Chain of Thought (CoT) Prompting Tutorial

Overview

This tutorial introduces Chain of Thought (CoT) prompting, a powerful technique in prompt engineering that encourages AI models to break down complex problems into step-by-step reasoning processes. We'll explore how to implement CoT prompting using OpenAI's GPT models and the LangChain library.

Motivation

As AI language models become more advanced, there's an increasing need to guide them towards producing more transparent, logical, and verifiable outputs. CoT prompting addresses this need by encouraging models to show their work, much like how humans approach complex problem-solving tasks. This technique not only improves the accuracy of AI responses but also makes them more interpretable and trustworthy.

Key Components

1. **Basic CoT Prompting**: Introduction to the concept and simple implementation.
2. **Advanced CoT Techniques**: Exploring more sophisticated CoT approaches.
3. **Comparative Analysis**: Examining the differences between standard and CoT prompting.
4. **Problem-Solving Applications**: Applying CoT to various complex tasks.

Method Details

The tutorial will guide learners through the following methods:

1. **Setting up the environment**: We'll start by importing necessary libraries and setting up the OpenAI API.

2. **Basic CoT Implementation**: We'll create simple CoT prompts and compare their outputs to standard prompts.

3. **Advanced CoT Techniques**: We'll explore more complex CoT strategies, including multi-step reasoning and self-consistency checks.

4. **Practical Applications**: We'll apply CoT prompting to various problem-solving scenarios, such as mathematical word problems and logical reasoning tasks.

Conclusion

By the end of this tutorial, learners will have a solid understanding of Chain of Thought prompting and its applications. They will be equipped with practical skills to implement CoT techniques in various scenarios, improving the quality and interpretability of AI-generated responses. This knowledge will be valuable for anyone working with large language models, from developers and researchers to business analysts and decision-makers relying on AI-powered insights.

Setup

Let's start by importing the necessary libraries and setting up our environment.

In [17]:

```
import os
from dotenv import load_dotenv
from langchain_openai import ChatOpenAI
from langchain.prompts import PromptTemplate

# Load environment variables
load_dotenv()

# Set up OpenAI API key
os.environ["OPENAI_API_KEY"] = os.getenv("OPENAI_API_KEY")

# Initialize the language model
llm = ChatOpenAI(model_name="gpt-3.5-turbo")
```

Basic Chain of Thought Prompting

Let's start with a simple example to demonstrate the difference between a standard prompt and a Chain of Thought prompt.

In [18]:

```
# Standard prompt
standard_prompt = PromptTemplate(
    input_variables=["question"],
    template="Answer the following question conciesly: {question}."
)

# Chain of Thought prompt
cot_prompt = PromptTemplate(
    input_variables=["question"],
    template="Answer the following question step by step conciesly: {que
)

# Create chains
standard_chain = standard_prompt | llm
cot_chain = cot_prompt | llm

# Example question
question = "If a train travels 120 km in 2 hours, what is its average sp
```

```
# Get responses
standard_response = standard_chain.invoke(question).content
cot_response = cot_chain.invoke(question).content

print("Standard Response:")
print(standard_response)
print("\nChain of Thought Response:")
print(cot_response)
```

```
Standard Response:
The average speed of the train is 60 km/h.

Chain of Thought Response:
Step 1: Calculate the average speed by dividing the total distance travele
d by the total time taken.

Step 2: Average speed = Total distance / Total time

Step 3: Average speed = 120 km / 2 hours

Step 4: Average speed = 60 km/h

Therefore, the average speed of the train is 60 km/h.
```

Advanced Chain of Thought Techniques

Now, let's explore a more advanced CoT technique that encourages multi-step reasoning.

