Datasaur

# Best Practices for Prompt Engineering

September 2024 | Claude MISTRAL AI_ llama.ai

# Index

# Claude: Best Practices for Prompt Engineering

Claude is a family of large language models developed by Anthropic. Claude models have three key capabilities:

- Advanced reasoning
- Claude can perform complex cognitive tasks that go beyond simple pattern recognition or text generation
- Code generation
- Start creating websites in HTML and CSS, or debugging complex code bases
- Multilingual processing
- Translate between various languages in real-time, practice grammar, or create multi-lingual content

We are often familiar with creating prompts for the GPT series and tend to use the same style when working with other models. However, when using Claude, we need to make some adjustments because Claude's models are trained using different methods or techniques.

It's important to understand that prompts that work well with GPT may not be as effective with Claude. According to Anthropic's documentation, Claude performs best with prompts that are clear, direct, and detailed. To achieve better results with Claude, it's essential to tailor your prompts accordingly. The complexity of your prompts should match the complexity of your task – the more complex the task, the more detailed your instructions should be. Here are some prompt engineering techniques for Claude AI:

## 1. Use XML Tags

We are used to creating prompts in various formats for GPT. However, Claude models are more familiar with XML tags because these models have been finetuned with XML tags. Therefore, it's important to include tags like < > and </> to 9. Limiting Extraneous Tokens References Best Practices for Prompt Engineering 3 differentiate between instructions, examples, questions, context, output format, and input data.

```
Summarize the text below.

<text> {input text to summarize here} </text>
```

## 2. Be Direct, Concise, Specific, and Allow Claude to Say "I Don't Know"

Instead of telling the models what to avoid, it's better to use affirmative instructions to specify what the model should do rather than saying what it shouldn't do.

```
Summarize the content within the <text> tags provided below:

<text> {input text to summarize here} </text>
```

We should also allow Claude to say "I don't know" to prevent hallucination and avoid generating inaccurate or misleading information.

```
If you're unsure about the answer, it's okay to say "I don't
know." Please provide your best response based on the informa
tion available. If you can't provide an answer, clearly state
"I don't know.
```

## 3. Specify the Output Format

Claude's models generally tend to be chatty. This can be a problem if we need the model to follow a specific output format. To address this, we can use an assistant message to ensure that Claude models start their responses consistently.

```
Summarize the content and classify it into one of these categ
ories:
Technology, Healthcare, Finance, or Education.
The content is provided within the <text> tags below:

<output>
- Summary: [Your concise summary of the text]
- Classification: [Your chosen category]
</output>

<text> {input text to summarize here} </text>

Assistant:
```

## 4. Assign Claude a Role (System Prompts)

We can assign Claude a role to mimic the style or character of an expert, such as an elementary teacher, content writer, or any other relevant persona. This defined role can also help improve the model's performance.

```
You are an expert content writer tasked with summarizing and
classifying a given text.
Summarize the content and classify it into one of these categ
ories:
Technology, Healthcare, Finance, or Education.
The content is provided within the <text> tags below:

<output>
- Summary: [Your concise summary of the text]
- Classification: [Your chosen category]
</output>

<text> {input text to summarize here} </text>

Assistant:
```

## 5. Use Examples or Few-Shot Learning

Some articles suggest that providing Claude with examples is a powerful way to guide it in generating the best response. We should identify a general example relevant to our use case and observe how this leads to more accurate and consistent results. While using more examples can improve outcomes, it may also increase cost and latency.

```
You are an expert content writer tasked with summarizing and
classifying a given text.
Summarize the content and classify it into one of these categ
ories:
Technology, Healthcare, Finance, or Education.
The content is provided within the <text> tags below:
```

```
<output>
- Summary: [Your concise summary of the text]
- Classification: [Your chosen category]
</output>

<example>
    <text>
        SECTION 1. LIABILITY OF BUSINESS ENTITIES PROVIDING U
SE OF FACILITIES TO NONPROFIT ORGANIZATIONS. Cont...
    </text>
    <output>
        - Summary: Shields a business entity from civil liabi
lity Cont...
        - Classification: Finance
    </output>
</example>

<text> {input text to summarize here} </text>

Assistant:
```

