

# Brief on 'A Prompt Pattern Catalog'

## Instructor

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

Google Developer Expert - ML & Cloud Champion Innovator

Published Author



# A Prompt Pattern Catalog

arXiv > cs > arXiv:2302.11382

Search...

Help

Computer Science > Software Engineering

[Submitted on 21 Feb 2023]

## A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, Douglas C. Schmidt

Prompt engineering is an increasingly important skill set needed to converse effectively with large language models (LLMs), such as ChatGPT. Prompts are instructions given to an LLM to enforce rules, automate processes, and ensure specific qualities (and quantities) of generated output. Prompts are also a form of programming that can customize the outputs and interactions with an LLM. This paper describes a catalog of prompt engineering techniques presented in pattern form that have been applied to solve common problems when conversing with LLMs. Prompt patterns are a knowledge transfer method analogous to software patterns since they provide reusable solutions to common problems faced in a particular context, i.e., output generation and interaction when working with LLMs. This paper provides the following contributions to research on prompt engineering that apply LLMs to automate software development tasks. First, it provides a framework for documenting patterns for structuring prompts to solve a range of problems so that they can be adapted to different domains. Second, it presents a catalog of patterns that have been applied successfully to improve the outputs of LLM conversations. Third, it explains how prompts can be built from multiple patterns and illustrates prompt patterns that benefit from combination with other prompt patterns.

Subjects: **Software Engineering (cs.SE)**; Artificial Intelligence (cs.AI)

Cite as: [arXiv:2302.11382](https://arxiv.org/abs/2302.11382) [cs.SE]

(or [arXiv:2302.11382v1](https://arxiv.org/abs/2302.11382v1) [cs.SE] for this version)

<https://doi.org/10.48550/arXiv.2302.11382> 



# A Prompt Pattern Catalog

Check out the [paper](#): A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT

arXiv:2302.11382v1 [cs.SE] 21 Feb 2023

## A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert,  
Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt

*Department of Computer Science*

*Vanderbilt University, Tennessee*

Nashville, TN, USA

{jules.white, quchen.fu, george.s.hays, michael.sandborn, carlos.olea, henry.gilbert,  
ashraf.elnashar, jesse.spencer-smith, douglas.c.schmidt}@vanderbilt.edu

**Abstract**—Prompt engineering is an increasingly important skill set needed to converse effectively with large language models (LLMs), such as ChatGPT. Prompts are instructions given to an LLM to enforce rules, automate processes, and ensure specific qualities (and quantities) of generated output. Prompts are also a form of programming that can customize the outputs and interactions with an LLM.

This paper describes a catalog of prompt engineering techniques presented in pattern form that have been applied to solve common problems when conversing with LLMs. Prompt patterns are a knowledge transfer method analogous to software patterns since they provide reusable solutions to common problems faced in a particular context, i.e., output generation and interaction when working with LLMs.

This paper provides the following contributions to research on prompt engineering that apply LLMs to automate software development tasks. First, it provides a framework for documenting patterns for structuring prompts to solve a range of problems so that they can be adapted to different domains. Second, it presents a catalog of patterns that have been applied successfully to improve the outputs of LLM conversations. Third, it explains how prompts can be built from multiple patterns and illustrates prompt patterns that benefit from combination with other prompt patterns.

**Index Terms**—large language models, prompt patterns, prompt engineering

### I. INTRODUCTION

Conversational large language models (LLMs) [1], such as ChatGPT [2], have generated immense interest in a range of domains for tasks ranging from answering questions on medical licensing exams [3] to generating code snippets. This paper focuses on enhancing the application of LLMs in several domains, such as helping developers code effectively and

LLM by providing specific rules and guidelines for an LLM conversation with a set of initial rules. In particular, a prompt sets the context for the conversation and tells the LLM what information is important and what the desired output form and content should be.

For example, a prompt could specify that an LLM should only generate code that follows a certain coding style or programming paradigm. Likewise, it could specify that an LLM should flag certain keywords or phrases in a generated document and provide additional information related to those keywords. By introducing these guidelines, prompts facilitate more structured and nuanced outputs to aid a large variety of software engineering tasks in the context of LLMs.

**Prompt engineering is the means by which LLMs are programmed via prompts.** To demonstrate the power of prompt engineering, we provide the following prompt:

**Prompt:** “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.”

This example prompt causes ChatGPT to begin asking the user questions about their software application. ChatGPT will drive the question-asking process until it reaches a point where it has sufficient information to generate a Python script that automates deployment. This example demonstrates the programming potential of prompts beyond conventional “generate a method that does X” style prompts or “answer this quiz question”.

Moreover, prompts can be engineered to program an LLM

# Thank You

---