Generative AI and Prompt Engineering: Basic Guide

Author: Nguyen Thai Ha (AI-CMT)

1. What is Generative AI?

Generative AI (Gen AI) is a field of artificial intelligence (AI) focused on creating new content. This content can include text, images, audio, video, and even code. Gen AI uses deep learning models to learn from existing data and automatically generate new content.

Copilot

2. How Gen AI Works

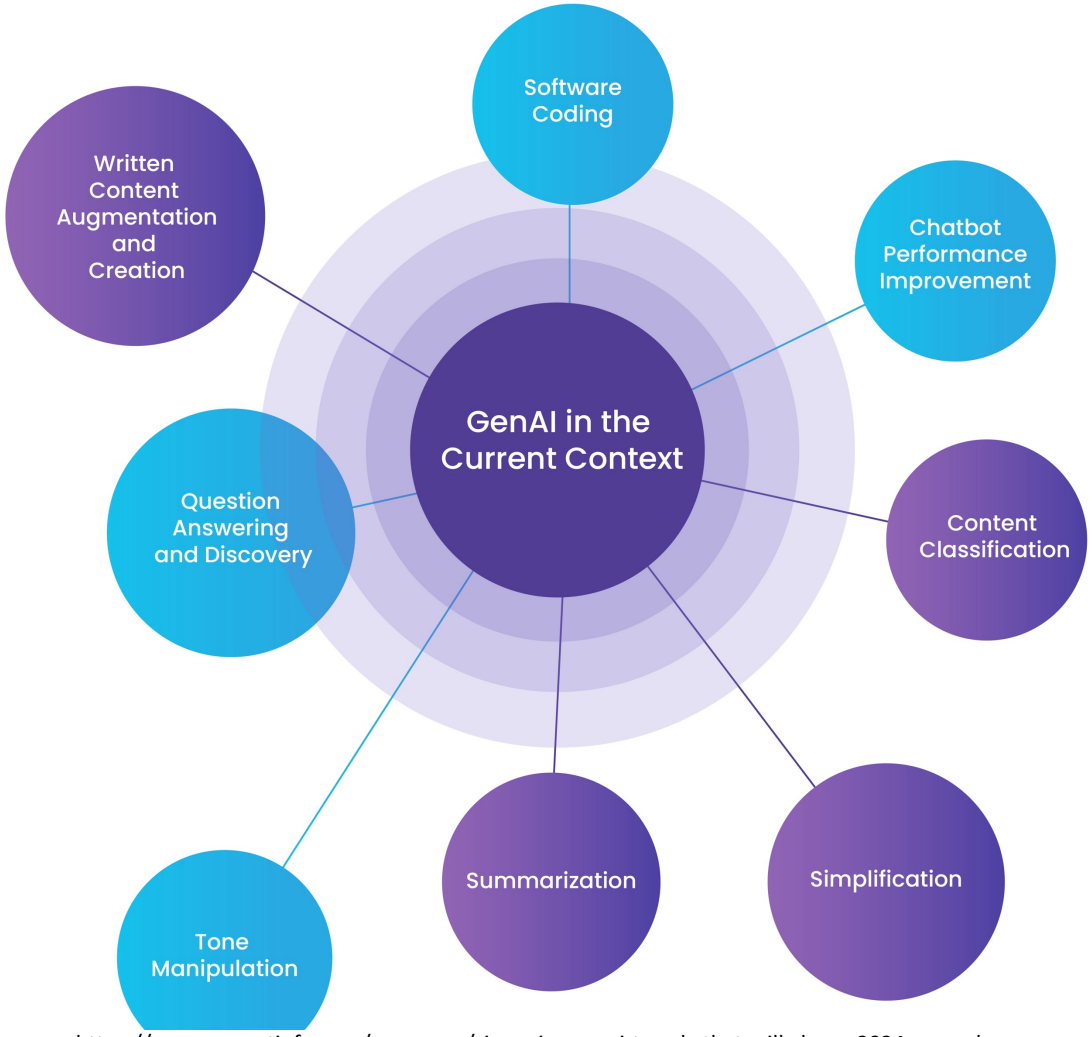
Gen AI operates based on deep learning models, particularly Transformer models. The training process of these models usually involves:

- **Collecting large datasets**: Models are trained on vast amounts of data, such as text, images, and audio, to learn patterns and relationships within the data.
- **Supervised learning**: The training process is often supervised, where the model learns from labeled examples.
- **Prediction and content generation**: For large language models (LLMs), content creation starts with predicting the next word in a text sequence. The model uses probabilities to predict which word is likely to appear next based on the current context. This process repeats until a complete output is achieved.
    - Eg. A cat like... (Sleeping: p=0.9, a dog: p=0.05, a duck: p=0.02 ...)

