Prompt Engineering

Prompt Enginnering là quá trình tối ưu hóa cách đưa ra yêu cầu cho AI để xác nhận được kết quả chính xác và hiệu quả nhất. Nó quan trọng vì:

1. Tận dụng đối đa khả năng của AI
    - Mô hình AI có hiệu suất phụ thuộc vào cách người dùng đặt câu hỏi.Một prompt rõ ràng, chi tiết giúp AI hiểu đúng yêu cầu và đưa ra phản hồi chất lượng cao.
2. Tiết kiệm thời gian
    - Prompt tốt giảm số lần chỉnh sửa và tương tác lại với AI. Thay vì phải hỏi đi hỏi lại, người dùng nhận ngay kết quả mong muốn.
3. Giảm sai sót
    - Prompt mơ hồ sẽ dễ khiến AI hiểu lầm, dẫn đến kết quả sai hoặc không như ý. Prompt rõ ràng giảm rủi ro này.

Các yếu tố ảnh hưởng đến hiệu quả của Prompt

1. **Độ rõ ràng**
   Prompt càng cụ thể, càng ít rủi ro cho AI.
   Ví dụ : “”Kể tên 5 loài động vật có vú sống ở châu Phi” tốt hơn “Kể tên vài loài động vật”

2. **Kích thước mô hình AI**
   Mô hình lớn (GPT-4,Claude 3) xử lí prompt phức tạp tốt hơn mô hình nhỏ (GPT-3.5)

3. **Độ phức tạp của nhiệm vụ**
   Đối với các nhiệm vụ phức tạp, prompt cần cấu trúc rõ ràng với từ khóa chính xác.
   Ví dụ: “Giải thích định lí Pythagore bằng ngôn ngữ dễ hiểu, kèm ví dụ minh họa” sẽ tốt hơn “Nói về Toán học”.

4. **Ngữ cảnh ràng buộc**
   Thêm ngữ cảnh hoặc các ràng buộc giúp AI tập trung vào mục tiêu.

Prompt từ phía hệ thống (System prompt)

1. **Định nghĩa:** Là những hướng dẫn ẩn hoặc ràng buộc do developer thiết lập trước khi AI trả lời người dùng.

2. **Vai trò:**
   - Định hình hành vi mặc định của AI
   - Giới hạn phạm vi trả lời
   - Tăng tính An toàn

3. Prompt hệ thống của các Chatbot lớn như ChatGpt, Gemini,.. có prompt hệ thống được thiết kế rất linh hoạt để cân bằng giữa tính mở (trả lời đa dạng các chủ đề) và kiểm soát tính an toàn.
   Ví dụ prompt hệ thống ẩn của ChatGPT sẽ :
   - Đảm bảo AI tuân thủ nguyên tắc đạo đức.
   - Định hướng phong cách: Trung lập, hữu ích, tự nhiên.
   - Xử lí đa nhiệm : Đa dạng tác vụ.

4. **Ví dụ:**
   {"role": "system", "content": "Bạn là trợ lý ẩm thực. Chỉ trả lời về nấu ăn."}

Prompt người dùng (User prompt)

1. **Định nghĩa**: Là những yêu cầu trực tiếp mà người dùng nhập vào AI

2. **Vai trò**:
   Quyết định ngữ cảnh trực tiếp của câu hỏi
   Ảnh hưởng trực tiếp đến chất lượng câu trả lời

3. **Ví dụ:**
   {"role": "user", "content": "Cách làm bánh flan?"}

Sau đó tất cả các prompt đều được xử lí qua các bước token hóa, lớp embedding và attention.

Nếu như prompt của người dùng có xung đột với prompt của hệ thống thì AI sẽ từ chối trả lời.

Các cấu trúc Prompt

1. Zero-shot prompt

1.1. Khái niệm:

Zero-shot prompt là một kỹ thuật yêu cầu mô hình thực hiện tác vụ mà không cần cung cấp bất kì ví dụ nào.

1.2. Đặc điểm:

- Mô hình dựa vào kiến thức đã học từ dữ liệu huấn luyện để trả lời.
- Không cần training hay fine-tuning thêm.
- Phù hợp với các tác vụ đơn giản, không đòi hỏi nhiều ngữ cảnh

1.3. Cấu trúc:

[Yêu cầu/ nhiệm vụ] + [Thông tin đầu vào] + [Định dạng (nếu có)]

Ví dụ:

- Dịch thuật : ‘’Dịch câu sau sang tiếng anh: … ’’
- Phân loại văn bản : “Đoạn văn sau thuộc thể loại gì : … ”
- Tóm tắt văn bản : “Tóm tắt ngắn gọn đoạn văn sau : …”

1.4. Ưu và nhược điểm:

1.4.1. Ưu điểm:

- Không cần ví dụ mẫu, tiết kiệm thời gian.
- Dễ sử dụng, phù hợp với tác vụ đơn giản
- Hoạt động tốt với các mô hình lớn (GPT-4,Gemini)

1.4.2. Nhược điểm:

- Độ chính xác thấp hơn Few-shot/Chain-of-Thought
- Không hiệu quả với nhiệm vụ phức tạp, đòi hỏi ngữ cảnh
- Có thể gây hiểu nhầm nếu prompt mơ hồ

