

[COMSE6998-015] Fall  
2024

# Introduction to Deep Learning and LLM based Generative AI Systems

# Agenda

- Prompt Engineering and LLM App Development
  - Use Cases of LLM
  - Prompt Engineering
  - Prompt Engineering Techniques
  - LLM App Development Frameworks

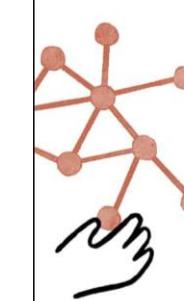


# What are the LLM Products you have used?



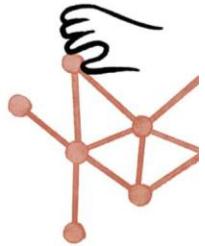
**ChatGPT**

**Gemini**



**Meet  
Claude**

A next-generation AI assistant for your tasks, no matter the scale



# Prompts and completions

## Prompt

*What is Large Language Model?*

Context window: 2k – 100K

## Model

LLM

## Completion

*What is Large Language Model?*

A **Large Language Model (LLM)** is a type of artificial intelligence (AI) model designed to understand, generate, and manipulate human language on a large scale. These models are typically built using deep learning techniques, particularly neural networks, and are trained on massive amounts of text data to learn the statistical patterns and structures of language.



# What are the LLM use cases you have seen?

- Other Use Cases
  - Translate
  - Summarize
  - Proof-read and correct
  - Explain words
  - Create article/email
  - Make restaurant suggestions
  - Chat with users
  - Do Math Calculations
  - Answer Questions on many subjects
  - ...



Certainly! Here's a di

 **Simplified**

Subje

Dear F

I hope

As Th  
guida  
deeply

Wishir

Thank

Best r

[Your Name]

Could you help me draft an email saluting my PhD advisor Prof. Kim for Thanksgiving?

## ChatGPT Writer

Free Chrome extension to generate entire emails and messages using ChatGPT AI. All sites are supported and enhanced support for Gmail.

 Download Free Extension



70,000+ happy users who don't like writing emails

- ✓ Highest quality responses that outcompete all other tools in the market
- ✓ Works in all Chromium-based browsers, such as Chrome, Brave, and Edge



SMARTLING

.ation



pairaplr



Google  
Translate

translate "塞翁失马，焉知非福" in English...



"... たまに うれしい こと が ある わよ。" can be translated



amazon Translate

"... a fable about



Falcon LLM

that who

that what



# GitHub Copilot



## Amazon Q Developer



Meet Cody,  
the most powerful  
& accurate AI coding  
assistant

```
app.run(debug=True)
```

tabnine

AI Assistant For Software Developers

Write Flask code  
response with 1  
request.

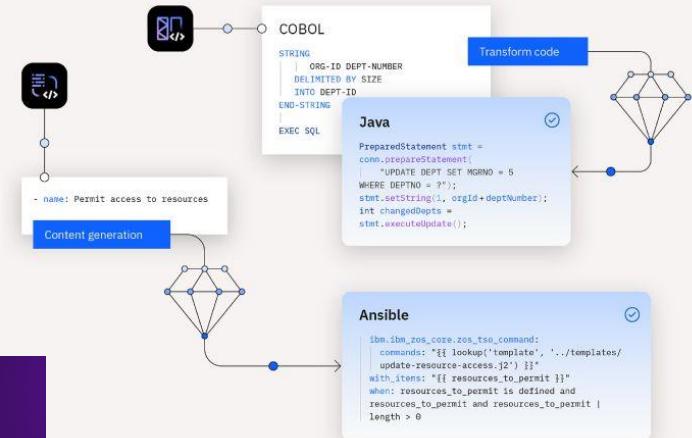
a simple Flask API that returns a JSON  
based via a GET request:

python

```
from flask import Flask, jsonify
```

IBM

## watsonX Code Assistant



CURSOR  
AI

@bindai

Boost productivity with  
the leading AI notetaker



# upword\*

The benchmark aims to push the boundaries of LM capabilities in practical and autonomous software engineering, providing a realistic and sustainable evaluation framework that can be continuously updated with new task instances.

## Summarize Documents in Seconds

Easily get a Summary of DOCX files as large as 500 pages or 150K words.

Select DOCX,PDF File &  
Choose Summary Size

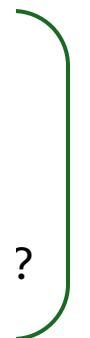
Read or Download  
Summary Easily



ironments.

on SWE-bench, including proprietary models and their training one (Claude 2), can resolve only a small fraction

d SWE-bench-train, which consists of 19,000 non-testing L on CodeLlama, which are specialized for repository-wide



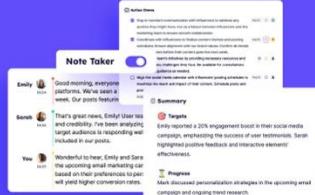
en  
s

# krisp

## AI Meeting Assistant

- ✓ Free unlimited transcriptions
- ✓ Meeting notes and summaries
- ✓ Meeting audio recording

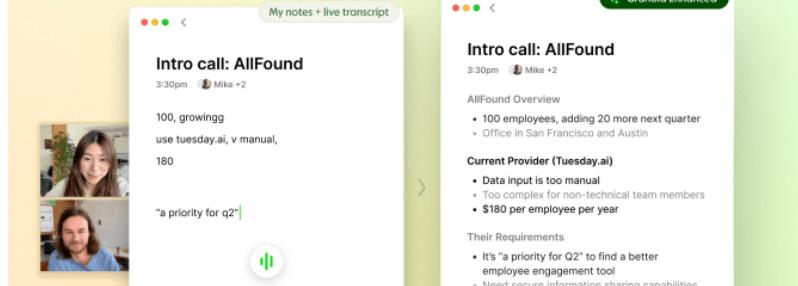
[Learn more about Krisp AI Meeting Assistant](#)



# Meeting Summarization Tool

# granola

The AI Notepad for people in back-to-back meetings



# Circleback

Automated meeting notes and follow-ups powered by AI.

Never take meeting notes again.

All-driven meeting notes and action items you can trust.  
[Try for free](#)

One second to record.

One second to review.

One second to act.

OII-O  
Otter.ai

# notta



Can you proof-read and correct this sentence?  
to the market to buy some fruits and vegetables.'



and vegetables





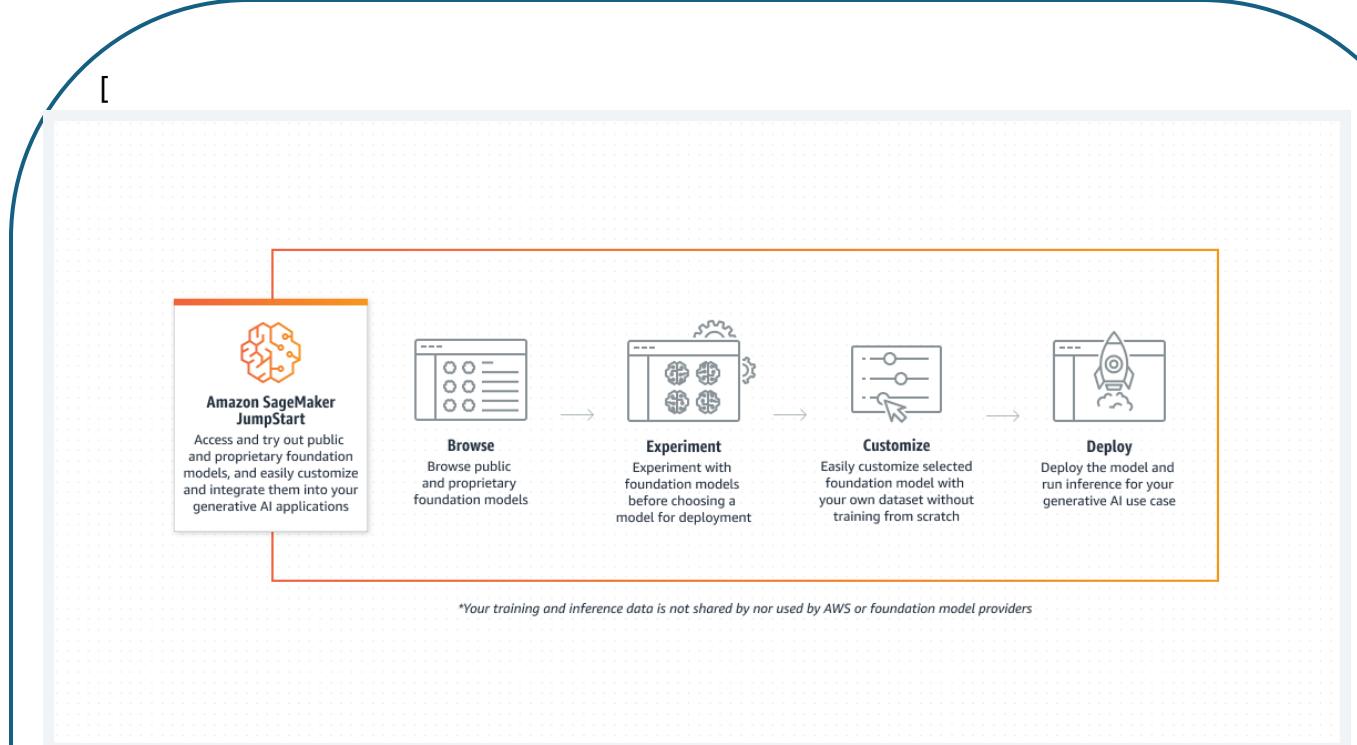
ction

Extract the following paragraph, including names of people, organizations, locations, dates, and any other relevant information. For each entity, provide its type (e.g., person, organization, location, date) and the text associated with it. Output the results in a list of JSON objects, where each object contains the entity and its type.

