

Research Article

The Pre-Informing Approach and a Synthesis of Existing Prompting Techniques for Improving Output Quality in Black-Box Large Language Models

Black-Box Büyük Dil Modellerinde Çıktı Kalitesini Artırmak için Pre-Informing Yaklaşımı ve Mevcut İstem Tekniklerinin Sentezi

Muhammed Lütfü ODABAŞOĞLU Yüksek Lisans Öğrencisi, Kafkas Üniversitesi İktisadi İdari Bilimler Fakültesi 242624002@ogr.kafkas.edu.tr https://orcid.org/0009-0001-7809-9157	Muhammed Akif YENİKAYA Dr. Öğr. Üyesi, Kafkas Üniversitesi İktisadi İdari Bilimler Fakültesi akif.yenikaya@kafkas.edu.tr https://orcid.org/0000-0002-3624-722X
--	--

Makale Geliş Tarihi	Makale Kabul Tarihi
04.11.2025	05.03.2026

Abstract

This study proceeds from the premise that the effectiveness of Large Language Models (LLMs) in synthetic data generation depends not only on model capacity but also on the quality of human-guided prompting strategies. Within this framework, it reviews twenty-three prompt-writing techniques reported in the literature and proposes a hybrid approach, pre-informing, which aims to improve output quality through structured contextual preparation before the main instruction. The study comparatively evaluates zero-shot and pre-informing prompting across 20 education-domain text-generation tasks, using 3 black-box LLMs: ChatGPT 5.2 Standard, Gemini 3 Fast, and Claude Sonnet 2.6. This design yields a total of 120 generated outputs.

The outputs are evaluated through a common quantitative framework consisting of Word Count, Unique Word Count, Average Sentence Length, and Technical Terms Count. The internal coherence of this indicator set is supported by a standardized Cronbach's alpha of .887. Descriptive comparisons show that the pre-informed condition yields higher values across all four indicators. These differences are further confirmed by Wilcoxon signed-rank tests, which show statistically significant improvements across all four measures. In addition, Friedman test results indicate that the magnitude of improvement differs significantly across models, with Claude showing the largest overall gains, Gemini the most limited gains, and ChatGPT generally occupying an intermediate position.

The findings indicate that pre-informing is associated with stronger output quality as reflected in measurable indicators, and that its effects are observable across multiple task types rather than being limited to isolated examples. At the same time, the results show that the effectiveness of pre-informing is not uniform across black-box LLMs and remains partly model-sensitive. The study positions pre-informing as a structured and reproducible prompt-based framework for improving quality-related output characteristics and underscores the continuing importance of human guidance in shaping LLM outputs.

Keywords: Large Language Models, Prompt Design, Prompt Engineering, Synthetic Data Generation, Black-box LLMs

Önerilen Atf /Suggested Citation

Odabaşoğlu, M.L. & Yenikaya, M.A., 2026, The Pre-Informing Approach and a Synthesis of Existing Prompting Techniques for Improving Output Quality in Black-Box Large Language Models, *Üçüncü Sektör Sosyal Ekonomi Dergisi*, 61(1), 1243-1270.

Öz

Bu çalışma, Büyük Dil Modellerinin (LLM) sentetik veri üretimindeki etkinliğinin yalnızca model kapasitesine değil, insan yönlendirmeli istem (prompt) yapısının niteliğine de bağlı olduğu kabulünden hareketle, literatürde yer alan yirmi üç istem yazım tekniğini incelemekte ve ana yönerge öncesinde yapılandırılmış bağlamsal hazırlık yoluyla çıktı kalitesini artırmayı amaçlayan pre-informing adlı hibrit bir yaklaşımı önermektedir. Çalışma, zero-shot ve pre-informing istemleme yaklaşımlarını, eğitim alanına ait 20 metin üretim görevi kapsamında ve 3 kapalı kutu LLM (ChatGPT 5.2 Standard, Gemini 3 Fast ve Claude Sonnet 2.6) kullanarak karşılaştırmalı olarak değerlendirmektedir. Bu tasarım toplam 120 çıktı üretmektedir.

Üretilen çıktılar, Kelime Sayısı (Word Count), Benzersiz Kelime Sayısı (Unique Word Count), Ortalama Cümle Uzunluğu (Average Sentence Length) ve Teknik Terim Sayısı (Technical Terms Count) göstergelerinden oluşan ortak bir nicel çerçeve aracılığıyla değerlendirilmektedir. Bu gösterge setinin içsel tutarlılığı, .887'lik standartlaştırılmış Cronbach's alpha katsayısı ile desteklenmektedir. Betimsel karşılaştırmalar, pre-informed koşulunun dört göstergenin tamamında daha yüksek değerler ürettiğini göstermektedir. Bu farklar ayrıca, dört ölçütün tamamında istatistiksel olarak anlamlı iyileşmelere işaret eden Wilcoxon işaretli sıralar testleri ile doğrulanmaktadır. Buna ek olarak, Friedman testi sonuçları iyileşme büyüklüğünün modeller arasında anlamlı biçimde farklılaştığını; Claude'un en yüksek genel kazanımları, Gemini'nin en sınırlı kazanımları gösterdiğini ve ChatGPT'nin genel olarak orta bir konumda yer aldığını ortaya koymaktadır.

Bulgular, pre-informing yaklaşımının, ölçülebilir göstergelere yansıdığı biçimiyle daha yüksek çıktı kalitesiyle ilişkili olduğunu ve etkilerinin yalnızca tekil örneklerle sınırlı kalmayıp birden fazla görev türünde gözlemlenebildiğini göstermektedir. Aynı zamanda sonuçlar, pre-informing'in etkisinin kapalı kutu LLM'ler arasında eşit dağılmadığını ve kısmen modele duyarlı kaldığını ortaya koymaktadır. Çalışma, pre-informing'i kaliteyle ilişkili çıktı özelliklerini iyileştirmeye yönelik yapılandırılmış ve tekrarlanabilir bir istem temelli çerçeve olarak konumlandırmakta ve LLM çıktılarının şekillenmesinde insan yönlendirmesinin süregelen önemini vurgulamaktadır.

Anahtar Kelimeler: *Büyük Dil Modelleri, İstem Tasarımı, İstem Mühendisliği, Sentetik Veri Üretimi, Black-box LLM'ler*

1. Introduction

Large Language Models (LLMs) have seen rapid development in recent years (Yao et al., 2023). Owing to this rapid progress, these systems now generate organized and useful content across numerous subject areas (Gero et al., 2022; Brown et al., 2020). As artificial intelligence and large language models continue to improve, LLM-based generative applications extend beyond text generation toward automation and operational efficiency, helping streamline day-to-day processes and reduce errors in different settings (Ibrahim et al., 2025). In turn, LLMs now function as key enablers of these advances and broaden AI's practical range.

LLMs are generative AI systems designed to interpret and produce human-like text by emulating linguistic behavior and replicating cognitive patterns observed in natural reasoning and understanding (Perković et al., 2024). These human-like processing abilities make them highly adaptable across both general-purpose and domain-specific applications.

As their cognitive design aligns with increasingly complex real-world tasks, LLMs are now more frequently employed in specialized, task-oriented systems. In such contexts, large language models can contribute to task-oriented systems by generating context-aware suggestions or strategic outputs without directly functioning as reinforcement learning agents, highlighting their broader role in decision-support and inference-driven processes (Lin et al., 2025).

Beyond architectural capabilities, much of this performance hinges on how tasks are communicated through prompts. This growing reliance on prompting has shifted attention toward the design of effective prompt formulations. Within this context, the present study specifically examines how prompt structure influences the quality of LLM outputs. More precisely, it focuses on whether organizing the prompt through a structured pre-task informational layer can improve the measurable quality of generated responses in black-box LLMs.

These models have already been trained on large-scale datasets encompassing a substantial portion of human-written digital content. Much of this training data is drawn from readily available sources created

by humans. As a result, the benefit of further training on the same data is diminishing, which raises the possibility of a future performance plateau and increases the importance of generating high-quality synthetic data. This brings LLMs into focus as one of the most capable tools for producing such content and positions prompt design as one of the most critical factors in shaping output quality.

Recognizing the central role of prompt design in the quality and controllability of generated outputs, this study focuses on pre-informing as a structured prompting strategy that provides targeted contextual preparation before the main instruction. The subject of the study is therefore not prompt engineering in general, but the specific question of whether a pre-task contextual layer can improve output performance in black-box LLMs.

Accordingly, the main aim of the study is to examine whether pre-informing improves output quality as reflected in measurable indicators when compared with zero-shot prompting. To address this question, the study compares the two prompting conditions across 20 text-generation tasks and 3 black-box LLMs, using a common quantitative indicator framework based on Word Count, Unique Word Count, Average Sentence Length, and Technical Terms Count. In this way, the study seeks to determine not only whether pre-informing is associated with measurable output gains, but also whether the magnitude of this effect varies across models.

2. Literature Review

The literature review of this study began with a broad exploration using the keywords “prompting”, “prompt engineering”, “prompting techniques”, and “large language model”. The initial scan returned a very large and topic-mixed set of records, which made focused theoretical synthesis difficult. We therefore limited the review to work explicitly situated in prompt engineering. To keep selection systematic, we ran the refined search mainly on Web of Science and Google Scholar.

To ensure a systematic selection of relevant sources, the refined search was conducted primarily through Web of Science and Google Scholar, two academically recognized indexing platforms. Based on this process, a total of 60 works were identified as foundational or highly relevant to the research scope. These include scientific articles, arXiv/preprint entries, conference proceedings, technical reports, books, and selected digital sources. The distribution of source types is summarized in Figure 1.

Figure 1: Document Type Distribution Among Cited Sources in the Literature Review

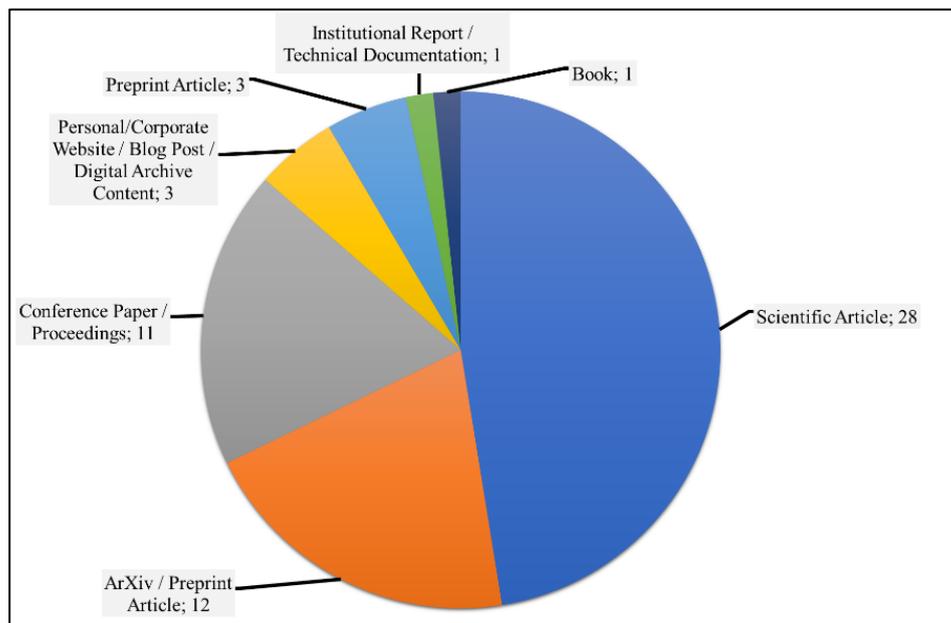


Figure 1 shows that scientific articles ($n = 29$), arXiv/preprint entries ($n = 12$), and conference proceedings ($n = 11$) are the three largest categories in the corpus. We next highlight the studies most relevant to this work.

