Accuracy And Consistency Of Prompt Engineering Techniques: A Comparative Study Of Large Language Models Across Everyday Tasks

Renuka Lakhangaonkar[1], Sujata Sathe[2]
[1,2]*Computer Science, Fergusson College, Pune, India*
doi.org/10.64643/IJIRT211669

***Abstract*—Large Language Models (LLMs) are widely used by students and professionals, and prompt engineering is commonly recommended to improve their output. However, much of the evidence for prompting techniques comes from earlier models evaluated through programmatic access. This study compared five prompting techniques zero-shot, few-shot, chain-of-thought (CoT), role-based, and structured prompting on three LLM chatbots (ChatGPT, Gemini, and Claude) accessed through their free web interfaces. Thirty everyday tasks covering reasoning, summarization, and sentiment classification, five harder multi-step reasoning items, and repeated runs for consistency produced 705 responses. All three models answered every routine reasoning and sentiment item correctly under every technique, gave identical answers across three repeated runs (TARa@3 = 100%), and followed every requested output format. On harder items, the accuracy was 96%; all three errors came from one model on a letter-counting item under zero-shot, few-shot, and structured prompts, while CoT and role-based prompts answered it correctly. Summary quality differed significantly across techniques (Friedman $\chi^2(4) = 27.1$, $p < 0.001$), with structured prompting scoring the highest (4.94 of 5). Technique strongly affected response length: CoT responses were on average 2.2 times longer than zero-shot responses. The results indicate a ceiling effect for routine tasks, where prompting technique mainly shapes the length, structure, and quality of answers rather than their correctness.**

***Index Terms*—Prompt engineering, chain-of-thought prompting, large language models, chain-of-thought prompting, zero-shot prompting.**

I. INTRODUCTION

Generative artificial intelligence is used everywhere from research laboratories to everyday businesses and academia. Large language models (LLMs), such as ChatGPT, Gemini, and Claude, are now used to solve problems, summarize documents, classify customer feedback, and support decision-making. Unlike conventional software, an LLM is not programmed for each task; it is instructed in natural language through a *prompt*. The same model may respond very differently to two prompts that ask for the same thing, which has made prompt engineering the deliberate design of model inputs a widely promoted skill [1], [2], [3], [4].

Earlier research has observed large gains from certain techniques. Few-shot prompting enables models to learn a task from a small number of examples [4], [5], and chain-of-thought (CoT) prompting, step-by-step reasoning asked to the model, and it helped to improve arithmetic and logical reasoning [6], [7], [8]. These findings have shaped popular advice on the use of AI tools. Recent studies have shown that the benefits are based on the model and the task to be performed, and explicit prompting is not required for newer models because they already use step-by-step reasoning by default [9] [10].

Two further issues were not explored. First, students and professionals generally use free chat interfaces whose setting cannot be controlled, and most find programmatic access with controlled settings. Second, most studies mentioned accuracy from a single run but the same prompt's answer varies at different attempts [11], [12]. For a user who asks only once, consistency is as important as the accuracy of the answer.

This study reports a controlled comparison of five prompting techniques across three everyday task types on three widely used LLMs, accessed exactly as ordinary users access them. It measures accuracy, consistency across repeated runs, format adherence, summary quality, and response length together, and

adds a small set of harder items to test whether any ceiling effect persists as difficulty increases.

II. OBJECTIVES

The objectives of this study are as follows:

1) To compare how five prompt engineering techniques affect the accuracy and quality of LLM outputs on reasoning, summarization and sentiment classification tasks
2) To verify the consistency of LLM answers after recurring the same prompt for each technique.
3) To investigate whether the effects of the techniques are consistent across the three different LLMs accessed through their free web interfaces.
4) To assess each technique based on response length and adherence to the requested format.
5) To develop prompting guidelines for professionals, students, and educators.

III. RESEARCH QUESTIONS

RQ1: Are the accuracy of LLM answers on everyday reasoning and classification tasks affected by different prompting techniques?
RQ2: Which prompting technique produces high-quality summaries?
RQ3: Do the techniques yield the same answers when a prompt is repeated, and is the most accurate technique also the most consistent?
RQ4: Do the effects of the techniques hold across different LLMs, and to what extent do they increase the response length?