Ion Stoica (born 1964 or 1965) is a Romanian–American computer scientist specializing in distributed systems, cloud computing and computer networking.[9][2][10][11] He is a professor of computer science at the University of California, Berkeley, and director of AMPLab. He co-founded Conviva and Databricks with other original developers of Apache Spark.[6][12] As of April 2022, Forbes ranked him and Matei Zaharia as the 3rd-richest people in Romania with a net worth of \$1.6 billion.[13]



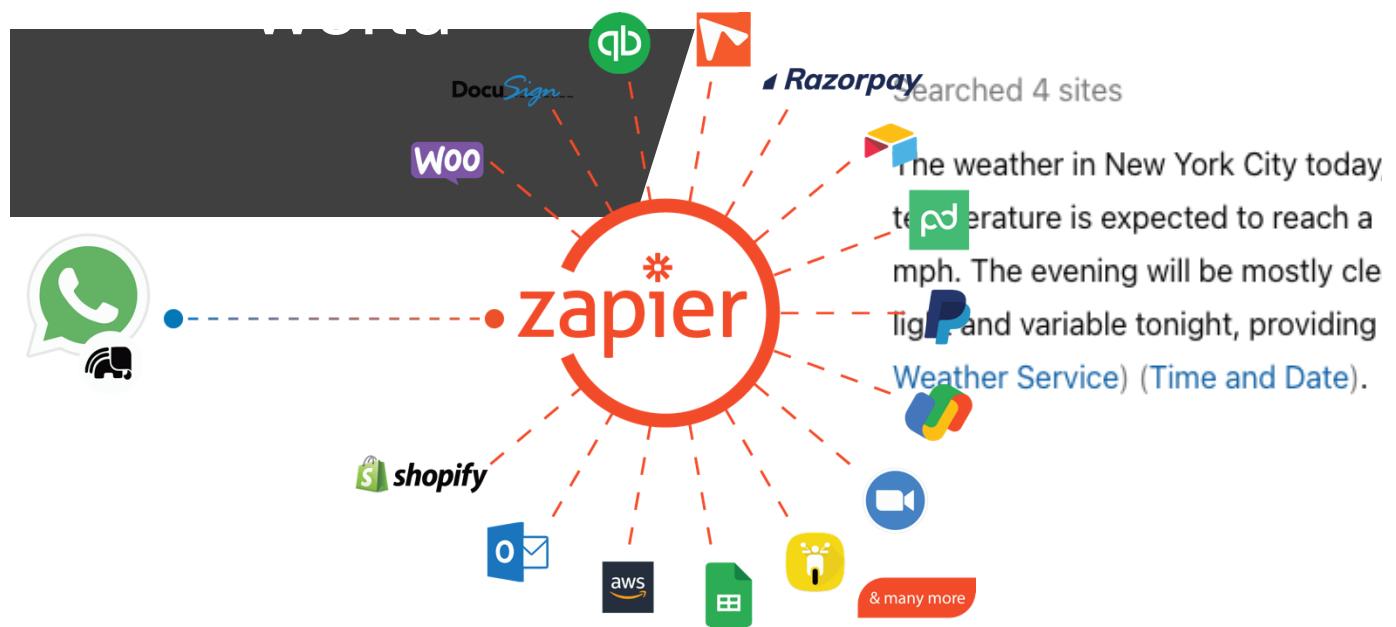
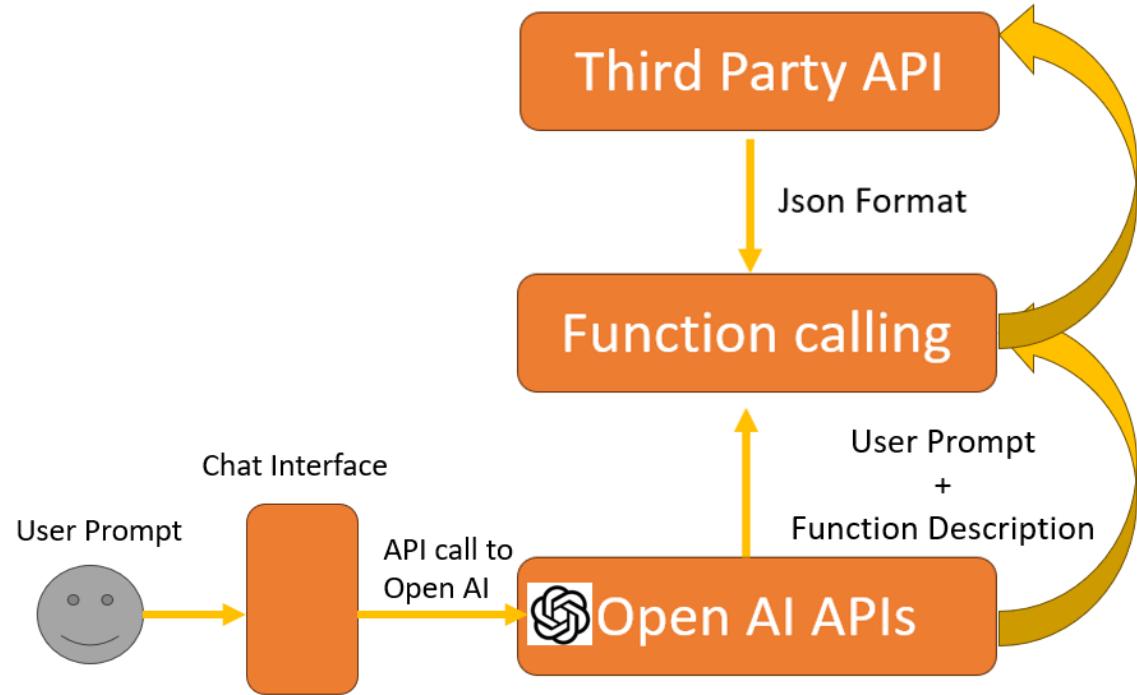
SINTELIX



```
"type": "occupation",
"text": "computer scientist"
},
{
"entity": "distributed system
ield",
istributed systems"
```

'University of Califor
organization",
niversity of Califorr





13

# Math Calculation

Let's solve this step-by-step:

1) First, let's set up the multiplication:

$$\begin{array}{r} 237 \\ \times 56 \\ \hline \end{array}$$



2) We'll multiply 237 by 6 (the ones digit of 56):

$$237 \times 6 = 1422$$

3) Now we'll multiply 237 by 50 (the tens digit of 56):

$$237 \times 50 = 11850$$

4) Now we add these two results:

$$\begin{array}{r} 1422 \\ + 11850 \\ \hline 13272 \end{array}$$

Therefore,  $237 \times 56 = 13,272$ .

Would you like me to explain any part of this calculation in more detail?

To multiply 237 by 56, you can follow these steps:

### Step 1: Break down the numbers

- Write the numbers in a column format:

$$\begin{array}{r} 237 \\ \times 56 \\ \hline \end{array}$$

### Step 2: Multiply 237 by 6 (the ones digit of 56)

-  $(237 \times 6 = 1422)$

### Step 3: Multiply 237 by 5 (the tens digit of 56), but remember it's actually 50

-  $(237 \times 50 = 11850)$

- You can calculate it as:

-  $(237 \times 5 = 1185)$

- Then, multiply by 10 to get  $(11850)$

### Step 4: Add the results of the two multiplications

- Align the results according to their place value:

$$\begin{array}{r} 1422 \\ + 11850 \\ \hline 13332 \end{array}$$

### Final Result

The product of 237 and 56 is **13,332**.



# Prompting and Prompt Engineering

## Prompt

*What is Large Language Model?*

Context window: 2k – 100K

## Model

LLM

## Completion

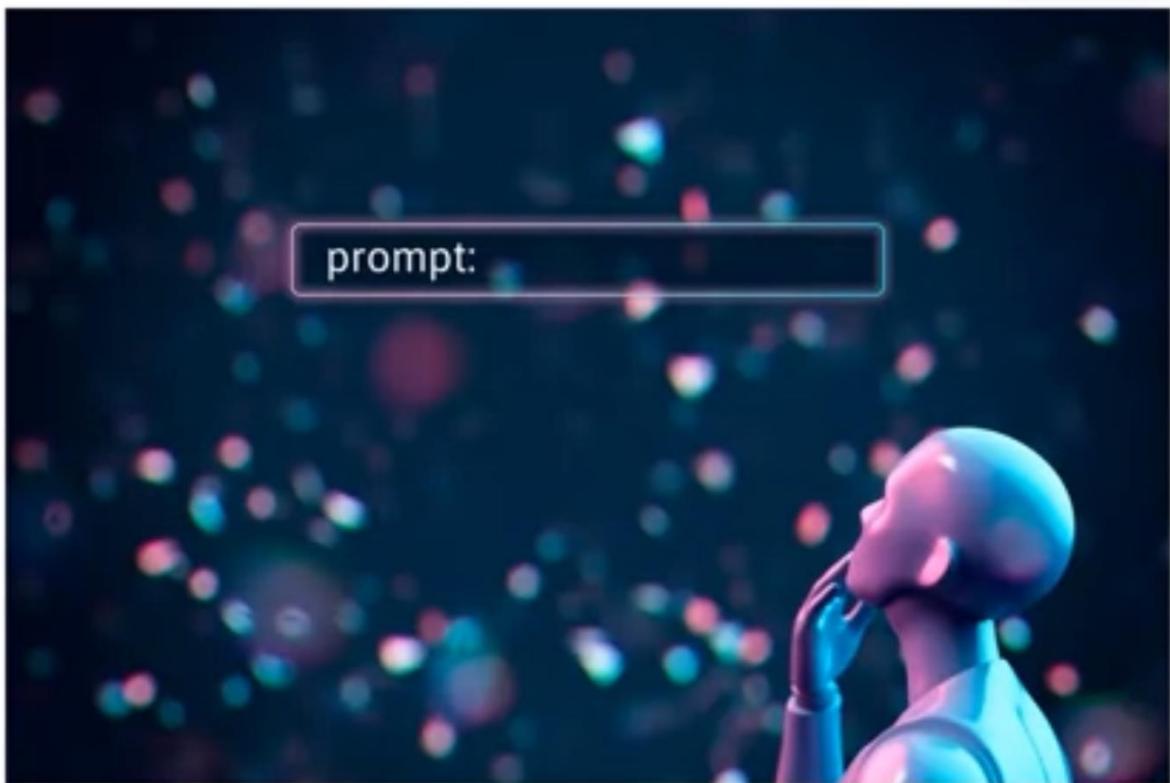
*What is Large Language Model?*

A **Large Language Model (LLM)** is a type of artificial intelligence (AI) model designed to understand, generate, and manipulate human language on a large scale. These models are typically built using deep learning techniques, particularly neural networks, and are trained on massive amounts of text data to learn the statistical patterns and structures of language.

