

Review

Not peer-reviewed version

---

# A Review of Generative Pretrained Multi-step Prompting Schemes –and a New Multi-step Prompting Framework

---

[Yuval Cohen](#) \* and [Yehudit Aperstein](#)

Posted Date: 13 May 2024

doi: 10.20944/preprints202405.0720.v1

Keywords: Iterative prompting, sequential prompting, recursive prompting, prompt engineering, adaptive prompting, chain of thought



Preprints.org is a free multidiscipline 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 is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Review

# A Review of Generative Pretrained Multi-Step Prompting Schemes –and a New Multi-Step Prompting Framework

Yuval Cohen \* and Yehudit Aperstein

Afeke Tel-Aviv College of Engineering, Tel-Aviv, Israel

\* Correspondence: yuvalc@afeke.ac.il

**Abstract:** This paper reviews the pioneering work related to the generation of a sequence of prompts in Pretrained Language Models (PLMs) to automatically complete a complex writing task. The paper identifies the main categories of automated multi-step prompting and discusses their advantages and disadvantages. The paper also proposes an additional approach for automated multi-step prompting with iterative four main phases: in the first phase a sequence of prompts is geared to elicit the structure of the requested task. This is done by identifying the task's main parts, the general contents of the different parts and their precedence relations. The structure of each main part is iteratively explored to find secondary and tertiary parts if they exist. A second phase follows by a sequence of prompts that fill in the primary, secondary, and tertiary parts, and seek to fine tune the result. Third phase reviews the generated text and criticizes it. Based on this criticism a revision is prepared. The fourth phase is fine tuning the document, with emphasis on sections length and text tone. Future research could deal with the system identification of points where the system should initiate interaction with the user for desired feedback and guidance.

**Keywords:** iterative prompting; sequential prompting; recursive prompting; prompt engineering; adaptive prompting; chain of thought

## 1. Introduction

Pretrained language models (PLMs) have been shown to struggle with multi-step reasoning tasks that require complex reasoning abilities. To address this limitation, researchers have proposed iterative prompting frameworks that progressively elicit relevant knowledge from PLMs for multi-step inference (Wang et al. 2022). These frameworks aim to equip PLMs with the ability to develop a "chain of thought" similar to humans when solving such tasks. Additionally, there is a focus on context-aware prompting, where prompts are dynamically synthesized based on the current step's contexts (Lu et al. 2022). Another approach involves using random walks over structured knowledge graphs to guide PLMs in composing multiple facts for multi-hop reasoning in question-answering tasks (Lu et al. 2022). These techniques have shown improvements in addressing the multi-step reasoning requirement in pretrained models.

While this paper reviews multiple approaches for sequential automated prompting, automatic self-refinement models are of special interest to our topic. Madaan et al. (2024) developed the SELF-REFINE model that relies on three consecutive prompts (initial generation (INIT), feedback (FEEDBACK), and refinement (REFINE)) on a suitable language model. few-shot prompting, feedback generation, and refinement, introspective feedback, and three components (INIT, FEEDBACK, and REFINE) for iterative refinement. It uses the same underlying language model to generate feedback and refine its outputs. The feedback is generated by the model itself based on the initial output, and the refinement is performed by the same model using the feedback. This iterative process continues until a desired condition is met. SELF-REFINE does not require any additional training or supervision for the refinement process.

Another especially relevant prompting scheme was presented by Kim et al. (2024) and called Recursive Criticism and Improvement (RCI). In this scheme after the initial prompt, the following two steps are recursively repeated until specific conditions are met.

- 1) *Criticism*: In the criticism step, the LLM generates an initial output based on the given prompt. The LLM is then prompted to identify problems or errors in its output. This is done by asking the LLM to review its previous answer and find problems with it. The LLM is encouraged to critically evaluate its own output and identify any issues.
- 2) *Improvement*: After the LLM has identified problems with its output, the improvement step follows. The LLM is prompted to generate an updated output based on the critique it provided. The LLM is asked to improve its answer based on the problems it found in the previous step. This prompts the LLM to refine its output and generate a better solution.

The RCI process is applied iteratively until termination conditions are met, such as reaching a maximum number of iterations, or receiving feedback from the environment, or reaching a low threshold of refinement intensity (Kim et al. 2024)). The goal of RCI is to enhance the reasoning abilities of LLMs and improve their performance in executing computer tasks. By incorporating the criticism and improvement steps, RCI helps the LLMs identify and rectify errors in their output, leading to more accurate and effective solutions.

Peng et al. (2019) presented early research on sequential prompting, suggesting that each prompt should use information of previous answers on prompts to explore and fill-in missing data.

A series of studies have explored the use of iterative prompting in various contexts. Wang et al. (2022) and Sun et al. (2023) both propose iterative prompting frameworks for enhancing the performance of pre-trained language models in multi-step reasoning tasks, and open-domain question answering, respectively. These frameworks involve dynamically synthesizing prompts based on the current context and have been shown to be effective in capturing variabilities across different inference steps.

