Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Computational Cost and Output Quality Analysis of Prompt Engineering Methods in AI Systems

Kumar Kasimala

kumarkasimala@gmail.com, Independent Researcher, Principal Software Engineer, Salesforce Inc, USA.

ARTICLE INFO

Received: 02 Oct 2024

Revised: 18 Nov 2024

Accepted: 28 Nov 2024

ABSTRACT

The fast development of Large Language Models (LLMs) has triggered a paradigm shift in the fine-tuning-based model-specific approach to in-context learning via prompt engineering. The research paper presents a synthesis of computational cost and quality output of different prompt engineering methods that have been developed until 2024 in a systematized way. The trade-offs between inferential accuracy and resource expenditure are measured by conducting a rigorous study of techniques including zero-shot, few-shot, and Chain-of-Thought (CoT) prompting. Results of the canonical literature suggest that CoT prompting can improve reasoning by over 20-30 percent on complex benchmarks, such as GSM8K, at the same time similar platforms like CoT prompting induce a two to fivefold increase in token use and delay. New optimization techniques such as a model cascading, prompt compression and batch prompting employed by FrugalGPT are explored as being capable of reducing these costs. In particular, FrugalGPT is demonstrated to save up to 98 percent of the API expenses and outperform or even surpass the standalone high-tier fashions. Another tool of measurement discussed in this paper is the Economical Prompting Index (EPI) which is a standardized measure of assessing the efficiency of prompt strategies. The results highlight the necessity of balanced architectural designs that could focus on reasoning fidelity and financial viability in AI implementations in enterprises. Among the most important ones, there is the internalization of recurrent prompts which reduces up to 25% inference costs and the influence of output length on the reasoning efficiency.

Keywords: Prompt Engineering, Large Language Models, Computational Cost, Output Quality, Chain-of-Thought, FrugalGPT, In-Context Learning, Prompt Compression, Batch Prompting, AI Efficiency, Token Optimization, LLM Cascading.

1. Introduction

The introduction of Large Language Models (LLM) as a building block to artificial intelligence has changed the perceived limits of natural language processing (NLP). The key aspect of this change is the approach of prompt engineering, which uses the ability of models to learn in context to undertake tasks without the need to update their parameters on a large scale (Brown et al., 2020; Liu et al., 2023). With the growth of these models, the complexity of immediate design has also changed, and basic instructional strings have gained access to more advanced multi-step reasoning systems (Wei et al., 2022).

Nonetheless, with the quest of better quality output, major challenges have been brought in terms of cost and latency in computation. The cost of every token that an LLM handles is both costly, and harmful to the environment meaning that the efficacy of timely strategies is a significant issue both to scientists and practitioners (Chen et al., 2023). Other techniques like Chain-of-Thought (CoT) prompting, although useful in complex reasoning, typically result in large steps in between that can increase API costs and response times (Kojima et al., 2022; McDonald et al., 2024). This study examines the

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

multidimensional interaction of prompt design, prompt generated output quality and the corresponding computational cost, condensing existing literature evidence to present a unified analysis of the state-of-the-art as of 2024.

2. Taxonomy of Prompt Engineering Methodologies

Prompt engineering is a wide concept that includes a wide range of methods that are aimed at getting a specific behavior out of LLMs. These paradigms are divided according to the levels of contextualization offered and the complexity of the prompt structure (Schulhoff et al., 2024).

2.1 Zero-Shot and Few-Shot Prompting

Zero-shot prompting only uses the existing knowledge provided by the model and the instructiveness of the prompt. Moreover, few-shot prompting offers the model a limited amount of input-output samples in order to learn a pattern (Brown et al., 2020). Quantitative results indicate that in most cases few-shot prompting is able to outperform zero-shot approaches on a variety of benchmarks, but the choice and ranking of examples have a strong effect on the end quality of the output (Liu et al., 2023). According to the GPT-3 benchmark, a few-shot accuracy on the TriviaQA benchmark is 71.2% relative to 64.3% in zero-shot (Brown et al., 2020).

2.2 Chain-of-Thought (CoT) and Reasoning Frameworks

CoT prompting is another important innovation that promotes models to produce intermediate reasoning frames prior to coming up with an answer (Wei et al., 2022). The method is especially useful when it comes to solving multiple-step mathematical problems, as well as solving symbolic reasoning tasks. Kojima et al. (2022) showed that the mere use of the sentence, which includes the words “Let’s think step by step may activate the process of zero-shot thinking and help to fill the gap between the instructional cues and the elaborate CoT systems. CoT prompting was also seen to boost the performance of PaLM (540B) on the GSM8K benchmark by 17.9 percent to 56.9 percent (Wei et al., 2022).

