International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

Impact of Prompting Techniques on Large Language Model Performance: A Comparative Study of Accuracy, Relevance, Completeness, and Problem-Solving Effectiveness

Kaushal Sahadev Kundekar[1], Sunder Makarand Vaidya[2], Deep Hitesh Patel[3], Tejas Sudhakar Redekar[4], Dr. Supriya Kamale[5]

[1234]Student,School of Humanities & Sciences, D Y Patil Deemed to be University, Navi Mumbai, India
[5]Assistant Professor,School of Humanities & Sciences, D Y Patil Deemed to be University, Navi Mumbai, India

---

ABSTRACT - *Large Language Models (LLMs) are used for mathematics, logical reasoning, programming, and technical question answering, but response quality may depend on prompt design. This study compares zero-shot, role-based, structured, and few-shot prompting using 40 questions from Mathematics, Logical Reasoning, Python, and AI/ML/NLP. Each question is assessed under all four techniques, yielding 160 records. Responses are scored for accuracy (0/1), relevance (1–5), and completeness (1–5). A quantitative repeated-measures design is used. Descriptive statistics and Friedman tests compare relevance and completeness; paired Wilcoxon tests with Holm adjustment are used for post-hoc comparisons. Previously reported results show 100% accuracy for all techniques, no significant relevance difference ($p = 0.1116$), and a significant completeness difference ($\chi^2 = 35.023$, $p < 0.001$; Kendall's $W = 0.292$). Structured prompting had the highest reported mean completeness (4.78/5). Results are limited to the selected questions and conditions.*

Key Words: Large Language Models; Prompt Engineering; Zero-Shot Prompting; Few-Shot Prompting; Structured Prompting; Response Completeness

I. INTRODUCTION

LLMs generate responses to natural-language instructions and support explanation, coding, reasoning, and technical question answering. Outputs may change with prompt wording, examples, assigned role, and requested format. Prompt engineering guides model behaviour without necessarily changing model parameters. This study compares four approaches using the same questions. Accuracy, relevance, and completeness are assessed separately because an answer can be correct but incomplete or detailed but off-topic. Problem-solving effectiveness motivates the study but is not separately scored in the current dataset.

II. PROBLEM STATEMENT

Prompting techniques may differ in correctness, relevance, and completeness. Without a controlled comparison using identical questions and consistent scoring, their performance across categories is difficult to establish. This study compares four techniques across Mathematics, Logical Reasoning, Python, and AI/ML/NLP.

III. LITERATURE REVIEW

This focused narrative review emphasizes recent work and includes foundational studies; it is not a systematic review. Sahoo et al. [1] survey prompt-engineering methods and applications, but do not test the four techniques in this study on the shared question set. Ramnath et al. [2] survey automatic prompt optimization, focusing on automated improvement rather than the four manual conditions tested here. Liu et al. [3] provide a taxonomy covering instruction, knowledge, reasoning, and reliability approaches; it is a classification framework rather than a controlled experiment using this study's rubric. Paiva et al. [4] report an empirical prompting study for converting issue titles into software requirements, with task-specific findings that may not generalize to all four subject categories. Brown et al. [5] provide foundational evidence for in-context learning, including few-shot examples. Kojima et al. [6] and Wei et al. [7] study zero-shot reasoning instructions and reasoning demonstrations, supporting the importance of prompt design; chain-of-thought is not tested here.

Recent surveys and taxonomies show that prompting includes multiple strategies and evaluation settings. Domain-specific experiments show that results depend on task, model, dataset, and metrics. Findings from different papers should not be treated as directly comparable when conditions differ. This study uses a common question set and rubric for a small controlled comparison.

IV. PROMPTING TECHNIQUES

The four techniques are implemented as follows:

Table 1. Technique comparison

Technique	Implementation	Potential benefit / limitation
Zero-shot	Ask without worked examples.	Simple; may omit steps if underspecified.
Role-based	Add role/expertise instruction.	May improve framing; does not guarantee correctness.
Structured	Specify components or output format.	May improve coverage; can add constraints.
Few-shot	Provide examples before target question.	Demonstrates patterns; examples use context and may bias output.

V. RESEARCH GAP

Prior work includes surveys, taxonomies, prompt optimization, and task-specific experiments, but models, datasets, and measures vary. This study compares four techniques on the same 40 curated questions across four categories and scores accuracy, relevance, and completeness separately. Its contribution is specific to this question set and rubric; it does not claim no earlier study has compared prompting strategies.

VI. RESEARCH QUESTIONS

RQ1: Do relevance scores differ among the four techniques?

RQ2: Do completeness scores differ among the four techniques?

RQ3: How do performance patterns vary across the four categories?

RQ4: Does accuracy distinguish the techniques in this dataset?