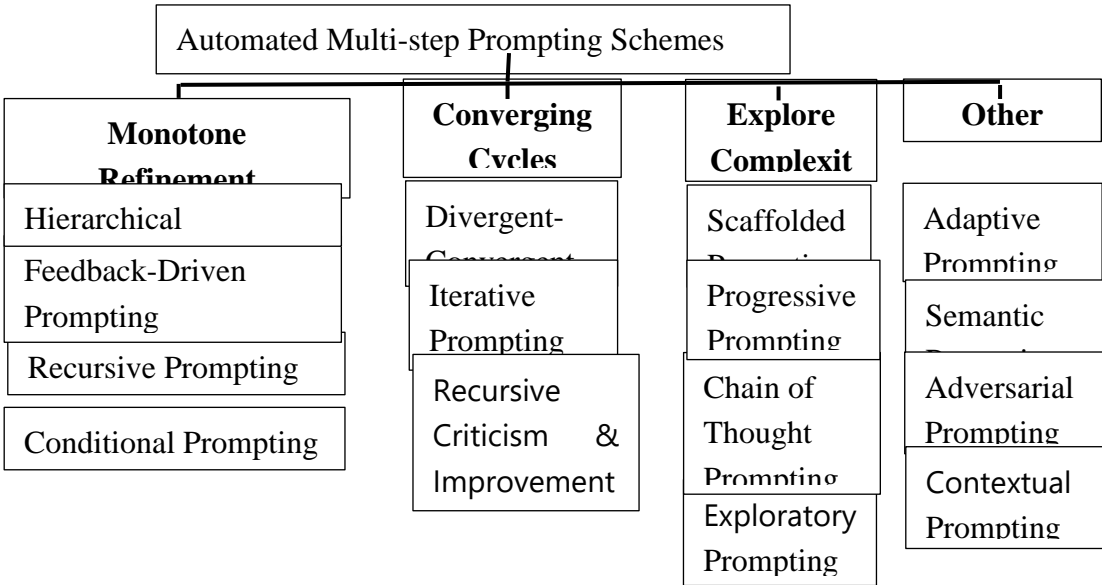
Fu (2022) introduced complexity-based prompting for multi-step reasoning, showing that prompts with higher reasoning complexity can significantly improve performance on multi-step reasoning tasks. Firdaus (2023) applied a two-step prompting method to generate empathetic responses in conversational systems, outperforming existing approaches. Tan (2021) proposed a multi-stage prompting approach for translation tasks, which significantly improved the translation performance of pre-trained language models. Liu (2022) further extended this concept to knowledge-grounded dialogue generation,

To fully grasp the significance of these and competing methods a comprehensive literature review is necessary. In the next section we explore literature related to the place of these methods in the realm of automated sequential multi-step prompting. We categorize the main approaches and discuss their effectiveness and suitability for various purposes.

Finally, we propose an additional new approach that works nicely with writing tasks that have a large number of close or similar examples in the LLM. The paper concludes with discussion on future research in close subjects.

## 2. Literature Review

This section is arranged according to the identified structure of main types of multi-step prompting schemes used to guide the generation process. The proposed arrangement is depicted in Figure 1.



**Figure 1.** Main types of multi-step LLM prompting schemes used to guide the generation process.

2.1. Monotone Refinement Prompting

*Monotone Refinement* refers to prompting schemes that progressively refine the prompts and the generated answers (Madaan et al. 2024). These techniques are characterized by a process of improving answers, so that each answer is better than its predecessors. The following are these schemes (Chen, Guo, Haddow, & Heafield, 2023).

2.1.1. Hierarchical Prompting

Hierarchical prompting involves structuring prompts into hierarchical levels, with each level providing increasing specificity or granularity (Lo et al. 2023). The model first receives a high-level prompt guiding the overall direction or theme of the generation (Li et al. 2023). Subsequent prompts at lower levels provide more detailed instructions or constraints, guiding the model towards finer-grained output (Sodhi et al. 2023, Li et al. 2023). Typical applications include Vision-Language Models (VLMs) (Liu et al. 2023; Wang et al. 2024) Image classification (Wang et al. 2024) web navigation (Lo et al. 2023; Sridhar et al. 2023; Sodhi et al. 2023) This approach allows for a balance between providing broad guidance and fine-tuning the model's output to meet specific criteria.

2.1.2. Feedback-Driven Prompting

Feedback-driven prompting involves incorporating feedback loops into the generation process, where the model's output is evaluated and used to inform subsequent prompts (Huang et al. 2023). Feedback can be provided by an answer to an automated evaluation prompt, or automated metrics, or reinforcement learning algorithms (Scarlatos et al. 2024), guiding the model towards desired outputs. This approach enables iterative refinement of the model's output based on close to real-time feedback, leading to continuous improvement over time (Madaan et al. 2024).

2.1.3. Recursive Prompting

Recursive prompting involves the generation of prompts that incorporate recursive elements, guiding the model through iterative cycles of self-reflection or refinement (Liu et al. 2023). In this scheme, prompts may ask the model to revisit and build upon its own output or previous responses in a recursive manner (Qi et al. 2023). The process resembles a feedback loop, where the model's own output becomes input for subsequent prompts, leading to deeper exploration or elaboration (Qi et al. 2023). Recursive prompting is particularly useful for tasks that require iterative problem-solving,

creative exploration, or self-referential reasoning. By incorporating recursive elements, this technique enables the model to revisit and refine its output iteratively. Recursive prompting was also used for image analysis (Batten et al. 2023), populating knowledge bases (Caufield et al. 2023)

#### 2.1.4. Conditional Prompting

Conditional prompting involves conditioning the prompts on specific attributes, contexts, or criteria to guide the generation process. Prompts may include conditional statements or constraints that influence the model's output based on predefined conditions. This technique allows for fine-grained control over the generated content, ensuring that it aligns with desired attributes or characteristics (Li et al. 2024).

#### 2.2. Converging Prompting Cycles

*Converging Cycles* refers to prompting schemes that iterate on cycles of prompts. While the stages in a cycle may have steps that do not refine or converge, each cycle leads closer toward convergence to the ultimate generated response.

##### 2.2.1. Divergent-Convergent Prompting

Divergent-convergent prompting involves guiding the model through alternating phases of divergent exploration and convergent refinement (Sim et al. 2024). During the divergent phase, the model is encouraged to explore a wide range of possibilities or ideas, generating diverse output. In the convergent phase, the focus shifts towards narrowing down the options and refining the output towards a specific goal or criteria (Tian et al. 2023). This approach balances exploration and exploitation, allowing the model to generate creative and varied output while also honing in on desired outcomes (Suh et al. 2023).

