Literature Review: LARGE LANGUAGE MODELS ARE HUMAN-LEVEL PROMPT ENGINEERS

Camille Challier | 311020 | camille.challier@epfl.ch
Chatbots-R-Us

1 Summary

This paper (Zhou et al., 2023) introduces Automatic Prompt Engineer (APE), a framework designed for the automated generation and selection of instructions or prompts. These prompts and instructions can be used to leverage Large Language Models (LLMs) for solving Natural Language Processing (NLP) tasks. The generation process is approached as natural language synthesis and treated as a black-box optimization problem. It employs LLMs to automatically create and select instructions, using an iterative Monte Carlo search method.

LLMs have demonstrated remarkable capabilities in various tasks, but their performance heavily depends on the quality of the prompts provided (Bsharat et al., 2024). Moreover, models do not seem to understand the instructions in the same way a human would (Webson and Pavlick, 2022). Effective prompt engineering can significantly enhance the ability of LLMs to produce accurate and relevant outputs, thereby improving their overall performance. Additionally, it eliminates the necessity for human users to experiment with a multitude of prompts in order to elicit desired behaviors, while not having full awareness of the compatibility of the instructions with the referenced model. Thus, the approach proposed in this paper is highly valuable, presenting a novel and intriguing idea.

The contribution of the research lies in the new formalism and theoretical introduction of the concept of Automatic Prompt Engineer. With "prompt engineering" defined as the practice of refining the wording within a prompt to elicit the optimal and desired response.

The authors also provide empirical insights into the effectiveness of APE-generated instructions by comparing and doing extensive qualitative and quantitative analyses of the performances of this method across diverse tasks. The authors compared APE against two baselines: human prompt engineers and a greedy version without a search and selection process.

Their achievement is that APE attains human-level performance on zero-shot learning with different datasets. It demonstrates the efficacy of APE applications in enhancing few-shot learning and discovering superior zero-shot chain of thought prompts. These findings underscore APE's potential in optimizing prompts to steer LLM behavior effectively, with implications for improving language understanding and task performance. Follow-up analyses delve into pertinent aspects of the main findings, examining prompt optimization effects and exploring trade-offs in the generated answers.

2 Strengths

This study makes several significant contributions to the field of Natural Language Processing, particularly in advancing our comprehension and utilization of prompt engineering.

The architecture of the APE framework is thoroughly justified and meticulously designed. The authors introduce an interesting and efficient method for evaluating the quality of instructions.

Initially, it's reasonable to anticipate that the method may incur significant computational intensity. However, through the implementation of adaptive filtering, which allocates more resources to high-quality candidates while minimizing costs for lower-quality ones, the approach efficiently optimizes the balance between accuracy and computational efficiency. This strategic allocation ensures that computational resources are utilized more judiciously, leading to improved performance without unnecessary overhead costs.

The experiments conducted in the study are well-designed. A lot of different tasks were tested, going from simple to more challenging ones as well. Indeed, the method was tested on diverse datasets, including MultiArith (Roy and Roth, 2015), Truth-

fulQA (Lin et al., 2022), Instruction Induction tasks (Honovich et al., 2022), and BIG-Bench Instruction Induction (BBII) tasks. This comprehensive experimental setup ensures a thorough evaluation of the APE method across various levels of difficulty, enhancing the robustness and reliability of the findings.

They employed metrics tailored to each task, such as Execution Accuracy, Truthfulness, and Informativeness. By considering the specific goals and demands of each task, these metrics provide a comprehensive evaluation of the capabilities of the framework, taking into account the overall performance.

The paper offers valuable insights and lots of annexed information that can aid researchers working in similar areas. The writing is clear and easily comprehensible, complemented by well-designed plots. Both the method and experiments demonstrate robustness, with the inclusion of multiple runs to estimate accuracy confidence. Finally, the authors have released their code, and the datasets are clearly presented, enhancing reproducibility.

Overall, the paper's suggestions correspond well with the paper's findings. The authors provide an insightful assessment of the method they introduced.

3 Weaknesses

While the paper delivers a valuable evaluation of the method, there are areas where further exploration could enhance its contribution. One of these is the lack of comparison with discrete instruction optimization techniques and other well-established prompt generation techniques. A comparative analysis with baselines such as AutoPrompt Gradient-based search (Shin et al., 2020), RLPrompt (Deng et al., 2022), GrIPS for gradient-free editing of instructions (Prasad et al., 2023), Prefix-Tuning (Li and Liang, 2021), P-tuning (Liu et al., 2023) or Prompt-Tuning (Lester et al., 2021) could have provided valuable context and insights. These frameworks have the potential to outperform instructions designed by humans, including the method under consideration.

In the study, completion-based models (such as GPT-3 and InstructGPT) and fill-in-the-blank models (like T5, GLM, InsertGPT) were assessed as prompt generators. In addition, InstructGPT served as the content generator. The exclusive focus on InstructGPT for generating output may limit the generalizability of the conclusions. Each language model possesses unique strengths and weaknesses. Over-reliance on GPT's performance could potentially bias the results and fail to provide an accurate representation of APE's capabilities. For instance, in some related papers, the developed technique was applied to GPT-2 and RoBERTa (Shin et al., 2020), or was examined with LaMDA and PaLM (Wang et al., 2023). Therefore, broadening the scope of the study to encompass a wider array of models could offer a more comprehensive understanding of language models' performance across various constraints.

Additionally, while experiments have shown that the proposed method demonstrates promising outcomes in tasks involving basic language comprehension, it occasionally underperforms compared to humans baseline in more challenging tasks. For instance, in tasks such as "Sentence Similarity", "Membership", "Second Letter" and "Passivization" in zero-shot scenarios across Instruction Induction experiments (as depicted in Figure 4 & Figure 7); and in various tasks of the BIG-Bench Instruction Induction, the method displays low performance. The authors' response to these limitations was limited, only stating that "APE achieves comparable or superior performance to the default human prompt on 17 out of 21 tasks" regarding the performance on tasks of the BIG-Bench Instruction Induction, without providing further elucidation or addressing potential solutions for improvement.

Addressing the choice of title; "LARGE LANGUAGE MODELS ARE HUMAN-LEVEL PROMPT ENGINEERS"; which may be perceived as pretentious, could also enhance the overall impact and reception of the study.

Finally, the paper didn't discuss ethical considerations related to LLMs. An extensive examination of potential discrimination, bias, and fairness would offer readers a comprehensive understanding of this important issue.

In conclusion, while this paper represents a significant contribution to the field, further refinements and expansions could lead to a more thorough and impactful investigation of the capabilities and challenges of this prompt generation method.

References

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled Instructions Are All

You Need for Questioning LLaMA-1/2, GPT-3.5/4. ArXiv:2312.16171 [cs].

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:2205.12548 [cs].

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2022. Instruction Induction: From Few Examples to Natural Language Task Descriptions. ArXiv:2205.10782 [cs].

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. ArXiv:2104.08691 [cs].

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. ArXiv:2101.00190 [cs].

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. ArXiv:2109.07958 [cs].

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. GPT Understands, Too. ArXiv:2103.10385 [cs].

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, Edit-based Instruction Search for Prompting Large Language Models. ArXiv:2203.07281 [cs].

Subhro Roy and Dan Roth. 2015. Solving General Arithmetic Word Problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. ArXiv:2010.15980 [cs].

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ArXiv:2203.11171 [cs].

Albert Webson and Ellie Pavlick. 2022. Do Prompt-Based Models Really Understand the Meaning of Their Prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large Language Models Are Human-Level Prompt Engineers. ArXiv:2211.01910 [cs].