Prompting LLMs to Solve Complex Tasks: A Review

Haochen Li[1] Jonathan Leung[1] Hao Wang[2] Zhiqi Shen[1]
Nanyang Technological University, Singapore
[2]The Hong Kong University of Science and Technology, China

1 Introduction

The recent trend of Large Language Models (LLMs) is evident through the investment of Big technology companies and the widespread discussion and fascination with LLMs in media and online communities. The GPT series by OpenAI, particularly GPT-3 and GPT-4 [OpenAI, 2023], have made headlines for their advanced text generation capabilities. Major corporations like Microsoft have integrated LLMs into their products, enhancing user experiences in applications such as Bing and the Office Suite. We could also see a surge in academics focused on LLMs, underscoring the growing interest in this field [Touvron et al., 2023a, Touvron et al., 2023b]. Additionally, the widespread discussion and fascination with LLMs in media and online communities highlight their impact and the general public's interest in AI advancements.

With such popularity and with LLMs demonstrating their capabilities in a wide variety of downstream tasks, how to leverage LLMs to solve complex tasks becomes an important question. Among all, prompt engineering has been the most direct and effective way to interact with LLMs [Liu et al., 2023b, Qiao et al., 2022]. By crafting precise and clear prompts, users can provide better instructions to LLMs, ensuring more accurate and contextually appropriate answers. This practice not only helps in controlling the tone and style of the LLM's output, making it suitable for varied purposes and audiences but also reduces ambiguities, leading to a more straightforward and efficient interaction. Therefore, prompt engineering serves as a crucial tool in harnessing the full potential of LLMs, ensuring their responses are as beneficial and relevant as possible.

Chain-of-thought prompting (CoT) [Wei et al., 2022] in working with LLMs involves breaking down complex problems into a series of logical steps, similar to how humans think through problems. This method is important as it enhances the ability of LLMs to handle complex multi-step reasoning tasks. For instance, in solving a math problem, the model first identifies the relevant information, and then sequentially applies mathematical operations, clearly articulating each step before reaching the final answer. Similarly, in a reasoning task about cause and effect, the model methodically assesses each aspect of the scenario before concluding. By doing so, CoT not only makes the model's reasoning process more transparent but also significantly improves its accuracy in problem-solving.

Inspired by the simplicity yet powerfulness of the CoT, we would like to dive further into the track of methods that decompose tasks into sub-tasks in prompts to enable LLMs to solve complex tasks. In this paper, we first review existing approaches that also focus on prompting LLMs to solve problems. Then, we pose a possible direction for further improvement. We hope this survey can lead interested researchers into prompt engineering for complex tasks and raise interest in further building on the field.

Section §2 would summarize current papers that decompose complex tasks into sub-tasks in prompt to guide LLMs for solving the problem. Two types of methods and their differences are discussed, namely Iterative decomposition and Plan-then-execute decomposition. Section §3 discusses the disadvantages of the current methods and how hierarchical decomposition can potentially prompt LLMs better in solving complex tasks.

2 Task Decomposition

Decomposing a complex task into simple tasks is particularly useful where one cannot solve it immediately without steps of reasoning in mind. In this section, we introduce the methods to decompose complex tasks and auxiliary techniques to help improve the performance of decomposition.

2.1 Iterative decomposition

Iterative decomposition generates a simple sub-task, performs actions to finish the sub-task, and then repeats this process with the knowledge of the previous results. In [Press et al., 2022], the authors empirically show that even if LLMs know the true answer to all the needed sub-questions for a complex question, LLMs are often wrong when asking them to answer the complex question directly. This finding indicates the significance of decomposing complex tasks into simple sub-tasks for LLMs.

Chain-of-thought prompting [Wei et al., 2022] can be considered the first work to try to decompose a task into sub-task sequences. By showing LLMs a series of intermediate natural language reasoning steps that lead to the final output in the prompt, LLMs can naturally imitate a human-like problem-solving process. Here, the intermediate reasoning steps can be considered as sub-tasks because they are all necessary to answer the question, and they are sequentially connected to form a sub-task sequence that leads to the final solution of the problem. Researchers even found that simply adding "*Let's think step by step.*" to the prompt can guide LLMs to perform chain-of-thought decomposition as well [Kojima et al., 2022]. The above two works implicitly follow iterative decomposition since LLMs generate tokens in an autoregressive way, which could be formulated as:

$$P(x_t|x_1, x_2, ..., x_{t-1}) = \text{softmax}(\text{LLM}(x_1, x_2, ..., x_{t-1})) \tag{1}$$

we could see that it is a conditional probability when generating x_t, which means that they can decide the next sub-task based on previous content.