- Brown et al. (2020) show that GPT-3, an LLM, achieves strong results across diverse tasks using zero-, one-, and few-shot natural-language prompts via in-context learning, demonstrating that clear textual instructions can steer behavior without retraining.
- Wang et al. (2022) build an iterative prompting scheme that composes step-by-step, context-aware prompts. The method helps the model recover intermediate facts and handle multi-hop reasoning, improving results on 2WikiMultiHopQA and R4C.
- Zhou et al. (2022) describe least-to-most prompting: the task is broken into simpler parts and solved in sequence with few-shot prompts. This improves generalization to harder cases and outperforms chain-of-thought on symbolic/compositional tasks (e.g., SCAN) and math reasoning.
- Diao et al. (2023) propose Active-Prompt, an uncertainty-guided selection of a small set of questions for chain-of-thought annotation.
- Yao et al. (2023) introduce Graph-of-Thought (GoT), which represents reasoning as a graph of interconnected ideas rather than a single linear chain and shows higher accuracy than CoT on AQUA-RAT and ScienceQA.
- Madaan et al. (2023) introduce SELF-REFINE, where the same LLM critiques and rewrites its own draft in short iterations, yielding consistently better outputs than one-pass generation without any extra training.
- Liu et al. (2023) provide a systematic survey of prompt-based learning for LLMs, formalizing the pre-train–prompt–predict paradigm with unified notation and a taxonomy covering prompt template/answer engineering, multi-prompt learning, tuning strategies, and applications.
- Marvin et al. (2023) offer a practitioner-oriented chapter that explains prompt engineering for LLMs, outlines a simple step-by-step process to craft prompts, catalogs widely used techniques with examples, and surveys tools and trends for real-world use.
- White et al. (2023) present a catalog of prompt patterns with a clear pattern format and concrete examples, offering reusable solutions for common LLM interaction and output challenges.
- Giray (2023) is a concise guide for academic writers that defines prompt engineering, explains core prompt elements and common techniques, and highlights pitfalls to help structure clearer and more responsible interactions with LLMs.
- Schulhoff et al. (2024) deliver a large-scale, machine-assisted survey that standardizes prompting terms, organizes 58 text-only and 40 multimodal techniques into a clear taxonomy, and summarizes common benchmarks and usage patterns to guide LLM prompt design and evaluation.
- Tian et al. (2024) introduce a domain-specific taxonomy of design prompts and structured patterns for design prompt synthesis and shows in a ChatGPT experiment that simple to moderately complex syntheses can boost the novelty and diversity of conceptual design ideas while overly complex chains reduce effectiveness.
- Sabbatella et al. (2024) present a black-box prompt optimization method that uses Bayesian optimization to automatically search for effective hard prompts, reporting competitive accuracy and shorter runtime on GLUE tasks with RoBERTa compared to other black-box baselines.

3. Materials and Methods

Grounded in the premise that human-authored prompting is a key determinant of large language model performance, this study compares two prompting conditions, zero-shot and pre-informing, across a broader multi-task setting. The study evaluates these two prompting conditions across 20 text-generation tasks and 3 black-box large language models, yielding a total of 120 generated outputs (20 tasks x 3 models x 2 prompting conditions). The models included in the analysis were ChatGPT 5.2 Standard, Gemini 3 Fast, and Claude Sonnet 2.6, and all generations were produced in February 2026.

All generations were produced in isolated sessions in order to prevent contextual carryover, with each run conducted under a clean interaction state. The same task prompt was applied to both prompting

conditions for each task-model pair, with the only systematic difference being the presence or absence of the pre-informing stage. No fine-tuning or parameter-level intervention was applied at any stage of the study. In this respect, the comparison was designed to isolate the effect of structured contextual preparation prior to the main task instruction.

The evaluation framework combined quantitative analysis with structured comparative interpretation. The quantitative component relied on a common set of four indicators applied consistently across all generated outputs: Word Count, Unique Word Count, Average Sentence Length, and Technical Terms Count. These indicators were selected because they capture observable and comparable dimensions of output differences across different tasks and models. Their definitions are presented in Table 6. Quantitative comparisons were conducted across prompting conditions and model groups using paired and non-parametric statistical procedures appropriate to the structure of the dataset.

In addition to descriptive comparisons, the study employed a staged analytical design. First, descriptive statistics were used to identify overall directional differences between the zero-shot and pre-informed conditions. Second, Wilcoxon signed-rank tests were used to evaluate whether the observed differences between the two prompting conditions were statistically significant across paired task-model observations. Third, model-level variation in improvement scores was examined using the Friedman test, allowing the study to assess whether the magnitude of the pre-informing effect differed across ChatGPT 5.2 Standard, Gemini 3 Fast, and Claude Sonnet 2.6. To support the internal coherence of the quantitative framework, a reliability analysis was also conducted on the four indicators.

This methodological structure was designed to evaluate not only whether pre-informing was associated with measurable changes in generated outputs, but also whether those changes were consistent across tasks and whether their magnitude varied by model. Accordingly, the study treats pre-informing as a prompt-based intervention whose effectiveness can be assessed through observable output differences rather than through access to internal model parameters.

3.1. Prompting and Prompt Engineering

A prompt is a structured natural-language input that guides the behavior of a large language model (DAIR.AI, 2023; Giray, 2023). Beyond a minimal definition, prompts function as the principal interface through which users elicit model behavior and generate content; because this interface constrains what the model attends to, prompt design has a direct bearing on output quality, and well-structured prompts have been reported to yield superior results across diverse tasks (Wei et al., 2022; Schulhoff et al., 2023). Within this framing, prompt engineering denotes the intentional construction of prompts to access latent capabilities in large models (Reynolds & McDonell, 2021).

Early studies on prompt wording and output quality reported mixed findings. Several papers found that clear instructions were not a dependable way to raise performance, especially in early models with weaker reasoning and factual accuracy (Michelet & Breitingner, 2024). More recent studies show that modest changes to prompt phrasing or design can meaningfully shift outputs (Maus et al., 2023; Lee & Shin, 2024). The gap between these results led to benchmarks like MMLU, which include 57 tasks across domains (Hendrycks et al., 2020).

Subsequent studies with more capable systems report sizeable effects of prompt design across a wide range of domains. In the medical domain, prompt engineering has been reported to enable open-source models to match or surpass fine-tuned systems (Maharjan et al., 2024). In healthcare triage, structured, multi-step prompts are reported to substantially increase accuracy while trading off response time and service-experience scores (Wang et al., 2025). In the legal domain, definition-driven prompts have been reported to support strong performance in zero-shot annotation with GPT-4, whereas larger in-prompt batches degrade quality (Savelka & Ashley, 2023). In code generation, optimized prompts increased a ChatGPT-4 similarity score from 85.93% to 93.87% (Ma et al., 2025). Other studies indicate that well-specified natural-language instructions can drive technically complex operations without specialized interfaces (Zhang et al., 2024; Ren et al., 2025). In higher education settings, components of AI literacy such as AI technology knowledge are positively associated with prompt engineering quality, and higher-quality prompts in turn predict higher-quality LLM outputs (Knoth et al., 2024). More recent work similarly finds that prompt structure and style affect output quality and can significantly influence performance (Elnashar, 2025; Cao et al., 2025).

Considered alongside these recent results, the early divergence is largely to be expected given the state of the technology at the time. At present, the literature broadly recognizes that prompt engineering, and specifically prompt structure, exerts a direct and measurable influence on output quality (e.g., Wei et al., 2022; Schulhoff et al., 2023; Sabbatella et al., 2024; Maharjan et al., 2024; Ma et al., 2025).

Taken together, the literature positions prompt engineering not as a peripheral concern but as a central, learnable interface for achieving accuracy, control, and task alignment in real-world applications. Beyond practical gains, specific prompting techniques may surface more advanced reasoning patterns in large models, suggesting that prompt engineering operates both as an interaction layer and as a potential key to unlocking emergent capabilities.

3.2. Prompt Structure and Stages

Effective prompt writing starts with a clear grasp of structure and essential components. A strong prompt is not simply a question or instruction. It is a structured input composed of multiple elements that jointly guide the behavior of a large language model (LLM). Structured and purposeful input selection has been shown to shape measured model behavior, including bias-related effects (Caliskan et al., 2017).

Recent frameworks converge on a modular understanding of what constitutes an effective prompt. DAIR.AI (2023) and Giray (2023) propose four core components: instruction, context, input data, and output indicator. The instruction defines the task, the context sets the background and constraints, the input data refers to the information to be processed, and the output indicator guides the response format. As a whole, these elements form a modular design that can improve precision, clarity, and relevance in the model's responses.

Marvin et al. (2023) set out a stage-based framework with five steps: goal definition, capability assessment, format selection, contextualization, and iterative refinement. Each step assists in mapping task needs into workable prompts and parallels the component view discussed above. Although framed procedurally, this model complements the component-based prompt frameworks proposed by DAIR.AI and Giray by emphasizing how construction order and content structure work in tandem.

This modular view is also reflected in task-specific adaptations developed for specialized domains. One example is provided by Taveekitworachai et al. (2023), who designed prompts for a procedural generation task in the ChatGPT4PCG competition. Their structure includes four comparable sections: instructions, definitions, environments, and tools. The definitions clarify domain-specific functions, environments define the variables or elements that shape possible outputs, and tools indicate functional operations available to the model. While tailored to a specific computational goal, this format aligns with the general-purpose modular framework proposed by DAIR.AI and Giray.

Building upon these frameworks, Tian et al. (2024) argue that effective prompt construction is not solely dependent on linguistic clarity but also on the deliberate organization of structural components that align with contextual and task-specific demands. Particularly in complex domains such as conceptual design, the quality of LLM outputs appears closely linked to the internal logic and arrangement of the prompt itself. These frameworks suggest that iterative methods and dynamic construction strategies may be at least as crucial as understanding the structural requirements of a well-designed prompt.

3.3. Existing Prompt Engineering Strategies for General-Purpose LLMs

Effective implementation of the pre-informing Method requires a solid understanding of established prompting strategies. This is because, rather than positioning itself as an alternative, it functions as an informative layer that organizes and refines these existing techniques.

To this end Table 1 consolidates the primary strategies discussed in literature. These approaches are specifically tailored for closed-source LLMs such as ChatGPT and are applicable across a wide range of user proficiency levels.