    => “A cat like sleeping” has the highest probabilities

3. Applications of Gen AI

- **Content creation**: Writing articles, creating images, composing music.
- **Software coding assistance**: Suggesting and auto-writing programming code.
- **Chatbot performance improvement**: Virtual assistants, chatbots.
- **Content classification:** Categorizing and organizing content.
- **Question Answering and Discovery:** Providing answers to questions and discovering relevant information.
- **Summarization:** Condensing large amounts of information into shorter, more digestible summaries.
- **Simplification:** Making complex information easier to understand.
- **Tone Manipulation:** Adjusting the tone of written content to suit different contexts.
- **And many others**



https://www.cogentinfo.com/resources/six-major-genai-trends-that-will-shape-2024s-agenda

4. What is Prompt Engineering and its attributes (1/3)

Prompt Engineering is the process of designing and optimizing prompts so that AI models can understand and respond accurately and effectively. A prompt is an input provided to the AI model to generate the desired response or content.

Key elements of an effective prompt include:

- **Clear and specific:**
  - Definition: The prompt must be clear, unambiguous, and specific about the request.
  - Example: Instead of "Write about a cat," use "Write a 100-word paragraph about a sleek black cat sleeping on a window sill."
- **Provide context:**
  - Definition: Context helps the model better understand the request and generate appropriate responses.
  - Example: "On a cold winter day, describe how you feel sitting by the fireplace."
- **Detailed and descriptive:**
  - Definition: The prompt should provide enough detail and description for the model to generate accurate and rich content.
  - Example: "Write a short story about a lost dog finding its way home, including details about the places it passes and the people it meets."

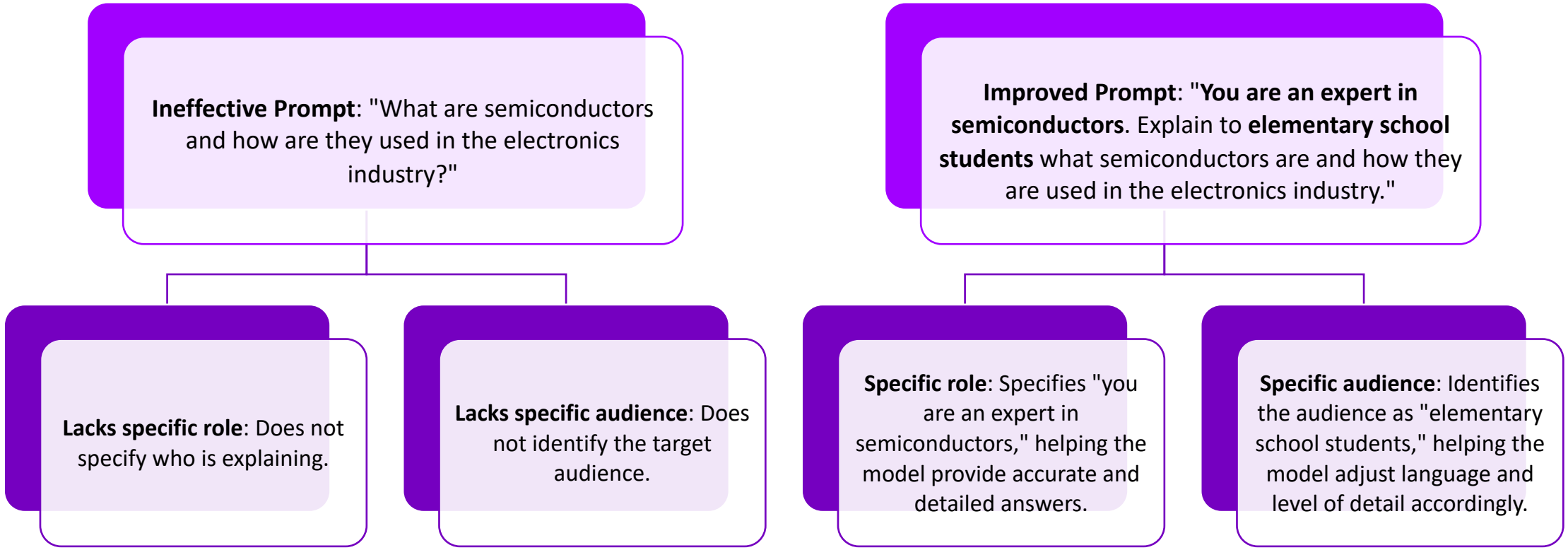
4. What is Prompt Engineering and its attributes (2/3)

