

Prompt Engineering, or Talking to LLMs

...

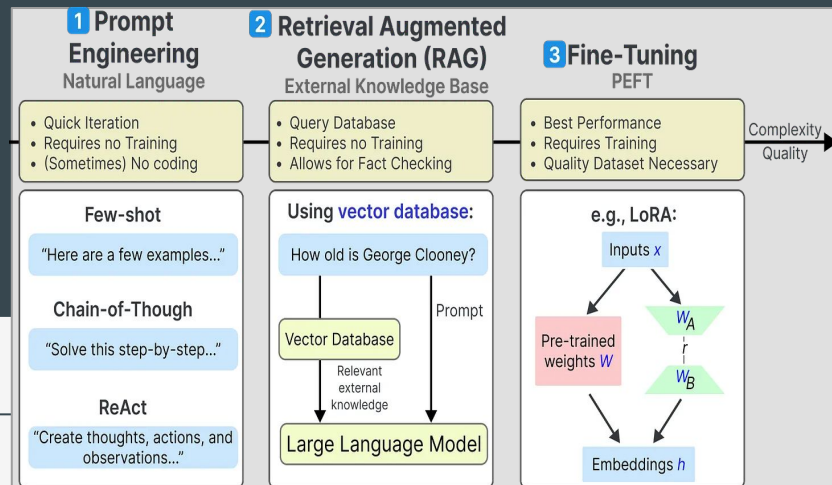
Krzysztof Nowak (Institute of Polish Language, Kraków)

Outline

- what is prompt engineering?
 - the anatomy of a prompt
 - rules and components
 - techniques
 - in-context learning
 - chain prompting
 - chain of thought
 - output verification
 - resources
 - hands-on session
-

How can we influence the results?

Model Selection	Different models for different tasks
Fine-tuning	Training on specific data
Parameter Tuning	Temperature, top-p, etc.
Prompt Engineering	Crafting effective inputs
System Design	RAG, tools, workflows



Source: [Maarten Grootendorst, 3 Ways To Improve Your Large Language Model](#)

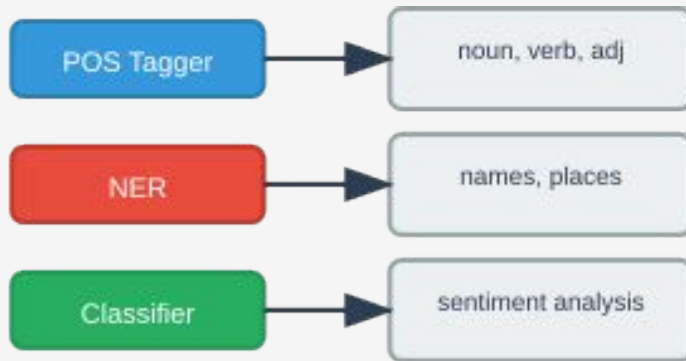
Why Do We Need to Talk to LLMs?

Traditional NLP: Predictable but Limited

- Tools like BERT, POS taggers designed for specific tasks

Input → **Processing** → **Predictable output**

- Consistent but narrow in scope



LLMs: Powerful but Probabilistic

- Generate text by predicting what comes next
- Same input can produce vastly different outputs
- Flexible capabilities but unpredictable behavior



What is prompting?

