Skills
Network

Advanced Methods of Prompt Engineering

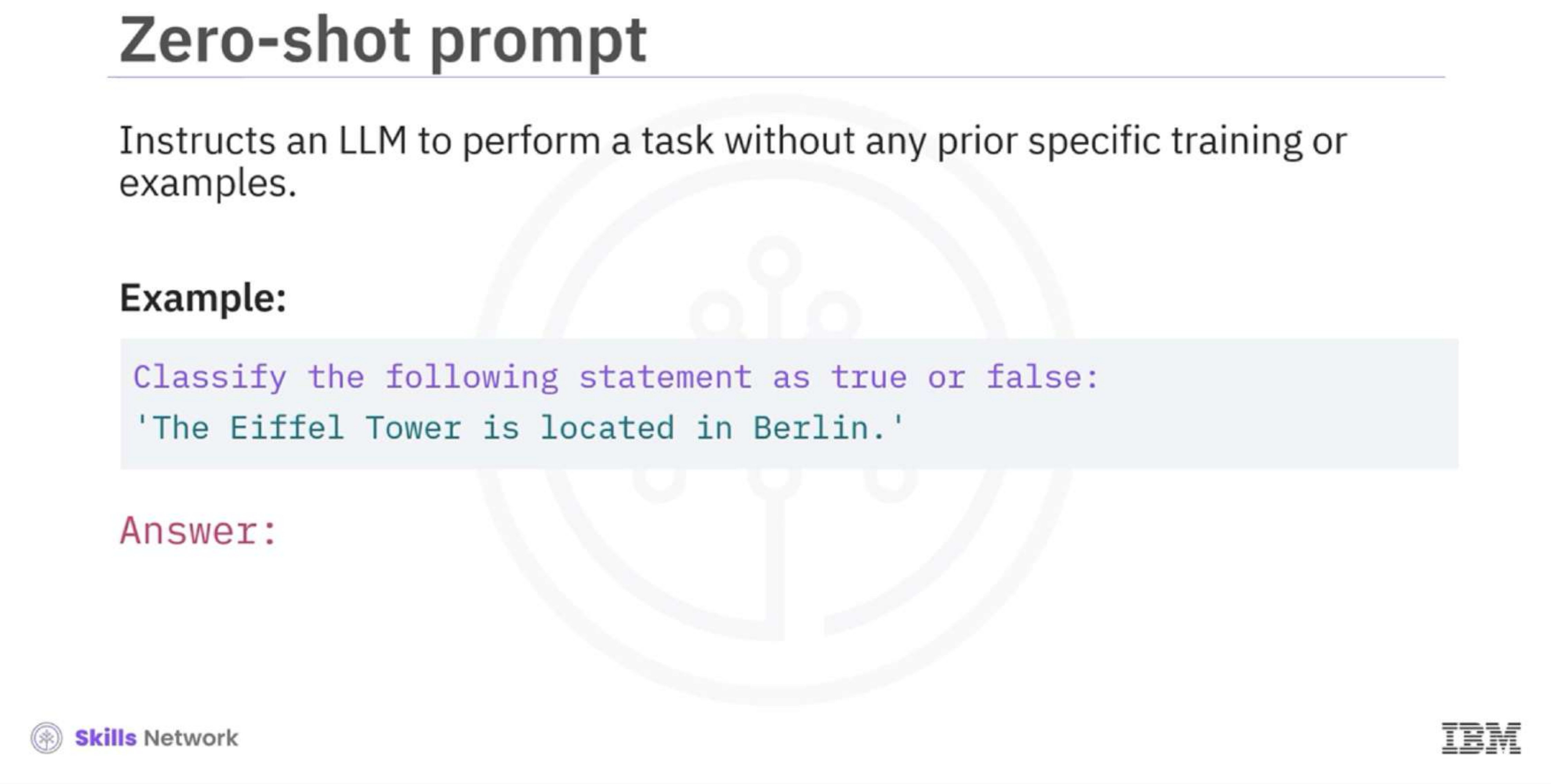
IBM

Advanced Methods of Prompt Engineering

© IBM Corporation. All rights reserved.

