Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Artificial Intelligence Center

Methods of Evolutionary Optimization of Prompts for Large Language Models

Vojtěch Klouda

Supervisor: Ing. Jan Drchal PhD.
Field of study: Artificial Intelligence
Subfield: Natural Language Processing
May 2025

Acknowledgements

=)))

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 10. May 2025

Abstract

TODO

Keywords: language model, evolutionary algorithm

Supervisor: Ing. Jan Drchal PhD.
Resslova 307/9 Praha, E-322

Abstrakt

TODO

Klíčová slova: jazykový model, evoluční algoritmus

Překlad názvu: Metody evoluční optimalizace vstupních řetězců pro velké jazykové modely

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Background

In recent years large language models (LLMs) emerged as a revolutionary force not only in the field natural language processing (NLP) but in everyday life of the general population $^{\text{cite smth}}$. The fundamental way to interact with these models is via text instructions or **prompts**. These prompts can be standalone or act as templates into which additional information, like a specific task or context, like from a database search, is inserted. This introduces the problem of **prompt engineering**, or finding the optimal prompt for a given situation. Prompt engineering emerges as a largely empirical field, with even experts experienced in the inner workings of LLMs relying on extensive trial-and-error bouts before finding a suitable prompt. With even experts struggling, it is significantly harder for non-experts to design an effective prompt. This motivates the recent research into automated prompt engineering. Several research directions have emerged, such as soft prompt tuning and discrete prompt optimization. Soft prompt tuning is effective but relies on access to inner states of the LLMs with are unavailable when using current state-of-the-art APIs such as the OpenAI API. Furthermore, prompts resulting from soft prompt tuning are often unreadable. In this work we focus on methods of discrete prompt optimization. This problem is very challenging because of the high dimesionality of the language space in which we conduct the search and non-differentiability as there are no gradients in the text space.

1.2 Objectives

The goal of this thesis is to compare several prompt optimization methods utilizing LLMs as the optimizing actors. That means, given a prompt p_t, where t denotes a step in the optimization process, the next prompt can be formulated as $p_{t+1} = \mathfrak{L}_{iter}(p_t)$, where $\mathfrak{L}_{iter}$ denotes an LLM initialized with specific instructions to facilitate the optimization mapping. The function $\mathfrak{L}_{iter}$ is an instance of a general LLM $\mathfrak{L}$ and

$$\mathfrak{L}_{iter}(p_t) = \mathfrak{L}(p_t|m), \tag{1.1}$$

where m denotes the set of instructions, or a prompt, for optimizing p_t. In context of prompt optimization when we are talking about a prompt for optimizing another prompts, we call it a **meta-prompt**. In this case, we can imagine m as

$$m \approx \text{"Criticise and improve the following prompt \{\}."}.$$

By changing the metaprompt, one can easily change the nature of the optimization operator. In the following chapters, we will examine options for designing such operators and compare their performance and effectivity on industry-standard datasets, as well as on a brand new custom dataset.

Chapter 2

Literature Review

2.1 Large Language Models

Since the Transformer architecture **Cite attention is all you need** brought us the first GPT and BERT models, LLMs have changed the NLP field. Now there are many powerful proprietary (OpenAI, Anthropic) and open-source (Llama, Qwen) models. **cite**

2.2 Prompts

Prompts are instructions we give to the LLM to get the output we want. It has been shown experimentally that small changes in prompt design, like simple paraphrasing, can influence the accuracy of the model's output. **cite** Naturally this inspired research into optimizing LLM performance by changing its prompts.

2.2.1 Prompting techniques

In solving challenging reasoning tasks, it is beneficial to get the model to show its reasoning steps. As LLMs just predict the next token, building on its previous thoughts improves performance on tasks like the GSM8k benchmark. This technique, called Chain-of-Thought has been built upon which resulted in methods such as Tree-of-Thought, ReAct and Reflexion. **cite prompt survey**

2.2.2 Prompt optimization methods

There has been work of optimizing soft and discrete prompts. As proprietary models usually do not allow access to its internal states, soft prompt optimization is not and option there. Several search and optimization methods have been employed in this task, like beam search and evolutionary algorithms. **sota rewrite here**

Appendix A

Bibliography

[1] W. Cui, J. Zhang, Z. Li, H. Sun, D. Lopez, K. Das, B. Malin, and S. Kumar, “Phaseevo: Towards unified in-context prompt optimization for large language models,” 2024.

[1]