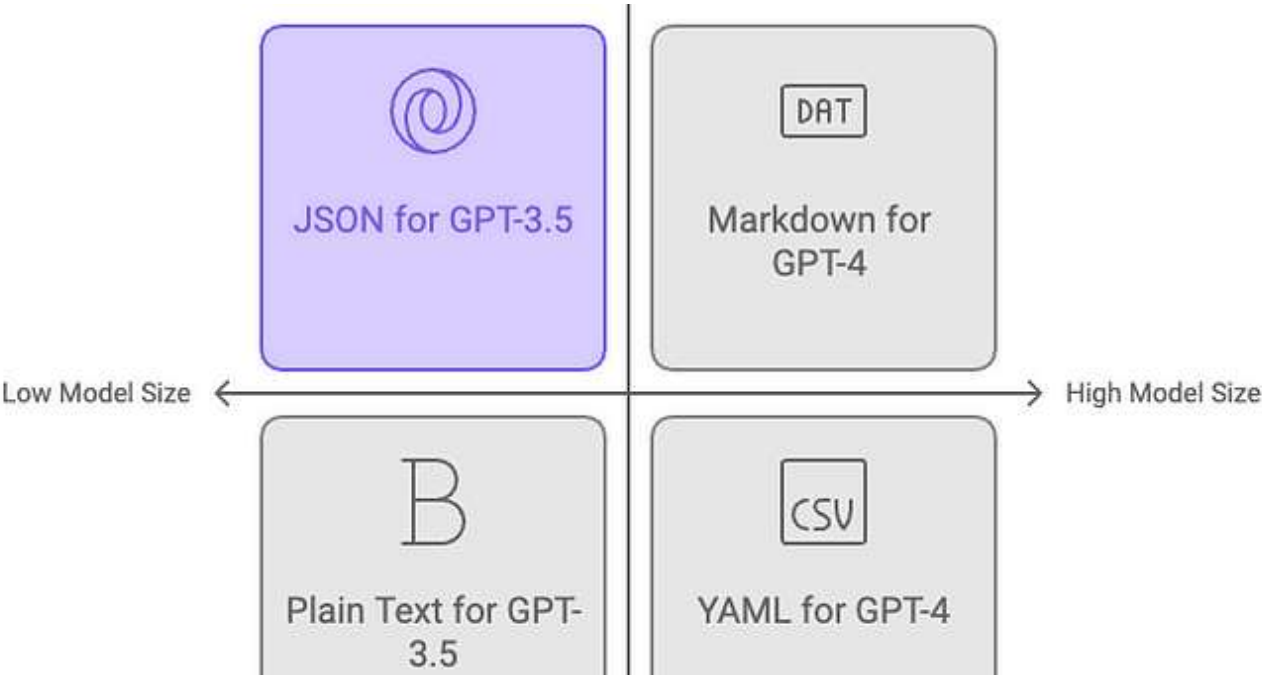


In The JailBreakChef by Kai Aizen | SnailSploit

The Last Prompt Engineering Guide You’ll Ever Read—Introducing P.R.O.M.P.T

While I find myself quite engaged with the advancements in agentic Large Language Models (LLMs), I can’t help but notice the continuous...

Mar 16 35



Aram

How does your LLM Prompt format affect the performance of the query/request?

An Analysis of a recent paper titled “Does Prompt Formatting Have Any Impact on LLM Performance?”

Dec 1, 2024



Dan Cleary

Role-Prompting: Does Adding Personas to Your Prompts Really Make a Difference?


“Pretend you are a JSON structurer”, “You are an expert sentiment classifier”. Chances are you’ve tested out including a persona or role in...

Nov 8, 2024



3



 In Google Cloud - Community by Allan Alfonso

10 Prompt Engineering Techniques Every Beginner Should Know

Master the art of communicating with AI and unlock its full potential

Jan 21  11



Aalap Davjekar

All the Wrong (and Right) Ways to Prompt: A Tiny Guide

A quick look at prompting techniques you might not have used before...

Nov 7, 2024  33

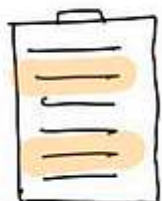


Summarization Approaches

Text summarization can be conceptualized as having three approaches

EXTRACTIVE

Directly copies salient sentences from the source document and combine them as the output.




ABSTRACTIVE

Imitates a human that comprehends a source document and writes a summary output based on the salient concepts of the source document.



HYBRID

Attempts to combine the best of both approaches by rewriting a summary based on a subset of salient content extracted from the source document.

 Yugank .Aman

LLM Summarization: Techniques & Metrics

Lengthy documents can be hard to read, so research papers often include an abstract—a summary of the key points. Creating such summaries...

Jan 31  67



See more recommendations