What you will learn

- Apply advanced methods of prompt engineering to enhance prompt design
- Discuss the implementation of tools and applications of prompt engineering



Zero-shot prompt

Instructs an LLM to perform a task without any prior specific training or examples.

Example:

```
Classify the following statement as true or false:
'The Eiffel Tower is located in Berlin.'
```

`Answer:`

One-shot prompt

Gives the LLM a single example to help it perform a similar task.

Example:

```
Translating a sentence from English to French:
English: “How is the weather today?”
French: “Comment est le temps aujourd'hui?”
Now, translate the following sentence from English to French:
English: “Where is the nearest supermarket?”
French:
```

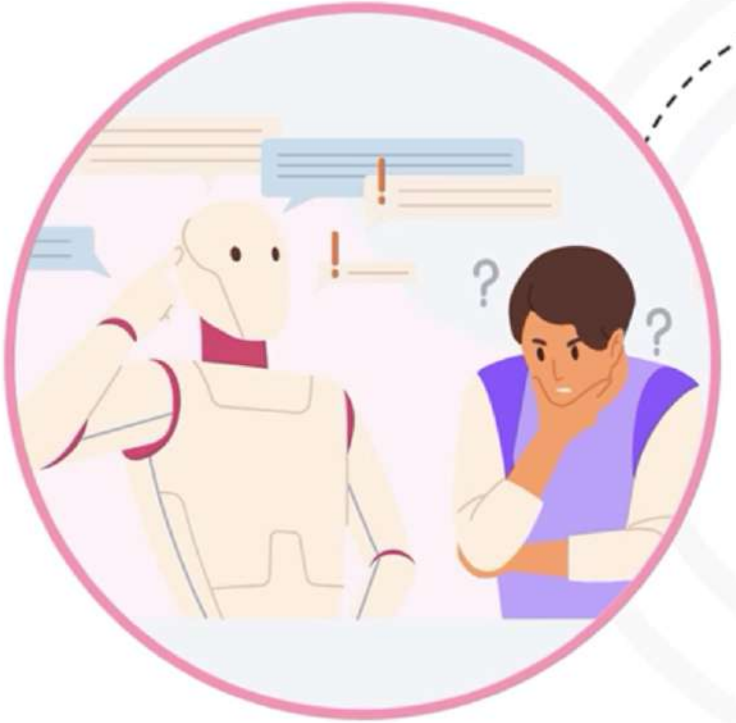
Few-shot prompt

Learns from a small set of examples before tackling a similar task to generalize from a few instances to new data.

Example:

```
Here are few examples of classifying emotions in statements:
Statement: 'I just won my first marathon!'
Emotion: Joy
Statement: 'I can't believe I lost my keys again.'
Emotion: Frustration
Statement: 'My best friend is moving to another country.'
Emotion: Sadness
Now, classify the emotion in the following statement:
Statement: 'That movie was so scary I had to cover my eyes.’
Emotion:
```

Chain-of-thought (CoT) prompting



Guides LLMs through complex reasoning in a step-by-step manner

Effective where the solution requires multiple intermediate steps or reasoning that mimics human thought processes

