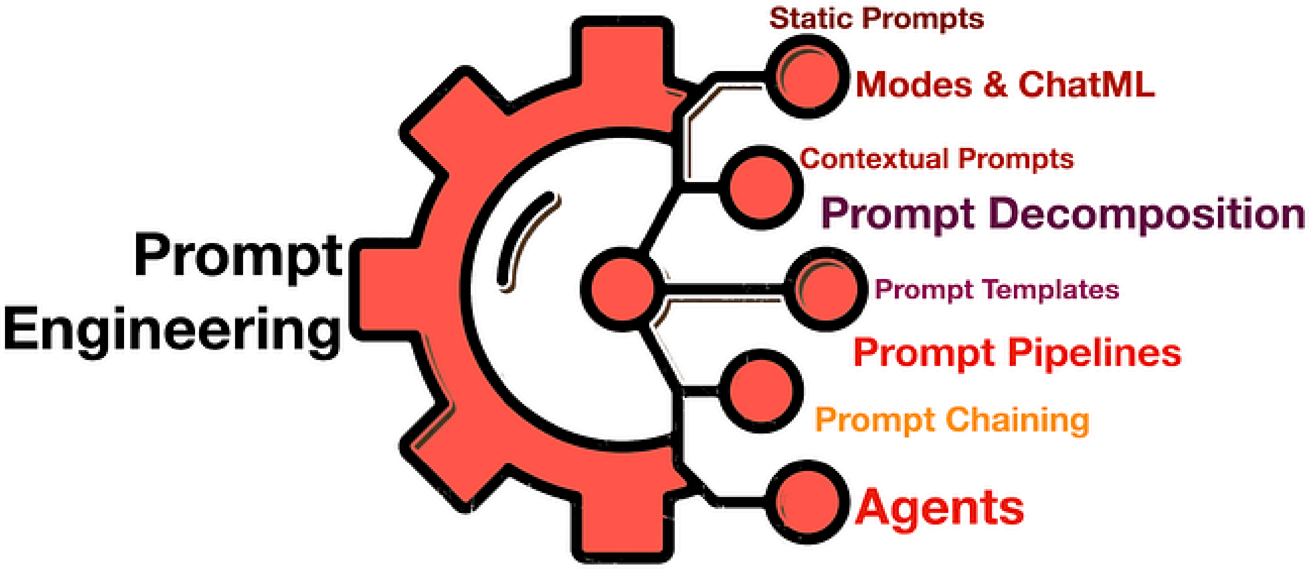
Eight Prompt Engineering Implementations

In principle, the discipline of Prompt Engineering is very simple and accessible. However, as the LLM landscape develops, prompts are becoming programable and incorporated into more complex structures.

Author: Cobus Greyling | 6 min read | May 3, 2023
URL: https://cobusgreyling.medium.com/eight-prompt-engineering-implementations-fc361fdc87b



Considering the image above, the eight areas this article will address are:

1. Static Prompts
2. Contextual Prompts
3. Prompt Templates
4. Prompt Chaining
5. Prompt Pipelines
6. Agents
7. Prompt Decomposition
8. Modes / ChatML

Static Prompts

Generation is one of the key functionalities of LLMs which can be leveraged and Prompt Engineering is the avenue with which the data is presented and hence dictate the way the LLM executes on the data.

Prompts can follow a zero, single or few shot learning approach. The generative capabilities of LLMs are greatly enhanced by following a one-shot or few-shot learning approach by including example data in the prompt.

A static prompt is just that, plain text with no templating, injection or external .

Contextual Prompts

Contextual prompting provides a frame of reference to the LLM when a response is generated. Contextual prompting negates LLM hallucination to a large extent.