IV. LITERATURE REVIEW

A. Prompting as the Interface to LLMs

Brown et al. [13] showed that a sufficiently large model can perform new tasks from a natural-language instruction or a few examples, a capability termed in-context learning.

Reynolds and McDonell [14] argued that this turns model use into a form of programming through language, and Liu et al. [1] described the resulting "pre-train, prompt, and predict" paradigm.
Kasneci et al. [15] noted that the educational value of LLMs depends on users' ability to formulate tasks and evaluate output, making prompting an everyday skill.

B. Prompting Techniques

1) Zero-shot and Few-shot Prompting:

Few-shot prompting generally improved performance over zero-shot instructions in early models [13], [16], but the gains were unstable: the choice and order of examples could change accuracy dramatically[17], [18], and examples appear to teach mainly the expected format and label space rather than the task itself [19].

2) Chain-of-thought prompting:

Lee et al.[6] showed that step-by-step worked examples improved reasoning, and Kojima et al. [8] obtained similar gains with the instruction "Let's think step by step." Self-consistency, which samples several reasoning paths and takes the majority answer, improved accuracy further and highlighted that a single model can reach different answers on different runs[20], [21]. A meta-analysis of over 100 papers found that CoT helps mainly on mathematical and symbolic problems[7], and tests on recent commercial models reported that CoT now adds little accuracy while substantially increasing response length and time [7].

3) Role-based prompting:

Assigning the model a role improved zero-shot reasoning in some settings[22], but a systematic evaluation of 162 personas found no reliable improvement on factual questions, with effects that were largely unpredictable [23].

4) Structured Prompting:

White et al. [24] catalogued reusable prompt patterns that fix the steps and output templates. Structure aids usability, but strict output formats can reduce reasoning performance [25], and minor formatting choices can change accuracy by large margins [26], [27], [28]. Surveys now catalogue dozens of techniques [29], and recent work stresses that whether a technique helps is contingent on the model, question, and how success is measured [10].

C. Summarization, Sentiment Analysis and Consistency

Human readers preferred summaries from prompted LLMs over earlier fine-tuned systems, whereas automatic metrics such as ROUGE failed to reflect this preference [30]. Instruction-tuned LLMs produce

summaries that are comparable to those of freelance writers [31]. For sentiment analysis, LLMs perform well in simple polarity classification in zero-shot settings but less well in complex cases [32]. Regarding consistency, Atil et al. [33] found accuracy variations of up to 15% across ten runs of five LLMs under supposedly deterministic settings and proposed the total agreement rate of parsed answers (TARa@N), which was the metric adopted in this study.

D. Research Gap

The existing research predominantly evaluates the single-run accuracy of one or two techniques or one task type using programmatic access. Some studies compare several techniques among different everyday tasks using free interfaces on which normal users rely. Few studies consider both consistency and accuracy. Table I summarizes the key studies and their relevance.

TABLE I Summary of Key Related Studies

Ref.	Focus	Key finding
[13]	Few-shot learning	Using few examples model learns the tasks
[17], [18]	Few-shot stability	The order and choice of example affect the accuracy
[6], [8]	Chain-of-thought	Maths and logic improved by progressive reasoning
[7], [8]	CoT on newer models	Gains primarily centered in maths; decreasing value, longer outputs
[22], [23]	Role prompting	Mixed; no reliable overall gain
[24], [25], [27]	Structure and format	Templates are used, format can change the accuracy
[31]	Summarization	Summary given by LLM is compared with human
[33]	Consistency	Accuracy differs upto 15% across runs

V. RESEARCH METHODOLOGY

A. Research Design

In the proposed study design, every test item was passed to each model using each prompting technique. Fig. 1 summarizes the proposed workflow of this study.

