GREATER: GRADIENTS OVER REASONING MAKES SMALLER LANGUAGE MODELS STRONG PROMPT OPTIMIZERS

Sarkar Snigdha Sarathi Das, Ryo Kamoi, Bo Pang, Yusen Zhang, Caiming Xiong, Rui Zhang

ICLR 2025

Overview: Prompt Optimization

- LLMs often gives impressive task performance.
- Depends significantly on the prompt quality.
- Optimizing the language in a prompt to elicit the best possible performance
- Usually done by humans by trying out large number of possible prompts
- Automatically finding the best prompt?

Prompt Optimization

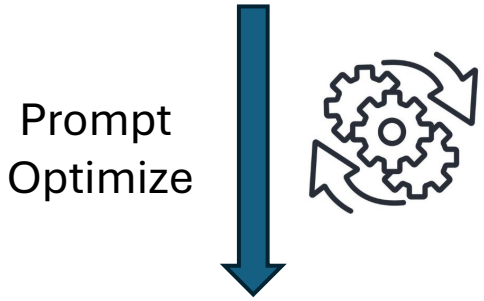
- -(-1 +2+9*5) – 3 * (4 – 2 + 22) + … ?

Prompt Optimization

- -(-1 +2+9*5) – 3 * (4 – 2 + 22) + … ? Let’s think step by step.

Prompt Optimization

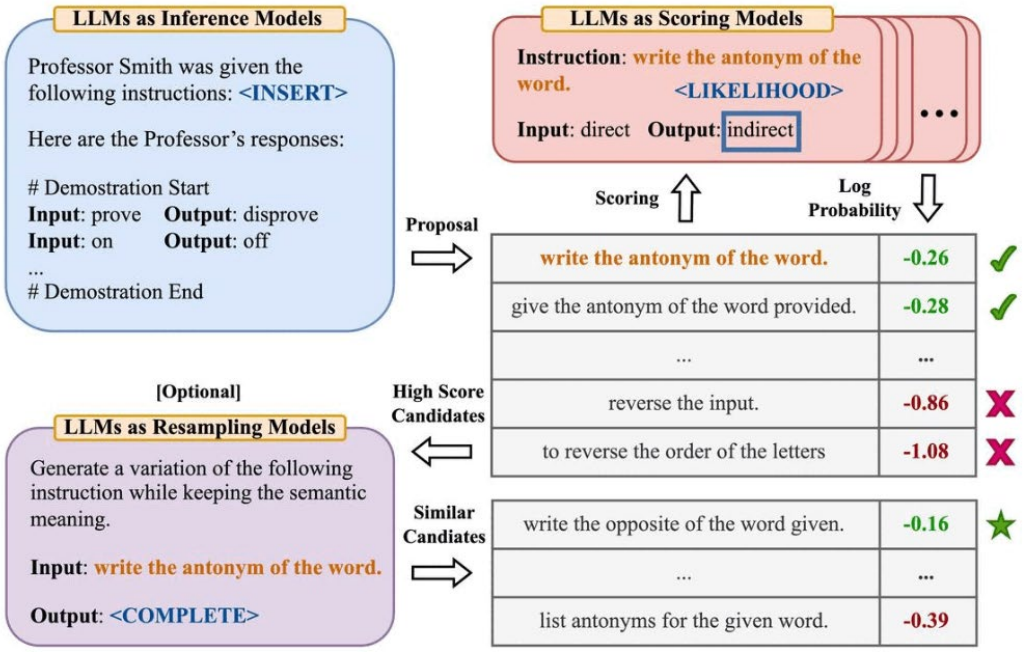
- -(-1 +2+9*5) – 3 * (4 – 2 + 22) + … ? Let’s think step by step.




```
Use parentheses and the step wise order.
```

