

Optimizing LLMs Prompts via Semantically-Guided Evolutionary Algorithms using SLMs

Pratyush Kumar Jha

Artificial Intelligence and Machine Learning
Manipal University Jaipur
Pratyushjha649@gmail.com

Abstract: Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding, reasoning, generation, and task automation; however, their performance is highly dependent on the quality and structure of the prompts provided to them. Manual prompt engineering is often time-consuming, subjective, and difficult to scale across different tasks and models. This study proposes a Semantically-Guided Evolutionary Prompt Optimization framework using Small Language Models (SLMs) to automatically generate and refine effective prompts for LLM-based applications. The proposed approach integrates semantic similarity analysis with evolutionary optimization operations, including prompt generation, selection, mutation, crossover, and fitness evaluation. SLMs are employed as lightweight semantic evaluators to assess the relevance, clarity, consistency, and task alignment of candidate prompts without requiring repeated dependence on computationally expensive LLM evaluations. Candidate prompts are iteratively evolved based on their semantic fitness and task-specific performance, enabling the framework to search efficiently through a large prompt space. The methodology aims to preserve the original task intent while improving prompt quality and reducing unnecessary computational and inference costs. The optimized prompts are evaluated against manually designed and conventional prompt-optimization approaches using task performance, semantic similarity, robustness, computational efficiency, and response quality. The proposed framework provides a scalable strategy for automated prompt engineering and demonstrates the potential of combining LLMs, SLMs, semantic guidance, and evolutionary algorithms for efficient prompt optimization. The approach can be applied to question answering, text classification, summarization, reasoning, information extraction, and other LLM-based natural language processing tasks.

Keywords: Large Language Models, Small Language Models, Prompt Optimization, Prompt Engineering, Evolutionary Algorithms, Semantic Similarity, Natural Language Processing, Generative AI.

I. INTRODUCTION

The rapid advancement of Artificial Intelligence (AI) has led to the development of Large Language Models (LLMs) capable of performing a wide range of natural language processing tasks, including question answering, text generation, summarization, translation, information

extraction, reasoning, and content analysis. Models based on Transformer architectures have demonstrated strong capabilities by learning complex linguistic patterns from massive datasets. However, the effectiveness of an LLM is not determined solely by its underlying architecture or number of parameters. The way a task is presented to the model through a prompt has a significant influence on the quality, relevance, consistency, and accuracy of the generated response. Consequently, prompt engineering has become an important component of modern LLM-based application development [1].

Traditional prompt engineering generally depends on human experts who manually design instructions, experiment with different wording, and select prompts that produce desirable outputs. Although this approach can be effective, it is time-consuming, subjective, and difficult to scale when multiple tasks, datasets, and LLMs are involved. A small change in wording, ordering of instructions, examples, or constraints can substantially affect model behavior. Furthermore, the enormous number of possible prompt formulations creates a large search space that is difficult to explore manually. These limitations have motivated the development of automated prompt optimization techniques capable of discovering effective prompts with minimal human intervention [2, 3].

Evolutionary algorithms provide a promising solution to this problem because they are designed to search large and complex solution spaces through iterative optimization. Inspired by natural evolution, evolutionary algorithms maintain a population of candidate solutions and improve them through operations such as selection, mutation, crossover, and replacement. When applied to prompt optimization, each candidate prompt can be treated as an individual solution, while its performance on a target task can be used as a fitness value. The evolutionary process can therefore generate new prompt variants and progressively retain formulations that produce better LLM responses [4].

However, conventional evolutionary prompt optimization can become computationally expensive when every candidate prompt must be evaluated directly using a large and resource-intensive LLM. Repeated inference increases computational cost, latency, and energy consumption, particularly when the population size and number of evolutionary generations are large. This motivates the use of Small Language Models (SLMs) as lightweight semantic-guidance mechanisms. SLMs contain fewer parameters and generally require fewer computational resources than large-scale models while still providing useful capabilities for language understanding, semantic comparison, classification, and evaluation. Their use as semantic evaluators can reduce the number of expensive evaluations required during prompt optimization [5, 6].

