A Critical Analysis of Prompt Engineering Strategies for Binary Question Answering

Sıla Keskin
Department of Computer Science
Università degli Studi di Milano
sila.keskin@studenti.unimi.it

Abstract

Prompt engineering has emerged as a key technique for influencing the behavior of large language models without modifying model parameters. This study presents a systematic analysis of different prompt engineering strategies for binary (Yes/No) question answering across tasks with varying levels of difficulty and reasoning requirements. We compare baseline, clarity-oriented, output-constrained, chain-of-thought, and self-verification prompts on factual, misconception-based, and engineering reasoning questions. Experimental results show that no single prompting strategy consistently outperforms the baseline across all tasks. While certain prompts yield task-specific gains, engineering and mathematical reasoning questions remain challenging for all strategies. These findings highlight the strong task dependency of prompt engineering and emphasize the importance of careful evaluation and critical interpretation of results.

Keywords: Prompt Engineering, Large Language Models, Evaluation, Binary Question Answering

1 Introduction

Large Language Models (LLMs) have recently achieved remarkable performance across a wide range of Natural Language Processing (NLP) tasks, including question answering, reasoning, and knowledge-intensive inference. Beyond architectural improvements, prompt engineering has emerged as a powerful mechanism for influencing model behavior at inference time without modifying model parameters. By carefully designing textual instructions, it is possible to steer models toward more structured reasoning, concise answers, or specific output formats. A growing body of research has explored the impact of prompting strategies on model performance. In particular, Chain-of-Thought (CoT) prompting has been shown to improve performance on tasks requiring multi-step reasoning by encouraging models to explicitly articulate intermediate reasoning steps [4]. Building on this idea, subsequent work has proposed self-verification and self-refinement strategies, in which models are prompted to review or correct their own reasoning before producing a final answer [2]. These approaches suggest that prompting can partially compensate for reasoning errors and improve reliability. However, recent studies have highlighted important limitations of reasoning-oriented prompting. In some settings, encouraging explicit reasoning does not improve—and may even degrade—performance, particularly on tasks that are simple, highly factual, or sensitive to output format [5]. Moreover, prompt engineering strategies often exhibit strong task dependency, making it difficult to identify universally optimal prompting schemes. An additional and often overlooked challenge concerns evaluation methodology. Many prompt-based studies rely on naive exact-match metrics that assume explicit answer tokens (e.g., “Yes” or “No”) appear in the model output. In practice, especially when reasoning is encouraged, model responses frequently express conclusions implicitly through natural language explanations. This can introduce evaluation artifacts that disproportionately penalize verbose or reasoning-based prompts, leading to potentially misleading conclusions. In this work, we conduct a focused empirical study on the effectiveness of different prompt engineering strategies for binary (Yes/No) question answering. We compare a baseline prompt, a clarity-oriented prompt, a strict Yes/No output constraint, chain-of-thought prompting, and self-verification prompting across multiple groups of questions with increasing difficulty. These include simple factual questions, misconception-based questions, and a set of engineering and mathematical reasoning questions derived from real university-level examination materials. To address evaluation challenges, we adopt a hybrid evaluation strategy that combines explicit rule-based extraction with lightweight semantic judgment, allowing us to fairly assess both concise and reasoning-heavy outputs. Rather than focusing on maximizing absolute accuracy, the goal of this study is to critically analyze how prompt design and evaluation choices interact, and to understand under which conditions specific prompting strategies are effective or counterproductive.

2 Research Question and Methodology

2.1 Research Question and Objectives

Prompt engineering techniques have been widely proposed as a means to improve the performance of large language models (LLMs) without modifying model parameters. While prior studies have demonstrated that certain prompting strategies can enhance reasoning performance, their effectiveness appears to be highly dependent on task characteristics and evaluation methodology.

