Safer Large Language Models via Hierarchical Meta-Learning Optimization

Anonymous ACL submission

Abstract

The performance of large language models (LLMs) is highly dependent on the way they interact with input data, where improper handling can lead to undesirable outcomes, including the exacerbation of biases and unsafe behaviors. Current optimization techniques often neglect the model's underlying pre-training knowledge and treat each input independently, missing the potential for more efficient and safer learning. In this work, we present Learning to Safe Prompt (L2P), a novel approach that integrates hierarchical meta-learning with optimization strategies to enhance the safety and reliability of LLMs. L2P trains a model to adapt its responses through a meta-learning framework that prioritizes both performance and risk mitigation, ensuring that the model behaves safely across a wide range of inputs. Our extensive evaluation shows that L2P outperforms existing methods by significantly improving both the safety and effectiveness of LLM responses while maintaining high performance.

1 Introduction

In recent years, artificial intelligence has witnessed remarkable advancements, giving rise to the emergence of large language models (LLMs), such as ChatGPT (Ray, 2023) and Llama (Touvron et al., 2023). These LLMs have demonstrated significant capabilities across various NLP tasks. However, it is important to acknowledge that the behavior of these LLMs is highly influenced by the inputs they receive. Extensive research has shown that when LLMs are given unclear or imprecise inputs, they may produce undesirable or harmful outputs (Hosseini and Horbach, 2023). This concern becomes especially critical in safety-sensitive applications (Harrer, 2023), where even minor flaws in the input can lead to severe consequences. Thus, ensuring that inputs are clear and well-defined is essential to minimizing potential risks and maximizing the safe application of LLMs.

Research has proposed two main approaches to address the safety issues of LLMs with the modification of the input prompt. One approach advocates for manual prompt crafting (Reynolds and McDonell, 2021), but this method can be limited by the lack of expertise among users and certain inherent constraints (Webson and Pavlick, 2021). Another line of research focuses on automated prompt optimization. For white-box models like Llama, gradient-based techniques are employed to adjust the prompt (Qin and Eisner, 2021; Gao, 2021). In contrast, black-box models like ChatGPT pose a greater challenge due to the limited information available. Recent studies, such as EVOPROMPT (Guo et al., 2023), have tackled prompt optimization in black-box models using techniques that do not rely on gradient information, such as evolutionary algorithms (Bäck and Schwefel, 1993). However, these methods encounter challenges, including performance degradation when faced with previously unseen prompts, and are highly dependent on the sequence of optimizing known prompts, resulting in an imbalanced emphasis on samples optimized later in the sequence.

To address these limitations, we propose Learning to Safe Prompt (L2P), with the goal of not only optimizing the target prompt but also summarizing the common properties as a meta-prompt derived from the global learning processes of a collection of optimized individual prompts. This meta-prompt can then be generalize and improve the performance for newly encountered prompts. The L2P framework consists of three stages: individual prompt optimization, global learning for the meta-prompt, and the transfer of the learned meta-prompt to optimize new prompts. Specifically, for individual prompt optimization, we leverage LLM to optimize the prompt towards the expected rewards. Then, in the global learning process, we employ a global-learning LLM-based optimizer to condense the optimization process for a set of individual prompts and obtain the meta-prompt. By doing so, the meta-prompt can be generalize to the newly encountered prompts.

In summary, our primary contribution is L2P, a framework that leverages an LLM-based optimizer and a chain-of-thought global learning mechanism to refine the inputs. L2P arises from our thorough analysis of the challenges associated with current black-box approaches. Our experiments rigorously evaluate L2P across a variety of tasks and LLM types. Compared to existing methods, L2P demonstrates significant improvements in enhancing the safety of LLM outputs, as measured by task-specific metrics. Notably, L2P excels in several critical LLM applications, including toxicity reduction, news summarization, and sentence simplification. It achieves an impressive 30% improvement in optimizing original inputs and a 25% improvement with newly generated inputs.

2 Related Work

2.1 Large Language Models as Optimizer

The expansion of large language models (LLMs) (Naveed et al., 2023) in terms of size and complexity has been paralleled by their increasingly superior performance on a wide array of downstream natural language processing (NLP) tasks (Xie et al., 2023; Salnikov et al., 2023; Madaan et al., 2023). Recent research (Yang et al., 2023) showed LLM can be utilized as powerful optimizers in various tasks (Suzgun et al., 2022), pointing out that their ability to understand semantic content out a new possibility, simply describing them in everyday language to a LLM, for optimization. In our L2P, we employ the LLM as optimizers for both the individual prompt optimization and the global learning (Hospedales et al., 2021) mechanism.