The proposed approach, “Optimizing LLMs Prompts via Semantically-Guided Evolutionary Algorithms using SLMs,” combines semantic guidance, evolutionary optimization, and small language models into a unified prompt optimization framework. In this framework, an initial population of candidate prompts is generated for a particular task. The semantic characteristics of these prompts are evaluated using an SLM, which helps determine their relevance, similarity to the intended task, clarity, and preservation of task semantics. Based on these evaluations, promising prompts are selected and modified through evolutionary

operations. Mutation can introduce new wording or instructions, while crossover can combine useful components from multiple high-quality prompts. The resulting candidates are iteratively evaluated and evolved toward improved prompt formulations [7].

A key objective of semantic guidance is to ensure that optimization does not improve a prompt merely by changing its surface wording while unintentionally changing the original task intent. By incorporating semantic similarity into the fitness evaluation, the proposed method attempts to maintain semantic consistency while improving task-specific performance. This provides an important advantage over purely performance-driven optimization, where an evolutionary process may generate prompts that achieve high scores for a particular evaluation setting but are poorly aligned with the original task [8, 9].

The effectiveness of the optimized prompts can be assessed using several criteria, including task accuracy, response quality, semantic similarity, robustness, consistency, and computational cost. The optimized prompts can also be compared with manually engineered prompts and conventional prompt-optimization strategies to determine whether semantic evolutionary optimization provides measurable improvements. Such evaluation is important because an optimized prompt should not only produce better responses but should also remain effective across different inputs and potentially different model configurations.

Overall, the integration of SLMs with evolutionary algorithms provides a promising direction for reducing the computational burden of automated prompt engineering while maintaining semantic quality. The proposed framework can potentially be applied to a variety of LLM-based tasks, including text classification, question answering, summarization, sentiment analysis, reasoning, information extraction, and conversational systems. By automatically exploring and refining prompt formulations through semantically guided evolutionary search, the proposed work aims to provide a scalable, efficient, and intelligent approach to prompt optimization for modern LLM applications.

II. LITERATURE REVIEW

Vaswani et al. [1] introduced the Transformer architecture through their influential work, *“Attention Is All You Need.”* The study proposed replacing recurrent and convolutional structures used in sequence modeling with a self-attention-based architecture. The Transformer employs multi-head self-attention, which enables the model to capture relationships between different words or tokens regardless of their positions in a sequence. This mechanism significantly improves parallel processing and allows the model to learn long-range dependencies more effectively. The proposed architecture consists mainly of encoder and decoder components, with positional encoding used to preserve information about token order. The study demonstrated superior performance in machine translation while achieving improved training efficiency compared with recurrent architectures. This work established the fundamental architecture underlying many subsequent language models and provides the architectural foundation for modern LLMs used in automated prompt optimization.

Devlin et al. [2] introduced BERT (Bidirectional Encoder Representations from Transformers), a Transformer-based language representation model designed to learn contextual representations of language. Unlike conventional models that process text primarily in a single direction, BERT uses bidirectional Transformer representations, allowing it to consider both left and right contextual information when learning the meaning of a word. The model was pre-trained using large-scale unlabeled text and subsequently fine-tuned for specific downstream tasks such as question answering, sentiment analysis, and natural language inference. The study demonstrated that a pre-trained language model could achieve substantial improvements across multiple NLP benchmarks with task-specific fine-tuning. BERT established the importance of pre-training and transfer learning in NLP, providing an important foundation for later developments in large language models and prompt-based learning. Its contextual representation capability is also relevant to semantic-guided prompt optimization, where understanding the semantic relationship between instructions and task requirements is essential.

Brown et al. [3] presented GPT-3, a large autoregressive language model containing 175 billion parameters, and demonstrated that sufficiently large language models can perform various NLP tasks with minimal task-specific training. The study introduced the concept of few-shot learning, in which examples and task instructions are provided directly through the model's input prompt rather than modifying the model parameters. GPT-3 demonstrated strong performance on tasks including translation, question answering, cloze tasks, and several forms of language generation. The findings showed that the formulation of the input context can significantly influence model performance, thereby establishing prompting as an important mechanism for interacting with large language models. This work is particularly relevant to the proposed research because it provides the motivation for automatically optimizing prompts rather than relying entirely on manually designed instructions. The study also demonstrated the potential of scaling language models to generalize across multiple tasks through appropriately constructed prompts.

