Are Large Language Models Good Prompt Optimizers?

Submitted on **3 Feb 2024**

1 Introduction

这篇paper探究了LLM optimizer在reflection时很难找到error真正的原因，往往会被LLM之前的knowledge影响而不是真正地reflect error。因此提出了**ABO(Automatic Behavior Optimization)**这个框架

LLM-based Automatic Prompt Optimization有两种方法：**Resampling-based methods**、**Reflection-based methods**

2 Background

LLM-based Automatic Optimization的框架通常有三个部分：

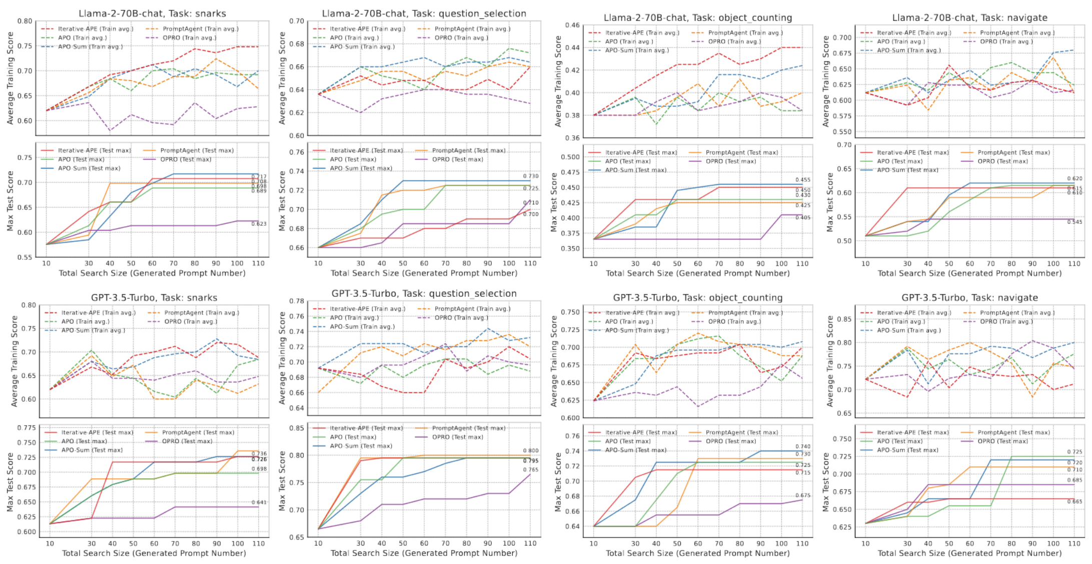
Prompt Initialization：可以是Manual Prompt Initialization，也可以是LLM-based Prompt Initialization
Search Strategy
Prompt Updating：分为 `Resampling-based Prompt Regeneration` (保持语义的同时在最优的prompt附近选取)、 `Explicit Reflection-based Prompt Optimization` (LLM-based prompt optimizer分析error然后产生feedback，之后refine current prompt)、 `Implicit Reflection-based Prompt Optimization` (根据历史prompt和对应的score，产生一系列refined prompt)

3 Basic Experiments

1 baseline：

`Iterative-APE` (resampling-based prompt regeneration method)
`APO` (explicit reflection-based method，每一步APO instruct LLM optimizer为现在的error生成三个reason，然后据此生成新的prompt)
`PromptAgent` (APO的一个扩展)
`APO-Sum` (APO的一个扩展，在reflection阶段总结reason)
`OPRO` (implicit reflection-based method)