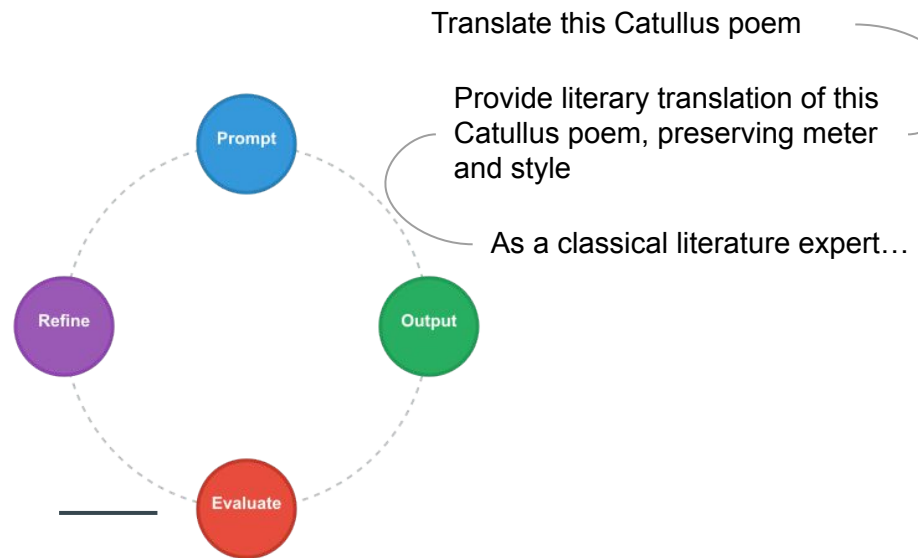
Goals

- Getting useful responses
- Evaluating the output
- Implementing safeguards

What is prompting?

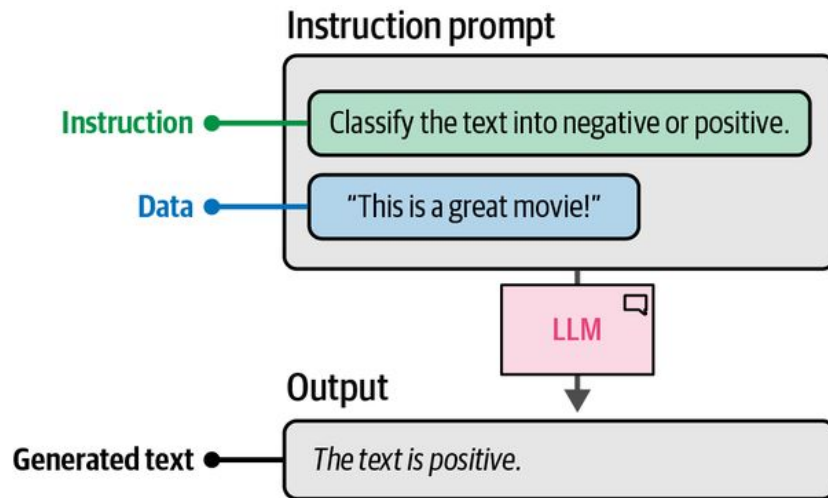
Key Properties

- Iterative process
- Requires experimentation
- Context-dependent



What is prompting?

Basic Structure



Decide if the following sentence is positive, negative, or neutral:

When, O Catiline, do you mean to cease abusing our patience?

Controlling Output Format

- Specify desired format explicitly
- Use structural cues
- Include examples

```
"Provide a morphological analysis in the following format:  
Word: [Latin word]  
Lemma: [dictionary form]  
Part of speech: [noun/verb/adjective/etc.]  
Case/Tense/etc.: [relevant grammatical information]"
```

[insert word here]