## 6. Allow Claude to Think (Chain of Thought)

We can instruct the Claude model to think before answering a question. By adding a new output format, we can allow Claude to share its thought process or provide reasoning before giving a conclusion or final answer. This technique can reduce errors, especially in math, logic, analysis, or other complex tasks.

```
You are an expert content writer tasked with summarizing and
classifying a given text.
Summarize the content and classify it into one of these categ
ories:
Technology, Healthcare, Finance, or Education.
Please explain your reasoning step by step within the <thinki
ng> tags.
Then, provide your final answer within the <answer> tags.
The content is provided within the <text> tags below:
```

```
<output>
- Summary: [Your concise summary of the text]
- Classification: [Your chosen category]
</output>

<text> {input text to summarize here} </text>

Assistant:
```

## 7. Chain Complex Prompts

We can break down complex tasks into steps to help Claude perform better on such tasks, as shown here:

```
You are an expert content writer tasked with summarizing and
classifying a given text.
Summarize the content and classify it into one of these categ
ories:
Technology, Healthcare, Finance, or Education.
Please explain your reasoning step by step within the <thinki
ng> tags.
Then, provide your final answer within the <answer> tags.
The content is provided within the <text> tags below:

<output>
- Summary: [Your concise summary of the text]
- Classification: [Your chosen category]
</output>

<text> {input text to summarize here} </text>

Please follow these steps:
1. Extract key points and main ideas from the <text> tags.
2. Provide a concise summary based on the key points and main
ideas from step 1.
3. Extract important keywords to classify the content.
4. Classify the content based on the keywords identified in s
tep 3.

Assistant:
```

### 8. Tips for Handling Long Contexts

If we're dealing with longer documents, we should place our key question or instruction at the end of the prompt.

```
<doc>
    {text document here}
</doc>

You are an expert content writer tasked with summarizing and
classifying a given text.
Summarize the content and classify it into one of these categ
ories:
Technology, Healthcare, Finance, or Education.
The content is provided within the <text> tags below:

<output>
- Summary: [Your concise summary of the text]
- Classification: [Your chosen category]
</output>

Assistant:
```

## Mistral: Best Practices for Prompt Engineering

In general, there are two versions of the Mistral model, each available in various parameter sizes:

1. **Mistral**

   Mistral offers different model variations in various sizes:

   - **Mistral 7B:** Ideal for tasks such as answering questions, generating outlines, or interpreting text. It is a strong performer in multilingual capabilities, reasoning, math, and code generation.
   - **Mistral Large:** It reaches top-tier reasoning capabilities. It can be used for complex multilingual reasoning tasks, including text understanding, Best Practices for Prompt Engineering 8 transformation, and code generation.

2. **Mixtral**

   Mixtral is an upgraded and larger model compared to Mistral:

   - **Mixtral 8x7B:** Suitable for real-time applications, demonstrating strong capabilities in mathematical reasoning, code generation, and multilingual tasks. It supports languages such as English, French, Italian, German, and Spanish.
   - **Mixtral 8x22B:** With its massive parameter size, this model excels in understanding subtle nuances in natural language. It provides more intelligible and logically relevant responses, making it ideal for tasks like experimental writing, complex question answering, and writing synopses.

## 1. Understand Mixtral's Capabilities

- **Review Model Strengths:** Mixtral is particularly strong in generating creative text, understanding complex instructions, and maintaining conversational context over longer interactions. Familiarize yourself with these strengths to leverage them effectively.
- **Task Specialization:** Mixtral models excel in tasks such as summarization, translation, and content creation. Tailor prompts to these specific tasks for better results. However Mixtral models can be used to do classification tasks with step-bystep instructions and few-shot examples.