You may have to revise the language in your prompt or the way that it's written several times to get the model to behave in the way that you want. This work to develop and improve the prompt is known as **prompt engineering**.

# What is a prompt?

---



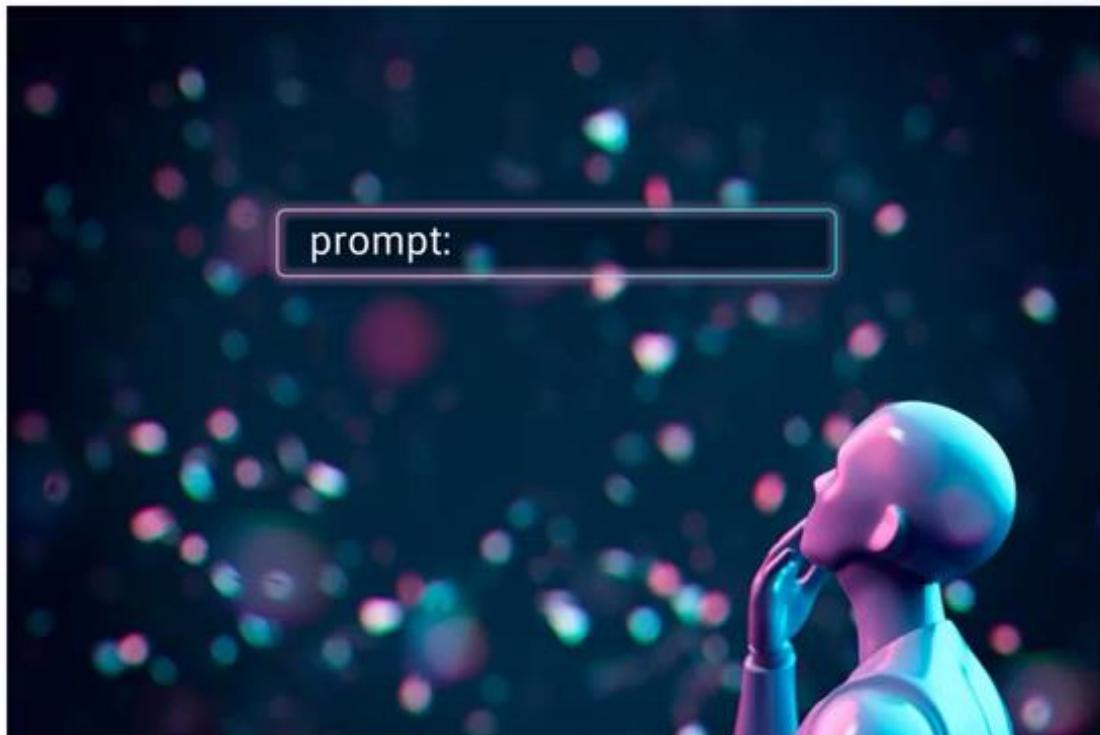
A prompt is an input that produces the desired output.

Example:

- Write a small paragraph describing your favorite holiday destination.
- Write HTML code to generate a dropdown selection of cities within an online form.

# What is a prompt?

---



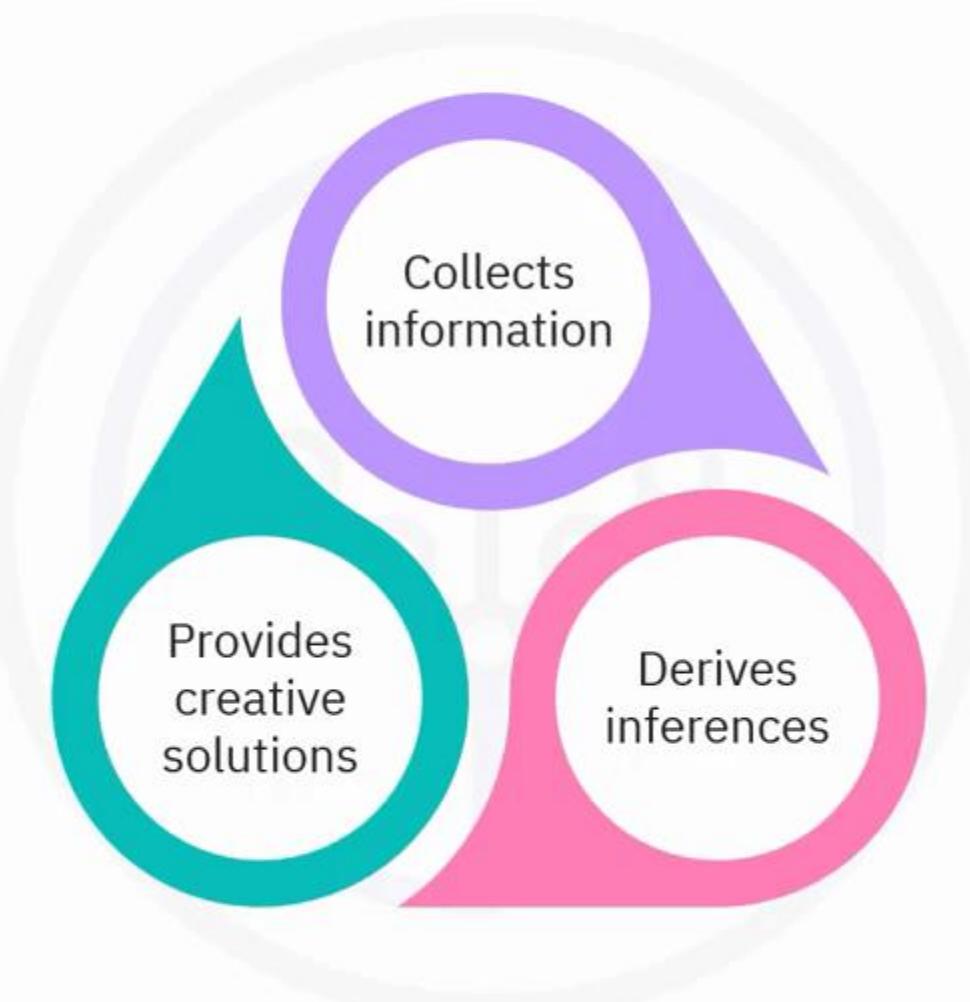
Prompts can also be a series of instructions.

Example:

- Write a short story about a scientist studying life on Mars.
- What were some of the challenges he faced during his research?

# What is a prompt?

---



# Correct and incorrect prompts

Rich man's story from a small town, his struggles, and achievements.



Write a short story about the struggles and achievements of a farmer who became a rich and influential businessman in 10 years.



Context

Proper structure

Comprehensible

# Correct and incorrect prompts

---



# Correct and incorrect prompts

Sunset image  
between  
mountains.



Too brief

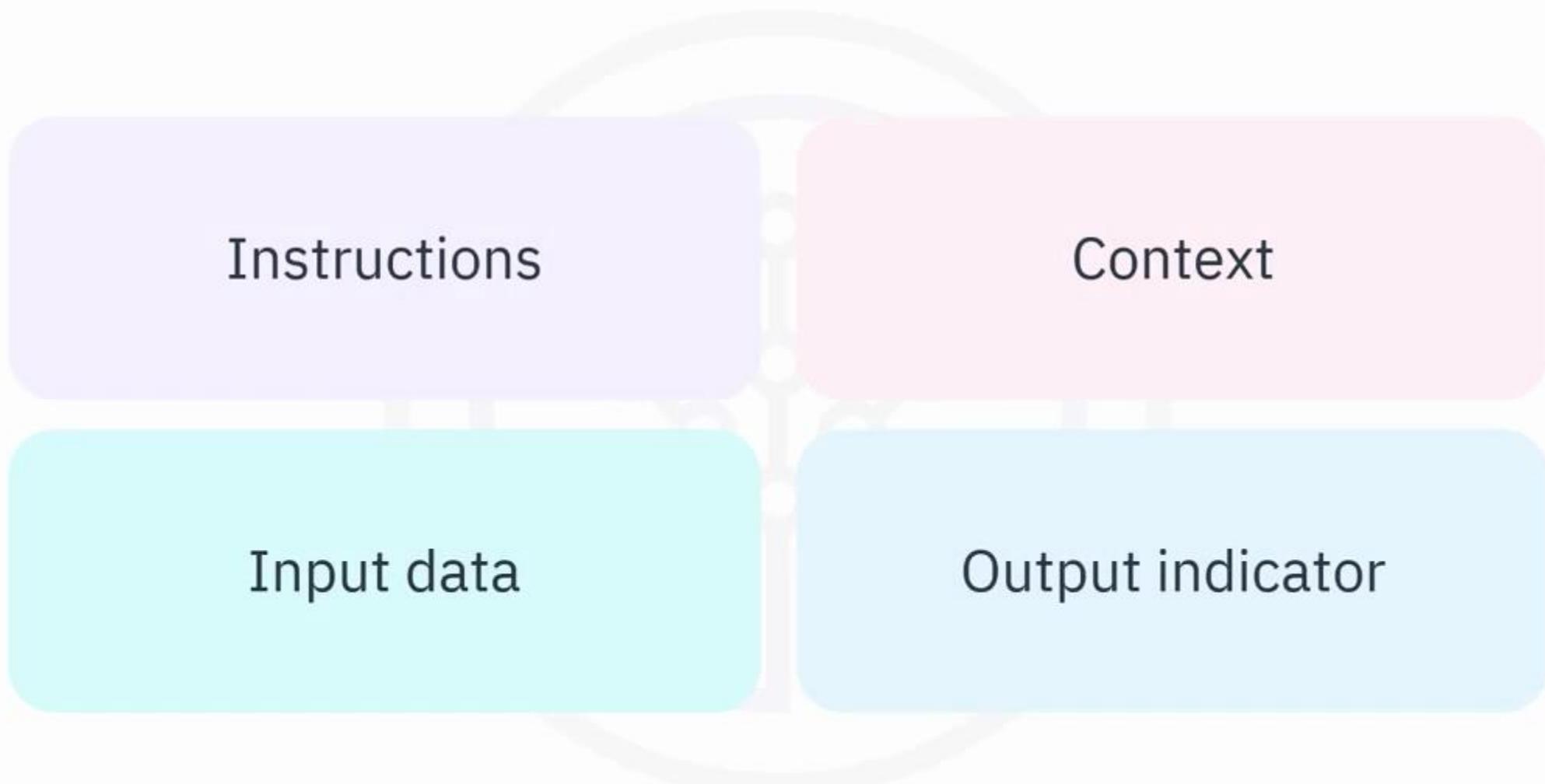


Lacks detailed  
outline

Generate an image  
depicting a calm  
sunset above a river  
valley that rests  
amidst mountains.