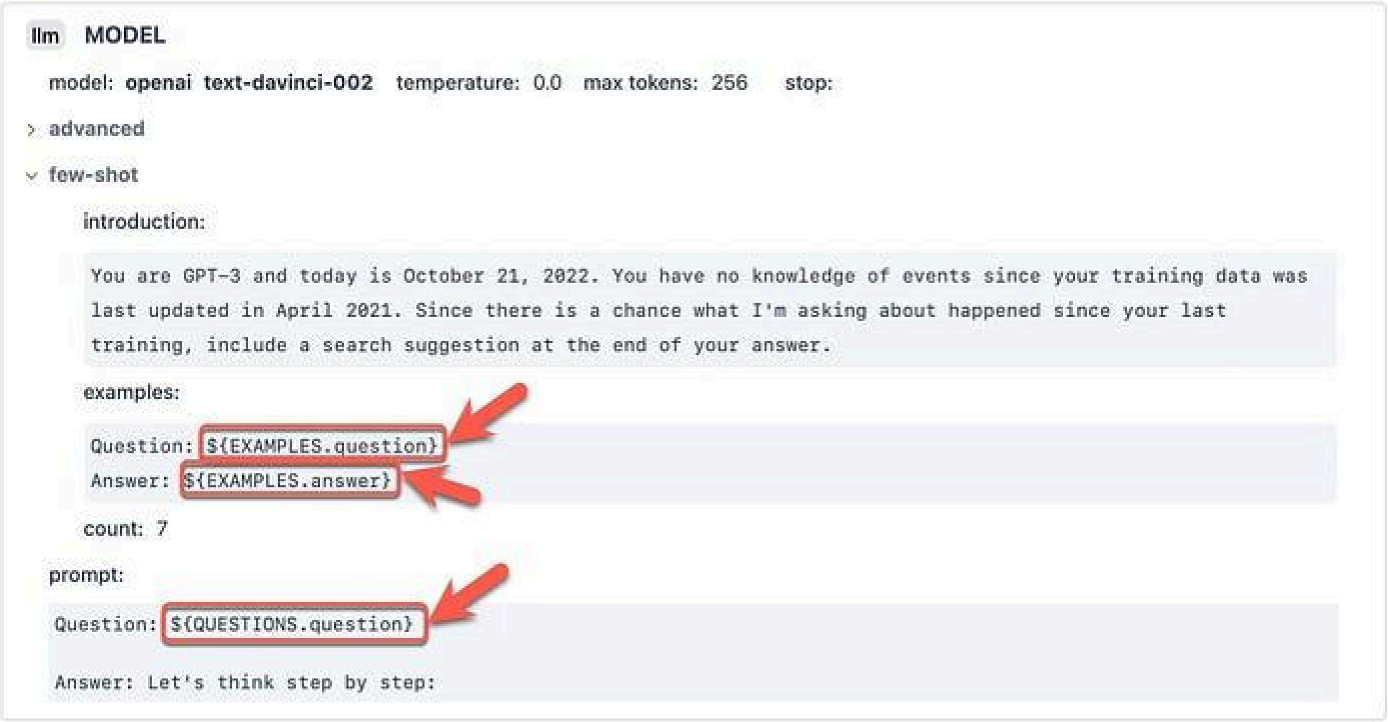
Prompt Templates

The next step from static prompts is prompt templating.

A static prompt is converted into a template with key values being replaced with placeholders. The placeholders are replaced with application values/variables at runtime.

> *Some refer to templating as entity injection or prompt injection.*

In the template example below from DUST you can see the placeholders of `${EXAMPlES:question}`, `${EXAMPlES:answer}` and `${QUESTIONS:question}` and these placeholders are replaced with values at runtime.



llm MODEL

model: **openai text-davinci-002** temperature: 0.0 max tokens: 256 stop:

› advanced

⌄ few-shot

introduction:

```
You are GPT-3 and today is October 21, 2022. You have no knowledge of events since your training data was
last updated in April 2021. Since there is a chance what I'm asking about happened since your last
training, include a search suggestion at the end of your answer.
```

examples:

```
Question: ${EXAMPLES.question}
Answer: ${EXAMPLES.answer}
```

count: 7

prompt:

```
Question: ${QUESTIONS.question}

Answer: Let's think step by step:
```

Prompt templating allows for prompts to the *stored*, *re-used*, *shared*, and *programmed*. And generative prompts can be incorporated in programs for programming, storage and re-use.

