International Journal on Robotics, Automation and Sciences

Enhancing LLM Efficiency: A Literature Review of Emerging Prompt Optimization Strategies

Asyafa Ditra Al Hauna*, Andi Prademon Yunus, Masanori Fukui and Siti Khomsah

***Abstract* – This study focuses on enhancing the performance of Large Language Models (LLMs) through innovative prompt engineering techniques aimed at optimizing outputs without the high computational costs of model fine-tuning or retraining. The primary objective is to investigate efficient alternatives, such as black-box prompt optimization and ontology-based prompt refinement, which improve LLM performance by refining prompts externally while maintaining the model's internal parameters. The study explores various prompt optimization techniques, including instruction-based, role-based, question-answering, and contextual prompting, alongside advanced methods like CoT and ToT prompting. Methodologically, the research involves a comprehensive literature review, benchmarking prompt optimization techniques against existing models using standard datasets such as Big-Bench Hard and GSM8K. The study evaluates the performance of approaches like APE, PromptAgent, self-consistency prompting, and many more. The results demonstrate that these techniques significantly enhance LLM performance, particularly in tasks requiring complex reasoning, multi-step problem-solving, and domain-specific knowledge integration. The findings suggest that prompt engineering is crucial for improving LLM efficiency without excessive resource demands. However, challenges remain in ensuring prompt scalability, transferability, and generalization across different models and tasks. The study highlights the need for further research on integrating ontologies and automated prompt generation to refine LLM precision and adaptability, particularly in low-resource settings. These advancements will be vital for maximizing the utility of LLMs in increasingly complex and diverse applications.**

Keywords—Prompt Optimization, Prompt Engineering, Black-Box, Ontology, Large Language Models.

I. INTRODUCTION

The rapid advancement of LLMs has led to their widespread use in automating diverse tasks. Despite their growing application, the effectiveness of LLMs in executing specific tasks remains heavily influenced by the quality of the prompts provided. Latest studies indicate that variations in prompt formatting can produce differing outcomes [1]. Recent approaches have been introduced to enhance the performance of LLMs through prompt engineering. For instance, ensemble prompt techniques have been proposed to boost the efficacy of in-context learning (ICL) [2]. Additionally, fine-tuning (adapting pre-trained LLM to task-specific data) strategies that incorporate prompts into the fine-tuning process have been suggested to address challenges such as hallucination and reproducibility issues [3]. Lately, a new role, termed prompt engineer, has emerged to address the challenge of prompt optimization. Considerable research efforts have been directed toward enhancing LLM prompts through various methodologies. These include directly utilizing LLMs' pre-existing knowledge and capabilities, fine-tuning them for particular

*Corresponding Author, email: alditra@student.telkomuniversity.ac.id ORCID: https://orcid.org/0009-0005-3098-5386
Asyafa Ditra Al Hauna is with Faculty of Informatics, Telkom University, Jl. DI. Panjaitan 128, Purwokerto, 53147, Jawa Tengah, Indonesia (e-mail: alditra@student.telkomuniversity.ac.id).
Andi Prademon Yunus is with Faculty of Informatics, Telkom University, Jl. DI. Panjaitan 128, Purwokerto, 53147, Jawa Tengah, Indonesia (e-mail: andiay@telkomuniversity.ac.id).
Masanori Fukui, Iwate Prefectural University, Japan (e-mail: fukui_m@iwate-pu.ac.jp).
Siti Khomsah is with Faculty of Informatics, Telkom University, Jl. DI. Panjaitan 128, Purwokerto, 53147, Jawa Tengah, Indonesia (e-mail: sitijk@telkomuniversity.ac.id).

International Journal on Robotics, Automation and Sciences (2025) 7, 1:72-82
https://doi.org/10.33093/ijoras.2025.7.1.9
Manuscript received: 3 Oct 2024 | Revised: 13 Dec 2024 | Accepted: 25 Dec 2024 |
Published: 31 Mar 2025

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

applications, or retraining the models for specialized tasks. However, fine-tuning or retraining often incurs significant computational and financial costs. To address these limitations, a technique known as black-box prompt optimization has been introduced, which seeks to improve LLM performance without modifying the model's internal parameters, offering a more cost-effective solution by optimizing solely through the prompt. Furthermore, an additional method, ontology-based enhanced prompt optimization, has been proposed, which incorporates knowledge representation to refine the prompts provided to LLMs, thereby enhancing their overall efficiency.

II. TRENDS

Figure 1 depicts a significant increase in research on prompt optimization in 2023. This surge can be attributed to several pivotal events, including the rise of open-source LLMs and the availability of powerful LLMs that offer APIs, enabling researchers to utilize and test these models with the methods they propose. Additionally, the growing urgency for effective human-AI collaboration has highlighted challenges in crafting prompts that guide LLMs in executing assigned tasks.

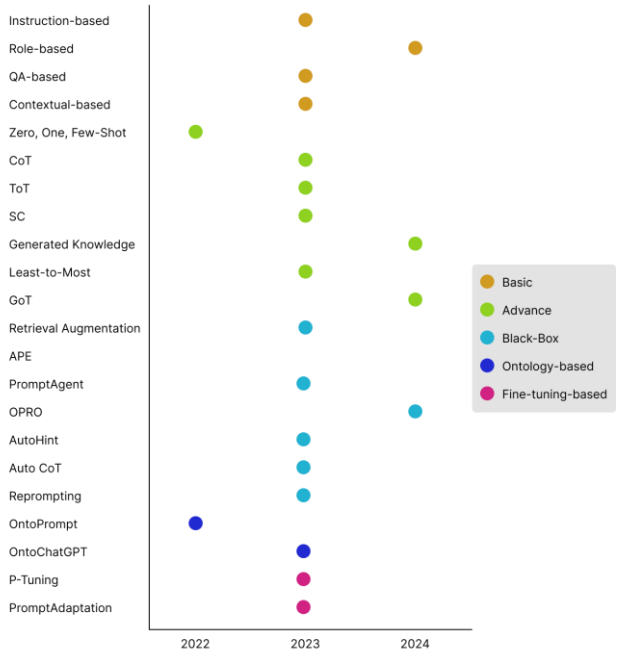

FIGURE 1. Trend of prompt optimization methods.

III. LITERATURE REVIEW

A. Prompt Engineering Techniques

(basics) Instruction-based prompt

In natural language processing (NLP), prompt engineering involves developing instruction-based input prompts that direct language models to perform specific tasks efficiently. These prompts are crafted to mirror human-like instructions. For instance, in [4], instruction-based prompting is utilized to delineate the scope and limitations of a given task and address tasks related to fallacy detection, using prompts such as "Given a text segment, identify the fallacy," A similar approach is utilized in [5] during the instruction tuning phase of the language model for further refinement and enhancement of its performance in zero-shot learning tasks, where the model is required to perform a task it has not been explicitly trained on by leveraging general knowledge and contextual understanding derived from its training data. With instruction tuning, the model becomes exposed to all types of instructions and learns how to generalize between different tasks for improved adaptability.

(basics) Role-based prompt

Since LLMs are typically trained on diversified sets of tasks, their general-purpose capabilities tend to be more general. It obviously may come at the cost of being less effective in specialized or domain-specific requests. In order to overcome this challenge, various techniques of role prompting have been advanced, especially instructing the LLM to behave like an expert in some domain. For example, a role may be specified thus: "You are a math professor, [task], [desired result format]," Moreover, research supports the view that role prompting enhances the reasoning capabilities of LLMs. In work [6], role prompting outperformed the zero-shot method on 12 datasets dedicated to reasoning. In [7], multi-domain adaptation is allowed for LLMs by introducing three key components: self-distillation, role prompting, and role integration. Its setting has been proven effective for tackling some particular challenges of LLMs, such as catastrophic forgetting or interdomain confusion.