Table 1: Prompting Strategies Summary

Source Control
The model is configured via prompts to restrict information access to user-approved sources and explicit rules (e.g., "use only the attached paper"). Favor verbatim extraction over free paraphrase, and return "Unknown" when evidence is absent; this aims to improve traceability and to reduce hallucinations.
Zero-shot Prompting & Direct Task Specification
Aims to instruct a language model using only a task description in natural language, without any examples. Another commonly known term is direct task specification, where the instruction is phrased in a more precise and structured way to help the model activate relevant task behavior (Reynolds & McDonell, 2021).
One-shot Prompting
Aims to guide the model by providing a single example along with the task instruction, helping it infer the expected format and response style (Brown et al., 2020). The example acts as a reference point, anchoring the output to a specific tone, structure, or scenario without requiring multiple demonstrations.
Few-shot Prompting
Few-shot prompting is widely regarded as one of the most effective and commonly used prompting strategy. This strategy involves providing the LLM with a small number of examples of the desired outputs within a single prompt, which helps constrain the potential output space and guides the model toward the expected response pattern (Gao, 2023). These examples demonstrate the structure and logic of the task, allowing the model to infer the appropriate response pattern. In addition to its easy-to-implement structure, several studies have shown that well-designed few-shot prompts significantly improve output accuracy and quality (Hendrycks et al., 2020; Lee et al., 2024; Sivarajkumar et al., 2024).
Constrained Prompting
Aims to shape the model's behavior by explicitly or implicitly restricting undesired outputs through carefully phrased instructions. This design approach reduces ambiguity by narrowing the possible response space, filtering out irrelevant or inappropriate continuations (Reynolds & McDonell, 2021).
Memetic Proxy Prompting
Uses culturally or conceptually recognizable characters, situations, or analogies as indirect cues to guide model behavior. These proxies help the model infer complex or nuanced intent by leveraging learned associations with shared cultural references (Reynolds & McDonell, 2021; Latitude Team, 2020).
The Persona Pattern
This strategy involves assigning the model a specific identity or role, such as a teacher, developer, or fictional character, to guide the type and tone of its output (White et al., 2023). Role cues like "Act as a security analyst" activate internal associations that help the model adopt appropriate language, priorities, and perspective. Applications range from educational use to simulation and expert emulation, with studies showing improvements in both quality and task-specific accuracy (Sivarajkumar et al., 2024). Similar techniques also appear in the literature, such as PersonaCraft, which builds multidimensional profiles from real user data and enhances them with model-generated narratives (Jung et al., 2025). These approaches serve comparable functions by shaping the model's communicative lens, though their construction may rely more on empirical data than imaginative role assignment. Although it shares surface similarities with memetic proxy prompting, the persona pattern differs in intent and focus. Memetic proxies emphasize stylistic expression by evoking familiar cultural figures, while persona-based prompting primarily influences the informational content and perspective by grounding the model in a defined identity.
Serialized Reasoning Prompting
Encourages the model to approach complex tasks by breaking them into manageable, logically ordered steps, rather than producing an immediate response (Reynolds & McDonell, 2021). Prompts such as "Let's think step by step" support structured reasoning and are linked to what the authors call "truth-seeking patterns" in model behavior. Structured guidance of this kind has also been observed to improve model performance in analytical and creative tasks (Branwen, 2020). Serialized reasoning shares conceptual ground with Chain-of-Thought (CoT) prompting, which improves multi-step inference by embedding intermediate reasoning steps in few-shot examples (Wei et al., 2022). A closely related approach, termed "AI Chains" (prompt chaining) by Wu et al. (2022), sequences prompt-defined sub-tasks so that each intermediate output conditions the next step, improving transparency and controllability. Unlike serialized reasoning within a single prompt, chaining structures a multi-prompt pipeline with user-visible stages. Extensions like AP-CoT further refine this by automating exemplar selection for better contextual fit (Diao, 2023). Similarly, Tree-of-Thought (ToT) prompting enables the model to explore multiple reasoning paths, enhancing its capacity for tasks involving planning and exploration (Yao et al., 2023).
Least To Most Prompting
Using the LLM to address complex tasks by first guiding it to decompose them into smaller, more manageable components, and then solving each subproblem in a logically ordered manner (Zhou et al., 2022a). The technique's two key phases are a decomposition stage that demonstrates how to break down the main task and subproblem-solving stage that guides the model through resolving each component individually. This structured two-phase approach aims to enhance the LLM's ability to maintain clarity and accuracy across multi-step reasoning tasks. Zhou et al. (2022a) showed that this method outperformed both standard prompting and chain-of-thought prompting, particularly in tasks involving cumulative reasoning or layered computations.

Metaprompt Programming
Provide a general seed instead of direct instructions, prompting the model to compose its own task-specific prompt and structure. For example, a phrase like “This problem asks us to...” allows the model to unpack the goal and proceed with reasoning accordingly (Reynolds & McDonell, 2021). Because the model authors the prompt, it often relies on familiar pretraining patterns, yielding more coherent and context-aware outputs when the topic aligns with well-represented domains.
Iterative Seed-Based Prompt Execution
Running the same prompt multiple times with different random seed values allows users to observe how output quality and consistency shift under non-deterministic conditions. This technique helps reveal a prompt’s generative potential by comparing several outputs from the same input, typically using three to nine runs (Liu & Chilton, 2022). Building on this logic, White et al. (2023) proposed a method that asks the model to produce multiple independent responses to better explore diverse problem-solving paths. Similarly, Taveekitworachai et al. (2023) executed each prompt ten times per character to average performance scores, aiming to reduce the effect of output variability caused by the model’s stochastic behavior. Marvin et al. (2023) described this practice as self-consistency prompting, although their framing may blur the distinction between seeking semantic agreement and controlling statistical variance. Overall, the strategy moves beyond mere randomness by encouraging the model to vary its reasoning approach and generate more robust, interpretable results.
Meta Language Creation
Enables users to construct a custom meta-language composed of symbols or structured elements, each carrying a predefined meaning that guides the model’s interpretation. In situations where natural language alone makes it difficult to convey complex or subtle meanings, this technique offers a solution by allowing users to define their own symbolic language and specify the meaning of each component for the model in advance (Strobelt et al., 2022). Once these definitions are established, the model can interpret and apply them in subsequent prompts. This leads to responses that are more precise and easier to control, particularly in tasks that require formal logic, emotional nuance, or customized notation.
The Output Automator Pattern
Minimizes manual effort by directing the model to generate executable content, such as scripts or tools, that carry out previously suggested actions (White et al., 2023). Rather than having users implement each recommendation themselves, this pattern enables the model to produce integrated solutions that automate those steps. It proves especially valuable in contexts like software development or system operations, where repeated actions or structured output is common. The prompt should clearly define the desired automation and offer enough context to ensure relevant and functional results. While this reduces user workload, human validation remains essential to confirm accuracy, functionality, and safety (White et al., 2023).
The Flipped Interaction Pattern
Reverses the usual user-to-model instruction flow by allowing the model to ask the user questions in order to collect the information it needs before producing an output (White et al., 2023). Instead of following static commands, the model leads the interaction by determining what details are essential for completing the task accurately. This dynamic exchange helps align the model’s internal reasoning with the task’s specific requirements and improves the contextual fit of the final response. This shift aligns with Vogelsang’s (2024) broader perspective on large language models evolving into more collaborative and dialog-oriented systems rather than remaining passive tools. Supporting research by Zhou et al. (2022b) also shows that models can generate and apply their own prompts to better complete complex tasks. Related techniques such as the Cognitive Verifier Pattern (White et al., 2023) and Automatic Instruction Generation (Marvin et al., 2023) further illustrate how allowing the model to take initiative in prompt construction enhances reasoning and reduces user burden. Similarly, Zamfirescu-Pereira et al. (2023) frame a complementary approach as end-user prompt engineering and introduce BotDesigner, a design probe that lets non-experts author and systematically evaluate prompt templates.. Unlike the flipped interaction pattern, which delegates questioning to the model, their approach keeps initiative with the user and adds error labeling and regression-style testing tools.
The Cognitive Verifier Pattern
Encourages the model to review and validate its own reasoning after generating an output or response. So the technique is aiming to detect logical flaws, contradictions, or uncertainties before finalizing the output (White et al., 2023).
The Question Refinement Pattern
Directs the LLM to seek clarification first rather than respond to a weak query immediately (White et al., 2023). By rewriting the question or posing targeted follow-ups to clarify intent, it yields more precise, context-aware outputs and fewer rounds of clarification.
The Context Manager Pattern
Aims to improve output quality by instructing the language model to review and assess its own response, thereby leveraging its internal reasoning capabilities to enhance reliability (White et al., 2023).
Reasoning and Acting (ReAct) Framework
Guides language models to alternate between generating reasoning steps and executing actions. As described by Yang et al. (2023), ReAct “prompts an LLM to generate reasoning texts that break down complex problems into intermediate steps, and action texts that allocate NLP tools for solving these steps” (Yang et al., 2023). This two-step structure helps reduce the difficulty of the task by spreading it over several smaller steps, and allows the model to bring in external tools to improve its answers.
Iterative Context-Aware Prompting & Dynamic Prompting
Iterative context-aware prompting refers to a strategy in which the model constructs new, context-sensitive prompts at each step by using both the original query and previously generated outputs to guide subsequent reasoning stages (Wang et al., 2022). Rather than operating with a fixed input throughout, the model dynamically adapts its prompt based on the evolving context. With this iterative approach, it becomes possible to achieve more accurate, focused, and coherent progress across multi-step tasks. The original formulation describes a system that generates each new input by integrating prior reasoning

<p>with the current query, enabling the model to build a structured reasoning path from one stage to the next. This structure was later classified by Marvin et al. (2023) under the broader concept of dynamic prompting, defined as “dynamically adjusting the prompt based on the language model’s previous responses to improve its performance over time.” The strategy is framed as a response-aware refinement mechanism, where each output contributes directly to shaping the next prompt in the sequence.</p> <p>Liu et al. (2023) further clarify the distinction between static and dynamic prompting functions by describing static approaches as those that apply the same template regardless of input, whereas dynamic approaches generate custom prompts tailored to the specific instance. They emphasize that dynamic prompting involves prompts that are not only input-dependent but also capable of incorporating contextual feedback and reflecting task structure. This perspective reinforces the view that iterative prompting is not merely about altering words in a template, but about adapting the prompt in real time to match the model’s evolving understanding and to identify the most effective way to guide the LLM in solving a specific problem. A similar applied pattern is termed “iterative prompt template engineering” by Caruccio et al. (2024) and is described as a “novel iterative prompt template engineering approach, which integrates files by exploiting prompts and evaluates the quality of responses provided by the LLM.” The authors also note that “multiple interactions with LLMs in the same context can progressively improve responses,” illustrating dynamic prompting in a multimodal setting.</p> <p>In short, dynamic prompting represents a powerful mechanism to manage complex instructions under both cognitive and computational constraints.</p>
<p>Format-Constrained Prompting</p> <p>Guides the model to produce outputs in a fixed, structured format like a multiple-choice selection or a machine-readable JSON block. Rather than allowing open-ended, natural language responses, this approach confines the output to a controlled schema that is easier to interpret, compare, and process automatically (Huang et al., 2024).</p>
<p>Self-Criticism</p> <p>Enables LLM to evaluate and revise their own outputs without external supervision by drawing on metacognitive processes. Models reflect on their initial responses and generate internal feedback aimed at improving them (Huang et al., 2022). This can involve a basic correctness check or a more structured revision process. Recent research has formalized such capabilities into prompting strategies designed specifically for closed-model settings (Schulhoff et al., 2024). This approach can be categorized under two strategies: Self-Calibration and Self-Refine.</p>
<p>Self-Calibration</p> <p>Enables a language model to assess the reliability of its own responses. The process begins with an initial answer to a prompt and followed by a secondary prompt that asks the model to evaluate whether its own answer is correct. This strategy helps to estimate internal confidence and guides the model in determining whether the output should be accepted or revised (Kadavath et al., 2022).</p>
<p>Self-Refine</p> <p>Enables the LLM to iteratively generate, critique, and revise its own outputs through alternating feedback and refinement stages. It uses the same model during the process without requiring retraining or external input which is making it a lightweight and flexible approach for improving responses in tasks such as dialogue, coding, and creative writing (Madaan et al., 2023).</p>
<p>Reversing Chain-of-Thought (RCoT)</p> <p>Detects and corrects factual inconsistencies in the reasoning process of large language models. The method begins by asking the model to reconstruct the original problem based on its own recent generated solution. The process is followed by a fine-grained comparison between the reconstructed and original versions. This comparison aims help reveal specific types of errors, such as hallucinated conditions, overlooked details, and misinterpreted questions. Once those inconsistencies identified, they are formulated into targeted, granular feedback that guides the model in revising its answer. RCoT not only improves the factual accuracy of outputs but also enhances interpretability by making reasoning errors more transparent and accessible to human evaluation (Xue et al., 2023).</p>
<p>Expert-Refined Prompting</p> <p>Expert-refined prompting is a technique in which domain experts craft rule-guided instructions, iteratively revise them using model feedback, and finalize the instructions once they meet predefined reliability targets, after which the prompts are evaluated against baseline approaches such as vanilla and one-shot prompting (Dauphin & Siefert, 2025).</p>

3.4. Pre-Informing Approach

In addition to established prompting strategies, this study introduces a complementary approach termed pre-informing. Designed specifically for black-box large language models that do not provide access to internal parameters, training data, or fine-tuning capabilities, pre-informing aims to improve task performance solely through prompt-based interactions. Instead of relying on internal optimization mechanisms, it operates entirely through the external user-model interface, with structured informational input as its central mechanism.