There are also approaches that explicitly instruct LLMs to adopt an iterative decomposition strategy. DecomP [Khot et al., 2022] and Successive Prompting [Dua et al., 2022] represent two contemporary techniques that employ a repetitive questioning approach to gather background information for tasks involving question answering. Each sub-question answered by the model serves as a sub-task to be accomplished. In contrast to CoT, which may sequentially generate sub-questions within a single output, these two methods explicitly guide LLMs to generate follow-up questions during the process. Empirical findings indicate that explicitly instructing LLMs to decompose complex tasks outperforms relying on implicit decomposition carried out by LLMs themselves.

2.2 Single-step decomposition

Unlike iterative decomposition, single-step decomposition approaches employ just one prompt to break down a task into smaller tasks. For instance, the Least-to-most prompting method, as noted in [Zhou et al., 2022], only requires two prompts for LLMs: one to create a plan breaking down the main task into smaller steps, and another to carry out these steps. The Plan-and-solve prompting technique, as described in [Wang et al., 2023a], enhances the efficiency found in Least-to-most prompting by combining the planning and execution phases into a single response. DEPS [Wang et al., 2023b] and GITM [Zhu et al., 2023] are specialized decomposition strategies for the game Minecraft, a sandbox game where players can create various items and tools. In this game, gathering basic materials is considered a series of sub-tasks necessary to construct the desired item. DEPS formulates a sequential plan for acquiring the needed items, whereas GITM prompts LLMs to break down the task into a sub-task tree structure.

Contrasting with methods like DecomP and Successive prompting, the single-step decomposition approach is more time-efficient as it reduces the number of prompts needed with LLMs. Nonetheless, DecomP and Successive prompting offer greater adaptability, allowing the next sub-task to be tailored based on the outcome of the preceding one, whereas the plan in a one-time decomposition approach remains static. Single-step decomposition is more apt for tasks where sub-tasks are confined to a relatively narrow range. For instance, the complexity and interconnections between tasks and prerequisites are more straightforward in Minecraft than in knowledge-intensive question answering. This simplicity allows for a higher accuracy in the plans generated through one-time decomposition. Ultimately, choosing between efficiency and precision depends on the specific nature of the task at hand.

2.3 External decomposition

The aforementioned categories both depend on the knowledge of LLMs to break down the task into smaller sub-tasks. However, they face challenges with hallucination issues, as pointed out in the literature [Ji et al., 2023]. Occasionally, these approaches generate sub-tasks that appear plausible but lack a firm grounding in reality. To ensure the precision of the decomposition process, LLM+P [Liu et al., 2023a] and SayPlan [Rana et al., 2023] adopt a different approach by integrating classical planning techniques. They employ LLMs to convert tasks expressed in natural language into the domain-specific language used by classical planners. This enables classical planners to work with the tasks more effectively. The results produced by the planners are subsequently translated back into natural language by LLMs.

2.4 Sub-task pre-definition

Opting for potential sub-tasks from a constrained pool offers the advantage of crafting a more precise and efficient sequence of sub-tasks. This approach helps in preventing LLMs from becoming sidetracked by irrelevant or erroneous sub-tasks. PEARL [Sun et al., 2023] is custom-tailored for the task of answering questions within lengthy documents. It employs a set of predefined sub-tasks, such as ”Locating the definition of A,” ”Comparing A and B,” and ”Summarizing A,” from which LLMs can select and organize valuable sub-tasks into a coherent plan. In a similar vein, ProCoT [Deng et al., 2023] establishes predefined sub-tasks encompassing query clarification, topic transition, and negotiation strategies, designed specifically for dialogue systems. DecomP [Khot et al., 2022] adopts a different approach by choosing sub-tasks from a collection of sub-task functions like ”split” and ”merge” for k-th letter concatenation. This method is evaluated across a diverse set of tasks, including tasks involving extensive context, open-domain question answering, and symbolic reasoning. Meanwhile, SayPlan [Rana et al., 2023] is tailored for robot planning tasks. Given a task instruction, it employs semantic search to identify a relevant subgraph within the entire 3D scene graph, serving as the planning environment. LLMs subsequently devise plans solely based on this identified subgraph.

