Prompt engineering guidelines

Topics

- Introduction
- What is a prompt?
- What is prompt engineering?
- General guidelines for Amazon Bedrock LLM users
- Prompt templates and examples for Amazon Bedrock text models

Introduction

Welcome to the prompt engineering guide for large language models (LLMs) on Amazon Bedrock. Amazon Bedrock is Amazon's service for foundation models (FMs), which offers access to a range of powerful FMs for text and images.

Prompt engineering refers to the practice of optimizing textual input to LLMs to obtain desired responses. Prompting helps LLMs perform a wide variety of tasks, including classification, question answering, code generation, creative writing, and more. The quality of prompts that you provide to LLMs can impact the quality of their responses. These guidelines provide you with all the necessary information to get started with prompt engineering. It also covers tools to help you find the best possible prompt format for your use case when using LLMs on Amazon Bedrock.

Whether you're a beginner in the world of generative AI and language models, or an expert with previous experience, these guidelines can help you optimize your prompts for Amazon Bedrock text models. Experienced users can skip to the General Guidelines for Amazon Bedrock LLM Users or Prompt Templates and Examples for Amazon Bedrock Text Models sections.



Note

All examples in this doc are obtained via API calls. The response may vary due to the stochastic nature of the LLM generation process. If not otherwise specified, the prompts are written by employees of AWS.

Disclaimer: The examples in this document use the current text models available within Amazon Bedrock. Also, this document is for general prompting guidelines. For model-specific guides, refer

Introduction 187

to their respective docs on Amazon Bedrock. This document provides a starting point. While the following example responses are generated using specific models on Amazon Bedrock, you can use other models in Amazon Bedrock to get results as well. The results may differ between models as each model has its own performance characteristics. The output that you generate using AI services is your content. Due to the nature of machine learning, output may not be unique across customers and the services may generate the same or similar results across customers.

Additional prompt resources

The following resources offer additional guidelines on prompt engineering.

- Anthropic Claude model prompt guide: https://docs.anthropic.com/claude/docs
- Anthropic Claude prompt engineering resources: https://docs.anthropic.com/claude/docs/guide-to-anthropics-prompt-engineering-resources
- Cohere prompt guide: https://txt.cohere.com/how-to-train-your-pet-llm-prompt-engineering
- AI21 Labs Jurassic model prompt guide: https://docs.ai21.com/docs/prompt-engineering
- Meta Llama 2 prompt guide: https://ai.meta.com/llama/get-started/#prompting
- Stability documentation: https://platform.stability.ai/docs/getting-started
- Mistral AI prompt guide: https://docs.mistral.ai/guides/prompting-capabilities/

What is a prompt?

Prompts are a specific set of inputs provided by you, the user, that guide LLMs on Amazon Bedrock to generate an appropriate response or output for a given task or instruction.

User Prompt:

Who invented the airplane?

When gueried by this prompt, Titan provides an output:

Output:

The Wright brothers, Orville and Wilbur Wright are widely credited with inventing and manufacturing the world's first successful airplane.

(Source of prompt: AWS, model used: Amazon Titan Text G1 - Express)

Additional prompt resources 188

Components of a prompt

A single prompt includes several components, such as the task or instruction you want the LLMs to perform, the context of the task (for example, a description of the relevant domain), demonstration examples, and the input text that you want LLMs on Amazon Bedrock to use in its response. Depending on your use case, the availability of the data, and the task, your prompt should combine one or more of these components.

Consider this example prompt asking Titan to summarize a review:

User Prompt:

The following is text from a restaurant review:

"I finally got to check out Alessandro's Brilliant Pizza and it is now one of my favorite restaurants in Seattle. The dining room has a beautiful view over the Puget Sound but it was surprisingly not crowded. I ordered the fried castelvetrano olives, a spicy Neapolitan-style pizza and a gnocchi dish. The olives were absolutely decadent, and the pizza came with a smoked mozzarella, which was delicious. The gnocchi was fresh and wonderful. The waitstaff were attentive, and overall the experience was lovely. I hope to return soon."

Summarize the above restaurant review in one sentence.

(Source of prompt: AWS)

Based on this prompt, Titan responds with a succinct one-line summary of the restaurant review. The review mentions key facts and conveys the main points, as desired.

