
A Survey of Techniques, Key Components, Strategies, Challenges, and Student Perspectives on Prompt Engineering for Large Language Models (LLMs) in Education

Wan Chong Choi^{*} and Chi In Chang

Posted Date: 25 March 2025

doi: 10.20944/preprints202503.1808.v1

Keywords: prompt engineering; large language models; LLM; Artificial intelligence generated content; AIGC; artificial intelligence in education; chatbot tutoring; prompt design strategies; chain-of-thought; AI ethics; instructional design with LLMs; ChatGPT; POE chatbot; natural language processing; programming education; student perspectives



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

A Survey of Techniques, Key Components, Strategies, Challenges, and Student Perspectives on Prompt Engineering for Large Language Models (LLMs) in Education

Wan Chong Choi ^{1,2,*} and Chi In Chang ²

¹ Department of Computer Science, Illinois Institute of Technology, U.S.

² Department of Psychology, Golden Gate University, U.S.; cchang@my.ggu.edu

* Correspondence: wchoi8@hawk.iit.edu

Abstract: This study presented a comprehensive investigation into prompt engineering for large language models (LLMs) within educational contexts, combining a systematic literature review with a 12-week empirical study involving primary school students using a chatbot-based tutor in a Python programming course. The research explored the breadth of prompt engineering techniques, identified essential components for effective educational prompts, examined strategic applications, highlighted key implementation challenges, and captured learner perspectives on interacting with LLMs. Our review categorized prompt engineering techniques into foundational (e.g., zero-shot, few-shot, and direct instruction), structured reasoning (e.g., chain-of-thought, tree-of-thought, and graph-based models), hallucination reduction (e.g., retrieval-augmented generation, CoVe, ReAct), user-centric strategies (e.g., automatic prompt engineering, active prompting), and domain-specific applications (e.g., emotion prompting, contrastive reasoning, and code generation tools like PoT and CoC). We also examined advanced optimization methods including prompt tuning, abstraction, and self-consistency approaches that enhanced both reasoning and factual reliability. Key components of effective educational prompt engineering were distilled into nine categories: content knowledge, critical thinking, iterative refinement, clarity, creativity, collaboration, digital literacy, ethical reasoning, and contextual integration. These elements collectively supported both the quality of LLM outputs and the development of students' cognitive and metacognitive skills. Strategically, we identified ten educational prompt engineering practices—contextual framing, task segmentation, prompt sequencing, role-based prompting, reflection, counterfactual exploration, constraint-based creativity, ethical consideration, interactive refinement, and comparative analysis—as essential for guiding LLM interactions aligned with pedagogical goals. We also addressed core challenges in prompt engineering for education, including ambiguity in model interpretation, balancing specificity and flexibility, ensuring consistency, mitigating hallucinations, safeguarding ethics and privacy, and maintaining student engagement. These challenges highlighted the need for explicit instructional support and adaptive prompt design in classrooms. Empirically, our study of primary school learners revealed a surprising level of sophistication in students' prompt construction and refinement. Students developed intuitive understandings of prompt clarity, used context to guide AI responses, adopted role-based and scenario-based prompting, applied constraints to improve learning outcomes, and created reusable prompt templates. Furthermore, they engaged in iterative refinement, developed evaluation criteria for AI responses, and differentiated between general and specific prompts based on their learning objectives. These findings underscored students' emerging metacognitive awareness and adaptability in AI-mediated learning.

Keywords: prompt engineering; large language models; LLM; Artificial intelligence generated content; AIGC; artificial intelligence in education; chatbot tutoring; prompt design strategies; chain-of-thought; AI ethics; instructional design with LLMs; ChatGPT; POE chatbot; natural language processing; programming education; student perspectives

1. Introduction

The rapid growth of large language models (LLMs) has created new prospects in education. Researchers examined how prompts act as important instructions that allow an LLM to generate the desired outcome from an open-ended text input [1]. In education, learners may use prompts to get explanations, code reviews, or tailored study materials, but they often struggle with formulating these prompts accurately or clearly.

Recent studies suggested that prompt engineering can be taught in various fields since prompts are unique textual inputs intended to regulate the outputs of an LLM [2]. While many universities and schools now allow or encourage the adoption of LLM-based tools, some educators remain uncertain about how to direct students to craft prompts that produce relevant responses for academic tasks. Moreover, students may lack awareness of the technical nuances behind how prompts affect performance [3]. These issues prompt a need to review fundamental aspects of prompt engineering, associated recommendations, and obstacles students face in educational contexts.

1.1. *What Is Prompt Engineering for LLMs*

Prompt engineering is often described as forming concise textual instructions and contextual hints to guide an LLM toward correct or meaningful outputs [4]. Prompt engineering relies on natural language interactions to reach outcomes, unlike conventional programming, where a programming language enforces syntax. Hence, it balances linguistic clarity with domain-specific requests.

Core components of prompt engineering involve specifying roles, limiting the scope, and emphasizing the format of requested outputs. LLMs can better interpret user queries when prompts are crafted clearly, including direct questions or explicit instructions. Conversely, ambiguous language tends to cause the model to offer unclear or incorrect answers. By studying effective prompt patterns, practitioners identified common structures—such as giving an example or providing short instructions—to build prompts that are generally robust across different educational tasks.

1.2. *The Role of Prompts in Interacting with LLMs*

Prompts serve as the foundation for interaction between the user and the LLM. They help shape responses by indicating the type of information needed and the preferred tone or style. For example, a teacher preparing a language arts assignment can craft a prompt that urges the LLM to respond analytically or highlight grammatical rules. This tailored communication fosters direct alignment between learning targets and AI-generated content.

In educational contexts, prompts can facilitate open-ended discussions. Students can propose questions about a historical event and refine their prompt to obtain specific timelines or analyses. This iterative approach reveals how better prompting leads to better answers. Nonetheless, the LLM response might be confusing if a learner's prompt is too vague or lacks context. These issues underscore the importance of prompt guidelines, which teach students to phrase questions that guide the LLM, resulting in outputs that meet classroom needs.

1.3. *Challenges for Students Using Prompts in Education*

Despite the potential advantages of LLMs in education, common challenges restrict student success in prompt engineering. A significant problem is that learners often do not know what they want to ask or how to phrase it.

1.3.1. *Students Unsure What to Ask*

When using an LLM, middle school, high school, or even college students might be overwhelmed by broad possibilities. Without guidance, they might type a trivial request and gain little insight. Encouraging them to break tasks into smaller pieces—"Explain the main argument of this article" or "Compare two opposing viewpoints on this topic"—helps students identify essential questions.

1.3.2. Students Unable to Form Efficient Prompts

Even if they know the question, students can struggle to frame concise and well-targeted prompts. Some might use wordy or ambiguous sentences, which produce unclear outputs. Simple, direct language, plus any relevant constraints—like word length, viewpoint, or structure—make LLM interactions more fruitful. Teaching students through examples (e.g., "Show me a 200-word summary" or "Outline three pros and cons") helps develop prompt engineering abilities.

1.3.3. Instructors Unsure How to Improve Student Prompt Skills

Instructors remained uncertain about how to support students in refining prompt skills. Traditional technology instruction may skip the subtlety of guiding a large language model with text prompts [5]. Educators might also be reluctant because they worry about the potential misuse of AI. Clear classroom exercises, showing how different prompts change the responses, can address teacher concerns. In particular, guided demonstrations and structured assignments let students practice prompt engineering, gather feedback, and realize how prompt variation influences the results.

1.4. Research Questions

According to the above background, this study was guided by the following research questions:

RQ1: What are the various prompt engineering techniques developed for effectively utilizing LLMs in education?

RQ2: What key components constitute effective prompt engineering in education?

RQ3: What strategies can educators and students employ to enhance interactions with LLMs through prompt engineering?

RQ4: What challenges are associated with prompt engineering within educational settings?

RQ5: What are students' perspectives and experiences with prompt engineering techniques integrated into chatbot-based tutoring systems?

2. RQ1: Prompt Engineering Techniques

Prompt engineering emerged as a critical skill for effectively utilizing LLMs in educational settings. We conducted a comprehensive review of existing literature [2,6–17] and concluded the findings. This section examines the techniques developed to optimize interactions with LLMs, categorizing them based on their methodological foundations and applications.

2.1. Foundational Techniques

2.1.1. Zero-Shot Prompting

Zero-shot prompting is a technique where an LLM performs a task based solely on instructions without exposure to examples during prompt formulation. This approach relies on the model's pre-existing knowledge acquired during pre-training to infer the desired output directly from a structured prompt without requiring labeled data for training on specific input-output mappings.

For example, zero-shot prompting can be effectively utilized for sentiment analysis, translation, and short-form text generation tasks. An illustrative prompt might be:

Classify the sentiment of the following sentence as positive, negative, or neutral:

"The lecture was engaging and insightful."