Pre-informing is defined as a preparatory phase within prompt engineering, in which contextual, technical, or task-related information is systematically provided to the model before issuing any direct instruction. This input may consist of academic literature, procedural guidelines, domain-specific reports, or any relevant material needed to establish background knowledge. Crucially, this phase is

designed not only to guide the model’s attention but also to optimize how limited computational and cognitive resources are allocated on a per-prompt basis.

In many API-based or platform-restricted deployments of LLMs, each prompt is processed under a fixed computational budget that is typically governed by token and context length limits. When multiple documents are submitted simultaneously, the model may distribute its attention too broadly, leading to a superficial understanding of key content. By contrast, submitting documents sequentially and requesting analysis after each step allows more processing capacity to be dedicated to each input. This approach helps minimize omissions, deepens content comprehension, and improves overall output quality.

Although pre-informing resembles pre-training in its goal of embedding contextual knowledge, it differs in a critical respect: it does not involve modifying the model’s parameters. Mainly it leverages the model’s short-term memory and localized attention within the current prompt window. This makes it especially well-suited to black-box LLMs such as ChatGPT, where user interaction is limited to discrete prompts without persistent model tuning or data injection.

Pre-informing does not replace existing prompting strategies; rather, it serves as a foundational layer that enhances their effectiveness. Once the model has been primed through pre-informing, techniques such as few-shot or chain-of-thought prompting can be applied with greater impact and reliability. Pre-informing acts as a scaffolding mechanism that enables more precise and context-aware applications of other strategies. Pre-informing does not function in isolation or as a standalone technique. On the contrary, it often integrates methods such as constraint prompting (to enforce structural or stylistic boundaries), few-shot prompting (to illustrate task expectations), and flipped interaction patterns (where the model assumes a temporary diagnostic role during input processing). By layering these strategies across the preparatory steps, pre-informing creates a dynamic prompting environment that amplifies the model’s ability to generate context-sensitive, pedagogically sound outputs.

3.4.1.Operational Structure of the Pre-Informing Method

First, the user introduces the task but clearly instructs the model not to respond yet. Phrases like “Do not proceed yet” are used to pause execution and prevent the model from generating content too early, which also helps conserve system resources.

Next, the user shares relevant materials one by one. The model is asked to analyze each input, summarize key points, and reflect on how the content relates to the task. The user might ask guiding questions or highlight important domain-specific details. The model can also request clarification or more data if needed. This creates a temporary role reversal, as the model actively explores the task rather than simply reacting to a command. Through this back-and-forth, a shared understanding of the task is gradually built.

The final stage begins when the user is confident that the model has sufficiently processed the provided materials. The user then asks for a short summary of what has been shared to confirm understanding. Once confirmed, the main instruction is given, along with any formatting rules, domain constraints, or performance expectations.

This structured approach turns prompting into a step-by-step process focused on alignment, understanding, and goal-oriented generation. It allows more efficient use of the model's available context window per prompt, which contributes to more developed and coherent outputs. Distributing information across multiple prompts helps the model focus more precisely and retain context more effectively, leading to more accurate and well-structured results. In this respect, pre-informing represents a powerful approach for guiding content generation in black-box LLMs.

3.4.2.Practical Implementation of Pre-Informing

The operational execution of the pre-informing method follows a structured, multi-step prompting sequence that mirrors instructional planning.

Table 2: Operational Workflow of Pre-Informing

Step 1: Task Declaration and Preliminary Framing
The LLM is explicitly informed about the overall objective, the intended audience, and the required output format, while being clearly instructed to refrain from content generation until further directives are provided through prompts.
Step 2: Incremental Data Feeding with Comprehension Check
Collected data and documents relevant to the task are introduced one at a time, and the model is prompted to extract relevant information and confirm comprehension before proceeding to the next input. This step is repeated for each individual resource. This iterative data-feeding process ensures that all preparatory materials are independently processed, thoroughly analyzed, and contextually integrated before advancing to the main task prompt.
Step 3: Integrated Knowledge Consolidation
The model is asked to synthesize the previously introduced materials into a coherent summary that outlines core themes, structural elements, and domain-specific constraints pertinent to the target task.
Step 4: Final Prompt Instruction and Task Execution
Following the confirmation of conceptual readiness, the main task prompt is given to the LLM and the primary output gets generated.
Step 5: Optional Iterative Refinement and Finalization
LLM’s output undergoes refinement through iterative techniques such as dynamic prompting, self-evaluation, feedback integration, or content-based adjustments.

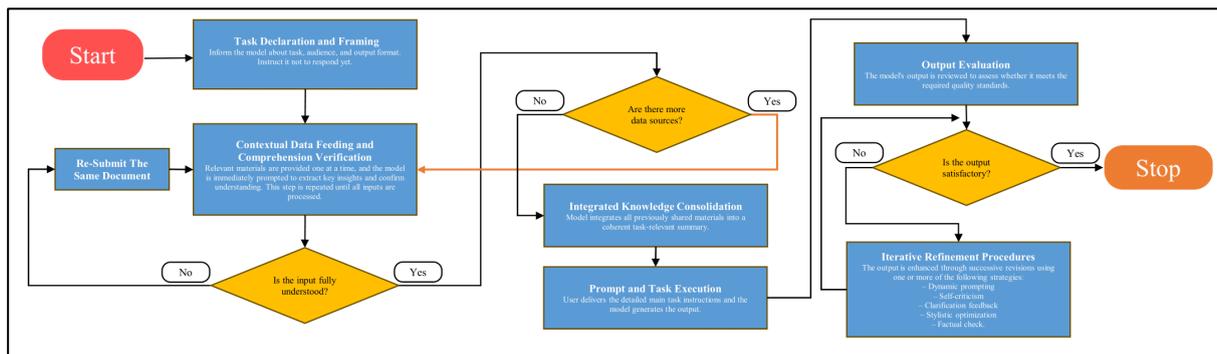


Figure 2: Operational Workflow of Pre-Informing on Flowchart

Figure 2 and Table 2 together depict the operational flow and structural logic of the pre-informing approach. Figure 2 provides a visual representation of the sequential stages, starting from task declaration and continuing through contextual feeding, consolidation, and final execution. Table 2 formalizes these phases and their core functions in textual format. Together, they clarify how pre-informing transitions a language model from raw task reception to context-aware generation, ensuring that each phase contributes meaningfully to alignment and instructional quality.

3.4.3. Application Across Twenty Educational-Domain Tasks

The pre-informing method is applicable to a wide range of text-based real-world tasks. In this study, its practical applicability was examined in the education domain through a broader task set rather than a limited number of illustrative examples. To evaluate the method under different writing demands, 20 education-oriented text-generation tasks were selected. These tasks were grouped into four categories: Instructional Design, Institutional Policy, Strategic Planning, and Socio-Economic Analysis. Each task was executed under both prompting conditions, zero-shot and pre-informing, in order to compare baseline outputs with outputs generated after structured contextual preparation.

To make the implementation process transparent, the step-by-step application of the pre-informing method is presented in detail for the first task. The first task is used to illustrate how the pre-informing sequence was operationalized in practice, including the staged contextual preparation applied before the main instruction. For the remaining 19 tasks, the same pre-informing logic and procedural pattern were followed.

The task set used in the study is as follows:

Instructional Design (5 Tasks)

Task 1 : Prepare a 40-minute lesson plan on renewable energy for secondary school students.

Task 2 : Design an instructional program on climate change adaptation for high school students.

Task 3 : Develop an interdisciplinary classroom activity integrating environmental science and economics.

Task 4 : Prepare a guide outlining evaluation criteria for student presentations on sustainable development.

Task 5 : Develop a guide for evaluating blended learning programs.

Institutional Policy (5 Tasks)

Task 6 : Develop a university classroom policy that balances digital engagement, student data privacy, attention management, and ethical use.

Task 7 : Draft a university policy on responsible artificial intelligence use.

Task 8 : Develop an academic integrity policy addressing AI-assisted writing tools.

Task 9 : Prepare a policy on student data collection and protection in digital learning systems.

Task 10: Develop a policy proposal to reduce digital distraction in university classrooms.

Strategic Planning (5 Tasks)

Task 11: Prepare a sustainability action plan for secondary education institutions.

Task 12: Develop a strategy to promote digital well-being in higher education institutions.

Task 13: Draft a strategic plan to improve STEM education.

Task 14: Develop strategic recommendations to reduce graduate brain drain.

Task 15: Propose strategies to improve alignment between vocational education and labor market needs.

Socio-Economic Analysis (5 Tasks)

Task 16: Analyze policy options to reduce youth unemployment.

Task 17: Analyze the economic and social impacts of rising housing rents.

Task 18: Analyze policy approaches to reduce gender-based employment inequality.

Task 19: Analyze the institutional risks associated with artificial intelligence use in educational management.

Task 20: Analyze the institutional impact of digital transformation in higher education.

This task structure was designed to test the pre-informing method across multiple forms of educational writing, ranging from instructional planning and policy formulation to strategic recommendation and socio-economic analysis. By applying the same comparative prompting framework across all twenty tasks, the study aimed to evaluate whether the effects of pre-informing remained observable across varied task types rather than being confined to a narrow set of examples.

3.4.3.1. Application on a Representative Task 1

The first task asked the model to generate a 40-minute lesson plan on renewable energy for secondary students (ages 13-15). The lesson-planning task was chosen because it goes beyond factual recall, requiring instructional sequencing, age-appropriate framing, and strategies for engagement, essentially simulating the reasoning of an educator. This complexity makes it a strong test case for pre-informing: it shows whether the model can internalize domain materials and translate them into a structured learning experience. The task was run on ChatGPT, first with a simple zero-shot prompt and then with the full pre-informing protocol.

Table 3: Zero-Shot Application for Task 1

Prompt:
Write a one page 40-minute lesson plan on renewable energy for secondary school students.
Output (Random Seed/Clean Conversation)
<p> Lesson Plan: Introduction to Renewable Energy Duration: 40 minutes Subject: Science / Geography / Environmental Studies Level: Secondary School</p> <hr/> <p> Learning Objectives By the end of the lesson, students will be able to:</p> <ol style="list-style-type: none"> 1. Define renewable energy. 2. Identify at least four types of renewable energy sources. 3. Explain basic advantages and challenges of renewable energy. 4. Compare renewable and non-renewable energy sources. <p>...</p> <p> Optional Homework Choose one renewable energy source and: Explain how it works (1 paragraph) Draw and label a diagram Give 2 advantages and 1 challenge</p> <hr/> <p> Extension (If Time Allows) Debate: “Should all countries switch to 100% renewable energy?”</p>

The zero-shot application for Task 1 is presented in Table 3. The following phase introduces the pre-informing approach, drawing on two preparatory documents selected for their complementary strengths in instructional clarity and technical depth. To implement this phase, the model was primed with external resources that combine pedagogical accessibility with academic rigor.

