

# Prompt Engineering

COMP4901Y

Binhang Yuan

# Overview of Prompt Engineering

# What Are Prompts?

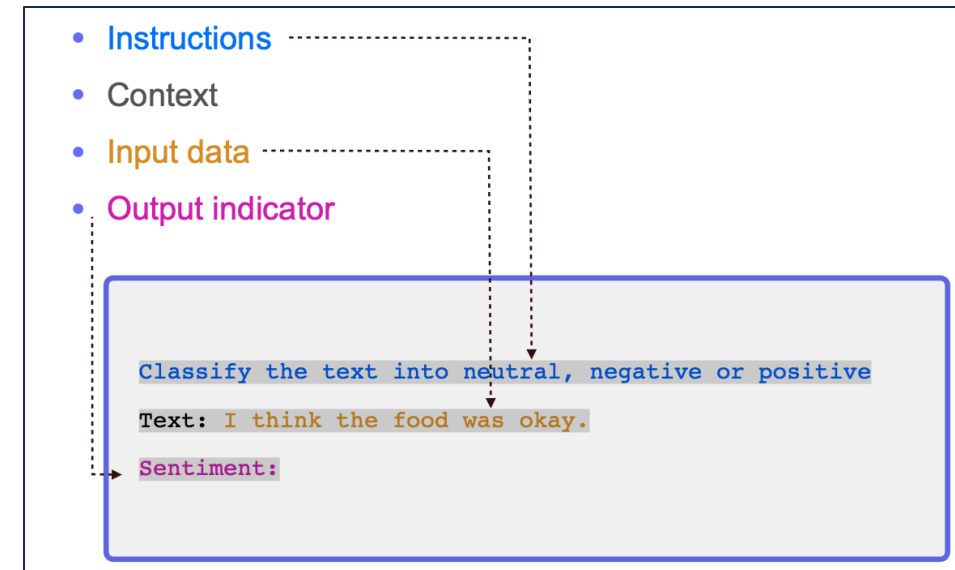
- *Prompts* involve instructions and context passed to a language model to accomplish a desired task.
- *Prompt engineering* is the practice of developing and optimizing prompts to efficiently use large language models (LLMs) for a variety of applications.
- Prompt engineering is a useful skill for AI engineers and researchers to improve and efficiently use language models.

# Why Prompt Engineering?

- Prompt engineering is a relatively new discipline for developing and optimizing prompts to efficiently use language models (LMs) for a wide variety of applications and research topics.
- Prompt engineering skills help to better understand the capabilities and limitations of large language models (LLMs).
- Researchers use prompt engineering to improve the capacity of LLMs on a wide range of common and complex tasks such as question answering and arithmetic reasoning.
- Developers use prompt engineering to design robust and effective prompting techniques that interface with LLMs and other tools.

# Elements of a Prompt

- A prompt is composed with the following components:
  - **Instruction:** a specific task or instruction you want the model to perform;
  - **Context:** external information or additional context that can steer the model to better responses;
  - **Input data:** the input or question that we are interested to find a response for;
  - **Output indicator:** the type or format of the output.
- You do not need all the four elements for a prompt and the format depends on the task at hand.



# Rules of Thumb from OpenAI

# Rule 1

- “Use the latest model.”
  - For best results, we generally recommend using the latest, most capable models. Newer models tend to be easier to prompt engineer.

## GPTs

Discover and create custom versions of ChatGPT that combine instructions, extra knowledge, and any combination of skills.

Top Picks

DALL·E

Writing

Productivity

Research & Analysis


Programming

Education


Lifestyle

### Featured


Curated top picks from this week




**Instant Website**  
**[Multipage]**  
Generates Functional Multipage Websites [in BETA]. Our mission is to simplify the creation of a...  
By Max & Kirill Dubovitsky




**AskYourPDF Research Assistant**  
Free Chat Unlimited PDFs, Access 400M+ Papers (PubMed, Nature, Arxiv, etc), Analyse PDF (Unlimite...  
By askyourpdf.com