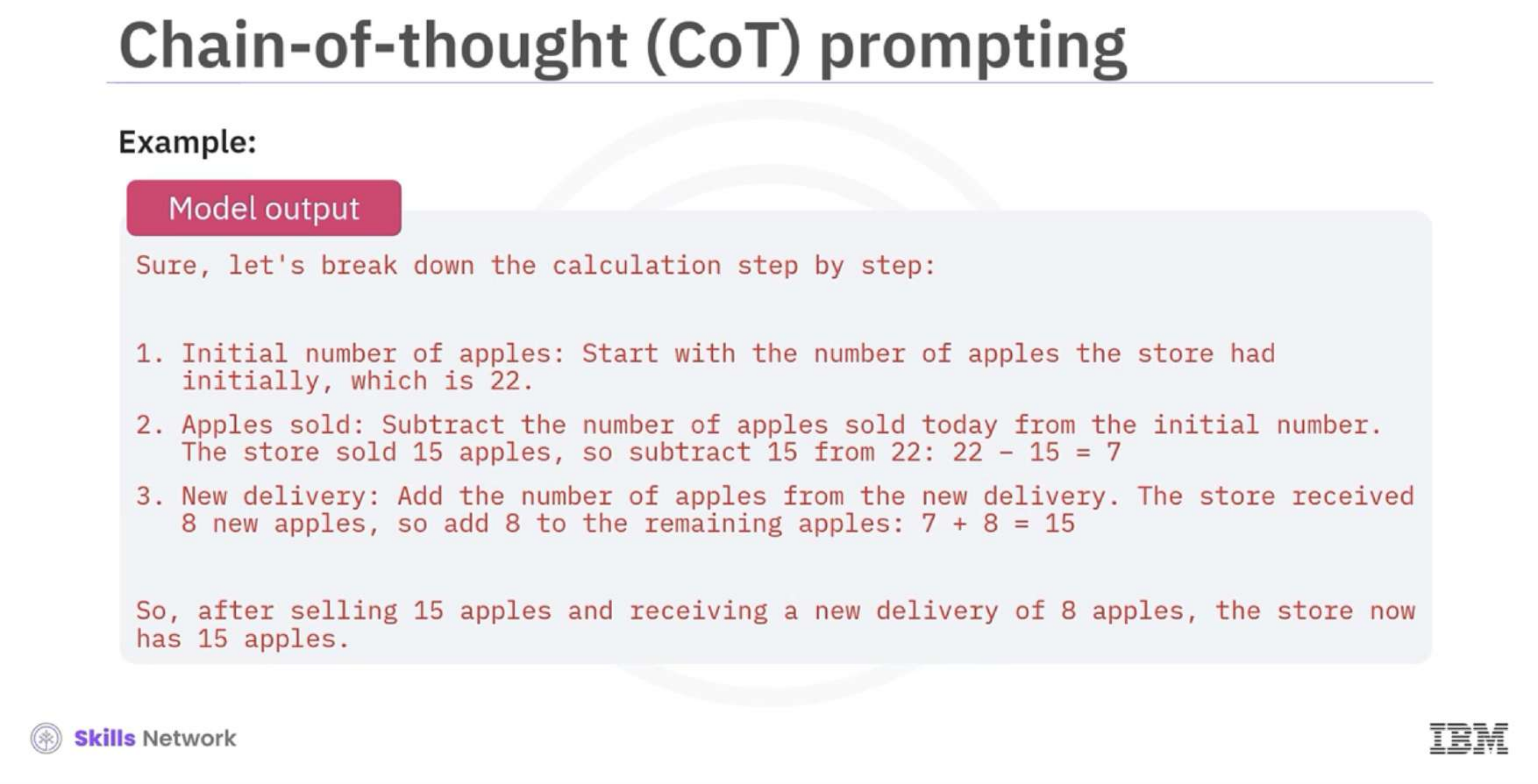
Chain-of-thought (CoT) prompting

Example:

Model input

```
Consider the problem: 'A store had 22 apples. They sold 15 apples today
and got a new delivery of 8 apples. How many apples are there now?'


Break down each step of your calculation
```



Chain-of-thought (CoT) prompting

Example:

Model output

```
Sure, let's break down the calculation step by step:


1. Initial number of apples: Start with the number of apples the store had
   initially, which is 22.

2. Apples sold: Subtract the number of apples sold today from the initial number.
   The store sold 15 apples, so subtract 15 from 22: 22 - 15 = 7

3. New delivery: Add the number of apples from the new delivery. The store received
   8 new apples, so add 8 to the remaining apples: 7 + 8 = 15


So, after selling 15 apples and receiving a new delivery of 8 apples, the store now
has 15 apples.
```

Self-consistency

Model input

When I was 6, my sister was half of my age. Now I am 70, what age is my sister?

Provide three independent calculations and explanations, then determine the most consistent result.

Self-consistency

Model output

When you were 6, your sister was half your age, making her 3 years old. The age difference between you and your sister is 6-3 = 3 years.

Now that you are 70, your sister, being 3 years younger, is 70 - 3 = 67.

You were 6 and your sister was half your age, so she was 3.

Since that time, 64 years have passed (from when you were 6 to when you are now 70).

Your sister's age would then be 3 + 64 = 67.

The ratio when you were 6 (you were twice as old as your sister) means she was 3.

The consistent age gap is 3 years, as previously calculated.