The significant advantages of zero-shot prompting include its flexibility, low resource requirements, and rapid deployment capability. However, its accuracy can be sensitive to prompt clarity and precision, as subtle wording variations might dramatically affect the interpretation and output.

2.1.2. Few-Shot Prompting

Few-shot prompting provides the model with limited task-specific examples within the prompt, which helps guide the model's reasoning and improve accuracy. This technique provides models with input-output examples to induce an understanding of a given task, unlike zero-shot prompting, where no examples are supplied.

Providing even a few high-quality examples has improved model performance on complex tasks compared to no demonstration. However, few-shot prompting requires additional tokens to include the examples, which may become prohibitive for longer text inputs. For instance, an illustrative prompt might be:

Q: What is the capital of France? A: Paris.

Q: What is the capital of Japan? A: Tokyo.

Q: What is the capital of Italy?

The selection and composition of prompt examples can significantly influence model behavior, and biases like favoring frequent words may still affect few-shot results. While few-shot prompting enhances capabilities for complex tasks, especially among large pre-trained models, careful prompt engineering is critical to achieve optimal performance and mitigate unintended model biases.

2.1.3. Direct Instruction Prompting

Direct instruction prompting explicitly states the task or question to be answered by the model without ambiguity. This approach typically provides precise, concise instructions with clear expected outcomes, significantly reducing interpretation errors by stating the central idea clearly and concisely. For instance, an illustrative prompt might be:

Summarize the following paragraph in two sentences:

"The rapid growth of large language models has transformed how students engage with digital learning tools. These AI systems can generate explanations, provide assignment feedback, and create personalized learning materials. However, educators remain concerned about potential overreliance and the need to develop proper usage guidelines."

Direct instruction prompts provide clarity that enhances model output consistency. However, overly detailed instructions might inadvertently restrict the model's creative or inferential capabilities. Thus, balancing specificity and flexibility is crucial.

2.2. Structured Reasoning Techniques

2.2.1. Chain-of-Thought (CoT) Prompting

Chain-of-Thought (CoT) prompting explicitly encourages the model to generate a logical, step-by-step reasoning pathway, facilitating more precise and structured responses. This method is beneficial for educational tasks involving mathematical problem-solving or multi-step reasoning, where LLMs often stumble in the face of complex reasoning, limiting their potential.

The primary contribution of CoT lies in its ability to guide LLMs through a logical reasoning chain, resulting in responses that reflect a deeper understanding of the given prompts. For example, the prompt would show the reasoning process and final answer for a multi-step math word problem and mimic how humans break down problems into logical intermediate steps:

Prompt: If a student has 12 pencils and gives 4 to friends, then buys 3 more pencils, how many pencils does the student have now?

LLM Thinking: The student starts with 12 pencils. Then gives away 4 pencils, leaving 8 pencils ($12 - 4 = 8$). The student then buys 3 more pencils. So the total number of pencils is $8 + 3 = 11$ pencils.

LLM Response: 11 pencils.

This structured approach significantly improved the interpretability and reliability of the model's output, aligning closely with human cognitive problem-solving strategies. The researchers

achieved state-of-the-art performance in math and commonsense reasoning benchmarks by utilizing CoT prompts for LLMs.

2.2.2. Automatic Chain-of-Thought (Auto-CoT) Prompting

Manual creation of high-quality CoT examples is both time-consuming and suboptimal. Auto-CoT automatically instructs LLMs with a "Let's think step-by-step" prompt to generate reasoning chains. Recognizing the possibility of errors in individually generated chains, Auto-CoT enhances robustness through diverse sampling.

It samples various questions and generates distinct reasoning chains for each, forming a final set of demonstrations. This automated diverse sampling minimizes errors and enhances few-shot learning, eliminating the need for labor-intensive manual creation of reasoning chains. Auto-CoT demonstrated enhanced performance, surpassing the CoT paradigm with average accuracy improvements on arithmetic and symbolic reasoning tasks.

2.2.3. Self-Consistency

Self-consistency is a decoding strategy enhancing reasoning performance compared to greedy decoding in CoT prompting. For complex reasoning tasks with multiple valid paths, self-consistency generates diverse reasoning chains by sampling from the language model's decoder. It then identifies the most consistent final answer by marginalizing these sampled chains.

This approach capitalizes on the observation that problems requiring thoughtful analysis often entail greater reasoning diversity, leading to a solution. Combining self-consistency and chain-of-thought prompting resulted in significant accuracy improvements across various benchmarks compared to the baseline chain-of-thought prompting.

2.2.4. Logical Chain-of-Thought (LogiCoT) Prompting

The ability to perform logical reasoning is critical for LLMs to solve complex, multi-step problems across diverse domains. Existing methods, like CoT prompting, encourage step-by-step reasoning but lack effective verification mechanisms.

LogiCoT is a neurosymbolic framework that leverages principles from symbolic logic to enhance reasoning coherently and in a structured way. Specifically, LogiCoT applies the concept of *reductio ad absurdum* to verify each step of reasoning generated by the model and provide targeted feedback to revise incorrect steps. LogiCoT reduced logical errors and hallucinations through a think-verify-revise loop, improving reasoning abilities on various datasets.

2.2.5. Chain-of-Symbol (CoS) Prompting

LLMs often struggle with tasks involving complex spatial relationships due to their reliance on natural language, which is susceptible to ambiguity and biases. CoS employs condensed symbols instead of natural language, providing distinct advantages: clear and concise prompts, heightened spatial reasoning for LLMs, and improved human interpretability.

Implementing CoS led to substantial improvements in model performance across spatial reasoning tasks. It also reduced prompt length considerably, making the process more efficient without sacrificing accuracy.

2.2.6. Tree-of-Thoughts (ToT) Prompting

ToT extends CoT prompting by managing a tree structure of intermediate reasoning steps, known as "thoughts". Each thought represents a coherent language sequence moving toward the final solution. This structure allows language models to deliberately reason by assessing the progress generated by thoughts in solving the problem.

ToT integrates the model's abilities to produce and evaluate thoughts with search algorithms like breadth-first or depth-first search. This enables systematic exploration among reasoning chains,

with a look-ahead to expand promising directions and to backtrack when solutions are incorrect. ToT excelled in tasks like the Game of 24, achieving a significantly higher success rate than CoT. Additionally, ToT outperformed CoT with a higher success rate in word-level tasks.

2.2.7. Graph-of-Thoughts (GoT) Prompting

The inherent non-linear nature of human thought processes challenges the conventional sequential approach of CoT prompting. GoT is a graph-based framework advancing traditional sequential methods to better align with the non-linear characteristics of human thinking.

This framework permits dynamic interplay, backtracking, and evaluation of ideas, allowing the aggregation and combination of thoughts from various branches, departing from the linear structure of the tree of thoughts. The key contributions encompass modeling the reasoning process as a directed graph, offering a modular architecture with diverse transformation operations. The GoT reasoning model demonstrated substantial gains over the CoT baseline, improving accuracy on various datasets.

2.2.8. System 2 Attention (S2A) Prompting

The soft attention mechanism in Transformer-based LLMs is prone to incorporating irrelevant context information, impacting token generation adversely. System 2 Attention (S2A) utilizes the reasoning abilities of LLMs to selectively attend to relevant portions by regenerating the input context.

S2A employs a two-step process to enhance attention and response quality by employing context regeneration and response generation with refined context. The effectiveness of S2A was evaluated across various tasks, including factual QA, long-form generation, and math word problems. In factual QA, S2A improved answer accuracy, clearly enhancing factual consistency.

2.2.9. Thread of Thought (ThoT) Prompting

Thread of Thought (ThoT) is a prompting technique designed to enhance the reasoning abilities of LLMs within chaotic contexts. ThoT, inspired by human cognition, systematically examines extensive contexts into manageable segments for incremental analysis, employing a two-phase approach where the LLM first summarizes and examines each segment before refining the information for a final response.

ThoT's flexibility shines as a versatile "plug-and-play" module, enhancing reasoning across different models and prompting methods. Evaluations of question-answering and conversation datasets revealed substantial performance improvements, especially in chaotic contexts.

2.2.10. Chain of Table Prompting

Approaches like CoT, PoT, and ToT represent reasoning steps through free-form text or code, which face challenges when dealing with intricate table scenarios. Chain-of-Table uses step-by-step tabular reasoning by dynamically generating and executing common SQL/DataFrame operations on tables.

The iterative nature of this process enhances intermediate results, empowering LLMs to make predictions through logically visualized reasoning chains. Significantly, Chain-of-Table consistently improved the performance of benchmark tabular datasets.

2.3. Hallucination Reduction Techniques

2.3.1. Retrieval Augmented Generation (RAG)

LLMs have revolutionized text generation, yet their reliance on limited, static training data hinders accurate responses, especially in tasks demanding external knowledge. Retrieval Augmented Generation (RAG) emerged as a novel solution, seamlessly weaving information retrieval into the prompting process.

