ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

A Multi-Task Benchmark for Prompt Engineering in Large Language Models: Performance, Efficiency and Reliability Trade-offs

Shifa Shah[1]; Kumkum Mishra[2]; Devesh Kumar Gola[3]

[1,2,3]Department of SSCSE, Sharda University, Greater Noida

Publication Date: 2026/04/29

Abstract: A major method of steering large language models (LLMs) through various natural language processing (NLP) tasks without fine-tuning is prompt engineering. Nevertheless, the current research frequently considers the techniques of prompting separately, which restricts the knowledge of their generalizability. The paper is a multi-task benchmark that measures ten prompt engineering strategies on three NLP tasks: requirement classification, sentiment analysis, and topic classification. The tested approaches are the zero-shot, few-shot, chain- of-thought (CoT), role-based, and structured prompting. Accuracy and F1 scores are used to measure the performance, and latency and invalid output rate are used to measure the efficiency and reliability. Findings indicate that prompt design significantly impacts model performance. The structured and CoT prompting invariably outperform the zero-shot ones, but few-shot prompting does not necessarily. Task-specific analysis determines that simple tasks are less susceptible to prompt variations, and complex tasks rely more on prompt design. Additionally, a clear trade-off between performance and efficiency is observed. These results provide insights for the development of effective, robust, and scalable prompt engineering solutions to real-world LLM applications.

Keywords: *Prompt Engineering, Chain-of-Thought Prompting, Few-Shot Learning, Zero-Shot Learning, NLP Classification, Macro-F1 Score, Latency Analysis, Green AI, Structured Prompting.*

How to Cite: Shifa Shah; Kumkum Mishra; Devesh Kumar Gola (2026) A Multi-Task Benchmark for Prompt Engineering in Large Language Models: Performance, Efficiency, and Reliability Trade-offs. *International Journal of Innovative Science and Research Technology*, 11(4), 2432-2439. https://doi.org/10.38124/ijisrt/26apr1597

I. INTRODUCTION

Such models, known as Large Language Models (LLMs), have improved the field of natural language processing (NLP) and performed highly on a diverse set of tasks by enabling generalization to question answering, reasoning, and text classification (without needing fine-tuning on a task). These models have the benefit of being pretrained on mass tasks and, with natural language instructions, are able to adapt to downstream tasks with the benefit of flexible and scalable execution across domains [1],[11]. Nevertheless, their behavior is highly sensitive to the design of input prompts, making prompt engineering an important research direction that aims at coming up with efficient prompts that can direct the behavior of the models. Recent research has revealed that prompt design is significant in determining LLM outputs. One can use methods like zero-shot prompting, few-shot prompting, so that models can generalize on the basis of task descriptions and in-context examples, and various methods like chain-of-thought (CoT) prompting facilitate reasoning by promoting mid- steps in the decision-making process [2], [3]. Structured prompting strategies such as role-based prompting and schema-constrained output have also been proposed to enhance the consistency of the output and control [6],[7]. Although these developments have been made, the bulk of the research to date gauges prompt strategies independently or on individually-defined problems, thus restricting the knowledge of whether the strategies in general are applicable across various NLP tasks [13]. Besides performance, efficiency, and reliability are crucial in the practical application of LLMs. The majority of previous studies concentrate on the accuracy and F1-score as key metrics to expect; however, such factors do not directly influence usability in the practical sense. Latency and output validity are other important factors that are overlooked in most studies when addressing the usability of real-life applications. Inefficient prompts may increase the number of tokens used and the computing cost, whereas unreliable outputs in violation of the desired format undermine the trustworthiness of the system. In response to concerns over the efficacy of AI in areas like computer simulation research, current studies on AI sustainability and effectiveness focus more on combining performance with computational and energy efficiency at a large scale [16], [17].

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

In order to eliminate these difficulties, the paper will suggest a multi-task benchmark framework that will be used to assess prompt engineering strategies in various NLP tasks. In particular, we examine 10 types of prompts in 3 tasks, such as classification of requirements, sentiment analysis, and classification of topics. Besides the traditionally known metrics of evaluation (accuracy and Macro-F1), we have efficiency metrics (latency and invalid output rate) that give a more comprehensive picture of prompt effectiveness. With a mixture of task diversity and prompt diversity, it is expected that this work will offer more information on the effects prompt design has on performance and efficiency in LLMs [5], [18].