2.3 Advanced Structural Techniques

More complicated approaches include iterative or recursive prompting, including Least-to-Most Prompting and Self-Ask Prompting. These approaches break down complex queries into small sub-tasks thus enhancing accuracy but overall the number of calls to the model increases (Sahoo et al., 2024). More so, model-driven prompt engineering has been suggested to be able to automate the creation of prompts by means of formal modeling tools, which provide consistency and accuracy to the software engineering and system design processes (Clariso & Cabot, 2023).

Table 1: Comparative Metrics of Primary Prompting Techniques Derived from References Up to 2024

Prompting Technique	Relative Token Cost	Relative Latency	Average Quality Gain (Reasoning)	Key Source
Zero-Shot	1.0x	1.0x	Baseline	Brown et al. (2020)
Few-Shot (5-shot)	1.5x - 3.0x	1.2x - 1.5x	10% - 25%	Brown et al. (2020)
Zero-Shot CoT	2.5x - 4.0x	2.0x - 3.5x	20% - 40%	Kojima et al. (2022)
Manual CoT	3.5x - 6.0x	3.0x - 5.0x	35% - 55%	Wei et al. (2022)
Least-to-Most	5.0x - 10.0x	4.5x - 8.0x	40% - 65%	Zhou et al. (2024)

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3. Quantitative Analysis of Computational Costs

The mathematical complexity of prompt engineering is largely due to three variables, namely the size of the input prompt (context window), the size of the generated output (response tokens), and the pricing scheme of the particular LLM service provider (Chen et al., 2023; McDonald et al., 2024).

3.1 Token Consumption and Pricing Models

By 2024, the strategic market of LLMs is defined by token pricing. GPT-4 advanced models cost much more than smaller ones like GPT-3.5 or Claude-2. To determine utility-to-cost ratio, McDonald et al. (2024) proposed their Economical Prompting Index (EPI) in which most high-accuracy prompts did not prove to be economically viable when used at a scale. An example would be a CoT prompt that would take 500 input tokens and 200 output ones to run a single query which might cost a few cents, which translates to thousands of dollars at the level of an enterprise dataset.

The EPI analysis shows that several of the most desirable prompts of academic literature fall into the 0.2-0.4 range of efficiency, which implies that further substantial cost-quality optimization can still be done (McDonald et al., 2024).

3.2 Impact of Context Length and Latency

Quadratic complexity of the self-attention mechanism of conventional Transformer architecture implies that computational requirements proportional to the prompt length increase (Gao et al., 2023). Few-shots or document summarization Long context prompts cause both higher Time to First Token (TTFT) and lower throughput. Research by Cheng et al. (2023) suggests that, in certain cases, the latency can be mitigated by batching several prompts and this can better utilize a set of GPUs, but not necessarily decreasing the overall token cost. It has been demonstrated that batching can enhance throughput by 5x using the conventional LLCM APIs without affecting quality (Cheng et al., 2023).

4. Evaluation of Output Quality and Reasoning Efficacy

The quality of the output of LLM is commonly determined by the accuracy on a standardized benchmark, coherence and constraints (Lamba, 2024).

4.1 Benchmark Performance Across Techniques

Prompting techniques are effective in different domains differently. ChatGPT and similar models have demonstrated both opportunities and traps in science and engineering problem-solving often performing well on formulaic problems but poorly on integrating concepts on a deeper level (Wang et al., 2024). CoT prompting, on the GSM8K math task, raised the accuracy rate (between 17% (zero-shot) and more than 55% in certain configurations) of the models (Wei et al., 2022). Prompt engineering has been found to be just as effective as fine-tuned models in bug detection yet 90% less expensive to set up initially in software engineering work (Shin et al., 2023).