Output:

Alessandro's Brilliant Pizza is a fantastic restaurant in Seattle with a beautiful view over Puget Sound, decadent and delicious food, and excellent service.

(Model used: Amazon Titan Text G1 - Express)

The instruction **Summarize the above restaurant review in one sentence** and the review text **I finally got to check out ...** were both necessary for this type of output. Without either one, the model would not have enough information to produce a sensible summary. The *instruction* tells the LLM what to do, and the text is the *input* on which the LLM operates.

Components of a prompt 189

The *context* (**The following is text from a restaurant review**) provides additional information and keywords that guide the model to use the input when formulating its output.

In the example below, the text Context: Climate change threatens people with increased flooding ... is the *input* which the LLM can use to perform the *task* of answering the question Question: What organization calls climate change the greatest threat to global health in the 21st century?".

User prompt:

Context: Climate change threatens people with increased flooding, extreme heat, increased food and water scarcity, more disease, and economic loss. Human migration and conflict can also be a result. The World Health Organization (WHO) calls climate change the greatest threat to global health in the 21st century. Adapting to climate change through efforts like flood control measures or drought-resistant crops partially reduces climate change risks, although some limits to adaptation have already been reached. Poorer communities are responsible for a small share of global emissions, yet have the least ability to adapt and are most vulnerable to climate change. The expense, time required, and limits of adaptation mean its success hinge on limiting global warming.

Question: What organization calls climate change the greatest threat to global health in the 21st century?

(Source of prompt: https://en.wikipedia.org/wiki/Climate_change)

Al21 Labs Jurassic responses with the correct name of the organization according to the context provided in the prompt.

Output:

The World Health Organization (WHO) calls climate change the greatest threat to global health in the 21st century.

(Model used: AI21 Labs Jurassic-2 Ultra v1)

Few-shot prompting vs. zero-shot prompting

It is sometimes useful to provide a few examples to help LLMs better calibrate their output to meet your expectations, also known as *few-shot prompting* or *in-context learning*, where a *shot* corresponds to a paired example input and the desired output. To illustrate, first here is an example

of a zero-shot sentiment classification prompt where no example input-output pair is provided in the prompt text:

User prompt:

Tell me the sentiment of the following headline and categorize it as either positive, negative or neutral:

New airline between Seattle and San Francisco offers a great opportunity for both passengers and investors.

(Source of prompt: AWS)

Output:

Positive

(Model used: Amazon Titan Text G1 - Express)

Here is the few-shot version of a sentiment classification prompt:

User prompt:

Tell me the sentiment of the following headline and categorize it as either positive, negative or neutral. Here are some examples:

Research firm fends off allegations of impropriety over new technology.

Answer: Negative

Offshore windfarms continue to thrive as vocal minority in opposition dwindles.

Answer: Positive

Manufacturing plant is the latest target in investigation by state officials.

Answer:

(Source of prompt: AWS)

Output:

Negative

(Model used: Amazon Titan Text G1 - Express)

The following example uses Anthropic Claude models. When using Anthropic Claude models, it's a good practice to use <example></example> tags to include demonstration examples. We also recommend using different delimiters such as H: and A: in the examples to avoid confusion with

the delimiters Human: and Assistant: for the whole prompt. Notice that for the last few-shot example, the final A: is left off in favor of Assistant:, prompting Anthropic Claude to generate the answer instead.

```
User prompt:
Human: Please classify the given email as "Personal" or "Commercial" related emails.
Here are some examples.
<example>
H: Hi Tom, it's been long time since we met last time. We plan to have a party at my
 house this weekend. Will you be able to come over?
A: Personal
</example>
<example>
H: Hi Tom, we have a special offer for you. For a limited time, our customers can save
 up to 35% of their total expense when you make reservations within two days. Book now
 and save money!
A: Commercial
</example>
H: Hi Tom, Have you heard that we have launched all-new set of products. Order now, you
 will save $100 for the new products. Please check our website.
Assistant:
Output:
Commercial
```

(Source of prompt: AWS, model used: Anthropic Claude)

