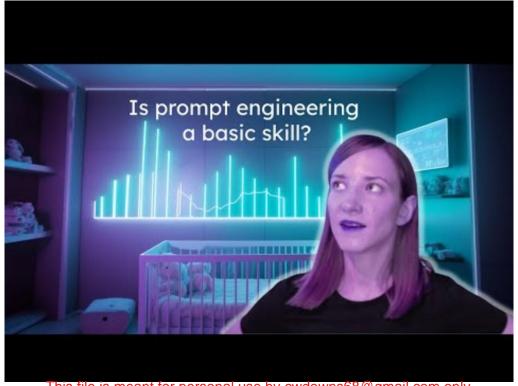


What is Prompt Engineering?







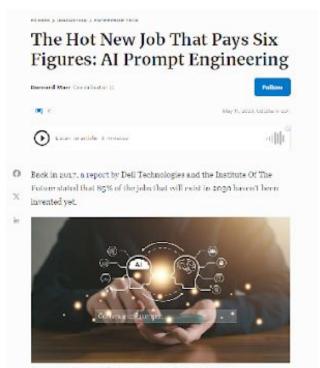
What is Prompt Engineering?

- Prompt Engineering is the art and science of designing and structuring prompts (questions or tasks) fed to language models.
- It is a collection of strategies and methods to describe a task in text for an AI model.
- Improve the performance of the model on tasks.





Why Do We Engage in Prompt Engineering?



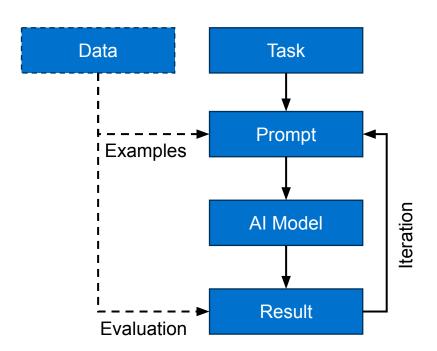
- To adapt the model to a wide range of tasks and applications, offering flexibility.
- To guide the model's responses, leading to more accurate and relevant results.
- To get the desired output efficiently, saving computational resources.

This file is meant for personal use by cwdowns68@gmail.com only.





The Basic Elements of an Effective Prompt

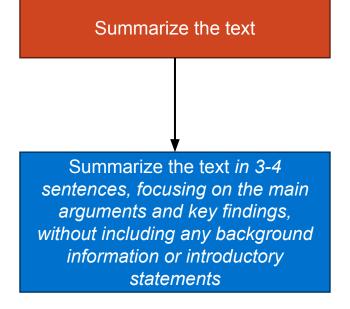


- Effective prompts normally have a couple of basic elements:
 - Clear instruction
 - Context
 - Output indicator
- Prompting frequently also benefits from additional elements:
 - Examples
 - Embodiment
 - Task breakdown





Clear Instructions in a Prompt



- Clearly define the task you want the model to do.
- Provide necessary details to do the task.
- Avoid vague language.
- For difficult to define tasks, providing examples (i.e., few-shot prompting) can help.





Adopting a Persona in a Prompt

You are literary expert who focuses on early medieval literature.

Analyze and describe the tone of this passage in a few sentences

Have the AI model play a role.

 The role should be related to the task you want the model to do.

 Providing descriptive elements to the role can also improve performance (e.g., "experienced", "from [location]", etc.).





Adding Context to a Prompt

"Why is cybersecurity important in healthcare?"

"Why is cybersecurity important in healthcare? Consider this context: Healthcare systems store sensitive patient information,...."

Using this context, explain why cybersecurity is especially crucial in the healthcare industry.

- Provide additional details to the prompt.
 - Forms the basis of Retrieval Augmented Generation.
- Greatly helps with problems like Hallucination.
- Can be used to augment AI models with knowledge outside of their training.