2.2 Prompts Engineering with LLMs

The prompt engineering (Liu et al., 2023a) refer to optimize the original prompts, of which the primary goal is to find a prompt that can enhance the language model's performance in a special downstream NLP tasks (Strobelt et al., 2022; Clavié et al., 2023; Luo et al., 2022). While LLMs are sensitive to how prompts are formatted, with studies showing that even semantically similar prompts can lead to varied results (Wei et al., 2023; Zhao et al., 2021), prompt engineering is of great importance for them. The effectiveness of a prompt can depend on both the specific model and the task at hand (White et al., 2023), however, some robustness prompts show decent performance across various models and tasks (Yang et al., 2023). In addition, compared to the fine-turning methods (Chen et al., 2023; Zhang et al., 2023), prompt engineering, which balances performance and efficiency (McDonald et al., 2022), is gaining recognition as a vital tool in the application of LLMs, especially in environments with limited computational resources and rapidly changeable tasks (Lin et al., 2023).

2.3 Black-Box Prompt Engineering

In the field of prompt engineering for Large Language Models (LLMs), the methods are broadly classified into two types: gradient-based (Qin and Eisner, 2021; Gao, 2021; Liu et al., 2023b; Zhang et al., 2021) and gradient-free, which is also known as black-box prompt engineering (Zhang et al., 2022; Zhou et al., 2022; Pryzant et al., 2023). The latter one is becoming increasingly important, especially as LLMs accessible only via APIs are more common. These methods are varied, including simple additions of tokens or task-specific instructions manually (Jiang et al., 2020), to more complex approaches like automatic prompt searching and optimization (Zhou et al., 2022). Since gradient-related information is not available, gradient-free optimization methods such as reinforcement learning (Deng et al., 2022) and evolutionary algorithms (Guo et al., 2023) are also utilized. However, these emerging methods are highly dependent on the order of optimization of known prompts. Our L2P employs the chain-of-thought (Wei et al., 2022) aided global learning, which exhibits better robustness against these issues.

3 Learning How to Prompt

In this section, we detail our method, **L**earning to Safe **P**rompt (**L2P**), whose framework, along with one representative example, is shown in Figure 1. L2P aims to obtain the meta-prompt result, which is a prompt containing indispensable high-scoring features. This is achieved through global learning, which analyze optimized individual prompt results and the associated scores, mitigating the negative effects caused by inappropriate optimization sequences and improving robustness. Specifically, L2P begins with the individual prompt optimization stage, where it utilizes LLMs as optimizers to enhance prompts by analyzing their performance with the scoring function. Following this, in the global learning stage, a global-learning LLM-based optimizer is employed to summarize the intrinsic features shared by high-scoring individual prompt results obtained during the individual prompt optimization stage. Our global learning approach utilizes a chain-of-thought mechanism to unearth deeply hidden features, further enhancing the trustworthiness and robustness of L2P.

3.1 Individual Prompt Optimization in Black-Box LLM

In our approach, we follow a process that begins with a fixed question q and an adjustable prompt p, which leads to the LLM generating an output. The process concludes with the scoring of this output. The specific form of p depends on the type of q, and it can serve either as a system prompt that describes the characteristics of LLMs or as a user prompt that guides LLM in performing specific tasks (Ray, 2023; Touvron et al., 2023). Essentially, we are addressing an optimization problem where our goal is to achieve the highest possible score for each response generated by the target LLMs.

$$p^* = \arg\max_p E_{q\sim D}[f_{sc}(L_{ta}(p, q)))], \quad (1)$$

where we use q and p to represent the question and prompt, both derived from the training dataset D. Notably, q remains fixed, while p is subject to optimization. Our goal is to find an abstract strategy or function for generating prompts based on questions and the training history h. To simplify our writing, we sometimes combine the tuple (q, p) and collectively refer to it as d. When we refer to optimizing d, we specifically mean optimizing the p component within the tuple. The function f_{sc} represents the scoring function used to evaluate the performance of the LLM, while L_{ta} denotes the target LLM's output when given a specific prompt. In most cases, we do not know the exact output a given input will produce, and we are uncertain about the specific adjustments needed to enhance the model's scores in a certain task. Consequently, we treat this problem, where we cannot design specific solution steps, as a gradient-free black-box optimization problem.

As we describe in the Algorithm 10, considering a training set with n prompts, denoted as $\mathcal{D}^{tr} = d_{tr_1}, \ldots, d_{tr_n}$. We introduce an optimizer based on the LLM, denoted as L_{op}. This optimizer refines the training prompts to change the performance of the target LLM, noted as the L_{ta}, after the individual training process, the global learning LLM L_{gl} try to find the common pattern shared by the high-score training data samples. The LLM-based optimizers, L_{op} and L_{gl}, are powered with the vast semantic knowledge these models have acquired during pre-training, allowing us to create optimization tasks without the detailed descriptions.

