Prompt Engineering: Unlocking better Responses from Large Language Models

A. Kamruzzaman, N. Samdrup
Department of Business and Economics, School of Business and IS, York College, the City University of NY, Queens, New York, USA

D. Nurse
Department of Mathematics and Computer Science, School of Arts & Sciences, York College, the City University of NY, Queens, New York, USA

ABSTRACT: The research explores how different prompt engineering strategies affect the performance of Large Language Models, using York GPT and ChatGPT as the primary test systems. Both models were used to evaluate three prompt engineering strategies: zero-shot, few-shot, and chain-of-thought. Research highlights the importance of prompt design and addresses the challenges in establishing a consistent strategy that generates well-defined output. By implementing a well-crafted prompt, the research demonstrates it can significantly improve the quality of a model response.

INTRODUCTION

Large Language Models (LLMs) have greatly improved in mimicking human language, answering queries, and completing tasks. However, the effectiveness of the LLMs often relies on how users present their prompts. Prompt engineering—developing structured and purposeful input—is vital for generating high-quality responses. As LLMs grow more sophisticated, trained on extensive data to perform tasks like summarization or question answering (Cao et al. 2023), the design of prompts becomes increasingly important. Studies show that using examples within prompts (few-shot learning) can significantly enhance response accuracy (Zhou et al.2022). Research also indicates that refining prompt strategies helps maintain performance across diverse tasks (Sahoo et al. 2024). Despite this progress, designing prompts that consistently yield high-quality output remains a challenge. Many existing studies lack clear and standardized frameworks for prompt construction, often resulting in inefficient or inconsistent responses from LLMs. To address this gap, this research explores various prompt engineering methods and evaluates their influence on LLM responses. The study uses two models: OpenAI's ChatGPT-4 Mini and YorkGPT, a model trained on York College content. It experiments with three prompting strategies: zero-shot, few-shot, and chain-of-thought. By using well-organized prompt techniques, such as zero-shot, few-shot, and chain-of-thought promptings, it will significantly improve the relevance, clarity, and completeness of LLM-generated responses.

GENERAL BACKGROUND

Prompt engineering refers to the practice of formulating clear instructions for LLMs, like ChatGPT, to generate meaningful results. As these models have evolved to handle more complex queries and tasks, prompt design has become essential for steering responses in useful directions (Sahoo et al. 2024). Initial applications of LLMs often suffered from vague outputs due to poorly structured prompts. Researchers found that carefully written prompts, which provide context and specify the task, improve model performance (Patel & Parmar 2024). This is further illustrated in Figure 1, where the user prompt is refined through prompt engineering with instructions and context, leading to an improved response. Over time, this field has expanded to include multiple strategies for improving prompt interpretation. Basic techniques include zero-shot prompting, where the model receives no examples, and few-shot prompting, where examples are embedded in the prompt (Cao et al. 2023). More recent methods, like chain-of-thought prompting, guide models to explain their reasoning, helping them tackle complex tasks more effectively (Jin et al. 2025). These advances underline the importance of prompt engineering in producing consistent and relevant LLM outputs across different applications.

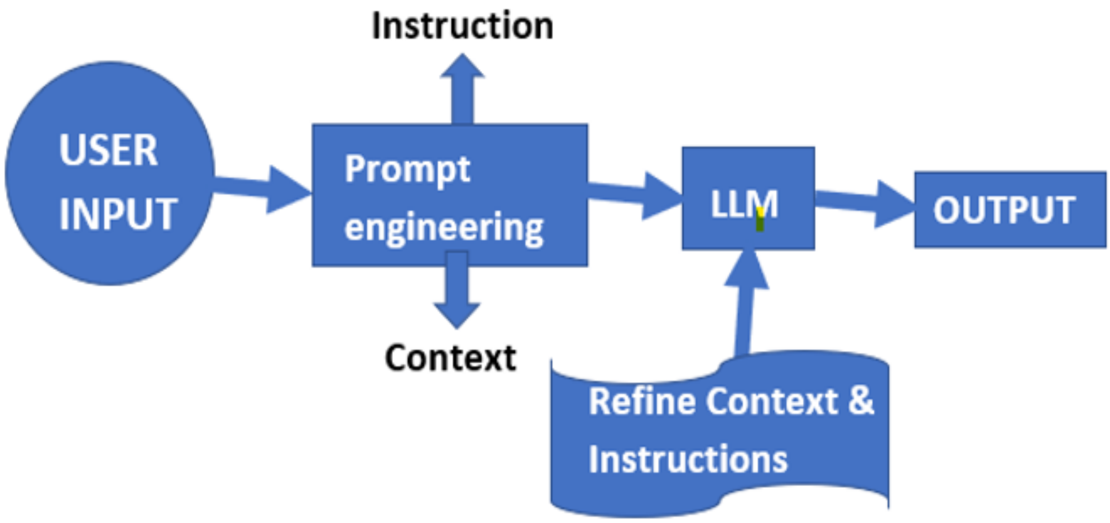

Figure 1. User input refined through prompt engineering before LLM output.