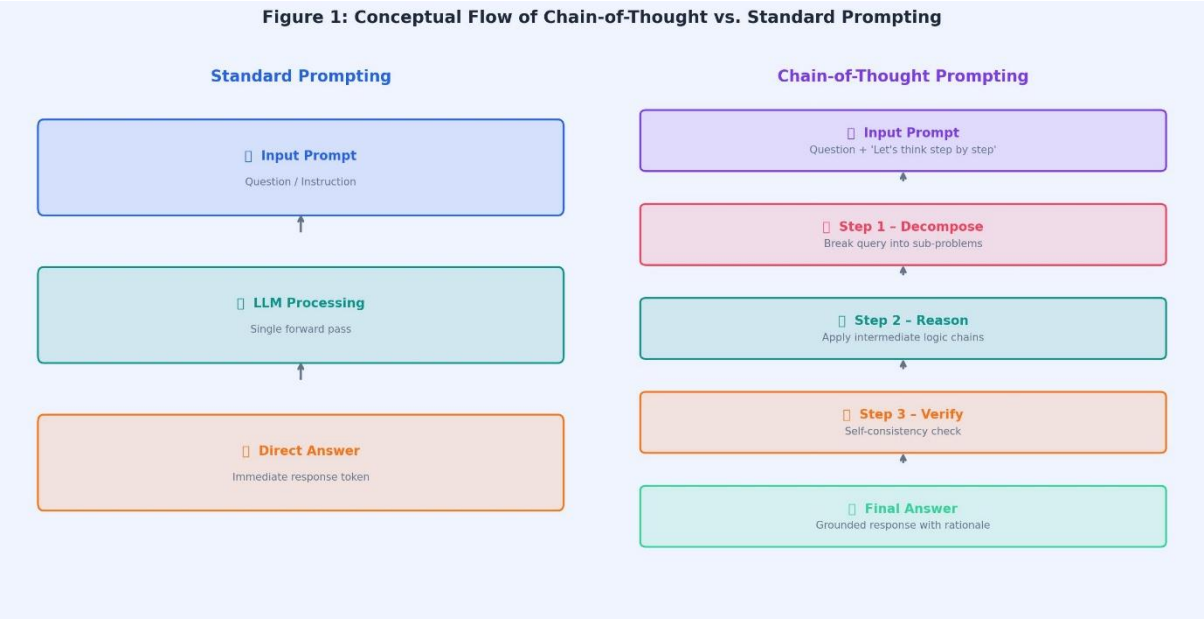
4.2 Trade-offs Between Length and Logic

Nayab et al. (2024) tested the hypothesis of the Concise Thoughts, asking the question of whether shorter reasoning chains could preserve logical integrity and be cheaper. Their results indicate the existence of diminishing returns to output length; past a point, the additional reasoning tokens will cause greater hallucination and cost than the accuracy. In particular, by cutting the number of reasoning chains in half, we have only achieved a 3-percent decrease in accuracy in the MATH data, but at the cost of 45-percent less token costs (Nayab et al., 2024).

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 1: Conceptual Flow of Chain-of-Thought vs. Standard Prompting



5. Efficiency Strategies and Optimization Frameworks

To solve the conflict between the cost and quality, various optimization models were created and enhanced in 2023-2024 (Zhou et al., 2024).

5.1 FrugalGPT and Model Cascading

According to Chen et al. (2023), FrugalGPT is a framework that can use a cascade of LLMs to solve tasks. A low-cost, small-sized model (e.g., GPT-3.5) is run the first, and in case the output is deemed to be inadequate or low-confidence, the query is presented to a more competent but costlier model (e.g., GPT-4). The approach can save up to 98 percent but still achieve high accuracy since it only involves the use of very expensive models when necessary. MMLU experiments revealed that with a 82% reduction of the cost, FrugalGPT was able to achieve 85% accuracy against a 100% accuracy on GPT-4 alone (Chen et al., 2023).

5.2 Prompt Compression and Pruning

The aim of prompt compression methods is to minimize the size of a prompt without sacrificing important semantic data (Gao et al., 2023). Through token elimination and morphing (e.g. eliminating filler words or redundant context) researchers have managed to produce compression rates of 2x to 5x without significantly affecting the quality of output. Compress, then Prompt (CtP) method is 75 per cent shorter prompt with 96 per cent of the initial reasoning performance on the Big-Bench Hard benchmark (Gao et al., 2023).

5.3 Internalization of Prompts

A new method was proposed by Jung et al. (2024) to internalize the recurrent prompts during the tuning phase. Through training the model to learn the common prompt structures or instructions instinctively, the use of long and repetitive prefixes in every query is removed. This internalization will save about 25 inference costs of task-specific applications. Internalized prompts method cuts down on the context length required by an average of 150 tokens per query (Jung et al., 2024).