# Building blocks of a well-constructed prompt

---



# Building blocks of a well-constructed prompt

---



**Instructions:** Give distinct guidelines regarding the task

**Example:**

Write an essay in 600 words analyzing the effects of global warming on marine life.

# Building blocks of a well-constructed prompt

---



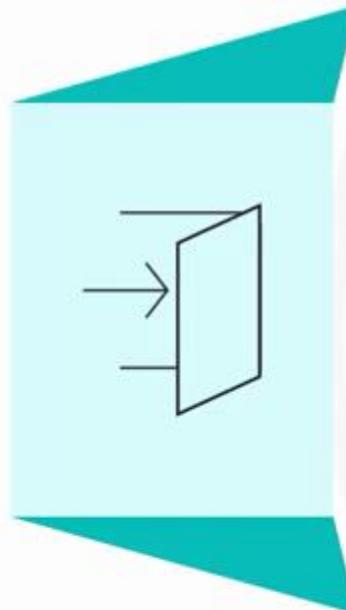
**Context:** Provides a framework for generating relevant content

**Example:**

In recent decades, global warming has undergone significant shifts, leading to rising sea levels, increased storm intensity, and changing weather patterns. These changes have had a severe impact on marine life. Write an essay in 600 words analyzing the effects of global warming on marine life.

# Building blocks of a well-constructed prompt

---



**Input data:** Any piece of information provided as part of prompt

**Example:**

You have been provided with a data set containing temperature records and measurements of sea levels in the Pacific Ocean. Write an essay in 600 words analyzing the effect of global warming on marine life in the Pacific Ocean.

# Building blocks of a well-constructed prompt

---



**Output indicator:** Offers benchmarks for assessing attributes of the output

**Example:**

The output generated should be an essay of 600 words. It will be evaluated based on the clarity of analysis and incorporation of relevant data or case studies.

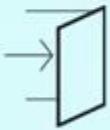
# Building blocks of a well-constructed prompt



**Instructions:** Give distinct guidelines regarding the task



**Context:** Provides framework for generating relevant content



**Input data:** Any piece of information provided as part of prompt



**Output indicator:** Offers benchmarks for assessing attributes of the output

# What Is Prompt Engineering?

---

# Prompt engineering

---

Process of designing effective prompts.

Critical analysis

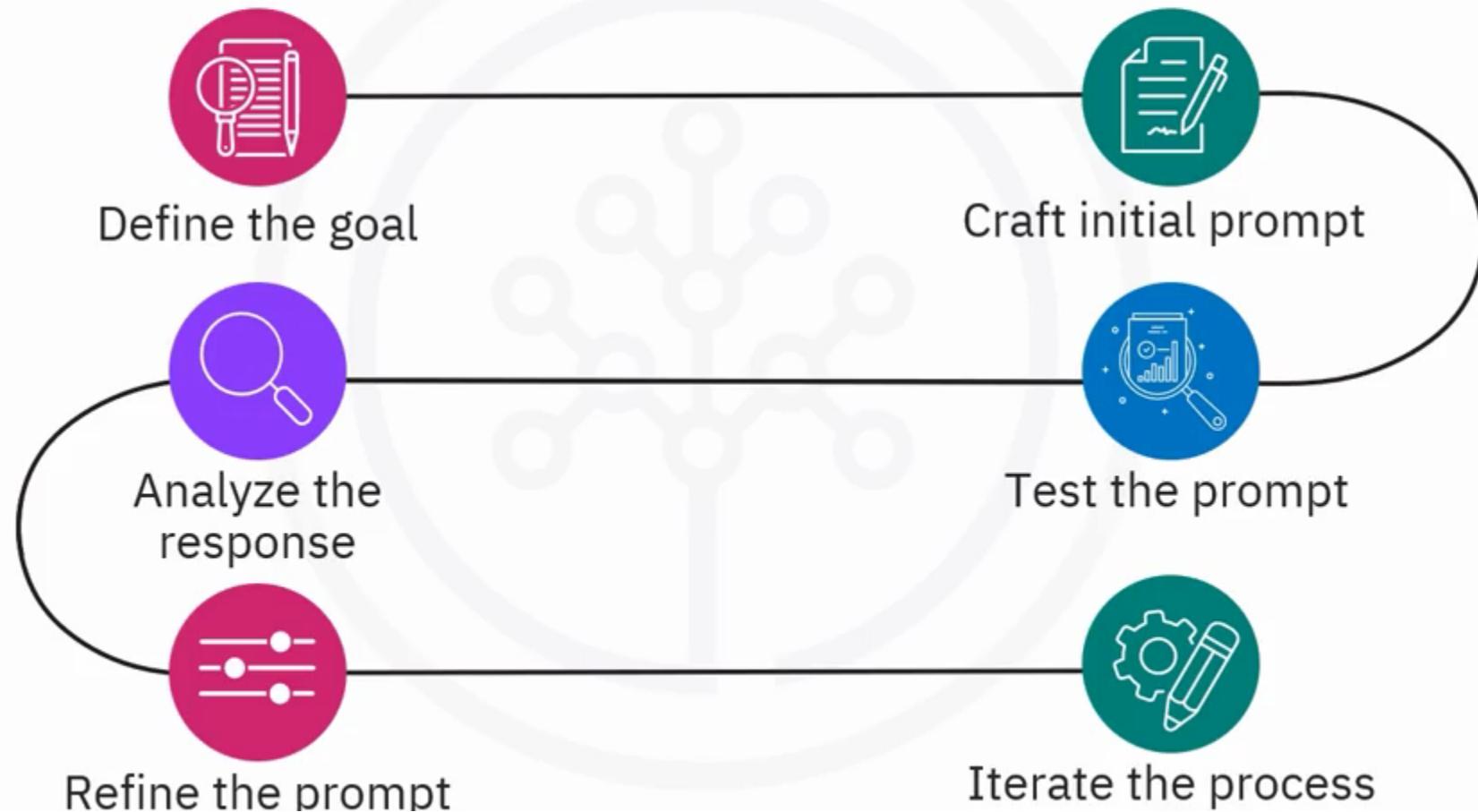
Creativity

Technical acumen



# Process involved in prompt engineering

## Well-structured iterative process



# Process involved in prompt engineering



## Define the goal:

Example: Form a brief overview of the benefits and risks associated with artificial intelligence in automobiles.



## Craft initial prompt

Example: Write an article that presents a well-rounded analysis of the benefits and drawbacks associated with the incorporation of artificial intelligence in the field of automobile industry.

# Process involved in prompt engineering



## Test the prompt

Example: Write an article that presents a well-rounded analysis of the benefits and drawbacks associated with the incorporation of artificial intelligence in the field of automobile industry.

Benefits

Drawbacks

Ethical concerns

Positive and negative implications

# Process involved in prompt engineering

---



## Analyze the response

Fails to cover: Comprehensive range of benefits and risks associated with artificial intelligence in the automobile industry.

# Process involved in prompt engineering



## Refine the prompt

Example: Write an informative article discussing the role of artificial intelligence in revolutionizing the automobile industry.

- Address key aspects such as benefits, drawbacks, ethical considerations, and both positive and negative implications.
- Cover specific domains like autonomous driving and real-time traffic analysis, while also examining potential challenges such as technical complexity and cybersecurity concerns.

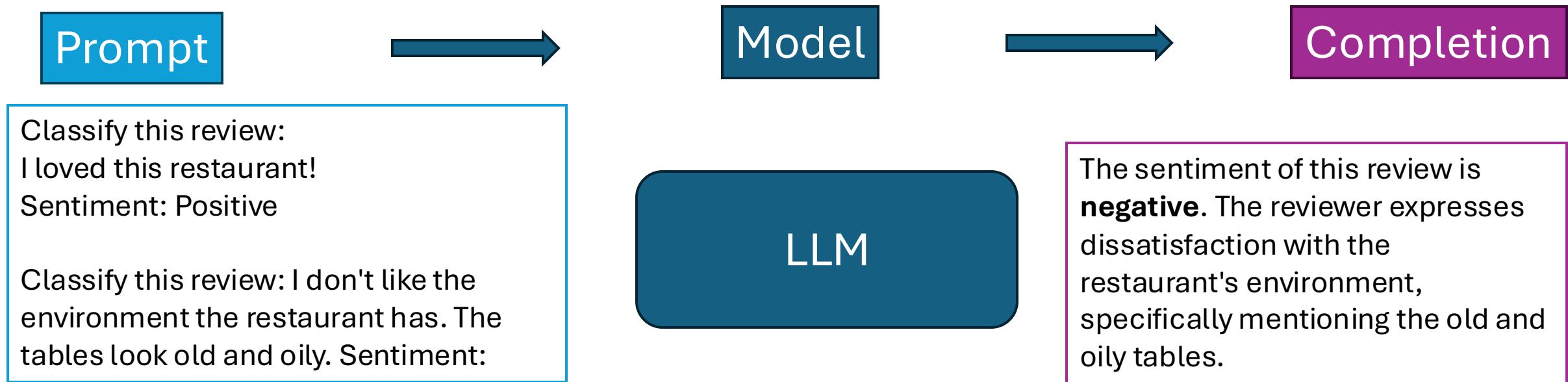
# Process involved in prompt engineering