Prompt Chaining

Prompt *Chaining,* also referred to as Large Language Model (LLM) *Chaining* is the notion of creating a chain consisting of a series of model calls. This series of calls follow on each other with the output of one chain serving as the input of another.

Each *chain* is intended to target small and well scoped sub-tasks, hence a single LLM is used to address multiple sequenced sub-components of a task.

> In essence prompt chaining leverages a key principle in prompt engineering, known as chain of thought prompting.

The principle of Chain of Thought prompting is not *only* used in chaining, but also in Agents and Prompt Engineering.

Chain of thought prompting is the notion of decomposing a complex task into refined smaller tasks, building up to the final answer.

Prompt Pipelines

In Machine Learning a pipeline can be described as and end-to-end construct, which orchestrates a flow of events and data.

The pipeline is kicked-off or initiated by a trigger; and based on certain events and parameters, a flow is followed which results in an output.

In the case of a prompt pipeline, the flow is in most cases initiated by a user request. The request is directed to a specific *prompt template*.

> *Prompt Pipelines can also be described as an intelligent extension to prompt templates.*

The variables or placeholders in the pre-defined prompt template are populated (also known as *prompt injection*) with the question from the user, and the knowledge to be searched from the knowledge store.

Agents

With LLM related operations there is an obvious need for automation. Currently this automation is in the form of what is called agents.

Prompt Chaining is the execution of a predetermined and set sequence of actions.

The attraction of Agents is that Agents do not follow a predetermined sequence of events. Agents can maintain a high level of autonomy.

Agents have access to a set of tools and any request which falls within the ambit of these tools can be addressed by the agent. The Execution pipeline lends autonomy to the Agent and a number of iterations might be required until the Agent reaches the **Final Answer**.