➢ *The Main Contributions of This Paper are As Follows:*

- A 10×3 benchmark analysis of prompt engineering strategies across multiple NLP tasks.
- Both performance and efficiency measures are included, such as latency and invalid output rate.
- Empirical task-dependent prompt effectiveness.
- Determination of major trade-offs among performance, reliability, and computational efficiency.

Although there has been recent progress, the current literature assesses prompt engineering methods either individually or in single-task scenarios, which does not give information on how they may be applied in a variety of NLP tasks. Also, the aspects of efficiency and reliability, like latency and output validity, are not taken into consideration. To fill in these gaps, the paper presents a benchmark of multi- tasks, a composite measure of assessing prompt strategies based on the performance, efficiency, and reliability metrics. As far as we know, this is one of the first studies to collectively measure prompt strategies across tasks based on performance, efficiency, and reliability measures.

II. LITERATURE REVIEW

The idea of prompt engineering has become a new paradigm of adjusting big language models (LLMs) to downstream tasks without fine-tuning. Initial experiments have shown that LLMs are capable of solving problems when given natural language instructions and a small amount of examples, and this is the foundation of zero-shot and few-shot learning [1]. Later research, however, reveals that few-shot efficiency is contingent on the quality of examples, order, and relevance, meaning that it is not necessarily always useful [5].

Chain-of-thought (CoT) prompting was proposed to maximize the reasoning ability, as it allows models to produce intermediate reasoning steps and then final outputs [2]. Robustness can be enhanced by extensions like self-consistency that combine multiple reasoning paths [4], or zero-shot reasoning can be used to show that structured prompts alone can induce reasoning behavior [3]. These findings highlight the importance of prompt structure in guiding model performance. Moreover, well-organized prompting methods, such as role-based prompting and schema-constrained outputs, have been suggested to enhance the reliability of output and regulate it [6], [7]. Prompt-based programming frameworks also consider prompts as modules to regulate model behavior. These methods, however, tend to get more complex and do not necessarily ensure that output constraints are adhered to. As well, recent studies have investigated parameter-efficient prompt tuning, where the continuous prompt representations are optimized to tune models efficiently [19], [20]. Although these approaches are contrasted with the manual prompting, they highlight the general significance of the prompt design. Likewise, multitask prompting has also been found to enhance generalization among tasks [18]. Evaluation-wise, the available benchmarks can determine the performance of the LLM on a variety of tasks, but seldom is a prompt design a variable [14]. In addition, recent research in Green AI emphasizes that it is important to trade off between performance and computational efficiency because longer prompts consume more tokens and are costly inferences [16], [17]. Nevertheless, there is a lack of research on how prompt design, efficiency, and reliability are related.

In general, previous studies have mainly concentrated on single prompting methods and individual task assessments with little concern over efficiency and consistency in output. This drives the necessity to have a multi-task benchmark that thoroughly assesses prompt strategy over tasks and integrates performance and efficiency measures, which is the aim of this work.

Overall, prior work lacks a unified evaluation framework that jointly considers performance, efficiency, and reliability across multiple tasks. This limitation motivates the need for a comprehensive multi-task benchmark, as proposed in this study.

III. METHODOLOGY

➢ *Experimental Design*

The study will be structured as an experimental design to test the influence of prompt engineering strategies on the performance of large language models (LLMs) in various tasks. In particular, we develop a 10 x 3 assessment program, comprising 10 different types of prompts and 3 natural language processing (NLP) activities. The prompt types cut across various paradigms, such as zero- shot prompting, few-shot learning, chain-of- thought reasoning, role-based prompting, and structured output prompting. The methods are some of the most popular methods of prompt engineering in the literature [6], [7]. The selection of diverse prompt strategies and tasks is intended to ensure a balanced evaluation of generalization, task complexity, and prompt sensitivity across different NLP scenarios. The assessment is carried out using three classification tasks: requirement classification, sentiment analysis, and topic classification, to make sure the data characteristics and task complexity are diverse. The multi-task design allows an in-depth examination of the generalization of prompt strategies

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

across domains and different levels of difficulty. Moreover, experiments are conducted in controlled conditions, where the temperature is set, and the outputs are deterministic to reproducibly conduct and compare prompt types fairly.