Adding Text Delimiters to a Prompt

Answer the question based on this passage

Based on the [Passage] provided, answer the [Question] accurately and concisely: [Passage]: " ..."

[Question]: "What are some of the fields impacted by recent Al developments?" [Answer]:

- Provide textual cues for elements of the prompt.
 - Task
 - Examples
 - Output indicator
- Cue words like "example:" or "output:" can mark particular actions fort he model.
- Delimiters like ""..." or [...] can demarcate special sections of text.





Breaking Down a Task in a Prompt

Analyze the following passage for tone, main ideas, and any notable rhetorical devices.

Analyze the following passage by completing each step below:

- Identify and describe the tone of the passage in one sentence.
- List the two main ideas conveyed in the passage.
- Highlight any rhetorical devices (e.g., metaphors, alliteration) used, and briefly use by cwdowns68@gmail.com only. explain their effect.

- Specify the steps required to do a task.
- Break the task into different sub-tasks which can be done over successive prompts.
- For tasks with lots of different instruction sets, classify a type of query and then use the appropriate instruction set.

in part or full is liable for legal action.





Specifying Output in a Prompt

Analyze and describe the sentiment of this passage

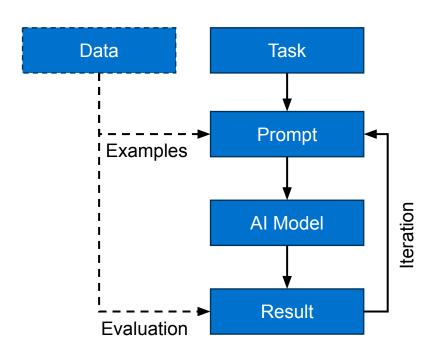
Analyze and describe the sentiment of this passage. Only respond with the words "positive", "negative", or "neutral".
Sentiment:

- Be descriptive about the output desired.
 - Word or sentence length
 - Special words to use (i.e., in a classification task)
- Provide an output indicator or word to cue the model (e.g., ":").





Evaluating the Quality of a Prompt

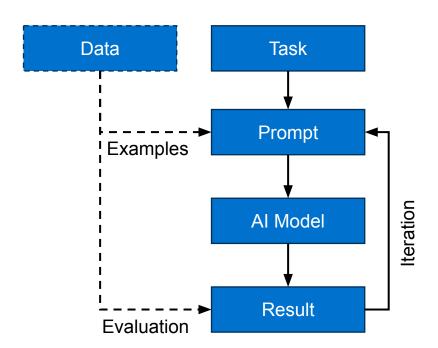


- Evaluation against performance indicators
 - Consistency
 - Completeness and specificity of the output
 - Comparison to benchmarks
- Test with variations and red-teaming
 - Ambiguous or harmful input
 - Especially for user facing applications