{word}

<word>

Word: gladius

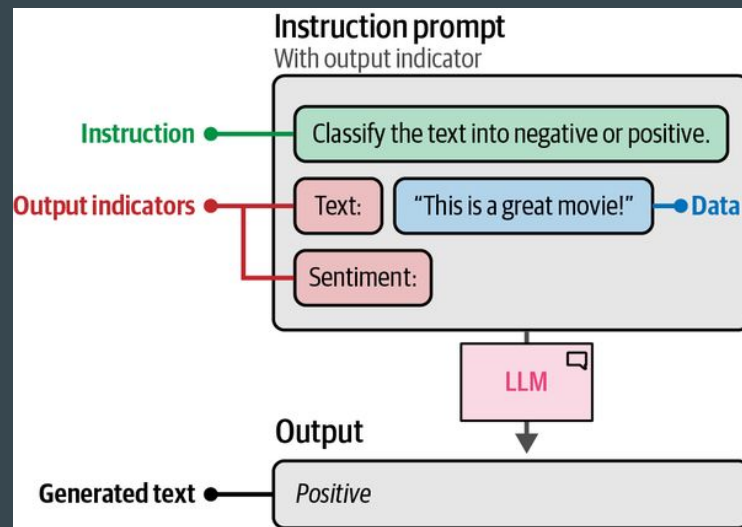


Figure: Jay Alammar, Maarten Grootendorst, Hands-On Large Language Models

Essential Rules

Specificity: Describe goals accurately



Analyze this sentence



Identify all nouns in this sentence and explain their grammatical function

Prevent hallucination

Only provide analysis if you can cite specific textual evidence

Strategic placement: Put key instructions at beginning or end of long prompts

- Beginning: Primacy effect
- End: Recency effect

Prompt Components

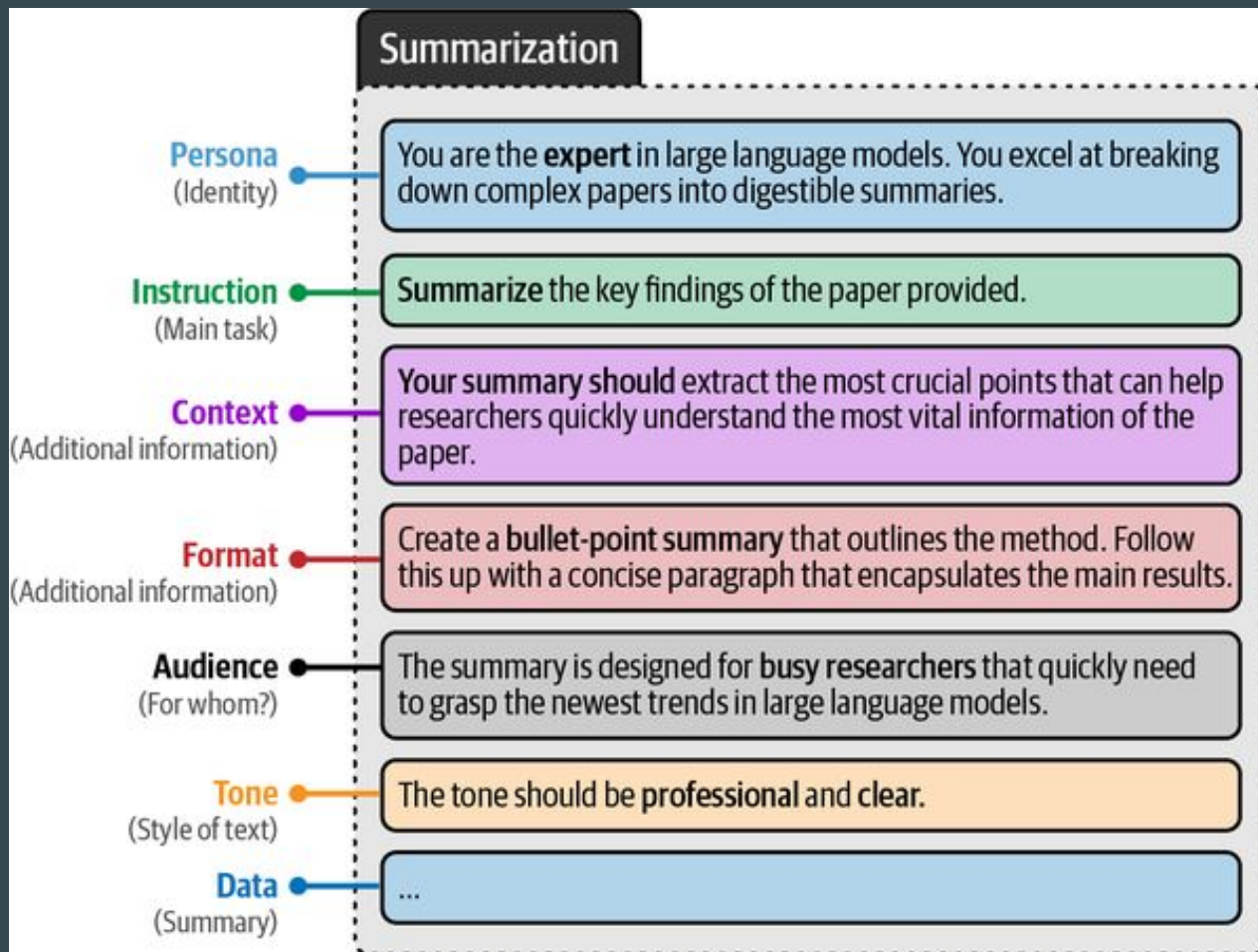


Figure: Jay Alammar, Maarten Grootendorst, Hands-On Large Language Models

Prompt Components

Persona: "You are a Latin philologist specializing in Augustan poetry"

Instruction: "Analyze the metrical patterns in the following hexameter lines"

Context: "This analysis is for a comparative study of Virgilian and Ovidian verse techniques"

Format: "Present findings as: Line number | Metrical pattern | Notable features"

Audience: "Write for graduate students in classics"

Tone: "Use scholarly but accessible language"

Data: [The actual Latin text to analyze]

Roles: User, System, Assistant

User

Provides instructions, data, examples

System

Provides high-level instructions

Assistant role

Model's response

User:

Classify the sentence as negative, positive, or neutral like in the example:

Sentence: When, O Catiline ...?

Sentiment: Negative

Sentence: How long is this madness...?

System:

You're a Latin philologist.

Assistant:

Sentiment: Negative

Roles: User, System, Assistant

User:

Classify the sentence as negative, positive, or neutral like in the example:

Sentence: When, O Catiline, do you mean to cease abusing our patience?
Sentiment: Negative

System:

You're a Latin philologist.

Assistant:

Sentiment: Negative

```
{  
  "model": "gpt-4",  
  "messages": [  
    {  
      "role": "user",  
      "content": "Classify the sentence as negative,  