- **Specific format and length instructions:**
  - Definition: Clearly state the format (e.g., paragraph, poem, list) and length of the output.
  - Example: "Write a 200-word essay on the benefits of reading books.”
- **Clear audience:**
  - Definition: Specify the audience to help the model adjust the language and tone appropriately.
  - Example: "Explain the concept of solar energy to a 6th-grade student.”
- **Detailed content domain:**
  - Definition: Provide context and details about the topic to ensure the model can generate accurate and relevant responses.
  - Example: "Write a brief report on the impact of blockchain technology in the financial industry."

4. What is Prompt Engineering and its attributes (3/3)

- **Clear perspective:**
  - Definition: Specify the viewpoint or stance you want the model to take in the response.
  - Example: "Write an essay supporting the use of renewable energy."
- **Specified tone:**
  - Definition: Clearly state the desired tone of the response, such as formal, friendly, humorous, etc.
  - Example: "Write a friendly letter to a friend about your recent trip."
- **Specific role:**
  - Definition: Assign a role or context for the model to generate responses from the appropriate perspective.
  - Example: "Write an article from the perspective of a scientist about the importance of wildlife conservation."
- **Switching between multiple roles:**
  - Definition: Use different roles to get responses from multiple perspectives.
  - Example: "Explain the benefits of artificial intelligence from the perspectives of an economist, an educator, and a consumer."

5. Ineffective prompt example and how to improve it



6. Key Parameters in Prompt Engineering

No.	Parameter	Parameter Meaning	Parameter Value	Use Case
1	Temperature	Controls the 'smoothness' of the softmax function by dividing the logits output by the temperature.	0.0 - 1.0	Non-random results (0.0) 'Sharper' results with some randomness (0.7 - 0.8) High creativity (>1.0)
2	Top_p	A parameter in the nucleus sampling technique, selecting the group of words with the highest probabilities exceeding a certain threshold.	0.1 - 0.9	Accurate answers (0.1 - 0.5) Creative answers (0.6 - 0.9)
3	Max Length	Adjusts the maximum number of words the model generates.	Depending on task needs	Limit the number of output words to fit the task.
4	Stop Sequences	Sets a stop sequence that prevents the model from generating further words.	Specific stop sequence	Stop generating words when the predefined stop sequence is encountered.
5	Frequency Penalty	Applies a penalty to the next word based on how many times it appears in the result and prompt.	0.0 - 1.0	Reduce word repetition (0.5 - 1.0) Not concerned about repetition (0.0 - 0.4)
6	Presence Penalty	Applies a penalty to repeated words, with the same penalty for all repeated words.	0.0 - 1.0	Create diverse results (0.6 - 1.0) Maintain accuracy (0.0 - 0.4)

7. Techniques in Prompt Engineering

No	Technique	Description	Comparison	Use Cases
1	Zero-Shot Prompting	Enables the model to perform tasks without specific examples.	No examples needed; relies on model's generalization abilities.	General Q&A, basic text generation.
2	Few-Shot Prompting	Provides a few specific examples to improve task performance.	Examples help guide the model, improving accuracy for specific tasks.	Task-specific scenarios, like sentiment analysis with few labeled examples.
3	Chain-of-Thought Prompting	Encourages the model to generate a natural sequence of intermediate steps.	Improves reasoning by breaking down complex tasks into steps.	Complex problem-solving, mathematical reasoning.
4	Self-Consistency	Generates multiple diverse thought processes and selects the most consistent answer.	Enhances accuracy by considering multiple solutions and selecting the best.	Enhanced accuracy in tasks requiring nuanced understanding.
5	Generated Knowledge Prompting	Generates relevant knowledge statements and uses them to answer questions.	Uses additional generated knowledge to enhance model responses.	Commonsense reasoning, tasks needing external knowledge.
6	Automatic Prompt Engineer (APE)	Automates the creation and evaluation of prompts using large language models.	Reduces manual effort by automating prompt engineering.	Efficiently generating and refining prompts for various applications.