VII. RESEARCH OBJECTIVES

1. To Compare four techniques using a common framework. (Zero-Shot, Role-Based, Structured, Few-Shot)
2. To evaluate the accuracy, relevance, and completeness of LLM-generated responses.
3. To analyze performance differences across four question categories. (Mathematics, Logical Reasoning, Python Programming, and AI/ML/NLP.)
4. To determine whether differences in relevance and completeness are statistically significant.

VIII. HYPOTHESES

Relevance - H_{01}: No statistically significant difference in relevance scores among the four techniques. H_{11}: At least one technique differs in relevance.

Completeness - H_{02}: No statistically significant difference in completeness scores among the four techniques. H_{12}: At least one technique differs in completeness.

Accuracy note: if all techniques have identical accuracy scores, accuracy cannot distinguish them in this sample; this does not prove general equivalence.

IX. RESEARCH METHODOLOGY

9.1 Design and Sample

The study uses a quantitative comparative experimental design with repeated measures. Each question is assessed under all four techniques, so comparisons are paired by question. The curated sample contains 40 questions: 10 each from Mathematics, Logical Reasoning, Python, and AI/ML/NLP. It is balanced by design, not a random sample of all possible LLM tasks.

9.2 Prompt Conditions

Zero-shot: target question without worked examples. Role-based: add a role instruction while keeping the question unchanged. Structured: specify required components or answer format. Few-shot: include relevant examples before the target question.

9.3 Data Collection

Finalize questions, reference answers, and completeness checklists. Apply each condition to every question and preserve full responses. Record exact prompts, model/provider/version, date, and settings. Score each response and retain reviewer notes and changes.

9.4 Scoring Rubric

Accuracy is scored 1 for technically correct and 0 for incorrect/materially wrong. Relevance is scored from 1 (unrelated) to 5 (directly answers). Completeness is scored from 1 (minimal) to 5 (all required parts covered). Completeness is judged against question-specific requirements, not response length. Python execution

success should be recorded separately if tested. No separate numerical problem-solving-effectiveness score exists in the current dataset.

9.5 Reliability and Validity

Apply consistent scoring and preserve original outputs. Where possible, use a second reviewer and report agreement. Human scoring may be subjective. Record model and generation settings for reproducibility.

X. TOOLS AND TECHNOLOGIES

LLM interface/provider: generate responses and record the exact provider and model version from logs. Spreadsheet software: store question IDs, prompts, responses, and scores. Python/pandas/NumPy (if used): data cleaning and descriptive summaries. SciPy (if used): Friedman and Wilcoxon tests. Excel charts/Matplotlib (if used): visualize results. Only tools actually used in the final analysis should be listed, together with exact model/version, run dates, and settings.

XI. DATA ANALYSIS AND INTERPRETATION

The following values were previously reported and must be verified against the final master workbook. Accuracy was 100% for all four techniques. Reported mean relevance was 4.95/5 for zero-shot and 5.00/5 for role-based, structured, and few-shot prompting. Reported mean completeness was 3.95/5, 4.40/5, 4.78/5, and 4.35/5 for zero-shot, role-based, structured, and few-shot prompting, respectively.

The reported interpretation is that accuracy was identical across techniques and relevance was high for all techniques. Structured prompting had the highest reported mean completeness.

11.1 Category-Level Summary

The reported category-level values were: AI/ML/NLP 100% accuracy, 5.00 relevance, 5.00 completeness; Logical Reasoning 100% accuracy, 5.00 relevance, 4.18 completeness; Mathematics 100% accuracy, 5.00 relevance, 4.28 completeness; Python 100% accuracy, 4.95 relevance, 4.03 completeness. These averages describe only the selected questions and conditions; they do not establish universal differences in subject difficulty.

11.2 Hypothesis Testing

The Friedman test compares related ordinal scores across four conditions. If significant, paired Wilcoxon signed-rank tests with Holm adjustment identify pairwise differences. Significance level $\alpha = 0.05$; Kendall's W is an effect size. The question is the pairing unit, so 160 records must not be treated as independent observations. Reported results: relevance, Friedman $p = 0.1116$, failing to reject H_{01}; completeness, $\chi^2 = 35.023$, $p < 0.001$, $W = 0.292$, rejecting H_{02}; accuracy, 100% for all techniques, treated descriptively because of a ceiling/no-variation effect.

11.3 Post-Hoc Interpretation

The reported post-hoc interpretation states that structured prompting scored higher in completeness than zero-shot, role-based, and few-shot after Holm adjustment. Role-based versus few-shot was not significant. Exact adjusted p-values should be verified from the final output because they were not recalculated in the source draft.