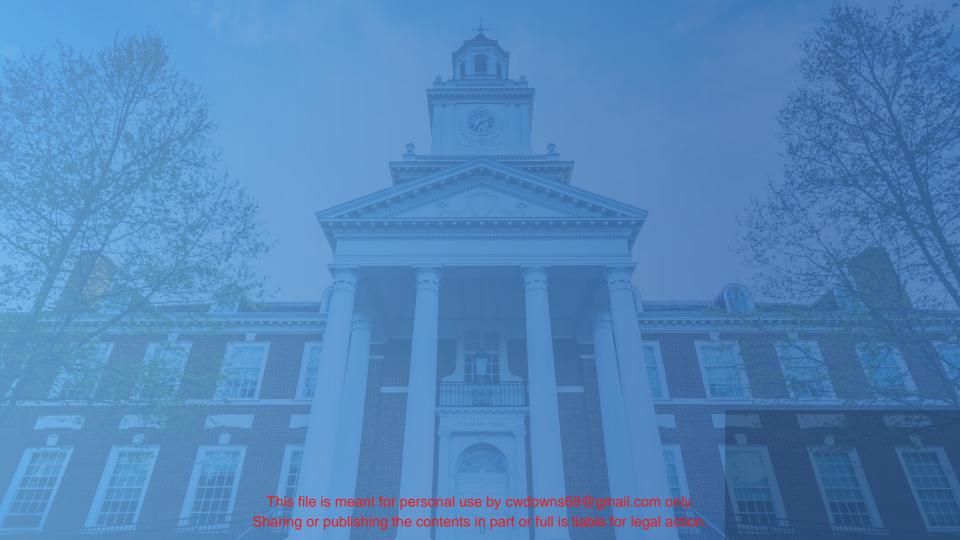




Bringing it All Together



- Prompt engineering is often an iterative process.
 - It helps to have some kind of validation data of what you are looking for.
 - Test changes systematically.
- Good prompts frequently use all or many of the aspects previously described.
- Ultimately, you want to structure your inputs to AI models to optimize your outputs from them.



Agentic AI Prompt Engineering: Common Design Patterns





Common Prompting Design Patterns

Prompt Engineering

(Effective communication & collaboration with AI)

Prompt Engineering is art and science of crafting inputs(prompts) to AI models to get the desired output

Techniques

Zero-shot
One-shot
Few-shot
Chain of Thought
Self Consistency
Generate Knowledge
Automatic Prompt Engineering
Active Prompt
Directional Stimulus
ReAct
Multimodal CoT
Graph Prompting

Use Cases

Text Summarization
Question Answering
Code Generation
Role Playing
Text Classification
Reasoning
Art Generation
Grammar correction
Bug finding
Language Translation
Idea Generation
& many more

- There are a number of prompt formats, or design patterns, for various types of tasks.
- Some of the most successful feature:
 - Using examples and/or context data.
 - Having the AI model engage in some form of deliberate reasoning.

Best Understand the model's capabilities and limitations
Use clear and specific language
Practices Provide examples and feed

Explain the context in as much detail as possible Experiment with different formats and styles Evaluate and refine $^{\uparrow}$

Vaj, Tiya, Prompt Engineer (2024)





Few Shot Prompting

```
Translate English to French: 

task description

sea otter => loutre de mer 

peppermint => menthe poivrée

plush girafe => girafe peluche

cheese => 

prompt
```

Brown et al., Language Models are Few-Shot Learners (2020)

- Provide the model with several examples of the task.
- Example: For a translation task, provide multiple pairs of sentences in two languages.
- Cautions on use:
 - The order and choice of examples can greatly influence the outcome.





Chain-of-Thought Prompting

Standard Prompting

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

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.

Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (2022)

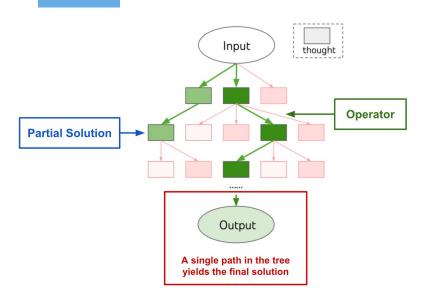
- Break down complex tasks into a series of simpler tasks, guiding the model through a chain of reasoning.
- Example: For an arithmetic task, start with simple operations and build up to the more complex operation.
- Cautions on use:
 - Increased computation cost
 - Design of intermediate steps

This file is meant for personal use by cwdowns68@gmail.com only. Sharing or publishing the contents in part or full is liable for legal action.





