Prompt Engineering in Large Language Models

Chapter · January 2024
DOI: 10.1007/978-981-99-7962-2_30

4 authors, including:

Ggaliwango Marvin
Makerere University

Nakayiza Hellen Raudha
Makerere University

Joyce Nakatumba-Nabende
Makerere University

Prompt Engineering in Large Language Models

Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende

1 Introduction

1.1 Overview of Large Language Models (LLMs)

A Large Language Model (LLM) is an AI algorithm that employs deep learning and vast datasets to comprehend, summarize, generate, and predict new content. LLMs are trained on extensive amounts of text using self-supervised learning and excel at a wide range of tasks [1]. LLMs are constructed using neural networks and multiple parameters usually billions of weights and more. These models are pre-trained on vast amounts of data to help them understand the complexity and relationships within language. By using techniques such as fine-tuning, in-context learning, and zero-/one-/few-shot learning, these models can be tailored for specific tasks [2]. A Large Language Model (LLM) is essentially a transformer-based neural network that predicts the text that is likely to come next, therefore its sophistication and performance can be evaluated by the number of parameters it possesses. The model's parameters are the factors it takes into account when generating output [3]. LLMs have generated much excitement recently due to their impressive capabilities as general-purpose computers when conditioned on natural language instructions. However, the effectiveness of the model is heavily influenced by the quality of the prompt used to guide it, and most successful prompts have been created manually by humans [4]. This has led to the rise of prompt engineering as an important skill set for NLP and AI engineers, conversational AI researchers, and most importantly,

G. Marvin (✉) · D. Jjingo · J. Nakatumba-Nabende
Makerere University, Kampala, Uganda
e-mail: ggaliwango.marvin@mak.ac.ug

N. Hellen
Muni University, Muni, Uganda

I. J. Jacob et al. (eds.), *Data Intelligence and Cognitive Informatics*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-99-7962-2_30

normal information seekers in various domains like education and health care were improved efficiency in the use of LLMs which is more valued.

1.2 Importance of Prompt Engineering for LLMs

Prompt engineering is the process of designing and refining input queries, or "prompts," to elicit desired responses from Large Language Models (LLMs). Prompts are crucial in guiding LLMs to generate useful and relevant outputs [5]. By understanding how to create effective prompts, information seekers and developers can improve LLM performance, explore new applications, and save valuable time and resources. This can be achieved by leveraging good quality prompts to guide the model toward generating more accurate and useful outputs. This in turn allows information seekers and developers to unlock the full potential of LLMs. Prompt engineering permits exploration of new applications for LLMs, saves valuable time and resources, and provides a form of programming that can customize the outputs and interactions with an LLM [6]. It offers reusable solutions to common problems faced in output generation and interaction when working with LLMs. By facilitating the development of more accurate, context-specific, and nuanced responses from LLMs, prompt engineering supports the advancement of research and discoveries in the field of AI most especially conversational AI.

Prompt engineering provides a framework for documenting patterns for structuring prompts to solve a range of problems and allows information seekers and developers to adapt prompts to different domains. It enables the combination of multiple prompt patterns to improve LLM outputs and facilitates the transfer of knowledge between information seekers and developers working with LLMs [7]. Prompt engineering supports the development of more sophisticated and effective LLM applications, helps developers to better understand the behavior and capabilities of LLMs, and enables information seekers and developers to steer LLMs toward truthfulness and/or informativeness. It also improves few-shot learning performance by prepending optimized prompts to standard in-context learning prompts and facilitates the development of more effective chatbots, virtual assistants, domain-specific prompt engineering tools, and other conversational AI systems [8, 9]. Therefore, prompt engineering supports the advancement of natural language processing (NLP) tasks by improving LLM performance.

Prompt engineering is likely to play an increasingly important role in unlocking the full potential of Large Language Models (LLMs) in the near future. If information seekers and developers and empowered to generate specific language output quickly and accurately, prompt engineering will become marketable element for increasing efficiency and streamlining business operations across various domains. This rapidly growing field may also lead to new job opportunities for those skilled in prompt engineering. As information seekers and developers continue to experiment with more sophisticated prompts [10, 11], we are likely to see the development of more efficient and understandable user interfaces for controlling LLM outputs. This enhanced control over LLM outputs will enable developers to fine-tune the generated content and unlock new applications for LLMs that were previously impossible.