## Iterate the process:

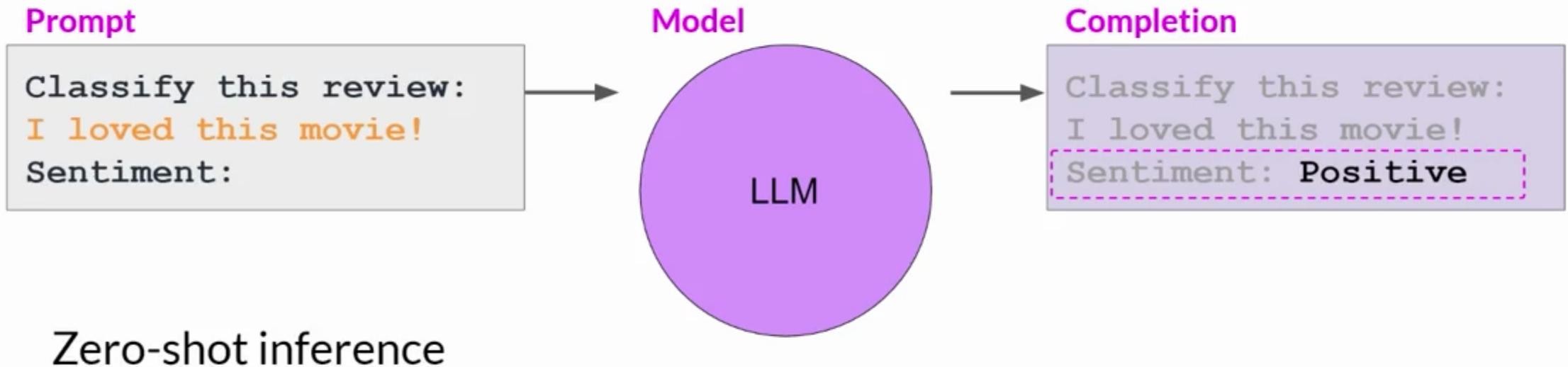
Example: Write an article highlighting how artificial intelligence is reshaping the automobile industry focusing on the positive advancements, particularly in autonomous driving and real-time traffic analysis, while thoroughly exploring concerns related to intricate technical aspects such as decision-making algorithms and potential cybersecurity breaches. Emphasize the implications these concerns may have on vehicle safety. Ensure that the analysis is thorough, backed with examples, and encourages critical thinking.

# In-context learning (ICL) – zero shot inference

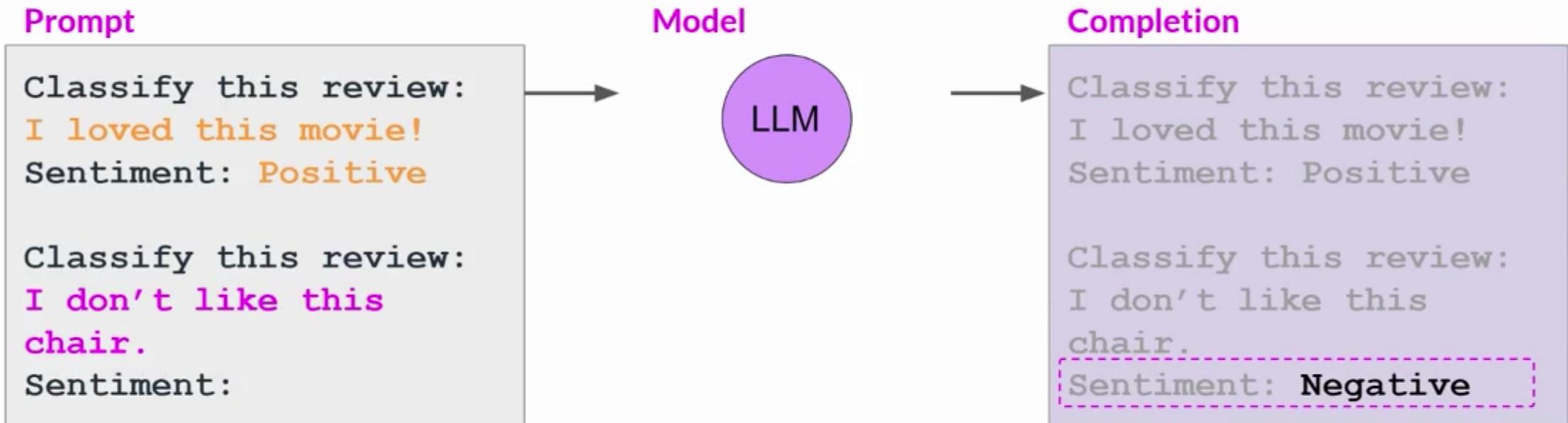


**Zero-shot Inference**

# In-context learning (ICL) - zero shot inference

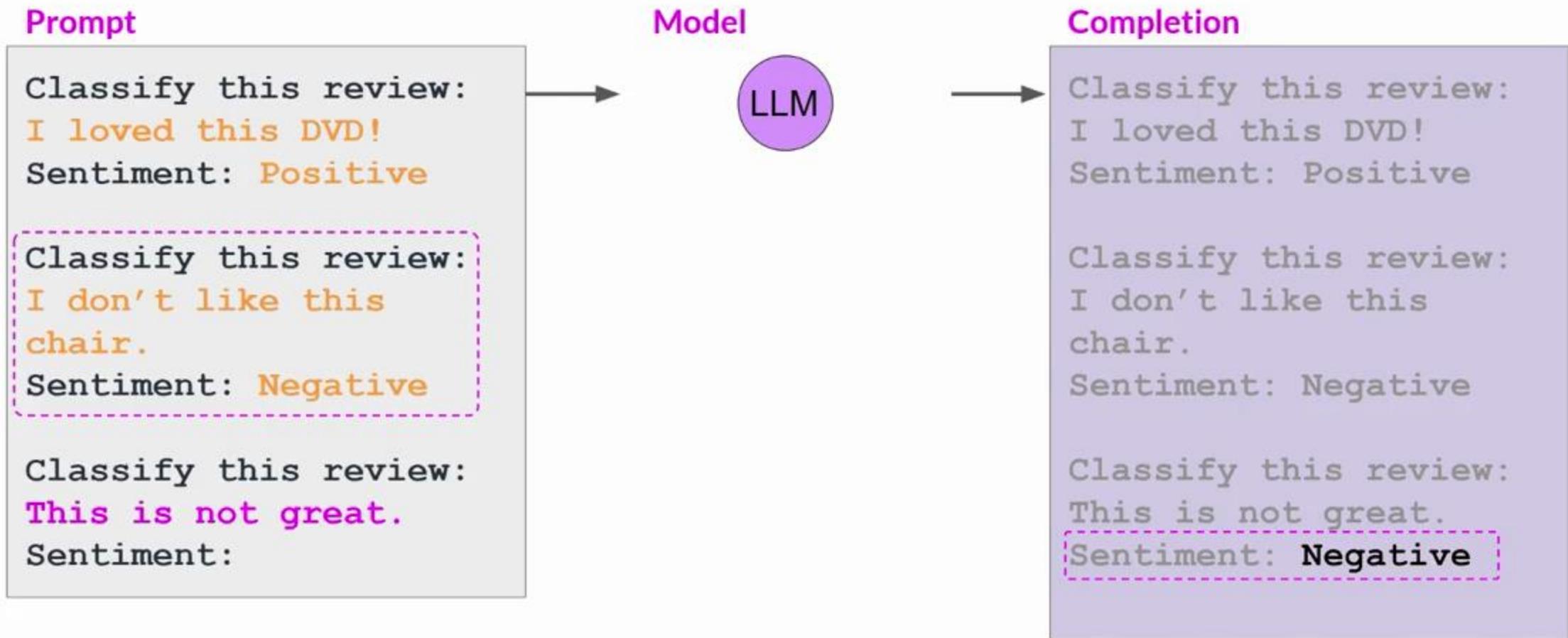


# In-context learning (ICL) - one shot inference



One-shot inference

# In-context learning (ICL) - few shot inference



# Summary of in-context learning (ICL)

**Prompt // Zero Shot**

Classify this review:  
**I loved this movie!**  
Sentiment:

**Context Window**  
(few thousand words)

**Prompt // One Shot**

Classify this review:  
**I loved this movie!**  
Sentiment: **Positive**

Classify this review:  
**I don't like this chair.**  
Sentiment:

**Prompt // Few Shot >5 or 6 examples**

Classify this review:  
**I loved this movie!**  
Sentiment: **Positive**

Classify this review:  
**I don't like this chair.**  
Sentiment: **Negative**

Classify this review:  
**Who would use this product?**  
Sentiment:

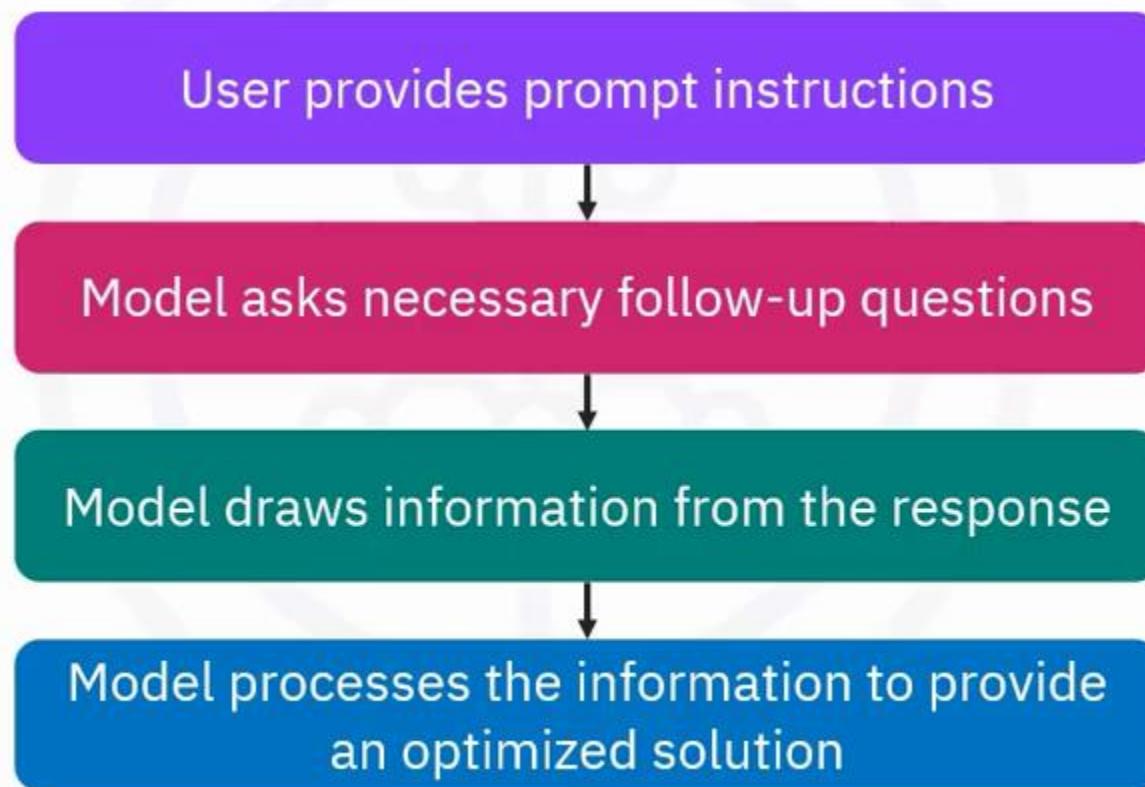
# **Interview Pattern Approach**

---

# Working behind the approach

---

The interview pattern approach involves designing prompts by simulating a conversation or interacting in the interview style.