LITERATURE REVIEW

Prompt engineering plays an essential role in enhancing how well large LLMs, such as ChatGPT, perform. Research indicates that the way a prompt is phrased can strongly influence the relevance, clarity, and precision of AI-generated responses. (Shin et al. 2023) examine several strategies, like zero-shot and few-shot prompting. These methods help LLMs perform tasks with minimal or no prior examples, making them particularly effective in settings where limited training data is available. Chain-of-thought (CoT) prompting has attracted interest due to its ability to support advanced reasoning. This approach guides models to solve problems step by step, resulting in more coherent and logically structured responses. Techniques like Auto-CoT and LogiCoT seek to improve this by automating prompt design and minimizing reasoning errors in outputs (Hsieh et al. 2023). Another method, known as Retrieval-Augmented Generation (RAG), incorporates external data into prompts, allowing models to pull in relevant facts when generating answers. This improves both the factual correctness and reliability of their responses. Although these methods have advanced prompt engineering, some challenges remain. There is currently no universal guideline for writing prompts, and the success of each technique can vary depending on the model and the task. The lack of adaptable frameworks makes it hard to implement prompt engineering across all use cases. Future work is needed to create more generalizable and versatile prompting strategies. In summary, prompt engineering continues to evolve. It acts as a connection point between what users want and how models behave, helping LLMs respond in more accurate and meaningful ways. As research in this area progresses, developing effective prompting techniques will be key to unlocking the full capabilities of LLMs.

RESEARCH DESIGN & METHODOLOGY

This study analyzes how effective three prompting strategies—zero-shot, few-shot, and chain-of-thought (CoT)—are in improving the performance of LLMs. Two models were tested: YorkGPT, which was fine-tuned for York College tasks, and OpenAI's Chat GPT-4o-mini, accessed through the OpenAI API. The goal was to assess how these prompting techniques influenced response quality in terms of relevance, clarity, completeness, and reasoning across both specific and general task domains. The YorkGPT test focused on prompts related to accessing York email and IT services and used the following prompt styles:

I. *Zero-Shot Prompting:*
A simple query, such as "How do I activate my York email and get help using it?" was used to evaluate the model's ability to generate brief and relevant responses without added context.

II. *Few-Shot Prompting:*
Before posing the main question, examples like "How do I create an email account?" were introduced, followed by "Now explain how to activate a York email account and where to find help." This approach examined how well the model could use examples to give structured replies.

III. *Chain-of-Thought Prompting:*
The prompt guided the model step by step, requiring it to describe the steps for accessing and activating a York email account, as well as how to find support services.

In the ChatGPT-4-mini experiment, the topic used was photosynthesis:

I. *Zero-Shot Prompting:*
The model was asked, "What is photosynthesis?" to test its ability to provide a brief definition.

II. *Few-Shot Prompting:*
Before asking the main question, two examples were shared—one explaining water absorption and the other discussing sunlight capture. These examples created a context, allowing the model to deliver a more thorough explanation.

III. *Chain-of-Thought Prompting:*
A detailed prompt asked the model to explain the roles of sunlight, water, and carbon dioxide in photosynthesis and describe how glucose and oxygen are produced.

The two models were tested on separate tasks aligned with their training data. YorkGPT, trained specifically in York College content, was evaluated using a campus-related task (email activation). In contrast, ChatGPT-4 mini was assessed on a general topic (photosynthesis). This comparison was designed to explore how a domain-specific model like YorkGPT performs against a general-purpose model, based on the nature of the task. To maintain fairness, the ChatGPT-4-mini experiment applied token limits: 100 tokens for zero-shot, 250 for few-shot, and 300 for CoT prompting (Zhou et al., 2022). YorkGPT had no token restrictions due to its narrower scope. Figure 2 (flowchart) illustrates the methodology used to conduct the ChatGPT -4-mini experiment, detailing each step from prompt creation to output analysis.

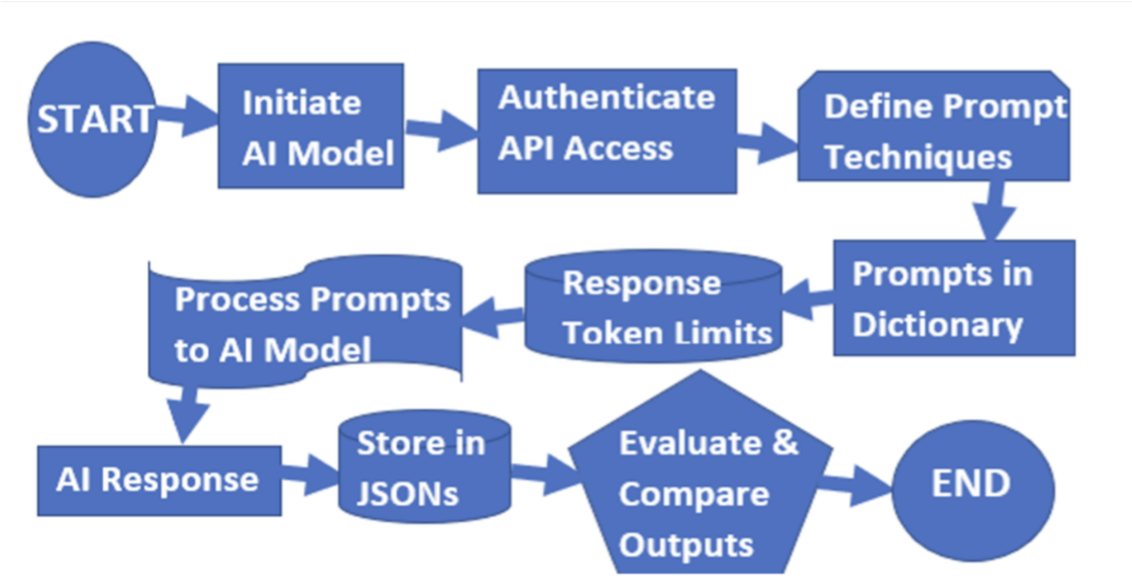

Figure 2. Flowchart showing the step-by-step process of the experiment.