In [19]:

```
advanced_cot_prompt = PromptTemplate(
    input_variables=["question"],
    template="""Solve the following problem step by step. For each step:
1. State what you're going to calculate
2. Write the formula you'll use (if applicable)
3. Perform the calculation
4. Explain the result

Question: {question}

Solution:"""
)

advanced_cot_chain = advanced_cot_prompt | llm

complex_question = "A car travels 150 km at 60 km/h, then another 100 km

advanced_cot_response = advanced_cot_chain.invoke(complex_question).cont
print(advanced_cot_response)
```

```
1. Calculate the total distance traveled and the total time taken for the
entire journey.
2. Total distance = 150 km + 100 km = 250 km.
   Total time = (150 km / 60 km/h) + (100 km / 50 km/h).
3. Total time = (2.5 hours) + (2 hours) = 4.5 hours.
4. The total distance traveled is 250 km, and the total time taken is 4.5
hours. To find the average speed, we divide the total distance by the tota
l time:
   Average speed = Total distance / Total time
                 = 250 km / 4.5 hours
                 ≈ 55.56 km/h.
5. Therefore, the average speed for the entire journey is approximately 5
5.56 km/h.
```

Comparative Analysis

Let's compare the effectiveness of standard prompting vs. CoT prompting on a more challenging problem.

In [23]:

```python
challenging_question = """
A cylindrical water tank with a radius of 1.5 meters and a height of 4 m
If water is being added at a rate of 10 liters per minute, how long will
Give your answer in hours and minutes, rounded to the nearest minute.
(Use 3.14159 for π and 1000 liters = 1 cubic meter)"""

standard_response = standard_chain.invoke(challenging_question).content
cot_response = advanced_cot_chain.invoke(challenging_question).content

print("Standard Response:")
print(standard_response)
print("\nChain of Thought Response:")
print(cot_response)
```

```
Standard Response:
It will take approximately 3 hours and 56 minutes for the tank to overflo
w.

Chain of Thought Response:
Step 1: Calculate the volume of the water in the tank when it is 2/3 full.
1. Calculate the volume of the cylinder
   Formula: V = πr^2h
   V = 3.14159 * (1.5)^2 * 4
   V ≈ 28.27433 cubic meters

2. Calculate the volume of water in the tank when it is 2/3 full
   Volume = 2/3 * 28.27433
   Volume ≈ 18.84955 cubic meters

Step 2: Calculate how long it will take for the tank to overflow.
1. Calculate the remaining volume until the tank overflows
   Remaining Volume = 28.27433 - 18.84955
   Remaining Volume ≈ 9.42478 cubic meters

2. Convert the remaining volume to liters
   Remaining Volume in liters = 9424.78 * 1000
   Remaining Volume in liters = 9424.78 liters

3. Calculate the time it will take for the tank to overflow
   Time = Remaining Volume / Rate of water addition
   Time = 9424.78 / 10
   Time ≈ 942.478 minutes

Step 3: Convert the time to hours and minutes
1. Convert the time to hours
   Hours = 942.478 / 60
   Hours ≈ 15.70797 hours

2. Calculate the remaining minutes
   Remaining Minutes = 0.70797 * 60
   Remaining Minutes ≈ 42.4782 minutes

Step 4: Final answer
It will take approximately 15 hours and 42 minutes for the tank to overflo
w when water is being added at a rate of 10 liters per minute.
```

Problem-Solving Applications

Now, let's apply CoT prompting to a more complex logical reasoning task.