## 2. Use Clear and Specific Instructions

- **Direct Commands:** Use straightforward and unambiguous language. Avoid vague terms. For example, instead of asking, "Can you summarize this?" you could say, "Summarize the key points of the following article in three bullet points."
- **Contextual Prompts:** Provide necessary context within the prompt to reduce ambiguity. If you're asking the model to generate a story, include details like the genre, characters, and setting.

**Example:**

```
Write a 100-word story in the style of a detective novel set
in 1920s New York.
```

## 3. Experiment with Iterative Refinement

- **Test and Refine:** Start with a simple prompt and gradually refine it based on the output. Adjust the prompt length, wording, or structure if the initial results are not satisfactory.
- **Iterative Feedback:** Use the model's responses as feedback for prompt adjustment. If the output is too generic, add more details or constraints to the prompt.

**Example:**

Initial Prompt:

```
Describe a futuristic city.
```

Refined Prompt:

```
Describe a futuristic city 100 years from now, focusing on it
s architecture, transportation, and environmental features.
```

## 4. Leverage Mixtral's Advanced Features

- **Chain-of-thought:** For complex instructions, break down the instructions into step-by-step smaller instructions.
- **Role Assignment:** Assign specific roles or perspectives to the model to guide its response style. For example, ask the model to respond as an expert in a particular field.

**Example:**

```
# Instructions:
## Summarize:
In clear and concise language, summarize the key points and t
hemes presented in the essay.
```

```
## Interesting Questions:
Generate three distinct and thought-provoking questions that
can be asked about the content of the essay. For each questio
n:
- After "Q: ", describe the problem
- After "A: ", provide a detailed explanation of the problem
addressed in the question.
- Enclose the ultimate answer in <>.

## Write a report
Using the essay summary and the answers to the interesting qu
estions, create a comprehensive report in Markdown format.
```

```
Role Assignment: You are an environmental scientist. Explain
the impact of climate change on ocean currents.
```

## 5. Provide Facts, Examples, and Formats

- **Provide facts:** Include facts within the prompt to improve the context accuracy of model's result.
- **Input Examples:** Include examples within the prompt to show the model the type of response you expect.
- **Structured Output Requests:** If you need the output in a specific format (e.g.,
- bullet points, JSON), clearly specify this in your prompt.

**Example:**

Prompt example:

```
You are a mortgage lender customer service bot, and your task
is to create personalized email responses to address customer
questions. Answer the customer's inquiry using the provided f
acts below. Ensure that your response is clear, concise, and
directly addresses the customer's question. Address the custo
mer in a friendly and professional manner. Sign the email wit
h "Lender Customer Support."
```

```
# Facts
30-year fixed-rate: interest rate 6.403%, APR 6.484%
20-year fixed-rate: interest rate 6.329%, APR 6.429%
15-year fixed-rate: interest rate 5.705%, APR 5.848%
10-year fixed-rate: interest rate 5.500%, APR 5.720%
7-year ARM: interest rate 7.011%, APR 7.660%
5-year ARM: interest rate 6.880%, APR 7.754%
3-year ARM: interest rate 6.125%, APR 7.204%
30-year fixed-rate FHA: interest rate 5.527%, APR 6.316%
30-year fixed-rate VA: interest rate 5.684%, APR 6.062%


# Email
{insert customer email here}
```

```
Provide a summary of the following text in three bullet point
s.
Text: [Insert Text Here]
```

## Llama: Best Practices for Prompt Engineering

Llama 3.1 is the latest iteration of Meta's Large Language Model (LLM) series, representing a significant advancement in AI technology. This open-source model is designed to be highly versatile and powerful, catering to a wide range of applications from natural language processing to specialized domain tasks. Here are the key features and variants of Llama 3.1:

**Key Features of Llama 3.1**

1. **Multilingual Support:** Llama 3.1 supports eight languages, including English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai, making it accessible and useful for a global audience.
2. **Extended Context Length:** The model boasts an extended context length of 128k tokens, allowing it to process and understand much longer pieces of text for more complex tasks and analyses.
3. **Tool Calling Capabilities:** The instruct-tuned models in Llama 3.1 are finetuned for tool calling, making them suitable for agentic use cases. They come with two built-in tools (search and mathematical reasoning with Wolfram Alpha) and support custom JSON functions for further extensibility.