# Example

---

Prompt instructions:

“You will act as a seasoned travel expert. Your objective is to engage in a comprehensive trip-planning session with me. Begin by asking a series of detailed questions, one at a time, to gather all the essential information required to craft the most tailored and memorable travel itinerary based on my specific preferences, interests, and budget.”

# Example

---

Follow-up questions:

**01**

What types of destinations do you enjoy traveling to the most?

**02**

Could you describe your ideal vacation in terms of activities and experiences?

**03**

How do you typically plan your trips, and what factors are most important to you when choosing a destination?

**04**

Do you find any specific cultural or historical aspects intriguing when planning your travel destination?

**05**

What kind of accommodation options do you prefer when you travel, and why?

**06**

How do you balance budget considerations with the desire for a memorable travel experience?

# Chain-of-Thought Approach

---

# Working behind the approach

Chain-of-Thought  
is a prompt-based  
learning approach.



# Working behind the approach

It involves breaking down a complex task into smaller and easier ones through a sequence of more straightforward prompts.



# Working behind the approach

---

Feed the model with related questions along with their corresponding solutions.

Prompt includes:

- Related question
- Accurate solution to the question
- Another question based on the same reasoning

# Example

---

Mathew has 6 eggs. He buys 2 more trays of eggs. Each tray has 12 eggs. How many eggs does he have now?

**Question:** Mary has 8 radishes. She used 5 radishes to prepare the dinner. The next morning, she bought 10 more radishes. How many radishes does she have now?

**Solution:** Mary had 8 radishes. She cooked dinner using 5 of them. So, she had  $8 - 5 = 3$  radishes left with her. The next morning, she bought 10 more. So, she has  $3 + 10 = 13$  radishes now.

# Example

---

## Model Input

**Q.** Mary has 8 radishes. She used 5 radishes to prepare the dinner. The next morning, she bought 10 more radishes. How many radishes does she have now?

**A.** Mary had 8 radishes. She cooked dinner using 5 of them. So, she had  $8 - 5 = 3$  radishes left with her. The next morning, she bought 10 more. So, she has  $3 + 10 = 13$  radishes now.

**Q.** Mathew has 6 eggs. He buys 2 more trays of eggs. Each tray has 12 eggs. How many eggs does he have now?

## Standard Prompting

### Model Input

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?

A: 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?

### Model Output

A: The answer is 27. 

## Chain-of-Thought Prompting

### Model Input

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?

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

### Model Output

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  $3 + 6 = 9$ . The answer is 9. 

### (a) Few-shot

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?

A: The answer is 11.

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?

A:

*(Output) The answer is 8. X*

### (c) Zero-shot

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?

A: The answer (arabic numerals) is

*(Output) 8 X*

### (b) Few-shot-CoT

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?

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

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?

A:

*(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. ✓*

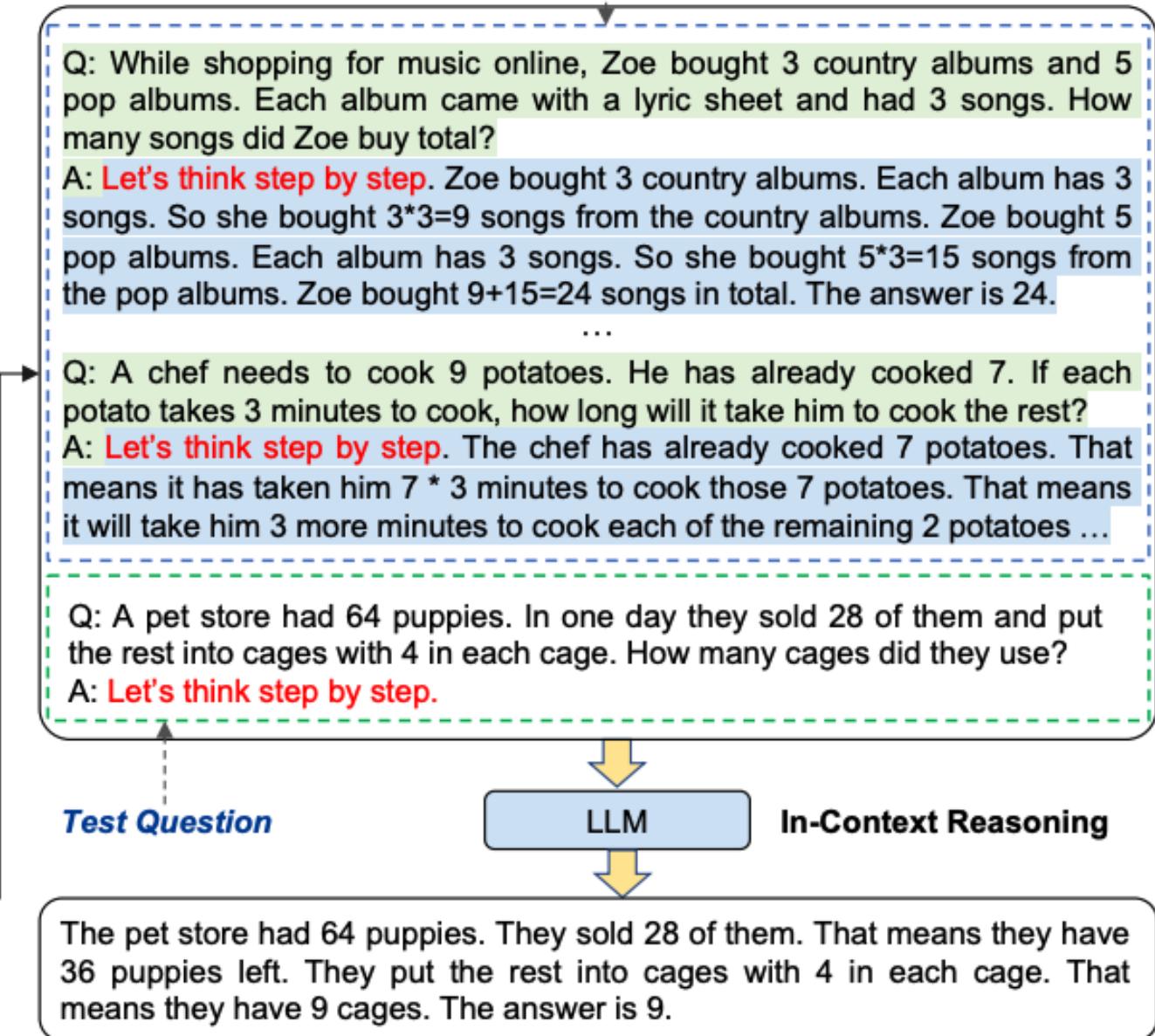
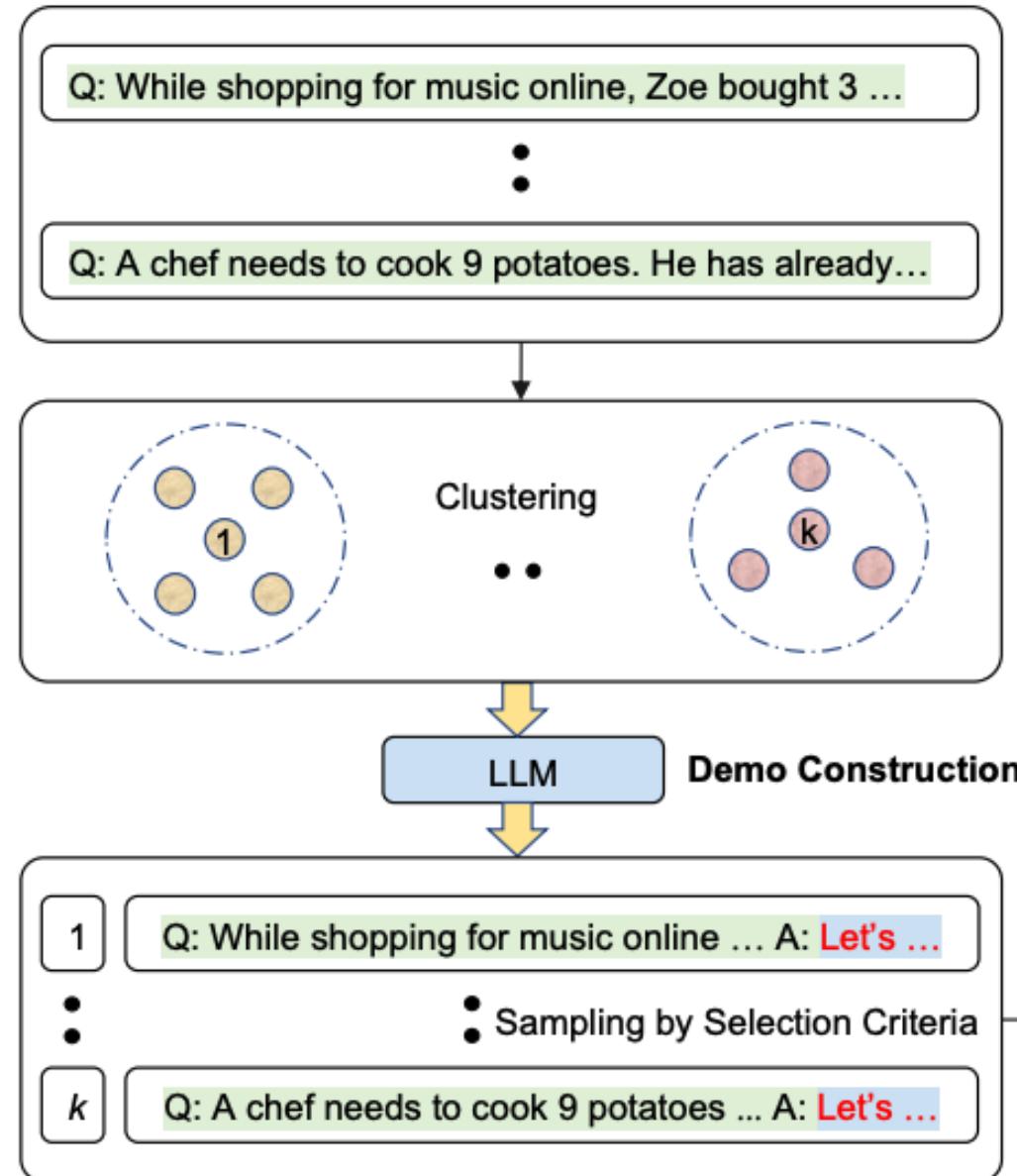
### (d) Zero-shot-CoT (Ours)

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?