**Diagrams & Data:**  
**Research, Analyze,...**  
Complex Visualizations (Diagram & Charts), Data Analysis & Reseach. For Coders: Visualize Databases,...  
By Max & Kirill Dubovitsky



**ChatPRD**  
An on-demand Chief Product Officer that drafts and improves your PRDs, while coaching you to...  
By Claire V Lawless



**Music Teacher**  
Regular ChatGPT isn't great at music theory and relative scales, so I trained Music Teacher to be an...  
By gryphonedm.com



**UX Design Mentor**  
I provide specific UX or Product Design feedback.  
By community builder

# Rule 2

- “Put instructions at the beginning of the prompt and use #### or """ to separate the instruction and context”

- Less effective :

Summarize the text below as a bullet point list of the most important points.

{text input here}

- Better :

Summarize the text below as a bullet point list of the most important points.

Text: """

{text input here}

"""



# Rule 3

- ‘Be specific, descriptive and as detailed as possible about the desired context, outcome, length, format, style, etc’

- Less effective :

Write a poem about OpenAI.

- Better :

Write a short inspiring poem about OpenAI, focusing on the recent DALL-E product launch (DALL-E is a text to image ML model) in the style of a {famous poet}

# Rule 4

- “Articulate the desired output format through examples”
  - Show, and tell - the models respond better when shown specific format requirements. This also makes it easier to programmatically parse out multiple outputs reliably.

- Less effective :

Extract the entities mentioned in the text below. Extract the following 4 entity types: company names, people names, specific topics and themes.

Text: {text}

- Better :

Extract the important entities mentioned in the text below. First extract all company names, then extract all people names, then extract specific topics which fit the content and finally extract general overarching themes

Desired format:

Company names:

<comma\_separated\_list\_of\_company\_names>

People names: -| |-

Specific topics: -| |-

General themes: -| |-

Text: {text}

# Rule 5


- “Start with zero-shot, then few-shot, neither of them worked, then fine-tune.”

-  Zero-shot:

Extract keywords from the below text.

Text: {text}

Keywords:

-  Fine-tune: to be discussed later in the lecture.

-  Few-shot - provide a couple of examples:

Extract keywords from the corresponding texts below.

Text 1: Stripe provides APIs that web developers can use to integrate payment processing into their websites and mobile applications.

Keywords 1: Stripe, payment processing, APIs, web developers, websites, mobile applications

##

Text 2: OpenAI has trained cutting-edge language models that are very good at understanding and generating text. Our API provides access to these models and can be used to solve virtually any task that involves processing language.

Keywords 2: OpenAI, language models, text processing, API.

##

Text 3: {text}

Keywords 3:

# Rule 6

- “Reduce “fluffy” and imprecise descriptions.”

- Less effective :

The description for this product should be fairly short, a few sentences only, and not too much more.

- Better :

Use a 3 to 5 sentence paragraph to describe this product.

# Rule 7

- “Instead of just saying what not to do, say what to do instead.”

- Less effective :

The following is a conversation between an Agent and a Customer. DO NOT ASK USERNAME OR PASSWORD. DO NOT REPEAT.

Customer: I can't log in to my account.

Agent:

- Better :

The following is a conversation between an Agent and a Customer. The agent will attempt to diagnose the problem and suggest a solution, whilst refraining from asking any questions related to PII. Instead of asking for PII, such as username or password, refer the user to the help article [www.samplewebsite.com/help/faq](http://www.samplewebsite.com/help/faq)

Customer: I can't log in to my account.

Agent:

# Rule 8

- “Code Generation Specific - Use “leading words” to nudge the model toward a particular pattern.”
  - In this code example below, adding “import” hints to the model that it should start writing in Python. (Similarly “SELECT” is a good hint for the start of a SQL statement.)

- Less effective :

```
# Write a simple python function that  
# 1. Ask me for a number in mile  
# 2. It converts miles to kilometers
```

- Better :

```
# Write a simple python function that  
# 1. Ask me for a number in mile  
# 2. It converts miles to kilometers  
  
import
```


# Advanced Prompt Engineering Techniques

# Baseline: Zero-Shot Prompting

- Large-scale training makes LLMs capable of performing some tasks in a "zero-shot" manner.
- Zero-shot prompting means that the prompt used to interact with the model won't contain examples or demonstrations.
- The zero-shot prompt directly instructs the model to perform a task without any additional examples to steer it.