In both cases, responses were evaluated based on predefined metrics: relevance to the task, clarity of instructions, completeness of details, and reasoning quality (Patel & Parmar 2024).
As highlighted by Patel & Parmar (2024) in their research proposal, Prompt Engineering for LLMs, their study explores how different prompting techniques influence AI model performance and effectiveness. The following Table 1 summarizes the performance of each prompting strategy for YorkGPT and ChatGPT 4 mini.

Table 1. Comparative Metrics for ChatGPT-4-mini and YorkGPT Across Prompting Strategies.

Metric	Zero-shot (OpenAI)	Few-shot (OpenAI)	Chain-of-thought (OpenAI)	Zero-shot (YorkGPT)	Few-shot (YorkGPT)	Chain-of-thought (YorkGPT)
Relevance	Clear but lacking detail	Detailed and relevant.	Detailed and logical, but a bit wordy.	Lacked details like IT support	Detailed and task specific.	Well-detailed and logical
Clarity	Easy to understand but brief.	Clear and well-structured.	Logical, clear but repetitive	Easy to understand but brief.	Clear with structured steps.	Logical and clear.
Completeness	Missed stages like glucose production.	Covered all stages comprehensively.	Covered all stages but wordy at times.	Missed key steps, e.g., password setup.	Covered most aspects thoroughly.	Comprehensive but wordy.
Reasoning & Quality	Simple ideas, no clear logic	Examples enhanced clarity	Step-by-step reasoning was strong.	Unstructured and lacked reasoning	Clearer with examples and logical flow	Step-by-step reasoning was strong.

This study followed a structured approach to compare the output of two language models—YorkGPT and ChatGPT-4-mini. Both were evaluated using the same three prompting formats: zero-shot, few-shot, and chain-of-thought. Each type of prompt was carefully created to ensure consistency in the tasks given to both models. The evaluation focused on four criteria—relevance, clarity, completeness, and reasoning (Patel & Parmar 2024)—and was applied to all prompt types to ensure a fair comparison. Table 1 outlines how each prompt and metric was applied throughout the experiment. This structure helped guide the data collection process and ensured consistency across test conditions. To assess the quality of model outputs across different prompting strategies, a rubric-based evaluation method was implemented. Each response was evaluated using four key criteria. Relevance measured how well the response addressed the prompt; clarity focused on the coherence and ease of understanding; completeness evaluated whether all required components of the answer were present; and reasoning & quality assessed the depth and quality of explanation provided. This scoring rubric from Table 2 was applied uniformly to all responses generated by both ChatGPT-4-mini and YorkGPT across the zero-shot, few-shot, and chain-of-thought prompting strategies. Each criterion was rated on a scale from 1 to 10, where a score of 10 indicated the highest quality performance. The resulting scores provided the raw data necessary for calculating mean scores and statistical analysis.

Table 2. Scoring Rubric (1–10 Scale)

Score Range	Level	Description	Example Phrases from Table 1
9-10	Excellent	Exceptionally clear, complete, and well-structured; logical reasoning throughout	"Well-detailed and logical (YorkGPT)", "Step-by-step reasoning was strong (ChatGPT)"
7-8	Good	Clear and mostly complete; some minor flaws or wordiness	"Detailed and relevant (ChatGPT)", "Clear and well-structured (York GPT)"
5-6	Fair / Average	Understandable but brief or missing some detail; minor issues in logic	"Clear but lacking detail (ChatGPT)", "Easy to understand but brief (YorkGPT)."
3-4	Weak	Incomplete or unclear; lacks depth or structure	"Missed key steps (YorkGPT)", "Simple ideas, no clear logic (ChatGPT)"
1-2	Poor	Confusing, irrelevant, or poorly reasoned; major issues in clarity or logic	"Unstructured and lacked reasoning (YorkGPT)", "Missed stages (ChatGPT)"

A total of three prompts were used for each model. Each model's response was evaluated using a structured scoring rubric across four key criteria. These four evaluation criteria served as the basis for statistical analysis. To determine whether the differences in performance between prompt strategies were statistically significant, a series of paired t-tests were conducted. The analysis focused on three strategy comparisons: (1) Zero-shot vs. Few-shot, (2) Few-shot vs. Chain-of-thought, and (3) Zero-shot vs. Chain-of-thought. For each comparison, the rubric scores across the four evaluation criteria were treated as matched pairs, resulting in a sample size of $n = 4$ per test.

The statistical analysis followed these steps:

1. Score differences were calculated by subtracting each criterion score of one prompt strategy from the corresponding score of the other.
2. The Mean Difference of the paired scores was computed to observe the average performance gap between the strategies.
3. The T-statistic (t) was obtained using the formula: t = (mean difference) / (standard error)
4. The P-value was then determined using two methods:
   - Manually, via a t-distribution table using degrees of freedom ($df = n – 1 = 3$)
   - Programmatically, using Python for greater precision.