1.3 Research Objective and Motivation

The problem that this chapter addresses is the need for effective prompt engineering to unlock the full potential of Large Language Models (LLMs). LLMs have generated much hype in the recent months due to their impressive capabilities as general-purpose computers when conditioned on natural language instructions [12]. However, the effectiveness of the model is heavily influenced by the quality of the prompt used to guide it, and most successful prompts have been created manually by humans. This has led to the rise of prompt engineering as an important skill set for information seekers, AI engineers and researchers to improve and efficiently use LLMs. Therefore, this study provides understanding of prompt engineering, presents an overview of the latest prompting techniques, and provides demonstrations and exercises to practice different prompting techniques. It also discusses the current and future trends of research on LLMs and prompt engineering, including the rise of automatic instruction generation and selection methods for prompt engineering. By understanding how to create effective prompts, information seekers and developers can improve LLM performance, explore new applications, and save valuable time and resources.

2 Prompt Engineering

What are Prompts?

A prompt is a text-based input that is fed to a language model to guide its output. A prompt can be audio but, in this case, the audio input would be transcribed into text and fed to the language model as a text-based prompt. The language model would then process the text-based prompt and generate an output based on the instructions and context provided in the prompt [13]. The primary purpose of a prompt is to provide the language model with instructions and context for achieving a desired task. Prompt engineering is the practice of developing and optimizing prompts to efficiently use language models (LMs) for a variety of applications. A prompt is composed of an instruction and can have input data, context, and some output indicator (Table 1).

Table 1 Examples of prompts for large language models (LLMs)

Prompt	Instruction	Input data	Context	Output indicator
Paint a tall African man dancing to cool Ugandan folk songs	Paint	NA	Tall African man dancing to cool Ugandan folk songs	NA
"Write a science fiction short story set in a future where humans particularly babies have colonized Mars, exploring the ethical implications of terraforming and the potential conflicts between Earth and Martian societies." (Creative writing)	Write a science fiction short story	Babies	Set in a future where humans have colonized Mars	Explore the ethical implications of terraforming and potential conflicts between Earth and Martian societies
"Write a short paragraph explaining the Pythagorean theorem and providing an example to demonstrate its use." (Mathematics)	Write a short paragraph	NA	Explaining a mathematical concept	Provide an example
"Write a philosophical essay exploring the concept of free will, incorporating arguments from determinism, compatibilism, and libertarianism." (Philosophy)	Write a philosophical essay	NA	Explore the concept of free will	Incorporate arguments from determinism, compatibilism, and libertarianism
"Generate a technical report on the feasibility of using nuclear fusion as a sustainable energy source, including an analysis of current technological limitations and potential solutions." (Engineering)	Generate a technical report	NA	Feasibility of using nuclear fusion as a sustainable energy source	Include an analysis of current technological limitations and potential solutions
"Write a historical fiction novel set during the French Revolution, incorporating accurate historical details and exploring themes such as social inequality, political upheaval, and personal agency." (Creative writing)	Write a historical fiction novel	NA	Set during the French revolution	Incorporate accurate historical details and explore themes such as social inequality, political upheaval, and personal agency

(continued)

Table 1 (continued)

Prompt	Instruction	Input data	Context	Output indicator
"Generate a comprehensive review of the current state of research on artificial intelligence and its potential impact on society, including an analysis of ethical considerations and potential risks." (Academia)	Generate a comprehensive review	NA	Current state of research on artificial intelligence and its potential impact on society	Include an analysis of ethical considerations and potential risks
"Write a legal brief arguing for the protection of privacy rights in the digital age, incorporating relevant case law and constitutional principles." (Law)	Write a legal brief	NA	Argue for the protection of privacy rights in the digital age	Incorporate relevant case law and constitutional principles
"Generate a detailed analysis of the environmental impact of large-scale deforestation, including effects on biodiversity, carbon sequestration, and soil erosion." (Environmental science)	Generate a detailed analysis	NA	Environmental impact of large-scale deforestation	Include effects on biodiversity, carbon sequestration, and soil erosion
"Write a screenplay for a romantic comedy set in New York City, incorporating elements of social commentary and exploring themes such as identity and self-discovery." (Entertainment)	Write a screenplay for a romantic comedy	NA	Set in New York City	Incorporate elements of social commentary and explore themes such as identity and self-discovery
"Generate a comprehensive treatment plan for someone with chronic pain, incorporating both traditional and alternative therapies and taking into account factors such as comorbid conditions and medication interactions." (Medicine)	Generate a comprehensive treatment plan	NA	For someone with chronic pain	Incorporating both traditional and alternative therapies and taking into account factors such as comorbid conditions and medication interactions