Prompt template

A prompt template specifies the formatting of the prompt with exchangeable content in it. Prompt templates are "recipes" for using LLMs for different use cases such as classification, summarization, question answering, and more. A prompt template may include instructions, few-shot examples, and specific context and questions appropriate for a given use case. The following example is a template that you can use to perform few-shot sentiment classification using Amazon Bedrock text models:

Prompt template 192

```
Prompt template:
"""Tell me the sentiment of the following
{{Text Type, e.g., "restaurant review"}} and categorize it
as either {{Sentiment A}} or {{Sentiment B}}.
Here are some examples:
Text: {{Example Input 1}}
Answer: {{Sentiment A}}
Text: {{Example Input 2}}
Answer: {{Sentiment B}}
Text: {{Input}}
Answer:"""
```

Users can copy and paste this template, filling in their own text and few-shot examples to complete prompts when using LLMs on Amazon Bedrock. More templates are provided in this section.



(i) Note

The double curly braces {{ and }} mark the places to put in data-specific information in the template, and should not be included in the prompt text.

Important notes on using Amazon Bedrock LLMs by API calls

If you are accessing LLMs on Amazon Bedrock via API calls, please be aware that they do not recall prior prompts and previous requests unless the previous interaction is included within the current prompt. You may need this for conversational-style interactions or follow up requests.

If you are accessing Anthropic Claude models using API calls instead of the Amazon Bedrock text playground, you wrap your prompts in a conversational style to get desired response (for the Titan Text and AI21 Labs Jurassic model family, it's not necessary). The following example demonstrates a conversational-style prompt:

```
User prompt:
Human: Describe where sea otters live in two sentences.
```

Assistant:

Output:

Here is a two sentence description of where sea otters live:

Sea otters live in the northern Pacific Ocean along the coasts of Russia, Alaska, Canada, and the continental United States.
They prefer coastal habitats like rocky shores, kelp forests, and estuaries where they have access to shallow, nearshore waters.

(Source of prompt: AWS, model used: Anthropic Claude)

Notice how the main content of the prompt is wrapped like this: \n\nHuman: {{Main Content}}\n\nAssistant:. For Anthropic Claude models, prompts sent via the API must contain \n\nHuman: and \n\nAssistant:.

To use conversational mode on Titan, you can use the format of User: $\{\{\}\}\$ \n Bot: when prompting the model.

What is prompt engineering?

Prompt engineering refers to the practice of crafting and optimizing input prompts by selecting appropriate words, phrases, sentences, punctuation, and separator characters to effectively use LLMs for a wide variety of applications. In other words, prompt engineering is the art of communicating with an LLM. High-quality prompts condition the LLM to generate desired or better responses. The detailed guidance provided within this document is applicable across all LLMs within Amazon Bedrock.

The best prompt engineering approach for your use case is dependent on both the task and the data. Common tasks supported by LLMs on Amazon Bedrock include:

- Classification: The prompt includes a question with several possible choices for the answer, and the model must respond with the correct choice. An example classification use case is sentiment analysis: the input is a text passage, and the model must classify the sentiment of the text, such as whether it's positive or negative, or harmless or toxic.
- Question-answer, without context: The model must answer the question with its internal knowledge without any context or document.
- **Question-answer, with context:** The user provides an input text with a question, and the model must answer the question based on information provided within the input text.

• **Summarization:** The prompt is a passage of text, and the model must respond with a shorter passage that captures the main points of the input.

- **Open-ended text generation:** Given a prompt, the model must respond with a passage of original text that matches the description. This also includes the generation of creative text such as stories, poems, or movie scripts.
- **Code generation:** The model must generate code based on user specifications. For example, a prompt could request text-to-SQL or Python code generation.
- Mathematics: The input describes a problem that requires mathematical reasoning at some level, which may be numerical, logical, geometric or otherwise.
- **Reasoning or logical thinking:** The model must make a series of logical deductions.

General guidelines for Amazon Bedrock LLM users

Design your prompt

Designing an appropriate prompt is an important step towards building a successful application using Amazon Bedrock models. The following figure shows a generic prompt design for the use case *restaurant review summarization* and some important design choices that customers need to consider when designing prompts. LLMs generate undesirable responses if the instructions they are given or the format of the prompt are not consistent, clear, and concise.