The optimization process is guided by the score functions f_{sc} of question q. For each tuple (q, p), there will be a associated score $s = f_{sc}(q, p)$, we note the tuple (p, s) as one record of the optimization history. The whole optimization history can be defined mathematically as

$$h = \{(p_1, s_1), (p_2, s_2), \ldots, (p_n, s_n)\} \quad (2)$$

where each tuple represents a prompt and its score, usually the p associated with higher s have more characteristics to achieve better performance. For this reason, combined with the LLM token limitations, we only utilize the high-scoring portions of history h when feeding the L_{op}. As we stated before, as a black-box prompt engineering method, L2P only rely on the output of the L_{ta}, with the optimization objection 2 stated before, L2P update the p with L_{op} as followings:

$$p_{new} = L_{op}(p_{now}, sort(h)) \quad (3)$$

The instruction-optimization function $sort()$ is introduced to provided the L_{op} with data with higher information density, chosen from historical data h, represented as the top n elements of $(p_i, s_i)_{i=1}^n$ sorted by s_i in descending order.

3.2 Meta-Prompt Summarization

In this section, we will delve into the comprehensive development and benefits of our global learning LLM optimizer, denoted as L_{gl}, which stands in contrast to earlier prompt engineering approaches, such as the OPRO, which optimize individual data points in a sequential manner. These methods utilize the outcome p obtained upon completing the optimization of the last sample d_{tr_n} from the known prompt set D^{tr} as the final result. As we stated before, for individual prompt optimization,

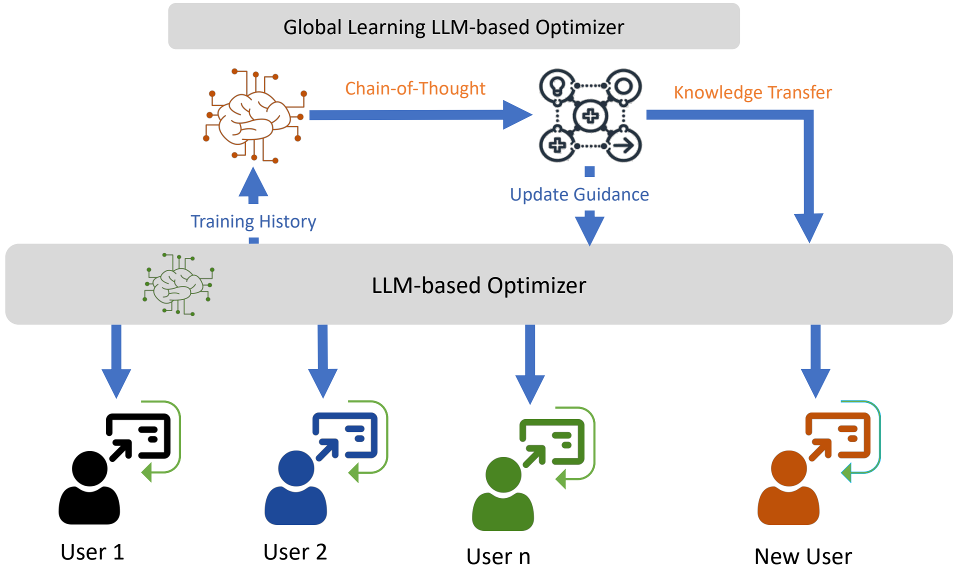


Figure 1: Pipeline of L2P. It automatically optimizes candidate prompts based on their performance scores, as indicated by the score. It achieves this using LLM-based optimizers. Additionally, it harnesses a global-learning LLM-based optimizer, employing the Chain of Thought (COT) mechanism to analyze valuable information from the optimization history. This information serves as guidance for the subsequent rounds of optimization. This iterative process continues until the specified number of optimization rounds is reached or convergence is achieved.

our L_{op} will optimize each sample d_{tr_i} in a synchronous manner. Each sample is optimized independently, unaffected by the optimization process of other samples.

