
Các Thành Phần Cơ Bản Của Một Prompt Trong Kỹ Thuật Prompt Engineering

Video giới thiệu các thành phần quan trọng trong thiết kế prompt khi làm việc với các mô hình ngôn ngữ lớn (LLMs). Tác giả phân tích bốn thành phần chính bao gồm: hướng dẫn (instruction), ngữ cảnh (context), dữ liệu đầu vào (input data) và chỉ báo đầu ra (output indicator). Qua ví dụ phân loại cảm xúc văn bản, video minh họa cách các thành phần này hoạt động cùng nhau và cách tận dụng vai trò hệ thống (system role) và vai trò người dùng (user role) trong giao diện playground để tối ưu hóa prompt. Tác giả nhấn mạnh rằng không phải tất cả các thành phần đều cần thiết cho mọi trường hợp sử dụng, và việc thiết kế prompt hiệu quả đòi hỏi sự cân nhắc kỹ lưỡng về cách kết hợp các thành phần này dựa trên nhiệm vụ cụ thể.

Giới Thiệu Về Các Thành Phần Của Prompt

Khi thiết kế prompt cho các mô hình ngôn ngữ lớn, việc hiểu rõ các thành phần cấu tạo nên một prompt hiệu quả là vô cùng quan trọng. Prompt design đóng vai trò quyết định trong việc điều hướng các mô hình ngôn ngữ để đạt được kết quả tối ưu. Các thành phần của prompt không chỉ giúp mô hình hiểu nhiệm vụ mà còn giúp định hình cách mô hình phản hồi với đầu vào được cung cấp^[1].

Video đã trình bày một cách có hệ thống về các thành phần cơ bản trong thiết kế prompt, đồng thời minh họa chúng thông qua ví dụ thực tế về phân loại văn bản. Người xem không chỉ được giới thiệu về lý thuyết mà còn được hướng dẫn cách áp dụng các thành phần này trong môi trường thực tế như OpenAI playground. Thông qua việc hiểu rõ các thành phần này, người dùng có thể thiết kế prompt hiệu quả hơn và tận dụng tối đa khả năng của các mô hình ngôn ngữ lớn^[1].

Định Nghĩa Các Thành Phần Cơ Bản

Trong video, tác giả đã xác định bốn thành phần chính của một prompt. Mỗi thành phần đóng một vai trò riêng biệt nhưng cùng hoạt động để tạo nên một prompt hoàn chỉnh và hiệu quả:

1. **Instruction (Hướng dẫn):** Đây là phần chỉ dẫn cụ thể cho mô hình về nhiệm vụ cần thực hiện. Hướng dẫn càng rõ ràng và cụ thể, mô hình càng dễ hiểu và thực hiện nhiệm vụ chính xác hơn. Thành phần này thường được coi là bắt buộc vì nó định hướng cho mô hình về việc phải làm gì^[1].
2. **Context (Ngữ cảnh):** Phần này cung cấp thông tin bổ sung hoặc bối cảnh cho nhiệm vụ. Tác giả lưu ý rằng ngữ cảnh không phải lúc nào cũng cần thiết và phụ thuộc vào trường hợp sử dụng cụ thể. Ví dụ, trong nhiệm vụ phân loại văn bản đơn giản, ngữ cảnh có thể không cần thiết nếu mô hình đã được đào tạo đủ tốt để hiểu nhiệm vụ mà không cần thông tin bổ sung^[1].
3. **Input data (Dữ liệu đầu vào):** Là dữ liệu cụ thể mà mô hình sẽ xử lý. Cách bạn truyền dữ liệu vào mô hình phụ thuộc vào nhiệm vụ. Trong ví dụ phân loại văn bản, dữ liệu đầu vào là câu "I think the food was okay". Tác giả nhấn mạnh việc chỉ rõ loại dữ liệu (như "text", "tweet"...) có thể giúp mô hình hiểu rõ hơn về nhiệm vụ^[1].
4. **Output indicator (Chỉ báo đầu ra):** Chỉ định loại đầu ra mà bạn mong đợi từ mô hình. Trong ví dụ phân loại cảm xúc, "sentiment" được sử dụng làm chỉ báo đầu ra để mô hình biết nó cần phân loại cảm xúc của văn bản. Điều này giúp hướng dẫn mô hình tạo ra định dạng hoặc loại phản hồi cụ thể^[1].

Các thành phần này tạo nên khung cơ bản cho việc thiết kế prompt, nhưng việc sử dụng chúng linh hoạt tùy thuộc vào nhiệm vụ cụ thể và cách bạn muốn mô hình phản hồi^[1].

