Dean&Francis

ISSN 2959-6157

Research and Analysis of Prompt Engineering Technology Systems Based on Large Language Model Generation Tasks

Ruoxi Zhang[1, *]

[1]Department of Computer Science and Technology, Southwest University of Science and Technology University, Mianyang, China

*Corresponding author: zrx@mails.swust.edu.cn

Abstract:

Due to the rapid development of artificial intelligence, prompt engineering technology has become a key technology for the development of Large Language Models in the field of generation tasks. Although the existing reviews have listed various technologies in detail, they still lack an analysis of the inherent logic and evolution laws among the technologies. Therefore, this article classifies the core issues that the technology solves, namely the functional goals, establishes the connection between the technology and the core issues, and presents the development path of the technology. First, let's introduce the basic prompt class to achieve something from nothing and solve the problem of task understanding. Then elaborate on how structured reasoning classes deal with logical reasoning problems. In this category, it is further divided into two subcategories: linear reasoning and nonlinear reasoning. Then, it discusses how the hallucination-correcting category overcomes the problems of knowledge deficiency and hallucinations. Finally, it explores how the automation optimization category can focus on the efficiency of manual design and promote the evolution of technology towards scale and industrialization. In the future, engineering technology will move towards a more intelligent direction, achieving a paradigm shift in human-computer interaction.

Keywords: Prompt engineering; Large language model; Chain of thought; Correcting illusion; Automated optimization.

1. Introduction

Traditional artificial intelligence models are only targeted at specific tasks, resulting in rigid input-output forms. However, Large Language Models and generative artificial intelligence are currently undergoing a paradigm shift. In the face of the huge gap between the explosive growth of large model capabilities and

application demands, it is suggested that engineering technology is precisely the key to unlocking these capabilities. For the vast majority of developers and enterprises, compared with the high cost and low efficiency of fine-tuning LLMs, by carefully constructing prompts, there is no need to update parameters, lowering the usage threshold and unlocking the huge potential of the model. At the same time, it gives rise to new application ecosystems such as AI agents, creating new paradigms.

Sahoo et al. classified the application scenarios of existing prompt engineering technologies, demonstrated the advantages and limitations of various methods, and provided users with a framework for selecting prompt words for specific tasks [1]. Chen et al. explored some fundamental and advanced prompt methods, providing more examples to help researchers or users better understand various technologies, and evaluated and analyzed the role of prompt engineering in AI security through subjective and objective indicators to unlock the potential of LLMs [2]. Moreover, people from different fields are also exploring the potential that prompt engineering can unleash in their professional fields. Oppenlaender et al. planned and discussed four future research directions of prompt engineering in creative skills [3], and Cain et al. explored the potential of this technology in reforming education [4].

Although a large number of existing reviews have detailed listed various technologies and classified them through application scenarios, etc., there is still a lack of systematic analysis of the internal logic and evolution laws among the technologies. Therefore, this article classifies different prompt engineering technologies based on the core problems they solve. The currently developing prompt engineering technologies are mainly divided into four categories. While explaining each technology, it sorts out the intrinsic relationships among them and directly connects them with the needs of users. Basic prompt types solve the problem of task understanding, that is, how to make the model understand what to do. Structured reasoning addresses the issue of "logical reasoning", figuring out how to make the model think more clearly and deeply. Knowledge enhancement technology addresses the issue of "knowledge deficiency and illusion", and how to make models answer more accurately and reliably. Automation optimization technology addresses the issue of "manual design efficiency", figuring out how to automatically find the best prompts and liberate human resources. Basic technology is the starting point, solving the problem of going from nothing to something. Structuring and knowledge technology are developments aimed at addressing the bottlenecks in generation quality (logical and factual). Automation technology is at the forefront. To scale up and industrialize the prompt engineering itself.

The structure of this article is arranged as follows: Chapter Two classifies through functional objectives and introduces representative technologies in different classifications. Chapter Three analyzes typical cases of structured reasoning, illusion correction, and automated optimization. This review aims to classify prompt engineering technologies, reveal the intrinsic logic among them, provide a clear framework for researchers in prompt engineering technologies, and promote future research and development in this field. And they are classified based on functional goals, which makes it easier for users to quickly understand and select appropriate prompt technologies, thereby enhancing practicality.

2. Prompt engineering technology

2.1 Basic prompt class

2.1.1 Zero-Shot Prompting

Radford et al. proposed that in the absence of prompt examples, that is, labeled data without input-output mappings, large language models utilize previously learned knowledge to predict results based on carefully constructed prompts [5]. The output may not meet the requirements of the task description, and it is unstable and the format is not uniform.