        positive, or neutral like in the example:\n        Sentence: When, O Catiline, do you mean to  
        cease abusing our patience? Sentiment:  
        Negative\n        Sentence: How long is this madness  
        ...?"  
    },  
    {  
      "role": "system",  
      "content": "You're a Latin philologist."  
    },  
    {  
      "role": "assistant",  
      "content": "Sentiment: Negative"  
    }  
  ]  
}
```

In-Context Learning - Teaching by Example

Zero-shot prompt

Prompting without examples

Classify the text into neutral, negative, or positive.

Text: I think the food was okay.

Sentiment: ...

One-shot prompt

Prompting with a single example

Classify the text into neutral, negative, or positive.

Text: I think the food was alright.

Sentiment: **Neutral**

Text: I think the food was okay.

Sentiment:

Few-shot prompt

Prompting with more than one example

Classify the text into neutral, negative, or positive.

Text: I think the food was alright.

Sentiment: **Neutral**.

Text: I think the food was great!

Sentiment: **Positive**.

Text: I think the food was horrible...

Sentiment: **Negative**.

Text: I think the food was okay.

Sentiment:

Figure: Jay Alammar, Maarten Grootendorst, Hands-On Large Language Models

Chain of Thought

Prompting the AI to show its reasoning process explicitly

Zero-shot chain-of-thought

Prompting without example

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Instruction

Let's think step-by-step.

Prime reasoning

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$.

Reasoning process (thought)

The answer is 9. ✓

Chain-of-thought prompt

Prompting with a reasoning example

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Example

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.
 $5 + 6 = 11$

Reasoning process (thought)

The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Instruction

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$.