##### 2.2.2. Iterative Prompting

Iterative prompting involves providing a series of prompts in a cyclical manner, refining the generated output with each iteration (Skreta, et al. 2023). After receiving an initial prompt, the model generates a response. The response is then evaluated, and feedback is provided as a new prompt to guide the subsequent iteration. This process continues until the desired output quality is achieved, or a stopping criterion is met (Jha et al. 2023).

##### 2.2.3. Recursive Criticism & Improvement

In this scheme after the initial prompt, cycles of two steps are recursively repeated until specific conditions are met. The steps are: (1) *Criticism*: (2) *Improvement* (Zhang & Dong et al. 2023).

#### 2.3. Prompting That Explore Complexity

*Prompting that Develops Details* refers to top-down prompting schemes that explore more and more complexity as they progress.

##### 2.3.1. Scaffolded Prompting

Scaffolded prompting involves structuring prompts in a sequential manner, with each prompt building upon the previous one to guide the generation process. Prompts are designed to provide increasing levels of support or complexity as the generation progresses. By scaffolding the prompts, the model is guided through a step-by-step process that facilitates learning or problem-solving (Lesmono & Astutik, 2023); (Suzgun & Kalai, 2024).



### 2.3.2. Progressive Prompting

Progressive prompting involves presenting prompts in a sequence that gradually increases in complexity or difficulty (Wu, Jiang, & Shen, 2024). The model starts with simple prompts and progresses to more challenging ones as it becomes more proficient or as the task requires (Zheng et al. 2023). Progressive prompting allows the model to incrementally build upon its knowledge and skills, leading to more sophisticated output (Razdaibiedina et al. 2023).

### 2.3.3. Chain of Thought Prompting

Chain of thought prompting guides the generation process by presenting a series of interconnected prompts that lead to a coherent narrative or solution (Hongru et al. 2023). Each prompt builds upon the previous one, encouraging the model to explore related ideas or concepts (Yu et al. 2023). By following a chain of thought, the model is guided towards generating output that is logically structured and coherent (Madaan et al. 2023).

### 2.3.4. Exploratory Prompting

Exploratory prompting encourages the model to explore different avenues or possibilities during the generation process (Parra Pennefather, 2023). Rather than providing strict instructions, prompts are designed to inspire curiosity and creativity, prompting the model to generate diverse and novel output (Armitage et al. 2023). This approach fosters exploration and experimentation, allowing the model to discover new ideas or perspectives through the generation process (Papakyriakopoulos & Mboya, 2023; Yan Zheng, & Wang, 2024). In educational settings, exploratory prompting has been shown to enhance skills like perspective taking and conflict negotiation (Sihite et al. 2023). Similarly, in the context of large language models (LLMs), prompting is crucial for evaluating linguistic knowledge (Ma et al. 2023).

## 2.4. Other Prompting Schemes

### 2.4.1. Adaptive Prompting

Adaptive prompting involves dynamically adjusting prompts based on the model's responses or performance (Kim et al. 2023). Feedback from the model's output or intermediate states is used to modify subsequent prompts, guiding the generation process in real-time (Shi; et al. 2023 Wu et al. 2024). Adaptive prompting enables the prompter to adapt to the model's behavior and steer it towards desired outcomes. This prompting process typically ends after few iterations (Zhou et al. 2023; Wu et al. 2024). Also, self-adaptive prompting may converge in one shot (Wan et al. 2023).

### 2.4.2. Semantic Prompting

Semantic prompting involves guiding the model's generation process by focusing on the semantic meaning or intent of the desired output (Gong et al 2023; Tanwar et al 2023). Instead of providing explicit instructions or constraints, prompts are formulated to convey the underlying meaning or concept that the model should capture in its output (Ma et al. 2023). This approach encourages the model to prioritize semantic coherence and relevance, leading to output that aligns with the intended meaning or purpose. Some implementations of Semantic Prompting were reported in: Multilingual LLMs (Tanwar et al 2023) Cross domain ((Gong et al 2023) Image creation (Han et al. 2024) and Chain of Thought (Ma et al. 2023).

### 2.4.3. Adversarial Prompting

Adversarial prompting involves challenging the model with prompts designed to push its boundaries and expose weaknesses or biases (Jin et al. 2023). These prompts may include adversarial examples, edge cases, or counterfactual scenarios that test the robustness and generalization capabilities of the model (Ciucă et al. 2023; Yang et al. 2024). By confronting the model with

adversarial prompts, weaknesses can be identified and addressed, leading to improvements in its performance and reliability (Chen et al. 2023).

#### 2.4.4. Contextual Prompting

Contextual prompting involves tailoring prompts to specific contexts or situations to guide the generation process effectively. Prompts are formulated to take into account the current context, such as previous interactions, user preferences, or environmental factors. By considering the context, prompts can provide more relevant and personalized guidance to the model, enhancing the quality of the generated output. Some applications of Contextual Prompting were reported in medical domain (Zhang et al. 2023), recommender systems (Chen & Sun, 2023), relation extraction (Chen et al. 2024) task-oriented dialogue (Swamy et al. 2023), and general conversation (Feldt et al. 2023).

### 3. The Proposed Framework

In this section, we present our proposed framework for automated sequential multi-step prompting, comprising four phases: (1) Structure Elicitation, (2) Content Generation, (3) Revision, and (4) Refinement. Each phase is designed to iteratively guide the Pretrained Language Model (PLM) through the process of completing a complex writing task. The framework is intended to enhance the PLM's ability to generate coherent and structured outputs by systematically eliciting, generating, revising, and refining content.