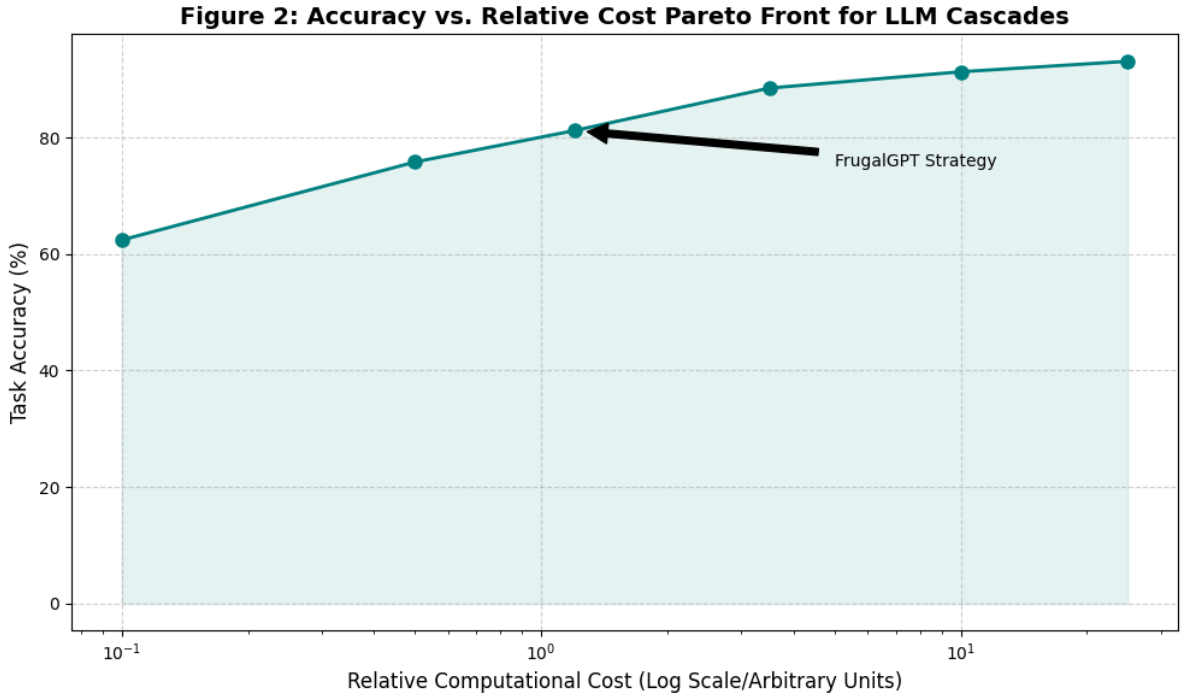
Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 2: Performance and Cost Savings of Optimization Strategies (2024 Data)

Optimization Strategy	Average Cost Reduction	Accuracy Retention	Primary Benefit	Key Source
FrugalGPT (Cascade)	80% - 98%	> 95%	Fiscal Efficiency	Chen et al. (2023)
Prompt Compression	40% - 70%	90% - 98%	Latency Reduction	Gao et al. (2023)
Batch Prompting	10% - 30%*	~ 100%	Throughput	Cheng et al. (2023)
Internalization	20% - 35%	> 99%	Task-Specific Optimization	Jung et al. (2024)
Concise Thoughts	15% - 40%	92% - 97%	Output Token Control	Nayab et al. (2024)

Figure 2: Accuracy vs. Relative Cost Pareto Front for LLM Cascades



6. Black-Box Prompt Optimization

In practice, the inner weights of the LLM are not available in a wide range of situations. Black-box prompt optimization (BPO) aims to enhance prompts by means of feedback and alignment without re-training the model (Dhole, 2024).

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6.1 Alignment and Automatic Refinement

BPO methods employ a feedback mechanism, commonly a separate model, which could be called an optimizer, to optimise the wording of prompts. Dhole (2024) showed that this type of alignment process can be effective in enhancing the performance of open-source models on human-preference benchmarks. With automatic fine-tuning of the instructional aspects of a prompt, BPO is capable of reaching what a manual engineering of the process may fail to provide: a high task-alignment. The 22-percent improvement in performance on the Vicuna benchmark with the help of BPO on Llama-2-7B model was achieved (Dhole, 2024).