Selected sources

Tromly, K. (2001): Provides an accessible, thematically structured overview suitable for secondary classrooms;

Twidell, J. (2021): Offers comprehensive technical depth on renewable energy resources.

Table 4: First 3 Steps of Pre-Informing Application for Task 1

Step 1: Task Declaration and Preliminary Framing
Prompt:
<p>You will be assigned a specific instructional design task. The objective is to generate a 40-minute lesson plan on the topic of renewable energy, targeting secondary school students aged 13 to 15. The final output should be approximately one page in length and include an overview, learning objectives, key topics to be covered, suggested teaching methods, required materials, and an optional assessment activity. The structure must be pedagogically sound, age-appropriate, and aligned with basic educational principles.</p> <p>Do not begin the task or generate any output yet. Before you are prompted to produce the final output, you will be sequentially provided with supporting materials. These documents are intended to establish your understanding of the domain. At each step, you will be asked to analyze the material and summarize its relevance.</p> <p>Please confirm whether you fully understand the task description and the instructions above. Respond only with a brief confirmation or clarification request.</p>
Output (Random Seed/Clean Conversation)
Understood. I confirm the task description and instructions.

Step 2: Contextual Data Feeding
Prompt: (With attached document: Tromly, K. (2001). <i>Renewable Energy: An Overview. Energy Efficiency and Renewable Energy Clearinghouse (EREC) Brochure. U.S. Department of Energy.</i>)
You are now provided with a document titled “Renewable Energy: An Overview.” Conduct a detailed analysis of its content. Identify the main thematic sections, core concepts, and key insights presented throughout the document. Summarize the material with particular attention to components that are suitable for educational narration, such as definitions, energy source classifications, benefits, challenges, and real-world applications. Highlight content that could be used directly or adapted into an instructional format for secondary-level students.
Output (Random Seed/Clean Conversation)
<i>*(The model produced a concise, instruction-ready synthesis of "Renewable Energy: An Overview" for students aged 13–15, distilling the main thematic sections, core concepts, narrative focal points, and classroom-adaptable content.)</i>
Feeding Continues with Second Document
Prompt: (With attached document: Twidell, J. (2021). <i>Renewable energy resources. Routledge.</i>)
You are now provided with a second reference document titled “Renewable Energy Resources.” Conduct a thorough analysis of its content. Focus on extracting advanced insights, classifications, and technical descriptions that can complement or extend the understanding developed from the first document. Pay particular attention to sections that could enrich an educational narrative for secondary-level students, such as case studies, comparative data, conceptual frameworks, or visual elements. Provide a structured summary that emphasizes how this material supports or deepens the instructional objective.
Output (Random Seed/Clean Conversation)
<i>*(The model generated an instruction-ready synthesis of Twidell (2021), extending the overview with themes, core frameworks, age-adaptable classifications, case-based comparisons, adaptable visuals, and clear guidance for secondary classroom use.)</i>
Step 3: Information Absorption Verification
Prompt:
You have now reviewed two foundational documents on renewable energy. Synthesize the key insights into an integrated conceptual framework. Extract the most valuable elements from each source including their unique contributions, pedagogically rich content, and domain-specific terminology. Emphasize recurring themes, conceptual consistencies, and complementary distinctions. Your summary should reflect a coherent understanding suitable for instructional design, particularly for constructing an age-appropriate lesson plan aligned with the original task objective.
Output (Random Seed/Clean Conversation)
<i>*(The model synthesized a unified framework from both documents for a 40-minute secondary lesson (ages 13-15), defining the core concept, recurring themes, age-appropriate activities and assessment, and a coherent lesson spine for engaging, rigorous instruction.)*</i>

**(...)" denotes summaries of long outputs; other entries are direct outputs.

Table 4 shows the first three steps of the pre-informing process applied to Task 1, including task initialization and two sequential document analyses of Tromly (2001) and Twidell (2021), culminating in a synthesized conceptual framework.

These early stages are especially critical for verifying whether the model has correctly understood and internalized the provided information. We observe the quality and effectiveness of assessing the model’s developing understanding before moving on to the final task instruction by highlighting this part of the process. The observation allows us to follow up how well the model assimilates domain-specific content and how effectively it can reorganize that content to meet the task objectives.

In this case, the model went beyond simple comprehension. It started transforming the information into structured, pedagogically meaningful formats. This indicates that the model was not just absorbing content but actively anticipating how to organize and deliver it. Such behavior appears to be triggered by prompt designs that activate latent educational patterns in the model’s training data. The model even inferred sequencing strategies and age-appropriate framing, suggesting that pre-informing can engage a dormant instructional capacity in which the model begins to design rather than merely describe.

Table 5: Pre-Informing Application for Task 1 (4th Step)

Step 4: Final Prompt Instruction and Task Execution
Prompt:
Based on your previous analysis and the conceptual framework you have developed; you are now ready to generate the final instructional output. Your task is to design a 40-minute lesson plan on renewable energy, targeting secondary school students aged 13 to 15. The plan should reflect the pedagogical understanding and structural insights gained from the prior materials. You are encouraged to draw upon the synthesized content, thematic organization, and age-appropriate framing that emerged during the consolidation process. Maintain a coherent and engaging structure that promotes conceptual clarity and student comprehension. Do not hold back in scope or depth. This is your opportunity to demonstrate a full and thoughtful application of the knowledge you have acquired. Proceed to generate the complete lesson plan.
Output (Same Conversation)
40-Minute Lesson Plan: Renewable Energy (Ages 13–15) Overview (Teacher-facing) Students explore what renewable energy is, how it differs from fossil fuels, and how several renewable technologies work. They use a simple decision framework to match energy solutions to places, emphasizing energy “flows vs stores,” conversion chains, variability, efficiency, and trade-offs. The lesson blends short explanations with a sorting task and a mini design challenge. ... Exit Ticket (optional, quick assessment): <ol style="list-style-type: none"> 1. Define renewable energy using “flows vs stores.” 2. Draw one conversion chain (e.g., wind → turbine → generator → electricity). 3. Choose the best renewable for a rainy, windy island and explain using two checklist points.
Optional Extension / Homework (if desired) <ul style="list-style-type: none"> • Create an energy poster: “Most renewables are solar in disguise” with arrows showing connections (sun → wind/hydro/biomass/waves). • Mini-research: Find one local renewable project (solar on a school, wind farm, hydro plant) and identify benefits/trade-offs.
Notes for Differentiation <ul style="list-style-type: none"> • Support: Provide sentence starters (“We recommend ___ because ___”); reduce options to 4 technologies. • Stretch: Add “compare two options” using capacity factor reasoning; include a constraint (budget, wildlife protection, space limits).

“(…)” denotes summaries of long outputs; other entries are direct outputs.

Table 5 displays the fourth step (the main output) of Task 1 as implemented by the GPT 5.2 model under the pre-informing approach. The output is generally well-composed, the components such as overview, learning objectives, content progression, instructional strategies, timing, and assessment mechanisms are all well-organized within a consistent pedagogical framework. It should not be forgotten that the table summarizes content through ellipses for brevity, the original output includes expanded descriptions, illustrative diagrams, and embedded interactive prompts. These broad and consistent improvements indicate that the model did not merely generate isolated instructional fragments, but engaged with the task holistically, adopting the role of an instructional planner.

A more detailed comparison is provided under section 4. Results and Discussion.

4. Results and Discussion

4.1. Quantitative Evaluation Indicators

To support the quantitative comparison of outputs generated under the zero-shot and pre-informed conditions, the study employs a common set of four quantitative indicators applied consistently across all generated outputs. These indicators were selected to capture core observable dimensions of output structure and content in a systematic and replicable way. Specifically, the analysis focuses on Word Count, Unique Word Count, Average Sentence Length, and Technical Terms Count. Together, these indicators provide a shared measurement framework for comparing outputs across tasks, models, and prompting conditions. The indicator set used in the analysis is presented in Table 6. The internal consistency of this indicator set is examined in the following section.

Table 6. Quantitative Indicators Used in the Analysis

Indicator	Definition
Word Count	Total number of words in the generated output.
Unique Word Count	Number of distinct lexical items used in the output.
Average Sentence Length	Mean number of words per sentence in the generated text.
Technical Terms Count	Number of domain-specific or conceptually specialized terms included in the output.

4.2. Test Results

Following the quantitative evaluation framework introduced in the previous section, the raw measurement results obtained from all task-model-prompting combinations are presented in Table 7. The table reports the directly recorded values for Word Count, Unique Word Count, Average Sentence Length, and Technical Terms Count across the full dataset. These measurements constitute the empirical basis of the subsequent reliability analysis, descriptive comparisons, and inferential tests.

Table 7. Raw Measurement Results for All Task-Model-Prompting Combinations

Task	LLM	Type	Word Count	Unique Word Count	Average Sentence Length	Technical Terms Count
Task1	ChatGPT	Zero-Shot	431	268	8,47	16
Task1	ChatGPT	Pre-Informed	1038	491	11,36	39
Task1	Gemini	Zero-Shot	502	265	9,52	20
Task1	Gemini	Pre-Informed	537	297	12,01	29
Task1	Claude	Zero-Shot	900	487	13,02	29
Task1	Claude	Pre-Informed	1769	802	15,81	33
Task2	ChatGPT	Zero-Shot	638	407	11,66	52
Task2	ChatGPT	Pre-Informed	970	478	14,78	58
Task2	Gemini	Zero-Shot	402	259	11,15	44
Task2	Gemini	Pre-Informed	435	276	15,08	46
Task2	Claude	Zero-Shot	1194	578	17,38	48
Task2	Claude	Pre-Informed	2546	937	21,64	64
Task3	ChatGPT	Zero-Shot	390	260	10,88	43
Task3	ChatGPT	Pre-Informed	910	452	15,54	47
Task3	Gemini	Zero-Shot	419	255	11,74	35
Task3	Gemini	Pre-Informed	465	291	13,61	40