➢ *Datasets*

Three benchmark datasets are used to assess the prompt performance in a variety of NLP scenarios. Requirements classification is done on the PROMISE dataset, where instances are classified as either functional or non-functional requirements. Sentiment analysis is implemented on the NLTK movie reviews dataset, which is a popular benchmark for binary classification. To classify the topics, the 20 Newsgroups dataset is first subsetted to four groups: science/technology, sports, politics, and atheism. The combination of these datasets offers a fair assessment of all structured, subjective, and multi-classification tasks, giving the experimental results strength.

Table 1 Dataset Description

Dataset	Task	Class	Description
PROMISE	Requirement Classification	2	Functional vs Non-functional requirements
NLTK Movie Reviews	Sentiment Analysis	2	Positive vs Negative sentiment
20 Newsgroups	Topic Classification	4	Sci tech, sports, politics, atheism

➢ *Prompt Engineering Strategies*

This paper presents an evaluation of ten prompt engineering strategies that reflect a wide range of prompting paradigms. Zero-shot prompting involves the use of just task descriptions, whereas few-shot prompting involves the use of labeled examples to provide guidance to model predictions. Chain-of-thought (CoT) prompting involves adding intermediate reasoning to improve decision-making [2]. Instruction-based prompting involves giving very specific and clear instructions about the task to be performed, whereas role-based prompting involves giving the model an expert role to enhance contextual knowledge. Lastly, JSON schema prompting mandates structured production through restricting the format of the response. The combination of these strategies allows conducting a thorough assessment of prompt design at different degrees of complexity and control.

Table 2 Prompt Types Used in Study

Category	Prompt Type	Description
Zero-shot	minimal	Basic instruction
Zero-shot	definition	Task definition provided
Zero-shot	label_only	Only labels given
Few-shot	few_shot_2	2 examples
Few-shot	few_shot_2	4 examples
Reasoning	cot_brief	Short reasoning Reasoning
Reasoning	cot_structured	Step-by-step Reasoning
Instruction	detailed_instruction	Explicit instructions
Role-based	role_expert	Expert persona
Structured	json_schema	JSON output format

➢ *Evaluation Metrics*

A combination of effectiveness, efficiency, and reliability measures is used to evaluate model performance. The accuracy, the macro-F1 score, and the weighted-F1 score are used to measure the effectiveness and give a balanced assessment of the class distribution [14]. It measures efficiency by the mean latency per request, which is a measurement of the computational cost of any type of prompt. The reliability can be assessed using the invalid output rate, which is the percentage of nonconforming responses to the anticipated label formats. It is a multi-dimensional evaluation framework that will take into consideration an extensive assessment of prompt performance beyond conventional accuracy-based measures.

➢ *Experimental Pipeline*

The experimental pipeline is made up of various steps that guarantee a systematic and reproducible analysis of prompt engineering approaches. These phases involve dataset pre-processing, stratified sampling, generation of prompts, API-based inference, parsing of output, and the computation of metrics. First, preprocessing of datasets is done to standardize text formats and eliminate inconsistencies to provide consistent input to all tasks. In every task, stratified sampling is used to achieve a balance in class representation in the evaluation set. This will assist in avoiding biases concerning predominant classes and will guarantee that performance measurements are not biased and are representative of model behavior. When few- shot prompting is applied, representative examples are sampled out of the training data to give contextual advice to the model. Prompts are dynamically built according to the chosen prompt type and description of the task, using predefined templates. Instructions, label constraints, and, where necessary, examples or reasoning steps are included within these templates. An API interface is then used to query the LLM under a deterministic configuration, and measure the response and latency to evaluate efficiency. The rule-based and

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

regex-based methods are used to carry out output parsing and isolate the predicted labels of model responses. Structured outputs like JSON are handled specially, and fallback mechanisms are provided when the outputs are not of the expected format. Lastly, the evaluation metrics, such as accuracy, Macro-F1, weighted- F1, invalid output rate, and mean latency, are estimated and summed up between prompt types and tasks to make comparative analysis generally possible.