2.1 The Process of Prompt Engineering

Steps involved in creating effective prompts:

1. Defining the goal: The first step in creating an effective prompt for a language model is to clearly define the goal of the prompt. What do you want the model to generate as a response to the prompt? Having a clear goal in mind enables you to focus your efforts and create a more effective prompt [14]. This is important because it provides direction and purpose for the rest of the prompt engineering process.
2. Understanding the model's capabilities: For every language model selected for prompting, it is very important to understand its limitations and abilities. An example is clarity on what kind of responses the model generates (text, audio, images, etc.) The strengths and weaknesses of a given model guide the synthesis of appropriate prompts that are effectively compatible with the model's capabilities and bypassing such limitations [15]. Fortunately, LLMs today have been built to optimize their performance by making use of some online and openly available tools via relevant APIs, e.g., access to diagraming tools.
3. Choosing the right prompt format: There is a huge impact on the quality of LLM generated responses based on prompt formats. Clear, precise, and concise prompting formats empower the model with all necessary details for generating coherent responses [16]. Selection of appropriate prompt formats enhance natural language understanding for LLMs hence improving quality of response given.
4. Providing Context: This is one of the most under estimated stages yet most influential in terms of information accuracy generated by the models. Without context LLMs provide generic coherent and relevant responses [17]. But with limited contexts, LLMs provide misinformation for less represented context within existing data. Context may include additional relevant/related information about the topic, setting or characters for inference within the prompt. Context provides deeper metrics for the LLM to understand the prompt and desired output.
5. Testing and Refining: For every synthesized prompt, it is vital to test it with appropriate LLMs based on the defined goal [18]. The generated LLM response can then be used to evaluate and refine the prompt before its output in a production setting. The whole aim of this testing is to obtain feedback for objective refining before making any data-driven decision about prompt optimization/improvement.

While the above procedure gives a structured approach to creating optimal and efficient prompts for LLMs like ChatGPT, information seekers and developers may choose to deviate from some of the above steps. This may usually happen if one has the expertise and extensive experience with a particular prompt engineering tool or LLM. In this case, the information seeker or developer can majorly rely on their creativity and intuition to synthesize prompts. The divergence may also happen cases of LLM testing and highly experimental circumstances. In cases of design and innovative thinking, validation and verification, information seekers may take exploratory approaches to prompt engineering. Please note that the synthesized procedure /framework above is not a set of rules and restrictions and not mandatory to be followed religiously. However, it is a guide that can help prompt engineers create effective prompts. Ultimately, the most effective approach to prompt engineering will depend on the specific goals, audience, and language model being used. A skilled prompt engineer will use their judgment and expertise to determine the best approach for each situation.

3 Prompt Engineering Techniques

3.1 Techniques for Optimizing Prompts

Optimizing prompts for language models like ChatGPT involves a combination of creativity, intuition, and data-driven approaches. By using the techniques described below, you can create effective and engaging prompts that help users achieve their goals;

Clearly define the goal of the prompt, understand the capabilities and limitations of the language model, choose a format that is clear and concise, provide context to help the model generate more relevant responses, test and refine your prompts to improve their effectiveness, use engaging language and visuals to capture the user's attention, Tailor your prompts to the interests and motivations of your audience, experiment with different formats and styles to see what works best, gather feedback from users to improve your prompts, use data-driven approaches to refine and optimize your prompts, keep your prompts focused and on-topic, avoid using overly complex or technical language, use examples and analogies to help explain complex concepts, break down complex topics into smaller, more manageable chunks, use humor and creativity to make your prompts more engaging, leverage the power of storytelling to make your prompts more memorable, use repetition and reinforcement to help users retain information, provide clear and concise instructions for the user, use visual aids such as images and diagrams to support your prompts, provide additional resources for users who want to learn more, use a conversational tone to make your prompts more approachable, encourage user interaction and engagement with your prompts, use personalization to tailor your prompts to individual users, continually update and refresh your prompts to keep them relevant, and most importantly, monitor the performance of your prompts and make adjustments as needed.

It is important to note that optimizing prompts for language models like ChatGPT is an ongoing process. As language models evolve and improve, and as user needs and expectations change, it is important to continually monitor and update your prompts to ensure that they remain effective and relevant most especially using data-driven techniques. This can involve gathering data on how users interact with your prompts, analyzing the responses generated by the language model, and using this information to make data-driven decisions about how to improve your prompts. By continually testing and refining your prompts, you can optimize their performance and achieve better results.

