

Prompting methods

Date: 03/02/2026

Tool: ChatGPT

Practice Chain of thought, Tabular format, fill in the blank, RGC, zero shot, one shot and few shots prompting

## 1. Chain of Thought (CoT) Prompting

Chain of Thought prompting encourages the model to generate intermediate reasoning steps before arriving at a final answer. This technique is particularly effective for complex arithmetic, symbolic reasoning, and multi-step logic tasks as it mimics human-like "thinking out loud."

**Prompt:** "Calculate the total number of parameters in a simplified CNN layer with 32 filters of size 5x5, an input depth of 3, and a bias term for each filter. Think step-by-step: First, calculate the parameters per filter, then multiply by the number of filters, and finally add the biases."

Calculate the total number of parameters in a simplified CNN layer with 32 filters of size 5x5, an input depth of 3, and a bias term for each filter. Think step-by-step: First, calculate the parameters per filter, then multiply by the number of filters, and finally add the biases

Let's walk through it cleanly and clearly.

Given

- Number of filters = 32
- Filter size =  $5 \times 5$
- Input depth (channels) = 3
- One bias per filter

### 1 Parameters per filter

Each filter spans all input channels, so:

$$5 \times 5 \times 3 = 75 \text{ weights}$$

Add 1 bias per filter:

## 2. Tabular Format Prompting

This technique instructs the AI to organize and present data or comparisons within a structured grid of rows and columns. It is used to improve the readability of multi-dimensional information, making it easier for users to perform quick data analysis and comparative evaluations.

**Prompt:** "Compare AlexNet, VGG16, and ResNet50 in terms of their release year, number of layers, and primary architectural innovation. Present this information in a clear Markdown table."

Compare AlexNet, VGG16, and ResNet50 in terms of their release year, number of layers, and primary architectural innovation. Present this information in a clear Markdown table

Here's a clean side-by-side comparison of AlexNet, VGG16, and ResNet50, focusing on when they came out, how deep they are, and *what made each of them special*.

Model	Release Year	Number of Layers	Primary Architectural Innovation
AlexNet	2012	8 layers (5 convolutional + 3 fully connected)	Popularized <b>deep CNNs</b> for large-scale image classification; introduced <b>ReLU activations</b> , <b>dropout</b> , and <b>GPU training</b>
VGG16	2014	16 layers (13 convolutional + 3 fully connected)	Used <b>very small (3×3) convolution filters</b> stacked deeply for better feature extraction and simplicity
ResNet50	2015	50 layers	Introduced <b>residual (skip) connections</b> , enabling training of very deep networks by mitigating the <b>vanishing gradient problem</b>

### 3. Fill in the Blank Prompting

Also known as Cloze prompting, this method requires the model to complete a sentence or paragraph by providing the missing information based on surrounding context. It is an excellent tool for testing the model's specific knowledge retrieval and its ability to maintain linguistic consistency.

**Prompt:** "Complete the following technical summary by filling in the blanks: 'In the context of deep learning, (1) \_\_\_\_\_ is a regularization technique used to prevent overfitting by randomly setting a fraction of input units to 0. It was famously used in the (2) + \_\_\_\_\_ architecture during the 2012 ImageNet challenge to improve generalization performance.'"

Complete the following technical summary by filling in the blanks: 'In the context of deep learning, (1) \_\_\_\_\_ is a regularization technique used to prevent overfitting by randomly setting a fraction of input units to 0. It was famously used in the (2) \_\_\_\_\_ architecture during the 2012 ImageNet challenge to improve generalization performance.'



Here is the completed technical summary:

"In the context of deep learning, (1) *Dropout* is a regularization technique used to prevent overfitting by randomly setting a fraction of input units to 0. It was famously used in the (2) *AlexNet* architecture during the 2012 ImageNet challenge to improve generalization performance."

#### Final Answers

1. Dropout
2. AlexNet

If you want, I can also explain *why dropout works* in simple exam-friendly terms.

#### 4. RGC (Role-Goal-Context) Prompting

The RGC framework is a structured approach where the user explicitly defines the AI's Role (persona), the Goal (specific task), and the Context (constraints and background). This provides the model with clear boundaries, ensuring the output is tailored to a specific professional or academic standard.