RAG analyzes user input, crafts a targeted query, and scours a pre-built knowledge base for relevant resources. Retrieved snippets are incorporated into the original prompt, enriching it with contextual background. The augmented prompt empowers the LLM to generate creative, factually accurate responses. RAG's agility overcame static limitations, making it a game-changer for tasks requiring up-to-date knowledge.

2.3.2. ReAct Prompting

Unlike previous studies that treated reasoning and action separately, ReAct enables LLMs to generate reasoning traces and task-specific actions concurrently. This interleaved process enhances the synergy between reasoning and action, facilitating the model in inducing, tracking, and updating action plans while handling exceptions.

ReAct was applied to diverse language and decision-making tasks, showcasing its effectiveness over state-of-the-art baselines. Notably, ReAct addressed hallucination and error propagation issues in question answering and fact verification by interacting with a simple Wikipedia API, producing more interpretable task-solving trajectories.

2.3.3. Chain-of-Verification (CoVe) Prompting

To address hallucinations in LLMs, Chain-of-Verification (CoVe) involves a systematic four-step process, including generating baseline responses, planning verification questions to check its work, answering the questions independently, and producing a revised response incorporating the verification.

By verifying its work through this deliberate multi-step approach, the LLM enhances logical reasoning abilities and reduces errors even with contradictory information. CoVe emulates human verification to bolster the coherence and precision of LLM output. Experiments demonstrated that CoVe decreased hallucinations while maintaining facts. Focused verification questions helped models identify and correct their inaccuracies.

2.3.4. Chain-of-Note (CoN) Prompting

Retrieval-augmented language models (RALMs) enhance LLMs by incorporating external knowledge to reduce factual hallucination. However, the reliability of retrieved information is not guaranteed, leading to potentially misguided responses.

Standard RALMs struggle to assess their knowledge adequacy and often fail to respond with "unknown" when lacking information. CoN systematically evaluates document relevance, emphasizing critical and reliable information to filter out irrelevant content, resulting in more precise and contextually relevant responses. Testing across diverse open-domain question-answering datasets demonstrated notable improvements.

2.3.5. Chain-of-Knowledge (CoK) Prompting

Traditional prompting techniques for LLMs have proven influential in tackling basic tasks. However, their efficacy diminishes due to complex reasoning challenges, often resulting in unreliable outputs plagued by factual hallucinations and opaque thought processes.

This limitation arises from their reliance on fixed knowledge sources, ineffective structured query generation, and lack of progressive correction that fails to guide the LLM adequately. Motivated by human problem-solving, CoK systematically breaks down intricate tasks into well-coordinated steps. The process initiates with a comprehensive reasoning preparation stage, where the context is established and the problem is framed, meticulously gathering evidence from various sources.

2.4. Interactive and User-Centric Techniques

2.4.1. Active Prompting

Active-Prompting addressed the challenge of adapting LLMs to diverse reasoning tasks by proposing to enhance LLMs' performance on complex question-and-answer tasks through task-specific example prompts with chain-of-thought (CoT) reasoning.

Unlike existing CoT methods that rely on fixed sets of human-annotated exemplars, Active-Prompt introduced a mechanism for determining the most impactful questions for annotation. Drawing inspiration from uncertainty-based active learning, the method utilized various metrics to characterize uncertainty and selected the most uncertain questions for annotation. Active-Prompting exhibited superior performance, outperforming self-consistency on complex reasoning tasks.

2.4.2. Automatic Prompt Engineer (APE)

While crafting effective prompts for LLMs has traditionally been a laborious task for expert annotators, Automatic Prompt Engineer (APE) is an innovative approach to automatic instruction generation and selection for LLMs.

APE shed the limitations of static, hand-designed prompts by dynamically generating and selecting the most impactful prompts for specific tasks. This ingenious method analyzes user input, crafts candidate instructions, and then leverages reinforcement learning to choose the optimal prompt, adapting it to different contexts. Extensive tests revealed APE's prowess, exceeding human-authored prompts in most cases and significantly boosting LLMs' reasoning abilities.

2.4.3. Automatic Reasoning and Tool-Use (ART)

The limited reasoning abilities and lack of external tool utilization hindered the potential of LLMs in complex tasks. Automatic Reasoning and Tool-use (ART) tackled this critical barrier that empowers LLMs to reason through multi-step processes and seamlessly integrate external expertise.

ART bridged the reasoning gap, enabling LLMs to tackle complex problems and expand beyond simple text generation. By integrating external tools for specialized knowledge and computations, ART unlocked unprecedented versatility and informed LLM outputs with real-world relevance. Moving beyond traditional prompting techniques, ART automated reasoning steps through structured programs, eliminating the need for laborious hand-crafting.

2.5. Domain-Specific Techniques

2.5.1. Contrastive Chain-of-Thought (CCoT) Prompting

Traditional CoT prompting for LLMs often misses a crucial element: learning from mistakes. Contrastive Chain-of-Thought Prompting (CCoT) dived in, providing valid and invalid reasoning demonstrations alongside original prompts.

This dual-perspective approach, tested on reasoning benchmarks, pushed LLMs to step-by-step reasoning, leading to substantial improvements in strategic and mathematical reasoning evaluations compared to traditional CoT, further improved when integrated with self-consistency techniques. This method is beneficial for educational contexts where learning from mistakes is essential.

2.5.2. Emotion Prompting

While LLMs demonstrate impressive capabilities on various tasks, their ability to comprehend psychological and emotional cues remains uncertain. EmotionPrompt addressed the uncertainty surrounding LLMs' ability to comprehend emotional cues by introducing emotional stimulus sentences to prompts to enhance LLM emotional intelligence.

Experimental results demonstrated seamless integration of these stimuli, significantly improving LLM performance across various tasks. EmotionPrompt demonstrated a relative

improvement in instruction induction and an impressive boost in tasks, underscoring its efficacy in augmenting LLM capabilities in processing affective signals. This technique has significant implications for educational contexts where emotional engagement is crucial for learning.

2.5.3. Code Generation Techniques

Scratchpad Prompting

Despite the prowess of Transformer-based language models in generating code for basic programming tasks, they encounter challenges in complex, multi-step algorithmic calculations requiring precise reasoning. Addressing this, a novel approach centered on task design rather than model modification introduced a 'scratchpad' concept.

The proposal enables the model to generate an arbitrary sequence of intermediate tokens before providing the final answer. Scratchpad Prompting technique outperformed with a higher success rate. Combining CodeNet and single-line datasets yielded the highest performance, achieving correct final outputs and perfect traces. However, the scratchpad prompting technique faced limitations, including a fixed context window size and a dependency on supervised learning for scratchpad utilization.

Program of Thoughts (PoT) Prompting

Language models are suboptimal for solving mathematical expressions due to their proneness to arithmetic errors, incapability in handling complex equations, and inefficiency in expressing extensive iterations. Program-of-Thoughts (PoT) prompting advocated using external language interpreters for computation steps to enhance numerical reasoning in language models.

PoT enabled models to express reasoning through executable Python programs, resulting in an average performance improvement compared to CoT prompting on datasets involving mathematical word problems and financial questions. This technique is particularly valuable in educational contexts requiring precise mathematical reasoning.

Structured Chain-of-Thought (SCoT) Prompting

LLMs have exhibited impressive proficiency in code generation. The widely used CoT prompting involves producing intermediate natural language reasoning steps before generating code. Despite its efficacy in natural language generation, CoT prompting demonstrated lower accuracy when applied to code generation tasks.

Structured Chain-of-Thought (SCoTs) is an innovative prompting technique tailored specifically for code generation. By incorporating program structures (sequence, branch, and loop structures) into reasoning steps, SCoT prompting enhanced LLMs' performance in generating structured source code. This approach explicitly guided LLMs to consider requirements from the source code perspective, improving their overall effectiveness in code generation compared to CoT prompting.

Chain of Code (CoC) Prompting

While CoT prompting has proven very effective for enhancing Language models (LMs) semantic reasoning skills, it struggles to handle questions requiring numeric or symbolic reasoning. Chain-of-Code (CoC) is an extension to improve LM reasoning by leveraging code writing for logic and semantic tasks.

CoC encouraged LMs to format semantic sub-tasks as flexible pseudocode, allowing an interpreter to catch undefined behaviors and simulate them with an "LMulator". Experiments demonstrated CoC's superiority over Chain of Thought and other baselines, achieving a higher accuracy on benchmarks. CoC proved effective with both large and small models, expanding LMs' ability to correctly answer reasoning questions by incorporating a "think in code" approach.

2.6. Advanced Optimization Techniques

2.6.1. Optimization by Prompting (OPRO)

In various domains, optimization is a fundamental process often involving iterative techniques. Optimization by PROMpting (OPRO) is a novel approach that leverages LLMs as optimizers. Unlike traditional methods, OPRO utilizes natural language prompts to iteratively generate solutions based on the problem description, enabling quick adaptation to different tasks and customization of the optimization process.