A p-value less than 0.05 was interpreted as statistically significant, indicating a meaningful difference in performance between the prompt strategies. P-values greater than 0.05 suggested that any observed differences may not be statistically meaningful. In summary, the experiment offered a reliable method to assess how different prompting strategies affect the performance and response quality of LLMs.

PROJECT REQUIREMENTS AND DATASET IMPLEMENTATION RESULTS

The study compared ChatGPT-4-mini and YorkGPT to evaluate their effectiveness in handling both general scientific questions and task-specific queries. The objective was to assess how each prompting technique affects clarity, accuracy, and completeness in responses. To achieve this, various tools and procedures were used to evaluate the impact of different prompting methods on both the model's performance. The experiments were conducted using Python as the programming language with the following tools and resources:

I. *ChatGPT-4-mini-Experiment:*
- Platform: VS Code served as the integrated development environment (IDE).
- Libraries:
  i. OpenAI: For generating responses from the ChatGPT-4-mini model.
  ii. JSON: To store and organize outputs.

II. *YorkGPT Experiment:*
- Platform: Google Colab provided an accessible and scalable environment for running YorkGPT-specific experiments.
- Libraries:
  i. Transformers: For loading and fine-tuning YorkGPT.
  ii. Google Colab Libraries: For handling files and runtime configurations.

  Storage: Google Drive was used for persistent file storage, including storing YorkGPT training data from the York College website.

The following Table 3 shows the structured prompts used in both experiments to evaluate each prompting method. It outlines the prompts applied for both ChatGPT-4-mini and YorkGPT, demonstrating how strategies like Zero-Shot, Few-Shot, and Chain-of-Thought were designed to measure their effect on response quality.

Table 3. Comparison of Prompting Strategies in ChatGPT -4-mini and YorkGPT Experiments

Prompt type	ChatGPT-4-mini example	YorkGPT example	Purpose
Zero-shot	"What is photosynthesis?"	"How do I activate my York email and get help using it?"	To test the model's ability to provide concise responses without prior context
Few-shot	Two examples of plant processes were given before asking: 'How does photosynthesis work?'	Two examples, account creation and password reset, were provided before asking: 'How do I activate a York email and get help?'	To assess whether guiding examples enhanced the depth and clarity of responses.
Chain-of-thought	A structured prompt guided the model to define and explain the process of photosynthesis.	A structured prompt guided the model to explain email activation, checking the spam folder, and accessing IT support.	To test logical reasoning and structured responses

To evaluate model performance, a custom scoring rubric was applied, as outlined in Table 4. Each model response was scored from 1 to 10 on four evaluation criteria, across all three prompt types, with higher scores reflecting better performance. This produced a total of 12 scores per model for each prompt strategy (3 prompts × 4 criteria). The raw scores assigned in Table 4 show quantitative values for each prompt strategy, enabling a structured assessment of their relative effectiveness.

Table 4. Evaluation Score

Metric	Zero-shot (ChatGPT)	Few-shot (ChatGPT)	Chain-of-thought (ChatGPT)	Zero-shot (YorkGPT)	Few-shot (YorkGPT)	Chain-of-thought (YorkGPT)
Relevance	6	8	7.5	4	8	9
Clarity	6	8	7	6	8	9
Completeness	4	9	8	4	7.5	8
Reasoning & Quality	4	7	9.5	3	7	9.5

Scores were assigned using Table 2 data, a standardized scoring rubric designed to ensure consistency across different prompt types and model responses. YorkGPT's Zero-shot responses performed the weakest, especially in reasoning and completeness, indicating that minimal guidance reduced quality. Few-shot prompting (York GPT) improved scores across relevance, clarity, completeness, and reasoning. Chain-of-Thought prompting (York GPT) consistently delivered the strongest results in reasoning and completeness. A similar pattern emerged with ChatGPT-4-mini, where Few-shot and Chain-of-Thought prompts generally outperformed Zero-shot. These observed trends in the raw scores provide preliminary evidence of performance variation across prompting techniques. To determine whether these differences were statistically significant, paired t-tests were conducted using the mean scores for each prompting strategy along with their corresponding p-values. The mean scores presented in Table 5 reflect the average performance ratings derived from the scoring values in Table 4. The Zero-shot strategy for ChatGPT produced a mean score of 5, while Few-shot and Chain-of-Thought (Cot) both achieved a mean score of 8, indicating that additional context or examples enhanced response quality. In contrast, YorkGPT demonstrated more variation. The Zero-shot approach scored 4.25, the lowest among all configurations, while Few-shot improved to 7.625, and Chain-of-Thought (Cot) further increased to 8.875, the highest mean observed in the study. These results suggest that both models benefited from more structured prompt strategies.

Table 5. Mean Evaluation Scores

Strategy & Model	Total Score	Mean = Total ÷ 4
Zero-shot (ChatGPT)	6 + 6 + 4 + 4 = 20	20 ÷ 4 = 5
Few-shot (ChatGPT)	8 + 8 + 9 + 7 = 32	32 ÷ 4 = 8
CoT (ChatGPT)	7.5 + 7 + 8 + 9.5 = 32	32 ÷ 4 = 8
Zero-shot (YorkGPT)	4 + 6 + 4 + 3 = 17	17 ÷ 4 = 4.25
Few-shot (YorkGPT)	8 + 8 + 7.5 + 7 = 30.5	30.5 ÷ 4 = 7.625
CoT (YorkGPT)	9 + 9 + 8 + 9.5 = 35.5	35.5 ÷ 4 = 8.875

To ensure consistency, prompts were implemented programmatically. Table 6 showcases the Python code used to define the ChatGPT-4-mini prompts and demonstrates how the dataset was structured for evaluation. The YorkGPT prompts followed a similar structure, tailored to a domain-specific task

Table 6. Examples of prompts used in the ChatGPT -4- mini experiments and YorkGPT .