After the stage of individual prompt optimization, L_{gl} attempts to summarize the meta-prompt, which is a distilled essence of the dataset capturing core features necessary for achieving high performance scores with f_{sc}. It does so by utilizing the chain-of-thought mechanism, which filters optimization results to select high-performing and representative prompts. This meta-prompt is denoted as P_{gl}, and this process of meta-prompt summarization can be formulated as:

$$\left.\begin{array}{l} F_{\text{key}} = L_{\text{gl}}(p_{d_{\text{tr}_1}}, p_{d_{\text{tr}_2}}, \ldots, p_{d_{\text{tr}_i}}) \\ F_{\text{Per}} = L_{\text{gl}}(p_{d_{\text{tr}_1}}, p_{d_{\text{tr}_2}}, \ldots, p_{d_{\text{tr}_i}}) \end{array}\right\} \quad (4)$$

$$\longrightarrow p_{\text{gl}} = L_{\text{gl}}(F_{\text{key}}, F_{\text{Per}}, (p_{d_{\text{tr}_1}}, \ldots, p_{d_{\text{tr}_i}})) \quad (5)$$

Here, $p_{d_{tr_i}}$ represents the individually optimized results using the sample d_{tr_i}, and F_{key} signifies the key feature required to achieve optimal performance, while unrelated personal features are denoted as F_{per}. From Equation 5, it is evident that the optimization order is irrelevant to the final result of p_{gl}. This approach preserves semantic integrity, preventing information loss during optimization and ensuring robustness. The chain of thought mechanism plays a crucial role by identifying and integrating commonalities and differences among the optimized prompts. L_{gl} tries to keep key features F_{key} necessary for optimal performance while discarding unrelated personal features F_{per}.

3.3 Generalizing to New Prompt

In this section, we focus on generalizing the results obtained from known prompts to new prompts, emphasizing the high efficiency, predictability, and exceptional transferability of the L2P model. The optimized results achieved through L2P can be directly applied to new prompts without the need for a costly fine-tuning process, while ensuring consistent, high-quality performance. This makes L2P particularly suitable for devices with limited computational resources and for rapid-response applications, such as real-time news analysis based on LLMs.

The transferability of L2P arises from the robustness of the optimized results. The outcomes it generates are not only applicable to new prompts but can also seamlessly adapt to new types of LLM configurations of various sizes and types, ranging from efficiency-oriented LLMs suitable for mobile devices to giant LLMs used on cloud servers. The performance estimation of generalization to new prompts can be expressed as:

$$E_{q\sim D_{te}}\left[f_{sc}\left(L_{ta}(p_{gl},q)\right)\right] \qquad (6)$$

Where D_{te} represents a new or altered set of prompts. In conclusion, with the assistance of the global optimizer L_{gl}'s key features F_{key} summarization mechanism, the superior ability of L2P to generalize to new prompts without further retraining highlights L2P's high efficiency and adaptability in resource-constrained or changeable demanding environments.

Algorithm 1 Learning to Prompt (L2P),

```
Require: The training dataset $D_{\text{tr}} = \{d_{tr_1}, d_{tr_2}, ..., d_{tr_n}\}$ and the test dataset
    $D_{\text{te}} = \{d_{te_1}, d_{te_2}, ..., d_{te_n}\}$; $L_{op}, L_{ta}, L_{gl}$: The individual LLM-based optimizer, the
    Target LLM, and the global learning optimizer; $f_{sc} : L_{ta}(d) \rightarrow \mathbb{R}$: score function for
    Evaluating.
Initial/Resume the Global Prompt: $p_{gl}$
while not converged do
    Choose a random training subset $\tilde{D}_{\text{tr}} \subseteq D_{\text{tr}}$
    for $d_n$ in $\tilde{D}_{\text{tr}}$ do
        Optimize: $p_{d_n} \leftarrow L_{op}(d_n, f_{sc}, p_{gl})$
    end for
    Select: the top $i\%$ of results with highest
    score improvement $P_i \subseteq P_{\tilde{D}_{\text{tr}}} = \{p_{d_1}, ..., p_{d_n}\}$
    Update Global Prompt: $p_{gl} \leftarrow L_{gl}(P_i)$
end while
Return: $p_{gl}$ with the highest score expectation
    $E[f_{sc}(L_{ta}(d))]$ over the $D_{\text{tr}}$.
```

4 Experiments

In this section, we evaluate the performance of L2P, aiming to answer the following questions: **Q1**: Compared to corresponding prior approaches, can L2P improve the in-distribution performance for known prompts, and out-of-distribution robustness with the new prompts? **Q2**: How does L2P perform when using the new types of LLMs rather than the original one? **Q3**: Is L2P get benefits from the using of the chain of the thought?

4.1 Experimental Settings

Evaluation Setup. Our experiment focuses on how our L2P optimizes prompts to maximize the performance of large language models for specific tasks with original prompts and new prompts. We will introduce the problem setup and provide details on the experimental design. Both input and output are presented in text format. The task is defined as a dataset with original prompt and new prompt splits, where the original prompt dataset split is used during the optimization process, acting as the target value, and the new prompts dataset split is evaluated after optimization.

