Literature Review: A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT

Pedro Chaparro | 374339 | pedro.chaparro@epfl.ch
My **N**ice **L**ong **P**enguin

1 Summary

The paper starts by explaining the expanding role of conversational large language models (LLMs), such as ChatGPT, in various domains, like software development and education, and follows with an emphasis on the crucial role that **prompts**, which guide LLMs by providing specific instructions and context, play in improving the LLM capabilities and its outputs.

The authors then illustrate the power of **prompt engineering**, the method used to configure LLMs through prompts, by providing some examples, showcasing the potential for prompts to facilitate complex interactions beyond conventional tasks. Furthermore, they emphasize the broader applications of prompts, such as creating new interaction paradigms and enabling self-adaptation.

Finally, the authors draw parallels between prompt patterns and software patterns, highlighting their similarities and motivations. They present a comprehensive list of prompt patterns, organized in different categories, where each pattern is thoroughly examined with discussions on its motivation, structure, key ideas, and consequences, while showing example implementations. This part spans most of the paper.

During some prompt pattern discussions, some of their limitations are presented, like potentially exceeding the output length limitation the LLM supports or the need for additional information. Despite that, for the scope of the paper, not many limitations are presented and, for some of the existing ones, solutions are given.
Regarding the limitations of the work itself, the authors note that the presented patterns are just a handful and that much more work can be done in this area, both by refining the presented patterns, as well as exploring new ones but, they also state that, as LLM capabilities evolve, some prompt patterns may become irrelevant.

This paper is relevant and current, appealing to both academics and everyday readers, as prompting LLMs is becoming a regular part of people's lives, and so understanding how to prompt them effectively is valuable. Also, it's written in an easy-to-understand manner and uses few technical terms, making it accessible to a wide audience.

However, the work is solely based on ChatGPT, although generalized to other LLMs, and the authors do not consider the LLM context window's size, which might invalidate some prompt patterns depending on the context. Also, no benchmarks are given for the improvement the discussed prompts generate.

This study provides relevant and current insights into both the fields of prompt engineering and LLMs in general, showcasing how to better interact with these models in a very practical way, which can be used onwards by researchers and ordinary users to make their interactions more efficient and less time-consuming.

2 Strengths

The authors delve into details of prompt engineering, a very fundamental aspect when interacting with LLMs using prompts. So, this knowledge is necessary and applicable across various LLM tasks, making it indispensable for anyone engaging with such systems.

Another property that sets this paper apart is its clarity and straightforward approach. The authors, instead of complex language, opted to use simple and direct writing that ensures accessibility to all readers. This simplicity enhances understanding, ensuring that even those unfamiliar with the technical terminology of the field can grasp the concepts presented, and it pairs well with the idea presented in the previous paragraph.

Also, to better explain the different prompt patterns, the authors provide concrete examples, showing prompts that can be used for the effect, like in the *Flipped Interaction* pattern, where they shared the 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.*". Such examples serve to illustrate the practical application of prompt engineering in real-world scenarios, making the concept more tangible and understandable for readers.

Furthermore, the paper offers comprehensive coverage of prompt engineering, delving deeply into various aspects and classifications of prompts. Each prompt is meticulously analyzed, with detailed explanations provided on topics such as example implementations and consequences. This thorough exploration ensures that readers gain a comprehensive understanding of prompt engineering, its implications across different contexts, and how to better distinguish patterns and identify the most suitable pattern to use for their specific needs.

3 Weaknesses

The authors stress how crucial prompts are in shaping the quality of conversational LLM outputs, pointing out that good prompts lead to better results. However, they seem to assume the conversation context is always unlimited, which might overlook some system limitations. For instance, ChatGPT's context window is 16385 tokens (OpenAI), which maps to less of the same number of words. So, depending on the circumstance, patterns like *Flipped Interaction* and *Visualization Generator*, where several questions may be asked or additional output may be generated to allow visualization, respectively, may not fully work if the context window is exceeded. Also, they don't really consider other restrictions the system might have.

Another problem is that, although the paper presents clear insights into prompt engineering, as implied by the name, it is mainly focused on ChatGPT. While the principles discussed could theoretically apply to other LLM frameworks, the lack of experimentation with those different models limits the scope of the findings.
For instance, despite some not being released at the time of the writing, there are currently numerous other LLMs available, such as OpenAI's ChatGPT Plus with DALL-E, Google's Gemini, Microsoft's Copilot, Anthropic's Claude, and Perplexity's AI, all with their purposes, advantages, and disadvantages. So, a possible new iteration of this research could leverage these multiple LLMs and provide benchmarks of the prompt patterns across all of them, while also possibly identifying redundant patterns in some, as well as discovering new prompt patterns, like for the image generation feature available in few.

Furthermore, while the paper provides numerous examples, most are tailored to systems or computer science concepts, such as graphs, Python, AWS, security reviewers, and cloud services. Instances of this include "*From now on, whenever I ask a question about a software artifact's security, ...*", "*Whenever I ask you to deploy an application to a specific cloud service, ...*" and "*From now on, whenever I type two identifiers separated by a "→", I am describing a graph. ...*". While this specificity may benefit readers within these domains, it could pose challenges for readers outside these fields to relate or adapt the ideas to their own needs.

Additionally, there's no clear way provided to measure the performance or improvement of queries while using these prompt patterns. Without a clear benchmark, it's tough for readers to assess their effectiveness in practice.

Lastly, regarding readability, during the Introduction and Section II, there's a tendency towards repetitive comparisons between software and prompt patterns, which could have been minimized for a smoother reading experience.

References

OpenAI. Gpt-3.5 turbo. OpenAI API Docs. Https://platform.openai.com/docs/models/gpt-3-5-turbo.