(basics) Question-Answering-based Prompt

Because ambiguous or irrelevant responses have occurred too frequently, LLMs rely on additional context to better understand the task or question. Related to this problem, the answer needs a clearer context and vague framing. The development of a question-answering-based prompt technique has addressed this problem. The approach utilizes templates such as "[context] [question] [desired output format]," [8] where context provides relevant background information to help the model understand the scope or domain of the problem. In contrast, the desired output format specifies the expected response type: a closed-one, for example, yes/no, or an opener with specific formats. Likewise, in [9], this question-answering prompt strategy was used, where they performed data augmentation, a technique applies to creating additional training data by modifying existing data. It created relevant data, and the consistency was strict because it combined cloze (fill-in-the-blank exercises designed to predict the missing words) and QA tasks. In addition, in [10], the authors constructed ProQA by constructing the input schema similarly in order for the model to balance knowledge generalization among the QA tasks and allow the model to tailor knowledge customization toward every individual QA task.

(basics) Contextual-based Prompt

While applications can be general, contextual prompting has become a major stride in increasing language models' powers, particularly in domain-specific applications. Although conventionally, LLMs have relied on graph-based structures to supplement their knowledge of domain-specific entities [11], this is

commonly insufficient as many entities are underrepresented or missing in today's knowledge bases. One way to address this challenge is the introduction of contextual prompting, which integrates information relevant to the task into the prompts to improve the model's understanding capability. This approach was explored in [12], where textual data were provided as a prompt, which fine-tuned the model and allowed it to learn from the particular tasks and knowledge encoded in texts. Additionally, contextual prompting works around the limitation of static prompting (pre-defined prompts that remain unchanged regardless of the task), which is mostly inappropriate for dynamic contexts. For example, static prompts in [13] cannot handle multi-turn dialogue scenarios. At the same time, contextual prompting allowed much flexibility and responsiveness, which was more befitting for task-oriented systems.

(Advance) Zero-shot, one-shot, and Few-Shot Prompting

Zero-shot prompting is a method where task instructions are given to the LLM alone, without any additional supporting task-specific information. The model draws only on knowledge acquired during previous training in this approach [14]. Countless research has been conducted showing that, without resorting to expensive task-specific fine-tuning, models can approach and solve acquainted complex tasks successfully: [15], [16], [17], [18]. In privacy areas, zero-shot prompting has been used to create systems that generate sanitized documents by minimizing the risks of deanonymization attacks, attempts to re-identify individuals from data that has been anonymized. This is preferred because it is important to maintain privacy; it avoids the use of sensitive data or particular tasks to train models [19]. Another field of application for zero-shot prompting has been clinical information extraction. Here, it helps address the challenges brought about by the scarcity of labeled data in clinical NLP [20].

On the other hand, one-shot prompting means that a single example or 'shot' is contained in the prompt given to the language model. This one sample represents how a task is accomplished [21]. One such application with one-shot prompting in SPARQLGEN reduces dependence on resource-intensive training by generating SPARQL queries from natural language input. Training in such a manner requires only the model to provide one SPARQL example query and a fragment of the knowledge graph to generalize and build needed queries without special training [22]. In the Automated Short Answer Grading (ASAG) model, a model designed to assess students' short textual responses, adaptation has been done so that one-shot prompting is used to extract key points from the justification in students' responses. Typically, in evaluation, multi-part questions, which are indispensable, especially in low-resource educational settings [23], would be critical for assessing correctness [80], [82].

Few-shot prompting involves giving a small number of task examples along with the prompt to enable the LLMs to recognize patterns and generalize to unseen queries. It is advantageous, as it is extremely difficult to collect large amounts of annotated or labeled data in many cases. Indeed, as shown in [24], few-shot prompting can be adapted to prompt-based fine-tuning by aligning prompt formats during pretraining (training process lets a model learn foundational patterns and representations) and fine-tuning. It would render the fine-tuning examples more informative in a few-shot learning setting. Few-shot prompting has also been performed in [25] to enhance the performance of in-context learning over downstream tasks without the instability within ICL due to variation in prompt construction. This technique significantly enhances model generalization by carefully selecting a few examples. Few-shot prompting has also been effectively used to handle the challenge of modeling heterogeneous graphs that use scarce labeled data, realized in the HGPROMPT framework. It greatly improves the performance of GNNs and HGNNs for few-shot tasks, as shown in [26].

(Advance) Chain-of-Thought (CoT) Prompting

Chain-of-thought prompting is a teaching methodology that's supposed to guide LLM models through the steps of reasoning to solve complicated problems by breaking them down into their granular components. To this end, as shown in the study by [27],[81] it allows for more organized and logical thinking processes, thus enabling their users to comprehend better and create valuable ideas. Moreover, [28] showed that applying this kind of prompting technique in LLMs like GPT-4 increased performance not only on diverse sets of tasks, from medical board exams up to multi-step science-based questions, which reached 0.83 by Krippendorff's alpha, statistical measure to evaluate the consistency of model outputs or their alignment with human judgments across various tasks. That means chain-of-thought prompting is an effective way to improve the LLM processing capability in complex situations. Similarly, in [29], an experiment conducted on two of the most simple mathematical settings—arithmetic expression evaluation and solving linear equations—provided evidence that for such basic tasks, bounded-depth transformer models cannot solve them without a chain of thought prompting if their model sizes are not increased polynomially concerning the input lengths. Regulated, however, was this performance by a chain of thought prompting, where guidance through step-by-step processes in LLMs used the structure of mathematical language to facilitate the completion of a task.

(Advance) Tree of Thought (ToT) Prompting

Traditional prompting techniques for LLMs often yield single-step responses, even when methods such as CoT prompting are employed. CoT primarily facilitates a linear reasoning path, which presents a challenge because LLMs lack the human-like ability to consider multiple possible steps before determining the optimal solution. To address this limitation, the TOT prompting technique was introduced, enabling models to engage in a branching thought process. This approach allows the model to consider intermediate steps, hypotheses, and various potential paths to solve problems, effectively generating a "tree" of possible solutions [30].

Subsequent research has demonstrated that TOT improves upon COT by allowing LLMs to explore multiple reasoning paths through mechanisms of self-evaluation and backtracking [31]. In this study, TOT proved to be more effective than COT, particularly in tasks requiring complex planning, such as the Game of 24, creative writing, and mini crosswords. Results indicated that GPT-4 using the TOT framework successfully completed 74% of tasks, compared to only 4% when using COT. Moreover, the TOT approach has been shown to outperform COT in multi-hop question answering (MHQA). A study implementing the STOC-TOT method demonstrated a significant improvement in LLM reasoning capabilities [32]. By breaking down complex questions into sub-questions, evaluating alternative reasoning paths, and applying probabilistic estimates to avoid dead ends, the method increased the accuracy by 7% and the F1 score by 7.8 points, highlighting the potential of TOT in addressing more complex problem-solving tasks.

(Advance) Self-Consistency (SC) prompting

The self-consistency prompting technique, similar to the ToT prompting method, enhances the CoT prompting strategy. The primary distinction between these approaches lies in the reasoning process employed. While ToT utilizes a tree-based paradigm to explore multiple reasoning paths simultaneously, self-consistency generates various reasoning pathways through sampling from the decoder, as opposed to the conventional greedy decoding method.