#### 3.1. Structure Elicitation Phase

The primary objective of the Structure Elicitation phase is to discern the essential components of the writing task and assign them appropriate titles. This phase can be likened to requirement engineering in software development, where the focus is on defining the fundamental structure of the document. Leveraging the Recursive Criticism and Improvement (RCI) process (Kim et al., 2024), this phase ensures the creation of a comprehensive and well-organized contents list.

#### 3.2. Content Generation Phase

The Content Generation phase aims to populate each identified subsection and section with relevant content. Building upon the contents list generated in the Structure Elicitation phase, prompts are designed to guide the PLM in writing the document. These prompts are tailored to solicit coherent and contextually appropriate content from the PLM, ensuring alignment with the predefined structure.

To facilitate this phase, prompts are strategically crafted to elicit responses from the PLM that align with the predefined structure and desired content. By providing clear and contextual instructions, the prompts guide the PLM through the process of generating content for each section and subsection. Additionally, the iterative nature of the framework allows for refinement and enhancement of the generated content over multiple iterations.

The Content Generation phase involves a dynamic interplay between the PLM and the prompting system. As the PLM generates content based on the provided prompts, the system evaluates the outputs and provides feedback to refine subsequent prompts. This feedback loop ensures continuous improvement in the quality and coherence of the generated content.

Furthermore, the Content Generation phase is characterized by flexibility and adaptability. The prompting system can dynamically adjust prompts based on the PLM's responses, allowing for real-time adaptation to changing requirements or preferences. This adaptability ensures that the generated content remains relevant and accurate throughout the writing process.

Overall, the Content Generation phase serves as a crucial step in the framework, laying the foundation for the subsequent phases of revision and refinement. By systematically guiding the PLM through the process of content generation, this phase enables the creation of well-structured and coherent documents that meet the desired objectives.

### 3.3. Revision Phase

Following the completion of the Content Generation phase, the document undergoes a thorough review and revision process. The primary goal of this phase is to critically evaluate the generated content, identify any inconsistencies or inaccuracies, and make necessary revisions to enhance clarity and coherence.

During the Revision phase, the document is systematically analyzed to ensure alignment with the predefined structure and objectives. Each section and subsection is scrutinized for logical flow, coherence, and relevance to the overall theme of the document. Any discrepancies or shortcomings are noted, and appropriate revisions are proposed to address them.

Additionally, the Revision phase involves soliciting feedback from human reviewers or domain experts to provide valuable insights and perspectives on the document's content. This external feedback serves as a valuable resource for identifying areas of improvement and refining the document to meet the highest standards of quality.

Throughout the Revision phase, emphasis is placed on iterative refinement and continuous improvement. Revisions are made iteratively based on feedback from both automated evaluation mechanisms and human reviewers, ensuring that the document evolves towards its final form.

### 3.4. Refinement Phase

The final phase of the framework is the Refinement phase, where the document undergoes final polishing and optimization. In this phase, attention is paid to fine-tuning various aspects of the document, including language style, tone, formatting, and overall presentation.

The Refinement phase aims to enhance the readability and aesthetic appeal of the document, making it more engaging and accessible to its intended audience. This may involve adjusting sentence structures, improving transitions between sections, and refining the overall writing style to achieve clarity and conciseness.

Furthermore, the Refinement phase may also involve optimization of technical aspects such as paragraph length, word choice, and sentence complexity. By carefully refining these elements, the document can effectively convey its message and capture the reader's attention.

Overall, the Refinement phase represents the culmination of the framework, where the document is polished to perfection before final publication or dissemination. Through meticulous attention to detail and iterative refinement, the document is transformed into a polished and professional piece of writing that meets the highest standards of excellence.

## 4. Case Study

In order to elucidate the innovative concept of automatic prompt generation in Pretrained Language Models (PLMs), we can delve into a concrete example, such as the task of creating a syllabus. For that purpose "Data Structures" course was chosen for the syllabus creation to illustrate the proposed approach that unfolds in four distinct phases, aiming to streamline and enhance the completion of a complex writing task.

### 4.1. Phase One: Eliciting Syllabus Structure

To begin the first phase of structuring the syllabus for the "Data Structures" course, we would first identify the main components and objectives of the course, such as introducing fundamental concepts, algorithms, and applications related to data structures. Next, we would categorize these components into primary, secondary, and tertiary sections, organizing them hierarchically based on their level of importance and complexity. This would involve delineating topics like arrays, linked lists, trees, and graphs, as well as their associated operations and algorithms. Additionally, we would consider prerequisites, learning outcomes, and assessment methods to ensure comprehensive coverage of the subject matter. Through iterative refinement using techniques like Recursive Criticism and Improvement, we would continuously revise the syllabus to optimize its structure and



content, ensuring coherence and alignment with educational objectives. The following is the result of the first phase (the GPT 3.5 output):

## Data Structures Course Syllabus

### 1 .Introduction to Data Structures

- Overview of data structures and their significance in computer science
- Basic terminology and concepts: data, information, algorithms
- Importance of efficient data storage and retrieval

### 2 .Linear Data Structures

Arrays; Linked Lists; Stacks; Queues; Applications and implementations

### 3 .Non-linear Data Structures

Trees; Binary Trees; Binary Search Trees; AVL Trees; B-Trees; Graphs.

Graph representations; Graph traversal algorithms; Shortest path algorithms

### 4 .Advanced Data Structures

Priority Queues; Heaps; Hash Tables; Disjoint Set Union; Trie

Segment Trees;

### 5 .Algorithmic Analysis

Asymptotic notation (Big O, Big Theta, Big Omega)

Time and space complexity analysis of data structure operations

### 6 .Applications of Data Structures

Searching and sorting algorithms; Graph algorithms (e.g., Dijkstra's algorithm, Prim's algorithm);

Dynamic programming.