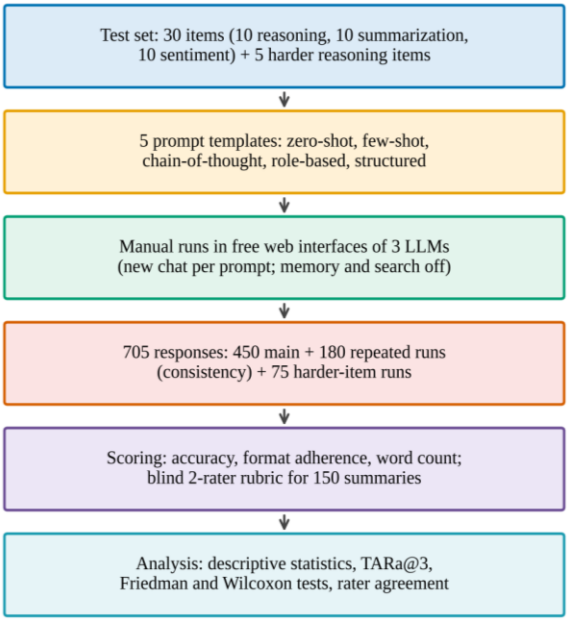


Fig. 1. Research design and workflow of the study

B. Models and Access

Three widely used LLM chatbots were tested through their free web interfaces in a desktop browser: ChatGPT (model shown in the interface as “GPT-6 Instant”), Gemini (shown as “Gemini 4 Hydrogen”), and Claude (shown as “Sonnet 5.5”).

Each model was tested by one team member on one device between November 16 and 21, 2026. Before testing, memory, chat-history referencing, personalization, custom instructions, web search, and connected-app features were disabled. Every prompt was run in a new chat so that earlier conversations could not influence the answer.

C. Prompting Techniques

Five templates were created for each task type (Table II). All templates for a task requested the same final output form (for example, a final line “Answer: <value>”), so that only the technique varied and answers could be scored uniformly. Few-shot examples were separated from the test data.

D. Test Items

The test set comprised 30 original items (Table III). Reasoning items covered arithmetic, percentages, time

and work, series, coding-decoding, and blood relations, including items designed to trap intuitive answers.

TABLE II Prompting Techniques Compared

Code	Technique	Prompt design
T1	Zero-shot	Task instruction only
T2	Few-shot	Instruction with 2–3 solved examples
T3	Chain-of-thought	Asks for step-by-step reasoning before the answer
T4	Role-based	Assigns an expert role (e.g., teacher, editor)
T5	Structured	Specifies steps, rules and exact output template

Passages (about 70 to 90 words) containing main points, key figures, and next steps on various domains such as education, business, agriculture, and technology are provided to get a summary of about 40 words. For the sentiment analysis prompt, straightforward reviews as well as sarcasm, negation, and mixed opinions were used. Difficult reasoning items (letter counting, inclusion-exclusion counting, modular arithmetic, a seating-arrangement puzzle, and work-rate problem) were added after the main results showed the ceiling effect. All the answer keys were verified computationally. To avoid models memorizing text, original passages were provided to the models.

TABLE III Composition of the Test Set and Responses

Component	Items	Responses
Reasoning (Task A)	10	150
Summarization (Task B)	10	150
Sentiment (Task C)	10	150
Consistency repeats (A1, A7, A13, C1, C14, C16; 2 extra runs)	6	180
Harder reasoning items	5	75
Total	35	705

Responses = items × five techniques × three models (× two extra runs for consistency).

E. Evaluation Measures

Accuracy was evaluated as correct or incorrect using answer keys for the reasoning, sentiment, and harder items. was scored as correct or incorrect against the answer key for reasoning, sentiment, and more difficult items.

Format adherence was determined based on the presence of the requested structure (for example, an "Answer:" line or a "Summary:" heading); unrequested closing suggestions were recorded separately. The word count of the full response was the *Response length*.

Consistency was evaluated using TARa@3 [33], which is the proportion of item–technique–model combinations whose parsed answers were identical across three independent runs. Using a five-point rubric covering relevance, completeness, clarity, and factual correctness, using a checklist of the key elements of each passage, two raters rated the *Summary quality*. The raters worked independently on shuffled coded responses and were blinded to the technique.

F. Statistical Analysis

The main runs showed no difference, so accuracy, consistency, and format adherence are reported descriptively. The Friedman test performed on differences in summary quality and response length among different techniques in which each model- item pair considered as a block and then Wilcoxon signed-rank tests with Bonferroni correction ($\alpha = 0.05/10 = 0.005$) was applied. Using the exact agreement and quadratic-weighted Cohen's kappa was used to assess the inter-rater agreement.