ChatGPT4- mini (Prompts)	YORKGPT (Prompts)

prompts = {
"zero shot": "What is photosynthe-
sis?",
"few shot": """
Example 1: Explain how plants absorb
water.
Plants absorb water through their
roots. The water travels through
tubes called xylem to reach the
leaves, where
It is used for photosynthesis.
Example 2: Explain how plants capture
sunlight.
Plants capture sunlight using chloro-
phyll, a green pigment in their chlo-
roplasts. This light energy powers
the
photosynthesis process.
Now, explain how photosynthesis
works.
"""
"Chain_of_thought": """
Think step by step and explain how
photosynthesis works.
1.Define photosynthesis briefly.
2.Explain the role of sunlight, wa-
ter, and carbon dioxide.
3.Describe the process of glucose
production and oxygen release.
Keep the explanation concise and easy
to understand.
"""
}
	Zero-Shot Prompt:
	#How do I activate my York email
	and get
	help using it.
	Few-Shot Prompt:
	#Example 1: How do I create an
	email
	account?
	#Go to the email provider's web-
	site, click
	"Sign Up," fill in your name,
	choose a username and password,
	and verify your email.
	#Example 2: How do I reset my
	email
	password?
	#Click "Forgot Password" on the
	login page,
	enter your registered email ad-
	dress, follow the instructions
	sent to your email.
	#Now, explain how to activate a
	York email
	account and where to get help.
	Chain-of-Thought Prompt:
	#Think step by step:
	#1. Explain how to activate a
	York email
	account (e.g., initial login,
	password setup) -
	#2. Describe how to access the
	email (e.g.,
	web portal or email app).
	#3. Suggest where to get help
	(e.g., York IT
	services, help desk, or support
	guides

Each prompt type was given a specific token limit to balance response length and clarity in the ChatGPT-4-mini experiment. Figure 3 illustrates the script used to assign these limits, which were set as follows:

- 100 tokens for zero-shot prompts to ensure concise responses.
- 250 tokens for few-shot prompts to allow for examples and details.
- 300 tokens for CoT prompts to accommodate logical reasoning.

```
token_sizes = {
"zero_shot": 100,
"few_shot": 250,
"Chain_of_thought": 300
}
```

Figure 3. Python code setting token limits for different prompt types.

The dataset for ChatGPT-4-mini experiments focused on general knowledge queries, such as explaining photosynthesis. In contrast, YorkGPT's dataset included York-specific queries, like email activation processes. For example, few-shot prompts in YorkGPT experiments featured examples of account creation to guide users through email activation steps. Similarly, ChatGPT-4-mini's experiments used examples of water absorption and sunlight capture to contextualize the explanation of photosynthesis processes. These tailored datasets enabled both models to deliver responses relevant to their respective domains (Hsieh et al. 2023). Token constraints were applied in the ChatGPT-4-mini experiments to ensure consistent outputs. This approach, influenced by (Zhou et al. 2022), helped control verbosity while maintaining clarity in the responses. The experiments demonstrated that adapting prompt strategies to the task and dataset significantly enhances model performance, reaffirming findings in prior studies on the impact of tailored prompting (Cao et al. 2023).

DISCUSSION

This research explored how effective three prompting methods—zero-shot, few-shot, and chain-of-thought (CoT)—were when applied to Chat GPT-4-mini and YorkGPT, which is tailored for York College use cases. The purpose was to analyze how each technique influenced clarity, completeness, relevance, and reasoning quality across both general and York-specific tasks. The findings showed that each strategy produced different outcomes. Zero-shot prompts often led to brief responses that missed important steps, such as troubleshooting in YorkGPT or photosynthesis stages in Chat GPT-4-mini 's responses. This aligns with prior research suggesting that zero-shot is suitable for straightforward tasks but less effective for complex, multi-step questions (Cao et al. 2023). Few-shot prompting, by contrast, incorporated guiding examples that made responses complete and more organized, especially when including IT-related context for YorkGPT or deeper science content for Chat GPT-4-mini. This supports previous findings on the strengths of few-shot methods in improving clarity and adaptability (Sahoo et al. 2024). CoT prompting worked particularly well in tasks that demanded logical thinking. It generated logically structured responses, but they were lengthy —an issue also mentioned by (Jin et al. 22025). Overall, the study emphasizes the importance of choosing an appropriate prompting technique to enhance the LLM's performance. Few-shot prompting balanced clarity and detail, CoT excelled in logical structure, and zero-shot was most effective for simple queries. Figures 4a and 4 b display the original outputs from YorkGPT and ChatGPT-4-mini for the three prompting types, comparing their responses to scientific explanation tasks (ChatGPT-4-mini) and email setup tasks (YorkGPT).

York GPT (Output)

Zero-Shot Prompt:

To activate your York email, use one of these methods. If you need help using it, visit the Student Accounts Activation page for assistance.