Apply this age gap to your current age: 70 - 3 = 67.

It's consistent that your sister is 67 years old now.

Tools and applications

Tools and applications

Facilitate interactions with LLMs:

Playground

LangChain

HuggingFace

AI classroom

Interactive experimentation: Allows real-time tweaking and testing of prompts

Diverse model access: Provides access to various pre-trained models suitable for different tasks and languages

Collaboration and sharing: Facilitates sharing and collaborative editing of prompts

Performance tracking: Offers tools to track changes, analyze results, and optimize prompts

Prompt template

🦜🔗 LangChain

- Uses prompt templates: Predefined recipes for generating effective prompts for LLMs
- Prompt templates include:
  - Instructions for the language model
  - A few-shot examples to help model understand context and expected responses
  - Specific question directed at the language model

Prompt template

```python
from langchain_core.prompts import PromptTemplate

prompt_template = PromptTemplate.from_template(

    "Tell me a {adjective} joke about {content}."
)
```

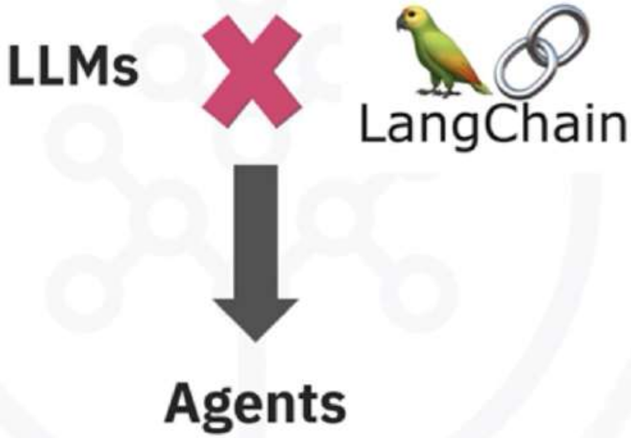
```python
prompt_template.format(adjective="funny", content="chickens")
```

↓

```
"Tell me a funny joke about chickens."
```

Applications

LLM- and LangChain-powered agents

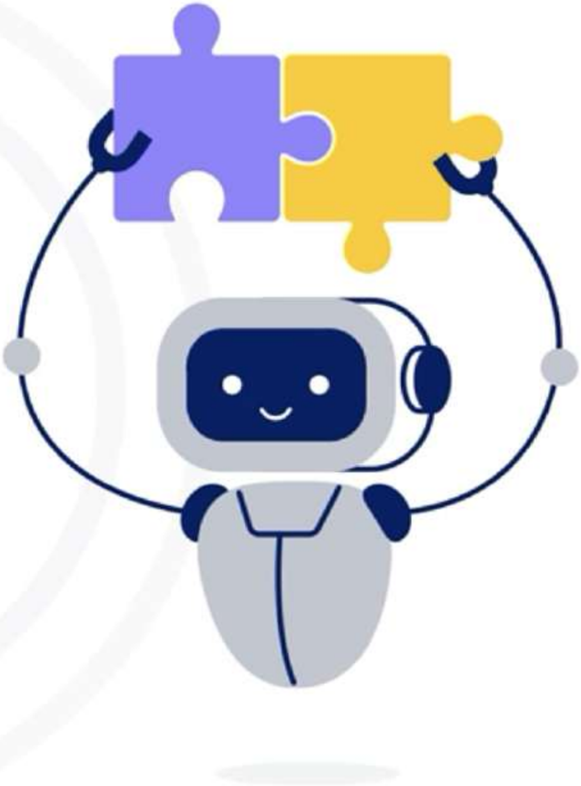


Applications

Agents perform complex tasks across domains using different prompts

Examples:

- Q and A agents with sources
- Content agents for creation and summarization
- Analytic agents for data analysis and business intelligence
- Multilingual agents for seamless, context-aware translation and communication



Recap

- Advanced methods for prompt engineering: Zero-shot prompt, few-shot prompt, chain-of-thought prompting, and self-consistency
- Prompt engineering tools facilitate interactions with LLMs
- LangChain uses ‘prompt templates,’ which are predefined recipes for generating effective prompts for LLMs
- Agent: Key component in prompt applications that can perform complex tasks across various domains using different prompts