RETRIEVAL AUGMENTED PROMPT OPTIMIZATION

Yifan Sun, Jean-Baptiste Tien, Karthik Lakshmanan
Google
{yifansun,jbtien,lakshmanan}@google.com

ABSTRACT

Prompt optimization for Large Language Models (LLMs) has recently made great strides in complex tasks such as solving arithmetic problems and reasoning. Yet, its efficacy remains limited in tasks demanding extensive domain expertise beyond the internal knowledge of LLMs. As context length increases, prompt optimization tends to plateau in performance, which limits the amount of domain knowledge we can provide in the prompt. We postulate that this difficulty stems from an inherent tradeoff between adding information and easing comprehension. To tackle this challenge, we present a divide-and-conquer approach (RAPO) to prompt optimization by means of retrieval augmentation. RAPO breaks the entire problem space into a number of subspaces, where each subspace can be handled separately by a local prompt specifically designed to cater to it. This approach not only scales more effectively to larger training datasets but also naturally accommodates domain knowledge (e.g., policy databases) and inference algorithms (e.g., re-ranking). Experimental results show that RAPO consistently outperforms recent methods (Yang et al., 2023; Pryzant et al., 2023) by a large margin across challenging datasets, including, a 7.4% relative AUCPR improvement on internal datasets by incorporating domain knowledge and 13.0% relative AUCPR gain on the public Sarcasm dataset (Abu Farha & Magdy, 2020). We hope our findings offer a new perspective of prompt optimization for knowledge-intensive tasks.

1 INTRODUCTION

Large Language Models (LLMs) have shown a variety of emergent capabilities (Wei et al., 2022). This success is significantly attributed to the art of prompt engineering - a process where the user provides an instruction in natural language or a set of demonstrations that guide the LLMs towards desired outputs. Traditionally, prompt crafting has been a manual process, demanding a deep understanding of both the LLMs' functioning and the task at hand. However, this method is not only time-consuming and labor-intensive but also prone to human biases and limitations. This setup is also difficult to maintain since we need to regenerate prompts as the underlying model versions change or we have slight change in the desired outcomes.

To tackle these challenges, the field has witnessed a surge in interest towards automating the process of prompt generation using LLMs themselves (Honovich et al., 2023; Zhou et al., 2023; Pryzant et al., 2023; Yang et al., 2023; Guo et al., 2023), known as prompt optimization. Prompt optimization leverages various techniques such as guided evolution (Guo et al., 2023), score maximization (Yang et al., 2023), and gradient estimation (Pryzant et al., 2023) to dynamically generate prompts that are tailored to specific tasks. Results in this domain have been promising, with automated prompts often outperforming their manually-engineered counterparts in a wide range of tasks.

Despite these advancements, the prompt optimization methods still suffer from an underfitting issue. Sometimes they even fail to fit a small training set with extensive prompt iterations. We hypothesize that this limitation stems from a fundamental tradeoff in prompt optimization: providing comprehensive information and ensuring the prompt comprehensible to the LLMs. On one hand, to be effective, a prompt must encompass all the necessary information about the downstream task. This often leads to longer, more detailed prompts. On the other hand, the prompt must be structured in a way that the LLMs can easily process and understand, which typically favors brevity and simplicity. Striking the right balance between these two objectives is a delicate and complex task.

In this work, we propose to address this inherent tradeoff through a divide-and-conquer approach. Specifically, we combine the strengths of retrieval augmentation with prompt optimization, breaking the optimization process over the entire problem into a number of smaller, more manageable sub-problems. Each sub-problem is addressed with a local prompt, specifically designed to cater to a particular aspect of the task. We then employ existing prompt optimization methods (e.g., OPRO Yang et al. (2023)) to optimize each local prompt. During inference, we construct an instance-specific prompt tailored to the input query. Concretely, we retrieve and rerank top-K relevant local prompts based on the input query and append them to the global task description and use the assembled prompt for inference.