### 7 .Advanced Topics

Advanced tree structures (e.g., Red-Black Trees, Splay Trees); Persistent data structures; Parallel data structures

### 8 .Practical Implementation and Problem Solving

- Hands-on coding exercises and programming assignments
- Problem-solving strategies using data structures.
- Project work on real-world applications

### 9 .Evaluation and Assessment

- Quizzes and tests on theoretical concepts
- Programming assignments and projects to evaluate practical implementation skills.
- Final examination covering the entire syllabus.

### 10 .Prerequisites

- Basic knowledge of programming concepts and algorithms
- Understanding of fundamental mathematical concepts such as sets, functions, and relations.

### 11 .References

- Data Structures and Algorithms textbooks
- Online resources and tutorials

- Research papers and articles on advanced topics in data structures.

#### 4.2. Phase Two: Content Generation

In the second phase of syllabus development for the "Data Structures" course, we would focus on generating content for each section identified in the structure. Compared to syllabus writing, this phase is much more intensive and important in tasks such as books, or large text documents. Generating such documents involves crafting prompts that guide the Pretrained Language Model (PLM) to provide detailed explanations, examples, and exercises for each topic. Prompts would be tailored to elicit responses that cover theoretical concepts, practical implementations, and problem-solving strategies related to data structures. The PLM would be prompted to explain key algorithms, demonstrate code implementations, and provide real-world applications of each data structure. Additionally, prompts would encourage the PLM to generate exercises and assignments for students to reinforce their understanding and skills. Iterative refinement of prompts based on feedback and evaluation ensures the completeness and coherence of the syllabus content. The following text illustrates the second phase on the first item of the syllabus: "Introduction to Data Structures".

##### **Introduction to Data Structures**

- *Overview of data structures and their significance in computer science*

*Data structures are essential in computer science for efficient data organization and manipulation. They find applications in arrays (sequential data storage), linked lists (dynamic memory allocation), trees (hierarchical data representation), and hash tables (fast data retrieval).*

- *Basic terminology and concepts: data, information, algorithms*

*In "Basic terminology and concepts," students will learn about fundamental elements such as data, information, and algorithms. Understanding these concepts is crucial as they form the building blocks of computer science.*

- *Importance of efficient data storage and retrieval*

*Most organizations generate very large amounts of data. So efficient data storage and retrieval are crucial in computer science for optimizing resource usage and enhancing system performance. They enable quick access to information, reducing latency and improving user experience. "*

#### 4.3. Phase Three: Revision

In phase 3, "Revision," the syllabus content is critically evaluated for coherence, accuracy, and alignment with educational goals. Iterative reviews are conducted to identify and address areas for improvement. Revisions are made based on these evaluations to enhance clarity, completeness, and relevance. The revised syllabus undergoes thorough scrutiny to ensure it meets high standards and effectively educates students on data structures. For example, note the addition below of the subject "Relationship with Algorithmic Complexity". As an example below is the revised version of the "" from the second phase.

##### **Introduction to Data Structures**

- *Overview: Data structures are fundamental in computer science, enabling efficient data organization and manipulation. They are utilized in arrays for sequential data storage, linked lists for dynamic memory allocation, trees for hierarchical data representation, and hash tables for fast data retrieval.*
- *Basic Terminology and Concepts: In this section, students will grasp essential elements such as data, information, and algorithms, which are the cornerstone of computer science.*
- *Significance of Efficient Data Storage and Retrieval: Given the vast amounts of data generated by organizations, efficient data storage and retrieval are paramount in optimizing resource usage and*

*enhancing system performance. They facilitate swift access to information, reducing latency, and enhancing user experience.*

- *Relationship with Algorithmic Complexity: Data structures significantly impact algorithmic efficiency, affecting factors like time and space complexity. Understanding this relationship helps in selecting the most suitable data structure for optimizing algorithm performance."*

#### 4.4. Phase Four: Refinement Phase

In the "Refinement phase" for the "Data Structures" syllabus, I would focus on fine-tuning the content to ensure clarity, coherence, and relevance. This involves reviewing each section for any inconsistencies or gaps and making necessary adjustments. Additionally, I would seek feedback from students and instructors to incorporate their perspectives and improve the overall quality of the syllabus. Finally, I would conduct a final review to ensure that the refined syllabus meets the educational objectives and effectively prepares students for mastering data structures.

The final revised text is as follows:

### Introduction to Data Structures

Overview: Data structures are essential in computer science, facilitating efficient data organization and manipulation. They include arrays for sequential storage, linked lists for dynamic allocation, trees for hierarchical representation, and hash tables for rapid retrieval.

Basic Terminology and Concepts: Students will grasp foundational elements such as data, information, and algorithms, pivotal in computer science.

Importance of Efficient Data Storage and Retrieval: With organizations generating vast data volumes, optimizing storage and retrieval is critical for resource utilization and system performance enhancement, ensuring swift access and improved user experience.

Relationship with Algorithmic Complexity: Data structures play a crucial role in algorithmic efficiency, impacting factors like time and space complexity. Understanding this relationship aids in selecting the most suitable structures for optimizing algorithm performance.

## 5. Discussion

This paper highlights the importance of iterative prompting schemes, such as Recursive Criticism and Improvement (RCI), in enhancing the reasoning abilities of Language Models (LLMs). The literature is categorized and analyzed to emphasize the effectiveness of various prompting approaches, including monotone refinement, converging prompting cycles, and complexity-based prompting, in improving the performance of pre-trained models across different tasks. Additionally, it underscores the significance of context-aware prompting and semantic prompting in guiding the generation process effectively.