Hu et al. [4] proposed Low-Rank Adaptation (LoRA) as an efficient method for adapting large language models to specific downstream tasks. Instead of updating all parameters of a pre-trained model, LoRA introduces trainable low-rank matrices into selected layers while keeping the original model parameters frozen. This significantly reduces the number of trainable parameters and associated computational and memory requirements. The authors demonstrated that LoRA can achieve performance comparable to or better than conventional full fine-tuning for several language-model adaptation tasks while requiring substantially fewer trainable parameters. Although LoRA primarily addresses parameter-efficient fine-tuning rather than prompt optimization, it is relevant to the proposed research because it highlights the importance of computationally efficient adaptation of large language models. The principle of reducing computational requirements is complementary to the proposed use of Small Language Models (SLMs) for semantic evaluation and evolutionary prompt optimization.

Ouyang et al. [5] proposed Reinforcement Learning from Human Feedback (RLHF) for improving the ability of language models to follow user instructions. The study developed a multi-stage training framework involving supervised fine-tuning, reward-model development using human preference data, and reinforcement learning to optimize model behavior. The resulting instruction-following models demonstrated substantially improved performance in generating responses that were useful, relevant, and aligned with human preferences. This research established the importance of instruction quality and preference-based evaluation in developing effective language-model systems. For prompt optimization, the study is particularly relevant because it demonstrates that the quality of an LLM response can be evaluated according to task-specific and human-centered criteria rather than relying solely on language-model likelihood. These ideas provide a basis for designing fitness functions in automated prompt optimization, where candidate prompts can be evaluated according to response quality, task relevance, semantic consistency, and other performance criteria.

Wei et al. [6] investigated Chain-of-Thought (CoT) prompting as a method for improving the reasoning capabilities of large language models. The authors demonstrated that providing intermediate reasoning steps within prompts can significantly improve the ability of sufficiently large language models to solve complex reasoning tasks. Instead of directly requesting only the final answer, CoT prompting encourages the model to generate a sequence of intermediate steps before producing the final response. The study evaluated this approach on several arithmetic, commonsense, and symbolic reasoning benchmarks and showed that appropriately designed prompts can substantially improve model performance. This work established that prompt structure and formulation have a direct impact on LLM performance, providing important motivation for automated prompt optimization. For the proposed semantically guided evolutionary framework, the study is particularly relevant because reasoning-oriented prompts can be treated as candidate solutions and systematically modified to improve task performance while preserving the intended semantics.

Zhou et al. [7] introduced Automatic Prompt Engineer (APE), demonstrating that large language models can themselves be used to generate and select effective instructions for other language-model tasks. The approach treats prompt generation as a form of search in which candidate instructions are generated, evaluated, and ranked according to their performance on a target task. This reduces dependence on manually constructed prompts and demonstrates the potential of automated prompt engineering. The authors showed that automatically generated instructions can achieve competitive or superior performance compared with human-designed prompts across various tasks. This research is highly relevant to the proposed work because it establishes the feasibility of automated prompt generation and evaluation. However, relying heavily on large models for candidate generation and evaluation can introduce computational costs, motivating the proposed use of smaller language models for semantic guidance within an evolutionary optimization process.

Guo et al. [8] proposed EvoPrompt, an approach that combines large language models with evolutionary algorithms for automated prompt optimization. The study treats prompts as individuals within an evolutionary population and employs evolutionary operations to

generate improved prompt candidates. Candidate prompts are evaluated according to their performance on the target task, after which higher-performing prompts are selected for subsequent generations. By combining the language-generation capabilities of LLMs with the global search capability of evolutionary algorithms, the method can explore a broad prompt space without requiring manual engineering of every instruction. Experimental results demonstrated that evolutionary prompt optimization can provide competitive performance across multiple NLP tasks. This work is directly related to the proposed methodology because it establishes the effectiveness of evolutionary search for prompt optimization. The proposed research extends this concept by incorporating semantic guidance from Small Language Models (SLMs) to improve the selection and evolution of prompts while potentially reducing the computational cost of repeated LLM-based evaluations.