8. Examples of Techniques in Prompt Engineering (1/2)

1. Zero-Shot Prompting
Description:
•Enables the model to perform tasks without specific examples.
Example Prompt:
•"Translate the following English sentence to French: 'The cat is on the roof.’”

2. Few-Shot Prompting
Description:
•Provides a few specific examples to improve task performance.
Example Prompt:
•"Translate the following English sentences to French. Example 1: 'The cat is on the roof.' -> 'Le chat est sur le toit.' Example 2: 'The dog is in the garden.' -> 'Le chien est dans le jardin.' Now translate: 'The bird is in the cage.'"

3. Chain-of-Thought Prompting
Description:
•Encourages the model to generate a natural sequence of intermediate steps.
Example Prompt:
•"To solve the math problem 24 + 18, first add 20 to 24 to get 44. Then add the remaining 2 + 18 to get 20. Finally, add 44 and 20 to get the answer, which is 64."

8. Examples of Techniques in Prompt Engineering (2/2)

4. Self-Consistency
Description:
•Generates multiple diverse thought processes and selects the most consistent answer.
Example Prompt:
•"Explain the process of photosynthesis. Provide two different explanations and choose the one that is most accurate."

5. Generated Knowledge Prompting
Description:
•Generates relevant knowledge statements and uses them to answer questions.
Example Prompt:
•"What are the benefits of exercise? Generate a list of benefits before providing a detailed explanation."

6. Automatic Prompt Engineer (APE)
Description:
•Automates the creation and evaluation of prompts using large language models.
Example Prompt:
•"Create a prompt for summarizing articles on climate change. Evaluate the prompt and refine it for better accuracy."

9. Challenges in Prompt Engineering

1. Ambiguity in Prompts

- Issue: Ambiguous or poorly defined prompts can lead to inaccurate or irrelevant responses.
- Solution: Develop clearer guidelines and best practices for prompt creation.

2. Model Limitations

- Issue: Current models may struggle with understanding complex or nuanced prompts.
- Solution: Continuous improvement and training of models with diverse datasets.

3. Ethical Considerations

- Issue: Ensuring AI-generated content is unbiased and ethical.
- Solution: Implement ethical guidelines and conduct regular audits.

4. Scalability

- Issue: Scaling prompt engineering techniques for large-scale applications.
- Solution: Develop automated tools and frameworks to streamline the process.

10. Future Directions

1. Large Language Models (LLM) to Large Multi-Models

- Focus: Expand capabilities by integrating multiple models (e.g., combining text, image, and speech models).
- Example: Developing systems that can understand and generate content across different modalities, such as a model that can answer questions based on both text and images.

2. Large Language Models (LLM) to Small Language Models (SLM)

- Focus: Optimize and compress large models to create smaller, more efficient models without significant loss in performance.
- Example: Techniques like knowledge distillation and model pruning to create SLMs that can be deployed on devices with limited computational resources.

3. Retrieval Augmented Generation (RAG)

- Focus: Enhance language models by integrating them with external knowledge databases to improve the relevance and accuracy of generated content.
- Example: Using search engines or domain-specific databases to provide additional context and information that the language model can draw upon when generating responses.

4. AI on Edge Devices

- Focus: Develop AI models that can run efficiently on edge devices, enabling real-time processing and decision-making without relying on cloud resources.
- Example: Implementing AI models in smartphones or IoT devices for applications like real-time language translation, personalized recommendations, and predictive maintenance.

Others: AI agent (devin), Large Action Models (LAMs) and etc

Conclusion

Recap of Key Points

- **Generative AI:** Significant applications and transformative potential across industries.
- **Prompt Engineering:** Crucial techniques and best practices for crafting effective **prompts** to improve AI responses.