The primary research question addressed in this project is how different prompt engineering strategies affect the accuracy and reliability of large language models on binary (Yes/No) question answering tasks with varying levels of difficulty and reasoning requirements. To address this question, the study compares the accuracy of multiple prompt engineering strategies on the same set of binary questions, analyzes how prompt effectiveness varies across different types of questions—including factual, misconception-based, and engineering reasoning tasks—assesses the stability of prompt strategies across question groups, and evaluates how different evaluation methodologies influence the observed performance of prompt engineering techniques. Rather than focusing on achieving maximal performance, the objective of this work is to critically analyze the conditions under which specific prompting strategies are effective or ineffective.

2.2 Problem Definition

Let $Q = \{q_1, q_2, \ldots, q_n\}$ be a set of binary questions, where each question q_i is associated with a ground-truth label $y_i \in \{\text{Yes}, \text{No}\}$.

Given a prompt template p and a language model M, the model generates a textual response:

$$r_i = M(p, q_i)$$

The task consists of inferring a predicted label $\hat{y}_i \in \text{Yes}, \text{No}$ from the response r_i and evaluating its correctness with respect to the gold label y_i. Unlike traditional classification tasks, the model output r_i is not restricted to a predefined label set and may include explicit binary answers, implicit conclusions expressed through natural language, or intermediate reasoning and verification statements. This variability makes the evaluation of binary correctness non-trivial and motivates the need for a robust evaluation strategy.

2.3 Prompt Engineering Strategies

We evaluate five prompt engineering strategies, each designed to elicit different model behaviors:

- **Baseline Prompt**: A minimal instruction asking the model to answer the question.
- **Clear Prompt**: Encourages concise and unambiguous responses.
- **Yes/No-Constrained Prompt (Y_N)**: Forces the model to respond strictly with "Yes" or "No".
- **Chain-of-Thought Prompt (CoT)**: Instructs the model to reason step by step before producing a final answer.
- **Self-Verification Prompt (Verify)**: Encourages the model to reason step by step and check its reasoning for consistency before giving the final answer.

All prompt strategies are applied to the same questions using the same model, ensuring that observed differences can be attributed solely to prompt design.

2.4 Dataset Construction

The dataset consists of multiple groups of binary questions designed to capture increasing levels of difficulty and distinct cognitive demands:

- **Groups A–D**: General factual and conceptual questions with varying levels of difficulty.
- **Group E (Misconceptions)**: Questions based on widely held but incorrect beliefs, constructed through online research and manual verification.
- **Group F (Engineering Reasoning)**: Questions requiring formal reasoning, mathematical derivations, physical laws, or algorithmic analysis. These questions were derived from publicly available university-level examination materials in mathematics, physics, and computer science.

Questions in Groups A–D were generated with the assistance of a large language model and subsequently reviewed and refined to ensure correctness, clarity, and controlled difficulty. Questions in Group E were gathered through internet research. Group F questions were collected from final and midterm exams of various universities. All questions were modified and converted into yes–no question format.

This grouping enables a systematic analysis of how prompt strategies behave across different task types.

2.5 Evaluation Methodology

A key challenge in evaluating binary question answering with LLMs lies in the free-form nature of model outputs. Responses may not explicitly contain the target labels “Yes” or “No”, particularly when reasoning-oriented prompts are used.

To address this issue, we adopt a hybrid evaluation strategy:

- **Explicit Extraction**: If the model response contains an explicit “Yes” or “No” in its final portion, this label is extracted directly.
- **Semantic Judgment**: If no explicit binary decision is present, a lightweight language-model-based judge is used to infer whether the response semantically corresponds to “Yes” or “No”.

This approach reduces evaluation artifacts while remaining transparent and reproducible. The semantic judgment component is invoked only for ambiguous responses, resulting in minimal additional inference cost. Performance is measured using accuracy, defined as the proportion of correctly classified answers.

2.6 Experimental Protocol

All experiments are conducted using the same language model and identical question sets across prompt strategies. Each prompt–question pair is evaluated independently.