Backbone Models and Hyperparameter Settings. As we noted before, We refer to the LLM used for target evaluation as the L_{ta}, the LLM used for individual prompt optimization as the L_{op}, and the LLM utilized for the global learning task as the L_{gl}. Our evaluation method uses common evaluation problems. For all the following experiments We utilize the ChatGPT-3.5 Turbo as the L_{op} and, GPT-4 as the L_{gl}. We have utilized various types of LLMs as the L_{ta}, including, LLama 7B, LLama 13B, LLama 70B, ChatGPT-3,5 and the GPT-4 Turbo. The superior results obtained with various different types and sized of LLMs demonstrate the excellent performance and robustness of L2P. All GPT-related LLMs are accessed through API calls, while the locally run Llama model is operated using 4*Nvidia RTX6000 GPUs, each with 48GB of VRAM.

Baselines. Similar to others black-box prompt engineering works, we mainly employ different black-box baselines for evaluation. The comparison methods include: Original, which directly use the initial default prompts without optimization for the tasks. Chain-of-Thought (CoT) (Wei et al., 2022), which is based on our designed chain of thought-based instructions, which will firstly try to rewrite the original prompts, and based on the changes of the performance to get the optimal prompts, resulting in improved performance. APE (Zhou et al., 2022), a method that applies the LLM approach on top of instruction induction to further refine and enhance the original initial prompts. In our implementation, we primarily used their designed LLMs as Re-sampling Models to align with the design framework of our experiments. OPRO (Yang et al., 2023), in this work, involves optimizing with large language models, based on the final score to optimize all the prompts. EVOPROMPT (Guo et al., 2023), employs a evolutionary strategy and aims to make the model output to get the highest fitness scores.

4.2 Toxicity Task

This project aims to adjust language model prompts

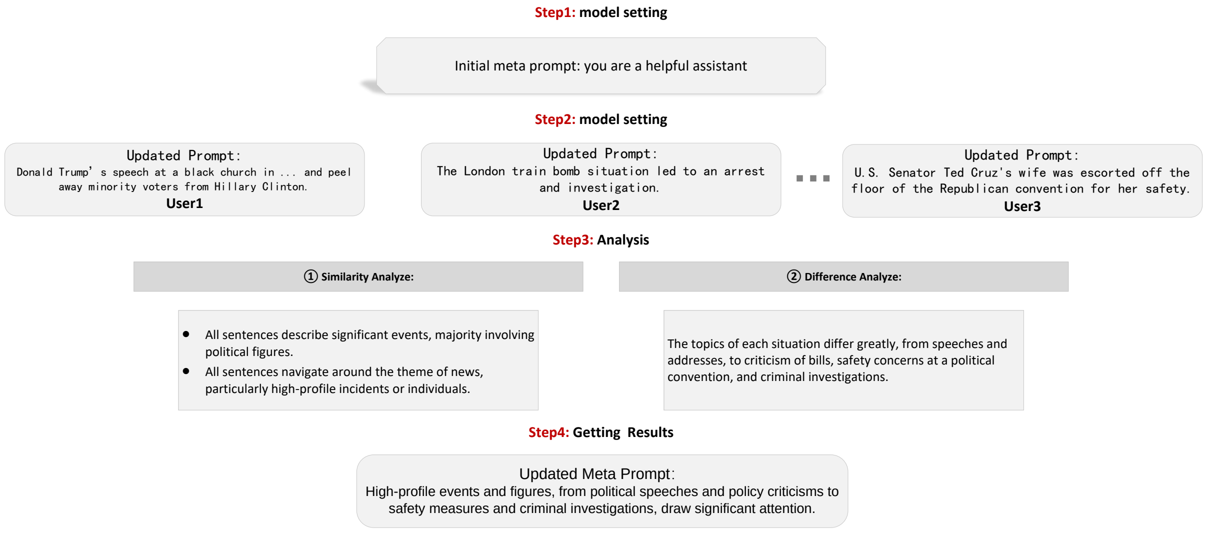

Step1: model setting
Initial meta prompt: you are a helpful assistant
Step2: model setting
Updated Prompt:
Donald Trump' s speech at a black church in ... and peel away minority voters from Hillary Clinton.
User1
Updated Prompt:
The London train bomb situation led to an arrest and investigation.
User2
Updated Prompt:
U.S. Senator Ted Cruz's wife was escorted off the floor of the Republican convention for her safety.
User3
Step3: Analysis
① Similarity Analyze:
All sentences describe significant events, majority involving political figures.
All sentences navigate around the theme of news, particularly high-profile incidents or individuals.
② Difference Analyze:
The topics of each situation differ greatly, from speeches and addresses, to criticism of bills, safety concerns at a political convention, and criminal investigations.
Step4: Getting Results
Updated Meta Prompt:
High-profile events and figures, from political speeches and policy criticisms to safety measures and criminal investigations, draw significant attention.