Challenges

- Ambiguity in prompts, model limitations, ethical considerations, and scalability issues.

Future Directions

- Advancements such as Large Multi-Models, Small Language Models, Retrieval Augmented Generation, and AI on edge devices.

Final Thoughts

- Generative AI and prompt engineering are rapidly evolving fields requiring continuous innovation and ethical focus.
- Stay informed, experiment with techniques, and collaborate with others to advance these technologies.

Appendix

Prompt Principle for Instructions (1/3)

Principle	Prompt Principle for Instructions
1	If you want a short answer, there's no need to be polite with the LLM, no need for phrases like "please", "if you don't mind", "thank you", "I would like to", etc. Get straight to the point.
2	Mention the target audience of the answer in the prompt, e.g., "experts in their field" for specific readers.
3	Break complex tasks into a sequence of simpler prompts in a conversational flow.
4	Use assertive statements like "do", while avoiding negative language like "don't".
5	When you want to recognize or understand a concept, idea, or any information better, use the following prompts: • Explain [insert specific topic] in simple terms. • Explain to me like I'm 11 years old. • Explain to me as if I'm a beginner in [field]. • Write the [essay/text/paragraph] using simple English like you're explaining something to a 5-year-old.
6	Add "I'm going to tip $xxx" (I will tip you $xxx) for better results.
7	Implement prompts based on examples (Use few-shot prompting).
8	When creating your prompt, start with "###Instruction###", followed by "###Example###" or "###Question###" if necessary. Present your context. Use one or multiple line breaks to separate sections like examples, questions, or contexts.
9	Use phrases like "Your task is" and "You MUST".
10	Use phrases like "You will be penalized".

REF: “Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4” Bsharat et al., 2024.

Prompt Principle for Instructions (2/3)

Principle	Prompt Principle for Instructions
11	Use phrases like "Answer a question given in a natural, human-like manner" (answer the question in a natural, human-like way).
12	Use the phrase "think step by step".
13	Add to your prompt: "Ensure that your answer is unbiased and avoids relying on stereotypes." (Ensure that the answer is unbiased and avoids stereotypes).
14	Allow the model to elaborate and provide specific and necessary information by adding phrases like "From now on, I would like you to ask me questions to clarify what I need to know."
15	To ask about a specific topic or any information and you want to test your understanding, you can use the following prompt: "Teach me any [theorem/topic/rule name] and include a test at the end, and let me know if my answers are correct after I respond, without providing the answers beforehand."
16	Assign roles to large language models.
17	Use punctuation marks.
18	Repeat words or phrases multiple times in the prompt.
19	Combine Chain-of-Thought with Few-Shot prompts.
20	Use introductory paragraphs, including suggestions that you want at the beginning of the answer. Use the opening paragraph by finishing the prompt with the beginning part of the desired response.

REF: “Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4” Bsharat et al., 2024.

Prompt Principle for Instructions (3/3)

Principle	Prompt Principle for Instructions
21	To write an essay/text/paragraph for me on [topic] in detail by adding all the necessary information: "Write a detailed [essay/text/paragraph] for me on [topic] including all the necessary details."
22	To revise/edit the text sent by the user without changing the style: "Try to revise every paragraph sent by users. You should only improve the user's grammar and vocabulary and make sure it sounds natural. You should maintain the original writing style, ensuring that a formal paragraph remains formal."
23	When you have a complex coding prompt involving multiple files: "From now on and whenever you generate code that spans more than one file, generate a [programming language] script that can be run to automatically create the specified files or
24	When you want to start or continue a text with specific sentences or phrases: • I'm providing you with the beginning [song lyrics/story/paragraph/essay...]: [Insert lyrics/words/sentence]. Finish it based on the words provided. Keep the flow consistent.
25	Specify exactly what the model must follow to create content, such as keywords, rules, suggestions, or guidelines.
26	To write any text, such as an essay or paragraph, and you want the writing style to be similar to the provided sample, include the following prompts: • Use the same language based on the provided paragraph [/title/text/essay/answer].

REF: "Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4" Bsharat et al., 2024.