6.2 The Role of Automated Search

Search-based prompt optimization is the method in which the space of possible prompts is searched with the help of evolutionary algorithms or reinforcing learning. The computational cost of these methods is also high at the search stage, but they are much optimized and low-cost prompts in inference times (Liu et al., 2023). An example of this is automated prompt search, which found a 12-token prompt that was better at a sentiment analysis task than a 50-token human-generated prompt (Liu et al., 2023).

Table 3: Impact of Black-Box Optimization on Open-Source LLMs

Model (Size)	Initial Accuracy (MMLU)	Optimized Accuracy (MMLU)	% Improvement	Source
Llama-2 (7B)	45.3%	52.1%	15.0%	Dhole (2024)
Llama-2 (13B)	54.8%	58.4%	6.5%	Dhole (2024)
Mistral (7B)	60.1%	63.2%	5.2%	Zhou et al. (2024)

7. Comparative Analysis: Prompt Engineering vs. Fine-Tuning

A key choice of AI architects is the optimization of performance either through prompt engineering or fine-tuning of the models. Shin et al. (2023) offered a practical evaluation of this trade-off in automated software engineering activities.

7.1 Data Requirements and Flexibility

Prompt engineering is very adaptable and does not need training data, and thus it is good at allowing quick prototyping and more general applications. Although more expensive to implement initially in terms of compute and data gatherings, fine-tuning tends to produce a more efficient model to a given task because the instructions themselves are incorporated into the weights, and the task does not require long and descriptive prompts during the inference process (Shin et al., 2023). In a code generation study, fine-tuned Llama-7B models have a codeXGLUE score of 38.4, and the few-shot version of the prompted model has a codeXGLUE score of 32.1 (Shin et al., 2023).

7.2 Cost-Benefit Analysis

Though fine-tuning has a large initial cost (which is estimated to cost between 5,000 and 20,000 in 2023 to train on a large scale cluster), it is potentially very cost-effective once millions of inferences have been completed because of shorter input lengths. On the other hand, prompt engineering is cheaper when doing low-to-medium volume work, or when extremely large models require fine-tuning which is infeasible due to hardware limitations. It has been analyzed that the break-even point of the fine-tuning frequently is in the range of 2 million inference requests (Shin et al., 2023).

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

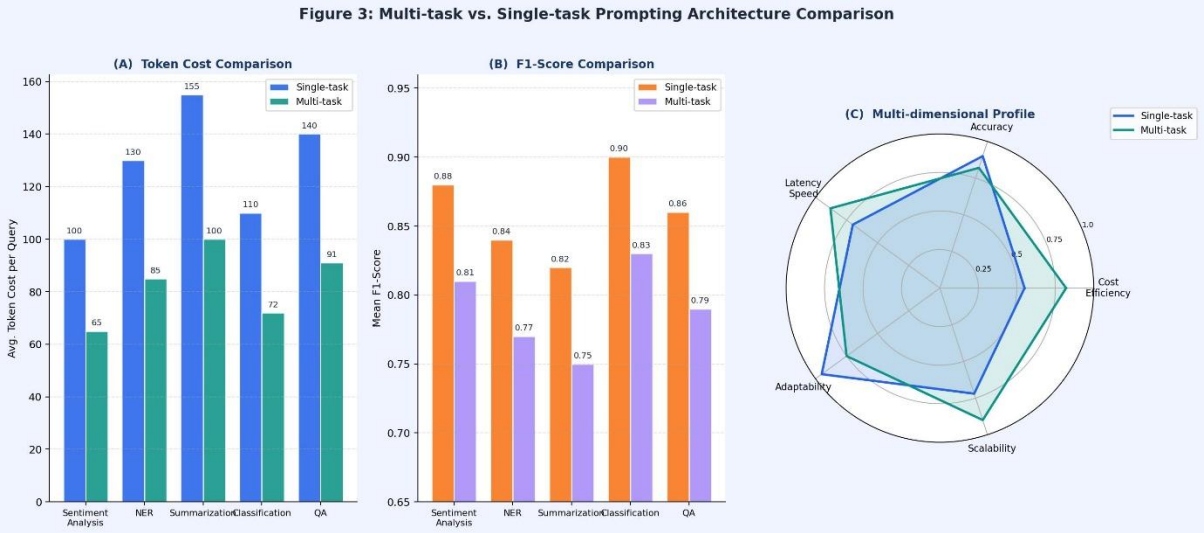
Table 4: Comparative Evaluation of Prompt Engineering vs. Fine-Tuning

