

20240918

A Systematic Survey of Prompt Engineering in Large Language
Models: Techniques and Applications

- 讀 prompt engineering 的 survey paper
- 挑選部分的測試案例和 RawData 來做比對，決定 prompt 的格式

Prompt engineering

- Zero-Shot Prompting
 - 在prompt中接收任務描述
 - 利用預先存在的知識並根據新任務的prompt產生預測
- Few-Shot Prompting
 - 提供了一些輸入輸出範例幫助模型理解任務
 - 需要額外的 token 來包含這些範例，可能對於長文本輸入造成負擔
 - 範例的品質可能會影響結果

Prompt engineering

- Chain-of-Thought (CoT) Prompting
 - 引導模型按照邏輯步驟逐步推理，而不是立即產生最終答案
 - 適合需要多步推理或邏輯推導的問題
- Automatic Chain-of-Thought (Auto-CoT) Prompting
 - 手動創建高質量的CoT範例既耗時且效果有限
→ “Let’s think step-by-step”
- Self-Consistency
 - 增強 CoT 的解碼策略
 - 通過比較不同推理路徑的最終答案，選擇最一致的答案來提高推理性能

Prompt engineering

- Logical Chain-of-Thought (LogiCoT) Prompting
 - 解決CoT缺乏驗證機制的問題
 - 使用反證法來驗證模型生成的推理步驟
 - think-verify-revise loop
 - 減少模型在推理過程中出現的邏輯錯誤和幻覺
- Graph-of-Thoughts (GoT) Prompting
 - 模擬人類非線性的思維過程
 - 允許將不同分支的推理過程進行聚合，形成一個更全面的解決方案

Prompt engineering

- System 2 Attention (S2A) Prompting
 - 解決Transformer-based LLM中的軟注意力機制容易合併不相關上下文訊息的問題
 - 重新生成輸入的上下文，過濾掉不相關的資訊
- Thread of Thought (ThoT) Prompting
 - 將大段的上下文分解成可管理的部分，逐步分析並總結每個片段，再將訊息精煉以生成最終回答
 - 適合在處理過於繁雜或沒有明確結構的文本