The potential of LLMs for optimization was demonstrated through case studies on classic problems like linear regression and the traveling salesman problem. Additionally, it explored the optimization of prompts to maximize accuracy in natural language processing tasks, highlighting the sensitivity of LLMs. The experiments showed that optimizing prompts for accuracy on a small training set effectively translated to high performance on the test set.

2.6.2. Rephrase and Respond (RaR) Prompting

The study highlighted an often-neglected dimension in exploring LLMs: the disparity between human thought frames and those of LLMs, and introduced Rephrase and Respond (RaR). RaR allowed LLMs to rephrase and expand questions in a single prompt, demonstrating improved comprehension and response accuracy.

The two-step RaR variant, incorporating rephrasing and response LLMs, achieved substantial performance enhancements across various tasks. The study highlighted that in contrast to casually posed human queries, the rephrased questions contributed to enhanced semantic clarity and the resolution of inherent ambiguity. These findings offered valuable insights for understanding and improving the efficacy of LLMs across various applications.

2.6.3. Take a Step Back Prompting

The Step-Back prompting technique was tailored explicitly for advanced language models to address the persistent challenge of complex multi-step reasoning. This innovative approach empowered models to engage in abstraction, extracting high-level concepts and fundamental principles from specific instances.

The Step-Back prompting method involved a two-step procedure, integrating Abstraction and Reasoning. The results demonstrated a substantial enhancement in reasoning capabilities through extensive experiments, applying Step-Back Prompting to LLMs in diverse reasoning-intensive tasks such as STEM, Knowledge QA, and Multi-Hop Reasoning. Noteworthy performance boosts were observed in various tasks.

3. RQ2: Key Components of Prompt Engineering in Education

Prompt engineering is a structured method to design practical inputs (prompts) for LLMs to enhance their performance in diverse applications, particularly education. Recent research highlighted several critical components integral to successful prompt engineering: content knowledge, critical thinking, iterative design, clarity and precision, and user comprehension. We conducted a comprehensive review of existing literature [2,8,9,18–23] and concluded the findings.

3.1. Content Knowledge

Content knowledge refers to the depth and specificity of the information embedded within prompts. Prompts should contain accurate, detailed, and domain-specific information to guide LLMs effectively. Cain [9] emphasized that the success of LLMs in educational applications significantly depends on the precision and relevance of the content knowledge included in the prompts. The underlying reason is that LLMs rely on contextual information in prompts to generate coherent and relevant outputs.

Effective prompt engineering in education must ensure that prompts align closely with the targeted learning objectives. For example, prompts used to assist students in writing essays should contain specific instructions about the essay structure, topic depth, and relevant terminologies. Similarly, prompts designed for generating Python code for educational purposes must incorporate clear algorithmic instructions to enable LLMs to produce accurate and pedagogically sound code.

3.2. Critical Thinking

Critical thinking within prompt engineering involves designing prompts that encourage LLMs to generate outputs demonstrating analytical reasoning and reflective judgment. Embedding critical thinking directives within prompts, such as requests for justifications, comparisons, or evaluative judgments is necessary. Such prompts compel LLMs to provide facts, synthesize information, evaluate evidence, and offer logically reasoned arguments.

In educational contexts, integrating critical thinking components in prompts is crucial for fostering higher-order cognitive skills among students. Different studies asserted that prompts encourage critical online reasoning, a subset of critical thinking that significantly enhances students' digital literacy and capacity to critically evaluate AI-generated information. Thus, prompts structured to challenge assumptions, question validity, and seek evidence-based reasoning substantially improve the educational effectiveness of LLM-generated outputs.

3.3. Iterative Design

Iterative design refers to the cyclical process of prompt refinement through repeated testing, feedback incorporation, and continual improvement. This approach acknowledges that initial prompt formulations are rarely optimal and thus require iterative revisions to achieve desired outcomes. The iterative design is essential in optimizing prompt engineering, underscoring the necessity of continuous refinement based on user feedback and output analysis.

The iterative design facilitates prompt enhancement through two principal methods: empirical evaluation of LLM outputs and systematic refinement based on clearly defined performance criteria. Iterative design involves adjusting prompt length, specificity, structure, and phrasing to maximize LLM performance on targeted tasks. This iterative approach is particularly beneficial in educational settings, where the precision of instructional prompts can significantly influence learning outcomes, learner engagement, and overall educational efficacy.

3.4. Clarity and Precision

Clarity and precision are essential for prompt effectiveness. Ambiguities or overly broad language within prompts often lead to inconsistent or irrelevant outputs from LLMs. Prompt clarity is achieved through precise language that unequivocally conveys the intended task, thus minimizing interpretive errors by the AI. Explicit and articulated prompts significantly enhance model predictability and the quality of generated responses.

Educational applications significantly benefit from prompts characterized by clarity and precision. For instance, when prompts are crafted with explicit instructional guidelines, such as specifying the exact format of desired responses or clearly defining the scope of inquiry, LLM outputs tend to align more closely with educational objectives. Such meticulous, prompt formulation reduces the cognitive load on learners, allowing them to focus more effectively on the educational content rather than deciphering ambiguous instructions.

3.5. Creativity

Creativity is vital in effective prompt engineering within educational settings, fostering innovative and engaging interactions between learners and Large Language Models (LLMs). Creative prompt design focuses on crafting prompts that meet clear educational goals while sparking learners' curiosity, imagination, and inventive thinking. For instance, in a history lesson, a creative prompt

might ask students to picture themselves as advisors to a historical figure, tasked with suggesting solutions to a major challenge of that time. This method pushes students beyond simple memorization, encouraging them to use their knowledge in new and imaginative ways.

Incorporating creativity into prompts helps LLMs generate responses that avoid being basic or repetitive, boosting student engagement and motivation. Well-designed creative prompts ignite learners' intrinsic interest and inspire them to dive deeper into subjects beyond conventional boundaries. These prompts often feature open-ended questions, hypothetical situations, or imaginative tasks that prompt thoughtful and original answers. Students engaging with such prompts tend to display greater enthusiasm and a stronger willingness to explore challenging topics.

Moreover, creative prompts enhance learners' problem-solving abilities and adaptability. By introducing scenarios that require innovative thinking and flexible application of knowledge, these prompts enable students to tackle challenges from various perspectives and develop diverse solutions. For example, in a science class, a prompt asking students to design an experiment under unusual constraints can spark inventive experimental approaches and deepen their understanding of scientific principles. The value of these prompts lies in their capacity to encourage creative knowledge application, which supports more profound learning and better retention.

Ultimately, weaving creativity into prompt engineering enriches the educational experience by promoting active participation in building knowledge. This approach aligns closely with modern educational aims, highlighting critical thinking, problem-solving, and adaptability—essential skills for success in today's rapidly evolving academic and professional landscapes.

3.6. Collaboration

Collaboration refers to the design of prompts that foster interaction, communication, and teamwork among learners. Effective prompt engineering incorporates collaborative elements to facilitate knowledge sharing, collective problem-solving, and enhanced social learning experiences. Collaborative prompts encourage learners to work together, leveraging diverse perspectives and expertise to tackle complex educational challenges.

Prompt designs may simulate group tasks, debates, or peer reviews, requiring students to construct responses or evaluate information collaboratively. Such collaborative learning scenarios deepen students' understanding and help develop essential interpersonal skills crucial for modern professional environments.

3.7. Digital Literacy

Digital literacy involves crafting prompts that enhance students' abilities to critically evaluate, use, and create digital content effectively and responsibly. Prompts to develop digital literacy emphasize critical assessment of online sources, ethical use of digital resources, and practical digital communication skills.

Educational prompts can incorporate scenarios requiring learners to identify credible digital resources, analyze digital content critically, and ethically navigate the digital information landscape. Digital literacy is essential for effective engagement with AI-generated content, underscoring its role in empowering learners to use digital tools proficiently and responsibly.

3.8. Ethical Judgment

Ethical judgment in prompt engineering involves developing prompts that enable learners to consider and evaluate ethical implications within educational contexts. These prompts challenge learners to reflect on moral dilemmas, assess the societal impacts of technology use, and critically evaluate the ethical considerations of information generated by LLMs.

Educational prompts structured around ethical judgment guide learners to explore scenarios involving privacy concerns, fairness, inclusivity, and societal impacts of AI technologies. This component of prompt engineering fosters learners' capacity for ethical reasoning, responsible

decision-making, and awareness of broader implications associated with technology integration in education.

3.9. Integration of Components

Integrating content knowledge, critical thinking, iterative design, clarity and precision, creativity, collaboration, digital literacy, and ethical judgment forms a comprehensive framework for effective educational prompt engineering. Each component uniquely contributes to the effectiveness of prompt design, while their integration creates a synergistic dynamic enhancing overall educational outcomes.