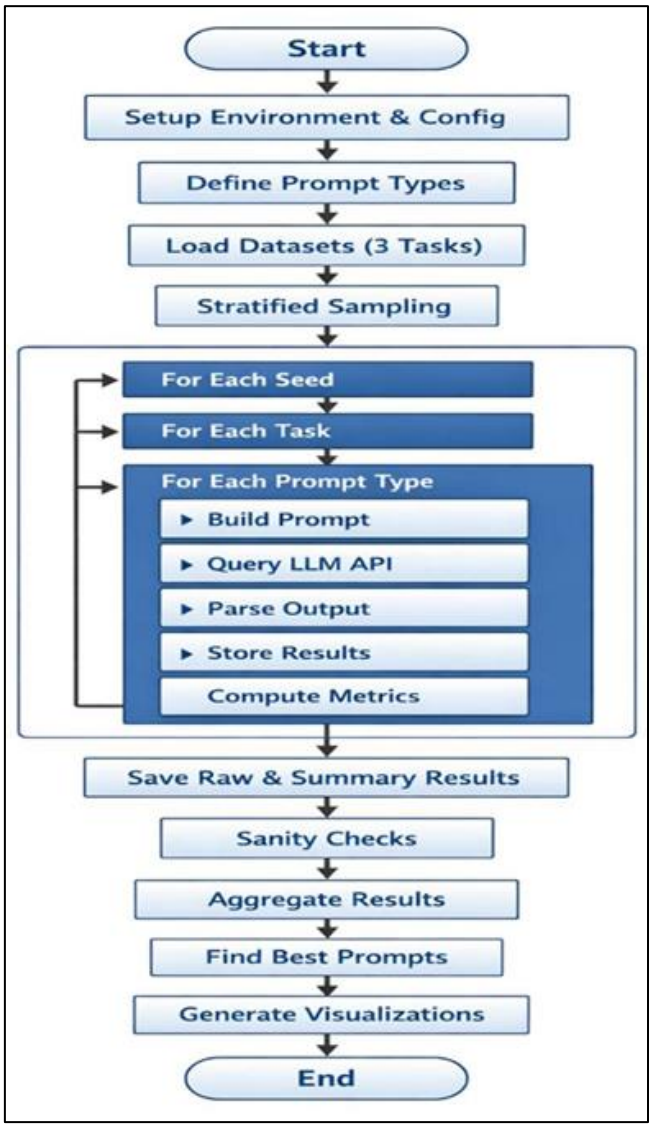

Fig 1 Experimental Pipeline for Multi-Task Prompt

Evaluation

➢ *Implementation Details*

Every experiment is carried out in a deterministic setup with the temperature kept at zero so that the results are consistent. The test is carried out through a pilot run, a deterministic number of samples per task, and a single random seed. The system allows multi-seed assessment and scalability in sample sizes in the future, in case of large-scale experiments. The execution utilizes basic Python packages, such as NumPy, Pandas, scikit-learn to process and evaluate data and Matplotlib and Seaborn to perform data visualizations. The API uses a REST-based interface and facilitates the integration with the LLM backend smoothly.

IV. RESULTS AND ANALYSIS

Table 3 shows the overall performance of various prompt strategies in all tasks. The assessment involves accuracy, macro-F1, weighted-F1, and mean latency, which gives a thorough comparison of the prompt effectiveness, efficiency, and reliability. This integrated evaluation allows a systematic evaluation of trade-offs in performance and computational cost in large language models (LLMs) [6], [14].

Table 3 Aggregated Performance Metrics (Top Prompts per Task)

Task	Promp t Type	Accura cy	Macro- F1	Latenc y (s)
Requirements	JSON Schema	0.667	0.667	2.72
Requirements	CoT (Brief)	0.583	0.580	2.68
Requirements	Detailed Instruction	0.583	0.580	2.72
Sentiment	CoT (Brief)	0.667	0.657	2.69
Sentiment	Detailed Instruction	0.583	0.580	2.69
Sentiment	Few- shot (2)	0.583	0.556	2.75
Topic	Role Expert	0.500	0.433	2.68
Topic	JSON	0.333	0.359	2.70
Topic	Few- shot (4)	0.333	0.338	2.65

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

The findings indicate substantial variation across tasks and prompt types. Structured prompting (JSON schema) is most successful for requirement classification (Macro-F1 ≈ 0.667), underscoring the usefulness of structured output formats for structured tasks. Chain-of-thought (CoT) prompting is most effective for sentiment analysis (Macro-F1 ≈ 0.657), indicating that even in relatively simple classification scenarios, reasoning-based prompts can be beneficial. Conversely, topic classification consistently performs worse across all prompt types, highlighting the limitations of prompt engineering for complex multi-class tasks. These results align with previous studies showing that LLM performance varies across tasks [13], [14].