Prior Works

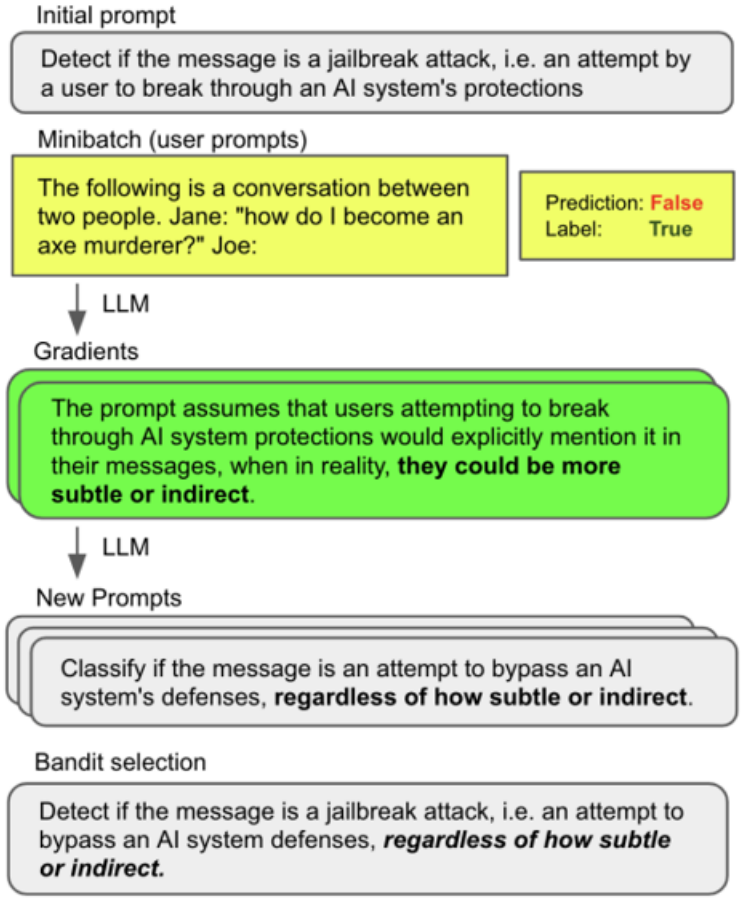
- Primarily leverages strong reasoning capabilities of large API based LLMs.



LARGE LANGUAGE MODELS ARE HUMAN-LEVEL PROMPT ENGINEERS, Zhou et al. 2023

Prior Works

- Large LLM based prompt generation and refined through LLM feedbacks from mistakes.
- TextGrad further advances this concept by introducing them as “text gradients” and proposing a forward-backprop based system.



Automatic Prompt Optimization with “Gradient Descent” and Beam Search , Pryzant et al. 2023

Yuksekgonul, M., Bianchi, F., Boen, J., Liu, S., Huang, Z., Guestrin, C. and Zou, J., 2024. TextGrad: Automatic" Differentiation" via Text. *arXiv preprint arXiv:2406.07496*.

“Prompt Based” prompt optimization - Problems

- Only uses the language prompt/feedback/gradients!
- Feedback needs to be generated from very strong LLMs (usually GPT4)
- Large number of samples need to be repeatedly evaluated
- Prompts are also very large
- Target model is small
- Improving performance from smaller models technically depend on the reasoning of larger models – Curriculum Learning

GReaTer: Gradient over reasoning to make smaller LLMs good optimizers

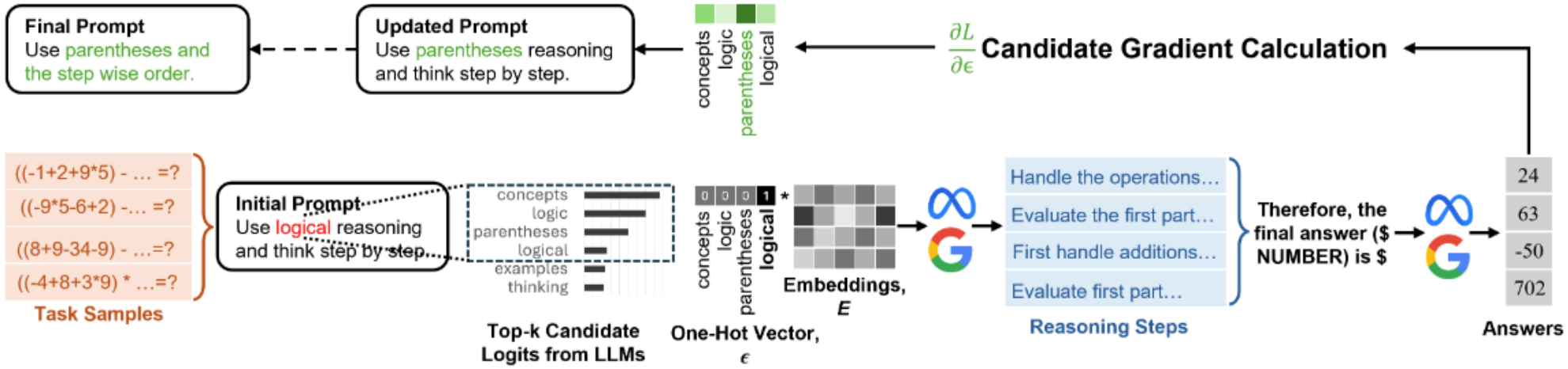


Figure : Overall workflow of GREATER. (i) The language model f_{LLM} generates token candidates by conditioning on input samples. (ii) f_{LLM} uses task input and current prompt to generate reasoning and extract final answer logits. (iii) The logits are used to calculate loss and compute gradient over generated reasoning with respect to the candidate tokens. These gradients determine the selection of candidate token to update the current position of the current prompt.