2.1.2 Few-Shot Prompting

Brown et al. enabled large language models to recognize patterns, formats, etc. from the provided examples through context learning by providing a small number of input-output examples (distinct from the parameters of training and updating the model), and then input the prediction results [6]. Since providing examples consumes the quota of tags, it may not be achievable for long text input. The combination order of examples will also affect the performance of the model.

2.1.3 Role-Play Prompting

In today's society, the demand for answering questions in the fields of marketing copywriting and professional skills has significantly increased. Role prompts guide LLMs to generate more professional, targeted, and context-appropriate content by endowing them with specific roles or identities. Zero-shot prompts and role prompts were tested and compared on 12 different datasets on ChatGPT-3.5. The performance of role prompts was superior to that of zero-shot prompts on 10 of the datasets. It demonstrates the potential of character hints in reasoning about task understanding.

ISSN 2959-6157

2.2 Structural reasoning class

2.2.1 Linear reasoning class

As traditional prompt techniques perform poorly in solving complex problems, Wei et al. proposed Chain-of-Thought (COT) [7]. Step-by-step reasoning is achieved by adding specific statements (such as "Let's think step by step"), guiding LLMS to imitate the way humans think when solving problems and decompose a complex problem into multiple sub-problems that can be solved step by step. This process is continuous, and each step is gradually reasoned and solved based on the result of the previous step. However, COT generates a single, linear inference chain. If any step goes wrong, the entire inference will fail, and LLM has no mechanism to detect and correct this error. In fact, when humans solve problems, they will consider multiple approaches, determine which one is more appropriate, and when a certain path fails, they will go back to the previous decision point to reconsider.

2.2.2 Nonlinear reasoning class

Aiming at the practical problems of the COT linear reasoning chain, the Tree-of-Thoughts (TOT) framework proposed by Yao et al. is generalized to enhance the performance of LLMS in exploration [8], prospect and decision-making when dealing with complex problems. Elevate LLM from a simple "content generator" to an "active problem solver" capable of managing its own thought process. Form a tree structure with the intermediate reasoning steps. The root node represents the initial problem, the intermediate nodes (also known as thinking) represent the possible states or intermediate steps for solving the problem, and finally the leaf nodes display the final answer. The TOT framework proposes multiple possible next "thoughts" through a generator, assesses these thoughts with an evaluator, and combines them with search algorithms such as breadth-first or depth-first, enabling LLMS to explore, anticipate in advance, and trace back when encountering incorrect solutions in the reasoning chain.

Humans do not merely reason and think from an independent chain of thought, but may have various different ways of thinking such as jumping, integrating, and cycling. TOT is difficult to support the process of cross-integration among such thinking paths. So, Yao et al. proposed the Graph-of-Thoughts (GOT) framework [9], which is more in line with the thinking neural network of the human brain. Replace the tree structure with the data structure of the graph. Nodes represent thinking units, and edges represent logical associations. It not only possesses the generation, evaluation and backtracking functions of the TOT framework, but also can aggregate different branches of thinking between each node step, thereby solving the problem that different branches of thinking in TOT are independent and unable to interact with each other. It has enhanced the performance of LLMS in handling complex tasks, making them more in line with the complexity of human thinking processes.

CoT, PoT, ToT, etc. represent reasoning steps through text or code. However, these methods are insufficient when dealing with specific fields such as complex tables. Wang et al. proposed the Chain-of-Table Prompting technology [10], materializing the data input and reasoning process as a series of gradually evolving tables themselves, and operating on the tables to achieve a step-by-step reasoning process. This form makes the logical process of reasoning clear and visible, making each step of reasoning transparent and explainable.

2.3 Correcting illusion class

2.3.1 Retrieval Augmented Generation (RAG)

The fundamental flaw of LLMS: The limited and static training data restricts their performance, causing them to be unable to answer the latest questions when confronted with external knowledge and to experience hallucinations. However, traditional methods such as "fine-tuning" have disadvantages such as high cost and complexity. Therefore, RAG was proposed, integrating information retrieval into the prompt process to solve the above problems.

The first step of this technology is retrieval. That is, the user inputs a question, and RAG will understand and analyze the user's question, optimize it into a search query that is more suitable for retrieval, and then search for relevant resources from the constructed knowledge base. Next comes "enhancement", which combines the retrieved resources with prompt instances and enriches them with context, thereby enhancing the prompt words of the LLM. Finally, "generate" and output the predicted result. The RAG technology overcomes the problems of static and illusion, enhancing accuracy, ensuring timeliness and reducing update costs.