VI. RESULTS AND DISCUSSION

A. Accuracy (RQ1)

All three models answered all 10 reasoning and sentiment items correctly under all five techniques, giving 100% accuracy across 300 responses. This included the items designed to trap intuitive answers and sarcastic, negated, and mixed-opinion reviews. Therefore, accuracy did not distinguish between techniques for routine tasks (Table IV).

Routine: 20 items per technique per model. Harder: 5 items per technique per model.

On the harder items, the overall accuracy was 96% (72 of 75). All three errors came from ChatGPT on the letter-counting item (A23), where it answered 12

instead of 13 under zero-shot, few-shot, and structured prompts but answered correctly under chain-of-thought and role-based prompts.

TABLE IV Accuracy by Technique and Model (%)

Technique	Routine (A, C)	Harder: ChatGPT	Harder: Gemini	Harder: Claude
Zero-shot	100	80	100	100
Few-shot	100	80	100	100
Chain-of-thought	100	100	100	100
Role-based	100	100	100	100
Structured	100	80	100	100

Letter counting is a known weakness of LLMs because they process text in sub-word tokens rather than individual characters. Prompts that led the model to work through the sentence explicitly avoided the error, which is consistent with reports that CoT benefits are concentrated in symbolic, step-wise tasks [7]. Gemini and Claude answered all the harder items correctly under every technique.

B. Consistency (RQ3)

After three separate runs (TAR@3 = 100%), all five techniques and three models retrieved identical answers for all 90 combinations of six consistency items. Unlike earlier reports of substantial run-to-run variation [33], no technique introduced inconsistency for these items. As the models improve day by day, accuracy and consistency both showed the ceiling effect for routine tasks, it will be considered open for more difficult tasks in future studies.

C. Format Adherence and Unrequested Content

All 450 main responses followed the requested output structure. However, the models varied in terms of the frequency with which they added unrequested content. Testers recorded closing offers or suggestions (e.g., offers to adjust the length or analyze more reviews) in 132 of 150 Gemini responses (88%) and 33 of 150 ChatGPT responses (22%); none were recorded for Claude’s responses. These added suggestions had no effect on the format but increased reading effort.

D. Summary Quality (RQ2)

Summary quality was high overall (Table V, Fig. 2). Structured prompting achieved the highest mean score for every model (overall 4.94 of 5), followed by chain-of-thought (4.78) and zero-shot (4.72); few-shot (4.58) and role-based (4.61) scored lowest.

The difference across techniques was statistically significant (Friedman $\chi^2(4) = 27.1$, $p < 0.001$, $n = 30$ blocks). After Bonferroni correction, it was showed that the structured prompting ahead of zero-shot ($p = 0.004$), few-shot ($p = 0.001$) and role-based prompting ($p = 0.001$); its advantage over chain-of-thought ($p = 0.014$) was not impactful after correction.

TABLE V Mean Summary Quality Score (1–5) by Technique and Model

Technique	ChatGPT	Gemini	Claude	All
Zero-shot	4.65	4.65	4.85	4.72
Few-shot	4.51	4.34	4.89	4.58
Chain-of-thought	4.76	4.68	4.90	4.78
Role-based	4.49	4.58	4.78	4.61
Structured	4.95	4.94	4.93	4.94

Mean of both raters’ averages across the four criteria; 10 summaries per cell.

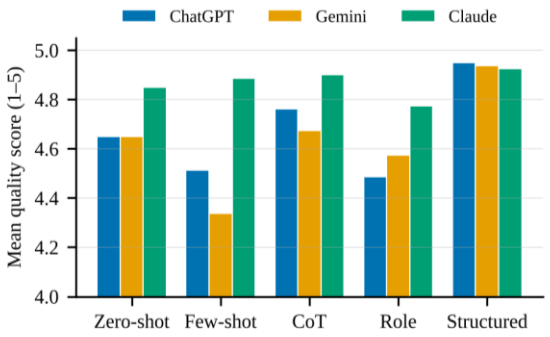


Fig. 2. Mean summary quality by technique and model (y-axis starts at 4.0 to show differences).

The criterion-level scores (Table VI) represent that the advantage of structured prompting was mainly due to completeness, clarity, and factual correctness. The structured template explicitly asked for the main point, key numbers, and any caution, and did not allow adding information that was not in the passage, which directly matched what readers valued in a summary. Claude produced the most uniform quality across techniques, while ChatGPT and Gemini were more sensitive to the prompt.