# Few-Shot Prompting

- *Few-shot prompting* allows us to provide exemplars in prompts to steer the model towards better performance.
  - Few-shot prompting can be used as a technique to enable in-context learning where we provide demonstrations in the prompt to steer the model to better performance.
  - The demonstrations serve as conditioning for subsequent examples where we would like the model to generate a response.
- Few-Shot Prompting Input:
- Output :

Great product, 10/10: positive  
Didn't work very well: negative  
Super helpful, worth it: positive  
It doesnt work!:

negative

# Chain-of-Thought Prompting

- *Chain-of-thought (CoT)* prompting enables complex reasoning capabilities through intermediate reasoning steps.
  - This is very useful for tasks that require reasoning;
  - It can be combined with both few-shot prompting and zero-shot prompting.

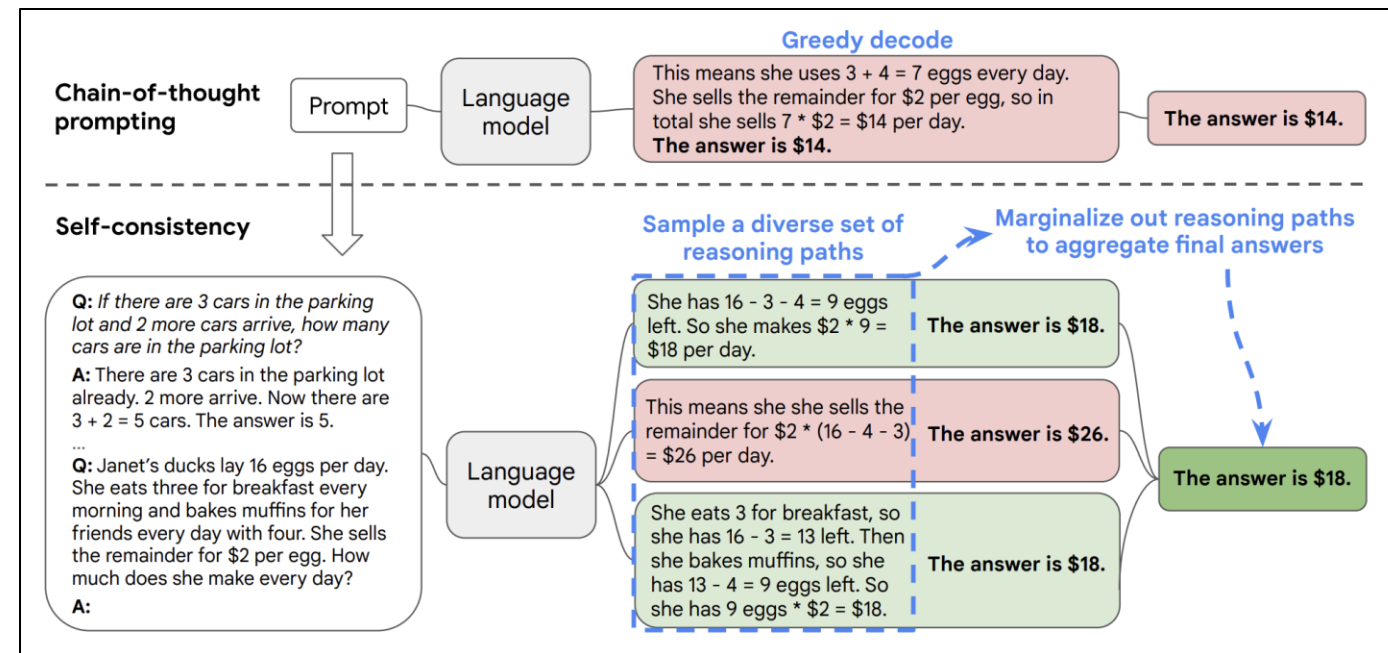
| (a) Few-shot  | (b) Few-shot-CoT  |
|---|---|
| <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?<br/>A: The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?<br/>A:</p> <p>(Output) The answer is 8. <b>X</b></p> | <p>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?<br/>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. <math>5 + 6 = 11</math>. The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?<br/>A:</p> <p>(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are <math>16 / 2 = 8</math> golf balls. Half of the golf balls are blue. So there are <math>8 / 2 = 4</math> blue golf balls. The answer is 4. <b>✓</b></p> |
| (c) Zero-shot   | (d) Zero-shot-CoT (Ours)  |
| <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?<br/>A: The answer (arabic numerals) is</p> <p>(Output) 8 <b>X</b></p>   | <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?<br/>A: <b>Let's think step by step.</b></p> <p>(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. <b>✓</b></p>   |