Due to the stochastic nature of large language models, repeated executions of the same prompt–question pair may yield slightly different outputs. In this study, results are reported from a single inference run, and emphasis is placed on relative comparisons between prompt strategies rather than absolute performance values.

The implementation follows a modular and reusable design, with core logic separated into Python modules for prompt definitions, model interaction, evaluation, and analysis. Jupyter notebooks are used exclusively for experimentation, visualization, and result interpretation.

2.7 Summary

This methodology enables a controlled and interpretable comparison of prompt engineering strategies across diverse task types. By explicitly addressing both dataset construction and evaluation challenges, the study aims to provide a nuanced understanding of when and why different prompting strategies are effective.

3 Experimental Results

3.1 Dataset and Evaluation Setup

Experiments were conducted on a dataset of binary (Yes/No) questions organized into six groups (A–F), designed to capture increasing levels of difficulty and distinct reasoning requirements. Groups A–D consist of general factual and conceptual questions, Group E targets common misconceptions identified through online research, and Group F contains engineering and mathematical reasoning questions derived from real university-level examinations.

Model performance is evaluated using accuracy, defined as the proportion of questions for which the predicted binary label matches the ground-truth answer. Due to the free-form nature of language model outputs, a hybrid evaluation strategy is employed to infer the final Yes/No decision, ensuring fair comparison across both concise and reasoning-oriented prompt strategies.

3.2 Overall Prompt Performance

Prompt	Accuracy
Baseline	0.950
Clear	0.933
CoT	0.933
Verify	0.917
Y_N	0.933

Table 1: Overall accuracy of each prompt strategy across all question groups.

As reported in Table 1, the baseline prompt achieves the highest overall accuracy. The remaining prompt strategies—clear instruction, chain-of-thought, self-verification, and Yes/No-constrained prompting—exhibit very similar average performance. This result suggests that, when aggregated across heterogeneous tasks, more sophisticated prompt engineering strategies do not consistently outperform a simple baseline instruction.

3.3 Performance Across Question Groups

Group	Baseline	Clear	CoT	Verify	Y_N
A	0.9	0.9	0.9	0.9	1.0
B	1.0	1.0	1.0	0.9	1.0
C	1.0	0.9	0.9	0.9	0.9
D	1.0	1.0	1.0	1.0	1.0
E	1.0	1.0	1.0	1.0	1.0
F	0.8	0.8	0.8	0.8	0.7

Table 2: Accuracy of each prompt strategy across different question groups.

Table 2 presents a breakdown of accuracy scores by question group. For Groups A–D, which include factual and conceptual questions, performance differences across prompt strategies are minimal. In

Group E, all prompts achieve perfect accuracy, indicating that the model robustly handles common misconceptions once evaluation artifacts are removed.

In contrast, Group F shows a consistent drop in performance across all prompt strategies. This group, composed of engineering and mathematical reasoning questions, represents the most challenging setting, highlighting the limitations of prompt-based interventions for tasks requiring formal computation or structured reasoning.

3.4 Prompt–Group Interaction Analysis

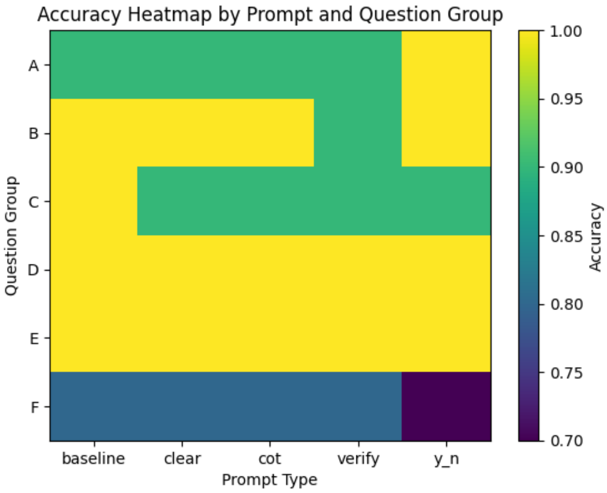

Figure 1: Accuracy heatmap by prompt type and question group.