The paper proposes a new four-phases framework for writing of structured documents such as course syllabus, book proposal, business plan and even a complete book. The proposed framework works well with standard documents that have clear structure that is easily identified. The approach will not work well if no structure is identified, or even exists. This is the limitation of this model.

The framework is illustrated using a case study of crafting a course syllabus for the course "Data Structure". This concrete example underscores the potential of generative pretrained multi-step prompting to revolutionize complex writing tasks. By review, revision and refinement, this approach not only streamlines the writing process but also ensures the coherence and relevance of the generated output.

### *Future Research*

The rest of the discussion is dedicated to future research topics and questions. We identify the following 7 directions for future research (each with its research questions):

1. Adaptive Prompting Mechanisms:

How can generative pretrained multi-step prompting adapt dynamically to different writing task complexities and contexts?

What mechanisms can be implemented to enable the model to self-adjust prompts based on the intricacy of the desired output?

2. Multimodal Integration:

How can the approach be extended to seamlessly integrate visual, auditory, and tactile information, expanding its applicability to tasks requiring a multimodal understanding?

What challenges and opportunities arise when incorporating diverse sensory inputs into the prompting process?

3. Ethical Implications and Bias Mitigation:

What measures can be implemented to mitigate potential biases in the generated content, ensuring fairness and ethical considerations?

How can the model be designed to handle sensitive topics and adhere to ethical guidelines in diverse cultural and societal contexts?

4. Real-time Feedback and Iterative Refinement:

How can the prompting process be enhanced to provide real-time feedback to users, facilitating an iterative refinement of the generated content?

In what ways can user input be leveraged during the prompt generation process to improve the relevance and accuracy of the outcomes?

5. Domain-Specific Adaptability:

How can the model be fine-tuned to exhibit domain-specific expertise, catering to the unique requirements of distinct professional fields?

What methodologies can be employed to enable users to customize the prompting approach for specialized applications?

6. Cognitive Load and User Experience:

How does the complexity of prompts impact the cognitive load on users, and what strategies can be employed to optimize user experience?

What insights can be gleaned from user studies to refine the prompting process and enhance user satisfaction?

7. Fine-Tuning for Specific Tasks:

What methodologies can be developed to allow users to fine-tune the model for specific writing tasks, ensuring task-specific nuances are captured accurately?

How can the model be trained to understand and generate content specific to distinct professional domains?

## **6. Conclusions**

As we embark on the next phase of AI evolution, embracing generative pretrained multi-step prompting offers a pathway towards more sophisticated, context-aware language models. The applications explored herein highlight its potential to enhance productivity, creativity, and overall efficacy across a spectrum of domains. Moving forward, research and development in this area promise to unlock new dimensions in natural language understanding and generation, cementing automated multi-step generative pretrained prompting as a pivotal advancement in the ever-evolving field of artificial intelligence.

We explored 7 future research topics and their related questions aim to guide the continued exploration and development of automated multi-step generative pretrained prompting, fostering

advancements in AI capabilities and its seamless integration into a myriad of writing tasks across various domains.

## References

1. Armitage, J., Magnusson, T., & McPherson, A. (2023). Studying Subtle and Detailed Digital Lutherie: Motivational Contexts and Technical Needs. In *Proc. New Interfaces for Musical Expression*.
2. Batten, J., Sinclair, M., Glocker, B., & Schaap, M. (2023). Image To Tree with Recursive Prompting. *arXiv preprint arXiv:2301.00447*.
3. Caufield, J. H., Hegde, H., Emonet, V., Harris, N. L., Joachimiak, M. P., Matentzoglou, N., ... & Mungall, C. J. (2023). Structured prompt interrogation and recursive extraction of semantics (SPIRES): A method for populating knowledge bases using zero-shot learning. *arXiv preprint arXiv:2304.02711*.
4. Chen, P., Guo, Z., Haddow, B., & Heafield, K. (2023). Iterative translation refinement with large language models. *arXiv preprint arXiv:2306.03856*.
5. Chen, A., Lorenz, P., Yao, Y., Chen, P. Y., & Liu, S. (2023, June). Visual prompting for adversarial robustness. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.
6. Chen, K., & Sun, S. (2023, June). CP-Rec: contextual prompting for conversational recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, No. 11, pp. 12635-12643).
7. Chen, Z., Li, Z., Zeng, Y., Zhang, C., & Ma, H. (2024). GAP: A novel Generative context-Aware Prompt-tuning method for relation extraction. *Expert Systems with Applications*, 123478.
8. Ciucă, I., Ting, Y. S., Kruk, S., & Iyer, K. (2023). Harnessing the Power of Adversarial Prompting and Large Language Models for Robust Hypothesis Generation in Astronomy. *arXiv preprint arXiv:2306.11648*.
9. Feldt, R., Kang, S., Yoon, J., & Yoo, S. (2023, September). Towards autonomous testing agents via conversational large language models. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 1688-1693). IEEE.
10. Firdaus, M., Singh, G., Ekbal, A., & Bhattacharyya, P. (2023, October). Multi-step Prompting for Few-shot Emotion-Grounded Conversations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (pp. 3886-3891).
11. Fu, Y., Peng, H., Sabharwal, A., Clark, P., & Khot, T. (2022, September). Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.
12. Gong, R., Danelljan, M., Sun, H., Mangas, J. D., & Van Gool, L. (2023). Prompting diffusion representations for cross-domain semantic segmentation. *arXiv preprint arXiv:2307.02138*.
13. Grunde-McLaughlin, M., Lam, M. S., Krishna, R., Weld, D. S., & Heer, J. (2023). Designing LLM Chains by Adapting Techniques from Crowdsourcing Workflows. *arXiv preprint arXiv:2312.11681*.
14. Han, J., Na, J., & Hwang, W. (2024). Semantic Prompting with Image-Token for Continual Learning. *arXiv preprint arXiv:2403.11537*.
15. Hongru, W. A. N. G., Wang, R., Mi, F., Deng, Y., Zezhong, W. A. N. G., Liang, B., ... & Wong, K. F. (2023, December). Cue-CoT: Chain-of-thought Prompting for Responding to In-depth Dialogue Questions with LLMs. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
16. Huang, Z., Zhou, J., Xiao, G., & Cheng, G. (2023, October). Enhancing in-context learning with answer feedback for multi-span question answering. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 744-756). Cham: Springer Nature Switzerland.
17. Jha, S., Jha, S. K., Lincoln, P., Bastian, N. D., Velasquez, A., & Neema, S. (2023, June). Dehallucinating large language models using formal methods guided iterative prompting. In *2023 IEEE International Conference on Assured Autonomy (ICAA)* (pp. 149-152). IEEE.
18. Jin, X., Lan, C., Zeng, W., & Chen, Z. (2023). Domain Prompt Tuning Via Meta Relabeling for Unsupervised Adversarial Adaptation. *IEEE Transactions on Multimedia*.
19. Kazemi, M., Kim, N., Bhatia, D., Xu, X., & Ramachandran, D. (2022). Lambada: Backward chaining for automated reasoning in natural language. *arXiv preprint arXiv:2212.13894*.
20. Kim, G., Baldi, P., & McAleer, S. (2024). Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.
21. Kim, D., Yoon, S., Park, D., Lee, Y., Song, H., Bang, J., & Lee, J. G. (2023). One Size Fits All for Semantic Shifts: Adaptive Prompt Tuning for Continual Learning. *arXiv preprint arXiv:2311.12048*.
22. Lesmono, A. D., & Astutik, S. (2023). The Effect of Scaffolding Prompting Questions on Scientific Writing Skills in the Inquiry Classroom. *JPI (Jurnal Pendidikan Indonesia)*, 12(1).
23. Li, S., Zhao, R., Li, M., Ji, H., Callison-Burch, C., & Han, J. (2023). Open-domain hierarchical event schema induction by incremental prompting and verification. *arXiv preprint arXiv:2307.01972*.
24. Li, Y., Yang, C., & Ettinger, A. (2024). When Hindsight is Not 20/20: Testing Limits on Reflective Thinking in Large Language Models. *arXiv preprint arXiv:2404.09129*.