Content knowledge provides the foundational substance for critical thinking, digital literacy, ethical judgment, and creativity, while iterative design systematically refines prompts through continuous feedback and evaluation. Creativity and collaboration enrich learner interactions, fostering engagement, exploration, and deeper cognitive processing. Digital literacy ensures responsible and effective use of digital tools and resources, enhancing the relevance and applicability of educational experiences. Ethical judgment adds a critical dimension, ensuring learners are cognizant of the moral implications and societal impacts of technology use.

This holistic approach aligns with constructivist learning theories, emphasizing students' active role in knowledge construction, collaborative engagement, critical evaluation, creative exploration, and ethical reflection. Effective prompt engineering thus balances these components according to specific educational goals and contexts, creating flexible, adaptive, and robust educational experiences. Ultimately, this integrated framework prepares students comprehensively, equipping them with essential skills for contemporary educational and professional landscapes.

4. RQ3: Strategies on Prompt Engineering in Education

Effective prompt engineering within educational contexts demands carefully developed strategies tailored specifically for educational settings. This section explores strategies educators and students can use to enhance LLMs interactions. Each strategy is discussed comprehensively and supplemented by relevant examples to demonstrate practical implementation. We conducted a comprehensive review of existing literature [1–5,7–11,18–21] and concluded the findings.

4.1. Contextual Framing

Contextual framing involves embedding specific educational contexts directly into prompts. This helps ensure that LLM-generated responses align with learning goals.

By adding context, the output becomes more relevant and precise, which improves its educational usefulness.

For example, instead of broadly prompting the LLM to "explain photosynthesis", an educator might say:

"Imagine you are explaining photosynthesis to a group of elementary students. Provide a simple yet accurate description."

This framing allows the LLM to adjust its response to the intended audience, making it more likely to meet the specific learning objective.

4.2. Task Segmentation

Task segmentation breaks down complex educational tasks into smaller, clearly structured components. This approach enhances response clarity and depth, making tasks more manageable and systematic.

For instance, a historical essay prompt might be segmented into distinct parts:

First prompt:

"List the key factors leading to the American Civil War."

Second prompt:

"Describe the economic impact of the Civil War."

Final prompt:

"Explain the social consequences of the Civil War."

This structured approach ensures that the LLM systematically addresses each task component, promoting deeper cognitive engagement and comprehensive understanding.

4.3. Prompt Sequencing

Prompt sequencing strategically organizes prompts into a coherent, logical progression to guide the LLM through structured reasoning. It enhances logical flow and clarity, especially in tasks involving sequential reasoning or analytical steps.

For example, in a scientific experiment context, an educator might use sequential prompts:

Initial prompt:

"Explain the hypothesis you aim to test."

Next prompt:

"Outline the method you will use to test this hypothesis."

Final prompt:

"Describe how you would analyze the results."

This step-by-step progression explicitly addresses each reasoning stage, ensuring complete and logically connected responses.

4.4. Persona or Role Specification

Specifying personas or roles within prompts guides the LLM in adopting disciplinary-specific language and perspectives. This helps improve the academic rigor of responses by setting clear expectations for tone, depth, and approach. For example, an educator might prompt:

Role-based prompt:

"As a historian specializing in World War II, provide an analysis of the causes of the conflict."

This explicit role designation helps the LLM generate responses that align with professional standards and reflect appropriate disciplinary methodologies.

4.5. Reflection Prompting

Reflection prompts encourage higher-order cognitive processes like analysis, synthesis, and evaluation. These prompts support more profound understanding and critical thinking by prompting students and the LLM to consider broader implications.

Reflective prompt example:

"Provide a solution to reduce plastic waste. Reflect on the potential societal and environmental implications of your solution."

This approach fosters critical engagement with the subject matter and enhances learning outcomes.

4.6. Encouraging Counterfactual Thinking

Counterfactual prompting encourages the exploration of alternative scenarios or outcomes. It supports creativity and deeper analytical thinking by challenging conventional perspectives.

Counterfactual prompt example:

"Describe what might have happened if the internet had been invented in the 19th century."

Such prompts encourage thoughtful consideration of alternative possibilities and their implications.

4.7. Constraint-Based Creativity Prompting

Constraint-based prompting introduces specific limits or conditions to guide the LLM's responses. These constraints help promote innovation and originality within a focused scope.

Constraint-based prompt example:

"Develop a plan to increase urban green spaces using only community-based initiatives."

This prompt encourages creative problem-solving within well-defined boundaries.

4.8. Ethical Consideration Prompts

Ethical prompts explicitly incorporate ethical dimensions into tasks, supporting ethical reflection and reasoning. They encourage students and LLMs to examine broader societal impacts and moral concerns.

Ethical prompt example:

"Discuss the ethical implications of using facial recognition technology in public surveillance."

These prompts enhance ethical awareness and promote thoughtful academic engagement.

4.9. Interactive and Iterative Prompting

Interactive prompting involves using an initial prompt followed by follow-up prompts based on the LLM's responses. This iterative method allows for refinement, clarification, and deeper exploration of ideas.

Initial prompt:

"Explain the concept of democracy."

Follow-up prompt:

"Clarify how democratic principles apply differently in direct versus representative democracy."

This step-by-step interaction supports a progressive understanding of layered concepts.

4.10. Comparative Analysis Prompting

Comparative analysis prompts focus on identifying similarities and differences between concepts, events, or perspectives. They encourage nuanced analysis and structured comparison.

Comparative prompt example:

"Compare and contrast the economic policies of the United States and China over the past two decades."

Such prompts strengthen analytical thinking and promote comprehensive understanding.

Through systematically applying these strategies, educators and students can significantly enhance their interactions with LLMs, fostering meaningful, relevant, and intellectually rigorous educational experiences.

5. RQ4: Challenges on Prompt Engineering in Education

Prompt engineering, a critical component in integrating LLMs in educational contexts, faces several challenges that educators, learners, and developers must consider to leverage these tools for improved learning outcomes effectively. This section identified and discussed the primary challenges associated with prompt engineering, specifically within educational settings, highlighting their implications on learning efficacy and user interaction. We conducted a comprehensive review of existing literature [2,9,10,20,22,24–30] and concluded the findings.

5.1. Ambiguity and Contextual Misinterpretation

One prominent challenge in prompt engineering for education is managing ambiguity inherent in human language. Given natural language's complexity, subtlety, and context-dependency, LLMs can misinterpret prompts, leading to responses that deviate from intended educational outcomes.

For example, a prompt such as *"Explain the process of division"* might result in the model describing mathematical division, cellular division, or social division. This ambiguity requires

prompt engineers—often educators themselves—to craft extremely precise instructions, significantly increasing the cognitive load on educators.

Additionally, contextual misinterpretation arises when prompts do not contain sufficient contextual cues, leading models to produce generic or irrelevant content. Educational prompts must include clear instructional goals, necessary context, and specificity to prevent unintended interpretations, thus ensuring the responses support learning objectives effectively.

5.2. Balancing Specificity and Flexibility

Striking an optimal balance between specificity and flexibility in prompts represents another critical challenge in education. Particular prompts limit responses' creativity and exploratory nature, constraining learners' critical thinking and problem-solving opportunities. Conversely, overly flexible prompts might lead to excessively broad or tangential responses that fail to meet educational objectives or standards.

For instance, a prompt like *"Describe the impact of climate change"* can be interpreted broadly. Refining this to *"Describe the impact of climate change on agricultural productivity in Southeast Asia"* narrows the response sufficiently to maintain educational rigor while allowing exploratory learning.

Educators face the ongoing task of fine-tuning prompts iteratively to maintain this delicate balance. This iterative process requires substantial effort and a deep understanding of subject matter and pedagogical methods, thus adding complexity and time investment to instructional planning.

5.3. Ensuring Response Consistency and Reliability

Consistency and reliability in the responses from LLMs are crucial for maintaining trust and efficacy in educational settings. Inconsistent responses, particularly in sequential interactions, can confuse learners, undermine confidence in educational tools, and disrupt the learning process. For instance, inconsistent tonal shifts or contradictory explanations across interactions could severely impact learner comprehension and engagement.

To address this, educators must implement structured prompt frameworks and incorporate explicit instructions for tone, format, and perspective within the prompts. Additionally, they must consistently validate and calibrate model responses against established educational standards, significantly increasing the preparatory and monitoring workload.

5.4. Managing Model Hallucinations and Accuracy

A notable challenge is managing the phenomenon of hallucinations in LLM outputs, where models generate coherent but factually incorrect or misleading information. This is particularly problematic in education, where the accuracy of content is paramount. Reliance on unverified or erroneous model-generated content could lead students to internalize incorrect information, negatively impacting learning outcomes and understanding.

Therefore, it becomes essential for educators to develop robust validation mechanisms and encourage critical evaluation among learners. This approach involves additional training for educators and students in identifying, cross-checking, and correcting inaccurate outputs from LLMs, further increasing the complexity of effectively integrating these tools into educational settings.