TABLE VI Mean Rubric Score by Criterion and Technique

Criterion	T1	T2	T3	T4	T5
Relevance	4.87	4.83	4.83	4.77	5.00
Completeness	4.68	4.40	4.78	4.52	4.83
Clarity	4.63	4.58	4.77	4.57	4.97
Factual correctness	4.68	4.50	4.73	4.60	4.95

The inter-rater agreement is presented in Table VII. The exact agreement lay between 69% and 70%, and the weighted kappa remained near zero. This phenomenon called as kappa paradox, occurs when almost all values lie in one category. Here, Rater 1 ratings are almost maximum score 5 and Rater 2 used score 3.

TABLE VII Inter-Rater Agreement

Criterion	Rater 1 mean	Rater 2 mean	Exact agreement	Weighted κ
Relevance	5.00	4.72	86%	0.00
Completeness	4.87	4.41	69%	0.19
Clarity	4.98	4.43	70%	0.00
Factual correctness	4.96	4.43	70%	0.05

The 150 summaries were rated by both raters.

E. Response Length (RQ4)

Technique had a large and highly significant effect on response length (Friedman χ^2 (4) = 159.3, $p < 0.001$, n = 90). Averaged over all tasks and models, chain-of-thought responses contained 112 words, compared with 52 for zero-shot, 49 for few-shot, 72 for role-based, and 80 for structured prompts, making CoT approximately 2.2 times longer than zero-shot. All pairwise differences were significant after correction, except for the structured versus role-based comparison. Fig. 3 shows the patterns by task. For sentiment classification, CoT responses averaged 69 words against 14 for structured prompts five times longer for the same correct label. For summarization, CoT more than doubled the length because it listed the main points before the summary. For the harder items, the gap is increased further; for ChatGPT, CoT responses averaged 180 words against 52 for the zero-shot.

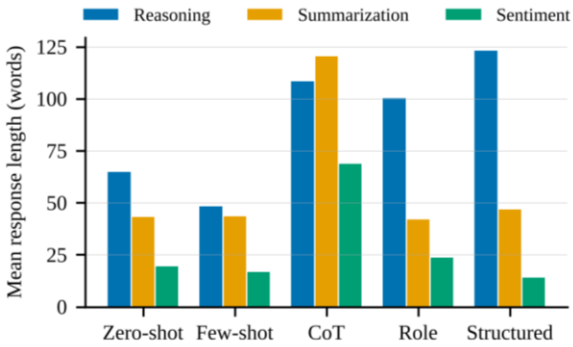


Fig. 3. Mean response length (words) by technique and task, averaged across three models.

The models also differed in terms of text length. ChatGPT provided the shortest zero-shot responses (an average of 32 words), whereas Gemini (65) and Claude (59) provided responses that were approximately double. This indicates that increasing the response length without improving the accuracy increases the reading time of the users and computation time for the providers[7].

F. Discussion

In response to RQ1, the results demonstrate that the prompting technique was not able to change the accuracy of everyday reasoning and sentimental tasks, as all models performed at the ceiling; differences appeared only on a harder letter-counting item, where step-by-step prompting prevented errors. In response to RQ2, structured prompting derived the best summaries, likely because its explicit requirements matched the criteria for a good summary. Regarding RQ3, all techniques provided perfectly consistent answers for everyday items. In response to RQ4, the effect of the technique on quality varied by model, with Claude being the least sensitive, whereas the effect on length was large and consistent across the models.

These findings agree with recent studies showing that current LLMs reason carefully by default; therefore, prompting techniques add less than they did for earlier models [7]. The experimental results suggest that zero-shot instructions are sufficient for everyday questions, and structured prompts are useful for summaries or when the content’s completeness matters, and in counting or multi-step calculation, problems with explicit symbolic work, step-by-step prompts are useful.

VII. LIMITATIONS AND FUTURE SCOPE

This study has several limitations. The test set was small (30 routine and 5 harder items), and current models are able to provide correct results, resulting in a ceiling effect that restricts the ability to differentiate techniques based on accuracy and consistency. Free web interfaces, where settings cannot be controlled and models may be updated without notice, were used for testing. Different team members tested each model on different accounts and devices while following the same settings. Inter-rater agreement on summary quality was limited by a strong ceiling in one rater’s score.