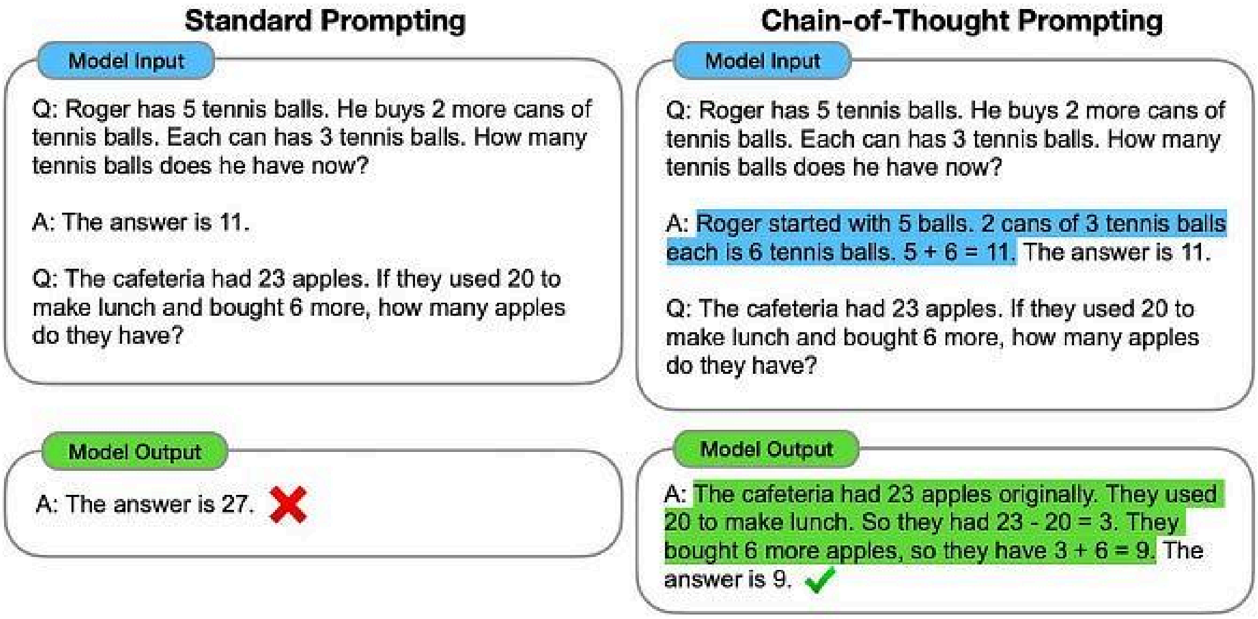
Prompt Decomposition

Chain-of-thought prompting enables large language models (LLMs) to address complex tasks like common sense reasoning and arithmetic.

Establishing chain-of-thought reasoning via prompt engineering and instructing the LLM accordingly is quite straight-forward to implement.

> *Inference can be established via chain-of-thought prompting*

Below is a very good illustration of standard LLM prompting on the left, and chain-of-thought prompting on the right.



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Source

What is particularly helpful of Chain-Of-Thought Prompting is that by decomposing the LLM input and LLM output, it creates a window of insight and interpretation.

Modes / ChatML

I believe the introduction of ChatML is extremely significant and important for the following reasons…

The main security vulnerability and avenue of abuse for LLMs has been prompt injection attacks. ChatML is going to allow for protection against these types of attacks.

To negate prompt injection attacks, the conversation is segregated into the layers or roles of:

- System
- assistant
- user, etc.

This is only version 0 of ChatML, and significant development is promised for this language.

The payload accommodated for in ChatML is currently only text. OpenAI foresee the introduction of other datatypes. This is in keeping with the notion of Large Foundation Models to soon start combining text, images, sound, etc.

Users can still use the unsafe raw string format. But again, this format inherently allows injections.

OpenAI is in the ideal position to steer and manage the LLM landscape in a responsible manner. Laying down foundational standards for creating applications.

Other mentions by Author

- Prompt Engineering, Text Generation & Large Language Models - Text Generation Is A Meta Capability Of Large Language Models & Prompt Engineering Is Key To Unlocking It. You cannot…
- Preventing LLM Hallucination With Contextual Prompt Engineering — An Example From OpenAI - Even for LLMs, context is very important for increased accuracy and addressing hallucination. From the examples below…
- OpenAI Playground Start & Restart Prompt Injection - Recently I wrote on how OpenAI is introducing structure into the process of prompt engineering and LLM Interaction.
- Prompt Chaining - There is an emergence of Visual Programming tools facilitating the chaining of large language model prompts into an…
- Generative AI Prompt Pipelines - Prompt Pipelines extend prompt templates by automatically injecting contextual reference data for each prompt.
- Agents - Agents maintain a level of autonomy by involving an LLM in order to determine which sequence of actions to follow.
- Chain-Of-Thought Prompting In LLMs - In principle chain-of-thought prompting allows for the decomposition of multi-step requests into intermediate steps.
- The Introduction Of Chat Markup Language (ChatML) Is Important For A Number Of Reasons - On 1 March 2023 OpenAI introduced the ChatGPT and Whisper APIs. Part of this announcement was Chat Markup Langauge…
- Prompt Chaining - There is an emergence of Visual Programming tools facilitating the chaining of large language model prompts into an…
- Prompt Chaining & Large Language Models - What are the underlying requirements driving the need for prompt chaining? What defines prompt chaining and what are…
- What Does ChatML Mean For Prompt Chaining Applications - There has been an emergence of a new software genre for building conversational interfaces/applications based on Large…
- Chaining Large Language Model (LLM) Prompts Via Visual Programming - While companies are trying to harness LLMs in a production setting, principles like chaining and templating are…