Research [33] outlines that the self-consistency technique involves three key steps. First, the language model is prompted using CoT to decompose a given task into smaller reasoning steps. Second, the model is directed to generate multiple reasoning pathways via sampling, replacing greedy decoding (method to construct sequences by iteratively selecting the token with the highest probability at each step) with exploring alternative approaches. It allows the model to avoid always choosing the token or word with the maximum immediate probability and, instead, investigates a wider variety of possible reasoning pathways. Third, the generated reasoning paths are scored, and the answer is selected based on the path showing the highest consistency, determined by marginalizing over the sampled reasoning paths. The RASC method [34] further develops this technique. This approach improves some of the issues inherently part of traditional self-consistency mechanisms (strategies designed to improve the reliability and coherence of a model's outputs by ensuring alignment across multiple predictions or reasoning paths), such as hallucination and inefficiency regarding computation. RASC adjusts the sample size dynamically based on the output's consistency and the quality of the reasoning pathways. As a result, this technique is more effective and efficient than prior SC techniques [30], Escape Sky-High Cost [35], and Adaptive-Consistency [36].

(Advance) Generated Knowledge Prompting

Generated knowledge prompting is an explicit approach whereby language models are taught to generate knowledge or context relevant to the test first before giving it the final relevant answer as required in the question paper [37]. Successful usages in many domains based on CLINGEN [38] have been reported among many applications. Clinical knowledge graphs serve as the external knowledge source to be infused for better performance in these experiments when the resources become scarce. This approach led to an average performance improvement of 8.7% with PubMedBERT base and 7.7% with PubMedBERT large scale, compared to the ZeroGen [39] and DemoGen [40], [41] methods. The CLINGEN framework has shown significant capability in improving the quality and diversity of synthetic clinical data, particularly in resource-constrained environments.

As well, research on the SPARTA framework [42] proposed a multi-stage knowledge transfer approach to address the challenges of zero-shot learning, particularly in conversational question generation. Applying generated knowledge prompting within this framework reduced the gap between single-turn and multi-turn conversations by synthesizing conversational history and integrating referential elements, such as anaphora. This approach facilitates the creation of realistic conversational questions under zero-shot conditions without annotated conversational data.

(Advance) Least-to-Most Prompting

The least to most prompting technique has been developed to address the limitations in reasoning processes within the Chain of Thought (COT) approach, mainly when dealing with tasks that are more complex than those provided as examples in the prompt. A key challenge CoT faces is its difficulty in managing compositional generalization, where its performance significantly declines when confronted with tasks that are more challenging than anticipated [43], [44].

The method was first introduced in a study [45], which proposed a two-step approach for solving complex problems: decomposition and subproblem solving. In the decomposition phase, the prompt is designed to break down the main problem into subquestions, allowing the reasoning process to be divided into smaller, more manageable parts. Subsequently, in the subproblem-solving phase, each prompt includes the subquestion and provides the intermediate answer from the preceding subquestion to guide the solution to the next problem. Findings from this study indicated that LLMs, such as GPT-3, employing the least to most prompting technique, demonstrated superior performance in compositional generalization tasks compared to the CoT approach. Using only 14 exemplars, this technique achieved an accuracy of up to 99%, whereas CoT attained only 16% accuracy.

(Advance) Graph of Thought (GoT) *Prompting*

Methods such as CoT and ToT rely on linear or sequential reasoning. While these approaches have demonstrated effectiveness in specific contexts, they exhibit limitations when confronted with the complexities inherent in human reasoning. Humans can connect diverse ideas and solutions through various relationships not confined to linearity. As a result, purely linear reasoning is often insufficient to

produce optimal solutions, particularly for tasks that require a high degree of complex reasoning.

To address the challenges posed by the non-linearity of human cognition, recent research on LLMs, as presented in the study [46], has proposed using a GoT method. This approach enables LLMs to mimic human cognitive processes by employing a graph structure, wherein nodes represent concepts and edges depict the relationships between those concepts. It represents an accuracy gain of 87.59%, a 2.4% increase compared with the CoT method on the test set AQUA-RAT when using the T5-base model. Another work [47] applied the method GoT to handle the challenges that arise in complex multi-step logical reasoning tasks—those in which CoT and ToT could not demonstrate peak performance. Next, an enhanced version of the approach was presented by adding a checker function to show a better precision score against the standard scoring systems. It demonstrated an even better result in GPT-4 with greater accuracy on tasks like the 24-point game, solving high-degree polynomial equations, and deriving formulas for recursive sequences. A study also introduces the Rex-GoT framework to address challenges in Dialogue Commonsense Multi-Choice Question Answering (DC-MCQ) [48]. This framework is put forth to deal with issues like option saturation, where an increase in the number of options confuses the model. The other challenge is the clue labyrinth, the terminology for the entangled processing of combinations of clues. In both cases, features are combinations of questions, options, and clues implicating complex information about predicted information.

(Advance) Retrieval augmentation Prompting

A retrieval augmentation prompt enslaves LLMs to retrieve relevant information from external sources and combines it with internally stored knowledge for more accurate and up-to-date answers [49]. It would help resolve several problems that are part of the general course of operation involved with LLMs, such as hallucinations and issues with truthfulness [50]. In the case of studies that have focused on Non-Knowledge-Intensive (NKI) tasks, two main challenges, such as the requirement of diversity in relevance ranking and the trade-off between training costs and performance of tasks [51], have been tackled effectively by prompting-based retrieval augmentation techniques. The results are that PGRA outperforms FiD [52] and RAG [53] using T5-large on more datasets, such as SST-2 [54] and CoLA [55]. Further research has used prompt-based RAG techniques to address issues of code completion. LLMs need more semantic understanding to complete code, which requires more depth than the training they have gotten [56]. This involves hallucinations and semantic lexical limitations that limit the model from fully comprehending the abstract structure of the code.

B. Blackbox Optimization

Blackbox prompt optimization is a technique for optimizing prompts using an LLM without modifying any parameters. In other words, it does not contain processes such as training or fine-tuning that modify the model's parameters; rather, it aims to maximize model capabilities using only prompt engineering [57]. Where most methods would stop, this one goes one step further: Blackbox prompt optimization presented here considers three widely known benchmark datasets to evaluate LLM performance on various tasks. These datasets include Big-Bench [58], GSM8K [59].

TABLE 1. Methods accuracy.

Method	Big-Bench	GSM8K
APE[60]	-	43.0
PromptAgent[61]	83.9	-
OPRO[62]	82	80.2
AutoHint[63]	90.15	-
AutoCoT[64]	-	62.8
Reprompting[65]	99.6	-

Automatic Prompt Engineering (APE)

Automation of creating expert-level LLM prompts serves as one of the most prominent tasks in increasing the models' performance. Manual prompt generation by humans often involves a time-consuming process of trial and error, particularly in specialized cases. To address this challenge, methods such as APE [66] have been introduced to automate the process. APE consists of a sequence of steps, including the generation, evaluation, and optimization of prompts in a continuous loop. Initially, this method inferentially generates candidate prompts, which are then evaluated using log probability metrics and execution accuracy. High-scoring candidates are resampled to maintain quality, while new candidates are also explored. This process is conducted iteratively using a Monte Carlo search algorithm.