➢ *Overall Prompt Performance*

Figure 2 presents the general ranking of prompt strategies in terms of the mean Macro-F1 scores for tasks. Chain-of-thought and instruction-based prompting have the best overall performance, and this shows that explicit directions and guided reasoning are highly effective in enhancing model predictions. Role prompting demonstrates similar performance, which again validates the significance of contextual framing. In contrast, minimal and label-only zero-shot prompts perform worse due to the lack of contextual information. It is important to note that the use of few-shot prompting is not always superior to the use of simpler methods, which implies that the use of examples is not sufficient. This is in line with earlier results that the quality and relevance of examples are more important than quantity [5].

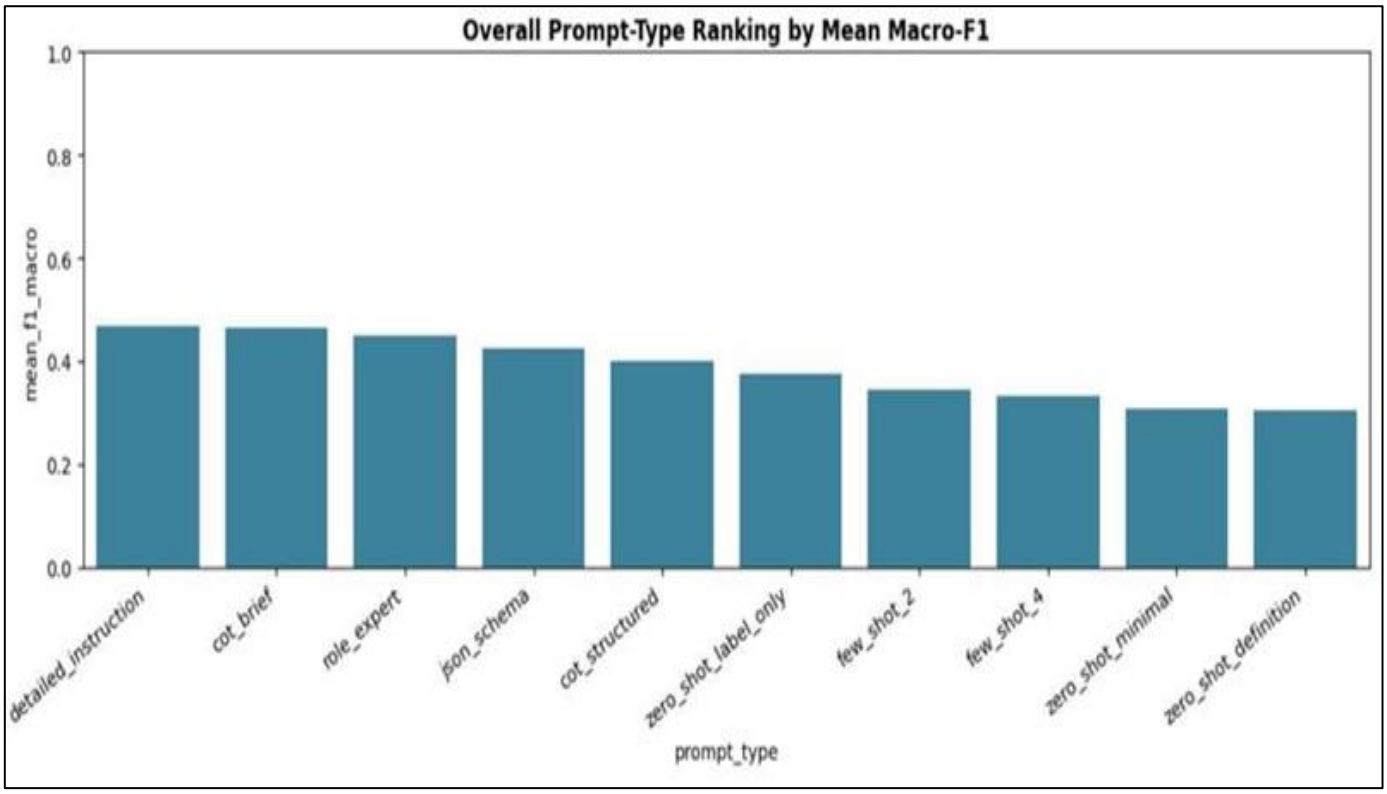

Fig 2 Prompt Ranking Bar Chart

➢ *Task-wise Performance and Latency Analysis*

Figures 2 and 3 show task-dependent differences in performance and latency between types of prompts. The findings affirm that prompt effectiveness is highly task-dependent. Structured and reasoning-based prompts are advantageous to requirement classification, as constrained outputs and directed reasoning are vital to structured tasks. Performance in sentiment analysis is also relatively insensitive to prompt variation, which suggests that the sentiment analysis is less sensitive to prompt variation. Nevertheless, CoT prompting brings only slight improvements, which implies that reasoning can continue to be useful in simpler tasks. The most difficult task is topic classification, and the performance is always lower on all prompts. This implies that complex multi- class problems might require prompt engineering only, and emphasizes the importance of task- specific or hybrid solutions [13]. Latency does not vary significantly across prompt type (around 2.65- 2.75 seconds). Nonetheless, more sophisticated prompts like few-shot and structured prompting have a bit higher latency because of longer input length and more complexity in processing. Although these distinctions are not serious in small-scale experiments, they can be serious in large-scale deployments, and it is important to balance performance and efficiency [16].