Tree-of-Thought Prompting



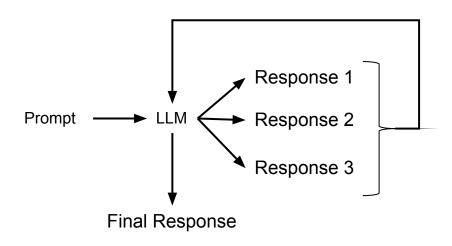
Yao et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Tree of Thoughts: Deliberate Problem Solving with Large Language Models (2023)

- Guide the model to explore multiple reasoning paths by branching into different lines of thought.
- Example: For solving a logic puzzle, prompt the model to generate multiple hypotheses and evaluate each one before converging on a solution.
- Cautions on use:
 - Increased computation cost (more than CoT)
 - Design of merge and prune steps





Self-Consistency Prompting



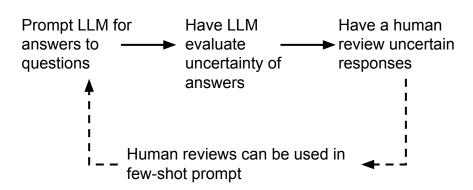
Wang et al., Self-consistency improves chain-of-thought reasoning in language models (2023)

- Reminded the AI model of previous parts of the conversation or ask it to ensure its responses are consistent with prior statements.
- Example: For a question-answering task, produce several possible answers and then seek consistency across the answers.
- Cautions on use:
 - Increased computation cost
 - Risk of Majority Bias





Active Prompting



Diao et al., Active Prompting with Chain-of-Thought for Large Language Models (2023)

- AI model proactively seeks additional information or clarification from the user to improve its understanding of the task or prompt.
- Example: Having the model ask questions of the user about the how to classify something before doing the task
- Cautions on use:
 - Relies on human expertise



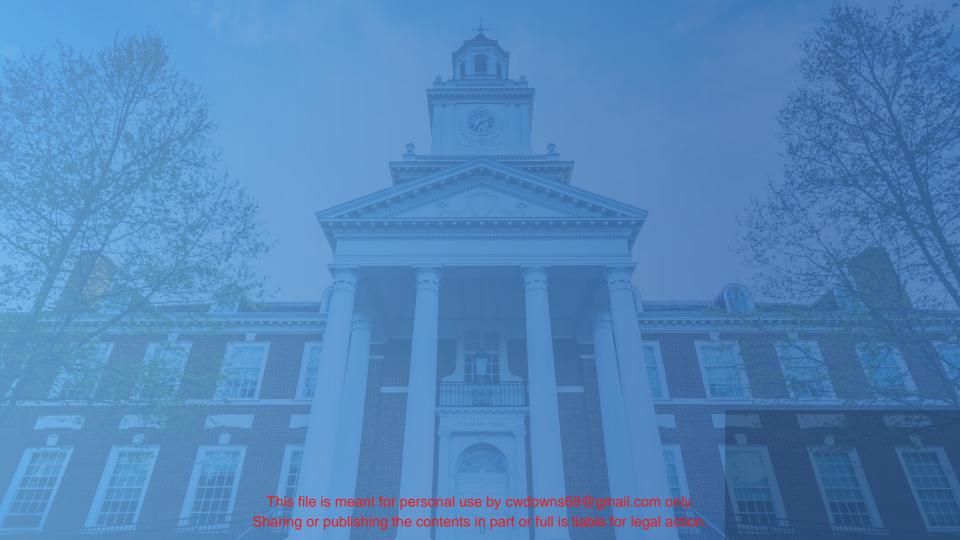


Action-Observation Prompting

```
(1d) ReAct (Reason + Act)
Thought 1: I need to search Apple Remote and find the
program it was originally designed to interact with.
Act 1: Search [Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in
October 2005 by Apple ... originally designed to control the
Front Row media center program ...
Thought 2: Apple Remote was originally designed to control
the Front Row media center program. I need to search
Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row
Seat to Earth', 'Front Row Motorsports',' Front Row
(software) ', ...]
Thought 3: Front Row is not found. I need to search
Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software ...
Thought 4: Front Row (software) is controlled by an Apple
Remote or the keyboard function keys. So the answer is
keyboard function keys.
Act 4: Finish[keyboard function keys
```