Figure 1 provides a compact visualization of the interaction between prompt strategies and question groups. The heatmap reveals strong task dependency: while most prompts perform uniformly well on simpler groups, accuracy drops sharply for Group F. Notably, the Yes/No-constrained prompt exhibits the lowest performance in this group, suggesting that strict output constraints are insufficient for complex reasoning tasks.

3.5 Prompt Gain and Loss Relative to Baseline

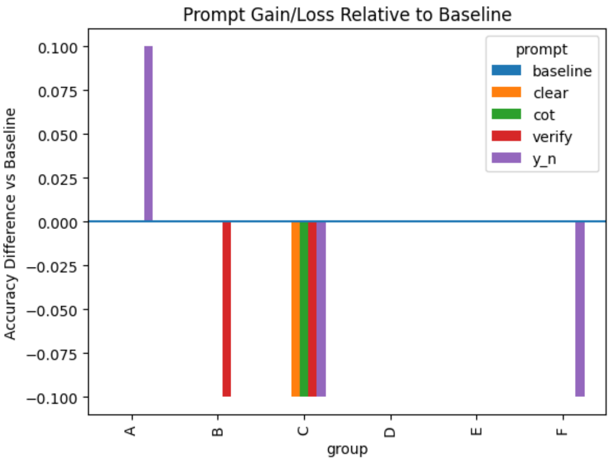

Figure 2: Accuracy gain or loss of each prompt strategy relative to the baseline prompt.

Figure 2 illustrates the relative gain or loss of each prompt strategy compared to the baseline. While certain prompts yield small improvements for specific groups, these gains are offset by performance degradation in others. In particular, reasoning-oriented and output-constrained prompts show negative relative performance in engineering reasoning tasks, reinforcing the absence of a universally optimal prompting strategy.

3.6 Stability of Prompt Strategies

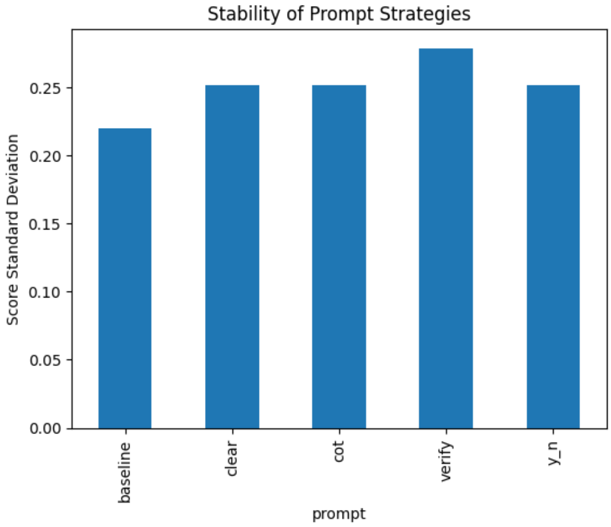

Figure 3: Standard deviation of accuracy across question groups for each prompt strategy.

Figure 3 reports the variability of prompt performance across groups. The baseline prompt exhibits the lowest variance, indicating the most stable behavior across tasks. In contrast, self-verification shows the highest variance, suggesting that more complex prompting strategies trade robustness for potential task-specific gains.

3.7 Summary of Experimental Findings

Overall, the experimental results demonstrate that prompt engineering strategies exhibit strong task dependency. While certain prompts outperform the baseline in narrowly defined scenarios, no single strategy consistently dominates across all question groups. Engineering and mathematical reasoning tasks remain challenging for all prompts, underscoring the limitations of prompt-based methods and the importance of careful evaluation and critical interpretation.