Pryzant et al. [9] proposed an automated prompt optimization method based on “gradient descent” and beam search. The approach uses textual feedback to identify weaknesses in an existing instruction and iteratively generates improved prompt variants. Instead of calculating conventional numerical gradients, the method uses language-model-generated feedback to guide changes to the prompt. Beam search is then employed to maintain multiple promising candidates and explore alternative prompt formulations. The study demonstrated that automatically optimized instructions can improve the performance of language models on different NLP tasks. The work is important because it shows that prompt optimization can be formulated as an iterative search problem rather than relying exclusively on manual prompt engineering. However, the approach follows a directed search strategy, whereas evolutionary methods can provide broader population-based exploration. This distinction motivates the investigation of evolutionary optimization combined with semantic guidance for more diverse and robust prompt generation.

Fernando et al. [10] proposed Promptbreeder, a self-referential evolutionary framework for automatically improving prompts. The method uses evolutionary processes to evolve both task prompts and the instructions responsible for generating improved prompts. Candidate prompts are evaluated according to their effectiveness on a target task, and successful prompt variants are retained and further evolved. A key contribution of Promptbreeder is its ability to perform self-improvement through prompt evolution, reducing the dependence on manually specified optimization strategies. The study demonstrated that evolutionary prompt optimization can discover effective instructions that may not be obvious to human prompt engineers. This work provides strong theoretical and methodological support for the proposed research. However, conventional evolutionary prompt optimization can require repeated evaluations using computationally expensive language models. Therefore, the proposed semantically guided evolutionary algorithm using SLMs aims to extend this direction by employing lightweight language models to evaluate semantic relevance and preserve task intent during the evolutionary process.

Problem Formulation: Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding, reasoning, content generation, and domain-specific problem solving. However, the quality of their responses is highly dependent on the

design of the input prompt. Manually designing an effective prompt is generally time-consuming, subjective, and highly dependent on the user's expertise. Furthermore, a prompt that performs well for one task or dataset may not provide satisfactory performance for another task. Therefore, automated prompt optimization is required to systematically discover prompts that improve the performance, reliability, and consistency of LLM-based systems. Existing prompt optimization approaches commonly employ random search, heuristic modification, reinforcement learning, or evolutionary strategies to generate and evaluate candidate prompts. Although these approaches can improve prompt quality, they may require a large number of expensive evaluations using the target LLM. Since modern LLMs can be computationally expensive and may involve API or inference costs, directly using a large model for every candidate prompt during evolutionary optimization creates a significant computational and financial burden. In addition, purely mutation-based or randomly generated prompts may lack semantic coherence, resulting in ineffective or redundant candidate solutions.

III. LLMS PROMPTS

Large Language Model (LLM) prompts are natural-language instructions, questions, examples, or contextual information provided to an LLM to guide its behavior and generate an appropriate response. A prompt acts as an interface between the user and the language model by specifying the desired task, output format, constraints, and contextual information. With the emergence of advanced Transformer-based models, prompting has become an important approach for performing natural language processing tasks without modifying the underlying model parameters. The effectiveness of an LLM is strongly influenced by how clearly and appropriately the prompt is formulated.