5.5. Ethical and Privacy Concerns

Prompt engineering in education also faces ethical and privacy-related challenges. Educational prompts may inadvertently expose sensitive personal or institutional information, raising concerns about confidentiality and data protection. Moreover, ethical issues may arise when prompts unintentionally reinforce biases or stereotypes embedded in training data, thus perpetuating prejudiced views and inequities.

Addressing these challenges necessitates clear guidelines and rigorous oversight to ensure prompts are ethically sound and respect privacy considerations. Educators must carefully curate

training materials, constantly evaluate model outputs for biases, and engage in ongoing professional development to understand and mitigate potential ethical and privacy implications.

5.6. Student Engagement and Interaction Dynamics

Lastly, maintaining student engagement when using prompt-driven educational methods presents challenges. Prompts that fail to align with students' interests or abilities can reduce motivation and engagement. Moreover, excessive reliance on prompts may lead students to become passive recipients rather than active participants in their learning processes.

To counteract these dynamics, educators must design prompts encouraging active learning, critical thinking, and meaningful interaction. It also involves continuous feedback from students to adjust and improve prompts dynamically, aligning more closely with student needs, preferences, and learning styles.

Effectively addressing these challenges requires concerted efforts from educators, researchers, and technologists to optimize prompt engineering practices, ultimately enhancing educational outcomes and enriching learning experiences.

6. RQ5: Student Perspective on Prompt Engineering in Education

In addition to reviewing the existing literature [2,9,19–21,31–33], we conducted a qualitative study on primary school students' experiences with prompt engineering techniques integrated into their chatbot-based tutoring system (POE chatbot) during a 12-week Python programming course using CodeCombat.

The study involved 30 students aged 9-12, who attended five classes per week, amounting to 60 sessions of 40 minutes each. The decision to focus on primary school students was driven by a significant research gap in this area, as few studies have explored the application of prompt engineering techniques in elementary education. This study aimed to address that gap by investigating how young learners interact with and benefit from these techniques in a programming context.

Before the study began, we obtained consent from students' parents and the school and informed the students about the research, ensuring their willingness to participate. Data were collected through structured and informal interviews to gain insights into students' perceptions of different prompt engineering strategies implemented in their learning environment. A detailed qualitative analysis was performed, and the findings are presented in the following thematic subsections.

6.1. Understanding Prompt Structure and Clarity

Students demonstrated remarkable awareness of how different prompt structures affected their interactions with the POE chatbot. They frequently noted how clear, specific prompts yielded more helpful responses than vague or ambiguous queries. Throughout the program, students gradually developed an intuitive understanding of effective prompt construction, refining their ability to communicate with the AI tutoring system.

S5 reflected on his growing awareness of how prompt construction affected the chatbot's responses:

S5: "I learned that when I ask POE questions, I need to be clear. At first, I would just say 'Help me!' but that didn't work well. Now I say exactly what part of my code I'm stuck on, like 'I don't understand how to use the if-else statement.' Then POE gives me much better help!"

S19 described her discovery of how providing context in prompts significantly improved the quality of assistance she received:

S19: "When I tell POE what I already tried or what I think might be the problem, it helps me much better. I used to just say 'This doesn't work,' but now I explain more details like 'I tried using a loop but my character keeps moving forever.' This makes POE give me exactly what I need."

Students also recognized that structuring prompts with specific learning goals helped them receive more targeted support. They learned to articulate their goals rather than simply describe problems.

S22 explained how he adapted his prompt strategies to achieve better learning outcomes:

S22: "I figured out that asking 'Can you explain how this works?' doesn't help as much as saying 'I want to understand this so I can build my own game later.' When I tell POE what I want to learn, not just what I'm stuck on, the explanations make much more sense to me."

The most significant finding was how students intuitively recognized the relationship between prompt specificity and response quality. Several students noted that providing detailed information about their current understanding level helped calibrate the chatbot's explanations to their needs.

6.2. Contextual Information and Setting Parameters

A crucial prompt engineering strategy that emerged from student interactions was the importance of establishing contextual information. Students discovered that informing the chatbot about the specific CodeCombat level, their age, and their current knowledge significantly improved response relevance and comprehensibility.

S13 explained how providing context about his specific CodeCombat level transformed the quality of assistance:

S13: "I learned to always start by telling POE exactly which CodeCombat level I'm working on. Like, I say 'I'm on the Dungeon level called Fire Dancing' before asking my question. When I do this, POE knows exactly what my character can do and what code I should use. Before, I didn't say which level, and sometimes POE told me to use commands I didn't have yet!"

S26 highlighted the importance of age-appropriate explanations:

S26: "My best trick is telling POE that I'm 10 years old at the beginning. When I do that, it explains things like it's talking to a kid, not a grown-up. It uses simple words and fun examples that I can understand. One time I forgot to say that, and POE used really complicated words that confused me!"

Teachers observed that students who consistently provided contextual information received more precise and age-appropriate guidance. This contextualization became a fundamental prompt engineering technique that students gradually adopted without explicit instruction.

S10 described how setting learning parameters helped him receive more useful explanations:

S10: "I always tell POE what I already know before asking for help. Like I'll say 'I understand what variables are, but I don't get how to change them in my code.' Then POE skips explaining stuff I already know and focuses on exactly what I'm confused about. It saves time and the explanations make more sense!"

Establishing contextual boundaries emerged as a sophisticated metacognitive skill that transferred beyond coding tasks. Students began applying this approach to other learning activities, demonstrating an enhanced awareness of their own knowledge gaps and learning needs.

S30 shared how this skill transferred to other learning contexts:

S30: "The way I talk to POE now helps me ask better questions to my teachers too! I learned to say what I already know and exactly what I'm confused about. My teacher said I ask really good questions now that are easier to answer. It's like I learned a super skill for getting the right help!"

6.3. Role-Based and Scenario Prompting

Students discovered the effectiveness of role-based and scenario prompting techniques where they would frame their interactions with the chatbot in creative contexts. This approach made learning more engaging and helped students receive explanations tailored to their specific needs and interests.

S3 described her excitement about using character-based prompts:

S3: *"It's fun when I ask POE to explain coding like it's a wizard teaching magic spells! I say 'Pretend you're a coding wizard and I'm your apprentice learning the magic spell of loops.' Then POE explains loops like they're magic spells, and it makes everything easier to remember because it's like a story!"*

S16 shared how scenario-based prompts helped him understand complex programming concepts:

S16: *"My favorite way to ask POE questions is making up coding adventures. Like when I couldn't understand variables, I asked POE to explain it like we're on a treasure hunt and variables are different treasure chests for storing things. Now I always remember that variables store things just like treasure chests!"*

Our analysis revealed that role-based prompting had significant pedagogical value beyond mere entertainment. Students who assigned specific teaching roles to the chatbot often received explanations that were more memorable and aligned with their cognitive frameworks.

S8 explained how assigning a specific teaching role to POE enhanced his comprehension:

S8: *"When I was stuck on functions, I asked POE to 'be like my football coach teaching a new play.' POE explained functions like they were football strategies with different players having different jobs, and suddenly it all made sense! Now whenever I write functions, I think about my football team and remember exactly how they work."*

Teachers noted that role-based prompting was particularly effective for students who struggled with traditional abstract explanations. Students could anchor new knowledge to existing mental models by framing coding concepts within familiar scenarios.

6.4. Constraint Specification and Error Prevention

Students developed an impressive understanding of how specifying constraints in their prompts could prevent misleading or incorrect responses. They learned to explicitly inform the chatbot about what not to do, establishing boundaries that significantly improved response accuracy and educational value.

S9 explained his discovery about setting explicit constraints:

S9: *"I learned an important trick – telling POE what NOT to do! When POE gave me answers that were too complicated, I started saying 'Please explain this simply without using any big coding words.' This worked really well! Now I always tell POE exactly how I want the answer and what I don't want."*

S25 shared how specifying constraints helped her receive more appropriate guidance:

S25: *"At first, POE sometimes gave me complete solutions that didn't help me learn. Then I started saying 'Please don't give me the full answer, just give me hints so I can figure it out myself.' Now POE gives me just enough help to learn without doing all the work for me. I feel prouder when I solve problems this way!"*

The ability to establish constraints emerged as a sophisticated prompt engineering skill that had significant implications for fostering independent learning. Students who effectively communicated boundaries received guidance that supported their development rather than circumventing the learning process.

S14 described how constraint-setting helped balance assistance with learning:

S14: *"I figured out that I can tell POE exactly how much help I want. When I say 'Don't solve the whole problem for me, just help me understand why my loop isn't working,' POE gives me the perfect amount of help. It points out my mistake but lets me fix it myself. This helps me learn much better than when POE just gives me the answer."*

Teachers observed that students who mastered constraint specification demonstrated greater metacognitive awareness and learning autonomy. These students were better able to identify precisely what assistance they needed and how that assistance should be delivered.

S28 shared his sophisticated approach to specifying ethical constraints:

S28: "I noticed that sometimes POE would show me a shortcut that wasn't what our teacher wanted us to learn. So now I say 'Please help me solve this using only the commands we've learned in class, and don't show me any shortcuts.' This makes sure I'm learning the right way and not taking the easy way out."