3 Future Direction

Current methods of prompting Large Language Models (LLMs) often involve splitting the final task into sequential subtasks or formulating a plan to execute all steps simultaneously. However, when dealing with complex tasks, this sequential approach can be limiting. The method might fall short in adequately addressing the intricacies of each subtask, especially in situations where each subtask itself is complex and multifaceted. Consequently, the model may struggle with accurately completing the final task due to insufficient breakdown of these complex components.

In contrast, the hierarchical decomposition of tasks, where each subtask is further broken down into smaller, more manageable parts, offers a more robust solution. This method allows for a deeper and more detailed exploration of each aspect of the task, ensuring that every element is thoroughly understood and addressed. For instance, in a complex problem-solving scenario, a subtask might involve several layers of reasoning or calculations, each requiring its own specific approach. Hierarchical decomposition would enable the model to tackle these layers individually, ensuring a more comprehensive and accurate completion of the final task. This approach not only enhances the problem-solving capabilities of LLMs but also mirrors human cognitive processes more closely, leading to solutions that are logical, well-structured, and more reliable.

4 Conclusion

This review has critically analyzed the application of various decomposition methods in prompting LLMs to solve complex tasks. We have seen that iterative, single-step, external, and predefined sub-task decompositions each offer unique benefits and limitations. Our analysis suggests that while current methods enhance LLMs' problem-solving abilities, there is significant room for improvement. Future research should focus on developing more advanced hierarchical decomposition strategies, which could better mimic human cognitive processes and offer more nuanced and reliable solutions.

References

[Deng et al., 2023] Deng, Y., Lei, W., Liao, L., and Chua, T.-S. (2023). Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626*.

[Dua et al., 2022] Dua, D., Gupta, S., Singh, S., and Gardner, M. (2022). Successive prompting for decomposing complex questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1251–1265.

[Ji et al., 2023] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

[Khot et al., 2022] Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. (2022). Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

[Kojima et al., 2022] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

[Liu et al., 2023a] Liu, B., Jiang, Y., Zhang, X., Liu, Q., Zhang, S., Biswas, J., and Stone, P. (2023a). Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477.*

[Liu et al., 2023b] Liu, X., Wang, J., Sun, J., Yuan, X., Dong, G., Di, P., Wang, W., and Wang, D. (2023b). Prompting frameworks for large language models: A survey. *arXiv preprint arXiv:2311.12785.*

[OpenAI, 2023] OpenAI (2023). Gpt-4 technical report. *ArXiv*, abs/2303.08774.

[Press et al., 2022] Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., and Lewis, M. (2022). Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350.*

[Qiao et al., 2022] Qiao, S., Ou, Y., Zhang, N., Chen, X., Yao, Y., Deng, S., Tan, C., Huang, F., and Chen, H. (2022). Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597.*

[Rana et al., 2023] Rana, K., Haviland, J., Garg, S., Abou-Chakra, J., Reid, I., and Suenderhauf, N. (2023). Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *arXiv preprint arXiv:2307.06135.*

[Sun et al., 2023] Sun, S., Liu, Y., Wang, S., Zhu, C., and Iyyer, M. (2023). Pearl: Prompting large language models to plan and execute actions over long documents. *arXiv preprint arXiv:2305.14564.*

[Touvron et al., 2023a] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023a). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971.*

[Touvron et al., 2023b] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288.*

[Wang et al., 2023a] Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P. (2023a). Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091.*

[Wang et al., 2023b] Wang, Z., Cai, S., Liu, A., Ma, X., and Liang, Y. (2023b). Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560.*

[Wei et al., 2022] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

[Zhou et al., 2022] Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q. V., et al. (2022). Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

[Zhu et al., 2023] Zhu, X., Chen, Y., Tian, H., Tao, C., Su, W., Yang, C., Huang, G., Li, B., Lu, L., Wang, X., et al. (2023). Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.