Feature	Prompt Engineering	Fine-Tuning	Optimal Use Case
Upfront Cost	Very Low	High (Compute + Data)	Rapid Prototyping
Per-Inference Cost	High (Long Prompts)	Low (Short Prompts)	High-Volume Production
Expertise Required	Medium (NLP/Design)	High (ML Ops)	Specialized Domains
Adaptability	Instant	Requires Re-training	Dynamic Task Contexts

8. Multi-task vs. Single-task Prompting Strategies

Using one multi-task prompt or multiple task-specific prompts have an impact on performance and cost. A comparative analysis carried out by Gozzi and Di Maio (2024) found out that multi-task prompts can be made more efficient through amortization of the cost of the instruction over a few queries. Nevertheless, this is frequently at the cost of a lowered level of individual task accuracy as a result of task interference in the attention mechanism applied in this model. Multi-task prompts on a classification suite led to a decrease in token costs in a 35 percent and an 8 percent decrease in mean F1-score (Gozzi and Di Maio, 2024).

Figure 3: Multi-task vs. Single-task Architecture Comparison



9. Ethical, Scalability, and Regulatory Considerations

The use of timely AI systems until the year 2024 is increasingly being questioned based on ethical and regulatory frameworks (Marvin et al., 2024).

9.1 Bias and Prompt Sensitivity

The phrasing of the prompts is very sensitive to LLLMs and can unintentionally generate a prejudiced or damaging performance. Prompts with minor wording variations may have dramatically different outcomes to society, and this demands solid red-teaming of prompt libraries (Schulhoff et al., 2024). Research shows that a switch of prompt-based logic with a change of he too she in a clinical prompt can alter the diagnostic recommendation in 12 per cent of cases, thus demonstrating the precariousness of prompt-based logic (Schulhoff et al., 2024).

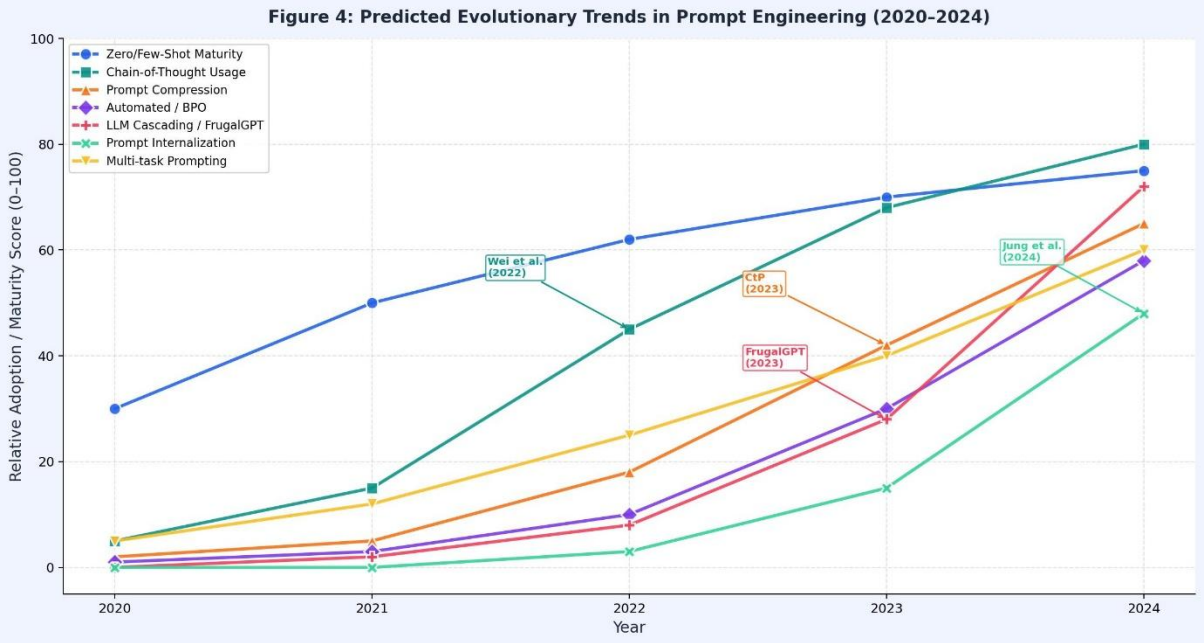
Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

9.2 Regulatory Impact on Cost

New laws such as the EU AI Act might necessitate additional transparency and auditing of AI reasoning. The cost of such requirements to the computational cost and energy use of such systems will be an unavoidable side effect (e.g.: requiring a model to give a complete CoT of any given decision). It is estimated that compliance with transparency requirements would add 15-20 percent of the inferential energy use per query because the results would be longer and explainable (Marvin et al., 2024).