6.5. Prompt Templates and Structure Recognition

Throughout the 12-week program, students began to recognize patterns in effective prompts and developed templates for different coding challenges. This development of prompt templates represented an important metacognitive skill that transferred to other problem-solving contexts.

S8 proudly described the prompt template he developed for debugging assistance:

S8: "I made my own special way to ask for help when my code has bugs. First, I tell POE what my code should do, then I paste my code, then I explain what's happening instead, and finally I ask what's wrong. POE almost always finds my mistake right away when I follow this pattern!"

S20 explained how she created different prompt templates for different learning needs:

S20: "I have different question patterns for when I need different help. If I want to understand something new, I start with 'Can you explain specific concept like I'm 9 years old with examples?' If I'm fixing code, I use 'Here's my code, here's what it should do, here's what's happening instead.' Having these patterns helps me get better answers from POE."

Further analysis revealed that students naturally evolved toward consistent template usage over the course of the study. By the final weeks, many students had developed sophisticated, multi-part templates that demonstrated a nuanced understanding of effective prompt construction.

S11 described her systematic template for learning new coding concepts:

S11: "I created my own four-step question formula for learning new things! Step 1: I ask 'What is [concept] in super simple words?' Step 2: I ask 'Can you show me a really easy example of [concept]?' Step 3: I ask 'How is [concept] used in games like CodeCombat?' Step 4: I ask 'What mistakes do people usually make with [concept]?' This pattern helps me understand everything completely!"

S29 shared how he developed templates based on response effectiveness:

S29: "I keep track of which questions get the best answers from POE in my notebook. When POE gives me a really good explanation, I write down exactly how I asked the question. Now I have a collection of perfect questions for different coding problems. It's like I discovered the secret codes for talking to POE!"

Teachers noted that prompt templating represented a sophisticated metacognitive strategy, as it required students to reflect on the communication process itself. Students who developed effective templates demonstrated enhanced problem-solving abilities and communication skills.

6.6. Iterative Prompt Refinement and Response Evaluation

As students progressed through the program, they began to employ sophisticated iterative prompt refinement strategies. Rather than accepting inadequate responses, students learned to systematically improve their prompts based on the quality of responses received, engaging in a collaborative dialogue with the chatbot that progressively approached optimal explanations.

S2 described her discovery of iterative prompt refinement:

S2: "At first, I would give up if POE's answer didn't help me. Then I realized I could keep asking better questions! Now if POE's explanation is confusing, I say 'I didn't understand that part about loops. Can you explain it differently using a game example?' I keep asking follow-up questions until I really understand."

S18 shared his systematic approach to refining prompts based on response evaluation:

S18: *"I learned to judge POE's answers in my head – was it too hard, too easy, or just right? If it's too complicated, I say 'Can you explain that like I'm younger?' If it's too simple, I say, 'I understand that part, but can you tell me more about specific details?' It's like training POE to give me perfect answers!"*

The development of response evaluation skills represented a sophisticated metacognitive capability that enhanced students' critical thinking. Students became more discerning about the quality and relevance of information, learning to identify gaps in explanations and request specific clarifications.

S6 articulated his process for evaluating and refining responses:

S6: *"I've become good at spotting when POE's explanation isn't quite right for me. Sometimes POE explains something using math examples, but I understand sports better. So I say 'That makes sense, but could you re-explain it using basketball instead of numbers?' Then the explanation suddenly clicks in my brain!"*

S23 described how she learned to identify specific knowledge gaps in responses:

S23: *"I noticed that sometimes POE assumes I know things that I don't. Now I'm not embarrassed to say 'Wait, I don't know what a parameter is yet. Can you explain that first?' I've learned it's OK to admit when I don't understand something and ask for more basic explanations before moving to the complicated stuff."*

Teachers observed that students who mastered iterative refinement demonstrated enhanced persistence when facing challenging concepts. These students were less likely to abandon complex problems and more likely to systematically work toward understanding through strategic dialogue with the chatbot.

S27 explained how iterative prompting improved her persistence and problem-solving:

S27: *"Before, if I didn't understand something right away, I would just give up and say coding is too hard. Now I know that if POE's first explanation doesn't help, I can try again with a better question. Sometimes it takes five or six tries, but eventually I always understand! This has made me more determined in everything I do – I don't give up easily anymore."*

6.7. Specific vs. General Prompt Strategies

Students developed a nuanced understanding of when to use specific versus general prompts depending on their learning objectives. They discovered that highly detailed prompts yielded precise solutions to immediate problems, while more open-ended prompts facilitated broader conceptual understanding and creative exploration.

S1 explained his strategic use of specific prompts for debugging:

S1: *"When my code is broken and I need to fix it quickly, I've learned to be super specific with POE. I say exactly what line is causing problems, what I expected to happen, and what's happening instead. The more details I give, the faster POE finds my exact mistake!"*

S12 contrasted this with her approach for concept exploration:

S12: *"When I want to really understand something new, I ask broader questions like 'What are different ways to use loops in games?' or 'What can I create with functions?' These questions make POE show me lots of cool possibilities I wouldn't have thought of myself. It's like opening a treasure chest of coding ideas!"*

This strategic alternation between specific and general prompts demonstrated sophisticated metacognitive awareness, as students learned to match their prompting strategies to different learning needs and contexts.

S19 shared his contextual approach to prompt specificity:

S19: *"I use different question styles for different situations. When I'm stuck and frustrated, I ask very specific questions to get unstuck quickly. But when I'm feeling curious and have extra time, I*

ask general questions to discover new coding tricks. Both ways are helpful, just for different times and goals."

Teachers noted that students who mastered this contextual approach demonstrated enhanced learning efficiency, appropriately balancing immediate problem-solving with deeper conceptual exploration.

S4 described how varying prompt specificity enhanced his learning strategy:

S4: "I've learned to use specific questions like tools for fixing problems and general questions like adventures for exploring. If I need to complete a level quickly, I ask exactly where my mistake is. But after finishing my work, I ask curious questions like 'What are cool things I could add to this code?' This helps me learn both the 'how' and the 'why' of coding."

The ability to strategically vary prompt specificity correlated with students' developing a more sophisticated and flexible approach to learning programming concepts.

6.8. Challenges and Limitations in Prompt Engineering

Despite their impressive developments in prompt engineering skills, students encountered several challenges and limitations that provided valuable insights for future educational implementations. These challenges highlight important considerations for integrating prompt engineering education into primary school curricula.

S7 expressed frustration with occasional mismatches between prompts and responses:

S7: "Sometimes even when I ask a really good question, POE still doesn't understand what I mean. I try using all the tricks I learned, but POE gives me information about something different. It's frustrating because I don't know how to ask better."

S14 highlighted difficulties with technical vocabulary limitations:

S14: "The hardest part is when I don't know the right coding words to use in my question. If I don't call something by its proper name, like 'variable' or 'parameter,' sometimes POE doesn't understand what I'm asking about. It's like we speak different languages sometimes."

Some students struggled with the metacognitive demands of prompt engineering, finding it challenging to simultaneously think about coding problems and how to communicate about those problems effectively.

S26 described this metacognitive challenge:

S26: "It's hard to think about coding AND think about how to ask good questions at the same time. Sometimes my brain gets too full trying to do both things! When I'm really stuck on a hard problem, it's difficult to remember all the question tricks I learned."

Teachers observed that younger students (ages 9-10) generally found prompt engineering more challenging than older students (ages 11-12), suggesting developmental factors in metacognitive capabilities relevant to effective prompt construction.

S5, one of the younger participants, articulated this developmental challenge:

S5: "The older kids are better at asking POE perfect questions. I try to copy how they ask questions, but sometimes I forget all the parts to include. My teacher says my question-asking skills will get better as I practice more and my brain grows."

Despite these challenges, students demonstrated remarkable resilience and adaptation. By the end of the program, even students who initially struggled had developed basic prompt engineering strategies that noticeably improved their learning interactions.

S10 reflected on his progress despite ongoing challenges:

S10: "At first, I was really bad at asking POE questions. My questions were too short, and POE's answers didn't help me. I'm still not the best at it, but I'm much better now! I learned to be specific, give examples, and keep trying different questions. Even when it's hard, I don't give up because I know good questions lead to good answers."

These challenges suggest important considerations for scaffolding prompt engineering education for young learners, including developmentally appropriate instruction, explicit vocabulary support, and gradual release of responsibility as students develop metacognitive capabilities.

7. Conclusion

This comprehensive review explored the multifaceted landscape of prompt engineering for LLMs in educational contexts. Our investigation identified a rich taxonomy of prompt engineering techniques applicable to educational settings, ranging from foundational approaches like zero-shot and few-shot prompting to more advanced methods and their derivatives, facilitating structured reasoning and problem-solving. Furthermore, we delineated key components constituting effective prompt engineering in education, including content knowledge, critical thinking, iterative design, clarity, creativity, collaboration, digital literacy, and ethical judgment—all synergistically contributing to enhanced educational outcomes.