Future work should focus on larger and harder item sets containing different domain-specific tasks in commerce, medicine, law, and computer; compare free interfaces with controlled API access; for consistency check, the number of repeated runs should be increased so that it can detect smaller variations; comparison should be extended to other Indian languages; and examine how engineered prompts differ from those used by non-expert users.

VIII. CONCLUSION

Using free interfaces that students and professionals use, this study evaluated five prompting techniques among three everyday task types on three widely used LLMs. Regardless of the techniques, all models answered every day reasoning and sentiment items consistently correctly, indicating a ceiling effect for everyday tasks. Prompting techniques are still important. Different techniques are useful for different purposes; for example, structured prompting is helpful in deriving better summaries, step-by-step prompting prevented errors on a symbolic letter-counting task, and the response length of chain-of-thought prompts was twice that of zero-shot responses. For current LLMs, prompting matters less for obtaining the correct answer and more for obtaining a complete, well-organized, and concise answer. Users obtain the best results by choosing a technique that suits the task, not by using complex prompts every time.

REFERENCES

[1] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, Jan. 2023, doi: 10.1145/3560815.

[2] M. Xia, “A comprehensive survey of prompt engineering for large language models: Techniques, applications, and future directions,” *Procedia Computer Science*, vol. 287, pp. 1243–1253, Jan. 2026, doi: 10.1016/j.procs.2026.08.169.

[3] P. Sahoo, A. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A systematic survey of prompt engineering in large language models: Techniques and applications,” Feb. 5, 2024, doi: 10.48550/arXiv.2402.07927.

[4] S. Ekin, “Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices,” Apr. 29, 2023, doi: 10.36227/techrxiv. 22683919.v1.

[5] N. Moratelli, M. Barraco, M. Cornia, L. Baraldi, and R. Cucchiara, “Are learnable prompts the right way of prompting? Adapting vision-and-language models with memory optimization,” *IEEE Intelligent Systems*, vol. 39, no. 3, pp. 26–34, May 2024, doi: 10.1109/MIS.2024.3386099.

[6] J. Lee et al., “Chain-of-thought prompting elicits reasoning in large language models,” Jan. 27, 2022, doi: 10.48550/arXiv.2201.11903.

[7] Z. Sprague et al., “To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning,” Sept. 18, 2024, doi: 10.48550/arXiv.2409.12183.

[8] T. Kojima, S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners,” Jan. 2022, pp. 22199–22213, doi: 10.52202/068431-1613.

[9] F. Jin and Y. Tan, “Large language models are multitasking chain-of-thought prompting optimizers,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 37, no. 7, pp. 1–10, Jan. 2026, doi: 10.1109/TNNLS.2025.3644134.

[10] J. Almeida, “Prompt engineering: A comparative study of prompting techniques in AI language models,” Mar. 2025, pp. 1–4, doi: 10.1109/ISEC64801.2025.11147384.

[11] M. Shukla, D. Pandey, S. Kaur, M. Agarwal, A. Goyal, and H. Sharma, “Evaluating the accuracy and explanatory quality of large language models

ChatGPT, Claude, DeepSeek, Gemini, Grok, and Le Chat in statistical test selection for hypothesis testing decisions,” *Cureus*, vol. 17, no. 10, Art. no. e94949, Oct. 2025, doi: 10.7759/cureus.94949.

[12] T. M. Kim, Y. Lee, C. Kim, and T. Ko, “Assessing the proficiency of LLMs with various tasks and evaluators,” *Studies in Health Technology and Informatics*, vol. 316, pp. 552–553, Aug. 2024, doi: 10.3233/SHTI240473.

[13] T. Brown et al., “Language models are few-shot learners,” May 28, 2020, doi: 10.48550/arXiv.2005.14165.

[14] L. Reynolds and K. McDonell, “Prompt programming for large language models: Beyond the few-shot paradigm,” May 2021, pp. 1–7, doi: 10.1145/3411763.3451760.

[15] E. Kasneci et al., “ChatGPT for good? On opportunities and challenges of large language models for education,” *Learning and Individual Differences*, vol. 103, Art. no. 102274, Mar. 2023, doi: 10.1016/j.lindif.2023.102274.