A: **Let's think step by step.**

*(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. ✓*

## Auto Demos One by One



# Tree-of-Thought Approach

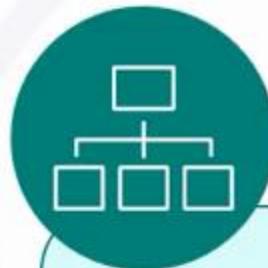
---

# Working behind the approach

Tree-of-Thought is built to expand the capabilities of Chain-of-Thought.



Enables generative AI models to demonstrate advanced reasoning capabilities



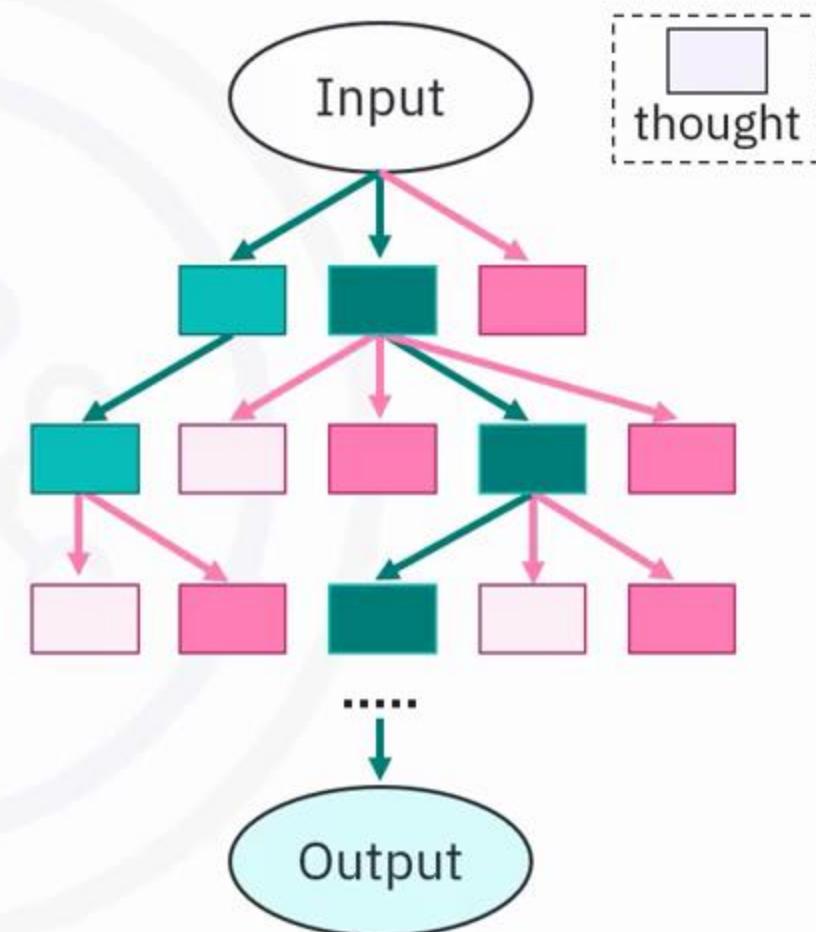
Involves hierarchically structuring a prompt or query, akin to a tree structure



Holds immense potential for unlocking new solutions and tackling complex problems

# Working behind the approach

- Involves generating multiple lines of thought, resembling a decision tree.
- Allows the model to evaluate and pursue multiple paths simultaneously.
- Each thought or idea branches out, creating a tree-like structure of interconnected thoughts.



# Example

---

Design recruitment and retention strategies for attracting skilled remote employees



# Example

---

## Prompt instructions:

Imagine three different experts answering this question. All experts will write down 1 step of their thinking, and then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point, then they leave.

## Prompt:

Act as a human resource specialist, design a recruitment and retention strategy for an e-commerce business, focusing on attracting and retaining skilled remote employees.

# REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao<sup>\*1</sup>, Jeffrey Zhao<sup>2</sup>, Dian Yu<sup>2</sup>, Nan Du<sup>2</sup>, Izhak Shafran<sup>2</sup>, Karthik Narasimhan<sup>1</sup>, Yuan Cao<sup>2</sup>

<sup>1</sup>Department of Computer Science, Princeton University

<sup>2</sup>Google Research, Brain team

<sup>1</sup>{shunyuy, karthikn}@princeton.edu

<sup>2</sup>{jeffreyzhao, dianyu, dunan, izhak, yuancao}@google.com

## ABSTRACT

While large language models (LLMs) have demonstrated impressive performance across tasks in language understanding and interactive decision making, their abilities for reasoning (e.g. chain-of-thought prompting) and acting (e.g. action plan generation) have primarily been studied as separate topics. In this paper, we explore the use of LLMs to generate both reasoning traces and task-specific actions in an interleaved manner, allowing for greater synergy between the two: reasoning traces help the model induce, track, and update action plans as well as handle exceptions, while actions allow it to interface with and gather additional information from external sources such as knowledge bases or environments. We apply our approach, named ReAct, to a diverse set of language and decision making tasks and demonstrate its effectiveness over state-of-the-art baselines in addition to improved human interpretability and trustworthiness. Concretely, on question answering (HotpotQA) and fact verification (Fever), ReAct overcomes prevalent issues of hallucination and error propagation in chain-of-thought reasoning by interacting with a simple Wikipedia API, and generating human-like task-solving trajectories that are more interpretable than baselines without reasoning traces. Furthermore, on two interactive decision making benchmarks (ALFWorld and WebShop), ReAct outperforms imitation and reinforcement learning methods by an absolute success rate of 34% and 10% respectively, while being prompted with only one or two in-context examples.

# How it Works?

## (1d) ReAct (Reason + Act)

**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.

**Act 1:** **Search**[Apple Remote]

**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

**Thought 2:** Apple Remote was originally designed to control the Front Row media center program. I need to search **Front Row** next and find what other device can control it.

**Act 2:** **Search**[Front Row]

**Obs 2:** Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

**Thought 3:** Front Row is not found. I need to search **Front Row (software)**.

**Act 3:** **Search**[Front Row (software)]

**Obs 3:** Front Row is a discontinued media center software ...

**Thought 4:** Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

**Act 4:** **Finish**[keyboard function keys]



# ReAct Example



- Question What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?
- Thought 1 I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, then find the elevation range of the area.
  - Action 1 Search[Colorado orogeny]
  - Observation 1 The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas.
  - Thought 2 It does not mention the eastern sector. So I need to look up eastern sector.
  - Action 2 Lookup[eastern sector]
  - Observation 2 (Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny.
  - Thought 3 The eastern sector of Colorado orogeny extends into the High Plains. So I need to search High Plains and find its elevation range.
- Action 3 Search[High Plains]
- Observation 3 High Plains refers to one of two distinct land regions
- Thought 4 I need to instead search High Plains (United States).
- Action 4 Search[High Plains (United States)]
- Observation 4 The High Plains are a subregion of the Great Plains. From east to west, the High Plains rise in elevation from around 1,800 to 7,000 ft (550 to 2,130 m).[3]
- Thought 5 High Plains rise in elevation from around 1,800 to 7,000 ft, so the answer is 1,800 to 7,000 ft.
- Action 5 Finish[1,800 to 7,000 ft]
- ...

# Overview

## LangChain

- Open-source development framework for LLM applications
- Python and JavaScript (TypeScript) packages
- Focused on composition and modularity

## Key value adds:

- Modular components
- Use cases: Common ways to combine components

# Why use prompt templates?

---

```
prompt = """
Your task is to determine if
the student's solution is
correct or not.
```

```
To solve the problem do the following:
- First, work out your own solution to the problem.
- Then compare your solution to the student's solution
and evaluate if the student's solution is correct or not.
...
Use the following format:
Question:
...
question here
...
Student's solution:
...
student's solution here
...
Actual solution:
...
...
steps to work out the solution and your solution here
...
Is the student's solution the same as actual solution \
just calculated?
...
yes or no
...
Student grade:
...
correct or incorrect
...

Question:
...
{question}
...
Student's solution:
...
{student_solution}
...
Actual solution:
..."
```

Prompts can be long and detailed.

Reuse good prompts when you can!

LangChain also provides prompts for common operations.

# LangChain output parsing works with prompt templates

---

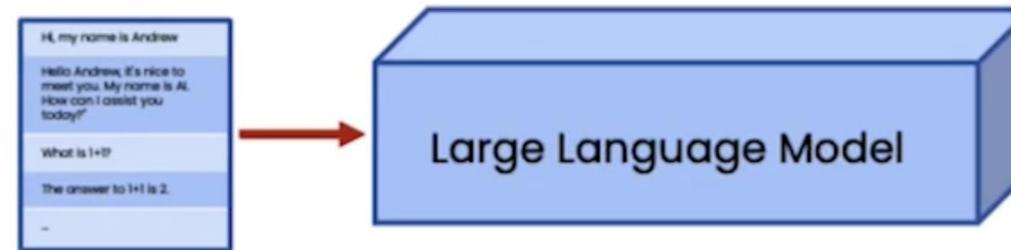
```
EXAMPLES = ["""  
Question: What is the elevation range  
for the area that the eastern sector  
of the Colorado orogeny extends into?  
  
Thought: need to search Colorado orogeny, find  
the area that the eastern sector of the Colorado  
orogeny extends into, then find the elevation range  
of the area.  
  
Action: Search[Colorado orogeny]  
  
Observation: The Colorado orogeny was an  
episode of mountain building (an orogeny) in  
Colorado and surrounding areas.  
  
Thought: It does not mention the eastern sector.  
So I need to look up eastern sector.  
Action: Lookup[eastern sector]  
  
...  
  
Thought: High Plains rise in elevation from  
around 1,800 to 7,000 ft, so the answer is 1,800 to  
7,000 ft.  
  
Action: Finish[1,800 to 7,000 ft]"""  
]
```

LangChain library  
functions parse the  
LLM's output  
assuming that it will  
use certain keywords.