The strategies examined in this study provided educators and students with practical frameworks for optimizing LLM interactions, including contextual framing, task segmentation, prompt sequencing, persona specification, reflection prompting, counterfactual thinking, constraint-based creativity, ethical consideration prompts, interactive prompting, and comparative analysis. However, significant challenges persisted in implementing prompt engineering in educational settings, including ambiguity and contextual misinterpretation, balancing specificity with flexibility, ensuring response consistency, managing model hallucinations, addressing ethical concerns, and maintaining student engagement.

Notably, our qualitative study with primary school students revealed remarkable insights into students' perspectives and experiences with prompt engineering techniques. Students demonstrated impressive intuitive development of sophisticated prompt engineering skills, including understanding the prompt structure, utilizing contextual information, employing role-based prompting, specifying constraints, developing prompt templates, engaging in iterative refinement, and strategically varying prompt specificity according to learning objectives. These findings suggested that even young learners could develop metacognitive awareness of effective communication strategies with AI systems, though developmental factors influenced their capacity to implement these strategies consistently.

This research contributed to the nascent field of AI literacy in education by highlighting the importance of explicit instruction in prompt engineering as a fundamental skill for the AI-augmented classroom. Future research should focus on developing age-appropriate pedagogical frameworks for teaching prompt engineering, investigating the long-term impact of prompt engineering skills on learning outcomes across disciplines, and exploring how prompt engineering competencies might transfer to other metacognitive domains. As LLMs become increasingly integrated into educational environments, the ability to effectively engineer prompts emerges not merely as a technical skill but as a critical component of digital literacy essential for 21st-century learning.

References

1. J. White et al., 'A prompt pattern catalog to enhance prompt engineering with chatgpt', *arXiv preprint arXiv:2302.11382*, 2023.
2. D. Federiakin, D. Molerov, O. Zlatkin-Troitschanskaia, and A. Maur, 'Prompt engineering as a new 21st century skill', in *Frontiers in Education*, Frontiers Media SA, 2024, p. 1366434.
3. S. Ekin, 'Prompt engineering for ChatGPT: a quick guide to techniques, tips, and best practices', *Authorea Preprints*, 2023.
4. S. Arvidsson and J. Axell, 'Prompt engineering guidelines for LLMs in Requirements Engineering', 2023.
5. G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende, 'Prompt engineering in large language models', in *International conference on data intelligence and cognitive informatics*, Springer, 2023, pp. 387–402.

6. H. Naveed et al., 'A Comprehensive Overview of Large Language Models', *arXiv preprint arXiv:2307.06435*, 2024.
7. T. Debnath, M. N. A. Siddiky, M. E. Rahman, P. Das, and A. K. Guha, 'A Comprehensive Survey of Prompt Engineering Techniques in Large Language Models', *TechRxiv*, 2025, doi: 10.36227/techrxiv.174140719.96375390.
8. P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, 'A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications', *arXiv preprint arXiv:2402.07927*, 2024.
9. W. Cain, 'Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education', *TechTrends*, vol. 68, pp. 47–57, 2024, doi: 10.1007/s11528-023-00896-0.
10. V. Geroimenko, *The Essential Guide to Prompt Engineering: Key Principles, Techniques, Challenges, and Security Risks*. in SpringerBriefs in Computer Science. Springer, 2025. doi: 10.1007/978-3-031-86206-9.
11. G. Beri and V. Srivastava, 'Advanced Techniques in Prompt Engineering for Large Language Models: A Comprehensive Study', in *2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG)*, IEEE, 2024, pp. 1–4.
12. W. C. Choi and C. I. Chang, 'Advantages and Limitations of Open-Source versus Commercial Large Language Models (LLMs): A Comparative Study of DeepSeek and OpenAI's ChatGPT', 2025, *Preprints.org*.
13. H. Hu, H. Lu, H. Zhang, Y.-Z. Song, W. Lam, and Y. Zhang, 'Chain-of-symbol prompting elicits planning in large language models', *arXiv preprint arXiv:2305.10276*, 2023.
14. Z. Zhang, A. Zhang, M. Li, and A. Smola, 'Automatic chain of thought prompting in large language models', *arXiv preprint arXiv:2210.03493*, 2022.
15. J. Weston and S. Sukhbaatar, 'System 2 attention (is something you might need too)', *arXiv preprint arXiv:2311.11829*, 2023.
16. S. Vatsal and H. Dubey, 'A survey of prompt engineering methods in large language models for different nlp tasks', *arXiv preprint arXiv:2407.12994*, 2024.
17. S. Schulhoff et al., 'The prompt report: A systematic survey of prompting techniques', *arXiv preprint arXiv:2406.06608*, 2024.
18. B. Chen, Z. Zhang, N. Langrené, and S. Zhu, 'Unleashing the potential of prompt engineering in large language models: a comprehensive review', *arXiv preprint arXiv:2310.14735*, 2023.
19. D. J. Woo, D. Wang, T. Yung, and K. Guo, 'Effects of a Prompt Engineering Intervention on Undergraduate Students' AI Self-Efficacy, AI Knowledge and Prompt Engineering Ability: A Mixed Methods Study', *arXiv preprint arXiv:2408.07302*, 2024.
20. Y. Walter, 'Embracing the future of Artificial Intelligence in the classroom: the relevance of AI literacy, prompt engineering, and critical thinking in modern education', *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, p. 15, 2024.
21. T. Wang, N. Zhou, and Z. Chen, 'Enhancing computer programming education with llms: A study on effective prompt engineering for python code generation', *arXiv preprint arXiv:2407.05437*, 2024.
22. R. Patil and V. Gudivada, 'A review of current trends, techniques, and challenges in large language models (llms)', *Applied Sciences*, vol. 14, no. 5, p. 2074, 2024.
23. E. Chen, D. Wang, L. Xu, C. Cao, X. Fang, and J. Lin, 'A Systematic Review on Prompt Engineering in Large Language Models for K-12 STEM Education', *arXiv preprint arXiv:2410.11123*, 2024.
24. C. R. Mohan, R. N. Sathvik, C. Kushal, S. Kiran, and A. Ashok Kumar, 'Exploring the Future of Prompt Engineering in Healthcare: Mission and Vision, Methods, Opportunities, Challenges, Issues and Their

- Remedies, Contributions, Advantages, Disadvantages, Applications, and Algorithms', *Journal of The Institution of Engineers (India): Series B*, pp. 1–24, 2024.
25. C. Schumacher and D. Ifenthaler, 'Investigating prompts for supporting students' self-regulation—A remaining challenge for learning analytics approaches?', *The Internet and higher education*, vol. 49, p. 100791, 2021.
 26. V. Geroimenko, 'Key Challenges in Prompt Engineering', in *The Essential Guide to Prompt Engineering: Key Principles, Techniques, Challenges, and Security Risks*, Springer, 2025, pp. 85–102.
 27. Z. Han and F. Battaglia, 'Transforming challenges into opportunities: Leveraging ChatGPT's limitations for active learning and prompt engineering skill', *The Innovation*, 2024.
 28. W. C. Choi and C. I. Chang, 'A Survey of Techniques, Design, Applications, Challenges, and Student Perspective of Chatbot-Based Learning Tutoring System Supporting Students to Learn in Education', 2025, *Preprints.org*.
 29. C. I. Chang, W. C. Choi, and I. C. Choi, 'Challenges and Limitations of Using Artificial Intelligence Generated Content (AIGC) with ChatGPT in Programming Curriculum: A Systematic Literature Review', in *Proceedings of the 2024 7th Artificial Intelligence and Cloud Computing Conference*, 2024.
 30. H. Subramonyam, R. Pea, C. Pondoc, M. Agrawala, and C. Seifert, 'Bridging the gulf of envisioning: Cognitive challenges in prompt based interactions with LLMs', in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–19.
 31. W. C. Choi, I. C. Choi, and C. I. Chang, 'The Impact of Artificial Intelligence on Education: The Applications, Advantages, Challenges and Researchers' Perspective', *Preprints.org*, 2025, doi: 10.20944/preprints202501.1420.v1.
 32. C. I. Chang, W. C. Choi, and I. C. Choi, 'A Systematic Literature Review of the Opportunities and Advantages for AIGC (OpenAI ChatGPT, Copilot, Codex) in Programming Course', in *Proceedings of the 2024 7th International Conference on Big Data and Education*, 2024.
 33. W. C. Choi, I. C. Choi, C. I. Chang, and L. C. Lam, 'Comparison of Claude (Sonnet and Opus) and ChatGPT (GPT-4, GPT-4o, GPT-o1) in Analyzing Educational Image-based Questions from Block-Based Programming Assessments', in *2025 14th International Conference on Information and Education Technology (ICIET)*, IEEE, 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.