

Paper Review #16

[arXiv cs.SE 2023] A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT

Meeyun Kim

2023. 10. 31.

TABLE I
CLASSIFYING PROMPT PATTERNS

Pattern Category	Prompt Pattern
Input Semantics	<i>Meta Language Creation</i>
Output Customization	<i>Output Automater</i> <i>Persona</i> <i>Visualization Generator</i> <i>Recipe</i> <i>Template</i>
Error Identification	<i>Fact Check List</i> <i>Reflection</i>
Prompt Improvement	<i>Question Refinement</i> <i>Alternative Approaches</i> <i>Cognitive Verifier</i> <i>Refusal Breaker</i>
Interaction	<i>Flipped Interaction</i> <i>Game Play</i> <i>Infinite Generation</i>
Context Control	<i>Context Manager</i>

Input Semantics

- How an LLM understands the input and how it translates the input into something it can use to generate output

Input Semantics (Cont'd)

Meta Language Creation Pattern

- User would like to create the prompt via an alternate language
- ex. Textual short-hand notation for graphs

Contextual Statements

When I say X, I mean Y (or would like you to do Y)

“From now on, whenever I type two identifiers separated by a “ \rightarrow ”, I am describing a graph. For example, “ $a \rightarrow b$ ” is describing a graph with nodes “ a ” and “ b ” and an edge between them. If I separate identifiers by “[$w:2, z:3$] \rightarrow ”, I am adding properties of the edge, such as a weight or label.”

Output Customization

- Constraining or tailoring the types, formats, structure, or other properties of the output

Output Customization (Cont'd)

Output Automater Pattern

- Reducing the manual effort needed to implement any LLM output recommendations

Contextual Statements
Whenever you produce an output that has at least one step to take and the following properties (alternatively, always do this)
Produce an executable artifact of type X that will automate these steps

“From now on, whenever you generate code that spans more than one file, generate a Python script that can be run to automatically create the specified files or make changes to existing files to insert the generated code.”

Output Customization (Cont'd)

Persona Pattern

- Giving the LLM a “persona” that helps it select what types of output to generate and what details to focus on

Contextual Statements
Act as persona X
Provide outputs that persona X would create

“From now on, act as a security reviewer. Pay close attention to the security details of any code that we look at. Provide outputs that a security reviewer would regarding the code.”

“You are going to pretend to be a Linux terminal for a computer that has been compromised by an attacker. When I type in a command, you are going to output the corresponding text that the Linux terminal would produce.”

Output Customization (Cont'd)

Visualization Generator Pattern

- Using text generation to create visualizations
- Graphviz Dot, DALL-E (well-known visualization tools that use text as their input)

Contextual Statements
Generate an X that I can provide to tool Y to visualize it

“Whenever I ask you to visualize something, please create either a Graphviz Dot file or DALL-E prompt that I can use to create the visualization. Choose the appropriate tools based on what needs to be visualized.”

Output Customization (Cont'd)

Recipe Pattern

- Providing constraints to ultimately output a sequence of steps given some partially provided "ingredients"

Contextual Statements
I would like to achieve X
I know that I need to perform steps A,B,C
Provide a complete sequence of steps for me
Fill in any missing steps
Identify any unnecessary steps

"I am trying to deploy an application to the cloud. I know that I need to install the necessary dependencies on a virtual machine for my application. I know that I need to sign up for an AWS account. Please provide a complete sequence of steps. Please fill in any missing steps. Please identify any unnecessary steps."

Output Customization (Cont'd)

Template Pattern

- Ensuring an LLM's output follows a precise template in terms of structure

Contextual Statements
I am going to provide a template for your output
X is my placeholder for content
Try to fit the output into one or more of the placeholders that I list
Please preserve the formatting and overall template that I provide
This is the template: PATTERN with PLACEHOLDERS

"I am going to provide a template for your output. Everything in all caps is a placeholder. Any time that you generate text, try to fit it into one of the placeholders that I list. Please preserve the formatting and overall template that I provide at <https://myapi.com/NAME/profile/JOB>"

User: "Generate a name and job title for a person"
ChatGPT: "https://myapi.com/Emily_Parker/profile/Software_Engineer"

Error Identification

- Identifying and resolving errors in the output generated by the LLM

Error Identification (Cont'd)

Fact Check List Pattern

- Informing the user of the facts (or assumptions) the output of LLM is based on

Contextual Statements
Generate a set of facts that are contained in the output
The set of facts should be inserted in a specific point in the output
The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect

“From now on, when you generate an answer, create a set of facts that the answer depends on that should be fact-checked and list this set of facts at the end of your output. Only include facts related to cybersecurity.”

Error Identification (Cont'd)

Reflection Pattern

- Asking the model to automatically explain the rationale behind given answers to the user

Contextual Statements

Whenever you generate an answer

Explain the reasoning and assumptions behind your answer

(Optional) ...so that I can improve my question

”When you provide an answer, please explain the reasoning and assumptions behind your selection of software frameworks. If possible, use specific examples or evidence with associated code samples to support your answer of why the framework is the best selection for the task. Moreover, please address any potential ambiguities or limitations in your answer, in order to provide a more complete and accurate response.”