Task3	Claude	Zero-Shot	789	416	14,12	36
Task3	Claude	Pre-Informed	1076	506	18,27	38
Task4	ChatGPT	Zero-Shot	525	296	9,74	24
Task4	ChatGPT	Pre-Informed	875	413	18,92	32
Task4	Gemini	Zero-Shot	362	257	9,28	24
Task4	Gemini	Pre-Informed	577	346	13,57	30
Task4	Claude	Zero-Shot	1139	539	16,86	26
Task4	Claude	Pre-Informed	1616	604	23,47	30
Task5	ChatGPT	Zero-Shot	692	408	11,82	38
Task5	ChatGPT	Pre-Informed	937	411	17,63	34
Task5	Gemini	Zero-Shot	378	249	12,09	24
Task5	Gemini	Pre-Informed	423	268	14,41	28
Task5	Claude	Zero-Shot	778	396	13,44	30
Task5	Claude	Pre-Informed	976	479	18,05	30
Task6	ChatGPT	Zero-Shot	621	342	17,98	43
Task6	ChatGPT	Pre-Informed	725	366	20,83	51
Task6	Gemini	Zero-Shot	407	258	17,65	31
Task6	Gemini	Pre-Informed	392	265	18,06	29
Task6	Claude	Zero-Shot	1021	461	19,51	48
Task6	Claude	Pre-Informed	3623	1129	24,86	66
Task7	ChatGPT	Zero-Shot	794	372	16,81	53
Task7	ChatGPT	Pre-Informed	1030	466	23,38	59
Task7	Gemini	Zero-Shot	414	228	18,83	51
Task7	Gemini	Pre-Informed	439	261	19,25	52
Task7	Claude	Zero-Shot	1811	685	19,88	83
Task7	Claude	Pre-Informed	2298	771	25,47	106
Task8	ChatGPT	Zero-Shot	621	323	18,6	49
Task8	ChatGPT	Pre-Informed	1083	507	21,37	70
Task8	Gemini	Zero-Shot	490	269	20,98	38
Task8	Gemini	Pre-Informed	541	301	23,06	48
Task8	Claude	Zero-Shot	1184	503	19,14	63
Task8	Claude	Pre-Informed	1374	493	17,48	68
Task9	ChatGPT	Zero-Shot	795	388	18,63	56
Task9	ChatGPT	Pre-Informed	992	453	20,74	60
Task9	Gemini	Zero-Shot	482	265	9,56	37
Task9	Gemini	Pre-Informed	512	296	14,89	41
Task9	Claude	Zero-Shot	1563	553	20,65	67
Task9	Claude	Pre-Informed	1660	529	26,32	88

Task10	ChatGPT	Zero-Shot	598	353	14,41	37
Task10	ChatGPT	Pre-Informed	1216	589	24,21	50
Task10	Gemini	Zero-Shot	439	277	7,94	28
Task10	Gemini	Pre-Informed	474	284	10,16	33
Task10	Claude	Zero-Shot	733	428	13,94	37
Task10	Claude	Pre-Informed	1952	708	17,76	57
Task11	ChatGPT	Zero-Shot	503	328	11,17	46
Task11	ChatGPT	Pre-Informed	830	427	17,55	50
Task11	Gemini	Zero-Shot	409	278	6,89	44
Task11	Gemini	Pre-Informed	488	314	10,24	55
Task11	Claude	Zero-Shot	2335	933	15,67	116
Task11	Claude	Pre-Informed	4630	1287	16,81	136
Task12	ChatGPT	Zero-Shot	529	317	7,74	68
Task12	ChatGPT	Pre-Informed	947	463	11,71	77
Task12	Gemini	Zero-Shot	454	285	13,88	32
Task12	Gemini	Pre-Informed	579	354	12,84	40
Task12	Claude	Zero-Shot	699	383	14,96	47
Task12	Claude	Pre-Informed	2852	991	19,53	97
Task13	ChatGPT	Zero-Shot	582	350	10,9	75
Task13	ChatGPT	Pre-Informed	913	439	13,52	81
Task13	Gemini	Zero-Shot	404	263	14,05	35
Task13	Gemini	Pre-Informed	520	337	14,29	48
Task13	Claude	Zero-Shot	670	402	13,61	55
Task13	Claude	Pre-Informed	2655	880	18,87	93
Task14	ChatGPT	Zero-Shot	871	530	13,62	77
Task14	ChatGPT	Pre-Informed	1120	519	15,88	89
Task14	Gemini	Zero-Shot	423	286	13,73	27
Task14	Gemini	Pre-Informed	511	323	14,52	33
Task14	Claude	Zero-Shot	798	475	15,44	68
Task14	Claude	Pre-Informed	2688	950	18,43	71
Task15	ChatGPT	Zero-Shot	413	275	10,77	46
Task15	ChatGPT	Pre-Informed	1043	505	14,91	64
Task15	Gemini	Zero-Shot	457	307	12,88	29
Task15	Gemini	Pre-Informed	468	304	16,24	28
Task15	Claude	Zero-Shot	489	306	12,64	37
Task15	Claude	Pre-Informed	2454	817	21,36	85
Task16	ChatGPT	Zero-Shot	1019	504	18,7	107
Task16	ChatGPT	Pre-Informed	2182	892	24	102

Task16	Gemini	Zero-Shot	561	333	21,8	60
Task16	Gemini	Pre-Informed	521	333	18,1	66
Task16	Claude	Zero-Shot	669	380	19	63
Task16	Claude	Pre-Informed	2721	961	25,7	111
Task17	ChatGPT	Zero-Shot	509	330	20,6	62
Task17	ChatGPT	Pre-Informed	1239	571	28	78
Task17	Gemini	Zero-Shot	359	243	19,7	34
Task17	Gemini	Pre-Informed	584	311	16,6	43
Task17	Claude	Zero-Shot	680	398	20,8	53
Task17	Claude	Pre-Informed	3016	965	24,4	104
Task18	ChatGPT	Zero-Shot	628	375	20,8	75
Task18	ChatGPT	Pre-Informed	1842	718	23,8	94
Task18	Gemini	Zero-Shot	501	323	17,7	54
Task18	Gemini	Pre-Informed	522	333	18,5	50
Task18	Claude	Zero-Shot	682	397	18,7	54
Task18	Claude	Pre-Informed	1777	685	21,7	86
Task19	ChatGPT	Zero-Shot	795	450	21,5	87
Task19	ChatGPT	Pre-Informed	1422	628	23	108
Task19	Gemini	Zero-Shot	497	299	19,3	46
Task19	Gemini	Pre-Informed	600	384	17,4	64
Task19	Claude	Zero-Shot	763	430	18,6	67
Task19	Claude	Pre-Informed	2363	901	22,8	103
Task20	ChatGPT	Zero-Shot	961	518	19,9	94
Task20	ChatGPT	Pre-Informed	1168	561	20,9	76
Task20	Gemini	Zero-Shot	414	284	18,4	47
Task20	Gemini	Pre-Informed	611	342	19,4	48
Task20	Claude	Zero-Shot	703	411	21,3	61
Task20	Claude	Pre-Informed	2150	782	24,5	107

4.3. Reliability and Internal Consistency of the Quantitative Indicator Set

To assess the internal consistency of the quantitative evaluation framework, a reliability analysis was conducted on the four quantitative indicators introduced in the previous section. The analysis was based on 120 valid observations, with no excluded cases, indicating that the dataset was complete and appropriate for reliability testing. As shown in Table 8, the raw Cronbach's alpha coefficient was .485, while the standardized Cronbach's alpha coefficient was **.887** across the four items. Because the indicators are measured on substantially different numerical scales, the standardized coefficient provides the more appropriate estimate in this context. Accordingly, the standardized alpha value indicates a high level of exploratory internal consistency among the indicators.

The difference between the raw and standardized alpha coefficients is methodologically expected and is mainly due to the unequal measurement scales of the variables. In particular, variables such as Word Count and Unique Word Count operate within much larger numerical ranges than Average Sentence Length, which lowers the raw alpha estimate. The inter-item correlation matrix further supports the

coherence of the indicator set, with correlations ranging from .508 to .980.

The strongest association was observed between Word Count and Unique Word Count ($r = .980$), while the remaining indicators showed moderate to strong positive relationships. In addition, the corrected item-total correlations ranged from **.517 to .981**, indicating that all four indicators contributed positively to the overall structure, although Average Sentence Length showed a comparatively weaker, yet still acceptable, contribution.

Taken together, these findings suggest that the quantitative evaluation framework is internally consistent as an exploratory indicator set. At the same time, the results indicate that these indicators should not be interpreted as a conventional psychometric scale. Rather, they represent related but non-identical dimensions of output quality, capturing complementary aspects of textual expansion, lexical diversity, structural elaboration, and domain-specific density.

Table 8. Reliability Statistics for the Quantitative Indicator Set

Statistic	Value
Valid cases (N)	120
Excluded cases	0
Number of items	4
Cronbach's Alpha	.485
Cronbach's Alpha Based on Standardized Items	.887
Inter-item correlation range	.508 to .980
Strongest inter-item correlation	.980
Corrected item-total correlation range	.517 to .981

4.4. Descriptive Statistics

Descriptive statistics indicate that outputs generated under the pre-informed condition yielded higher mean values than those produced under zero-shot prompting across all four quantitative indicators. The mean Word Count increased from **688.15** to **1314.62**, the mean Unique Word Count rose from **373.97** to **545.77**, the mean Average Sentence Length increased from **15.34** to **18.62**, and the mean Technical Terms Count increased from **49.60** to **62.38**. These descriptive differences suggest that the pre-informed condition produced outputs that were generally longer, more lexically diverse, more structurally elaborated, and more densely populated with domain-specific terminology. At the same time, the higher standard deviation values observed under the pre-informed condition indicate that the magnitude of improvement varied across tasks and models. Overall, these descriptive patterns provide an initial indication that the pre-informing approach is associated with broader and more developed outputs. The detailed descriptive values used in this comparison are reported in Table 9.

Table 9. Descriptive Statistics for the Quantitative Indicators by Prompting Condition

Prompting Condition	Indicator	N	Minimum	Maximum	Mean	Std. Deviation
Zero-Shot	Word Count	60	359	2335	688.15	359.159
Zero-Shot	Unique Word Count	60	228	933	373.97	124.079
Zero-Shot	Average Sentence Length	60	6.89	21.80	15.34	4.193
Zero-Shot	Technical Terms Count	60	16	116	49.60	20.897
Pre-Informed	Word Count	60	392	4630	1314.62	921.084
Pre-Informed	Unique Word Count	60	261	1287	545.77	250.826
Pre-Informed	Average Sentence Length	60	10.16	28.00	18.62	4.396
Pre-Informed	Technical Terms Count	60	28	136	62.38	26.821

4.5. Wilcoxon Signed-Rank Test Results

To test whether the differences observed between the zero-shot and pre-informed conditions were statistically significant, Wilcoxon signed-rank tests were conducted for all four quantitative indicators across 60 paired model-task observations. The results showed statistically significant differences in all four indicators. Specifically, significant increases were found for Word Count ($Z = -6.655, p < .001$), Unique Word Count ($Z = -6.495, p < .001$), Average Sentence Length ($Z = -6.044, p < .001$), and Technical Terms Count ($Z = -5.997, p < .001$).

The rank distributions also showed that the pre-informed condition outperformed the zero-shot condition in the large majority of paired comparisons. Positive ranks, which indicate higher values under the pre-informed condition, were observed in 58 of 60 comparisons for Word Count, 55 of 60 for Unique Word Count, 55 of 60 for Average Sentence Length, and 53 of 60 for Technical Terms Count. Taken together, these findings indicate that the improvements observed under the pre-informed condition were not merely descriptive, but statistically robust across all four quantitative indicators. The detailed rank distributions and Wilcoxon test statistics are reported in Table 10.

Table 10. Wilcoxon Signed-Rank Test Results

Indicator	Negative Ranks	Positive Ranks	Ties	Z	p
Word Count	2	58	0	-6.655	< .001
Unique Word Count	4	55	1	-6.495	< .001
Average Sentence Length	5	55	0	-6.044	< .001
Technical Terms Count	6	53	1	-5.997	< .001

4.6. Friedman Test Results

To examine whether the magnitude of improvement associated with the pre-informed condition differed across the three LLMs, Friedman tests were conducted on the delta scores for all four quantitative indicators. The results showed statistically significant model-level differences across all indicators. Specifically, significant differences were found for Word Count ($\chi^2(2) = 32.500, p < .001$), Unique Word Count ($\chi^2(2) = 16.300, p < .001$), Average Sentence Length ($\chi^2(2) = 18.538, p < .001$), and Technical Terms Count ($\chi^2(2) = 7.063, p = .029$). These findings indicate that the gains associated with pre-informing were not uniform across models.