We evaluate the proposed retrieval-augmented prompt optimization (RAPO) on both internal abuse detection datasets and a public Sarcasm dataset. RAPO demonstrates substantial performance improvements across all tasks, outperforming the state-of-art APO baseline (Pryzant et al., 2023) with reranking by 7.4% in relative test AUCPR and further incorporating domain policy to achieve a 13.0% relative test AUCPR increase.

2 METHOD

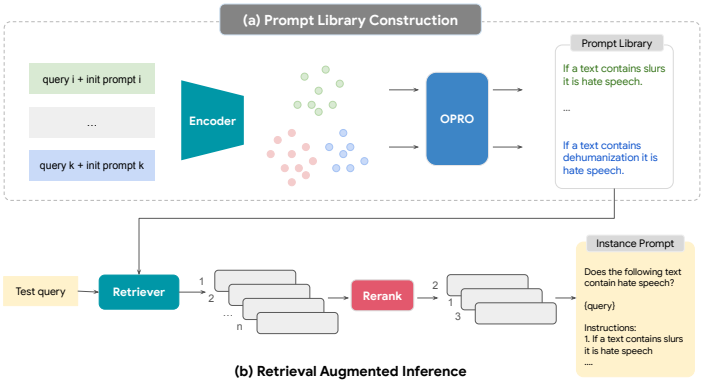

Figure 1: An overview of the retrieval-augmented prompt optimization framework. (a) Prompt Library Construction: group training data into clusters based on their task-dependent similarity and use OPRO to refine the instance prompt library. (b) Retrieval augmented inference: given an input query, retrieve and rerank top-K relevant instance prompts from the prompt library and construct a coherent prompt tailored to the input query.

At the core of our method is to divide prompt optimization in the entire problem space into smaller ones, wherein we can optimize a local prompt more effectively. This is achieved through two parts:

Prompt library construction. We design a prompt library that serves as an extensive knowledge base and contains instance prompts tailored to each individual training instance. This design is motivated by the fact that existing prompt optimization techniques are already capable of searching effective instance-prompt for a few queries.

Retrieval augmented inference. Once a prompt library has been constructed, we dynamically construct instance prompts for any input query during inference. This involves selecting the most relevant instance prompts and composing them into a coherent one tailored to the input query.

Figure 1 illustrates the framework of RAPO. We will next describe each component in more detail.

2.1 PROMPT LIBRARY CONSTRUCTION

To construct a specialized prompt library for our task, we start with designing instance prompts that (1) allow for dynamic adaptation to input queries and (2) encapsulate relevant knowledge for specific aspects of the task. Our approach involves:

Prompt Structure: We design each instance prompt with a common structure as $p = [p_g, p_l]$, consisting of a global component p_g with a customizable local component p_l. This uniform structure facilitates prompt reconfiguration for individual queries by reusing p_g and modifying p_l tailored to individual input queries.

Data Grouping: We split the training data into groups based on knowledge relevance for task aspects. Specifically, we use an off-the-shelf embedding model (e.g. the Vertex AI PaLM Embedding API) to estimate semantic similarity between different examples. For each training example, we preserve the top-K nearest neighbors measured by cosine similarity, thus offloading prompt optimization in a local group manner.

Instance Prompt Optimization: Having established groups, we optimize the local component p_l of each instance prompt using an off-the-shelf prompt optimizer (e.g., OPRO Yang et al. (2023)). To mitigate the risk of exploiting spurious features, we explicitly incorporate the domain knowledge as a constraint in each optimization step, as detailed in Appendix B.

This approach ensures each instance prompt is flexible enough to adapt to varying input queries, specialized for particular task aspects and adheres to the text constraints comprehensible by LLMs.