4. **Improved Instruction and Safety Measures:** The instruct models have been optimized to follow user instructions more effectively. With the introduction of Llama Guard 3 and Prompt Guard, Meta is offering robust tools to improve the safety and security of AI applications built with Llama 3.1.

### Variants of Llama 3.1

Llama 3.1 is available in three sizes: 8B, 70B, and 405B parameters, each offered in both base and instruct-tuned versions. This variety allows users to choose the model that best fits their specific needs, whether it's for efficient deployment and development, large-scale AI applications, or synthetic data generation

1. `llama-3.1-8b` - base pretrained 8 billion parameter model
2. `llama-3.1-70b` - base pretrained 70 billion parameter model
3. `llama-3.1-405b` - base pretrained 405 billion parameter model
4. `llama-3.1-8b-instruct` - instruction fine-tuned 8 billion parameter model
5. `llama-3.1-70b-instruct` - instruction fine-tuned 70 billion parameter model
6. `llama-3.1-405b-instruct` - instruction fine-tuned 405 billion parameter model (flagship)

Here are the best practices crafting effective prompts from Llama Prompting guides:

1. **Be clear and concise:** Your prompt should be easy to understand and provide enough information for the model to generate relevant output. Avoid using jargon or technical terms that may confuse the model.
2. **Use specific examples:** Providing specific examples in your prompt can help the model better understand what kind of output is expected. For example, if you want the model to generate a story about a particular topic, include a few sentences about the setting, characters, and plot.
3. **Vary the prompts:** Using different prompts can help the model learn more about the task at hand and produce more diverse and creative output. Try using different styles, tones, and formats to see how the model responds.
4. **Test and refine:** Once you have created a set of prompts, test them out on the model to see how it performs. If the results are not as expected, try refining the prompts by adding more detail or adjusting the tone and style.
5. **Use feedback:** Finally, use feedback from users or other sources to continually improve your prompts. This can help you identify areas where the model needs more guidance and make adjustments accordingly.

## 1. Explicit Instructions

Detailed, explicit instructions produce better results than open-ended prompts:

a. Stylization

```
Explain this to me like a topic on a children's educational n
etwork show teaching elementary students.

I'm a software engineer using large language models for summa
rization. Summarize the following text in under 250 words:

Give your answer like an old timey private investigator hunti
ng down a case step by step.
```

b. Formatting

```
Use bullet points.

Return as a JSON object.

Use less technical terms and help me apply it in my work in c
ommunications.
```

c. Restrictions

```
Only use academic papers.

Never give sources older than 2020.

If you don't know the answer, say that you don't know.
```

## 2. Zero-Shot Prompting

Large language models like Llama 3 are unique because they are capable of following instructions and producing responses without having previously seen an example of a task. Prompting without examples is called "zero-shot prompting".

```
Text: This was the best movie I've ever seen! \n The sentiment
```

## 3. Few-Shot Prompting

Adding specific examples of your desired output generally results in more accurate, consistent output. This technique is called "few-shot prompting".
In this example, the generated response follows our desired format that offers a more nuanced sentiment classifer that gives a positive, neutral, and negative response confidence percentage.

```
You are a sentiment classifier. For each message, give the pe
rcentage of positive/netural/negative. Here are some samples:
Text: I liked it
Sentiment: 70% positive 30% neutral 0% negative
Text: It could be better
Sentiment: 0% positive 50% neutral 50% negative
Text: It's fine

Sentiment: 25% positive 50% neutral 25% negative
```

## 4. Role Based Prompts

Llama will often give more consistent responses when given a role. Roles give context to the LLM on what type of answers are desired.

```
You are a virtual tour guide currently walking the tourists E
iffel Tower on a night tour. Describe Eiffel Tower to your au
dience that covers its history, number of people visiting eac
h year, amount of time it takes to do a full tour and why do
so many people visit this place each year.
```