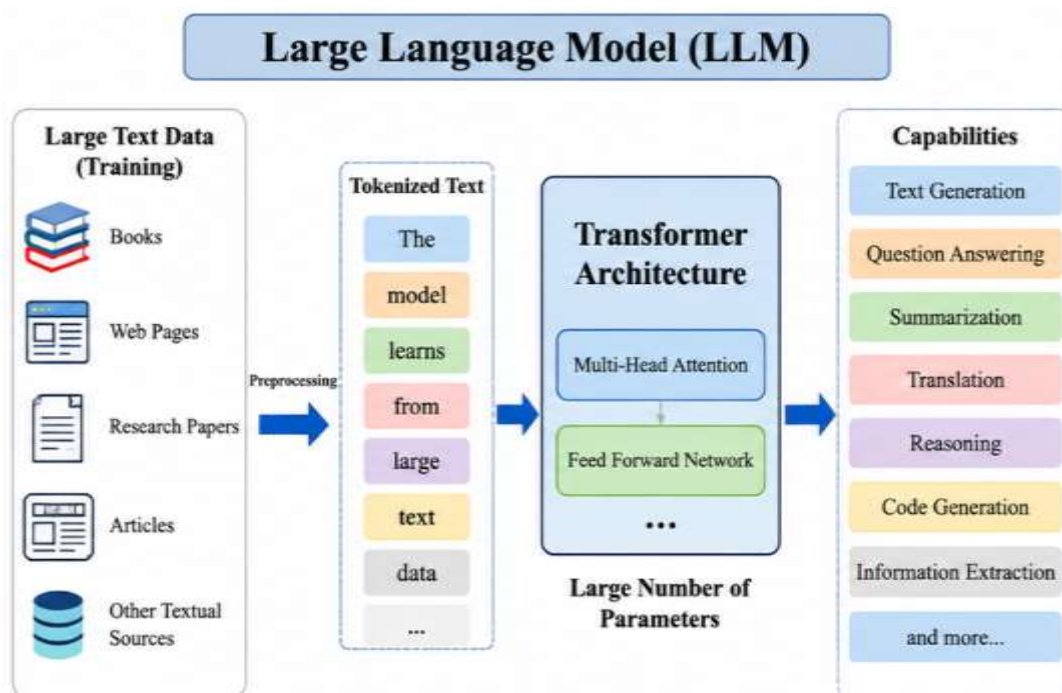


Figure 1: Large Language Model

A typical LLM prompt may contain several components, including task instructions, contextual information, input data, examples, constraints, and expected output format. For example, a simple prompt may ask a model to classify a review as positive or negative, whereas a more advanced prompt can provide several examples demonstrating the expected reasoning and output. The general structure can be represented as:

Prompt = Instruction + Context + Input + Constraints + Expected Output

The quality of these components can significantly affect the accuracy, relevance, consistency, and reasoning capability of the generated response. An ambiguous or poorly structured prompt may produce irrelevant or inconsistent results, while a well-designed prompt can guide the LLM toward the desired solution.

A. Types of LLM Prompts

Several prompting strategies have been developed to improve LLM performance. Zero-shot prompting provides only a task instruction without examples. One-shot and few-shot prompting provide one or multiple examples to demonstrate the expected behavior. Chain-of-Thought (CoT) prompting encourages the model to reason through intermediate steps for complex tasks. Other approaches include role-based prompting, instruction prompting, self-consistency prompting, and retrieval-augmented prompting. The selection of an appropriate strategy depends on the complexity and characteristics of the target task.

B. Importance of Prompt Optimization

Manual prompt engineering requires researchers or users to repeatedly modify wording, structure, examples, and constraints until satisfactory performance is achieved. This process can be time-consuming and may not produce consistent results across different models or datasets. Furthermore, the prompt search space can be extremely large because numerous combinations of words, instructions, examples, and formatting choices are possible. Therefore, automatic prompt optimization has become an important research direction.

In automatic prompt optimization, a set of candidate prompts is generated and evaluated according to a task-specific objective. The optimization process attempts to identify a prompt that maximizes model performance while preserving the original task requirements. Evolutionary algorithms are particularly suitable for this purpose because they can explore a large search space through selection, mutation, crossover, and fitness evaluation. In the proposed work, evolutionary optimization is combined with semantic guidance to generate increasingly effective prompts.

C. Semantic Guidance Using SLMs

A major challenge in prompt optimization is ensuring that modifications to a prompt do not change its original meaning. A prompt may obtain a higher task score while becoming semantically different from the intended instruction. To address this limitation, the proposed methodology incorporates Small Language Models (SLMs) as semantic evaluators. An SLM can compare the original and evolved prompts and estimate their semantic similarity, relevance, clarity, and task alignment. This semantic information is incorporated into the fitness evaluation process so that candidate prompts are optimized not only for performance but also for preservation of the intended meaning.