# Self-Consistency

- **Self-consistency** aims to improve the naive greedy decoding used in chain-of-thought prompting.
  - The idea is to sample multiple, diverse reasoning paths through few-shot CoT, and use the generations to select the most consistent answer.

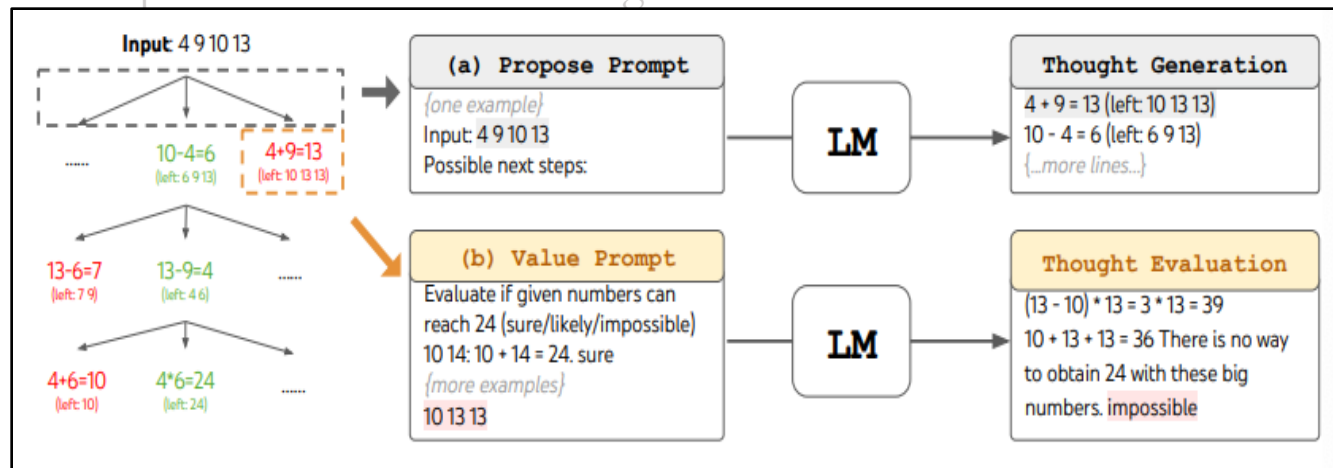
## Three steps:

1. Prompt a language model using chain-of-thought (CoT) prompting.
2. Replace the “greedy decode” in CoT prompting by sampling from the language model’s decoder to generate a diverse set of reasoning paths.
3. Marginalize the reasoning paths and aggregate by choosing the most consistent answer in the final answer set.