2.3.2 Chain-of-Verification (CoVe)

To address the illusion problem of LLMS, Dhuliawala et al. proposed the validation chain technique [11], which consists of four main steps: generating initial responses, generating validation questions, independently answering validation questions without referring to the initial responses, separating "generation" from "validation", correcting the model's responses, checking the correctness of the answers, and then conducting validation. Compare it with the initial answer, and finally correct the initial answer to output a more accurate one. By simulating the human verification thinking mode through multi-step ver-

ification, the logical reasoning ability of LLMS has been enhanced, errors have been reduced, and accuracy has been improved.

2.4 Automatic optimization technology class

2.4.1 Automatic Prompt Engineering (APE)

Designing high-quality prompt words is an extremely arduous task. Then, how to solve manpower problems and improve efficiency has become a hot research topic. Zhou et al. proposed APE [12]. An innovative method of using machines to automatically generate and optimize prompts. APE overcomes the limitations of manually designing prompt words. First, the large language model analyzes the user input to generate candidate instructions. Then, another model evaluates these candidate instructions to guide the LLM to generate or select higher-quality prompt words. This makes the LLM's task processing more efficient and broadens the application field.

2.4.2 Optimization by Prompting (OPRO)

Only change the prompt words without altering the model parameters. Yang et al. used LLM as an optimizer to achieve optimization through natural language description and historical iteration [13]. Its workflow is actually an iterative cycle: provide the LLM with a prompt, including a problem description and previously attempted solutions along with their scores. Based on the above historical information, the LLM generates a better solution, then evaluates this solution, and re-inserts this solution, its score, and historical records into the prompt. This cycle of iteration continues, gradually optimizing. Since all problem descriptions are made using natural language, which is different from the design of traditional optimization algorithms and requires specific modifications, it can be applied to multiple fields and has high flexibility.

3. Case Analysis

3.1 The performance of COT when dealing with complex reasoning

CoT technology is currently the most commonly used method to unlock the reasoning potential of LLMS. Sprague et al. explored in which tasks or application scenarios CoT has actual improvements, not just in mathematical tasks, to fill this gap [14].

This case combines COT technology with basic hint technology, namely Zero-Shot COT and Few-Shot COT. Zero-Shot was evaluated based on 20 different types of datasets in 14 models. Through experimental analysis, it is found that CoT's reasoning in mathematical and symbolic problems has significantly improved, and its gains mainly come from symbolic execution. For instance, on the MMLU dataset, CoT's accuracy has only significantly improved when the problem or model response includes "=". Among them, based on GPT-4, the Mathematical dataset increased from 36.5% to 63.3%, and on Symbolic it increased from 55.7% to 68.3%. However, in terms of common sense responses and other questions, there is almost no difference in performance compared to the directly generated ones. This reflects a significant improvement in the performance of structured reasoning classes in logical reasoning problems.

Meanwhile, this case combines different technologies, compensating for the advantages and disadvantages among them. In the future, this technology can develop in the direction of a hybrid element prompt architecture. Enhance the application of this technology in a wider range of generation task fields.

3.2 The implementation of ReAct Paradigm in question-answering systems

Because the traditional prompting methods only rely on the internal knowledge of the model, they cannot interact with external information sources to obtain the latest information or perform specific actions. Yao et al. propose the ReAct paradigm [15], which is not a new model but an innovative interaction framework based on prompt engineering that overcomes the pervasive illusion and error propagation problems in chain-thinking reasoning and generates human-like task-solving trajectories that are more interpretable than baselines without reasoning trajectories.

The core of the ReAct paradigm lies in its structured loop hinting framework. Given a problem, ReAct performs Thought - reasoning about the current situation to decide what to do next, Action - deciding what action to take, Observation - interacting with an external source to obtain precise information, and reasoning again based on that information to correct errors. Then repeat the above operation, and finally get a more accurate and real-time answer. And the test on HotPotQA and FEVER datasets also significantly outperforms CoT only and Act only. In addition, it outperforms imitation learning and reinforcement learning methods by 34% and 10%, respectively, based on only a small number of examples prompts. This reflects that the rectifying illusion class overcomes the rectifying illusion problem based on the inference chain and improves the accuracy.

At the same time, it is still a great challenge to balance the retrieval quality with speed and efficiency in the face of huge amount of external information sources.

ISSN 2959-6157

3.3 Promptomatix: An Automated Prompt Optimization Framework for Large Language Models

In order to solve the problem that manual design of prompt words is time-consuming and labor-intensive, and it highly depends on professionalism while ordinary users do not have relevant technical background, Murthy et al. proposed the Promptomatix framework [16], which is a significant change from „craftsmanship“ to „industrialization“ of prompt engineering.