2.2 Retrieval Augmented Inference

This section outlines the dynamic construction of instance prompts tailored to specific queries. Drawing inspiration from the Retrieval-Augmented Generation (RAG) approach, the adaptation process comprises two crucial steps:

2.2.1 Retrieval

To create an adaptive prompt for an input, a crucial element is retrieving highly relevant instance prompts based on the input query. Our approach consists of two steps to address this challenge:

Candidate Retrieval: We employ the Vertex AI PaLM Embedding API for retrieving relevant instance prompts. However, any embedding model with good semantic similarity measurement capabilities can be used. Given an input x, this process returns a list of local components $\{p_{l_1}, \ldots, p_{l_m}\}$ by computing cosine similarity between the input x and each prompt component p_{l_i}.

Candidate Reranking: While our embedding model retrieves instance prompts through generic semantic similarity, these top candidates may not guarantee task-specific relevance. To refine that, we employ a pairwise ranking prompting method, as detailed by Qin et al. (2023). This approach includes the input x and a candidate pair (p_{li}, p_{lj}) into a single prompt for the LLM to rank. We then aggregate these rankings globally to assign a score s_i to each candidate p_{l_i}, enabling the selection of the top-K most relevant instance prompts. Further details are in Appendix C.

2.2.2 Composition

After retrieving the relevant local components, we construct an adaptive instance prompt by appending the top-K local components $\{p_{l_1}, \ldots, p_{l_k}\}$ to the global task description p_g sorted in ascending order. This approach ensures a tailoring of instance prompts to a given query, thereby enhancing both the information content and comprehension of the prompt in relation to the query.

3 Experiments

3.1 Data

The evaluation datasets employed in this study include an internal abuse detection dataset and a publicly available Sarcasm dataset. The internal dataset labeling process adheres to a sophisticated domain policy, which is developed by experts in community guidelines and policy development experts. This complexity of the datasets renders them natural testbeds for assessing the effectiveness of RAPO. The datasets are as follows:

- **Mandarin Hate Speech Classification**: a dataset consisting of Mandarin text annotated with hate speech labels. We randomly select 180 examples for training and 180 for testing.

- **Self-harm Classification**: an English dataset of text, each annotated with relevant Self-harm labels. We randomly sample 300 examples for training and 300 for testing.
- **Harassment Classification**: an English dataset of text, each labeled for the presence of harassment content. We randomly select 250 examples for training and 250 for testing.
- **Sarcasm** (Abu Farha & Magdy, 2020): an Arabic sarcasm detection dataset. We randomly sample 300 examples for training and 300 for testing.

3.2 Evaluation Setup

Models. The LLMs we use as the optimizer and the scorer are:

- Optimizer: instruction-tuned PaLM 2-L in the PaLM-2 model family (Anil et al., 2023).
- Scorer: instruction-tuned PaLM 2-S.

Implementation details. During prompt library construction, we employ the OPRO prompt optimization algorithm (Yang et al., 2023) to optimize each instance prompt. Further details about OPRO are in Appendix D. During inference, we first retrieve 10 candidates using the embedding model and subsequently refine to the top 3 through reranking.

Evaluation metrics. When evaluating the performance of generated instructions, we leverage the LLM in score mode. This involves computing a probability vector $y \in \mathbb{R}^C$ over all classes. We report the Area Under the Precision-Recall Curve (AUCPR) score relative to original starting prompt on the test set, determined by the formula:

$$S_{\text{method}} = \frac{\text{AUCPR}_{\text{method}} - \text{AUCPR}_{\text{original\_prompt}}}{1 - \text{AUCPR}_{\text{original\_prompt}}} \tag{1}$$

3.3 Baselines

We evaluate the proposed RAPO framework against the following baseline methods:

Human-Engineered Prompts. This consists of manually crafted prompts from domain experts.

Large Language Models as Optimizer (OPRO). The algorithm proposed by Yang et al. (2023) guides the LLM to generate new prompts based on previously discovered solutions and their scores. We adhere to the default parameters in the original paper and run 200 steps for each dataset.