Reasoning process (thought)

The answer is 9. ✓

Figure: Jay Alammar, Maarten Grootendorst, Hands-On Large Language Models

Chain Prompting

Taking output from one prompt as input for the next

Classify the sentence into negative, positive, and neutral.

When, O Catiline, do you mean to cease abusing our patience?

Negative

Identify elements of the sentence that contribute to **Negative** sentiment

abusing, cease, when ... do you?

For each of the elements 'abusing, cease, when ... do you?' indicate part of speech.

Verb (present participle/gerund),
Verb (infinitive), Rhetorical
structure

Output Verification

Structured output

This is the famous opening line from Cicero's first oration against Catiline (Quo usque tandem abutere, Catilina, patientia nostra?), delivered in 63 BCE to the Roman Senate. Cicero was confronting Lucius Sergius Catilina, who was allegedly plotting against the Roman Republic.

The sentence expresses frustration, accusation, and condemnation toward Catiline, making it decidedly negative in sentiment despite its formal, elevated language.

```
[
  {
    "sentence" : "When, O Catiline, do
you mean to cease abusing our
patience?",
    "sentiment" : "Negative"
  },
  {
    "sentence" : "How long is that
madness of yours still to mock us?",
    "sentiment" : "Negative"
  }
]
```

Output Verification

Valid output

```
[{  
  "sentence" : "How long is that  
madness of yours still to mock  
us?",  
  "sentiment" : "Condemnation"  
}]
```

```
[{  
  "sentence" : "How long is that  
madness of yours still to mock  
us?",  
  "sentiment" : "Negative"  
}]
```

Ethics

Bias

- cultural
- gender
- linguistic

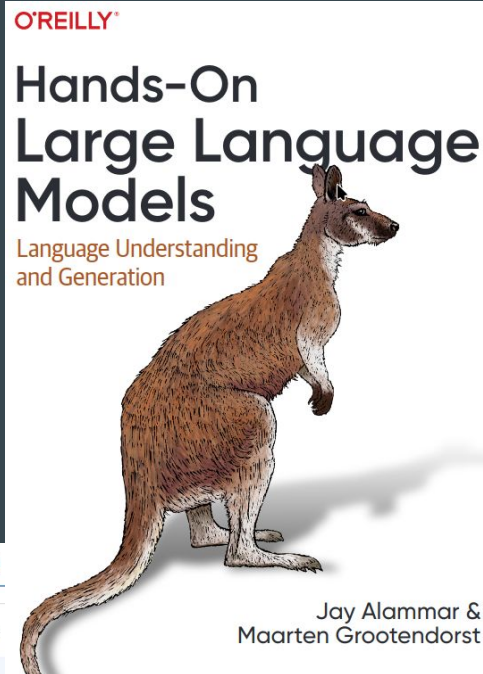
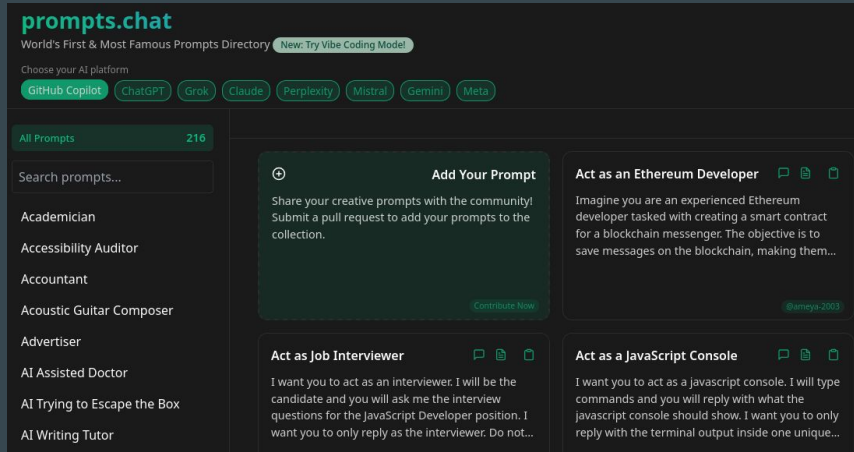
Intellectual property

