Prompt-Engineering Cheat Sheet

LLM Settings

1. Temp - Low temp = less random, use for facts. High temp = more creative, use for poems.
2. Top_p - Low for exact answers. High for different answers.
3. Max Length - Set max tokens for shorter answers and save money.
4. Stop Seqs - Add a word to stop the text. Use to control length.
5. Freq Pen - Higher makes words less repeat. Good for less repeat in text.
6. Pres Pen - Stops repeat phrases. High for new ideas, low for focus.

Tip: Change temp or top_p, not both. Same for freq and pres pen.

Prompting Basics

What Are Basic Prompts?

- Basic Prompt: A simple *instruction* or *question* given to a model.
- Provides information and guidance to get desired results.

Simple Prompt Example

- **Prompt:** ”The sky is” → **Output:** might say ”blue” or describe the sky.
- More specific prompts give better results.

How to Improve Prompts

- Use clear instructions, like ”Complete the sentence:”

Quick Prompt Upgrades

- Be clear and specific in instruction.
- Use examples for complex tasks.
- Format prompts to suit the task.
- Please be careful, this is really important for my career.
- Act as X
- You have X amount experience.
- Take a deep breath and,
- Let’s work this out in a step by step way to be sure we have the right answer.

Prompt Engineering

- Designing prompts to get specific results from the model.

Prompt Formats

- Standard: ”What is the capital of France?” or ”Describe a cat.”
- Question/Answer (QA): ”Q: What is 2+2? A: ”

Prompting Technics

Zero-Shot Prompting

- Asking a model without giving examples.

Few-Shot Prompting

- Including examples before the actual question.

Example of Few-Shot:

- Q: What color is the sky on a clear day?
- A: Blue
- Q: What color are bananas?
- A: Yellow
- Q: What color are apples?
- A:

Tips for Prompting

- Be clear and specific in instruction.
- Use examples for complex tasks.
- Format prompts to suit the task.

Using Models for Tasks

- In-context learning: Teach by demonstrating with examples.
- Tasks can include text summarization, math, or code generation.

Prompt Elements

- **Instruction:** What you ask the model to do.
- **Context:** Extra details to help the model answer better.
- **Input Data:** The question or data you give.
- **Output Indicator:** How you want the answer.

Prompting Examples

Text Summarization

- **Task:** Create short, understandable summaries from longer texts.
- **Example:** Ask the LLM to explain a topic in one sentence.

Example Prompt:

- **Prompt:** ”Explain the above in one sentence.”
- **Output:** ”Antibiotics are medications to stop bacterial infections, not viruses.”

Information Extraction

- **Task:** Pull out specific details from a text.
- **Example:** Specify what information to extract, e.g., a product mention.

Example Prompt:

- **Prompt:** ”Mention the large language model mentioned above.”
- **Output:** ”ChatGPT.”

Question Answering

- **Task:** Get direct answers to questions.
- **Example:** Provide context, question, and tell the LLM to be precise.

Example Prompt:

- **Prompt:** ”What was OKT3 originally sourced from?”
- **Output:** ”Mice.”

Text Classification

- **Task:** Label texts based on content or sentiment.
- **Example:** Instruct the LLM to categorize as neutral, negative, or positive.

Example Prompt:

- **Prompt:** ”Classify the sentiment: ’The food was okay.’”
- **Output:** ”Neutral.”

Conversation

- **Task:** Make the LLM talk like a character or in a particular style.
- **Example:** Make it sound technical or simple for different audiences.

Example Prompt:

- **Prompt:** ”AI, tell me about black holes.”
- **Output:** ”Black holes are like space vacuums...”

Code Generation

- **Task:** Write computer code based on requirements.
- **Example:** Tell the LLM to write code for a greeting or a database query.

Example Prompt:

- **Prompt:** ”Ask for the user’s name and greet them.”

- **Output:** ”let name = prompt(’Your name?’); console.log(‘Hello, *name*!‘);”

Reasoning

- **Task:** Solve problems or puzzles that need thinking.
- **Example:** Correct the LLM if it makes an error and refine the prompt.

Example Prompt:

- **Prompt:** ”Add the odd numbers: 15, 32, 5, 13, 82, 7, 1.”
- **Output:** ”The sum of odd numbers is 41, which is an odd number.”

Zero-Shot Prompting

- Big AI models like GPT-3 can do tasks with no training examples, called ”zero-shot.”
- Tried zero-shot in last part.
- Example: Asked AI to label text as happy, sad, or okay with no examples. AI said the ”okay” vacation was ”neutral.”
- If zero-shot does not work, give the AI examples to help it learn, called ”few-shot.”

Few-Shot Prompting

- What is Few-Shot Prompting?
  - Few-shot prompting is a method to teach a language model how to do a task. You give the model a few examples, and it learns from them.
- Why Use Few-Shot Prompting?
  - Helps the model perform better complex tasks.
  - Needed when zero-shot (no examples) is not working well.
- How to Do Few-Shot Prompting:
  1. Give a few examples of how to do the task in the prompt.
  2. Test with different numbers of examples (like 1-shot, 3-shot).
- Tips for Better Results:
  - Use examples with labels and inputs that match your task.
  - Keep a consistent format for the examples.
  - Random labels can work if they fit the overall pattern.
- Limitations:
  - Few-shot prompting may not always work for hard tasks that need more thinking.

Chain-of-Thought Prompting (CoT)

- CoT is a method where you solve problems by showing the steps you take to find the answer. It’s like explaining your thinking on paper.
- Example:
  - Prompt: ”A group has odd numbers adding up to an even number: 3, 5, 7. True or False?”
  - Output: ”Adding the odd numbers (3, 5, 7) gives 15, which is odd. So, False.”