The application of the APE method, combined with the CoT approach, as demonstrated in prior research [14], has been shown to enhance the performance of existing CoT models on datasets such as MultiArith [67] and GSM8K. On the MultiArith dataset, performance increased from 78.7 to 82.0, while on GSM8K, it improved from 40.7 to 43.0 as shown in Table 1 measured in normalize performance. Furthermore, on the BIG-Bench Hard (BBH) dataset, dataset designed to evaluate the reasoning capability of LLMs, the APE method exhibited superior performance, outperforming human-generated prompts in 17 out of 21 tasks under Few-Shot and Zero-Shot experimental settings. Despite this method's effectiveness, specific limitations exist in its implementation. Larger and more advanced LLMs are more effective at generating high-quality prompts, but they also incur higher per-token costs. While smaller models can occasionally generate effective prompts, their success rate is significantly lower. Consequently, even with a large number of iterations using smaller models, achieving optimal results remains challenging.

PromptAgent

PromptAgent [61] attacks the same challenge as APE. By contrast, a key feature of PromptAgent is that

it frames the optimization of prompts in terms of a strategic planning problem—maintaining a more systematic way to improve the quality of prompts. Indeed, empirical evidence supports that PromptAgent outperforms the APE method in this respect. This approach couples Monte Carlo Tree Search with an error feedback mechanism to explore a large space of possible prompts. During this, PromptAgent assesses the intermediate prompts and adjusts them according to the error feedback to find the optimal path that yields the highest reward.

The main advantage of this approach is that it uses MCTS as a search strategy. It has already proved to perform considerably better than other approaches. For instance, PromptAgent using MCTS reached an accuracy of 0.754 against the BBH task, outperforming other alternatives such as MC [60], Beam Search [68], and Greedy. Additionally, the optimized prompts demonstrated strong transferability, achieving accuracies of 0.776 on the base GPT-3.5 model, 0.839 on GPT-4, and 0.441 on the weaker LLM model, PaLM 2—consistently outperforming both human-crafted prompts and the APE. However, the method has its limitations. One notable drawback is that its accuracy needs to be improved in the Chain-of-Thought (CoT) approach in specific tasks, such as object counting, which requires step-by-step reasoning to achieve optimal results as shown in Table 1.

Optimization by Prompting (OPRO)

Optimal Prompt Refinement (OPRO) is an iterative method that optimizes tasks described within a meta-prompt (a prompt in natural language) by leveraging a LLM as both the optimizer and evaluator [69]. This process involves using a meta-prompt containing task descriptions and solution-score pairs, which the LLM optimizes to generate several candidate solutions. The process terminates if the best solution is identified; otherwise, the generated solutions are evaluated and assigned scores. These scores are then fed back into the meta-prompt for further optimization. The cycle continues until the highest-scoring optimal solution is found.

OPRO has demonstrated significant performance improvements across various tasks, such as achieving 80.2% accuracy on the GSM8K dataset and 82% accuracy on BBH tasks compared to baseline prompts as shown in Table 1. These results exceed the performance of manually designed human prompts, underscoring the effectiveness of this LLM-based optimization method. The adaptability of OPRO across different LLM architectures is noteworthy. In studies, it has been successfully applied to a range of models, including PaLM [70], Text-Bison, and GPT [71], indicating that OPRO functions effectively across diverse LLM architectures.

However, OPRO is not intended to replace gradient-based optimization algorithms used for continuous mathematical optimization, nor is it designed for more specialized methods required for classic combinatorial optimization problems, such as the Traveling Salesman Problem (TSP). Additionally, OPRO encounters challenges when dealing with large-scale problems due to LLMs' limited context window size, which constrains the amount of data or problem descriptions that can be included in a single prompt. This limitation is particularly problematic for tasks like high-dimensional linear regression (task to predict a continuous number) or more complex problems such as TSP [72].

AutoHint

The AutoHint framework is designed to generate enriched instructions automatically, or “hints,” from input-output demonstrations to refine original prompts and improve the performance of LLMs [73]. This framework addresses challenges in zero-shot and few-shot learning settings, where LLMs often struggle to understand tasks when no examples are provided fully. Even in few-shot settings, where answers tend to be more detailed, performance can be influenced by the order or selection of samples included in the prompt. To mitigate these issues, a residual-sampling-summarize (three steps method to optimize prompts consists of residual sampling, hint generation, summarization and refinement) technique is employed.

In terms of strengths, the framework demonstrates strong performance, particularly in zero-shot settings. It increases accuracy across five of six BIG-Bench Instruction Induction (BBII) tasks, with notable improvements in epistemic reasoning and hyperbaton tasks (task to evaluate LLM’s ability to determine the correct order of adjectives in english sentences), where balanced accuracy (performance metric used to evaluate the model’s classification ability) significantly rises to 90.15 as shown in Table 1 from the baseline values of 82.9. Additionally, it reduces evaluation costs by employing random sampling when the validation set (part of a whole data used during the training of a model to evaluate its performance) is too large. However, the framework has limitations. Performance decreases with additional iterations, especially on epistemic reasoning and Winowhy tasks. This decline also highlights the framework’s restricted generalization capability across all tasks, as evidenced by a drop in accuracy for logical fallacy detection from 86.76 to 84.64 in few-shot settings.

Auto CoT

CoT prompting technique is frequently employed in LLMs to optimize performance on multi-reasoning tasks by encouraging step-by-step thinking. However, this approach may fail in some cases, particularly under a Zero-Shot setting. Although prompts like “let us think step by step” [14] can reduce such failures, they do not fully address one of the leading causes: the diversity of demonstration questions. To tackle this issue, the AutoCoT method [64] was developed, which designs step-by-step reasoning demonstrations through two primary stages: question clustering and demonstration sampling, both conducted automatically. This method has proven effective in addressing the challenge of question diversity, showing adaptability and flexibility, as the demonstrations are tailored automatically to the specific task. In the study, CoT outperformed previous methods such as Zero-Shot [14], Zero-Shot-CoT [14],

Few-Shot [74], and Manual-CoT [74], especially in arithmetic and symbolic tasks, task designed to process and manipulate symbols such as numbers, letters, or logical expression. As shown in the Table 1, AutoCoT achieve 62.8 accuracy in GSM8K task while using the codex LLM.However, its performance on commonsense tasks did not exceed these methods by more than 5 points. While suitable for reasoning tasks involving arithmetic or commonsense tasks, this approach still requires testing on open-ended reasoning tasks where the answers are less structured.

Reprompting

Reprompting method integrates the CoT paradigm with Gibbs sampling (a technique to generate samples from a probability distribution of two or more dimensions) techniques for prompt optimization [65]. This method was proposed to address the limitations in scalability and generalizability often seen with CoT techniques, which in some cases still rely on human experts to design prompts for tasks requiring multi-step reasoning. The reprompting method demonstrates superior performance compared to self-consistency decoding [33], Auto-CoT [64], and Automatic Prompt Optimization [75]. With improvements ranging from an average of 11 to 33 points on BBH tasks. Notably, it has shown potential for effective combinations; for example, using ChatGPT to generate initial sample guidance for InstructGPT improved performance by up to 71 points compared to using ChatGPT alone. In ObjectCount task the recorded performance is 99.6 as shown in Table 1

However, the experimental setup in the research involved up to 20,000 iterations, with costs for running the experiments on ChatGPT and text-DaVinci-003 ranging from $80 to $800, which could rise with additional iterations. Despite its promise, the method still needs more consistency in cross-modal generalization (LLM's ability to perform tasks on different types of data that differ from training data type). The study reported that CoT recipes developed using InstructGPT and later tested on ChatGPT resulted in an 18% performance drop.