- attribution
- plagiarism
- originality

Accuracy

- factually accurate
- coherent
- free from hallucination

Resources



Prompt Engineering for ChatGPT

This course is part of multiple programs. [Learn more](#)

Instructor: [Dr. Jules White](#) [Top Instructor](#)

Prompt Libraries

OpenAI Platform

Q Search

CTRL K

CORE CONCEPTS

Text and prompting

Images and vision

Audio and speech

Structured Outputs

Function calling

Conversation state

Streaming

File inputs

Reasoning

BUILT-IN TOOLS

Usina built-in tools

</> Cookbook

🗯️ Forum

🔍 Help

Prompt examples

Explore what's possible with

Q Search...



Grammar correction

Convert ungrammatical st
standard English.



Parse unstructured dat

Create tables from unstruc



Keywords

Extract keywords from a block of text.



Tweet classifier

Detect sentiment in a tweet.

ANTHROPIC

English

Q Search...

Ctrl K

Rese

Welcome

Developer Guide

API Guide

Claude Code

Resources

Release Notes

Alien anthropologist

Data organizer

Brand builder

Efficiency estimator

Review classifier

Direction decoder

Motivational muse

Email extractor

Master moderator

Lesson planner

Socratic sage

Alliteration alchemist

Futuristic fashion advisor

Polyglot superpowers

Product naming pro

Philosophical musings



Product name generator

Generate product names from a
description and seed words.



Airport code extractor

Extract airport codes from text.

PROMPT LIBRARY

Tweet tone detector

Detect the tone and sentiment behind tweets.

Copy this prompt into our developer [Console](#) to try it for yourself!

Content

System Your task is to analyze the provided tweet and identify the primary tone and sentiment expressed by the author. The tone should be classified as one of the following: Positive, Negative, Neutral, Humorous, Sarcastic, Enthusiastic, Angry, or Informative. The sentiment should be classified as Positive, Negative, or Neutral. Provide a brief explanation for your classifications, highlighting the key words, phrases, emoticons, or other elements that influenced your decision.

User Wow, I'm so impressed by the company's handling of this crisis. 🙄 They really have their priorities straight. #sarcasm #fail

Example output

Tone: Sarcastic Sentiment: Negative

Model Documentation

OpenAI Platform

Search Ctrl K

- Quickstart
- Models
- Pricing
- Libraries

CORE CONCEPTS

Text and prompting

- Images and vision
- Audio and speech
- Structured Outputs
- Function calling
- Conversation state
- Streaming
- File inputs
- Reasoning

Cookbook

Forum

Help

Few-shot learning

Few-shot learning lets you steer a large language model toward a new task by including a handful of input/output examples in the prompt, rather than [fine-tuning](#) the model. The model implicitly "picks up" the pattern from those examples and applies it to a prompt. When providing examples, try to show a diverse range of possible inputs with the desired outputs.

Typically, you will provide examples as part of a `developer` message in your API request. Here's an example `developer` message containing examples that show a model how to classify positive or negative customer service reviews.

```
1 # Identity
2
3 You are a helpful assistant that labels short product reviews as
4 Positive, Negative, or Neutral.
5
6 # Instructions
7
8 * Only output a single word in your response with no additional formatting
9   or commentary.
10 * Your response should only be one of the words "Positive", "Negative", or
11   "Neutral" depending on the sentiment of the product review you are given.
12
13 # Examples
14
15 <product_review id="example-1">
16 I absolutely love this headphones - sound quality is amazing!
17 </product_review>
```

ANTHROPIC

English

Search...