3.2 Advanced Techniques for Prompt Engineering

Prompt engineering can be used to create effective prompts for a wide range of natural language processing tasks, including Text Summarization, Question Answering, Text Classification, Role Playing, Code Generation, Reasoning, Text Generation, Text Translation, Sentiment Analysis, Named Entity Recognition, Text Completion, Dialog Generation, Paraphrasing, Text Simplification, Text-to-Speech, Speech-to-Text, Image Captioning, Text-based Game Playing, Poetry Generation, Lyric Generation, Story Generation, Joke Generation, Recipe Generation, Email Generation, Resume Generation, News Article Generation, Text-based Adventure Game Playing, Text-based Puzzle Solving, Text-based Strategy Game Playing, Text-based Simulation Game Playing, Text-based Interactive Fiction, Text-based Virtual Assistant, Text-based Customer Service, Text-based Personal Shopping Assistant, Text-based Personal Finance Assistant, and Text-based Personal Health Assistant among others [1–30]. Accomplishing such tasks often requires basic and advanced prompt engineering techniques. These kinds of techniques are explained in Table 2.

The prompt engineering techniques can be used to improve the performance of language models like ChatGPT, allowing them to generate more coherent, relevant, and sophisticated responses to user inputs. Besides those, there are some much more advanced prompts engineering techniques with include;

Automatic Instruction Generation: It refers to the use of technology to automatically generate instructions or prompts. This can be done in various contexts, such as creating how-to guides or generating prompts for machine learning models. One example of an advanced technique for automatic instruction generation tool is the Automatic Prompt Engineer (APE), which was proposed for generating and selecting instructions for Large Language Models. APE considers the instruction as a "program" and optimizes it by searching through a pool of instruction candidates suggested by a Large Language Model to maximize a chosen score function [37]. The effectiveness of the chosen instruction is then assessed by measuring the zero-shot performance of another Large Language Model that follows the selected instruction. Experiments on 24 NLP tasks showed that instructions automatically generated using APE outperformed previous benchmarks and achieved better or similar performance to instructions created by human annotators on 19 out of 24 tasks [38]. This demonstrates the potential of automatic instruction generation as an advanced prompt engineering technique.

Program Synthesis for Prompt Engineering: Classical program synthesis and human approach prompt engineering is what inspires this technique. It entirely relies on automatically generating instructions or prompts by using techniques from program synthesis [38]. APE is still a good example for selected LLMs. Here, it searches and optimized instruction as a program within a pool of LLMs while

Table 2 Prompt engineering techniques

Technique	Description	Most recommended large language models (LLMs)
Few-shot prompts [19]	It involves using a small number of carefully crafted prompts to enable a language model to perform a new task or adapt to a new domain [19]	GPT-3, GPT-4
Chain-of-thought (CoT) prompting [20]	Using a sequence of interconnected prompts to guide the language model's responses in a coherent and logical manner [20]	GPT-3, GPT-4
Self-consistency [21]	Using prompts to encourage the language model to generate responses that are consistent with its previous responses [21]	GPT-3, GPT-4
Knowledge generation prompting [22]	Using prompts to encourage the language model to generate new knowledge or insights based on its existing knowledge and understanding of the world [23]	GPT-3, GPT-4
Reasoning and acting (ReAc) [24]	Using prompts to encourage the language model to reason about a given situation and generate appropriate actions or responses [24]	GPT-3, GPT-4
Contextual prompting [25]	Providing additional context to the language model to help it generate more coherent and relevant responses [25]	Transformer-based models such as GPT-3, GPT-4, and BERT
Dynamic prompting [26]	Dynamically adjusting the prompt based on the language model's previous responses to improve its performance over time [27]	Transformer-based models such as GPT-3, GPT-4, and BERT
Multi-modal prompting [28]	Using multiple modalities, such as text and images, to provide richer and more detailed prompts to the language model [29]	Multi-modal models such as CLIP and DALL-E
Adversarial prompting [30]	Using adversarial examples to test and improve the robustness of the language model's responses to prompts [30]	Transformer-based models such as GPT-3, GPT-4, and BERT
Transfer learning prompting [31]	Using transfer learning to adapt a pre-trained language model to new tasks or domains using carefully crafted prompts [31]	Transformer-based models such as GPT-3, GPT-4, and BERT

(continued)

Table 2 (continued)