Automatic Prompt Optimization (APO): Introduced by Pryzant et al. (2023), APO utilizes mini-batches of data to construct textual "gradients" that critique the current prompt. These gradients are then used to guide the LLM to generate improved revisions. We use the default parameters in the original paper and run 50 steps for each dataset.

For both OPRO and APO, we make slight adjustments to the meta prompts to ensure optimal performance with the PaLM models, facilitating a fair comparison.

3.4 Experimental Results

Overall Results. Table 1 presents our main results. The results suggest that RAPO can outperform all baseline methods across all four datasets considered in the study. On average, RAPO-Rerank improved over the APO and OPRO baselines by 4.27% and 13.7% relative test AUCPR respectively, while also improving over the original prompt p_0 by 31.0% and human-engineered prompts by 13.3%.

Reranking Ablation. We evaluate the effectiveness of a post-retrieval reranking strategy for relevant instance prompt retrieval. Table 1 demonstrates that applying reranking subsequent to a similarity search based on embedding similarity improved over RAPO-Base by 10.4% in relative AUCPR.

Domain knowledge Ablation. We next conduct experiments on the internal abuse detection datasets to investigate whether injecting domain knowledge info could benefit the rule library quality. Table 1 compares the outcomes of instance prompts initialization with and without domain expert policies. The results indicate that adding domain policy to RAPO-Base and RAPO-Rerank leads to respective increases of 10.9% and 6.1% in relative AUCPR scores, suggesting incorporating domain policy can induce effective domain knowledge to produce expert-level prompts.

Table 1: Relative test AUCPR on (a) Mandarin Hate Speech classification, (b) Self Harm classification, (c) Harassment classification and (d) public Sarcasm Dataset.

Method	Hate	Self Harm	Harassment	Sarcasm	Average
Baselines					
Starting	0.00	0.00	0.00	0.00	0.00
Human	24.67	24.23	17.84	4.02	17.69
OPRO (Yang et al., 2023)	26.46	16.18	4.14	22.36	17.29
APO (Pryzant et al., 2023)	33.07	37.02	22.40	14.47	26.74
Ours					
RAPO - Base	27.07	17.13	14.72	23.64	20.64
RAPO - Rerank	43.75	33.23	19.57	**27.48**	31.01
RAPO - Base + Policy	41.47	29.44	20.77	-	30.56
RAPO - Rerank + Policy	**46.98**	**44.12**	**23.65**	-	**38.25**

4 Related Works

Automatic Prompt Engineering. To alleviate the intensive trial-and-error efforts in manual prompt engineering, the research community has developed various strategies to automate this process with techniques such as incremental editing (Prasad et al., 2023), reinforcement learning (Deng et al., 2022; Zhang et al., 2022), algorithmic search (Xu et al., 2022; Guo et al., 2023), among others. A notable line of work focuses on leveraging LLMs themselves for automatic prompt engineering (Honovich et al., 2023; Zhou et al., 2023; Pryzant et al., 2023; Yang et al., 2023; Guo et al., 2023). We differ in optimizing and constructing instance prompts tailored to individual examples.

Retrieval Augmented Generation (RAG). Retrieval-augmented Generation has been developed to address key limitations of Large Language Models (LLMs) such as hallucinations (Ji et al., 2023; Shuster et al., 2021) and factuality (Wang et al., 2023). Initial RAG implementations employed sparse retrievers such as BM25 (Robertson & Zaragoza, 2009) and TF-IDF (Wu et al., 2008) which, despite their effectiveness, struggled to grasp the nuanced semantic meanings in texts (Guo et al., 2022). This limitation led to the emergence of dense retrieval approaches, which use language model-based dense vector encodings to more accurately capture text semantics (Bruch et al., 2023; Karpukhin et al., 2020; Li et al., 2023). In addition, Glass et al. (2022). propose a retrieve and re-rank framework for combining advantages of sparse retrieval and dense retrieval.