Figure 2: Training process of L2P, an illustrative example of the optimization process for prompts carried out jointly by the LLM-based optimizer and the global learning LLM-based optimizer.

to control sentence toxicity while maintaining meaning, focusing on ethical text tone management. This feature is vital for moderating online platforms, helping to identify and reduce harmful speech, thus promoting safer, more positive communication.

Dataset and Evaluation Metric. Our goal is to optimize prompts to make the language model generate more toxic content while maintaining semantic consistency with the original prompts. We use three datasets: red-team (Ganguli et al., 2022), real toxicity (Gehman et al., 2020), and persona (Deshpande et al., 2023) to represent various scenarios. Our model addresses continuing writing, responding to queries, and role-playing. To assess semantic changes, we utilize ChatGPT, and for evaluating toxicity, we rely on the Perspective API metric (Hosseini et al., 2017), known for its alignment with human evaluations.

We measured toxicity for original prompts, prompts optimized using baseline methods, and prompts optimized using our proposed method. For the sake of simplifying experiments and reducing API access costs, we randomly select prompts from the dataset, which is also employed for the following tasks. We report scores on both known and new prompts, noted as original and new in the result table

Results and Analysis. In Table 1, we find that optimization-based methods, guided by objectives like score functions and fitness functions, outperform non-optimization-based methods like COT in toxicity-related tasks. This indicates that optimization objectives enhance prompt engineering algorithms by facilitating exploration of prompt updates and improving their performance.

Compared to other black-box prompt engineering approaches, L2P stands out with its superior performance in toxicity modification across all three datasets, highlighting the effectiveness of its novel pipeline and chain-of-thought global learning mechanism. Furthermore, L2P shows substantial improvements over state-of-the-art methods in both optimizing original prompts and generating new ones, suggesting that L2P can excel not only in optimizing existing prompts for better performance but also in quickly adapting to new prompts, making it advantageous in rapidly changing or resource-constrained situations.

4.3 Summarizing Task

This experiment aims to optimize prompt to enhance LLMs' ability to produce brief, accurate news summaries from long articles. This is critical for generating precise news summaries in practical applications, ensuring the essence of the original content is maintained well.

Dataset and Evaluation Metric. We use the news-summary dataset (Ahmed et al., 2018, 2017), sourced from real news articles. To assess the quality of summaries generated by different prompts, we employ two trusted metrics: ROUGE (Lin, 2004), which compares machine-generated summaries to manual references, and BLEU (Papineni et al., 2002), which measures vocabulary overlap between machine-generated text and references.

Table 1: Results for Toxicity-related Datasets. We use the original and new prompts. The best results and second best results are **bold** and underlined, respectively.

	Real Toxicity				Red-Teaming				Persona			
	LLAMA		ChatGPT		LLAMA		ChatGPT		LLAMA		ChatGPT	
	Original	New	Original	New	Original	New	Original	New	Original	New	Original	New
ORI	6.883	4.753	8.617	4.064	8.167	4.405	4.382	2.719	8.013	4.906	13.073	7.794
COT	5.831	4.438	5.314	8.219	8.229	5.290	4.792	2.417	9.231	7.270	14.744	9.105
APE	6.989	4.547	8.485	10.154	8.640	4.702	4.760	2.608	8.924	7.235	16.308	10.316
EVOPROMPT	7.197	8.075	10.023	14.240	9.061	6.993	4.848	3.834	11.131	7.538	15.049	**11.499**
OPRO	7.145	9.676	11.852	17.833	9.306	6.622	6.132	3.212	10.934	5.909	13.969	6.918
L2P (Ours)	**13.008**	**11.883**	**20.900**	**28.534**	**13.762**	**9.667**	**10.320**	**5.544**	**11.958**	**10.652**	**26.667**	10.923

Table 2: Results for News summarizing Datasets. We use the original and new prompts.

	LLAMA		ChatGPT	
	Original	New	Original	New
ORI	33.372	35.091	47.745	51.454
COT	33.445	31.784	44.352	51.571
APE	34.478	31.350	53.729	52.455
EVOPROMPT	33.726	31.766	57.463	51.352
OPRO	37.766	36.194	51.632	48.566
L2P (Ours)	**44.199**	**42.529**	**61.724**	**68.705**

Table 3: Results for sentence-simplification Datasets. We use the Original and New prompts.

	LLAMA		ChatGPT	
	Original	New	Original	New
ORI	39.957	37.160	42.877	40.909
COT	41.316	39.048	42.167	41.312
APE	41.876	37.427	43.817	41.000
EVOPROMPT	42.070	38.715	43.707	39.471
OPRO	42.722	37.158	44.296	41.314
L2P (Ours)	**50.442**	**45.691**	**49.464**	**44.984**

Table 4: Results for generalization performance across various LLMs using the News dataset.