Technique	Description	Most recommended large language models (LLMs)
Meta-learning prompting [32]	Using meta-learning to train a language model to quickly adapt to new tasks or domains using a small number of carefully crafted prompts [32]	Meta-learning models such as MAML and reptile
Zero-shot prompting [33]	Using carefully crafted prompts to enable a language model to perform tasks that it has not been explicitly trained on [33]	Transformer-based models such as GPT-3, GPT-4, and BERT
Active learning prompting [34]	Using active learning to iteratively improve the performance of a language model by selecting the most informative prompts for training [34]	Transformer-based models such as GPT-3, GPT-4, and BERT
Curriculum learning prompting [35]	Using curriculum learning to gradually increase the difficulty of the prompts used to train a language model, allowing it to learn more effectively [35]	Transformer-based models such as GPT-3, GPT-4, and BERT
Reinforcement learning prompting [36]	Using reinforcement learning to train a language model to generate responses that maximize a reward signal based on the quality of its responses to prompts [36]	Reinforcement learning models such as PPO and A2C

continuously maximizing its score functions [38]. Still, zero-shot performance is used measure the effectiveness of the prompt for overall evaluation within other LLMs.

Program synthesis for prompt engineering and automatic instruction generation is related but distinct concepts. Automatic instruction generation refers to the use of technology to automatically generate instructions or prompts. This can be done in various contexts, such as creating how-to guides or generating prompts for machine learning models. Program synthesis for prompt engineering, on the other hand, refers to the use of techniques from program synthesis to automatically generate instructions or prompts [39]. Program synthesis is fundamentally a field of research that focuses on automatically generating programs from high-level specifications, such as examples or natural language descriptions. In the context of prompt engineering, program synthesis techniques are used to generate instructions or prompts that meet certain criteria or specifications. The primary difference between the two concepts is that automatic instruction generation is a broader term that encompasses various techniques for generating instructions automatically, while program synthesis for prompt engineering refers specifically to the use of program synthesis techniques for this purpose.

3.3 Demonstration Tasks for Prompt Engineering

Few-shot learning: This technique enables in-context learning where demonstrations are provided in the prompt to steer the model to better performance [19]. The demonstrations serve as conditioning for subsequent examples where the model is expected to generate a response.

Exercise 1: Experiment with few-shot learning by providing a series of messages between the user and the assistant in the prompt as few-shot examples. Understand how this technique improves accuracy and response grounding in LLMs.

Exercise 2: Execute few-shot learning for NLP tasks like text classification, sentiment analysis, and language translation using LLMs.

Non-chat scenarios: This provides a better understanding of application of prompt engineering can for specific tasks, e.g., answering questions on specific topics [40].

Exercise 1: Use ChatGPT or GPT-4 for a non-chat scenario task like generation text that satisfies specific questions, e.g., "What is the capital city of Uganda?"

Exercise 2: Explore how prompt engineering can be applied to non-chat scenarios, such as generating text for a specific purpose or answering questions on a specific topic with Large Language Models.

Start with clear instructions: Experiment with providing clear instructions to the model in the prompt to see how it affects the generated responses.

Exercise 1: Experiment with providing clear instructions to the model in the prompt to see how it affects the generated responses. Fine tune the prompt to understand how this technique can be used with alternative LLMs to improve accuracy and grounding of given responses.

Exercise 2: Explore different techniques for providing clear instructions to the model in the prompt when working with Large Language Models.

Explore different APIs: For Azure OpenAI GPT models, there are two separate APIs where prompt engineering is important: the Chat Completion API and the Completion API. Each API requires the input data to be structured in a specific way, which affects the overall design of the prompt.

Exercise 1: Experiment with using different APIs for interacting with Azure OpenAI GPT models. Explore how these APIs can be used with Large Language Models to improve their accuracy and grounding of responses.

Exercise 2: Explore how different APIs require input data to be formatted differently and how this impacts overall prompt design when working with Large Language Models.

Validate responses: Even when using prompt engineering effectively, it is crucial to verify the responses generated by the models.

Exercise 1: Experiment with different techniques for validating responses generated by LLMs. Fine tune the prompts to comprehend the limitations of LLMs based on the known strengths and weakness.

Exercise 2: Explore different techniques for validating responses generated by LLMs when working with Large Language Models.