Few-Shot Prompt:

To activate a York email account, follow these steps:
1.Check your York College Network Directory for the default email format using [SYNACUNY]
2.If you haven't received an activation email within 30 minutes of creating your account, check your spam folder and ensure no filters are blocking it.
3.Contact the IT Helpdesk if problems persist.

Chain-of-Thought Prompt:

To activate a York email account, follow these steps. Firstly, you will need to create your account using the link provided by York College.Once created, use the default password assigned during setup and change it immediately for l security purposes. To access your email, log in through any standard web browser or install the Outlook app on your mobile device. If you encounter issues with activating your account or re-

ChatGPT 4-mini (Output)
Prompt Type: zero_shot
Response: Photosynthesis is a biochemical process through which green plants,
algae, and some bacteria convert light energy into chemical energy in the form of glucose. This process primarily occurs in the chloroplasts of plant cells, which contain chlorophyll, the green pigment that captures light energy, primarily from the sun.

Prompt Type: few_shot
Response: Photosynthesis is the process by which plants, algae, and some bacteria convert light energy into chemical energy stored in glucose. This process primarily occurs in the chloroplasts of plant cells. It begins when chlorophyll captures sunlight, which provides the energy needed for the reaction. Water(H_2O) absorbed by the roots and carbon dioxide (CO_2) from the air enter the chloroplasts. In the presence of sunlight, water is split into oxygen (O2) and hydrogen (H) during a light-dependent reaction. The oxygen is released as a byproduct.The energy from the sunlight is then used to convert carbon dioxide and
the hydrogen into glucose ($C_6H_{12}O_6$) through a series of reactions known as the Calvin cycle. In summary, photosynthesis transforms light energy into chemical energy, producing glucose for energy and growth while releasing oxygen as a vital byproduct.

Prompt Type: chain_of_ thought
Response: #1.Definition of Photosynthesis
Photosynthesis is a process used by plants, algae, and some bacteria to convert light energy into chemical energy in the form of glucose. It mainly occurs in the chloroplasts of plant cells, where this energy conversion supports the growth and energy needs of the organism.

#2.Role of Sunlight, Water, and Carbon Dioxide
-Sunlight:This is the primary energy source for photosynthesis. Chlorophyll, the green pigment in plants, captures sunlight.
- Water (H_2O): Plants absorb water from the soil through their roots. Water serves as a reactant in the process, which is split into hydrogen and oxygen.
- Carbon Dioxide: This gas is taken from the air through small openings called stomata. Plants use carbon dioxide as one of the key raw materials to produce glucose.

#3. Process of Glucose Production and Oxygen Release
I.Light Reactions: In the presence of sunlight, chlorophyll absorbs light energy, which is then used to split water molecules into oxygen and hydrogen. This process releases oxygen as a byproduct into the atmosphere.
II.Calvin Cycle (Dark Reactions):The hydrogen produced in the light reactions combines with carbon dioxide in a series of reactions to form glucose. This cycle occurs in the stroma of the chloroplasts and does not require light.

Figure 4. (a) YorkGPT Output. (b) ChatGPT 4 mini Output.

Results from zero-shot prompts produced fast but shallow replies, often missing important details like email setup steps in YorkGPT or the full explanation of photosynthesis stages in ChatGPT-4-mini. These responses were short and efficient but lacked the depth needed for more complex or multi-step tasks. Few-shot prompting generated more well-rounded and thorough responses by including examples that helped the model better grasp the task. This approach was especially useful in YorkGPT, where example-based prompts led to clearer instructions, and in ChatGPT 4-mini, where scientific explanations became more detailed and structured (Sahoo et al. 2024). CoT prompts gave the most logical and in-depth answers by walking through tasks step by step, which helped in tasks requiring reasoning or multiple stages of explanation. However, this strategy occasionally resulted in longer responses than necessary, which may not always be ideal for users looking for quick answers. To further support these observations, a series of paired t-tests was conducted to determine whether the observed differences in performance across prompt strategies were statistically significant. The results are presented in Table 7, which displays the mean difference, standard deviation, standard error, t-statistics, and p-value for each comparison.

Table 7. Statistical Test Results for Prompt Strategy Comparisons

Group A	Group B	Differences	Mean Difference	Standard Deviation	Standard Error	T-statistic	P-value
Zero-shot (ChatGPT)	Few-shot (ChatGPT)	[-2, -2, -5, -3]	-3	1.4142	0.7071	-4.2426	0.024
Few-shot (ChatGPT)	Chain-of-thought (ChatGPT)	[0.5, 1.0, 1.0, -2.5]	0	1.6833	0.8416	0	1
Zero-shot (ChatGPT)	Chain-of-thought (ChatGPT)	[-1.5, -1.0, -4.0, -5.5]	-3	2.1213	1.0607	-2.8284	0.0663
Zero-shot (YorkGPT)	Few-shot (YorkGPT)	[-4.0, -2.0, -3.5, -4.0]	-3.375	0.9465	0.4732	-7.1317	0.0057
Few-shot (YorkGPT)	Chain-of-thought (YorkGPT)	[-1.0, -1.0, -0.5, -2.5]	-1.25	0.866	0.433	-2.8868	0.0632
Zero-shot (YorkGPT)	Chain-of-thought (YorkGPT)	[-5.0, -3.0, -4.0, -6.5]	-4.625	1.493	0.7465	-6.1954	0.0085