C. Ontological-based Prompt Optimization

Ontological-based prompt optimization is a technique for optimizing prompts in LLMs by leveraging knowledge representations, such as ontologies, to enhance the quality and effectiveness of the prompts. Below is a review of research conducted in efforts to implement this approach.

Ontoprompt

OntoPrompt [76] is designed to transform structured knowledge, specifically ontologies, into textual prompts, leveraging these prompts to enhance few-shot learning performance. According to the findings, this method addresses the challenges of missing information, noise, and heterogeneity through span-sensitive knowledge (only adding helpful information to the prompts while ignoring irrelevant details) and collective training (a way to train both the added knowledge and the model together, ensuring they work well as a team and make better predictions). One of the critical strengths of OntoPrompt is its model-agnostic nature, allowing it to integrate seamlessly with any pre-trained language model, such as BERT or BART. This flexibility underscores its potential for use across various model architectures to tackle specific tasks within few-shot learning settings.

In terms of performance, OntoPrompt has demonstrated notable success, particularly with extraction tasks, where it achieved an F1 score (metric used to evaluate the performance of the model's classification ability by combining precision and recall into a single value) of 52.6 in an 8-shot learning scenario, outperforming Fine-tuning (24.8) and GDPNET (25.3). For event extraction, with only 1% of training data, OntoPrompt recorded an F1 score of 25.6, significantly higher than MQAEE (5.2) and TEXT2EVENT (3.4). Additionally, the knowledge graph completion task on the FB15K-237 mini dataset attained a Hit@10 (metric used to evaluate the performance of recommendation and ranking systems) of 0.111, surpassing models like KG-BERT (0.0451) and GRL (0.0300). However, this approach's limitation is its reliance on high-quality external knowledge, which can restrict its applicability in cases where external sources need more comprehensive ontologies.

OntoChatGPT

OntoChatGPT [77] is a meta-learning framework that integrates ontology-driven structured prompts with ChatGPT to enhance performance in domain-specific dialogue systems. This approach involves the creation of formal models for structuring knowledge and designing prompts, enabling ChatGPT to interpret, extract, and infer information based on predefined ontologies. By employing a meta-learning approach, this framework overcomes limitations in the training data and continuously improves the prompt generation process, adapting to new challenges and fine-tuning responses. The method has shown promising results in specific domains, such as rehabilitation medicine, where it achieved an accuracy of 0.7059, precision of 0.6534, recall of 0.9444, and an F1 score of 0.7724. In the confusion matrix, the model produced 17 true positives, seven true negatives, nine false positives, and one false negative. However, further testing is necessary in languages other than Ukrainian to demonstrate its applicability across various languages and to ensure that the syntax of a specific task does not influence the framework's performance. Additionally, similar to previous methods, the performance of this ontology-based approach depends heavily on the quality and comprehensiveness of the predefined ontologies.

D. Fine Tuning based Prompt Optimization

Fine-tuning-based prompt optimization represents an approach wherein a fine-tuned language model has been used to improve prompts. Fine-tuning-based prompt optimization modifies the model for better generation and refinement of prompts, as opposed to black-box prompt optimization, which makes no internal changes to the model. This approach fine-tunes the model to create better prompts, ultimately

leading to more contextually appropriate and accurate outputs.

P-Tuning

P-tuning [78] is specially designed to improve the performance and stability of LLMs by using trainable continuous prompt embeddings. It contrasts manually designed discrete prompts, which are mostly unstable and sensitive to minor changes. P-Tuning uses continuous embeddings (numerical vectors that represent words to capture their meanings and relationships) optimized by back-propagation so that the model can learn more robust patterns for natural language understanding tasks by combining continuous and discrete prompts. This approach offers several advantages, such as improved performance on a wide range of benchmarks and consistently outperforming traditional prompt-based methods like PET and manually crafted prompts, especially in tasks like SuperGLUE. These gains are evident in both fully-supervised and few-shot learning, enhancing the performance of models like BERT and GPT-2 across various tasks.

Important contribution of P-Tuning is the instability decoupling effect that discrete prompts have through continuous embeddings, which significantly reduces performance variance, as seen in the LAMA knowledge probing task. Moreover, its flexibility allows it to apply effectively to both unidirectional and bidirectional architectures, which increases its applicability to a wide range of NLP tasks. However, P-Tuning introduces more complexity and computational overhead since optimizing continuous prompt embeddings requires backpropagation. This may not be feasible in real-time or resource-constrained environments. Furthermore, this method is heavily dependent on the underlying quality of pre-trained models such as GPT and BERT, meaning the biases and weaknesses in these models persist despite the improved prompt stability.

Prompt Adaptation

Prompt Adaptation, according to the framework in [79], the latest proposal is intended to boost the efficiency of text-to-image model by the automatic optimization of user-generated prompts. The principal task of the model is to transform user input into prompts so the model can better interpret and use them when producing high-quality images. The first phase is the supervised fine-tuning stage, which is a learning process that involves adjusting a pre-trained language model like GPT using a small set of pre-specified prompts. This way, the end user's language feature will be familiar with the model's suggestion. Later on, the application of the reinforcement learning technique will further refine these prompts. Thereby, this step will increase the visual quality and the resemblance of the generated images to the user's idea. The goal of optimization is to have input as a starting point, but at the same time to achieve the highest artistic standard in the output. This approach adds resilience to the model for tackling more varieties of prompts and domains, thus enhancing its universality.

One of the key positive sides of this automation process is that the prompt optimization process becomes completely automated so the prompt engineering work at the program level is less needed as it is manifested in lower costs for the company. The method is made versatile, as it can be applied to various models and across different domains by the built-in support of reinforcement learning. The approach draws out-of-domain input cases for which it particularly excels in performance through the use of reinforcement learning as well as the criteria of pertinence and aesthetics in prompt design. This dissertation's elegance is based on its ability to be general throughout diverse patterns and input sources. In contrast, the others use fixed reinforcement procedures only.

Nonetheless, Prompt Adaptation entails certain limitations. It relies heavily on prompt snippets formulated by humans during the training of complex models like Lexica, densely packed with artistic theme vocabulary terms (i.e., using artist names), and thus, may limit the model's capability of producing realistic images and result in a narrower range of model output. The approach's real-time efficiency as well as its diversity in training data, crucially in the fine-tuning step, are also proven to be the keys to success in the methodology. If the data set content is contained the bias of some creative manner, for example, the target domains or the model may not be able to perform its tasks properly. Refinements in reinforcement learning are very demanding tasks, which can be particularly challenging when it comes to employing them in real-time settings. Finally, the disparity in performance in more difficult or subjective tasks is subsided by the fact that it transfers well between the domains. But, pre-specified reward functions for relevance and art mostly miss the subtlety of user interests, especially in more nuanced or subjective tasks.

IV. Conclusion

The explosion of LLMs all across the world has given exposure to the request to prompt engineering methods that will increase the task quality without the additional computational and financial overheads that usually come in the wake of fine-tuning or retraining. Techniques such as black-box optimization and the ontology-based prompt refinement are two examples of this process. The latter have emerged as successful options for optimization, thus the quality of the final LLM outputs is strengthened even as the original inward structure of the model is retained. These methods have been found to be effective in various applications, from general problem-solving to domain-specific tasks, at the same time optimizing LLM efficiency in the most cost-effective way.