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

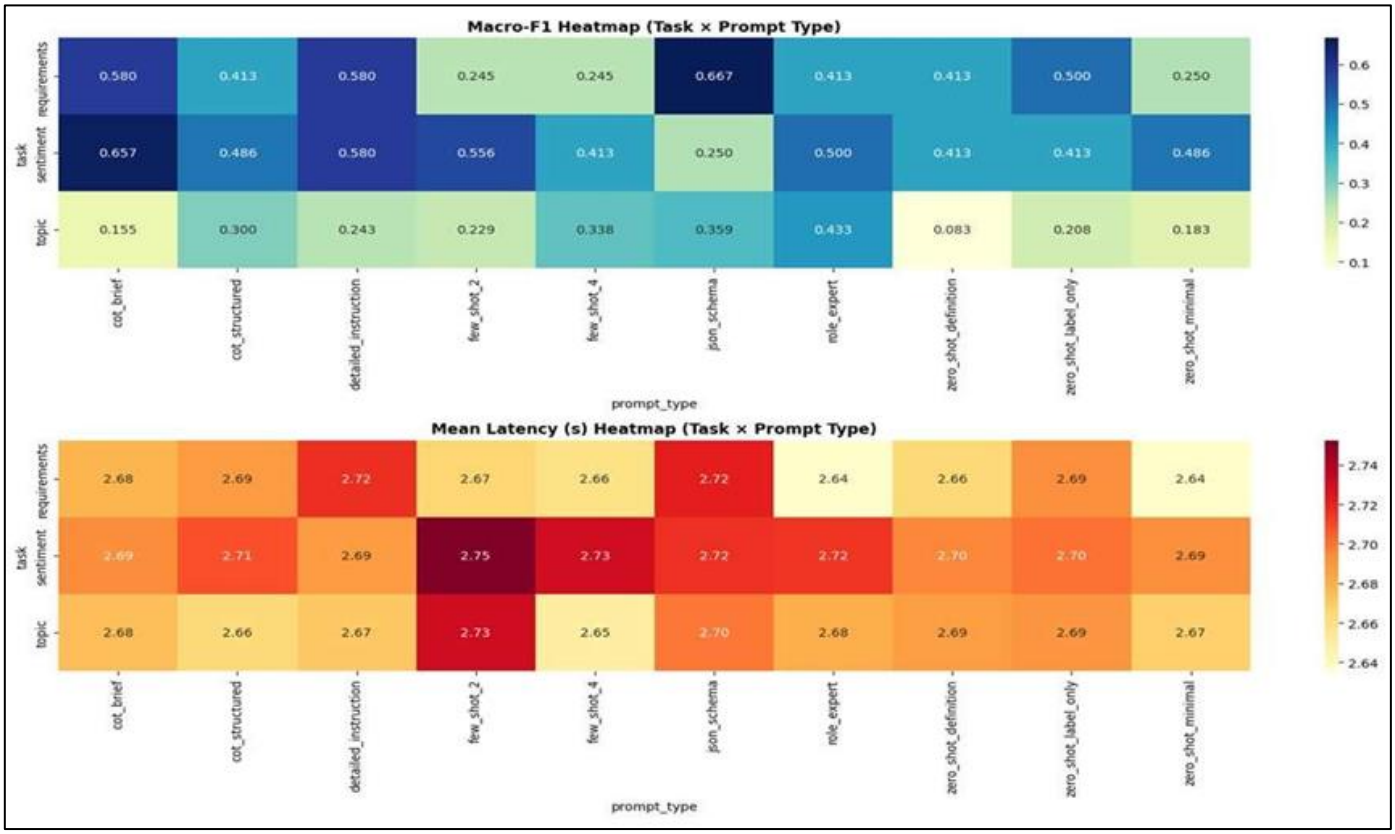

Fig 3 Macro-F1 Heatmap and Latency Heatmap

➢ *Performance–Efficiency Trade-off*

The findings indicate that there is an evident trade- off between performance and computational efficiency. Detailed and structured prompts are more accurate but have a higher latency because of a longer input and a more complicated processing model. Conversely, less complex prompts have quicker responses but poorer performance. This shows the importance of effectiveness and efficiency in prompt design to implement in the real world. Also, few-shot prompting does not always enhance performance and raise computational cost, which means that naïve prompt scaling is not very efficient. These results indicate that quality and structure rather than quantity should be the first priorities when optimizing prompting, as previous research has proposed [19].

➢ *Invalid Output Analysis*

There is a high rate of invalid output that is constant in all the prompting strategies. The model often does not comply with the necessary output format even when structured means, including JSON-constrained prompting, are used. This shows one important weakness of LLMs in being constrained in their output. The practical implications of this issue are immense because good and well-organized outputs are essential in the real-world setting. These findings suggest that prompt engineering is inadequate to guarantee output validity, and other mechanisms like post- processing, validation layers, or constrained decoding are needed [10].

➢ *Summary of Key Findings*

In general, the results indicate that prompt engineering is a determining factor of the performance of LLM, with the effectiveness depending on tasks and the type of prompt. Instruction-based and structured prompts always outperform minimal ones, and few-shot prompting does not ensure any improvements. A direct trade- off between performance and efficiency, as well as ongoing problems with reliability in output, can also be identified by this study. These findings highlight the significance of efficient, effective, and robust prompt design to the practical use of LLM.

V. DISCUSSION