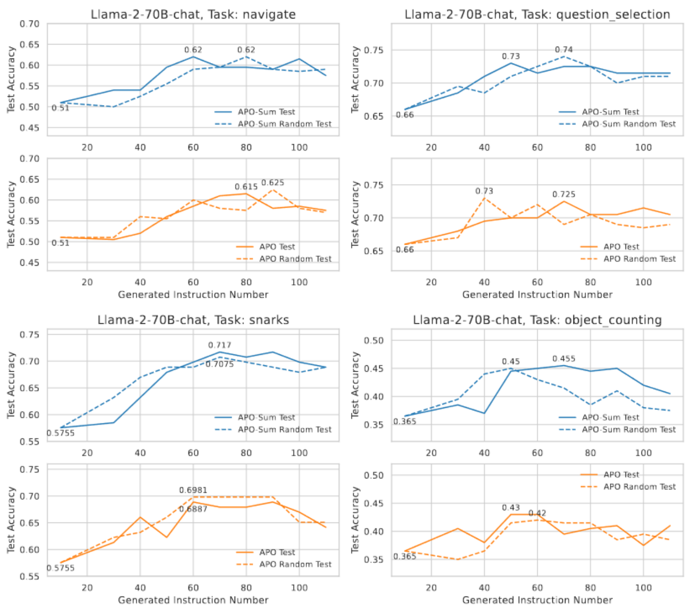
2 Result：



LLM optimizer可能不是一个好的implicit reflection-based prompt optimizer

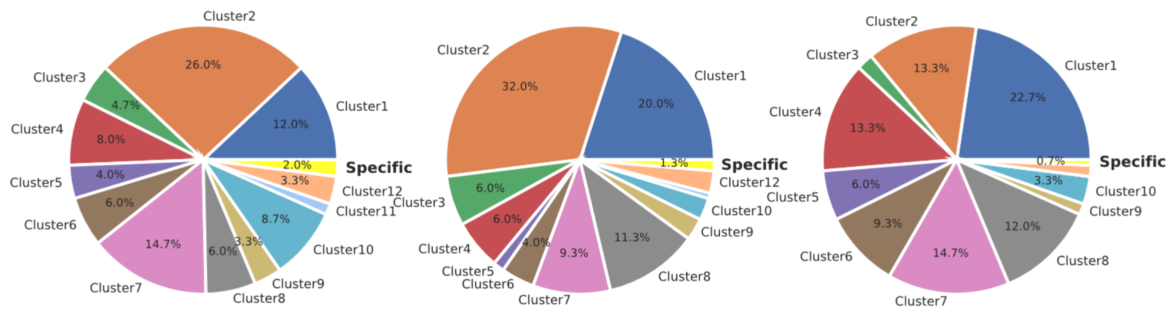
4 Did the LLM-based Prompt Optimizers Perform Valid Reflection?

Experiment1：两个setting，一个为 Vanilla (standard reflection setting)，另一个为 Pseudo (将real error examples换为pseudo error examples，生成方法为uniformly flip the LLM's predictions for all training examples)。结果为：



，可以看到最终的accuracy并无太大差距，说明无论是true error distribution还是pseudo error distribution，LLM optimizer都可以产生类似的feedback从而使得结果相差不大

Experiment2：探究了几种setting下feedback的分布问题，同时也探究了在Instance层面上的feedback重复问题

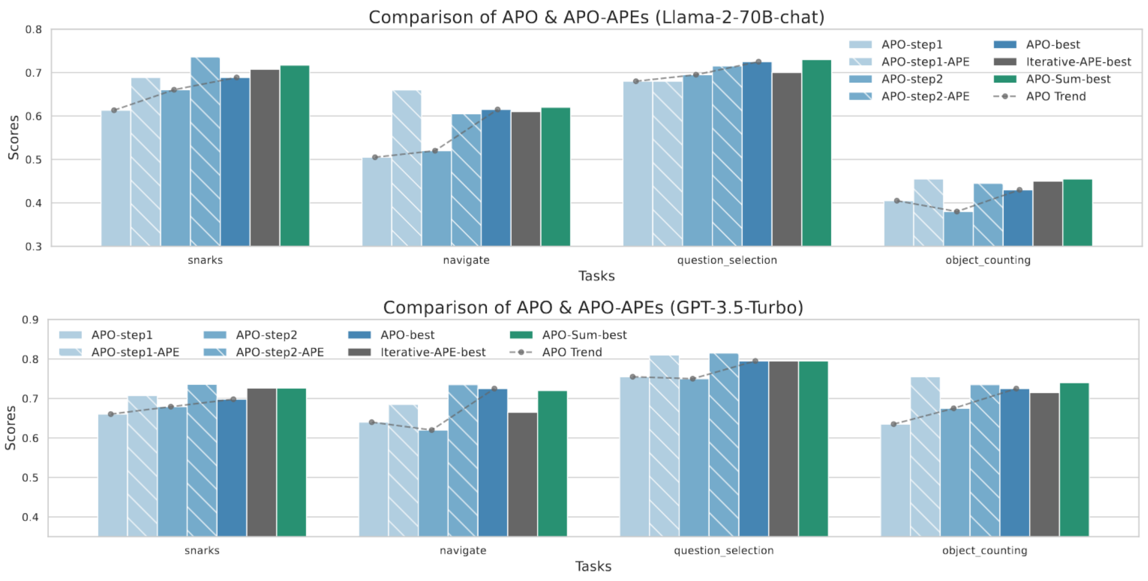


(a) Llama-2-70B-chat-Pseudo (b) Llama-2-70B-chat-Vanilla (c) GPT-3.5-turbo-Vanilla

LLM-based reflection过程更可能是LLM根据它prior knowledge进行分析，而不是根据真正的reason

5 How is the Quality of the Refined Prompts and How They Affect the Target Models' Behavior?

Experiment1： 从语义上来测试prompt，也就是将 iterative-APE 用到 APO 的第一步/第二步上



APO能够在某个reflection的过程中做出valid的semantic alternation，但并不总是valid

Experiment2： 探究了target model的instruction following的能力

6 Automatic Behavior Optimization

该框架的具体步骤：

1. At each step, the LLM optimizer is instructed to generate step-by-step prompts.

2. Next, we utilize the LLM optimizer to write an "Instruction-following Demonstration" for each prompt, i.e., an example illustrating how to strictly follow every detail of the given prompt. This practice aims to enhance the controllability of the prompt optimization process by ensuring any improvement made will be strictly followed by the target models.

3. During the reflection and prompt refinement process, given error examples, the LLM optimizer is required to identify the failure step of the target model and refine the prompt by further breaking down the solution at the problematic step. This aims to avoid invalid feedback by utilizing the LLM optimizer to perform more objective tasks during reflection.

4. For each refined prompt, the instruction-following demonstration is also updated to illustrate how to strictly follow the refined steps.