The core architecture of this framework is like a precision factory, and its four core components are like four key workshops. After the user describes the problem in natural language, the workshop is configured for task analysis, and then the optimization engine workshop generates or optimizes data and prompt words, and then outputs the workshop for delivery, and feedback the workshop for continuous improvement. Technologies such as fully automated end-to-end processes, intelligent classification to accurately identify the nature of tasks, and design mathematical formulas to balance quality and efficiency. The framework is tested on multiple data sets, such as GSM8K, XSum, News, CommonGE, etc., and its performance on text classification tasks is significantly improved by nearly 15%-20%, and its performance in other test fields is also excellent. The „democratization“ of improving engineering technology has been realized, which marks that prompt engineering has moved from the „craftsmanship“ era relying on personal experience to the „industrialization“ era based on algorithms and systems.

Table 1. The current development level and main challenges of four types of prompt technologies

Technology Category	Current development level	Main Challenges
Basic prompt	Mature and stable	Weak in complex tasks, long text processing
Structural reasoning	Frontier exploration	High computational overhead, dependence on large model size, and high requirements for the design of search strategies
Correcting illusion	Wide application	Retrieval quality and speed, management and update of external knowledge sources, multi-source information fusion
Automatic optimization technology	Emerging frontier	Search space is large, optimization of the high cost, prompt interpretability and controllable performance degradation

In the future, this technique may be integrated with multimodal models to achieve automatic optimization in vertical domains.

In order to summarize the various types of technologies discussed in the previous sections more clearly and systematically, and analyze some typical cases, Table 1 summarizes the current development level and the main challenges of different types of technologies. At the same time, it is these challenges that constitute the main breakthrough for future research.

4. Conclusion

In this paper, the functional goal of prompt engineering technology is classified, that is, the core problem solved by the technology, and a classification system that clearly reveals the internal logic and development law between technologies is constructed, which is more practical. The existing technologies are summarized into basic prompt class, structured reasoning class, hallucinatory correction class and automatic optimization class. This review reveals that prompt engineering technology presents an evolution trend from traditional manual design and static to automatic optimization and generation and dynamic. In the early stage, cue examples were carefully constructed by humans, while the current focus is more on the improvement of their reasoning performance and how to automatically optimize prompts while balancing efficiency and quality. In the future, the research on prompting engineering is developing towards meta-learning, multimodal prompting, and hybrid meta-prompting architectures. However, this technology still faces great challenges in ethics and safety, as well as the establishment of standardized benchmarks. Through the systematic classification and review of the existing technologies, this paper hopes to provide readers with a structured knowledge navigation for understanding this field, encourage more researchers to devote themselves to the research directions with both challenges and opportunities, and jointly promote the future development of prompt engineering and even artificial intelligence technology.

References

[1] Sahoo P, Singh AK, et al. A systematic survey of prompt engineering in large language models: Techniques and applications. arxiv preprint arxiv:2402.07927. 2024 Feb 5.

[2] Chen B, Zhang Z, et al. Unleashing the potential of prompt engineering for large language models. Patterns. 2025 May 8.

[3] Oppenlaender J, Linder R, et al. Prompting AI art: An investigation into the creative skill of prompt engineering. International journal of human–computer interaction. 2025 Aug 18;41(16):10207-29.

[4] Cain W. Prompting change: Exploring prompt engineering in large language model AI and its potential to transform education. TechTrends 68.1 2024: 47-57.

[5] Radford A, Wu J, et al. Language models are unsupervised multitask learners. OpenAI blog. 2019 Feb 24;1(8):9.

[6] Brown T, Mann B, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020;33:1877-901.

[7] Wei J, Wang X, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems. 2022 Dec 6;35:24824-37.

[8] Yao, Shunyu, et al. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 36 2023: 11809-11822.

[9] Yao S, Yu D, et al. Tree of thoughts: Deliberate problem solving with large language models. Advances in neural information processing systems. 2023 Dec 15;36:11809-22.

[10] Wang Z, Zhang H, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. arxiv preprint arxiv:2401.04398. 2024 Jan 9.

[11] Dhuliawala S, Komeili M, et al. Chain-of-verification reduces hallucination in large language models. arxiv preprint arxiv:2309.11495. 2023 Sep 20.

[12] Zhou Y, Muresanu AI, et al. Large language models are human-level prompt engineers. InThe eleventh international conference on learning representations 2022 Nov 3.

[13] Yang C, Wang X, et al. Large language models as optimizers. InThe Twelfth International Conference on Learning Representations 2023 Sep 7.

[14] Sprague Z, Yin F, et al. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. arxiv preprint arxiv:2409.12183. 2024 Sep 18.

[15] Yao S, Zhao J, et al. React: Synergizing reasoning and acting in language models. InInternational Conference on Learning Representations (ICLR) 2023 Jan.

[16] Murthy R, Zhu M, et al. Promptomatix: An Automatic Prompt Optimization Framework for Large Language Models. arxiv preprint arxiv:2507.14241. 2025 Jul 17.