Techniques in Prompt Engineering

No.	Concept	Meaning	Activities	Example	Use Case
1	Zero-shot Prompting	Helps the model perform tasks without specific examples	Fine-tune the language model on a range of tasks described in natural language prompts	The model analyzes and evaluates customer service employee attitudes via email without specific examples	Perform new tasks that haven't been specifically trained for
2	Few-shot Prompting	Helps the model learn and perform tasks with a few specific examples	Provide a few examples in the prompt for the model to learn and adapt	The model quickly classifies user comments with a few illustrative examples	Adapt and perform new tasks based on a few specific examples
3	Chain-of-thought Prompting	Improves the model's reasoning ability by creating thought chains	Stimulate the model to generate intermediate thinking steps leading to the final result	The model solves complex arithmetic reasoning tasks by creating a sequence of thought steps	Solve tasks requiring complex reasoning
4	Zero-shot Chain-of-thought Prompting	Create thought chains without specific examples	Add the phrase "Let's think step by step" to the prompt	When asked "What is 5 plus 3?", the model thinks step by step: "5+3, the result is 8."	Tasks requiring logical reasoning without specific examples
5	Automatic Chain-of-thought Prompting	Automatically generates thought chains in large language models	Leverage the model's ability to generate thought chains on its own	When asked "How to cook pho?", the model automatically generates steps: "Prepare ingredients, cook the broth..."	Solve complex reasoning tasks
6	Self-Consistency	Generate multiple thought directions and choose the most common answer	Generate multiple thought chains and select the most common answer	When asked "What is 10 divided by 2?", the model generates multiple answers and selects the most frequent one	Reasoning tasks, ensuring consistency and accuracy
7	Generated Knowledge Prompting	Uses generated knowledge to answer complex questions	Generate knowledge statements related to the question and use them as input	With the question "Who was the first U.S. president?", the model generates statements and selects the correct answer	Improve common sense reasoning, answer questions based on generated knowledge
8	Tree of Thoughts	Explores different thought steps to solve problems	Generate potential thoughts from each state and evaluate the states	When solving "3+5×2", the model considers different thought steps and chooses the correct solution	Tasks requiring multiple reasoning steps and decisions
9	Automatic Prompt Engineer (APE)	Automates the prompt creation process	Use LLM to generate and evaluate prompts, filter out the most effective ones	Generate prompt proposals, evaluate and filter out the best prompts	Automate prompt engineering, create more effective prompts than human-generated ones
10	Active-Prompt	Improves Chain-of-Thought method by selecting the best examples	Label training data, calculate uncertainty, and select the best examples for specific tasks	Calculate uncertainty for questions and select those with the highest uncertainty for manual labeling	Select appropriate illustrative examples for reasoning tasks
11	Directional Stimulus Prompting	Uses directional cues to guide the model	Use specific directives or cues to guide the model in generating more accurate answers	Use specific keywords to help the model understand the request and generate appropriate answers	Create accurate summaries or answers based on directional cues
12	PAL: Program-aided Language Models	Uses programs to support language model in solving complex tasks	Combine language models with programs written in programming languages, delegate problem-solving to interpreters	Use intermediate reasoning steps and Python code to solve problems	Solve natural language and programming problems, enhance processing accuracy
13	ReAct Prompting	Combines reasoning and actions	The model reasons and performs actions based on its reasoning	When answering questions, the model not only reasons but also performs actions to gather additional information	Tasks requiring a combination of reasoning and actions
14	Reflexion	Uses verbal reinforcement learning to improve the model	The model learns from its own feedback and the environment, enhancing its self-learning and self-improvement	The model reflects and adjusts actions based on feedback from the environment	Enhance the model's self-learning ability in interactive environments
15	Multimodal CoT Prompting	Uses multimodal data to create thought chains	Combine information from multiple sources (text and images) to generate rich and accurate thought chains	Combine text and image inputs for the model to reason and generate answers	Improve reasoning and generate answers for large language models using multimodal data
16	Synthetic Prompting	Automatically generates illustrative examples of thought chains	Use large language models to automatically generate illustrative examples	Generate illustrative thought chain examples based on a few manual examples	Solve complex problems without needing manual example creation

References

1. https://www.promptingguide.ai/jp
2. https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api
3. “Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4” Bsharat et al., 2024.