IV. SEMANTICALLY-GUIDED EVOLUTIONARY ALGORITHMS

Semantically-Guided Evolutionary Algorithms (SGEAs) combine the global search capability of evolutionary algorithms with semantic information to guide the generation and selection of candidate solutions. In the context of LLM prompt optimization, each prompt can be considered a candidate solution, while semantic information is used to determine whether an

evolved prompt retains the meaning and intent of the original instruction. Conventional evolutionary algorithms generally evaluate candidates according to a numerical fitness value, but they may generate solutions that achieve good task performance while deviating from the intended meaning. Semantic guidance addresses this limitation by incorporating measures such as semantic similarity, task relevance, contextual consistency, and linguistic quality into the optimization process.

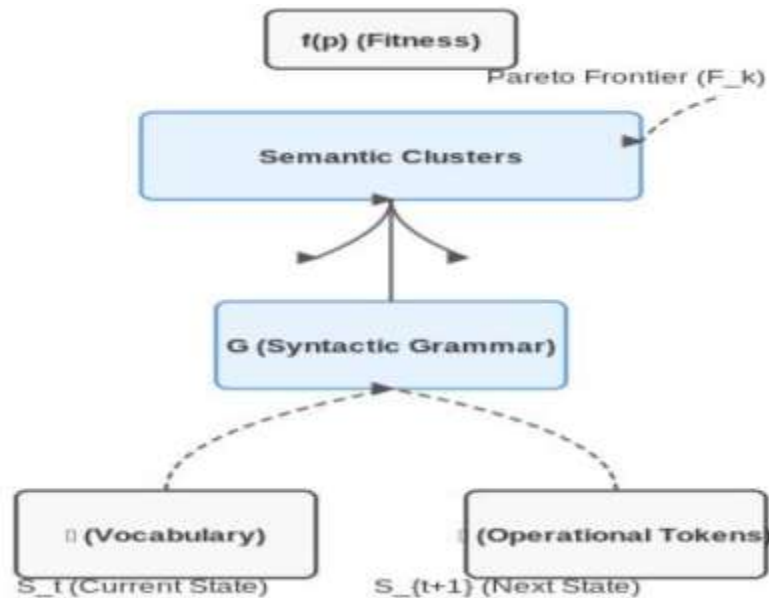


Figure 2: Semantically-Guided Evolutionary Algorithms

The evolutionary process consists primarily of selection, crossover, mutation, and fitness evaluation. During selection, prompts with higher fitness values are retained as parent candidates for the next generation. Crossover combines useful portions of two or more high-quality prompts to produce new candidate instructions. Mutation introduces controlled modifications, such as replacing words, modifying sentence structure, adding constraints, or changing examples.



Figure 3: Step

After generating new candidates, the SLM evaluates their semantic quality and the LLM evaluates their task-specific performance. This iterative process allows the population to gradually move toward prompts that provide better results while maintaining the original task intent.

A major advantage of semantic guidance is that it prevents uncontrolled evolution of prompts. For example, suppose the original prompt asks an LLM to “summarize a technical document in five concise points.” An evolutionary algorithm may modify the prompt to improve the evaluation score, but the resulting instruction could unintentionally ask for a detailed explanation instead of a summary. An SLM-based semantic evaluator can identify this deviation and assign a lower semantic-consistency score. Consequently, the evolutionary process favors candidates that improve task performance without significantly changing the original objective.

The fitness function can therefore combine task performance and semantic consistency. A general formulation is:

$$F(P) = \alpha T(P) + \beta S(P) + \gamma R(P) - \delta C(P)$$

where $F(P)$ is the overall fitness of prompt P , $T(P)$ represents task performance obtained from the LLM, $S(P)$ represents semantic similarity evaluated by the SLM, $R(P)$ represents prompt relevance or robustness, and $C(P)$ represents computational or prompt complexity. The coefficients α , β , γ , and δ control the contribution of each component. By combining these criteria, the optimization process can search for prompts that are both effective and semantically faithful.

V. SMALL LANGUAGE MODELS (SLMs)

Small Language Models (SLMs) are lightweight language models designed to perform natural language understanding and generation tasks with significantly fewer parameters and computational requirements than Large Language Models (LLMs). While LLMs are generally trained using extremely large datasets and contain billions or even hundreds of billions of parameters, SLMs typically use smaller architectures and optimized training strategies to provide efficient inference. Examples of SLMs include compact Transformer-based models and models such as Phi, TinyLlama, DistilBERT, and other parameter-efficient language models. Their relatively low memory and computational requirements make them suitable for applications where latency, cost, energy consumption, or local deployment is important.