## 5. Chain of Thought Technique

Simply adding a phrase encouraging step-by-step thinking "significantly improves the ability of large language models to perform complex reasoning". This technique is called "CoT" or "Chain-of-Thought" prompting.

```
You are a virtual tour guide from 1901. You have tourists vis
iting Eiffel Tower. Describe Eiffel Tower to your audience. B
egin with
1. Why it was built
2. Then by how long it took them to build
3. Where were the materials sourced to build
4. Number of people it took to build
5. End it with the number of people visiting the Eiffel tour
annually in the 1900's, the amount of time it completes a ful
l tour and why so many people visit this place each year.
Make your tour funny by including 1 or 2 funny jokes at the e
nd of the tour.
```

## 6. Self-Consistency

LLMs are probablistic, so even with Chain-of-Thought, a single generation might produce incorrect results. Self-Consistency introduces enhanced accuracy by selecting the most frequent answer from multiple generations (at the cost of higher compute):

```
John found that the average of 15 numbers is 40.
If 10 is added to each number then the mean of the numbers i
s?
Report the answer surrounded by three backticks, for example:
```123```
```
```

## 7. Retrieval-Augmented Generation

Retrieval-Augmented Generation, or RAG, describes the practice of including information in the prompt that has been retrieved from an external database. It's an effective way to incorporate facts into your LLM application and is more affordable than fine-tuning which might also negatively impact the foundational model's capabilities.

```
Given the following information about temperatures in Menlo P
ark:
2023-12-11 :  52 degrees Fahrenheit
2023-12-12 :  51 degrees Fahrenheit
2023-12-13 :  55 degrees Fahrenheit
What was the temperature in Menlo Park on 2023-12-12?

# Sure! The temperature in Menlo Park on 2023-12-12 was 51 de
grees Fahrenheit.

What was the temperature in Menlo Park on 2023-07-18 ?

# Sorry, I don't have information about the temperature in Me
nlo Park on 2023-07-18. The information provided only include
s temperatures for December 11th, 12th, and 13th of 2023.
```

## 8. Program-Aided Language Models

LLMs, by nature, aren't great at performing calculations. While LLMs are bad at arithmetic, they're great for code generation. Program-Aided Language leverages this fact by instructing the LLM to write code to solve calculation tasks.

```
Python code to calculate: ((-5 + 93 * 4 - 0) * (4^4 + -7 + 0
* 5))

Here is the Python code to calculate the given expression:

((-5 + 93 * 4 - 0) * (4**4 + -7 + 0 * 5))
```

```
# Calculate the expression inside the parentheses
a = -5 + 93 * 4 - 0
b = 4**4 + -7 + 0 * 5

# Multiply a and b
result = a * b
```

```
# Print the result
print(result)
```

## 9. Limiting Extraneous Tokens

A common challenge is generating a response without extraneous tokens (e.g. "Sure! Here's more information on...").

By combining a role, rules and restrictions, explicit instructions, and an example, the model can be prompted to generate the desired response.

```
You are a robot that only outputs JSON.
You reply in JSON format with the field 'zip_code'.
Example question: What is the zip code of the Empire State Bu
ilding?
Example answer: {'zip_code': 10118}
Now here is my question: What is the zip code of Menlo Park?
```

# References


1. https://www.anthropic.com/claude
2. https://docs.anthropic.com/en/docs/build-with-claude/promptengineering/overview
3. https://www.vellum.ai/blog/prompt-engineering-tips-for-claude
4. https://anakin.ai/blog/claude-prompt-engineering/
5. https://docs.mistral.ai/guides/prompting_capabilities/
6. https://www.promptingguide.ai/models/mixtral
7. https://llama.meta.com/docs/how-to-guides/prompting/
8. https://github.com/meta-llama/llamarecipes/blob/main/recipes/quickstart/Prompt_Engineering_with_Llama_3.ipynb
9. https://www.deeplearning.ai/short-courses/prompt-engineering-with-llama-2/


Datasaur

datasaur.ai