**Prompt:** " Role: You are a Senior Academic Advisor at RV University. Goal: Write a formal recommendation letter for a student named Satyatma Chincholi (USN: 1RVU23CSE414) for a Research Internship in Genomics and Generative AI. Context: The letter should highlight their proficiency in Python, their understanding of the RGC framework, and their interest in personalized medicine. Keep the tone professional and limit it to two paragraphs."

ole: You are a Senior Academic Advisor at RV University. Goal: Write a formal recommendation letter for a student named Satyatma Chincholi (USN: 1RVU23CSE414) for a Research Internship in Genomics and Generative AI. Context: The letter should highlight their proficiency in Python, their understanding of the RGC framework, and their interest in personalized medicine. Keep the tone professional and limit it to two paragraphs



Here is a polished, **formal recommendation letter**, limited to **two paragraphs** and aligned with your context:

I am pleased to write this letter of recommendation for Mr. Satyatma Chincholi (USN: 1RVU23CSE414) in support of his application for a **Research Internship in Genomics and Generative Artificial Intelligence**. As a Senior Academic Advisor at RV University, I have had the opportunity to observe his academic progress and technical aptitude closely. Mr. Chincholi has demonstrated strong proficiency in **Python programming**, which he applies effectively to data analysis, model development, and research-oriented problem solving. His ability to translate theoretical concepts into practical implementations reflects both technical maturity and intellectual curiosity.

## 5. Zero-Shot Prompting

Zero-shot prompting involves giving the model a task without providing any prior examples or demonstrations of how to perform it. It relies entirely on the model's pre-existing training and internal knowledge to understand the intent and generate a relevant response from scratch.

**Prompt:** "Translate the following technical sentence into high-level medical terminology: 'The AI model looks at the patient's DNA to find the best medicine for their specific type of cancer.'"

Translate the following technical sentence into high-level medical terminology: 'The AI model looks at the patient's DNA to find the best medicine for their specific type of cancer.'

A high-level medical translation of the sentence would be:

"The artificial intelligence system analyzes the patient's genomic profile to identify the most appropriate targeted therapeutic intervention for their specific oncological subtype."

If you want it even more clinical (journal-style) or slightly simplified for patient-facing communication, I can tune the phrasing.

📄 👍 🗨️ ⬆️ ↺️ ...

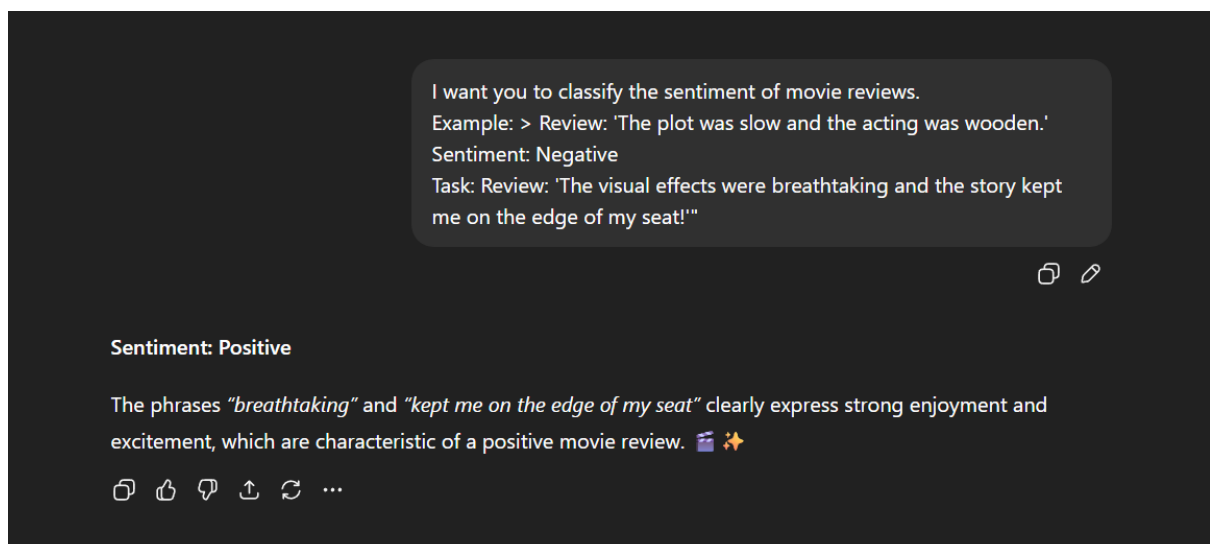
## 6. One-Shot Prompting

In this technique, the user provides the model with exactly **one example** of the desired input-output pair before asking it to complete a new task. This single reference point helps the model grasp the specific format, tone, or logic required for the response.

**Prompt:** "I want you to classify the sentiment of movie reviews.

Example: > Review: 'The plot was slow and the acting was wooden.' Sentiment: Negative

Task: Review: 'The visual effects were breathtaking and the story kept me on the edge of my seat!'"