Changes in the command given strategies like CoT and ToT provide particular strengths for enhancing multi-step reasoning and complex problem-solving. Nevertheless, there still are challenges of adaptability, scalability, and speed under the more and more complicated task. The future researches might work on the prompt transfer and generalization that promote the across models and tasks, especially in low-resource settings. Additionally, integrating ontologies and automated prompt generation methods, such as

PromptAgent and APE, presents significant opportunities for refining LLMs' precision and contextual awareness. These advancements will be critical for maximizing LLM performance while minimizing resource demands in increasingly complex and dynamic tasks.

ACKNOWLEDGMENT

I sincerely thank Telkom University for its support throughout this research.

FUNDING STATEMENT

I would like to clarify that no funding agencies have provided financial assistance for this work. I am profoundly thankful to the researchers and collaborators whose invaluable contributions and insights significantly enriched the quality of this study.

AUTHOR CONTRIBUTIONS

Asyafa Ditra Al Hauna: Conceptualization, Data Curation, Visualization, Methodology, Writing – Original Draft Preparation, Writing – Review & Editing;

Andi Prademon Yunus: Project Administration, Methodology, Writing – Review & Editing, Validation, Supervision;

Masanori Fukui: Supervision and Study Design;

Siti Khomsah: Supervision.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our research work follows The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org.

REFERENCES

[1] J. He, M. Rungta, D. Koleczek, A. Sekhon, F. X. Wang, and S. Hasan, “Does Prompt Formatting Have Any Impact on LLM Performance?,” *arXiv preprint arXiv:2411.10541*, 2024. DOI: http://arxiv.org/abs/2411.10541

[2] C. Tang, Z. Wang, and Y. Wu, “Large Language Models Might Not Care What You Are Saying: Prompt Format Beats Descriptions,” *arXiv preprint arXiv:* 2408.08780, 2024. DOI: http://arxiv.org/abs/2408.08780

[3] D. Sulimov, “Prompt-Efficient Fine-Tuning for GPT-like Deep Models to Reduce Hallucination and to Improve Reproducibility in Scientific Text Generation Using Stochastic Optimisation Techniques,” *arXiv preprint arXiv:* 2411.06445, 2024. DOI: http://arxiv.org/abs/2411.06445

[4] T. Alhindi, T. Chakrabarty, E. Musi and S. Muresan, “Multitask Instruction-based Prompting for Fallacy Recognition,” *Conference on Empirical Methods in Natural Language Processing*, vol. 2022-Dec, pp. 8172–8187, 2022. DOI: http://arxiv.org/abs/2301.09992

[5] J. Wei *et al.*, “Finetuned Language Models Are Zero-Shot Learners,” *Tenth International Conference on Learning Representations*, 2021. DOI: https://doi.org/10.48550/arXiv.2109.01652

[6] A. Kong *et al.*, “Better Zero-Shot Reasoning with Role-Play Prompting,” *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4099–4113, 2024. DOI: https://doi.org/10.48550/arXiv.2308.07702

[7] R. Wang, F. Mi, Y. Chen, B. Xue, H. Wang, Q. Zhu, K. Wong and R. Xu, "Role Prompting Guided Domain Adaptation with General Capability Preserve for Large Language Models," *arXiv*, 2024. DOI: https://doi.org/10.48550/arXiv.2403.02756

[8] W. Zhou and T.H. Ngo, “Using Pretrained Large Language Model with Prompt Engineering to Answer Biomedical Questions,” *Conference and Labs of the Evaluation Forum*, 2024. DOI: https://doi.org/10.48550/arXiv.2407.06779

[9] X. Chen, Y. Zhang, J. Deng, J.-Y. Jiang and W. Wang, “Gotta: Generative Few-shot Question Answering by Prompt-based Cloze Data Augmentation,” *Proceedings of the 2023 SIAM International Conference on Data Mining*, pp. 909-917, 2023. DOI: https://doi.org/10.48550/arXiv.2306.04101

[10] W. Zhong, Y. Gao, N. Ding, Y. Qin, Z. Liu, M. Zhou, J. Wang, J. Yin and N. Duan, "ProQA: Structural Prompt-based Pre-training for Unified Question Answering," *arXiv*, 2022. DOI: https://doi.org/10.48550/arXiv.2205.04040

[11] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, “Unifying Large Language Models and Knowledge Graphs: A Roadmap,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, pp. 3580-3599, 2023. DOI: https://doi.org/10.1109/TKDE.2024.3352100

[12] K. Vasisht, B. Ganesan, V. Kumar and V. Bhatnagar, “Infusing Knowledge into Large Language Models with Contextual Prompts,” *Proceedings of the 20th International Conference on Natural Language Processing*, pp. 657–662, 2024. DOI: https://doi.org/10.48550/arXiv.2403.01481

[13] S. Swamy, N. Tabari, C. Chen and R. Gangadharaiah, “Contextual Dynamic Prompting for Response Generation in Task-oriented Dialog Systems,” *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 17, pp. 3102–3111, 2023. DOI: https://doi.org/10.48550/arXiv.2301.13268

[14] T. Kojima, S. Shane Gu, M. Reid Google Research, Y. Matsuo and Y. Iwasawa, “Large Language Models are Zero-Shot Reasoners,” *Conference on Neural Information Processing Systems*, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2205.11916

[15] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language Models are Few-Shot Learners," *arXiv*, 2020. DOI: https://doi.org/10.48550/arXiv.2005.14165

[16] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A.M. Dai, T.S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov and N. Fiedel, "PaLM: Scaling Language Modeling with Pathways," *arXiv*, 2022. DOI: https://doi.org/10.48550/arXiv.2204.02311

[17] H.W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S.S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E.H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q.V. Le and J. Wei, "Scaling Instruction-Finetuned Language Models," *arXiv*, 2022. DOI: https://doi.org/10.48550/arXiv.2210.11416

[18] OpenAI *et al.*, “GPT-4 Technical Report,” *arXiv*, vol. abs/2303.08774, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2303.08774

[19] S. Utpala, C. For, S. Hooker and P.Y. Chen, “Locally Differentially Private Document Generation Using Zero Shot Prompting,” *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. DOI: http://dx.doi.org/10.18653/v1/2023.findings-emnlp.566

[20] S. Sivarajkumar, M. Kelley, A. Samolyk-Mazzanti, S. Visweswaran and Y. Wang, “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*, vol. 12, 2024. DOI: http://dx.doi.org/10.2196/55318

[21] S. Sivarajkumar and Y. Wang, “HealthPrompt: A Zero-shot Learning Paradigm for Clinical Natural Language Processing,” *AMIA Annual Symposium*, vol. 2022, pp. 972-981, 2022. DOI: http://dx.doi.org/10.48550/arXiv.2203.05061

[22] L. Kovriguina, R. Teucher, D. Radyush and D. Mouromtsev, “SPARQLGEN: One-Shot Prompt-based Approach for SPARQL Query Generation,” *AMIA Annual Symposium*, vol. 2022, pp. 972-981, 2023. DOI: http://dx.doi.org/10.1007/978-3-642-31600-5_49

[23] S.-Y. Yoon, “Short Answer Grading Using One-shot Prompting and Text Similarity Scoring Model,” *Arxiv*, vol. abs/2305.18638, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2305.18638

[24] A.M. Marasovi´c, I. Beltagy, D. Downey and M.E. Peters, “Few-Shot Self-Rationalization with Natural Language Prompts,” 2022 *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, vol. 2022, pp. 410-424, 2022. DOI: http://dx.doi.org/10.48550/arXiv.2111.08284