4 Applications, Tools, and Trends of Prompt Engineering

4.1 Applications and Tools

Since prompt engineering is a technique used to improve the performance of natural language processing (NLP) models through providing better and more focused input, there are several tools and applications available to facilitate this process for example; LangChain, a library aimed at assisting in the development of applications that combine large language models with other sources of computation or knowledge [41]. Dust.tt which is a platform that helps build large language model applications as a series of prompted calls to external models [42]. OpenPrompt which is a PyTorch-based library that offers a standard, adaptable, and expandable structure for implementing the prompt-learning process. [43]. BetterPrompt, another test suite for large language model prompts before pushing them to production [44]. Prompt Engine which is an NPM utility library for creating and maintaining prompts for large language models [45]. Promptify, a library aimed at assisting in developing a pipeline for using large language model APIs in production [46]. TextBox 2.0 which is a modern text generation library that uses Python and PyTorch to create a consistent and standardized process for using pre-trained language models for text generation [47]. ThoughtSource is a central, open resource, and community centered on data and tools for chain-of-thought reasoning in large language models [48]. There is also GPT Index. It is a very useful project constituting data structures utilized for easing the usage of large external knowledge bases with LLMs [49, 50]. Moreover, a community-driven initiative was also established to gather numerous prompts for different cases. The community can be accessed via https://huggingface.co/datasets/fka/awesome-chatgpt-prompts.

4.2 Current Research and Future Trends of Prompt Engineering

We have to acknowledge the fact that the rapid development of conversational AI has underlined lots of emerging Natural Language Processing (NLP) innovations. One of which are Large Language Models (LLMs). Such innovations have attracted attention which has particularly facilitated the growth in size, capabilities, and transformative potential of NLP driven technologies. We observe how such technologies have been scaled to various domains like education and health care. It is important to note that LLM operations involve creating and optimizing methods for input (prompts) in order to obtain appropriate good quality output (responses). This interdependency and need for good quality outputs across various domains of application is what has inspired research initiatives and defined for us various research trend in prompt engineering.

Among the inspired directions of prompt engineering research is the development of better and more sophisticated domains specific prompt engineering tools and techniques for interfacing with LLMs. The other trend is utilization of advanced techniques in machine learning to automate the synthesis and optimization of prompts for specific tasks. Opportunities are also opening up to integrate prompt engineering in other NLP research areas like transfer learning and domain adoption. This focuses on developing methods to adapt existing prompts to new domains or tasks. It also fits into transfer of knowledge learned from one task to another prompt driven task. As the world is working toward Society 5.0, a vision where every technology responsibly connects and interacts with other technologies, it opens up research opportunities on responsible utilization of prompt engineering for improved reliability and safety of LLMs. Such research initiatives usually involve developing methods for detection and mitigation of potential biases and errors in LLM inputs and outputs. It also covers aspects for ensuring that LLMs behave in predictable, traceable, trustworthy, and explainable means when applied in sensitive domains like health care.

Since prompt engineering is becoming a programming paradigm in conversational AI, it opens up opportunities for utilizing prompts to improve explainability, interpretability, and reliability of conversational AI applications most especially LLMs. This can involve developing of methods for generating prompts that enable users to understand how LLMs arrive to their outputs. This requires providing LLM explainers for LLM output decisions. Building an equitable and well-connected world drives attention on utilization of prompt engineering for efficiently and scalable deployment of LLMs. This opens up research opportunities on methods for reducing computational costs of LLMs. Some these include how humans can enable usage of large-scale datasets and distributed computing environments for LLMs.

4.3 Current Research and Future Trends of Large Language Models (LLMs)

We are certain of increasing applications of LLMs across domains. This increase comes with attention to the processes of developing larger and more powerful conversational AI tools that accurately perform a variety of tasks that require human intelligence. This will likely manifest as Conversational Artificial General Intelligence (CAGI).

This opens up research trends building methods for developing, deployment, and monitoring LLMs. It opens up research opportunities on equitable training and utilization of LLMs among various social groups. Methods of transfer learning, domain adaptation, and training LLMs on small amounts of data also open up. There are inspired research trends to investigate reasoning, planning, and decision making within various architectures of LLMs. For matters of generalizability and scalability, the future and trend of research on LLMs needs to focus on combination of technologies (Multi-modeling) for NLP, computer vision, robotics, and quantum AI. This requires development of optimal methods for integrating those technologies for effective operations. The other trend in LLMs is Responsible Conversational AI, this involves interpretability, explainability, inclusivity, trustworthiness, and responsibility in modeling, development, deployment, and monitoring LLMs as conversational AI systems. Finally, human–machine interaction is another interest developing among multiple disciplines who want to understand domain-specific human language and behavior for effective communication and control of LLMs across domains.