- Alternates between performing actions (e.g., reasoning steps) and observing outcomes, using feedback to refine responses.
- Example: For a question-answering task, the model forms a hypothesis, retrieves relevant data, adjusts based on findings, and refines the answer.
- Cautions on use:
 - Complexity in prompt design

Yao et al., ReAct: Synergizing Reasoning and Acting in Language Models (2022)



Agentic AI Prompt Engineering: Working with LangChain &

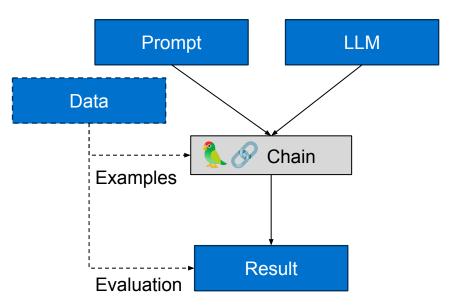
This file is meant for personal use by cwdowns68@gmail.com only.

Sharing or publishing the contents in part or full is liable for legal action





LangChain Core Concepts

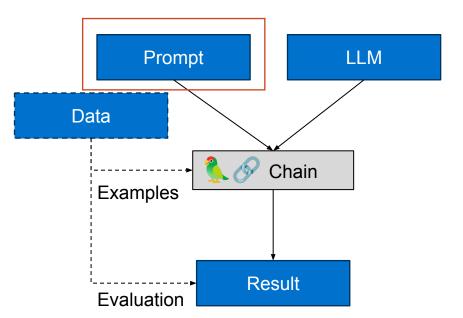


- LangChain uses the concept of "Chains" to link together things like models and prompts, and control their execution.
- For prompt engineering, the components of our chains will center around:
 - Prompt templates
 - LLMs
 - Chains





Prompt Templates

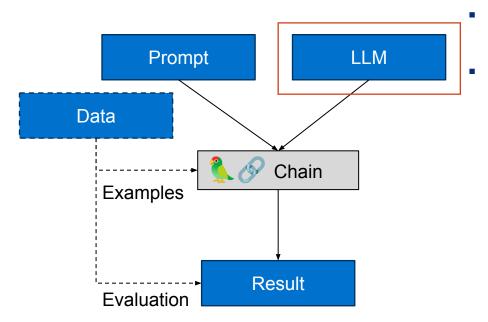


- Prompts are stored in Prompt templates, which consist of:
 - Text for the prompt with `{}' for inserting variables
 - A Prompt Template object that controls inputs and outputs
- There are a number of Prompt Template classes, but the most important are the basic BasePromptTemplate and FewShotPromptTemplate





LLMs

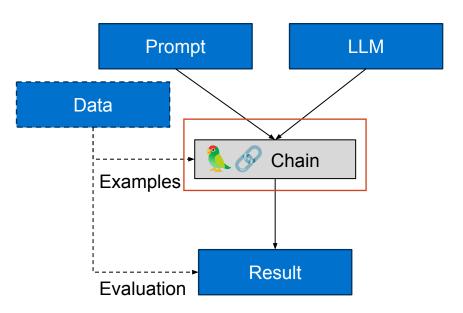


- LangChain uses an abstraction to wrap models for inclusion into Chains.
- Typically each major source of LLMs (i.e., OpenAI, Cohere, Huggingface) has a specific class for wrapping those models (or API calls to those models)
 - Model classes are also typically broken between "chat" and "llm" classes depending on the particular model





Chains

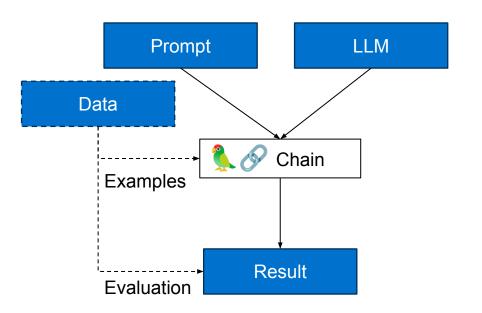


- Chains are the reusable components that link together things like LLMs and prompts
- Chains are composable with other objects or "runnables", including other chains
- Chains typically link together runnables through the "|" operator