[25] H. Ma, C. Zhang, Y. Bian, L. Liu, Z. Zhang, P. Zhao, S. Zhang, H. Fu, Q. Hu and B. Wu, "Fairness-guided Few-shot Prompting for Large Language Models," *arXiv*, 2023. DOI: https://doi.org/10.48550/arXiv.2303.13217

[26] X. Yu, Y. Fang, Z. Liu and X. Zhang, “HGPROMPT: Bridging Homogeneous and Heterogeneous Graphs for Few-shot Prompt Learning,” in *Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, 2024. DOI: http://dx.doi.org/10.1609/aaai.v38i15.29596

[27] A.V.Y. Lee, C.L. Teo, and S.C. Tan, “Prompt Engineering for Knowledge Creation: Using Chain-of-Thought to Support Students’ Improvable Ideas,” *AI*, vol. 5, no. 3, pp. 1446–1461, 2024. DOI: http://dx.doi.org/10.3390/ai5030069

[28] K. Hebenstreit, R. Praas, L.P. Kiesewetter and M. Samwald, “An automatically discovered chain-of-thought prompt generalizes to novel models and datasets,” *Arxiv*, vol. abs/2305.02897, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2305.02897

[29] G. Feng, B. Zhang, Y. Gu, H. Ye, D. He and L. Wang, “Towards Revealing the Mystery behind Chain of Thought: A Theoretical Perspective,” *37th Annual Conference on Neural Information Processing Systems*, vol. 2305.15408, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2305.15408

[30] J. Long, “Large Language Model Guided Tree-of-Thought,” *Arxiv preprint*, vol. 2305.08291, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2305.08291

[31] S. Yao, D. Yu, J. Zhao, I. Shafran, T.L. Griffiths, Y. Cao and K. Narasimhan, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," *arXiv*, 2023. DOI: https://doi.org/10.48550/arXiv.2305.10601

[32] Z. Bi, D. Hajialigol, Z. Sun, J. Hao and X. Wang, “STOC-TOT: Stochastic Tree-of-Thought with Constrained Decoding for Complex Reasoning in Multi-Hop Question Answering,” *Arxiv* , vol. abs/2407.03687, 2024. DOI: http://dx.doi.org/10.48550/arXiv.2407.03687

[33] X. Wang *et al.*, “Self-Consistency Improves Chain of Thought Reasoning in Language Models,” *Arxiv*, vol. abs/2203.11171, 2023. DOI: http://arxiv.org/abs/2203.11171

[34] G. Wan, Y. Wu, J. Chen and S. Li, “Dynamic Self-Consistency: Leveraging Reasoning Paths for Efficient LLM Sampling,” *Arxiv*, vol. abs/ 2408.17017, 2024. DOI: http://dx.doi.org/10.48550/arXiv.2408.17017

[35] Y. Li, P. Yuan, S. Feng, B. Pan, X. Wang, B. Sun, H. Wang and K. Li, "Escape Sky-high Cost: Early-stopping Self-Consistency for Multi-step Reasoning," *arXiv*, 2024. DOI: https://doi.org/10.48550/arXiv.2401.10480

[36] P. Aggarwal, A. Madaan, Y. Yang and Mausam, “Let’s Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs,” *Conference on Empirical Methods in Natural Language Processing*, vol. 2, 2023. DOI: https://doi.org/10.48550/arXiv.2305.11860

[37] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R.L. Bras, Y. Choi and H. Hajishirzi, "Generated Knowledge Prompting for Commonsense Reasoning," *arXiv*, 2021. DOI: https://doi.org/10.48550/arXiv.2110.08387

[38] R. Xu, H. Cui, Y. Yu, X. Kan, W. Shi, Y. Zhuang, M.D. Wang, W. Jin, J. Ho and C. Yang, "Knowledge-Infused Prompting: Assessing and Advancing Clinical Text Data Generation with Large Language Models," *Findings of the Association for Computational Linguistics ACL 2024*, pp. 15496-15523, 2024. DOI: https://doi.org/10.18653/v1/2024.findings-acl.916

[39] Y. Meng, J. Huang, Y. Zhang and J. Han, “Generating Training Data with Language Models: Towards Zero-Shot Language Understanding,” *Conference on Neural Information Processing Systems*, vol. 35, pp. 462-477, 2022. DOI: http://arxiv.org/abs/2202.04538

[40] Y. Meng, M. Michalski, J. Huang, Y. Zhang, T. Abdelzaher and J. Han, “Tuning Language Models as Training Data Generators for Augmentation-Enhanced Few-Shot Learning,” *Proceedings of the 40th International Conference on Machine Learning*, 2022. DOI: http://arxiv.org/abs/2211.03044

[41] K.M. Yoo, D. Park, J. Kang, S.-W. Lee and W. Park, “GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation,” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, vol. EMNLP 2021, pp. 2225–2239, 2021. DOI: http://arxiv.org/abs/2104.08826

[42] H. Zeng, B. Wei, J. Liu and W. Fu, “Synthesize, Prompt and Transfer: Zero-shot Conversational Question Generation with Pre-trained Language Model,” *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics,* vol. 1, pp. 8989–9010, 2023. DOI: https://doi.org/10.18653/v1/2023.acl-long.500

[43] B.M. Lake and M. Baroni, “Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks,” *Proceedings of the 35th International Conference on Machine Learning*, 2017. DOI: http://arxiv.org/abs/1711.00350

[44] J. Ouyang, S. Liang, S. Chen, S. Li, Y. Zhou and Q. Liwen, "Design and Realization of Data Application Architecture Oriented to the Requirements of Distribution Network," *2020 IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 2354-2359, 2020. DOI: https://doi.org/10.1109/iSPEC50848.2020.9351123

[45] D. Zhou *et al.*, “Least-to-Most Prompting Enables Complex Reasoning in Large Language Models,” *The Eleventh International Conference on Learning Representations,* 2023. DOI: http://arxiv.org/abs/2205.10625

[46] Y. Yao, Z. Li and H. Zhao, “Beyond Chain-of-Thought, Effective Graph-of-Thought Reasoning in Language Models,” *Arxiv*, vol. abs/2305.16582, 2023. DOI: http://arxiv.org/abs/2305.16582

[47] B. Lei, pei-H. Lin, C. Liao and C. Ding, “Boosting Logical Reasoning in Large Language Models through a New Framework: The Graph of Thought,” *Arxiv*, vol. abs/2308.08614, 2023. DOI: http://dx.doi.org/10.48550/arXiv.2308.08614

[48] L. Zheng *et al.*, “Reverse Multi-Choice Dialogue Commonsense Inference with Graph-of-Thought,” *Thirty-Seventh AAAI Conference on Artificial Intelligence,* 2023. DOI: http://dx.doi.org/10.1609/aaai.v38i17.29942

[49] O. Ram *et al.*, “In-Context Retrieval-Augmented Language Models”, *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1316-1331, 2023. DOI: http://dx.doi.org/10.1162/tacl_a_00605

[50] S. Lin, J.H. Openai and O. Evans, “TruthfulQA: Measuring How Models Mimic Human Falsehoods,” *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 3214-3252, 2022. DOI: http://dx.doi.org/10.48550/arXiv.2109.07958

[51] Z. Guo, S. Cheng, Y. Wang, P. Li and Y. Liu, "Prompt-Guided Retrieval Augmentation for Non-Knowledge-Intensive Tasks," *61st Annual Meeting of the Association for Computational Linguistics*, vol. ACL 2023, pp. 10896–10912, 2023.
DOI: http://dx.doi.org/10.48550/arXiv.2305.17653