Whereas LLMs are useful tools, they are still limited by correctness, lack of enterprise context, stale training data, limited controllability, and private data risks. These make them perpetuate stereotypes and harm disadvantaged groups, they also spread misinformation, outright disinformation and are still constrained by computing power.

5 Conclusion

The potential and significance of prompt engineering on Large Language Models (LLMs) and conversational AI systems is undeniable. With the impressive performance demonstrated by LLMs on a wide range of NLP tasks in complex applications and use cases, we declare urgent and extra attention on prompt engineering processes and procedures. Since prompt engineering is now a programming paradigm for conversational AI, information seekers, AI engineers, and researchers need to get excited about the new opportunities and challenges that come with developing more powerful and capable Conversational AI systems.

By leveraging the latest advances in these fields, engineers and researchers can build systems that can perform a wide range of tasks more effectively and efficiently. Part of such tasks is building domain specific prompt engineering platfoms based on responisble data standards and AI practices. These will be very imporant for prompt engineering research, digital equality, coversational AI research and industrial practices.

For ordinary users and information seekers, LLMs and prompt engineering have the potential to significantly improve their interactions with Conversational AI systems. By enabling AI systems to better understand and respond to human language and behavior, LLMs and prompt engineering will make it easier for users to communicate with and control these systems. Therefore, active participation and staying up-to-date with the latest advances in these fields, AI engineers, researchers, and ordinary information seekers can all benefit from the exciting new opportunities offered by LLMs and prompt engineering.

References

1. Brants T, Popat AC, Xu P, Och FJ, Dean J (2023) Large language models in machine translation. Research.google. Online. Available: http://research.google/pubs/pub33278.pdf. Accessed 01 May 2023
2. Du Y et al (2023) Guiding pretraining in reinforcement learning with large language models. arXiv cs.LG
3. Wang Y et al (2022) AdaMix: mixture-of-adaptations for parameter-efficient model tuning. arXivcs.CL
4. Wei J et al (2022) Emergent abilities of large language models. arXiv cs.CL
5. Oppenlaender J (2022) A taxonomy of prompt modifiers for text-to-image generation. arXiv cs.MM
6. White J et al (2023) A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv cs.SE
7. Lo LS (2023) The CLEAR path: a framework for enhancing information literacy through prompt engineering. J Acad Libr 49(4):102720
8. Short CE, Short JC (2023) The artificially intelligent entrepreneur: ChatGPT, prompt engineering, and entrepreneurial rhetoric creation. J Bus Ventur Insights 19(e00388):e00388
9. Strobelt H et al (2023) Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. IEEE Trans Vis Comput Graph 29(1):1146–1156
10. Abukhalaf S, Hamdaqa M, Khomh F (2023) On codex prompt engineering for OCL generation: an empirical study. arXiv cs.SE
11. Oppenlaender J, Linder R, Silvennoinen J (2023) Prompting AI art: an investigation into the creative skill of prompt engineering. arXiv cs.HC
12. Chalkidis I (2023) ChatGPT may pass the bar exam soon, but has a long way to go for the LexGLUE benchmark. arXiv cs.CL
13. Johnson C, Rodríguez-Fernández N, Rebelo SM (2023) Artificial intelligence in music, sound, art and design. In: 12th international conference, EvoMUSART 2023, held as part of EvoStar 2023, Brno, Czech Republic, Apr 12–14, 2023, proceedings. Springer Nature, Cham, Switzerland
14. Shtedritski A, Rupprecht C, Vedaldi A (2023) What does CLIP know about a red circle? Visual prompt engineering for VLMs. arXiv cs.CV
15. Polak MP, Morgan D (2023) Extracting accurate materials data from research papers with conversational language models and prompt engineering—example of ChatGPT. arXiv cs.CL
16. Busch K, Rochlitzer A, Sola D, Leopold H (2023) Just tell me: prompt engineering in business process management. arXiv cs.AI
17. Kumar K (2023) Geotechnical parrot tales (GPT): harnessing large language models in geotechnical engineering. arXiv cs.CL
18. Trautmann D, Petrova A, Schilder F (2022) Legal prompt engineering for multilingual legal judgement prediction. arXiv cs.CL
19. Ahmed T, Pai KS, Devanbu P, Barr ET (2023) Improving few-shot prompts with relevant static analysis products. arXiv cs.SE
20. Diao S, Wang P, Lin Y, Zhang T (2023) Active prompting with chain-of-thought for large language models. arXiv cs.CL
21. Taveekitworachai P, Abdullah F, Dewantoro MF, Thawonmas R, Togelius J, Renz J (2023) ChatGPT4PCG competition: character-like level generation for science birds. arXiv cs.AI
22. Kather JN, Ghaffari Laleh N, Foersch S, Truhn D (2022) Medical domain knowledge in domain-agnostic generative AI. NPJ Digit Med 5(1):90