For both ChatGPT-4-mini and YorkGPT, zero-shot vs. few-shot comparisons yielded p-values well below the $\alpha = 0.05$ significance threshold (0.024 and 0.0057, respectively), indicating that few-shot prompting significantly outperformed zero-shot in both models. These findings statistically validate the earlier observation that zero-shot responses were consistently weaker in depth and clarity. However, when comparing zero-shot vs. chain-of-thought, the p-values (0.0663 for ChatGPT-4-mini and 0.0085 for YorkGPT) revealed mixed outcomes. While YorkGPT showed a statistically significant improvement with chain-of-thought prompting, ChatGPT's result narrowly missed the significance threshold. This suggests that although chain-of-thought was descriptively better, the evidence was not strong enough to claim a statistically meaningful difference for ChatGPT at the $\alpha = 0.05$ level. In the few-shot vs. chain-of-thought comparison, both models returned p-values above 0.05 (1.0 for ChatGPT-4-mini, 0.0632 for YorkGPT), suggesting no statistically significant difference between these two strategies. This implies that while CoT may appear more logical or structured, it is not consistently superior to few-shot prompting from a statistical standpoint. These findings are visualized in Figure 5, a bar chart that displays the p-values for each pairwise comparison between prompt strategies. The visual layout reinforces the numerical insights presented in Table 7, highlighting which differences reached statistical significance and offering a more accessible interpretation of the model performance trends.

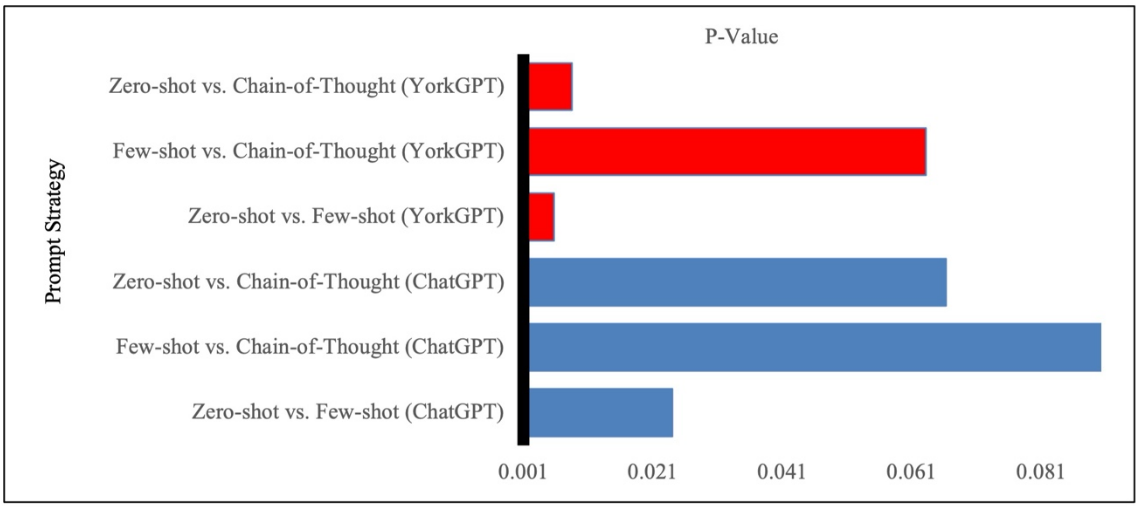

Figure 5. P-values for prompt strategy comparisons.

Together, the statistical results and observational findings support the conclusion that few-shot prompting is consistently effective. Zero-shot approaches tend to underperform, especially on complex tasks. Chain-of-thought prompting shows strength in logical reasoning but yields mixed statistical significance. These differences highlight that each prompt strategy has its strengths depending on the task at hand. Selecting the right strategy based on task complexity is essential for improving performance across contexts (Sahoo et al. 2024).

LIMITATIONS AND FUTURE WORK

This study showed that prompt engineering can strongly influence the performance of LLMs, but several limitations were identified. A major challenge involved the inconsistency in responses, which stemmed from the inherent randomness in how LLMs produce text. This aligns with (Liu et al. 2024), who observed that the unpredictable output of models like ChatGPT is largely due to their design architecture. In this study, for example, ChatGPT-4-mini occasionally produced different outputs for the same prompt, especially in zero-shot and few-shot conditions. This suggests that even well-designed prompts can yield unstable results depending on how the model samples text. This highlights the need for more refined prompt strategies that help produce consistent outputs (Liu et al. 2024). Another limitation involved token truncation during the ChatGPT-4-mini experiment. In chain-of-thought (CoT) prompts, some responses were cut off when they exceeded the token limit, resulting in incomplete answers. (Cao et al. 2023) also observed that strict token constraints can limit a model's ability to generate complete, detailed, or multi-step responses. This issue was especially noticeable in longer CoT responses, where key parts of reasoning were sometimes omitted. To mitigate this, future work could explore dynamic token allocation or truncation warnings during response generation to alert users. A further limitation relates to the task-specific performance differences observed between ChatGPT-4-mini and YorkGPT. While YorkGPT performed well on institution-specific tasks—such as answering questions related to York College—its performance declined in more general scientific or open-domain prompts. In contrast, ChatGPT-4-mini was more robust across general domains but lacked depth on York-specific content. This difference underscores the tradeoff between specialization and generalizability in LLMs.The use of a domain-specific dataset for YorkGPT also presented both strengths and weaknesses. While YorkGPT performed well in tasks related to York College, its smaller and narrower dataset limited its ability to handle general topics. (Liu et al. 2024) emphasized that diverse and large-scale datasets are important for building models that perform well across different subjects. The YorkGPT training dataset, sourced from the publicly available York College website, is valuable for building contextual responses in a campus-specific setting. However, its limited scope restricts the model's adaptability outside of that niche.To address these limitations, future work should explore ways to reduce response variability and improve the model's ability to carry context over longer outputs. Incorporating methods such as temperature tuning or response re-ranking could help reduce randomness in generated answers. Adding a system that allows the model to adjust its responses based on previous user input—like a feedback loop—could also help the model generate more accurate and personalized answers (Cao et al. 2023). Broadening the YorkGPT dataset to cover more diverse topics could enhance its ability to generalize while maintaining its effectiveness in specialized tasks (Liu et al. 2024). Additionally, (Liu et al. 2024) highlighted ethical concerns related to how prompts are designed. As prompt engineering becomes more powerful, future research should include clear safeguards to prevent harmful use and ensure that these technologies are applied responsibly for public benefit.