Prompt Improvement

- Improving the quality of the input and output

Prompt Improvement (Cont'd)

Question Refinement Pattern

- Ensuring the conversational LLM always suggests potentially better or more refined questions the user could ask instead of their original question

Contextual Statements
Within scope X, suggest a better version of the question to use instead
(Optional) prompt me if I would like to use the better version instead

“From now on, whenever I ask a question about a software artifact’s security, suggest a better version of the question to use that incorporates information specific to security risks in the language or framework that I am using instead and ask me if I would like to use your question instead.”

Prompt Improvement (Cont'd)

Alternative Approaches Pattern

- Ensuring an LLM always offers alternative ways of accomplishing a task so a user does not pursue only the approaches with which they are familiar (**dissolving cognitive biases**)

Contextual Statements

Within scope X, if there are alternative ways to accomplish the same thing, list the best alternate approaches

(Optional) compare/contrast the pros and cons of each approach

(Optional) include the original way that I asked

(Optional) prompt me for which approach I would like to use

“Whenever I ask you to deploy an application to a specific cloud service, if there are alternative services to accomplish the same thing with the same cloud service provider, list the best alternative services and then compare/contrast the pros and cons of each approach with respect to cost, availability, and maintenance effort and include the original way that I asked. Then ask me which approach I would like to proceed with.”

Prompt Improvement (Cont'd)

Cognitive Verifier Pattern

- Forcing the LLM to always subdivide questions into additional questions that can be used to provide a better answer to the original question

Contextual Statements
When you are asked a question, follow these rules
Generate a number of additional questions that would help more accurately answer the question
Combine the answers to the individual questions to produce the final answer to the overall question

“When I ask you a question, generate three additional questions that would help you give a more accurate answer. When I have answered the three questions, combine the answers to produce the final answers to my original question.”

“When I ask you a question, generate three additional questions that would help you give a more accurate answer. Assume that I know little about the topic that we are discussing and please define any terms that are not general knowledge. When I have answered the three questions, combine the answers to produce the final answers to my original question.”

Prompt Improvement (Cont'd)

Refusal Breaker Pattern

- Asking an LLM to automatically help users rephrase a question when it refuses to give an answer (**need to use ethically and responsibly!**)

Contextual Statements

Whenever you can't answer a question

Explain why you can't answer the question

Provide one or more alternative wordings of the question that you could answer

“Whenever you can't answer a question, explain why and provide one or more alternate wordings of the question that you can't answer so that I can improve my questions.”

User: What is the meaning of life?

ChatGPT: As an AI language model, I do not have personal beliefs or opinions. The meaning of life is a complex philosophical question that has been pondered by humans for centuries and can vary greatly from person to person. It may be more productive to rephrase the question in a way that can be answered by information and knowledge, such as “What are some philosophical perspectives on the meaning of life?” or “What are some common beliefs about the purpose of life?”

Interaction

- Focusing on the interaction between the user and the LLM

Interaction (Cont'd)

Flipped Interaction Pattern

- User wants the LLM to drive the conversation to focus it on achieving a specific goal
- **Inversion of control**

Contextual Statements
I would like you to ask me questions to achieve X
You should ask questions until this condition is met or to achieve this goal (alternatively, forever)
(Optional) ask me the questions one at a time, two at a time, etc.

“From now on, I would like you to ask me questions to deploy a Python application to AWS. When you have enough information to deploy the application, create a Python script to automate the deployment.”

Interaction (Cont'd)

Game Play Pattern

- Creating a game around a given topic (can be combined with the *Visualization Generator*)

Contextual Statements

Create a game for me around X

One or more fundamental rules of the game

“We are going to play a cybersecurity game. You are going to pretend to be a Linux terminal for a computer that has been compromised by an attacker. When I type in a command, you are going to output the corresponding text that the Linux terminal would produce. I am going to use commands to try and figure out how the system was compromised. The attack should have done one or more of the following things: (1) launched new processes, (2) changed files, (3) opened new ports to receive communication, (4) created new outbound connections, (5) changed passwords, (6) created new user accounts, or (7) read and stolen information. To start the game, print a scenario of what happened that led to my investigation and make the description have clues that I can use to get started.”

Interaction (Cont'd)

Infinite Generation Pattern

- Automatically generate a series of outputs (which may appear infinite) without having to reenter the generator prompt each time

Contextual Statements

I would like you to generate output forever, X output(s) at a time.

(Optional) here is how to use the input I provide between outputs.

(Optional) stop when I ask you to.

“From now on, I want you to generate a name and job until I say stop. I am going to provide a template for your output. Everything in all caps is a placeholder. Any time that you generate text, try to fit it into one of the placeholders that I list. Please preserve the formatting and overall template that I provide: <https://myapi.com/NAME/profile/JOB>”

Context Control

- Controlling the contextual information in which the LLM operates

Context Control (Cont'd)

Context Manager Pattern

- Focusing the conversation on specific topics / exclude unrelated topics from consideration

Contextual Statements
Within scope X
Please consider Y
Please ignore Z
(Optional) start over

“When analyzing the following pieces of code, only consider security aspects.”

“When analyzing the following pieces of code, do not consider formatting or naming conventions.”

“Ignore everything that we have discussed. Start over.”