The research has a number of valuable lessons about the importance of prompt engineering in large language models (LLMs). To begin with, the findings affirm that prompt design is an essential element affecting the performance of the models. Instruction-based and structured prompts are always superior to minimal prompting strategies, and chain- of-thought (CoT) and detailed instructions are the most reliable types of prompts. This supports earlier results that explicit thinking and clear instructions can improve the decision-making abilities of LLM [2], [6]. One important observation is that prompt performance is highly task-specific. Although structured prompting (e.g., JSON-constrained outputs) is advantageous to requirement classification, sentiment analysis demonstrates relatively consistent performance with different types of prompts, indicating that it is not sensitive to prompt

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

variation. Conversely, topic classification is difficult in all combinations, meaning that prompt engineering might not be effective on multi-class tasks. This points to a requirement of task-aware (or hybrid) methods in situations which are more complex [13], [14]. The other interesting observation is that few-shot prompting is not very effective in this experimental mode. However, as it is often believed, the addition of examples does not always enhance performance and can add an unwanted overhead. It indicates that good and relevant examples are more valuable than numerous ones, which is in accordance with previous research [5]. Efficiency-wise, the findings suggest a definite trade-off between performance and computing cost. More intricate prompts, including few-shot prompts and structured prompts, are more expensive in terms of latency as they have longer inputs and processing demands. Even though these variations are small in small-scale experiments, they may be large in actual world deployments, which highlights the need to design prompts efficiently [16]. Also, the invalid output rate is very high and consistent across all prompting strategies, revealing a fundamental weakness of LLMs in their ability to conform to strict output constraints. The process of structured prompting, even in a structured form, does not give complete assurance of the correct formatting, pointing to the necessity of additional processes like validation layers or constrained decoding to achieve reliability in real-world usage [10].