More recent advancements have explored using LLMs as retrievers (Ma et al., 2023; Shen et al., 2023; Sun et al., 2023; Qin et al., 2023). Shen et al. (2023) demonstrates the potential of LLMs as zero-shot retrievers across multiple benchmarks. Ma et al. (2023) introduce a Listwise Reranker and Qin et al. (2023) propose a pairwise Reranker, both harnessing LLMs for improved reranking without requiring task-specific training. Sun et al. (2023) explored the use of generative LLMs like ChatGPT and GPT-4 and found properly instructed LLMs could rival or exceed the performance of leading supervised methods in popular information retrieval benchmarks. Their research also suggests the possibility of distilling ChatGPT's ranking capabilities into smaller, more efficient models for practical applications.

To the best of our knowledge, our work is the first to apply RAG in the domain of prompt optimization.

5 Conclusion

In this paper, we introduce Retrieval-Augmented Prompt Optimization (RAPO), a novel method in the realm of prompt optimization. Our method leverages a divide-and-conquer technique to enhance prompt efficacy by breaking down complex tasks into manageable subspaces. We showcased RAPO not only surpassing state-of-art APO by a significant margin, including a 7.4% relative AUCPR improvement on internal datasets that require domain knowledge and 13.0% relative AUCPR gain on the publicly available Sarcasm dataset (Abu Farha & Magdy, 2020), but also facilitating seamless integration with domain-specific knowledge to enhance prompt quality, particularly when the domain knowledge is either absent or diverges/contradicts the LLMs' internal knowledge.

REFERENCES

Ibrahim Abu Farha and Walid Magdy. From Arabic sentiment analysis to sarcasm detection: The ArSarcasm dataset. In Hend Al-Khalifa, Walid Magdy, Kareem Darwish, Tamer Elsayed, and Hamdy Mubarak (eds.), *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 32–39, Marseille, France, May 2020. European Language Resource Association.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Sebastian Bruch, Siyu Gai, and Amir Ingber. An analysis of fusion functions for hybrid retrieval. *ACM Trans. Inf. Syst.*, 42(1), aug 2023. ISSN 1046-8188. doi: 10.1145/3596512. URL https://doi.org/10.1145/3596512.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3369–3391, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2G: Retrieve, rerank, generate. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2701–2715, Seattle, United States, July 2022. Association for Computational Linguistics.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Trans. Inf. Syst.*, 40(4), mar 2022. ISSN 1046-8188. doi: 10.1145/3486250. URL https://doi.org/10.1145/3486250.

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

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1935–1952, Toronto, Canada, July 2023. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-Tau Yih. Dense passage retrieval for Open-Domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics.

Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Trans. Inf. Syst.*, 41(3), apr 2023. ISSN 1046-8188. doi: 10.1145/3570724. URL https://doi.org/10.1145/3570724.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*, 2023.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3845–3864, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

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

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*, 2023.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*, 2023.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-Tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as Re-Ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14918–14937, Singapore, December 2023. Association for Computational Linguistics.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval and Domain-Specificity. October 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst. Secur.*, 26(3):1–37, June 2008.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. GPS: Genetic prompt search for efficient Few-Shot learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8162–8171, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

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

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. Tempera: Test-time prompting via reinforcement learning. *arXiv preprint arXiv:2211.11890*, 2022.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=92gvk82DE-.

A INITIALIZATION PROMPT

We use the following prompt to initialize an instance prompt for each training example. This approach allows for the incorporation of prior domain knowledge by embedding it within the $\{domain\_knowledge\}$ section.

Role: You are a prompt designer, trying to find a prompt that can correctly classify a given example.