23. van Dis EAM, Bollen J, Zuidema W, van Rooij R, Bockting CL (2023) ChatGPT: five priorities for research. Nature 614(7947):224–226
24. Yang Z et al (2023) MM-REACT: prompting ChatGPT for multimodal reasoning and action. arXiv cs.CV
25. Khattak MU, Rasheed H, Maaz M, Khan S, Khan FS (2022) MaPLe: multi-modal prompt learning. arXiv cs.CV
26. Wang B, Deng X, Sun H (2022) Iteratively prompt pre-trained language models for chain of thought. arXiv cs.CL, pp 2714–2730
27. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2023) Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comput Surv 55(9):1–35
28. Yang Z, Li Z, Zheng F, Leonardis A, Song J (2022) Prompting for multi-modal tracking. In: Proceedings of the 30th ACM international conference on multimedia
29. Zhu J, Lai S, Chen X, Wang D, Lu H (2023) Visual prompt multi-modal tracking. arXiv cs.CV
30. Maus N, Chao P, Wong E, Gardner J (2023) Adversarial prompting for black box foundation models. arXiv cs.LG
31. Wang Z, Panda R, Karlinsky L, Feris R, Sun H, Kim Y (2023) Multitask prompt tuning enables parameter-efficient transfer learning. arXiv cs.CL
32. Zhang H, Zhang X, Huang H, Yu L (2022) Prompt-based meta-learning for few-shot text classification. In: Proceedings of the 2022 conference on empirical methods in natural language processing, pp 1342–1357
33. Kojima T, Gu SS, Reid MM, Matsuo Y, Iwasawa Y (2022) Large language models are zero-shot reasoners. arXiv cs.CL
34. Köksal A, Schick T, Schütze H (2022) MEAL: stable and active learning for few-shot prompting. arXiv cs.CL
35. Lin J, Chen Q, Zhou J, Jin J, He L (2022) CUP: curriculum learning based prompt tuning for implicit event argument extraction. arXiv cs.CL
36. Zhang T, Wang X, Zhou D, Schuurmans D, Gonzalez JE (2022) TEMPERA: test-time prompting via Reinforcement learning. arXiv cs.CL
37. Zhou Y et al (2022) Steering large language models using APE
38. Zhou Y et al (2022) Large language models are human-level prompt engineers. arXiv cs.LG
39. Austin J et al (2021) Program synthesis with large language models. arXiv cs.PL
40. Sun K et al (2020) Adding chit-chat to enhance task-oriented dialogues. arXiv cs.CL
41. Chase H (2023) Welcome to langchain—langchain 0.0.154. Langchain.com. Online. Available: https://python.langchain.com/en/latest/index.html. Accessed 01 May 2023
42. Dust—design and deploy large language models apps. Dust.tt. Online. Available: https://dust.tt/. Accessed 01 May 2023
43. "OpenPrompt," Openprompt.co. Online. Available: https://openprompt.co/. Accessed 01 May 2023
44. "The art & science of AI prompts," The Art & Science of AI Prompts. Online. Available: https://www.betterprompts.ai/. Accessed 01 May 2023
45. "Promptengines.com," Afternic.com. Online. Available: https://www.afternic.com/forsale/promptengines.com?traffic_id=GoDaddy_DLS&traffic_type=TDFS&utm_campaign=TDFS_GoDaddy_DLS&utm_medium=sn_affiliate_click&utm_source=TDFS. Accessed 01 May 2023
46. "Promptify.Ai," Promptify.ai. Online. Available: https://www.promptify.ai/. Accessed 01 May 2023
47. TextBox: TextBox 2.0 is a text generation library with pre-trained language models
48. ThoughtSource: A central, open resource for data and tools related to chain-of-thought reasoning in large language models. Developed @ Samwald research group: https://samwald.info/
49. G.-3 Demo, "GPT index," Gpt3demo.com. Online. Available: https://gpt3demo.com/apps/gpt-index. Accessed 01 May 2023
50. "llamaindex (LlamaIndex)," Huggingface.co. Online. Available: https://huggingface.co/llamaindex. Accessed 01 May 2023