In [29]:

```
llm = ChatOpenAI(model_name="gpt-4o")

logical_reasoning_prompt = PromptTemplate(
    input_variables=["scenario"],
    template="""Analyze the following logical puzzle thoroughly. Follow

List the Facts:

Summarize all the given information and statements clearly.
Identify all the characters or elements involved.
Identify Possible Roles or Conditions:
```

```
Determine all possible roles, behaviors, or states applicable to the cha
Note the Constraints:

Outline any rules, constraints, or relationships specified in the puzzle
Generate Possible Scenarios:

Systematically consider all possible combinations of roles or conditions
Ensure that all permutations are accounted for.
Test Each Scenario:

For each possible scenario:
Assume the roles or conditions you've assigned.
Analyze each statement based on these assumptions.
Check for consistency or contradictions within the scenario.
Eliminate Inconsistent Scenarios:

Discard any scenarios that lead to contradictions or violate the constra
Keep track of the reasoning for eliminating each scenario.
Conclude the Solution:

Identify the scenario(s) that remain consistent after testing.
Summarize the findings.
Provide a Clear Answer:

State definitively the role or condition of each character or element.
Explain why this is the only possible solution based on your analysis.
Scenario:

{scenario}

Analysis:""")

logical_reasoning_chain = logical_reasoning_prompt | llm

logical_puzzle = """In a room, there are three people: Amy, Bob, and Cha
One of them always tells the truth, one always lies, and one alternates
Amy says, 'Bob is a liar.'
Bob says, 'Charlie alternates between truth and lies.'
Charlie says, 'Amy and I are both liars.'
Determine the nature (truth-teller, liar, or alternator) of each person.'

logical_reasoning_response = logical_reasoning_chain.invoke(logical_puzz
print(logical_reasoning_response)
```

```
Let's analyze the logical puzzle step by step.

### List the Facts:

1. **Characters Involved:**
   - Amy
   - Bob
   - Charlie

2. **Statements:**
   - Amy says, "Bob is a liar."
   - Bob says, "Charlie alternates between truth and lies."
   - Charlie says, "Amy and I are both liars."

3. **Roles:**
   - One person is a truth-teller (always tells the truth).
```

   - One person is a liar (always lies).
   - One person alternates between truth and lies.

Identify Possible Roles or Conditions:

- Each character can be either:
  - A truth-teller
  - A liar
  - An alternator

Note the Constraints:

1. There is exactly one truth-teller, one liar, and one alternator.
2. The statements made by each character must align with their assigned ro
les.

Generate Possible Scenarios:

Let's analyze each possible assignment of roles systematically:

Scenario 1: Amy is the Truth-teller

- **Amy (Truth-teller):** "Bob is a liar."
- **Bob (Liar):** This would imply Bob is lying about Charlie alternating.
- **Charlie (Alternator):** "Amy and I are both liars."

  - If Charlie is alternating, his statement must be a lie since he would
alternate from a previous truth. However, for him to be a liar in this sta
tement, it must be false, which means Amy isn't a liar (consistent with he
r being a truth-teller), but he would be contradicting himself by saying h
e is a liar (which is a lie).

Scenario 2: Amy is the Liar

- **Amy (Liar):** "Bob is a liar." (False, so Bob is not a liar)
- **Bob (Truth-teller):** "Charlie alternates between truth and lies."
- **Charlie (Alternator):** "Amy and I are both liars."

  - Charlie’s statement would have to be false (right now) as Amy is indee
d a liar, but Charlie is not (since he’s an alternator). This matches his
alternating nature.

Scenario 3: Amy is the Alternator

- **Amy (Alternator):** "Bob is a liar."
- **Bob (Truth-teller):** "Charlie alternates between truth and lies."
- **Charlie (Liar):** "Amy and I are both liars."

  - Bob’s statement is true, meaning Charlie is indeed alternating, which
contradicts the assumption of Charlie being a liar.

Test Each Scenario:

After testing each scenario, only Scenario 2 holds consistently:

- **Amy (Liar):** Her statement "Bob is a liar" is false, which is consist
ent with Bob being the truth-teller.
- **Bob (Truth-teller):** His statement "Charlie alternates between truth
and lies" is true.
- **Charlie (Alternator):** His alternating nature allows him to say "Amy
and I are both liars," which aligns with him alternating and being false a

t that moment.

Eliminate Inconsistent Scenarios:

- Scenario 1 and Scenario 3 lead to contradictions and are therefore elimi
nated.

Conclude the Solution:

- **Amy is the Liar.**
- **Bob is the Truth-teller.**
- **Charlie is the Alternator.**

Provide a Clear Answer:

Amy is the liar because her statement is false. Bob is the truth-teller be
cause his statement is true. Charlie is the alternator because his stateme
nt is false at this instance, consistent with his alternating nature. This
is the only scenario that fits all the constraints without contradiction.