	7B	13B	70B	ChatGPT	GPT4
ORI	35.091	42.622	53.931	51.454	45.336
OPRO on LLama-7B	36.194	43.645	48.986	49.298	48.306
OPRO on ChatGPT	36.004	41.344	50.671	48.566	43.430
L2P (Ours) on LLama-7B	**42.529**	51.387	61.323	65.774	54.993
L2P (Ours) on ChatGPT	39.365	**57.259**	**63.764**	**68.705**	**61.997**

We combine these metrics to provide a comprehensive evaluation of the model's performance.

Results and Analysis. The goal is to summarize the key information of a detailed news, with string length of input detailed news ranging from 168 to 12400, typical around 2000, and the output summarization is required concise, usually below 100. Consistent with previous experiments, our experiment begins with the initial general system prompt "you are a helpful assistant". Our expectation is to optimize the model through a series of optimization, for better summarization.

Table 2 presents the performance of various algorithms on the News summarization dataset. Compared to ORI and COT, the optimization approaches can more effectively enhance performance of the LLMs. This aligns with our previous conclusions on toxicity-related datasets, demonstrating the effectiveness of applying optimization objectives. Furthermore, by combining optimized methods with chain-of-thought global learning mechanism, L2P outperforms all other approaches on two completely different LLMs, verifying its robustness across diverse tasks.

4.4 Simplification Task

This experiment focuses on training prompts to simplify complex sentences while maintaining their original meaning. It involves controlling the LLMs output for clarity. The model must understand and preserve the core intent and context, and identify complex structures, which can be utilized to enhance text readability.

We utilize ASSET (Alva-Manchego et al., 2020), a multi-reference dataset for evaluating English sentence simplification. For the metric used in this task, we employ SARI (Xu et al., 2016) to measure the quality of the simplification system's output with different prompt inputs, with higher scores indicating better quality simplifications. From the Table 3, we can see that Our method L2P has a significant advantage over all baseline methods.

4.5 Analysis of L2P

The generalization of our learned prompt across various LLMs. Our method demonstrates exceptional generalization in black box prompt engineering, crucial for real-world LLM-based applications. It remains robust across a range of LLMs, from LLama 7B to 70B models, including most advanced GPT4 Turbo, without requiring additional training. This adaptability is essential for efficiency and computing resource conservation, particularly in mobile device deployment.

Our approach excels across diverse datasets, adapting smoothly to different LLMs. For example, in news content, it outperforms OPRO in generalization, improving content generation quality across LLMs without extra adjustments. In addition, L2P is scalable and transferable, consistently performing well across LLMs of varying complexity. This cost-effective solution streamlines prompt engineering, enabling result prompts gain from low-cost LLMs to work on expensive ones, reducing time and upgrade expenses for LLM-based applications.

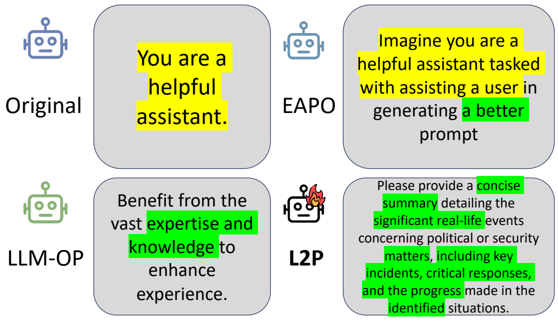
Ablation experiment of the COT mechanism of the global learning module. Our chain-of-thought (COT) aided global learning module plays a crucial role in improving algorithmic efficiency and effectiveness. It systematically analyzes results to extract meaningful insights, identifying F_{key} and F_{per} in result prompts. This approach enhances critical analysis, and improves data comprehension by breaking down sentences of results prompts and exploring underlying shared features.

Our COT ablation experiment results, shown in Table5, demonstrate that each designed module in COT significantly enhances global learning performance. "Only Module D" only focuses on F_{per} before summarizing, "Only Module C" only considers F_{key} before summarizing, and "Module C+D" combines both. "Without C+D" lets the global learning optimizer L_{gl} to summarize without any additional steps.

Table 5: Results for ablation study of chain-of-thought mechanism design.