The mean rank patterns showed a consistent distribution across the four indicators. In all cases, Claude obtained the highest mean rank, Gemini obtained the lowest mean rank, and ChatGPT generally occupied an intermediate position. This pattern suggests that Claude benefited the most from the pre-informed condition, while Gemini showed the most limited improvement. Although all three models benefited from pre-informing in the earlier paired comparisons, the Friedman results demonstrate that the magnitude of this benefit varied significantly by model. The model-level mean ranks and Friedman test statistics are reported in Table 11.

Table 11. Friedman Test Results

Indicator	ChatGPT Mean Rank	Gemini Mean Rank	Claude Mean Rank	N	Chi-Square	df	p
Word Count	2.25	1.00	2.75	20	32.500	2	< .001
Unique Word Count	2.15	1.30	2.55	20	16.300	2	< .001
Average Sentence Length	2.35	1.23	2.43	20	18.538	2	< .001
Technical Terms Count	1.93	1.63	2.45	20	7.063	2	.029

4.7. Interpretation of Findings

The findings indicate that the pre-informed condition was associated with consistent improvement across all four quantitative indicators used in the study. The descriptive results in table 9 showed increases in Word Count from **688.15** to **1314.62**, Unique Word Count from **373.97** to **545.77**, Average Sentence Length from **15.34** to **18.62**, and Technical Terms Count from **49.60** to **62.38**. Considered together, these increases indicate that outputs generated under the pre-informed condition were longer, contained a higher number of unique lexical items, showed higher average sentence length, and included more domain-specific terms than those produced under the zero-shot condition.

The Wilcoxon signed-rank results in table 10 showed that these differences were statistically significant across all four indicators: Word Count ($Z = -6.655$, $p < .001$), Unique Word Count ($Z = -6.495$, $p < .001$), Average Sentence Length ($Z = -6.044$, $p < .001$), and Technical Terms Count ($Z = -5.997$, $p < .001$). This indicates that the observed increases were not limited to descriptive averages, but reflected a stable pattern across the paired comparisons. The rank distributions support this interpretation: positive ranks were observed in **58 of 60** comparisons for Word Count, **55 of 60** for Unique Word Count, **55 of 60** for Average Sentence Length, and **53 of 60** for Technical Terms Count. These results show that the pre-informed condition outperformed the zero-shot condition in the large majority of model-task pairings.

The Friedman test results in table 11 further showed that the magnitude of improvement differed significantly across the three LLMs. Significant model-level differences were found for Word Count ($\chi^2(2) = 32.500$, $p < .001$), Unique Word Count ($\chi^2(2) = 16.300$, $p < .001$), Average Sentence Length ($\chi^2(2) = 18.538$, $p < .001$), and Technical Terms Count ($\chi^2(2) = 7.063$, $p = .029$). Across all four indicators, Claude obtained the highest mean ranks, Gemini obtained the lowest, and ChatGPT generally remained in an intermediate position. This pattern indicates that all three models benefited from pre-informing, but the extent of improvement was not uniform. In particular, Claude showed the largest overall gains, whereas Gemini showed the most limited gains.

Overall, the findings support the central claim of the study: the pre-informing approach was associated with measurable and statistically supported improvements across the selected quantitative indicators. At the same time, the model-level differences indicate that this improvement was not uniform across LLMs, and that the strength of the pre-informing effect varied depending on the model receiving the structured contextual input.

4.8. Potential for Further Gains

The pre-informing approach was applied through the first four stages; the iterative refinement stage was omitted. Applying this final stage is likely to enhance narrative flow, sharpen conceptual precision, and tighten contextual alignment, with any improvements still contingent on human expertise in shaping prompts and judging revisions. In short, AI-assisted writing improves as structured human guidance deepens from initial preparation to iterative refinement.

5. Conclusion and Recommendations

This study examined whether pre-informing can improve output quality as reflected in measurable indicators in black-box LLMs by supplying structured contextual input before the main task instruction. The findings indicate that, relative to zero-shot prompting, pre-informing was associated with more developed outputs across the indicator framework used in the study. In overall terms, the pre-informed condition produced consistent gains across all four indicators, and the paired comparisons showed that these differences were statistically robust rather than limited to descriptive tendencies.

The results also showed that the effect of pre-informing was not uniform across models. Although all three LLMs benefited from the pre-informed condition, the magnitude of improvement varied by model. Claude showed the largest overall gains, Gemini showed the most limited gains, and ChatGPT generally occupied an intermediate position. This indicates that pre-informing can function as a meaningful prompt-based enhancement strategy, while its effectiveness remains partly dependent on the model receiving the structured contextual input.

These findings support the central claim of the study: output quality, insofar as it is reflected in

measurable indicators, depends not only on model capacity, but also on how relevant context is structured and delivered before generation. In this respect, pre-informing may be understood as a structured preparation layer within prompt engineering that can improve observable output patterns without requiring parameter access, fine-tuning, or retraining.

These conclusions should nevertheless be interpreted with caution. Although the study expands the empirical basis of evaluation by incorporating multiple tasks and multiple LLMs, the analysis remains bounded by a specific indicator framework and a single coding structure, which limits the scope of generalization. Future research should specify clearer coding manuals, involve at least two independent raters, report inter-rater agreement, expand the diversity of tasks and models further, institute blinded review, and conduct ablation-style comparisons that isolate the mechanism of pre-informing relative to related techniques.

Practically, pre-informing and allied prompting strategies can play an important role in improving output quality, as reflected in measurable indicators, and can be deployed to improve results across tasks, particularly when zero-shot generations exhibit organizational gaps or unsupported inferences. To support replication, authors should report full prompts, random seeds, and evaluation protocols; and even as AI and LLMs advance, human expertise, creativity, and effort remain decisive, since the person who designs the prompt ultimately shapes what the model attends to and the reliability, usefulness, and accountability of the outcomes that follow.

References

- Branwen, G. (2020). *GPT-3 creative fiction*. Retrieved April 3, 2025, from <https://gvern.net/GPT-3>
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.
- Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 183-186.
- Cao, J., Li, M., Wen, M., & Cheung, S. C. (2025). A study on prompt design, advantages and limitations of chatgpt for deep learning program repair. *Automated Software Engineering*, 32(1), 1-29.
- Caruccio, L., Cirillo, S., Polese, G., Solimando, G., Sundaramurthy, S., & Tortora, G. (2024). Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach. *Intelligent Systems with Applications*, 21, 200336.
- DAIR.AI. Elements of a prompt. 2023. <https://www.promptingguide.ai/introduction/elements>
- Dauphin, B., & Siefert, C. (2025). From Llama to language: prompt-engineering allows general-purpose artificial intelligence to rate narratives like expert psychologists. *Frontiers in Artificial Intelligence*, 8, 1398885.
- Diao, S., Wang, P., Lin, Y., & Zhang, T. (2023). Active prompting with chain-of-thought for large language models. arXiv. *arXiv preprint arXiv:2302.12246*.
- Elnashar, A., White, J., & Schmidt, D. C. (2025). Enhancing structured data generation with GPT-4o evaluating prompt efficiency across prompt styles. *Frontiers in Artificial Intelligence*, 8, 1558938.
- Gao, A. (2023). Prompt engineering for large language models. *Available at SSRN 4504303*.
- Gero, K. I., Liu, V., & Chilton, L. (2022, June). Sparks: Inspiration for science writing using language models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference* (pp. 1002-1019).
- Giray, L. (2023). Prompt engineering with ChatGPT: a guide for academic writers. *Annals of biomedical engineering*, 51(12), 2629-2633.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

- Huang, J., Gu, S. S., Hou, L., Wu, Y., Wang, X., Yu, H., & Han, J. (2022). Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.
- Huang, J., Yang, D. M., Rong, R., Nezafati, K., Treager, C., Chi, Z., ... & Xie, Y. (2024). A critical assessment of using ChatGPT for extracting structured data from clinical notes. *npj Digital Medicine*, 7(1), 106.
- Ibrahim, K. A., Luk, P. C. K., Luo, Z., Ng, S. Y., & Harrison, L. (2025). Revolutionizing power electronics design through large language models: Applications and future directions. *Computers and Electrical Engineering*, 123, 110248.
- Jung, S. G., Salminen, J., Aldous, K. K., & Jansen, B. J. (2025). PersonaCraft: Leveraging language models for data-driven persona development. *International Journal of Human-Computer Studies*, 197, 103445.
- Kadavath, S., Conerly, T., Askeel, A., Henighan, T., Drain, D., Perez, E., ... & Kaplan, J. (2022). Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Knoth, N., Tolzin, A., Janson, A., & Leimeister, J. M. (2024). AI literacy and its implications for prompt engineering strategies. *Computers and Education: Artificial Intelligence*, 6, 100225.
- Lee, U., Jung, H., Jeon, Y., Sohn, Y., Hwang, W., Moon, J., & Kim, H. (2024). Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in english education. *Education and Information Technologies*, 29(9), 11483-11515.
- Lee, J. H., & Shin, J. (2024). How to optimize prompting for large language models in clinical research. *Korean Journal of Radiology*, 25(10), 869.
- Lin, X., Chen, W., Zhou, Z., Li, J., Zhao, Y., & Zhang, X. (2025). A five-dimensional digital twin framework driven by large language models-enhanced RL for CNC systems. *Robotics and Computer-Integrated Manufacturing*, 95, 103009.
- 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.
- Liu, V., & Chilton, L. B. (2022, April). Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI conference on human factors in computing systems* (pp. 1-23).
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., ... & Clark, P. (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 46534-46594.
- Ma, F., Li, D., Liu, Y., Lan, D., & Pang, Z. (2025). STEP: A structured prompt optimization method for SCADA system tag generation using LLMs. *Journal of Industrial Information Integration*, 45, 100832.
- Maharjan, J., Garikipati, A., Singh, N. P., Cyrus, L., Sharma, M., Ciobanu, M., ... & Das, R. (2024). OpenMedLM: prompt engineering can out-perform fine-tuning in medical question-answering with open-source large language models. *Scientific Reports*, 14(1), 14156.
- Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023, June). Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics* (pp. 387-402). Singapore: Springer Nature Singapore.
- Maus, N., Chao, P., Wong, E., & Gardner, J. (2023). Adversarial prompting for black box foundation models. *arXiv preprint arXiv:2302.04237*, 1(2).
- Michelet, G., & Breitingner, F. (2024). ChatGPT, Llama, can you write my report? An experiment on assisted digital forensics reports written using (local) large language models. *Forensic Science International: Digital Investigation*, 48, 301683.
- Perković, G., Drobnjak, A., & Botički, I. (2024, May). Hallucinations in llms: Understanding and

- addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)* (pp. 2084-2088). IEEE.
- Ren, Z., Ju, X., Chen, X., & Qu, Y. (2025). Improving distributed learning-based vulnerability detection via multi-modal prompt tuning. *Journal of Systems and Software*, 226, 112442.
- Reynolds, L., & McDonell, K. (2021, May). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems* (pp. 1-7).
- Sabbatella, A., Ponti, A., Giordani, I., Candelieri, A., & Archetti, F. (2024). Prompt optimization in large language models. *Mathematics*, 12(6), 929.
- Schulhoff, S., Pinto, J., Khan, A., Bouchard, L. F., Si, C., Anati, S., ... & Boyd-Graber, J. (2023, December). Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global scale prompt hacking competition. Association for Computational Linguistics (ACL).
- Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., ... & Resnik, P. (2024). The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*, 5.
- Sivarajkumar, S., Kelley, M., Samolyk-Mazzanti, A., Visweswaran, S., & Wang, Y. (2024). An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing: algorithm development and validation study. *JMIR Medical Informatics*, 12, e55318.
- Savelka, J., & Ashley, K. D. (2023). The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6, 1279794.
- Strobelt, H., Webson, A., Sanh, V., Hoover, B., Beyer, J., Pfister, H., & Rush, A. M. (2022). Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE transactions on visualization and computer graphics*, 29(1), 1146-1156.
- Taveekitworachai, P., Abdullah, F., Dewantoro, M. F., Thawonmas, R., Togelius, J., & Renz, J. (2023). ChatGPT4PCG competition: character-like level generation for science birds. *arXiv cs*.
- Team, L. (2020). *World Creation by Analogy*. <https://aidungeon.medium.com/world-creation-by-analogy-f26e3791d35f>
- Tian, Y., Liu, A., Dai, Y., Nagato, K., & Nakao, M. (2024). Systematic synthesis of design prompts for large language models in conceptual design. *CIRP Annals*, 73(1), 85-88.
- Tromly, K. (2001). Renewable energy: An overview. energy efficiency and renewable energy clearinghouse (EREC) Brochure. *Department of Energy: USA*, 200.
- Twidell, J. (2021). *Renewable energy resources*. Routledge.
- Vogelsang, A. (2024). From specifications to prompts: On the future of generative large language models in requirements engineering. *IEEE Software*, 41(5), 9-13.
- Wan, Y., Chen, Z., Liu, Y., Chen, C., & Packianather, M. (2025). Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing. *Advanced Engineering Informatics*, 65, 103212.
- Wang, B., Deng, X., & Sun, H. (2022). Iteratively prompt pre-trained language models for chain of thought. *arXiv preprint arXiv:2203.08383*.
- Wang, M. H., Jiang, X., Zeng, P., Li, X., Chong, K. K. L., Hou, G., ... & Pan, Y. (2025). Balancing accuracy and user satisfaction: the role of prompt engineering in AI-driven healthcare solutions. *Frontiers in Artificial Intelligence*, 8, 1517918.
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... & Fedus, W. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.
- White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

- Wu, T., Terry, M., & Cai, C. J. (2022, April). Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems* (pp. 1-22).
- Xue, T., Wang, Z., Wang, Z., Han, C., Yu, P., & Ji, H. (2023). Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought. *arXiv preprint arXiv:2305.11499*.
- Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., ... & Wang, L. (2023). Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.
- Yao, Y., Li, Z., & Zhao, H. (2023). Beyond chain-of-thought, effective graph-of-thought reasoning in language models. *arXiv preprint arXiv:2305.16582*.
- Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B., & Yang, Q. (2023, April). Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems* (pp. 1-21).
- Zhang, K., Zhou, F., Wu, L., Xie, N., & He, Z. (2024). Semantic understanding and prompt engineering for large-scale traffic data imputation. *Information Fusion, 102*, 102038.
- Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., ... & Chi, E. (2022). Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.
- Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022, November). Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Arastırma Makalesi

The Pre-Informing Approach and a Synthesis of Existing Prompting Techniques for Improving Output Quality in Black-Box Large Language Models

Black-Box Büyük Dil Modellerinde Çıktı Kalitesini Artırmak için Pre-Informing Yaklaşımı ve Mevcut İstem Tekniklerinin Sentezi

Muhammed Lütfü ODABAŞOĐLU Yüksek Lisans Öğrencisi, Kafkas Üniversitesi İktisadi İdari Bilimler Fakültesi 242624002@ogr.kafkas.edu.tr https://orcid.org/0009-0001-7809-9157	Muhammed Akif YENİKAYA Dr. Öğr. Üyesi, Kafkas Üniversitesi İktisadi İdari Bilimler Fakültesi akif.yenikaya@kafkas.edu.tr https://orcid.org/0000-0002-3624-722X
--	--

Genişletilmiş Özet

Bu çalışma, Büyük Dil Modellerinin (LLM) sentetik veri üretimindeki etkinliğinin yalnızca model kapasitesine değil, insan yönlendirmeli istem (prompt) yapısının niteliğine de bağlı olduğu kabulünden hareketle, literatürde yer alan yirmi üç istem yazım tekniğini incelemekte ve ana görev talimatı öncesinde yapılandırılmış bağlamsal hazırlık yoluyla çıktı kalitesini artırmayı amaçlayan pre-informing adlı hibrit bir yaklaşımı önermektedir.

Çalışmada pre-informing yaklaşımı şu adımlarla tanımlanmaktadır: LLM'lere ana görev talimatı verilmeden önce görevin tanımı yapılmakta, ancak çıktı üretilmemesi istenmektedir. Görev anlaşıldıktan sonra toplanmış kavramsal veriler LLM ile sistematik olarak paylaşılmakta, bu veri besleme adımı tamamlandığında ana görev talebi iletilmekte ve çıktı alınmaktadır.

Yaklaşımında, LLM'ye doğrudan görev talimatı verilmeden önce göreve ilişkin kavramsal ve doğrulanmış bilgilerin (makaleler, dokümantasyon vb.) toplanması, ardından bu verilerle LLM'nin beslenmesi ve her beslemeden sonra paylaşılan bilgilerin özetlenerek anlaşıldığının teyit edilmesi yoluyla alınacak çıktının niteliğinde kayda değer bir artış hedeflenmektedir. Her besleme adımının ardından teyit alınarak, birden çok kaynağın tek seferde gönderilmesiyle ortaya çıkabilen yüzeysel tarama eğiliminin azaltılması, sınırlı bağlam penceresinin daha verimli kullanılması ve modelin göreve ilişkin bağlamsal odağının güçlendirilmesi amaçlanmaktadır. Nihai çıktı sonrasında uygulanabilecek iteratif ve dinamik istem adımları yaklaşımın genişletilebilir bir parçası olmakla birlikte, bu çalışmada temel karşılaştırma pre-informing aşamasının ana üretim üzerindeki etkisine odaklanmaktadır.

Pre-informing yaklaşımının pratik testi için eğitim alanına ait 20 yazma ve analiz görevi seçilmektedir. Bu görevler öğretim tasarımı, kurumsal politika, stratejik planlama ve sosyo-ekonomik analiz olmak üzere dört başlık altında yapılandırılmaktadır. İlk görevde pre-informing uygulama adımları ayrıntılı biçimde tek tek gösterilmekte, kalan on dokuz görevde ise aynı istem örüntüsü ve işlem mantığı korunarak uygulama sürdürülmektedir. Böylece yöntem, yalnızca sınırlı iki örnek üzerinden değil, daha geniş ve çeşitlendirilmiş bir görev kümesi üzerinde sınanmaktadır.

Tüm üretimler Şubat 2026 tarihinde, ChatGPT 5.2 Standard, Gemini 3 Fast ve Claude Sonnet 2.6 modelleri kullanılarak, temiz oturum koşullarında, sabit örnekleme ayarlarıyla ve ince ayar uygulanmaksızın gerçekleştirilmektedir. Değerlendirme ortak bir nicel çerçeve üzerinden kurgulanmaktadır. Bu çerçevede Kelime Sayısı (Word Count), Benzersiz Kelime Sayısı (Unique Word Count), Ortalama Cümle Uzunluğu (Average Sentence Length) ve Teknik Terim Sayısı (Technical Terms Count) göstergeleri kullanılmaktadır. Çıktılar önce betimsel olarak karşılaştırılmakta, ardından eşleştirilmiş karşılaştırmalar için Wilcoxon işaretli sıralar testi, modeller arası iyileşme farklarını

incelemek için ise Friedman testi uygulanmaktadır. Ayrıca kullanılan gösterge setinin birlikte yorumlanabilirliğini desteklemek amacıyla güvenilirlik analizi yapılmakta ve standartlaştırılmış Cronbach's alpha katsayısı .887 olarak hesaplanmaktadır.

Bulgular, pre-informing koşulunun dört nicel göstergenin tamamında zero-shot koşuluna göre daha yüksek değerler ürettiğini göstermektedir. Ortalama Kelime Sayısı 688.15'ten 1314.62'ye, Benzersiz Kelime Sayısı 373.97'den 545.77'ye, Ortalama Cümle Uzunluğu 15.34'ten 18.62'ye ve Teknik Terim Sayısı 49.60'tan 62.38'e yükselmektedir. Bu örüntü, pre-informing'in çıktıları daha kapsamlı, daha çeşitli ve görev bağlamına daha zengin biçimde duyarlı hale getirdiğini göstermektedir. Wilcoxon işaretli sıralar testi sonuçları da bu farkların yalnızca betimsel düzeyde kalmadığını; dört göstergenin tamamında istatistiksel olarak anlamlı olduğunu ortaya koymaktadır.

Elde edilen sonuçlar, pre-informing'in etkisinin tüm modellerde aynı büyüklükte olmadığını da göstermektedir. Friedman testi sonuçları, iyileşme düzeylerinin modeller arasında anlamlı biçimde farklılaştığını ortaya koymaktadır. Tüm göstergelerde Claude en yüksek ortalama sıra değerlerini alırken, Gemini en düşük değerleri almakta, ChatGPT ise genel olarak ara bir konumda bulunmaktadır. Bu bulgu, pre-informing yaklaşımının genel olarak yararlı olduğunu, ancak etkisinin modele duyarlı biçimde değiştiğini göstermektedir.

Elde edilen sonuçlar, pre-informing'in muhtemel etki örüntüsünü anlamaya da ışık tutmaktadır. Girdilerin ardışık biçimde beslenmesi ve her adımda anlama teyidi alınması, modelin nihai görev talimatına daha hazırlıklı bir bağlamsal zeminle yaklaşmasını sağlayabilmektedir. Bu durum, üretimin göreve daha yakın biçimde yönlenmesine ve çıktının ölçülebilir göstergeler bakımından güçlenmesine katkı sunmaktadır. Bu gözlem, çalışmanın başlangıçta ortaya koyduğu araştırma sorunsalına uygun biçimde, insan yönlendirmesinin ve istem kurgusunun model davranışını şekillendirmede belirleyici rol oynadığını desteklemektedir.

Çalışmada sınırlılıklar açıkça belirtilmektedir. Her ne kadar görev sayısı ve model çeşitliliği artırılmış olsa da değerlendirme hâlen dört nicel gösterge ve tek bir kodlama çerçevesi ile sınırlandırılmıştır. Bağımsız değerlendirici, kör değerlendirme ve değerlendiriciler arası uyum gibi unsurların gelecekte güçlendirilmesine ihtiyaç bulunmaktadır. Ayrıca pre-informing protokolünün isteğe bağlı yinelemeli iyileştirme adımı bu çalışmada karşılaştırmanın merkezine alınmamaktadır. Farklı görev aileleri, farklı model sürümleri ve çoklu değerlendiricili tasarımlarla yapılacak genişletmelerin, yöntemin hangi alt adımlarının asıl katkısını sağladığını daha açık biçimde ortaya koyması beklenmektedir.

Bulgular ışığında yapılan genel çıkarım, LLM çıktılarının niteliğinin yalnızca model kapasitesine değil, üretim öncesi hazırlığın kurgusuna, istem yapısına ve dolayısıyla insan faktörüne de güçlü biçimde bağlı olduğudur. Bu çerçevede çalışma, pre-informing yaklaşımını kapalı kutu ortamlarda ölçülebilir göstergelere yansıyan çıktı kalitesini artırmaya yönelik yapılandırılmış bir çerçeve olarak konumlandırmakta ve model davranışını yönlendirmede insan unsurunun merkezi önemini vurgulamaktadır.