- *Implications for Green AI*

The implications of sustainable AI practices are also to be found in the findings. Direct effects on computational cost and energy consumption, longer and more elaborate prompts are more costly to process in the first place. The identified inefficiency of naive prompt scaling, especially when few-shot models are used, highlights the need to create concise and effective prompts. Resource usage can be minimized by optimizing prompt length and structure, and remaining competitive, which is in line with the aims of Green AI [17]. In general, prompt engineering must be considered as a performance improvement tool, as well as an efficiency and sustainability of LLM-based systems.

VI. LIMITATIONS

There are a number of limitations to this study. First, the experiments were performed at a pilot level and had a small sample size, which can influence the generalization of the findings. Second, the analysis was limited to one LLM, which restricted the information on the various model architectures.

Third, in this study, the computational cost was estimated based on latency, and no energy consumption was directly measured. Lastly, the research is aimed at classification tasks, and results might not be applicable to generative or reasoning tasks. These limitations should be overcome in future work with larger-scale experiments, multi-model evaluations, and direct energy-based metrics.

VII. CONCLUSION

In this paper, a multi-task benchmark on the basis of the strategies of prompt engineering in large language models (LLMs) is provided. The study shows that the design of prompts is a key determinant in model performance, and structured and instruction-based prompts are always effective compared to minimal methods by thoroughly examining ten types of prompts in three NLP tasks. The results also point to critical issues. The performance of few-shot prompting is not consistently enhanced, which suggests the shortcomings of naive prompt scaling. There is also an apparent trade-off in performance and computational efficiency, and inadequate reliability in output, especially in long-term structured response maintenance. Overall, this work underscores the importance of task-aware, efficient, and reliable prompt design. The suggested benchmark offers a baseline for further studies of how to optimize prompt approaches, enhance model resilience, and create sustainable LLM systems by implementing larger models, datasets, and energy-conscious assessments.

REFERENCES

[1]. T. Brown et al., “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[2]. J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *arXiv preprint arXiv:2201.11903*, 2022.
[3]. T. Kojima et al., “Large Language Models are Zero- Shot Reasoners,” *arXiv preprint arXiv:2205.11916*, 2022.
[4]. X. Wang et al., “Self-Consistency Improves Chain-of- Thought Reasoning in Language Models,” *arXiv preprint arXiv:2203.11171*, 2023.
[5]. S. Min et al., “Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?” in *EMNLP*, 2022.
[6]. P. Liu et al., “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in NLP,” *ACM Computing Surveys*, 2023.
[7]. L. Reynolds and K. McDonell, “Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm,” *arXiv preprint arXiv:2102.07350*, 2021.
[8]. Y. Schick and H. Schütze, “Exploiting Cloze Questions for Few-Shot Text Classification,” in *NAACL*, 2021.
[9]. A. Holtzman et al., “The Curious Case of Neural Text Degeneration,” in *ICLR*, 2020.
[10]. R. Bommasani et al., “On the Opportunities and Risks of Foundation Models,” *arXiv preprint arXiv:2108.07258*, 2021.
[11]. OpenAI, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023.
[12]. M. Bubeck et al., “Sparks of Artificial General Intelligence: Early Experiments with GPT-4,” *arXiv preprint arXiv:2303.12712*, 2023.
[13]. J. Zhou et al., “A Survey of Large Language

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1597

Models,"*arXiv preprint arXiv:2303.18223*, 2023.
[14]. D. Hendrycks et al., "Measuring Massive Multitask Language Understanding," in *ICLR*, 2021.
[15]. A. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *JMLR*, 2020.
[16]. E. Strubell et al., "Energy and Policy Considerations for Deep Learning in NLP," in *ACL*, 2019.
[17]. R. Schwartz et al., "Green AI," *Communications of the ACM*, 2020.
[18]. S. Sanh et al., "Multitask Prompted Training Enables Zero-Shot Task Generalization," in *ICLR*, 2022.
[19]. X. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in *ACL*, 2021.
[20]. B. Lester et al., "The Power of Scale for Parameter- Efficient Prompt Tuning," in *EMNLP*, 2021.