[52] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics,* vol. EACL 2021, pp. 874–880, 2021.
DOI: http://dx.doi.org/10.48550/arXiv.2007.01282

[53] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv*, 2020.
DOI: https://doi.org/10.48550/arXiv.2005.11401

[54] R. Socher *et al.*, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.
DOI: https://aclanthology.org/D13-1170/

[55] A. Warstadt, A. Singh and S.R. Bowman, "Neural Network Acceptability Judgments," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, 2018.
DOI: https://doi.org/10.48550/arXiv.1805.12471

[56] H. Tan *et al.*, "Prompt-based Code Completion via Multi-Retrieval Augmented Generation," *Arxiv*, vol. abs/2405.07530, 2024.
DOI: https://doi.org/10.48550/arXiv.2405.07530

[57] J. Cheng *et al.*, "Black-Box Prompt Optimization: Aligning Large Language Models without Model Training," *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 3201–3219, 2023.
DOI: https://doi.org/10.18653/v1/2024.acl-long.176

[58] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H.W. Chung, A. Chowdhery, Q. Le, E. Chi, D. Zhou and J. Wei, "Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them," *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.
DOI: https://doi.org/10.18653/v1/2023.findings-acl.824

[59] K. Cobbe *et al.*, "Training Verifiers to Solve Math Word Problems," *Arxiv*, vol. abs/2110.14168, 2021.
DOI: http://arxiv.org/abs/2110.14168

[60] Y. Zhou *et al.*, "Large Language Models Are Human-Level Prompt Engineers," *The Eleventh International Conference on Learning Representations*, 2023.
DOI: https://doi.org/10.48550/arXiv.2211.01910

[61] Y. Zhou, A.I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan and J. Ba, "Large Language Models Are Human-Level Prompt Engineers," *arXiv*, 2022.
DOI: https://doi.org/10.48550/arXiv.2211.01910

[62] C. Yang, X. Wang, Y. Lu, H. Liu, Q.V. Le, D. Zhou and X. Chen, "Large Language Models as Optimizers," *arXiv*, 2023.
DOI: https://doi.org/10.48550/arXiv.2309.03409

[63] H. Sun, X. Li, Y. Xu, Y. Homma, Q. Cao, M. Wu, J. Jiao and D. Charles, "AutoHint: Automatic Prompt Optimization with Hint Generation," *arXiv*, 2023.
DOI: https://doi.org/10.48550/arXiv.2307.07415

[64] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic Chain of Thought Prompting in Large Language Models," *The Eleventh International Conference on Learning Representations*, 2023.
DOI: https://doi.org/10.48550/arXiv.2210.03493

[65] W. Xu, A. Banburski-Fahey and N. Jojic, "Reprompting: Automated Chain-of-Thought Prompt Inference Through Gibbs Sampling," *41st International Conference on Machine Learning*, vol. 235, pp. 54852-54865, 2023.
DOI: https://doi.org/10.48550/arXiv.2305.09993

[66] Y. Zhou *et al.*, "Large Language Models Are Human-Level Prompt Engineers," *The Eleventh International Conference on Learning Representations*, 2023.
DOI: http://arxiv.org/abs/2211.01910

[67] S. Roy and D. Roth, "Solving General Arithmetic Word Problems," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752, 2016.
DOI: https://doi.org/10.18653/v1/D15-1202

[68] R. Pryzant, D. Iter, J. Li, Y.T. Lee, C. Zhu, and M. Zeng, "Automatic Prompt Optimization with 'Gradient Descent' and Beam Search," *Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968, 2023.
DOI: https://doi.org/10.48550/arXiv.2305.03495

[69] Y. Ishimizu, J. Li, J. Xu, J. Cai, H. Iba and K. Tei, "Automatic Adaptation Rule Optimization via Large Language Models," *2024 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, pp. 180-181, 2024.
DOI: https://doi.org/10.1109/ACSOS-C63493.2024.00057

[70] R. Anil *et al.*, "PaLM 2 Technical Report," *Arxiv*, 2023, vol. abs/2305.10403, 2023.
DOI: http://arxiv.org/abs/2305.10403

[71] OpenAI *et al.*, "GPT-4 Technical Report," *Arxiv*, vol. abs/2303.08774, 2023.
DOI: http://arxiv.org/abs/2303.08774

[72] S. Lin and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," *Operations Research*, vol. 21, pp. 498-516, 1973.
DOI: https://www.jstor.org/stable/169020

[73] N. P, S. Eliyas, S. K. M and B. Balusamy, "Individualized Mastery Quest: Crafting Customized Question Papers and Dynamic Hint Generation for Personalized Learning Journeys Using Cutting-Edge Rank-Based Algorithm," *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, pp. 1-6, 2024.
DOI: https://doi.org/10.1109/ICEECT61758.2024.10739213

[74] T. Kojima, S.S. Gu, M. Reid, Y. Matsuo and Y. Iwasawa, "Large Language Models are Zero-Shot Reasoners," *arXiv*, 2022.
DOI: https://doi.org/10.48550/arXiv.2205.11916

[75] R. Pryzant, D. Iter, J. Li, Y.T. Lee, C. Zhu, and M. Zeng, "Automatic Prompt Optimization with 'Gradient Descent' and Beam Search," *Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968, 2023.
DOI: http://dx.doi.org/10.48550/arXiv.2201.11903

[76] H. Ye *et al.*, "Ontology-enhanced Prompt-tuning for Few-shot Learning," in *WWW 2022 - Proceedings of the ACM Web Conference 2022*, Association for Computing Machinery, Inc, pp. 778–787, 2022.
DOI: http://dx.doi.org/10.48550/arXiv.2201.11332

[77] O. Palagin, V. Kaverinsky, A. Litvin and K. Malakhov, "OntoChatGPT Information System: Ontology-Driven Structured Prompts for ChatGPT Meta-Learning," *International Journal of Computing*, vol. 22, no. 2, pp. 170–183, 2023.
DOI: http://dx.doi.org/10.48550/arXiv.2307.05082

[78] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang and J. Tang, "GPT understands, too," *AI Open*, vol. 5, pp. 208-215, 2024.
DOI: https://doi.org/10.1016/j.aiopen.2023.08.012

[79] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing Prompts for Text-to-Image Generation," *Proceedings of 36th International Conference on Neural Information Processing Systems*, 2022.
DOI: https://doi.org/10.48550/arXiv.2212.09611

[80] M. Too, S. H. Lau, and C. K. Tan, "Validity and reliability of a conceptual framework on enhancing learning for students via Kinect: A pilot test," *International Journal on Robotics, Automation and Sciences*, vol. 6, no. 1, pp. 59–63, 2024.
DOI: https://doi.org/10.33093/ijoras.2024.6.1.8

[81] T.-E. Tai, S.-C. Haw, W.-E. Kong, and K.-W. Ng, "Performance evaluation of machine learning techniques on resolution time prediction in helpdesk support system," *International Journal on Robotics, Automation and Sciences*, vol. 6, no. 2, pp. 59–68, 2024.
DOI: https://doi.org/10.33093/ijoras.2024.6.2.9 0.33093/ijoras.2024.6.2.9

[82] M. Too and R. Chang, "A fundamental study of an alternative learning framework utilizing natural user interface (NUI) for physically disabled students," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 1, pp. 1–5, 2023.
DOI: https://doi.org/10.33093/ijoras.2023.5.1.1