4 Concluding Remarks

This study investigated the effectiveness of different prompt engineering strategies for binary (Yes/No) question answering across tasks with varying levels of difficulty and reasoning requirements. Rather than aiming to maximize absolute performance, the primary goal was to critically analyze how prompt design and evaluation methodology interact to shape observed outcomes.

The experimental results reveal a strong task dependency in the effectiveness of prompt engineering strategies. For simple factual and conceptual questions, prompt choice has minimal impact on performance, and a simple baseline prompt proves both accurate and stable. In misconception-based questions, all prompt strategies perform equally well once evaluation artifacts are removed, suggesting that the model's underlying knowledge is sufficient to resolve these cases. These findings align with recent observations that increased prompting complexity does not necessarily translate into improved reasoning or factual correctness [5].

In contrast, engineering and mathematical reasoning questions remain challenging across all prompt strategies. Even explicit chain-of-thought and self-verification prompts fail to consistently outperform the baseline, indicating that structured reasoning prompts alone are insufficient to guarantee correct formal reasoning. This result supports prior work showing that reasoning-oriented prompting improves performance only under specific conditions and task structures [4]. Furthermore, the increased variability observed for more complex prompts highlights a trade-off between task-specific gains and overall robustness, echoing concerns raised in the literature about over-reliance on elaborate prompting schemes [3].

A key methodological contribution of this work lies in the adoption of a hybrid evaluation strategy. By combining explicit answer extraction with semantic judgment for ambiguous outputs, the evaluation pipeline mitigates biases that disproportionately affect reasoning-based prompts. This finding underscores the importance of evaluation design in prompt-based studies and reinforces the notion that naive exact-match metrics can lead to misleading conclusions when applied to free-form language model outputs [1].

Despite these insights, the study has several limitations. Results are reported from a single inference run, and the inherent stochasticity of large language models may introduce minor variations across executions. Additionally, the question sets, while carefully constructed, remain limited in size and scope. Future work could address these limitations by incorporating multiple inference runs, expanding the dataset to include additional reasoning domains, and exploring alternative evaluation strategies such as model-based confidence estimation or supervised answer classification.

Overall, the findings of this study suggest that prompt engineering should be treated as a task-adaptive design choice rather than a universally beneficial technique. Careful consideration of task characteristics, evaluation methodology, and robustness is essential for drawing meaningful conclusions about the effectiveness of prompting strategies in practical NLP applications.

AI Usage Disclaimer

Parts of this project were developed with the assistance of generative AI tools, specifically OpenAI's ChatGPT (GPT-5) and related large language models. These tools were used to support the development of project ideas, the structuring of methodological workflows, the drafting and refinement of descriptive text, and the generation of code examples and experimental scaffolding. In addition, large language models were used as objects of investigation within the experimental framework, serving as the underlying models evaluated through different prompt engineering strategies. All AI-generated content was carefully reviewed, verified, and, where necessary, modified by the author. The final structure, methodological choices, experimental design, analysis, and interpretations reflect the author's own understanding and critical judgment. Full responsibility for the correctness, originality, and academic integrity of the submitted work is assumed by the author. Generative AI tools were employed as creativity and productivity aids rather than as substitutes for independent reasoning, problem solving, or technical decision-making.

References

[1] Pengfei Liu, Weizhe Yuan, Jinlan Fu, et al. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2307.03109*, 2023.

[2] Aman Madaan, Niket Tandon, Prakhar Gupta, et al. Self-refine: Iterative refinement with self-feedback. *Proceedings of the Association for Computational Linguistics*, 2023.

[3] Sewon Min, Kalpesh Krishna, Xinxi Lyu, et al. Rethinking the role of demonstrations: What makes in-context learning work? *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.

[5] Xingyi Zhou, Shuyan Wang, Yujia Zheng, et al. Large language models are not always better reasoners. *arXiv preprint arXiv:2305.18654*, 2023.