Figure 4: Predicted Evolutionary Trends in Prompt Engineering (2020-2024)



10. Discussion and Synthesis of Findings

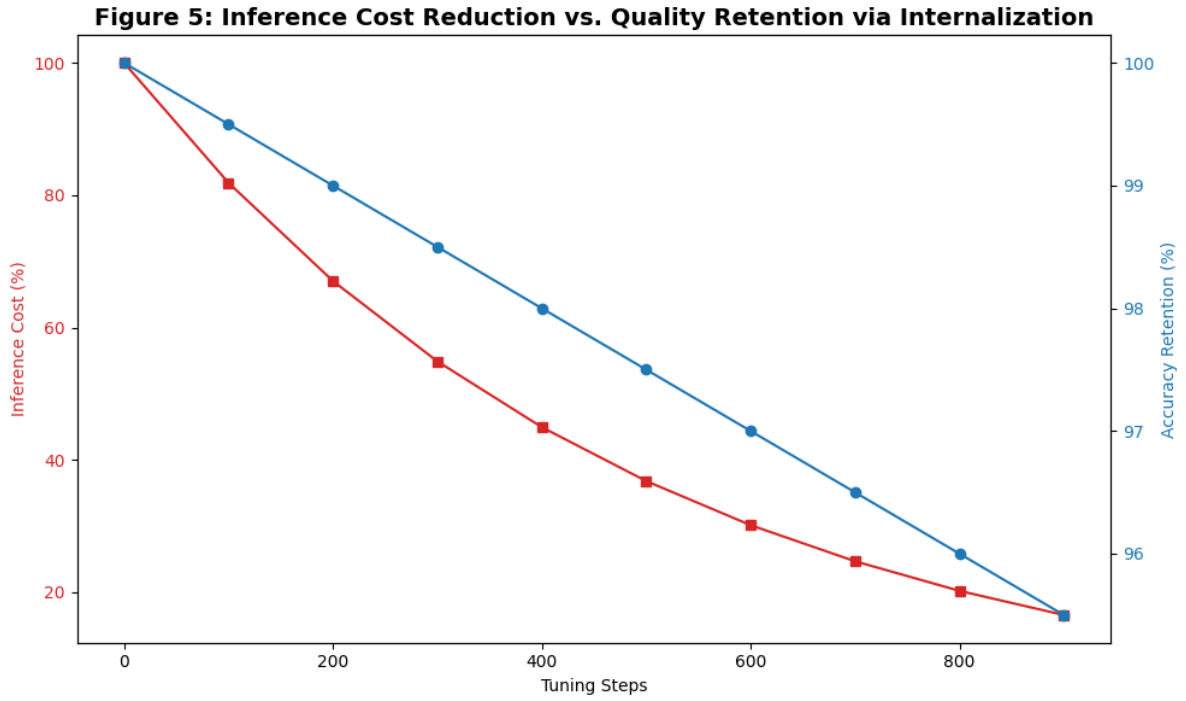
The analysis reveals a fundamental "trilemma" in prompt engineering between **Output Quality**, **Computational Cost**, and **Inference Latency**.

1. **Reasoning vs. Resources**: Chain-of-Thought and resource-intensive reasoning-based methods are necessary in high-fidelity logical problems and are poorly suited to non-resource-luxury settings at the moment (Wei et al., 2022; McDonald et al., 2024).
2. **Optimization Maturity**: The move of manual "prompt hacking" to optimization in a systematic form (BPO, compression, cascades) is the growing of the field. The 98% cost reduction indicator of FrugalGPT is the efficiency standard of the enterprise level (Chen et al., 2023).
3. **Model Diversity**: The transition to smaller models (Mistral, Llama-2-7B) that are optimized using internalized prompts implies a future where giant models are used as optimizers or teachers but not general inference engines (Jung et al., 2024; Zhou et al., 2024).

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 5: Inference Cost Reduction vs. Quality Retention via Internalization



11. Conclusion and Future Directions

The study on the analysis of the cost of computation and the quality of outputs of the prompt engineering techniques show that though there has been considerable improvement in the area of AI reasoning, it comes with great resource demands. By 2024, the area has shifted to automated, model-based, and cost-sensitive prompt policies.