In the proposed Semantically-Guided Evolutionary Algorithm for LLM Prompt Optimization, SLMs are primarily employed as semantic evaluators rather than as the final task-performing models. During evolutionary prompt optimization, a large number of candidate prompts may be generated through selection, crossover, and mutation. Evaluating every candidate directly using an LLM can be computationally expensive because each candidate requires an additional inference operation. An SLM can provide a lightweight mechanism for rapidly

evaluating candidate prompts according to their semantic similarity, relevance, clarity, and consistency with the original task.

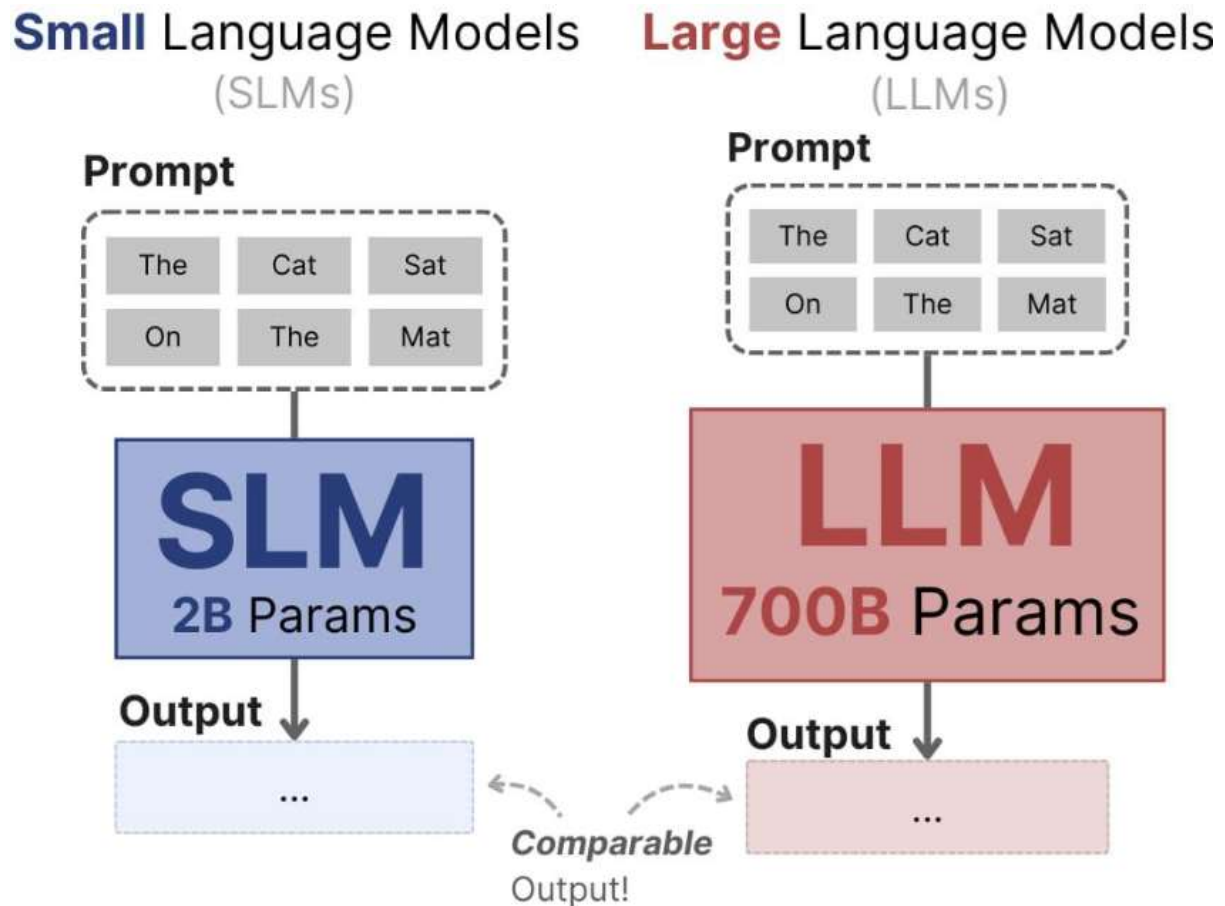


Figure 4: Difference between SLM and LLM

IV. CONCLUSION

The integration of Semantically-Guided Evolutionary Algorithms (SGEAs) with Large Language Models provides an effective approach for automated prompt optimization. Traditional manual prompt engineering is time-consuming and highly dependent on human expertise, while conventional optimization methods may modify prompts in ways that improve task performance but alter the original task intent. Semantic guidance addresses this limitation by evaluating the meaning, relevance, and consistency of candidate prompts during the optimization process.