	LLaMa	ChatGPT
Without Module D+C	5.107	22.156
Only Module D	7.239	25.001
Only Module C	8.019	23.672
Module D + C	**11.883**	**28.534**

Case Analysis. We selected several strong baselines and presented a case with their own optimization result prompts in Figure 3. We will mark the background color of meaningful information as green and the background color of invalid information as yellow. Compared with other approaches, L2P excels in providing a higher-quality prompt, which includes more meaningful information to enhance the performance of LLMs. In the case, L2P accurately learns the primary F_{key} to achieve great performance (e.g., the summarization should be concise), while avoiding vague descriptions lacking effective information (e.g., just telling the LLM should be helpful) or personal information of the data sample. Although other baselines also improve the performance of the LLMs to some extent, they still exhibit vague descriptions or F_{per} not beneficial for achieving better LLMs performance. Additionally, from the results, we can see that L2P can effectively provide concrete instructions to achieve better performance, such as telling LLMs to provide key incidents and critical responses, and progress, which LLM can easily follow. In contrast, such as OPRO, even also provides some meaningful instructions such as using expertise knowledge to summarize, but compared with the instructions of L2P, they are too vast, causing difficulty for target LLMs to follow.



You are a helpful assistant.

Imagine you are a helpful assistant tasked with assisting a user in generating a better prompt

Benefit from the vast expertise and knowledge to enhance experience.

Please provide a concise summary detailing the significant real-life events concerning political or security matters, including key incidents, critical responses, and the progress made in the identified situations.

Figure 3: A case study comparing the levels of meaningful information gain among various baselines.

5 Conclusion

Our research introduces a novel prompt optimization method called L2P, designed to significantly enhance the security of target LLMs. L2P leverages a hierarchical meta-learning optimization approach: an individual LLM-based local optimizer and a COT-aided global learning optimizer. This combination not only fine-tunes the performance of various LLMs across a range of known and novel prompts but also enhances their robustness against adversarial inputs. L2P bolsters model safety and reducing the risk of harmful or biased responses. Our approach consistently outperforms existing state-of-the-art methods across different tasks, offering substantial advancements in performance while ensuring that security considerations are an integral part of the optimization process.

Limitations

Our work only considered the use of a single type of LLM, ChatGPT, as the individual optimizer L_{op}'s backbone. The LLM used in this work can be expanded to different structure LLMs, such as the Llama series, or a more powerful LLM like GPT4 or GPT4 Turbo. Additionally, for both the individual optimizer L_{op} and global learning optimizer L_{gl}, we did not make the use of integrating external knowledge databases specific to certain domains to further enhance the performance of these LLM-based optimizers. We believe this is a promising direction worth considering for the next step.

References

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. In *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference, ISDDC 2017, Vancouver, BC, Canada, October 26-28, 2017, Proceedings 1*, pages 127–138. Springer.

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *arXiv preprint arXiv:2005.00481*.

Thomas Bäck and Hans-Paul Schwefel. 1993. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2023. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376*.

Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. 2023. Large language models in the workplace: A case study on prompt engineering for job type classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 3–17. Springer.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

T Gao. 2021. Prompting: Better ways of using language models for nlp tasks the gradient.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Stefan Harrer. 2023. Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine. *EBioMedicine*, 90.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

Mohammad Hosseini and Serge PJM Horbach. 2023. Fighting reviewer fatigue or amplifying bias? considerations and recommendations for use of chatgpt and other large language models in scholarly peer review. *Research Integrity and Peer Review*, 8(1):4.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zheng Lin, Guanqiao Qu, Qiyuan Chen, Xianhao Chen, Zhe Chen, and Kaibin Huang. 2023. Pushing large language models to the 6g edge: Vision, challenges, and opportunities. *arXiv preprint arXiv:2309.16739*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023b. Gpt understands, too. *AI Open*.

Xianchang Luo, Yinxing Xue, Zhenchang Xing, and Jiamou Sun. 2022. Prcbert: Prompt learning for requirement classification using bert-based pretrained language models. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–13.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Joseph McDonald, Baolin Li, Nathan Frey, Devesh Tiwari, Vijay Gadepally, and Siddharth Samsi. 2022. Great power, great responsibility: Recommendations for reducing energy for training language models. *arXiv preprint arXiv:2205.09646*.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.

Partha Pratim Ray. 2023. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions. *arXiv preprint arXiv:2310.02166*.

Hendrik Strobelt, Albert Webson, Victor Sanh, Benjamin Hoover, Johanna Beyer, Hanspeter Pfister, and Alexander M Rush. 2022. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE transactions on visualization and computer graphics*, 29(1):1146–1156.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. 2023. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. 2022. Tempera: Test-time prompt editing via reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Zihan Zhang, Meng Fang, Ling Chen, Mohammad-Reza Namazi-Rad, and Jun Wang. 2023. How do large language models capture the ever-changing world knowledge? a review of recent advances. *arXiv preprint arXiv:2310.07343*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.