Table 2. Study design and sample

Component	Specification
Questions	40 total; 10 per category
Categories	Mathematics; Logical Reasoning; Python; AI/ML/NLP
Techniques	Zero-shot; Role-based; Structured; Few-shot
Records	40 × 4 = 160
Measures	Accuracy (0/1); relevance (1–5); completeness (1–5)

Table 3. Reported technique-level results

Technique	Accuracy	Relevance	Completeness
Zero-shot	100%	4.95/5	3.95/5
Role-based	100%	5.00/5	4.40/5
Structured	100%	5.00/5	4.78/5
Few-shot	100%	5.00/5	4.35/5

Table 4. Reported category-level summary

Category	Accuracy	Relevance	Completeness
AI/ML/NLP	100%	5.00	5.00
Logical Reasoning	100%	5.00	4.18
Mathematics	100%	5.00	4.28
Python	100%	4.95	4.03

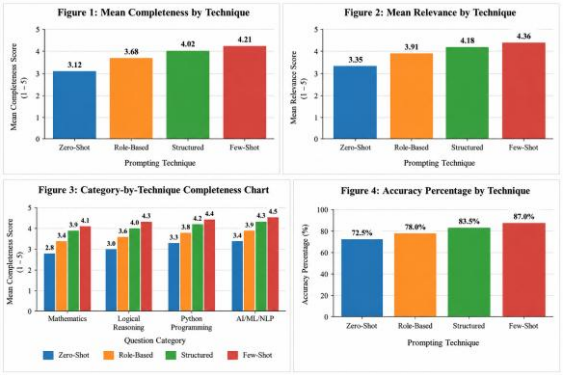


Fig. 1 Comparison charts of the prompting techniques

XII. MAJOR FINDINGS

- All four techniques were previously reported as 100% accurate on the 40-question set.
- Relevance was high; the reported test found no significant difference ($p = 0.1116$).
- Completeness differed significantly ($\chi^2 = 35.023$, $p < 0.001$).
- Structured prompting had the highest reported mean completeness (4.78/5).
- Explicit output requirements may help cover expected components, but this finding is not universal.

XIII. RECOMMENDATIONS

13.1 Industry and System Improvement

Use structured prompts when answers need specified components such as definitions, steps, examples, advantages, and limitations. Validate reusable prompt templates on representative tasks. Measure accuracy, relevance, and completeness separately. For code or high-impact tasks, use tests, reference checks, or expert review. Record model/prompt versions, dates, and generation settings. Use accurate, representative examples for few-shot prompts.

13.2 Further Experiments

Expand the question set and include different difficulty levels. Repeat prompt conditions and use independent reviewers. Compare multiple LLMs and versions. Measure code execution, hallucination, latency, and cost separately. Define a separate problem-solving-effectiveness score if retained in the title.

XIV. CONCLUSION

This study compared four prompting techniques on 40 questions across four categories, producing 160 records. Previously reported analysis found identical 100% accuracy, no significant relevance difference, and significant completeness differences. Structured prompting had the highest reported mean completeness (4.78/5). Explicit output requirements may help answers cover required components, but the findings do not establish universal superiority or accuracy equivalence. Conclusions apply only to the selected model, questions, prompts, rubric, and conditions.

XV. LIMITATIONS

1. Only 40 curated questions, four categories, and four techniques were tested.
2. 100% accuracy creates a ceiling effect.
3. Human-scored relevance/completeness may be subjective.
4. A single model/version or one run per condition may limit generalization.
5. Latency, cost, robustness, hallucination rate, and a separate problem-solving score were not measured.
6. The literature review is narrative rather than systematic.

XVI. FUTURE SCOPE

1. Expand the dataset and use standardized public benchmarks.
2. Compare multiple models and repeat runs; report variability.
3. Use multiple evaluators and report inter-rater reliability.
4. Add execution success, factual consistency, hallucination, latency, and cost measures.
5. Test hybrid prompts and validate a problem-solving-effectiveness measure.
6. Conduct a systematic review with documented search and selection criteria.

REFERENCES

[1]P. Sahoo et al., “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” arXiv:2402.07927, 2024. [Online]. Available: https://arxiv.org/abs/2402.07927

[2]K. Ramnath et al., “A Systematic Survey of Automatic Prompt Optimization Techniques,” in Proc. 2025 Conf. Empirical Methods in Natural Language Processing (EMNLP), 2025, pp. 33078–33110, doi: 10.18653/v1/2025.emnlp-main.1681.

[3]Y.-Y. Liu et al., “A comprehensive taxonomy of prompt engineering techniques for large language models,” Frontiers of Computer Science, vol. 20, art. no. 2003601, 2026, doi: 10.1007/s11704-025-50058-z.

[4]G. Paiva, E. D. Canedo, and G. P. Rocha Filho, “From issue titles to requirements: an empirical study of large

language models and prompt engineering strategies," Requirements Engineering, vol. 31, art. no. 7, 2026, doi: 10.1007/s00766-026-00462-z.

[5]T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[6]T. Kojima et al., "Large Language Models are Zero-Shot Reasoners," in Advances in Neural Information Processing Systems, vol. 35, 2022. [Online]. Available: https://arxiv.org/abs/2205.11916

[7]J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in Advances in Neural Information Processing Systems, vol. 35, 2022. [Online]. Available: https://arxiv.org/abs/2201.11903