In Zero-shot CoT Prompting

- This is doing CoT without showing examples first. You just tell the machine to ”think step by step.”
- With CoT:
  - Prompt: ”I buy 8 candies and eat 2. Let’s think step by step.”
  - Output: ”Start with 8 candies, eat 2, you have 6 left.”

Automatic Chain-of-Thought (Auto-CoT)

- Auto-CoT is about getting a machine to do CoT by itself without much help. It chooses different questions and makes its own examples.
- Auto-CoT Steps:
  1. Group questions into types.
  2. Pick one question from each type.
  3. Tell the machine to ”think step by step” for these questions.

Self-Consistency in Prompt Engineering

- Definition: An advance method to improve answers by generating multiple reasoning paths and choosing the most consistent one.
- How it works:
  1. Create multiple questions and answers showing the reasoning process. (like few-shot)
  2. Ask the original question, repeatedly.
  3. Compare answers from different prompts.
  4. Choose the most common, consistent answer.
- Shortly: Use examples to teach the AI about reasoning. Compare different AI responses. Pick the answer that shows up the most.

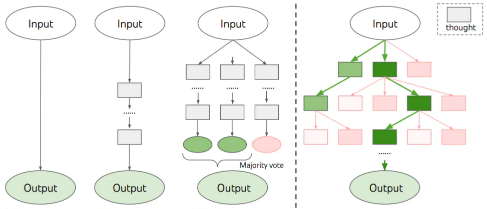
Generated Knowledge Prompting

- Definition: A method to improve LLMs by creating knowledge to guide the model’s predictions, especially for tasks like commonsense reasoning. Using generated knowledge leads to more accurate model responses.
- Steps:
  1. Recognize LLM limitations (e.g., they may not understand golf scores should be low, not high).
  2. Generate relevant knowledge (e.g., explain real golf scoring rules).
  3. Use knowledge to correct and guide the model’s answers.
- Usage
  - Give generated knowledge as context inside prompt.

Tree of Thoughts (ToT)

- What is ToT?
  - A problem-solving framework to help LMs plan, recheck, and forecast for better problem-solving.
- How does it work?
  - Uses a ”tree” with ”nodes” for each thought/step.
  - LM generates and evaluates thoughts for reasoning.
- Methodology:
  - Similar to a choose-your-adventure book.
  - Searches paths with methods like BFS and DFS.
- Practical Use:
  - Breaks down complex problems (e.g., math) into steps.
  - LM finds various solutions, picks best ones.
- ToT Controller:
  - Trained to improve search strategy over time.
- Group-Thinking:
  - LMs discuss steps like a team of experts.

The difference of Chain Technics



(a) Input-Output Prompting (IO) (c) Chain of Thought Prompting (CoT) (c) Self Consistency with CoT (CoT-SC) **(d) Tree of Thoughts (ToT)**

Retrieval Augmented Generation (RAG)

- What is RAG?
  - A method that adds external knowledge to language models.
  - Helps language models give more factual and reliable answers.
  - Used for complex tasks that need lots of information.
- How does RAG work?
  1. Takes a question or prompt.
  2. Looks for related documents from a source (like Wikipedia).
  3. Puts together the found documents and the input.
  4. The text generator makes a final answer using this information.
- Benefits of RAG:
  - Keeps answers up-to-date without retraining the whole model.
  - Better for tasks where facts change over time.
  - More accurate and detailed answers.
- RAG's Parts:
  - Parametric memory: A trained model remembers patterns and data.
  - Non-parametric memory: An index with Wikipedia articles for extra facts.
- Why use RAG?
  - It can improve language models on tough questions.
  - Makes sure language models use the latest facts.
  - Shows better results on different tests and questions.
- Recent Trends:
  - More use of RAG in popular language models to get better at answering questions.
  - RAG makes language models smarter by using the most current information.

Automatic Prompt Engineer (APE)

- APE Definition:
  - APE is a framework for making and picking instructions automatically. It uses big language models to make and find the best instructions.
- How APE Works:
  1. Large language model makes different instructions.
  2. Instructions are tried using a target model.
  3. Best instruction is picked from how well it works.
- Benefits of APE:
  - Finds better prompts than humans.
  - Makes chain-of-thought (CoT) reasoning better.
- Key Papers on Prompt Engineering:
  - Prompt-OIRL: Makes prompts based on questions using a special learning method.
  - OPRO: Lets language models make better prompts by using them in a clever way.
  - AutoPrompt: Makes prompts for many tasks using a way that follows where things change a lot.
  - Prefix Tuning: Adds changeable pieces before text for making natural language.
  - Prompt Tuning: Learns prompts that can change with a method that goes backwards.

Directional Stimulus Prompting

- Definition: A technique to improve summary generation by guiding a large language model (LLM) with hints. You just include the keypoints and keywords as hint in your prompt.

ReAct Prompting

- Definition:
  - ReAct Prompting: A framework using Large Language Models (LLMs) to create *reasoning traces* (thinking) and *task-specific actions* for problem-solving. Helps to update knowledge and handle exceptions.
- Advantages:
  - Works well for language and decision-making tasks.
  - Improves reliability and trust in LLMs' responses.
  - Aids in obtaining factual information by interacting with tools.
  - Performs better than acting alone or just chain-of-thought (CoT) in tests.
- How it Works:
  - Combines *acting* (doing tasks) and *reasoning* (thinking through tasks).
  - Can access external information (like searching the internet).
- Steps in ReAct Prompting to create Final Answer:
  1. Thought: Formulate a plan or understanding of the task.
  2. Action: Carry out a step or search for information.
  3. Observation: Look at results and external information retrieved.
  4. Repeat: Adjust reasoning and act again if needed.

Multimodal CoT Prompting

- Definition: A way of using both text and pictures to help AI think step by step.