CONCLUSIONS

This research demonstrates how a prompt is framed directly influences the clarity and effectiveness of responses produced by LLMs. Through the evaluation of three prompting strategies—zero-shot, few-shot, and chain-of-thought—on Chat GPT-4-mini and YorkGPT, the results showed that few-shot and chain-of-thought methods led to responses that were more thorough, coherent, and logically structured. These findings support the original hypothesis that well-designed prompts improve LLM output quality. The results are in line with existing studies by (Sahoo et al. 2024) and (Jin et al. 2025), which emphasize that example-based and step-by-step prompting improve the reasoning and clarity of language model responses. This outcome is particularly important as LLMs are increasingly used in education, customer support, and workplace tools where accuracy and completeness are essential. In these practical applications, few-shot prompting can support student learning by offering structured examples that clarify expectations and guide understanding. Similarly, chain-of-thought prompting enhances multi-step reasoning in customer service by helping the model produce clearer, step-by-step responses that improve user comprehension and satisfaction. By aligning prompt strategies with specific use cases, developers and practitioners can significantly enhance the quality and reliability of AI-generated outputs. Overall, this study reinforces the importance of selecting the right prompting technique based on task complexity. Effective prompt engineering allows users to optimize LLM performance, ensuring AI-generated responses are more accurate, reliable, and useful across different applications.

REFERENCES

Cao, D. *et al.* (2023) “TEMPO: Prompt-based Generative Pre-trained Transformer for time series forecasting,” *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/2310.04948.

Cao, J. *et al.* (2025) “A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair,” *Automated software engineering*, 32(1). Available at: https://doi.org/10.1007/s10515-025-00492-x.

Hewing, M. and Leinhos, V. (2024) “The Prompt Canvas: A literature-based practitioner guide for creating effective prompts in Large Language Models,” *arXiv [cs.AI]*. Available at: http://arxiv.org/abs/2412.05127.

Hsieh, C.-J. *et al.* (2023) “Automatic Engineering of Long Prompts,” *arXiv [cs.AI]*. Available at: http://arxiv.org/abs/2311.10117.

Jin, C. *et al.* (2025) “APEER : Automatic prompt engineering enhances large language model reranking,” in *Companion Proceedings of the ACM on Web Conference 2025*. New York, NY, USA: ACM, pp. 2494–2502.

Knoth, N. *et al.* (2024) “AI literacy and its implications for prompt engineering strategies,” *Computers and Education: Artificial Intelligence*, 6(100225), p. 100225. Available at: https://doi.org/10.1016/j.caeai.2024.100225.

Li, H., Leung, J. and Shen, Z. (2024) “Towards goal-oriented prompt engineering for large Language Models: A survey,” *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/2401.14043.

Lin, Z. (2024) “How to write effective prompts for large language models,” *Nature human behaviour*, 8(4), pp. 611–615. Available at: https://doi.org/10.1038/s41562-024-01847-2.

Liu, Yi *et al.* (2023) “Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study,” *arXiv [cs.SE]*. Available at: http://arxiv.org/abs/2305.13860.

Pornprasit, C. and Tantithamthavorn, C. (2024) “Fine-tuning and prompt engineering for large language models-based code review automation,” *Information and software technology*, 175(107523), p. 107523. Available at: https://doi.org/10.1016/j.infsof.2024.107523.

Patel, H. & Parmar, S. (2024) “Prompt engineering for large language model”,ResearchGate. Available at: https://www.researchgate.net

Sahoo, P. *et al.* (2024) “A systematic survey of prompt engineering in large language models: Techniques and applications,” *arXiv [cs.AI]*. Available at: https://rotmandigital.ca/wp-content/uploads/2024/09/A-Systematic-Survey-of-Prompt-Engineering-in-Large-Language-Models.pdf (Accessed: June 6, 2025).

Shin, J. *et al.* (2023) “Prompt engineering or fine-tuning: An empirical assessment of LLMs for code,” *arXiv [cs.SE]*. Available at: http://arxiv.org/abs/2310.10508.

Vaswani, A. *et al.* (2017) *Attention is all you need*, *Neurips.cc*. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (Accessed: June 6, 2025).

Zhou, Y. *et al.* (2022) “Large language models are human-level prompt engineers,” *arXiv [cs.LG]*. Available at: https://openreview.net/pdf?id=92gvk82DE- (Accessed: June 6, 2025).