25. Liu, Y., Lu, Y., Liu, H., An, Y., Xu, Z., Yao, Z., ... & Gui, C. (2023). Hierarchical prompt learning for multi-task learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10888-10898).
26. Liu, Z., Patwary, M., Prenger, R., Prabhumoye, S., Ping, W., Shoeybi, M., & Catanzaro, B. (2022). Multi-stage prompting for knowledgeable dialogue generation. arXiv preprint arXiv:2203.08745.
27. Liu, C., Zhao, F., Kang, Y., Zhang, J., Zhou, X., Sun, C., ... & Kuang, K. (2023). Rexuie: A recursive method with explicit schema instructor for universal information extraction. arXiv preprint arXiv:2304.14770.
28. Lo, R., Sridhar, A., Xu, F. F., Zhu, H., & Zhou, S. (2023, December). Hierarchical prompting assists large language model on web navigation. In Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 10217-10244).
29. Lu, P., Qiu, L., Chang, K. W., Wu, Y. N., Zhu, S. C., Rajpurohit, T., ... & Kalyan, A. (2022). Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. arXiv preprint arXiv:2209.14610.
30. Ma, Y., Yu, Y., Li, S., Jiang, Y., Guo, Y., Zhang, Y., ... & Liao, X. (2023). Bridging Code Semantic and LLMs: Semantic Chain-of-Thought Prompting for Code Generation. arXiv preprint arXiv:2310.10698.
31. Ma, P., Ding, R., Wang, S., Han, S., & Zhang, D. (2023, December). InsightPilot: An LLM-Empowered Automated Data Exploration System. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 346-352).
32. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., ... & Clark, P. (2024). Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems, 36. 37th Conference on Neural Information Processing Systems (NeurIPS 2023).
33. Madaan, A., Hermann, K., & Yazdanbakhsh, A. (2023, December). What makes chain-of-thought prompting effective? a counterfactual study. In Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 1448-1535).
34. Misra, K., Santos, C. N. D., & Shakeri, S. (2023). Triggering Multi-Hop Reasoning for Question Answering in Language Models using Soft Prompts and Random Walks. arXiv preprint arXiv:2306.04009.
35. Papakyriakopoulos, O., & Mboya, A. M. (2023). Beyond algorithmic bias: A socio-computational interrogation of the Google search by image algorithm. Social Science Computer Review, 41(4), 1100-1125.
36. Parra Pennefather, P. (2023). The Art of the Prompt. In Creative Prototyping with Generative AI: Augmenting Creative Workflows with Generative AI (pp. 197-246). Berkeley, CA: Apress.
37. Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation.
38. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2590–2602, Hong Kong, China. Association for Computational Linguistics.
39. Qi, J., Xu, Z., Shen, Y., Liu, M., Jin, D., Wang, Q., & Huang, L. (2023, December). The art of SOCRATIC QUESTIONING: Recursive thinking with large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 4177-4199).
40. Razdaibiedina, A., Mao, Y., Hou, R., Khabsa, M., Lewis, M., & Almahairi, A. (2023). Progressive prompts: Continual learning for language models. arXiv preprint arXiv:2301.12314.
41. Scarlatos, A., Smith, D., Woodhead, S., & Lan, A. (2024). Improving the Validity of Automatically Generated Feedback via Reinforcement Learning. arXiv preprint arXiv:2403.01304.
42. Shi, C., Zhai, R., Song, Y., Yu, J., Li, H., Wang, Y., & Wang, L. (2023). Few-shot Sentiment Analysis Based on Adaptive Prompt Learning and Contrastive Learning. Information Technology and Control, 52(4), 1058-1072.
43. Sihite, R., Mawardi, A. L., & Marjanah, M. (2023). Implementation of Probing Prompting to Improving Student Learning Outcomes on Ecosystem Material in Senior High School. Biodidaktika: Jurnal Biologi dan Pembelajarannya, 18(1), 48-55.
44. Sim, W. L., Mun, N., & Silva, A. (2024). Using AI-Enabled Divergence and Convergence Patterns as a Quantitative Artifact in Design Education. Journal of Mechanical Design, 146, 032301-1.
45. Skreta, M., Yoshikawa, N., Arellano-Rubach, S., Ji, Z., Kristensen, L. B., Darvish, K., ... & Garg, A. (2023). Errors are useful prompts: Instruction guided task programming with verifier-assisted iterative prompting. arXiv preprint arXiv:2303.14100.
46. Sodhi, P., Branavan, S. R. K., & McDonald, R. (2023). HeaP: Hierarchical Policies for Web Actions using LLMs. arXiv preprint arXiv:2310.03720.
47. Sridhar, A., Lo, R., Xu, F. F., Zhu, H., & Zhou, S. (2023). Hierarchical prompting assists large language model on web navigation. arXiv preprint arXiv:2305.14257.
48. Suh, S., Chen, M., Min, B., Li, T. J. J., & Xia, H. (2023). Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. arXiv preprint arXiv:2310.12953.
49. Sun, W., Cai, H., Chen, H., Ren, P., Chen, Z., de Rijke, M., & Ren, Z. (2023). Answering ambiguous questions via iterative prompting. arXiv preprint arXiv:2307.03897.