1.5. Khi nào nên dùng Zero-shot Prompt

- Khi nhiệm vụ đơn giản,không cần giải thích phức tạp
- Khi không có sẵn dữ liệu mẫu để làm Few-shot
- Khi muốn thử nhanh khả năng của mô hình

Ví dụ không phù hợp:

- Giải toán logic phức tạp
- Viết code dài

1.6. Kết luận

Zero-shot prompt là phương pháp đơn giản để tương tác với AI, nhưng hiệu quả phụ thuộc vào:

- Độ rõ ràng của prompt
- Khả năng của mô hình
- Độ phức tạp của tác vụ

2. Few-shot prompt

2.1. Khái niệm

- Few-shot Prompt là một kỹ thuật yêu cầu mô hình thực hiện tác vụ bằng cách cung cấp một vài ví dụ mẫu trước khi đưa ra yêu cầu chính
- Mục đích là giúp mô hình hiểu rõ hơn về định dạng, ngữ cảnh hoặc cách thực hiện nhiệm vụ.

2.2. Cấu trúc

[Ví dụ 1] + [Ví dụ 2] + … + [Yêu cầu chính]

Ví dụ:

Dịch thuật:

- "Hello" → "Xin chào"
- "Good morning" → "Chào buổi sáng"
- "Happy birthday" → ?

→Output mong đợi : "Chúc mừng sinh nhật"

Phân loại cảm xúc:

- "Tôi rất vui!" → "Tích cực"
- "Điều này thật tệ." → "Tiêu cực"
- "Chiếc bánh ngon quá." → ?

→Output mong đợi : "Tích cực"

2.3. Ưu và nhược điểm

2.3.1. Ưu điểm

- Hiệu quả hơn Zero-shot với các tác vụ phức tạp
- Giảm sai sót nhờ học từ ví dụ

2.3.2. Nhược điểm

- Tốn token (tăng chi phí)
- Quá nhiều ví dụ có thể gây nhiễu

2.4. Khi nào nên dùng Few-shot Prompt

- Dịch thuật (cần đúng cấu trúc)
- Phân loại văn bản
- Trả lời câu hỏi có format
- Tạo văn bản theo khuôn mẫu nhất định

Không nên dùng khi:

- Tác vụ quá đơn giản(dùng Zero-shot)
- Khi các ví dụ đưa ra không rõ ràng gây nhiễu thông tin

2.5. Kết luận

Few-shot Prompt giúp tăng độ chính xác của AI khi :

- Những ví dụ đưa ra chất lượng
- Nhiệm vụ cần ngữ cảnh để tránh hiểu lầm
- Mong muốn mô hình bắt chước cách làm của ví dụ đưa ra

3. Chain of thought(CoT)

3.1. Khái niệm:

Một phương pháp khuyến khích AI suy nghĩ trước khi đưa ra kết quả.

Mục đích:

- Cải thiện độ chính xác với các bài toán logic, toán học, hoặc suy luận phức tạp.
- Giúp người dùng hiểu quá trình AI suy nghĩ, dễ kiểm tra và sửa lỗi.

3.2. Cấu trúc

[Bài toán] + Hãy giải từng bước

Ví dụ:

- Toán học:
  - Prompt: "Nếu 3 quả táo giá 60k, hỏi 5 quả táo giá bao nhiêu? Hãy giải từng bước."
  - Output:
    - 1. Giá 1 quả táo = 60k / 3 = 20k.
    - 2. Giá 5 quả táo = 5 × 20k = 100k.
    - → Đáp án: 100k.
- Logic:
  - Prompt: "Anh ấy đến sân bay lúc 8h, chuyến bay cất cánh sau 2 tiếng. Nhưng bị hoãn 30 phút. Hỏi bay lúc mấy giờ? Giải thích từng bước."
  - Output:
    - 1. Thời gian dự kiến cất cánh: 8h + 2h = 10h.

- 2. Thời gian bị hoãn: 10h + 30 phút = 10h30.
- → Bay lúc 10h30.

3.3. Các biến thể của CoT

3.3.1. Self-Consistency CoT

Yêu cầu AI đưa ra nhiều cách giải → Chọn phương án xuất hiện

Ví dụ:

"Giải bài toán sau bằng 3 cách khác nhau: 48 ÷ (6 + 2) = ?"

3.3.2. Automatic CoT (Auto-CoT)

Tự động suy luận các bước thay vì viết tay

Ví dụ:

"Hãy tự chia nhỏ bài toán này và giải: Nếu 5 người làm xong công việc trong 10 ngày, 10 người làm trong mấy ngày?"

3.4. Khi nào nên dùng Chain of Thought

- Bài toán cần đến tính toán , logic
- Câu hỏi suy luận phức tạp
- Kiểm tra tính hợp lí

Sử dụng kết hợpvới Few-shot prompt để vừa kết hợp ngữ cảnh và vừa yêu cầu suy luận.

Nguồn:

1. https://tinhte.vn/thread/huong-dan-prompt-tu-co-ban-den-nang-cao-p1-zero-shot-va-few-shot-prompting.4011566/
2. https://tinhte.vn/thread/huong-dan-prompt-tu-co-ban-den-nang-cao-p3-step-back-prompting-va-chain-of-thought-cot.4012561/
3. https://chatgpt.com/
4. https://chat.deepseek.com/