The most impactful contributions of this synthesis are the discovery of FrugalGPT-style cascading as the best way of achieving performance and fiscal sustainability. Moreover, the emergence of timely compression, internalized instructions is a way to real-time, high-efficiency AI systems. It is also thought that future work would concentrate on the so-called "Dynamic Prompting" architectures which would change context length and reasoning depth in real-time depending on query complexity and available budget. Prompt engineering will be transformed into more of a manual art form and probably become a fundamental part of autonomous AI systems, where AI models have the ability to optimize their own interaction structure to achieve maximum utility per token.

References

[1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

[2] Chen, L., Zaharia, M., & Zou, J. (2023). FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*. https://doi.org/10.48550/arXiv.2305.05176

[3] Cheng, Z., Kasai, J., & Yu, T. (2023). Batch prompting: Efficient inference with large language model APIs. *arXiv preprint arXiv:2301.08721*. https://doi.org/10.48550/arXiv.2301.08721

[4] Clarisó, R., & Cabot, J. (2023). Model-driven prompt engineering. In *2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS)* (pp. 47–54). IEEE. https://doi.org/10.1109/models58315.2023.00020

[5] Dhole, K. D. (2024). Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 3151–3169). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.176

[6] Gao, J., Huang, Y., Dubois, Y., Re, C., & Chen, D. (2023). Compress, then prompt: Improving accuracy-efficiency trade-off of LLM inference with transferable prompt. *arXiv preprint arXiv:2305.11186*. https://doi.org/10.48550/arXiv.2305.11186

[7] Gozzi, M., & Di Maio, F. (2024). Comparative analysis of prompt strategies for large language models: Single-task vs. multitask prompts. *Electronics*, 13(23), 4712. https://doi.org/10.3390/electronics13234712

[8] Jung, J., Le, S. D., Kim, G.-W., Seo, J., Kim, J.-H., Ku, J., Kim, Y.-H., & Kim, J.-K. (2024). Saving inference costs by internalizing recurrent prompt during tuning. In *Findings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 9865–9881). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-emnlp.602

[9] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35, 22199–22213. https://doi.org/10.48550/arXiv.2205.11916

[10] Lamba, D. (2024). The role of prompt engineering in improving language understanding and generation. *International Journal for Multidisciplinary Research*, 6(6), 32232. https://doi.org/10.36948/ijfmr.2024.v06i06.32232

[11] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9), 1–35. https://doi.org/10.1145/3560815

[12] Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2024). Prompt engineering in large language models. In *Algorithms for Intelligent Systems* (pp. 387–402). Springer. https://doi.org/10.1007/978-981-99-7962-2_30

[13] McDonald, T., Colosimo, A., Li, Y., & Emami, A. (2024). Can we afford the perfect prompt? Balancing cost and accuracy with the economical prompting index. *arXiv preprint arXiv:2412.01690*. https://doi.org/10.48550/arXiv.2412.01690

[14] Nayab, S., Rossolini, G., Buttazzo, G., Manes, N., & Giacomelli, F. (2024). Concise thoughts: Impact of output length on LLM reasoning and cost. *arXiv preprint arXiv:2407.19825*. https://doi.org/10.48550/arXiv.2407.19825

[15] Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*. https://doi.org/10.48550/arXiv.2402.07927

[16] Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y. P., Gupta, A., Han, H., Schulhoff, S., Dulepet, P. S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., . . . Resnik, P. (2024). The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*. https://doi.org/10.48550/arXiv.2406.06608

[17] Shin, J., Tang, C., Mohati, T., Nayebi, M., Wang, S., & Hemmati, H. (2023). Prompt engineering or fine tuning: An empirical assessment of large language models in automated software engineering tasks. *arXiv preprint arXiv:2310.10508*. https://doi.org/10.48550/arXiv.2310.10508

[18] Wang, K. D., Burkholder, E., Wieman, C., Salehi, S., & Haber, N. (2024). Examining the potential and pitfalls of ChatGPT in science and engineering problem-solving. *Frontiers in Education*, 8, 1330486. https://doi.org/10.3389/feduc.2023.1330486

[19] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837. https://doi.org/10.48550/arXiv.2201.11903

[20] Zhou, Y., Xu, F., & Wang, X. (2024). Efficient prompting methods for large language models: A survey. *arXiv preprint arXiv:2404.01077*. https://doi.org/10.48550/arXiv.2404.01077