50. Suzgun, M., & Kalai, A. T. (2024). Meta-prompting: Enhancing language models with task-agnostic scaffolding. arXiv preprint arXiv:2401.12954.
51. Swamy, S., Tabari, N., Chen, C., & Gangadharaiyah, R. (2023). Contextual dynamic prompting for response generation in task-oriented dialog systems. arXiv preprint arXiv:2301.13268.
52. Tan, Z., Zhang, X., Wang, S., & Liu, Y. (2021). MSP: Multi-stage prompting for making pre-trained language models better translators. arXiv preprint arXiv:2110.06609.
53. Tanwar, E., Borthakur, M., Dutta, S., & Chakraborty, T. (2023). Multilingual llms are better cross-lingual in-context learners with alignment. arXiv preprint arXiv:2305.05940.
54. Tian, Y., Ravichander, A., Qin, L., Bras, R. L., Marjeh, R., Peng, N., ... & Brahman, F. (2023). MacGyver: Are Large Language Models Creative Problem Solvers?. arXiv preprint arXiv:2311.09682.
55. Yan, Y., Zheng, P., & Wang, Y. (2024). Enhancing large language model capabilities for rumor detection with Knowledge-Powered Prompting. *Engineering Applications of Artificial Intelligence*, 133, 108259.
56. Yang, Y., Huang, P., Cao, J., Li, J., Lin, Y., & Ma, F. (2024). A prompt-based approach to adversarial example generation and robustness enhancement. *Frontiers of Computer Science*, 18(4), 184318.
57. Yang, F. J., Su, C. Y., Xu, W. W., & Hu, Y. (2023). Effects of developing prompt scaffolding to support collaborative scientific argumentation in simulation-based physics learning. *Interactive Learning Environments*, 31(10), 6526-6541.
58. Yu, Z., He, L., Wu, Z., Dai, X., & Chen, J. (2023). Towards better chain-of-thought prompting strategies: A survey. arXiv preprint arXiv:2310.04959.
- Wan, X., Sun, R., Dai, H., Arik, S. O., & Pfister, T. (2023). Better zero-shot reasoning with self-adaptive prompting. arXiv preprint arXiv:2305.14106.
59. Wang, B., Deng, X., & Sun, H. (2022). Iteratively prompt pre-trained language models for chain of thought. arXiv preprint arXiv:2203.08383.
60. Wang, W., Sun, Y., Li, W., & Yang, Y. (2024). Transhp: Image classification with hierarchical prompting. *Advances in Neural Information Processing Systems*, 36.
61. Wu, Z., Jiang, M., & Shen, C. (2024, March). Get an A in Math: Progressive Rectification Prompting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 17, pp. 19288-19296).
62. Wu, J., Liu, X., Yin, X., Zhang, T., & Zhang, Y. (2024, March). Task-Adaptive Prompted Transformer for Cross-Domain Few-Shot Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 6, pp. 6012-6020).
63. Zhang, X., Li, S., Yang, X., Tian, C., Qin, Y., & Petzold, L. R. (2023). Enhancing small medical learners with privacy-preserving contextual prompting. arXiv preprint arXiv:2305.12723.
64. Zhang, P., Chai, T., & Xu, Y. (2023). Adaptive prompt learning-based few-shot sentiment analysis. *Neural Processing Letters*, 55(6), 7259-7272.
65. Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., ... & Wang, G. (2023). Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792.
66. Zheng, C., Liu, Z., Xie, E., Li, Z., & Li, Y. (2023). Progressive-hint prompting improves reasoning in large language models. arXiv preprint arXiv:2304.09797.
67. Zhou, H., Li, M., Xiao, Y., Yang, H., & Zhang, R. (2023). LLM Instruction-Example Adaptive Prompting (LEAP) Framework for Clinical Relation Extraction. *medRxiv*, 2023-12.

**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.