Ctrl K

Research

- Welcome
- Developer Guide
- API Guide
- Claude Code
- Resources
- Release Notes

Models overview

Pricing

Model deprecations

Crafting effective examples

For maximum effectiveness, make sure that your examples are:

- Relevant:** Your examples mirror your actual use case.
- Diverse:** Your examples cover edge cases and potential challenges, and vary enough that Claude doesn't inadvertently pick up on unintended patterns.
- Clear:** Your examples are wrapped in `<example>` tags (if multiple, nested within `<examples>` tags) for structure.

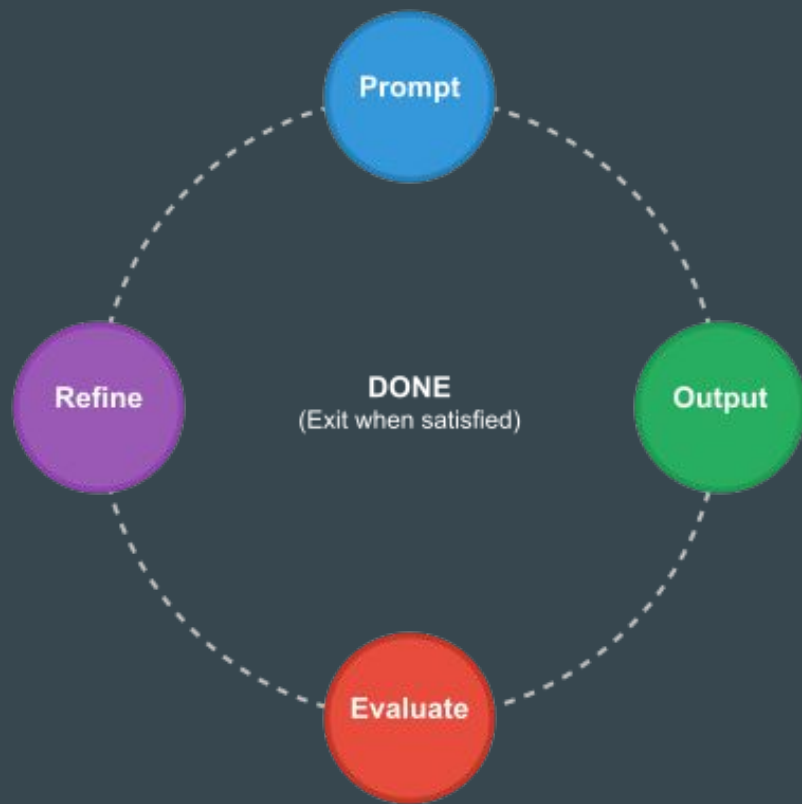


Ask Claude to evaluate your examples for relevance, diversity, or clarity. Or have Claude generate more examples based on your initial set.

Example: Analyzing customer feedback

Notice that in the "no examples" version, Claude does not list multiple options for each category, despite its written analysis indicating that there are multiple categories for certain feedback. It also includes a long explanation when we may not want it to.

Role	No Examples	With Examples
User	Analyze this customer feedback and categorize the issues. Use these categories: UI/UX, Performance, Feature Request, Integration, Pricing, and Other. Also rate the sentiment (Positive/Neutral/Negative) and priority (High/Medium/Low). Here is the feedback: <code>{{FEEDBACK}}</code>	Our CS team is overwhelmed with unstructured feedback. Your task is to analyze feedback and categorize issues for our product and engineering teams. Use these categories: UI/UX, Performance, Feature Request, Integration, Pricing, and Other. Also rate the sentiment (Positive/Neutral/Negative) and priority (High/Medium/Low). Here is an example:





1. Prompt Component Analysis:
What Elements Matter Most?
2. Learning by example: Zero-shot
and Few-Shot Learning
3. Taming the LLM: Restraining the
Output
4. Explain me like I'm Five: Chain
of Thought