Example here uses  
**Thought**, **Action**,  
**Observation** as  
keywords for Chain-  
of-Thought  
Reasoning. (ReAct)

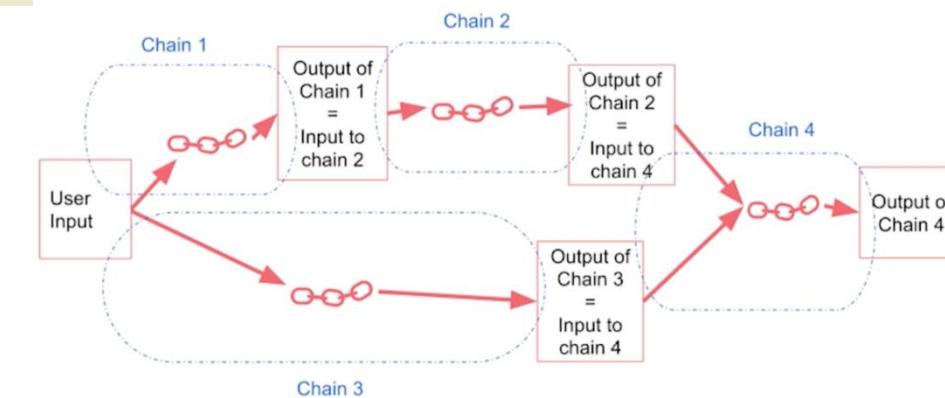
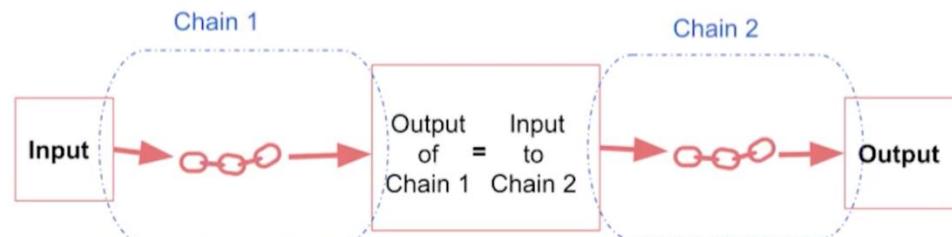
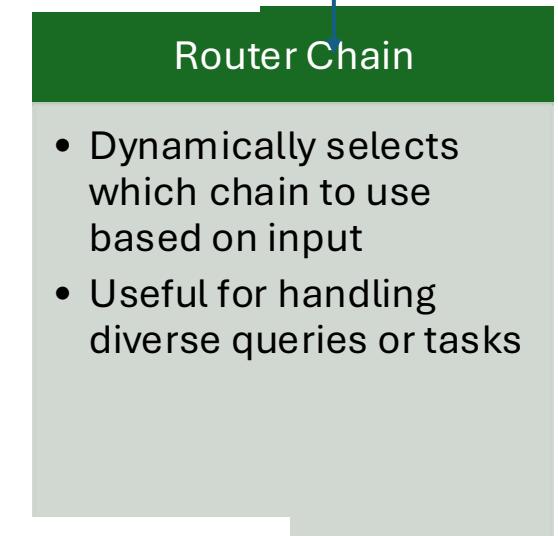
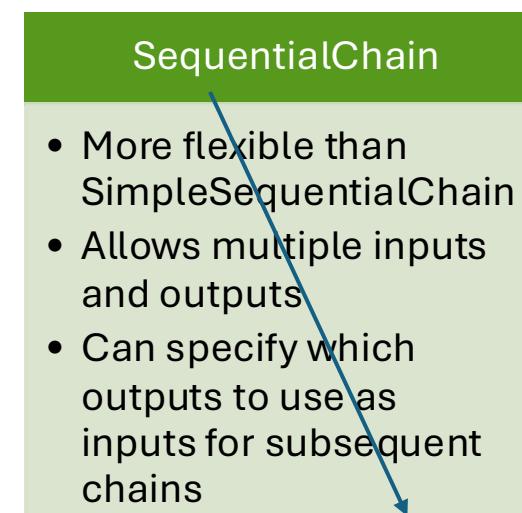
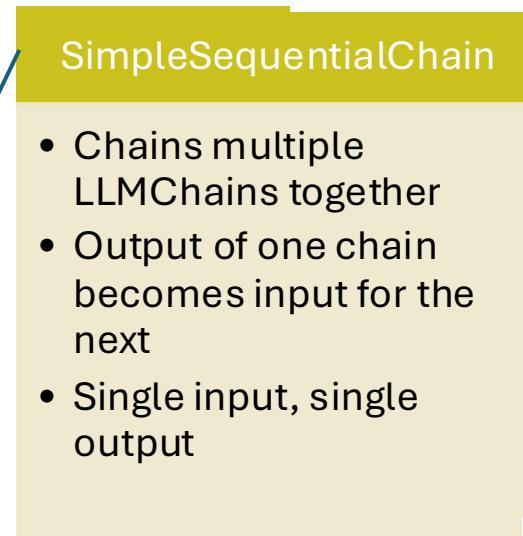
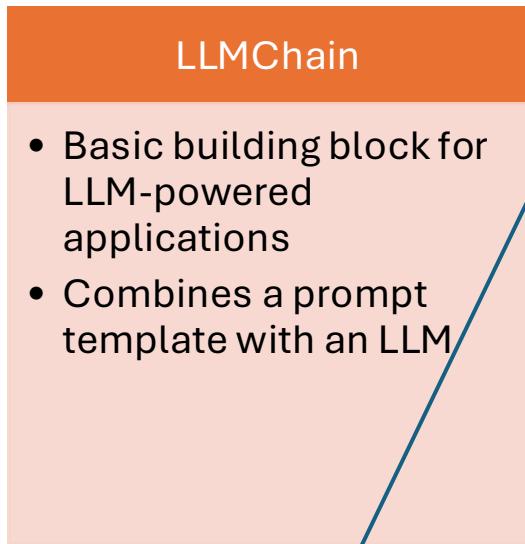
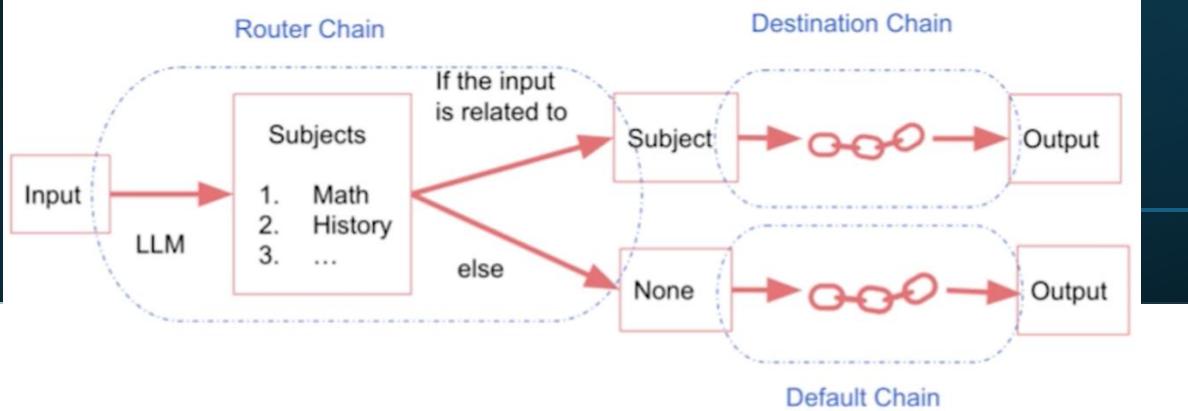
# Memory

- Large Language Models are ‘stateless’
  - Each transaction is independent
- Chatbots appear to have memory by providing the full conversation as ‘context’



- LangChain provides several kinds of ‘memory’ to store and accumulate the conversation.

# Chain



# What is Llamaindex?

---



Llamaindex is a powerful framework for building context-augmented generative AI applications using large language models (LLMs)

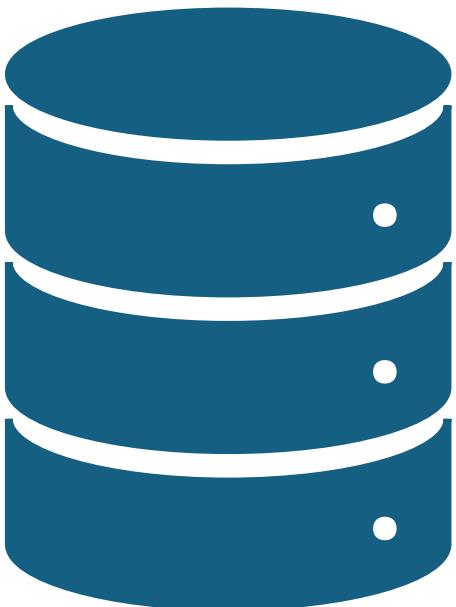
It provides tools to:

- Ingest data from various sources
- Structure and index that data
- Create natural language interfaces to query and interact with the data



Key Benefits:

- Simplifies data integration for LLM apps
- Enables use of private/domain-specific data
- Provides flexible query and retrieval capabilities



# Core Components

1. Data Connectors: Ingest data from various sources (APIs, PDFs, databases, etc.)
2. Indexing: Structure data for efficient retrievalVector stores
3. Graph-based indexes
4. Query Engines: Process natural language queriesRetrieval-Augmented Generation (RAG)
5. Multi-step reasoning
6. Response Synthesizers: Generate coherent answers
7. Chat Engines: Enable conversational interfaces

# Use Cases and Getting Started



- Common Use Cases:
  - Question-answering over documents
  - Chatbots with domain knowledge
  - Data analysis and summarization
  - Augmented content generation