INITIAL PROMPT

{initial_prompt}

EXAMPLE

Here's an example:

{error_string}

HELPFUL INFO

{domain_knowledge}

TASKS

Step 1. Your objective is to articulate why the above example should be classified as '{ground_truth_label}' based on its content. Provide a detailed analysis, explicitly identifying the features or characteristics that contribute to this classification.
Step 2. Additionally, derive a generalized rule or guideline that can be universally applied to categorize similar examples. Avoid specifying exact words or phrases; instead, aim to formulate a broader statement that encapsulates the key criteria for classification. If extracting specific rules proves challenging, use your best judgment to offer a broader statement.

OUTPUT

Step 1.

B META PROMPT FOR INSTANCE PROMPT OPTIMIZATION

For the optimization of each instance prompt, we utilize a modified OPRO meta prompt, incorporating initialized local components p_l from the training data as "policies." This strategy introduces a constraint in the optimization process to mitigate the risk of exploiting spurious features in small-sized groups.

Given a list of policies, I have some texts rephrased from it along with their corresponding scores. The texts are arranged in ascending order based on their scores, where higher scores indicate better quality.

Policies

{policies}

Texts with Scores

Here's an example:

{old_instructions_and_scores}

The following exemplars include a few examplars. They show how to apply your text: you replace <INS> in each input with your text, then read the input and give a output. We say your output is wrong if your output is different from the given output, and we say your output is correct if they are the same. When replacing <INS> with an old piece of text above, we get wrong outputs on the following inputs.

{exemplars}

Task

Your task is to produce a new text by adhering to the policies. The next text should be different from the old ones with a highest possible score on the exemplars. Write the text in square brackets.

C RERANKING IMPLEMENTATION

We incorporate a pairwise ranking method using LLM as introduced in Qin et al. (2023). The prompt used for pairwise reranking is as follows:

Given a query {context}, which of the following rules is more relevant to classify the query?

Rule A: Rule: {critique_A}. Example: {example_A}
Rule B: Rule: {critique_B}. Example: {example_B}

Which rule (Rule A / Rule B) is more relevant to the query?

To aggregate the score globally, we adopt the AllPair implementation as detailed in Qin et al. (2023).

D DETAILS OF OPRO IMPLEMENTATION

We optimize each local component in the instance prompt library using the OPRO method. The OPRO setup is as follows: we set the default temperature to be 1.0 for optimizer LLMs to encourage the generation of diverse and creative instructions. In each optimization step for an instance prompt, the optimizer LLM generates 8 instruction candidates. The top-10 best candidates along with their respective training negated cross-entropy loss scores, are fed into the meta-prompt. We conduct 50 iterations for each local component optimization using the associated group of training data. In contrast, we run 200 iterations using the entire training dataset for the OPRO baseline experiment.

E SAMPLE PROMPTS OPTIMIZED BY OPRO AND RAPO

We present sample prompts optimized by OPRO and RAPO for the Sarcasm classification task below. As can be seen: the prompt optimized by OPRO provides a broad definition about the downstream task, whereas RAPO's retrieval-augmented instance prompt provides more specific instructions tailored to the individual examples.

Sample input tweet translated by GPT4:

> The world is moving towards reducing its regular and continuous reliance on coal, and Egypt wants to increase it! StopCoal

OPRO optimized prompt:

> Is the following text sarcasm? Answer with True or False.
>
> Text: {tweet}
>
> Instructions:
> - Sarcasm is a form of humor that is often used to make fun of someone or something. It is usually expressed through irony, in which someone says the opposite of what they mean. For example, if someone says "I love getting up at 5 am to go to work," they are probably being sarcastic.
>
> A tweet is considered sarcastic if it is mocking or making fun of someone or something. A positive tweet is not sarcasm.

RAPO's retrieval augmented prompt:

> Is the following text sarcasm? Answer with True or False.
> Text: {tweet}
>
> Instructions:
> 1. A serious statement about a current event is considered to not be sarcastic.
> 2. A tweet is likely to be sarcastic if it is unlikely that the person who wrote it actually believes what they are saying, and it is not a serious statement, or a statement of fact.
> 3. A statement of fact is not sarcastic.