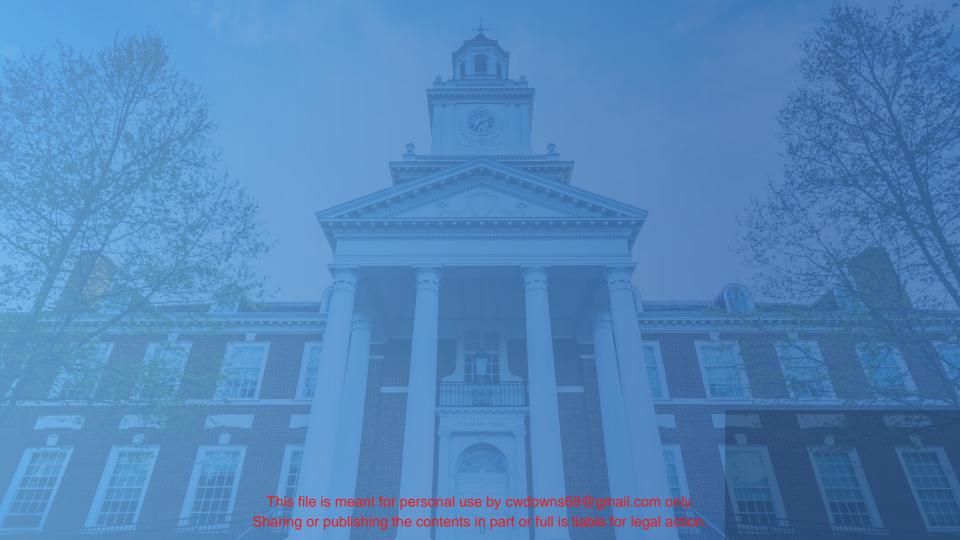




A Final Note about LangChain



- In this section, we focused in on the elements of LangChain that most directly apply to prompt engineering
- LangChain has many other elements for things like Agents, memory, and API calls, which can also be incorporated in chains
- I invite you to explore the documentation to find other chains or components of chains!



References

- Briggs, J., & Ingham, F. (n.d.). LangChain AI Handbook. Pinecone. Retrieved from https://www.pinecone.io/learn/series/langchain/
- Kadel, S. N. (2024). Prompt Engineering: The Art of Getting What You Need From Generative AI. Ivan Allen College of Liberal Arts. Retrieved from https://iac.gatech.edu/featured-news/2024/02/AI-prompt-engineering-ChatGPT
- Cummings, L. (2023, September 26). Why AI Whispering is a Myth: There's nothing magical about structured prompt operations.
 Cyborgs Writing. Retrieved from https://www.isophist.com/p/the-anatomy-of-a-prompt
- OpenAI Platform Team. (n.d.). Prompt engineering. OpenAI. Retrieved from https://platform.openai.com/docs/quides/prompt-engineering
- DAIR.AI. (2024). Prompt Engineering Guide. Retrieved from https://www.promptingquide.ai/
- Diao, S., Wang, Y., Chen, Y., Zhang, R., & Tao, D. (2023). Active prompting with chain-of-thought for large language models. In Proceedings of the 37th International Conference on Machine Learning (ICML). arXiv preprint arXiv:2302.12246.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. In Proceedings of the 39th International Conference on Machine Learning (ICML). arXiv preprint arXiv:2210.03629.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. In Proceedings of the 39th International Conference on Machine Learning (ICML). arXiv preprint arXiv:2203.11171.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems (NeurIPS), 33, 1877-1901. arXiv preprint arXiv:2005.14165.

This file is meant for personal use by cwdowns68@gmail.com only. Sharing or publishing the contents in part or full is liable for legal action.