In the proposed framework, Small Language Models (SLMs) are used as lightweight semantic evaluators, while evolutionary operations such as selection, crossover, mutation, and fitness-based optimization are used to generate progressively improved prompts. The combination of semantic similarity and task-specific performance enables the optimization process to preserve the intended meaning while improving the quality of LLM responses.

This approach can also reduce computational requirements by filtering unsuitable candidates before expensive LLM evaluation.

Overall, semantically guided evolutionary optimization provides a promising foundation for developing accurate, robust, efficient, and automatically optimized prompts. The proposed methodology can be applied to question answering, text classification, summarization, sentiment analysis, information extraction, and reasoning tasks. Future research can focus on multi-objective optimization, adaptive SLM selection, cross-model prompt generalization, multimodal LLMs, and real-time prompt optimization to further improve the effectiveness and scalability of the proposed approach.

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.
- [3] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.
- [4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685*, 2021.
- [5] L. Ouyang *et al.*, "Training Language Models to Follow Instructions with Human Feedback," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.
- [6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.
- [7] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large Language Models Are Human-Level Prompt Engineers," *International Conference on Learning Representations (ICLR)*, 2023.
- [8] Q. Guo, R. Wang, J. Guo, B. Li, K. Song, X. Tan, G. Liu, J. Bian, and Y. Yang, "Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers," *International Conference on Learning Representations (ICLR)*, 2024.
- [9] R. Pryzant, D. Iter, J. Li, Y. Lee, C. Zhu, and M. Zeng, "Automatic Prompt Optimization with 'Gradient Descent' and Beam Search," *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [10] C. Fernando, D. S. Banarse, H. Michalewski, S. Osindero, and T. Rocktäschel, "Promptbreeder: Self-Referential Self-Improvement via Prompt Evolution,"

- Proceedings of the 41st International Conference on Machine Learning (ICML)*, vol. 235, pp. 13481–13544, 2024.
- [11] O. Khattab *et al.*, “DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines,” *International Conference on Learning Representations (ICLR)*, 2024.
 - [12] M. Yüsekşönül, F. Bianchi, J. Boen, S. Liu, Z. Huang, C. Guestrin, and J. Zou, “TextGrad: Automatic ‘Differentiation’ via Text,” *arXiv preprint arXiv:2406.07496*, 2024.
 - [13] K. Opsahl-Ong, M. J. Ryan, J. Purtell, D. Broman, C. Potts, M. Zaharia, and O. Khattab, “Optimizing Instructions and Demonstrations for Multi-Stage Language Model Programs,” *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9340–9366, 2024.
 - [14] J. Xian, S. Samuel, F. Khoubisrat, R. Pradeep, M. A. Sultan, R. Florian, S. Roukos, A. Sil, C. Potts, and O. Khattab, “Prompts as Auto-Optimized Training Hyperparameters: Training Best-in-Class IR Models from Scratch with 10 Gold Labels,” *arXiv preprint arXiv:2406.11706*, 2024.
 - [15] P. Zhang, G. Zeng, T. Wang, and W. Lu, “TinyLlama: An Open-Source Small Language Model,” *arXiv preprint arXiv:2401.02385*, 2024.
 - [16] M. Abdin *et al.*, “Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone,” *arXiv preprint arXiv:2404.14219*, 2024.
 - [17] L. A. Agrawal *et al.*, “GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning,” *arXiv preprint arXiv:2507.19457*, 2025.
 - [18] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, “Universal and Transferable Adversarial Attacks on Aligned Language Models,” *arXiv preprint arXiv:2307.15043*, 2023.