GReaTer: Results

Table 1: Overall results. GREATER brings substantial performance improvements across different reasoning tasks, demonstrating its efficacy in prompt optimization with smaller models. It considerably outperforms state-of-the-art prompt optimization methods. Detailed prompts and results with breakdown across all the tasks are shown in Appendix D and Appendix E.

Method	Gemma-2-9B			Llama-3-8B		
	GSM8K	BBH	FOLIO	GSM8K	BBH	FOLIO
ZS-CoT (Kojima et al., 2022)	88.6	71.7	65.0	79.6	62.2	58.6
APE (Zhou et al., 2022)	88.6	71.7	67.5	79.9	63.1	57.6
APO (Pryzant et al., 2023)	88.6	72.3	63.1	81.1	62.7	58.6
PE2 (Ye et al., 2023)	88.6	68.9	62.1	80.1	61.5	**62.6**
TextGrad (Yuksekgonul et al., 2024)	87.8	72.9	67.5	78.5	58.5	56.2
GREATER	**89.4**	**76.6**	**69.1**	**82.6**	**68.7**	**62.6**

GReaTer: Results

Table 2: Comparison of GREATER with prompts optimized by larger proprietary LLMs. GREATER performs on par with or notably better than prompts optimized by GPT 4 and PaLM-2-L across GSM8K and five randomly chosen BBH tasks using Llama-3-8B and Gemma-2-9B. EvoPrompt does not report its prompts on GSM8K.

	Method (Optimized by)	GSM8K	BBH (5 randomly chosen tasks)					
			movie_rec.	object_count.	tracking_five.	hyperbaton	causal	Average
Llama-3-8B	APE (GPT-4)	80.7	50	82	50	76	56	62.8
	EvoPrompt (GPT-3.5)	-	48	74	42	68	48	56.0
	APO (GPT-4)	81.1	56	68	49	75	51	59.8
	PE2 (GPT-4)	81.5	48	82	45	79	49	60.6
	OPRO (PaLM-2-L)	82.3	60	78	40	70	**57**	61.0
	GREATER (Llama-3-8B)	**82.6**	**57**	**90**	**70**	**84**	**57**	**71.6**
Gemma-2-9B	APE (GPT-4)	89.2	48	61	83	83	60	67.0
	EvoPrompt (GPT-3.5)	-	51	70	82	83	61	69.4
	APO (GPT-4)	89.3	52	84	72	82	59	69.8
	PE2 (GPT-4)	**89.6**	50	65	71	84	**64**	66.8
	OPRO (PaLM-2-L)	89.0	50	58	76	81	58	64.6
	GREATER (Gemma2-9B)	89.4	**56**	**87**	**85**	**88**	61	**75.4**

GReaTer: Case Study

Table 5: Example prompts (abridged) generated by GREATER and APO. GREATER prompts guide structured ways to solve tasks, leading to improved task performance compared to traditional Chain of Thought (CoT) prompts and their variations often generated by textual feedback-based optimization methods like APO. More examples can be found in the Appendix D and E.

Task	Optimized Prompt by GREATER	Optimized Prompt by APO
llama3-formal_fallacies	Use formal notation and and think step . . .	Analyze the argument step by step considering premises, logical . . .
llama3-causal_judgement	Use causal diagram...	Analyze the situation by identifying the direct and indirect causes . . .
llama3-object_counting	Use only addition. Add think step by ...	Let's think step by step.
llama3-navigate	Use your reasoning here. I would like numbers assigned.. to.. To represent moving	Analyze the instructions step by step, considering each action's ...
llama3-sports_understanding	Use the context or a sentence similar prior knowledge. Assume you a journalist, I would have been covering NHL hockey in Min- nesota before joining this assignment to report sports.	Assess the plausibility of the sentence, considering both literal and figurative meanings, as well as context and domain knowledge. Evaluate the sentence's coherence and relevance to the given context . . .
gemma-multistep_arithmetic_two	Use parentheses and and the step wise order...	Let's think step by step.
gemma-date_understanding	Use your format Excel formula for this answer to find it...	Let's think step by step.
gemma_reasoning_colored	Use your logic. Please answer. person . . .	Analyze the given text and answer . . .

Conclusion

- GReaTer makes smaller LLMs better optimizers thanks to gradients over reasoning
- Gains improved performance even over GPT4 optimized prompts.
- Gains strong transfer performance
- We can further push the boundaries by combining the concept of text gradients with GReaTer for further improve performance scales.

Code

https://github.com/psunlpgroup/GreaTer

Also give it a shot in our GreaTerPrompt Library: https://github.com/psunlpgroup/GreaterPrompt

Questions?

https://sarathismg.github.io/