[16] V. Kulikov and R. Neychev, “A brief overview of few-shot prompting in the large language models,” *Zenodo*, May 2023, doi: 10.5281/zenodo.8005317.

[17] M. Li, X. Wang, and M. Yu, “Comparative evaluation of zero-shot and few-shot performance of large language models in low-resource language machine translation,” *JGER*, vol. 3, no. 2, pp. 59–68, Jul. 2025, doi: 10.66372/jger.v3i2.5.

[18] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, “Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity,” Jan. 2022, pp. 8086–8098, doi: 10.18653/v1/2022.acl-long.556.

[19] S. Min et al., “Rethinking the role of demonstrations: What makes in-context learning work?” Jan. 2022, pp. 11048–11064, doi: 10.18653/v1/2022.emnlp-main.759.

[20] J. Ren and K. Zhao, “SERC-PD: Self-evaluation and relative confidence-based path-level reasoning decoding for large language models,” Apr. 2026, pp. 290–295, doi: 10.1145/3829441.3829479.

[21] X. Wang, D. Schuurmans, E. Chi, J. Lee, Q. Le, and D. Zhou, “Self-consistency improves chain of thought reasoning in language models,” Mar. 21, 2022, doi: 10.48550/arXiv.2203.11171.

[22] A. Kong et al., “Better zero-shot reasoning with role-play prompting,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, Jan. 2024, pp. 4099–4113, doi: 10.18653/v1/2024.naacl-long.228.

[23] M. Zheng, J. Pei, L. Logeswaran, M. Lee, and D. Jurgens, “When ‘a helpful assistant’ is not really helpful: Personas in system prompts do not improve performances of large language models,” Jan. 2024, pp. 15126–15154, doi: 10.18653/v1/2024.findings-emnlp.888.

[24] J. White et al., “A prompt pattern catalog to enhance prompt engineering with ChatGPT,” Feb. 21, 2023, doi: 10.48550/arXiv.2302.11382.

[25] Z. R. Tam, C.-K. Wu, Y.-L. Tsai, C.-Y. Lin, H.-Y. Lee, and Y.-N. Chen, “Let me speak freely? A study on the impact of format restrictions on large language model performance,” Jan. 2024, pp. 1218–1236, doi: 10.18653/v1/2024.emnlp-industry.91.

[26] L. Zhou, W. Schellaert, F. Martínez-Plumed, Y. Moros-Daval, C. Ferri, and J. Hernández-Orallo, “Larger and more instructable language models become less reliable,” *Nature*, vol. 634, no. 8032, pp. 61–68, 2024, doi: 10.1038/s41586-024-07930-y.

[27] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, “Quantifying language models’ sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting,” Oct. 17, 2023, doi: 10.48550/arXiv.2310.11324.

[28] Y. Sasaki, H. Washizaki, J. Li, N. Yoshioka, N. Ubayashi, and Y. Fukazawa, “Landscape and taxonomy of prompt engineering patterns in software engineering,” *IT Professional*, vol. 27, no. 1, pp. 41–49, Jan. 2025, doi: 10.1109/MITP.2024.3525458.

[29] O. Fagbohun, R. M. Harrison, and A. Dereventsov, “An empirical categorization of prompting techniques for large language models: A practitioner’s guide,” *JAIMLD*, vol. 1, no. 4, pp. 1–11, Dec. 2023, doi: 10.51219/jaimld/oluwole-fagbohun/15.

[30] M. M. Islam, U. Muhammad, and M. Oussalah, “Evaluating text summarization techniques and factual consistency with language models,” Dec. 2024, pp. 116–122, doi: 10.1109/BigData62323.2024.10826032.

[31] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, “Benchmarking large language models for news summarization,” *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 39–57, 2024, doi: 10.1162/tacl_a_00632.

[32] W. Zhang, Y. Deng, B. Liu, S. Pan, and L. Bing, “Sentiment analysis in the era of large language models: A reality check,” Jan. 2024, pp. 3881–3906, doi: 10.18653/v1/2024.findings-naacl.246.

[33] B. Atil et al., “Non-determinism of ‘deterministic’ LLM settings,” Aug. 6, 2024, doi: 10.48550/arXiv.2408.04667.