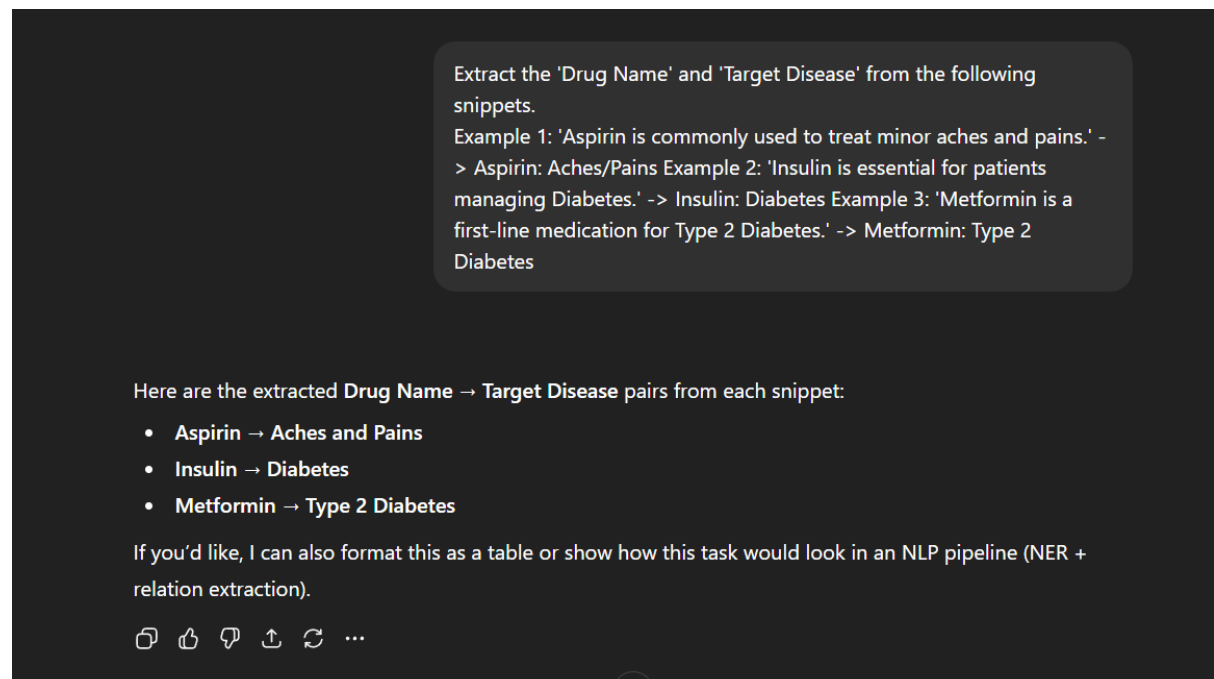


## 7. Few-Shot Prompting

Few-shot prompting provides the model with a small set of examples (typically between 3 and 5) to establish a clear pattern for the task. This is the most robust way to "teach" the model complex formatting rules or specific classification behaviors that might be ambiguous in a zero-shot setting.

**Prompt:** "Extract the 'Drug Name' and 'Target Disease' from the following snippets.

Example 1: 'Aspirin is commonly used to treat minor aches and pains.' -> Aspirin: Aches/Pains  
Example 2: 'Insulin is essential for patients managing Diabetes.' -> Insulin: Diabetes  
Example 3: 'Metformin is a first-line medication for Type 2 Diabetes.' -> Metformin: Type 2 Diabetes"



The screenshot shows a chat window with a dark background. At the top, a light gray message box contains the prompt: "Extract the 'Drug Name' and 'Target Disease' from the following snippets. Example 1: 'Aspirin is commonly used to treat minor aches and pains.' -> Aspirin: Aches/Pains Example 2: 'Insulin is essential for patients managing Diabetes.' -> Insulin: Diabetes Example 3: 'Metformin is a first-line medication for Type 2 Diabetes.' -> Metformin: Type 2 Diabetes". Below this, a response message says: "Here are the extracted Drug Name → Target Disease pairs from each snippet:" followed by a bulleted list: "• Aspirin → Aches and Pains", "• Insulin → Diabetes", and "• Metformin → Type 2 Diabetes". At the bottom of the response, it says: "If you'd like, I can also format this as a table or show how this task would look in an NLP pipeline (NER + relation extraction)." and a row of icons for copy, like, dislike, upvote, downvote, and more options.

Extract the 'Drug Name' and 'Target Disease' from the following snippets.

Example 1: 'Aspirin is commonly used to treat minor aches and pains.' -> Aspirin: Aches/Pains  
Example 2: 'Insulin is essential for patients managing Diabetes.' -> Insulin: Diabetes  
Example 3: 'Metformin is a first-line medication for Type 2 Diabetes.' -> Metformin: Type 2 Diabetes

Here are the extracted Drug Name → Target Disease pairs from each snippet:

- Aspirin → Aches and Pains
- Insulin → Diabetes
- Metformin → Type 2 Diabetes

If you'd like, I can also format this as a table or show how this task would look in an NLP pipeline (NER + relation extraction).

📄 👍 👎 ⬆️ ⬆️ ⋮