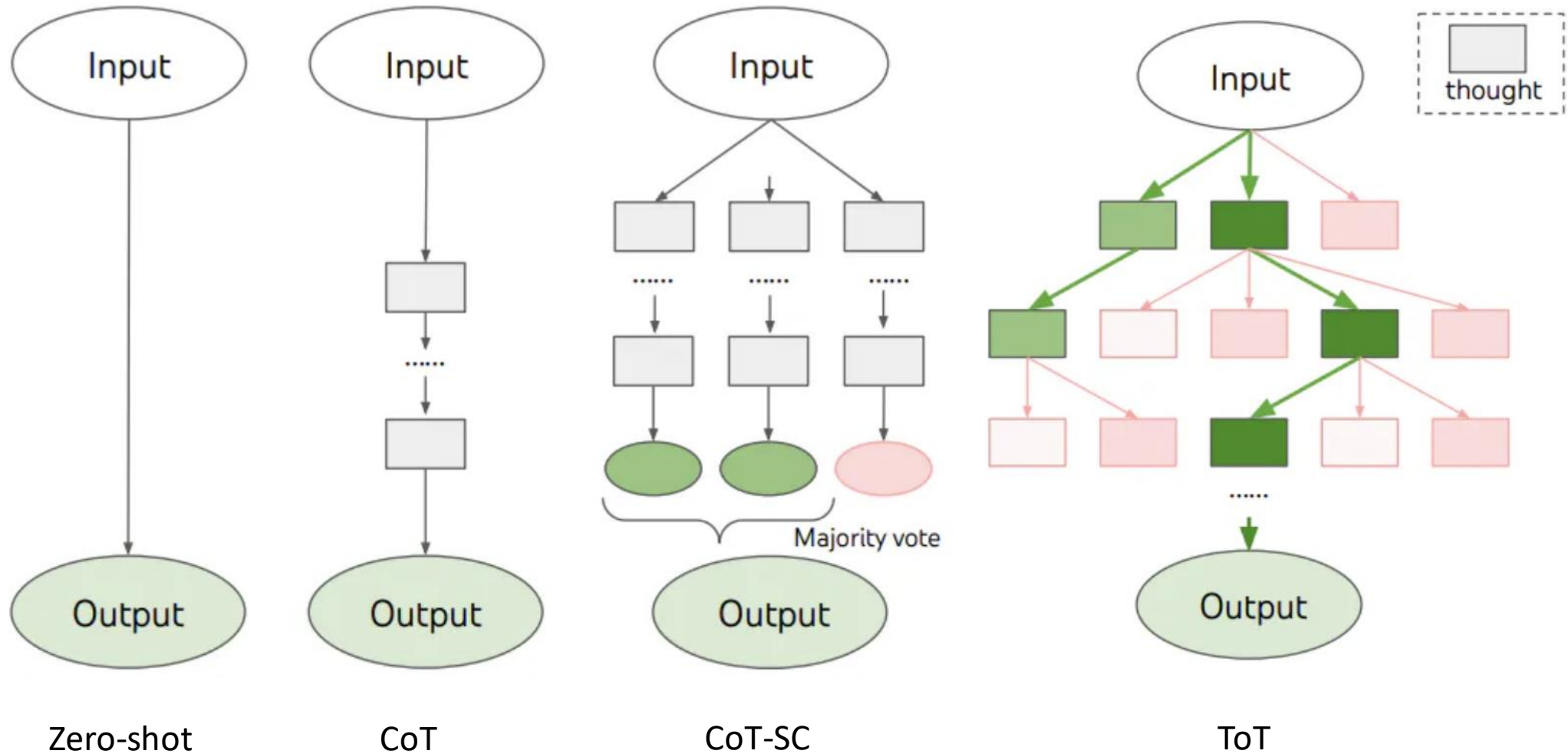


# Tree of Thoughts (ToT)

- **Tree of Thoughts (ToT)** maintains a tree of thoughts where thoughts represent coherent language sequences that serve as intermediate steps to solve a problem.
- The LM's ability to generate and evaluate thoughts is then combined with search algorithms (e.g., breadth-first search and depth-first search) to enable systematic exploration of thoughts with lookahead and backtracking.
- Example: Game of 24:
  - To perform BFS in ToT for the Game of 24 task, the LLM is prompted to evaluate each thought candidate as sure/maybe/impossible w.r.t reaching 24.
  - The aim is to promote correct partial solutions that can be verified within a few lookahead trials, eliminate impossible partial solutions based on too big/small commonsense, and keep the rest maybe.
  - Values are sampled 3 times for each thought.



# Some Comparison and Summary



# References

- <https://www.promptingguide.ai/>
- <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api>
- <https://arxiv.org/pdf/2203.11171.pdf>
- <https://arxiv.org/pdf/2305.10601.pdf>