Ứng Dụng Các Thành Phần Trong Thiết Kế Prompt

Việc áp dụng hiệu quả các thành phần prompt đòi hỏi sự hiểu biết về cách chúng hoạt động cùng nhau và cách tận dụng các công cụ có sẵn. Trong video, tác giả đã minh họa cách các thành phần này được ứng dụng trong thực tế thông qua ví dụ và demo trực tiếp trên OpenAI playground.

Ví Dụ Về Phân Loại Văn Bản

Tác giả sử dụng một ví dụ phân loại cảm xúc để minh họa cách các thành phần prompt hoạt động. Trong ví dụ này, prompt đầy đủ bao gồm:

- Instruction: "Classify the sentiment of the text"
- Input data: "I think the food was okay"

- Output indicator: "Sentiment"

Khi chạy prompt này trên mô hình GPT-3.5 Turbo, kết quả trả về là "neutral", phù hợp với nội dung của văn bản đầu vào. Ví dụ này cho thấy cách một prompt đơn giản với các thành phần cơ bản có thể hướng dẫn mô hình thực hiện nhiệm vụ phân loại hiệu quả^[1].

Tận Dụng Vai Trò Hệ Thống Và Người Dùng

Một điểm quan trọng được nhấn mạnh trong video là cách tận dụng các vai trò khác nhau trong giao diện của các playground như OpenAI. Tác giả đã trình bày hai cách tiếp cận:

1. **Sử dụng system role:** Đặt toàn bộ prompt (bao gồm instruction, input data và output indicator) trong vai trò hệ thống. Cách này giúp định nghĩa hành vi tổng thể mà bạn mong đợi từ mô hình^[1].

VD: You are a sentiment analysis expert. Your task is to classify the sentiment of any given text into one of the following categories: Positive, Neutral, or Negative. Always provide a concise explanation for your classification.

2. **Kết hợp system role và user role:** Tách các thành phần, đặt instruction trong system role và input data trong user role. Cách tiếp cận này có thể đơn giản hóa prompt và tận dụng cách mô hình đã được đào tạo để xử lý các vai trò khác nhau^[1].

VD:

System Role (Vai trò hệ thống):

"Bạn là trợ lý dịch thuật chuyên nghiệp [1]. Nhiệm vụ của bạn là dịch văn bản từ tiếng Việt sang tiếng Anh một cách chính xác và giữ nguyên ngữ nghĩa."

User Role (Vai trò người dùng):

"Hãy dịch câu sau:

'Công nghệ AI đang thay đổi cách chúng ta làm việc và giao tiếp hàng ngày.'"

Tác giả đã thực hiện thử nghiệm và chỉ ra rằng cả hai cách đều có thể mang lại kết quả tương tự, nhưng việc tách các thành phần vào các vai trò khác nhau có thể làm cho prompt dễ quản lý hơn và có thể cải thiện độ tin cậy của mô hình trong một số trường hợp^[1].

Các Cân Nhắc Trong Thiết Kế Prompt

Thiết kế prompt không phải là một công thức cứng nhắc mà đòi hỏi sự linh hoạt và thích ứng với từng trường hợp cụ thể. Tác giả trong video đã chia sẻ một số cân nhắc quan trọng khi thiết kế prompt.

Tính Linh Hoạt Của Các Thành Phần

Một điểm quan trọng được nhấn mạnh là không phải tất cả các thành phần đều cần thiết cho mọi trường hợp sử dụng. Việc sử dụng thành phần nào phụ thuộc nhiều vào nhiệm vụ cụ thể:

- **Instruction** thường được coi là **cần thiết** vì nó chỉ định nhiệm vụ cho mô hình^[1].
- **Context có thể không cần thiết** trong các nhiệm vụ đơn giản hoặc khi mô hình đã có đủ kiến thức để thực hiện nhiệm vụ mà không cần thông tin bổ sung^[1].
- **Input data có thể không cần thiết** trong các nhiệm vụ tạo sinh như "tạo một email với giọng điệu chuyên nghiệp" nơi không có dữ liệu đầu vào cụ thể^[1].
- **Output indicator có thể được sử dụng hoặc không tùy thuộc** vào việc bạn có cần chỉ định định dạng cụ thể cho đầu ra hay không^[1].

Tính linh hoạt này cho phép thiết kế prompt phù hợp với yêu cầu cụ thể của từng nhiệm vụ, không bị giới hạn bởi một cấu trúc cứng nhắc.

Tối Ưu Hóa Prompt Cho Hiệu Suất Tốt Hơn

Để tối ưu hóa prompt và cải thiện hiệu suất của mô hình, tác giả đề xuất một số chiến lược:

1. **Thử nghiệm với các cấu trúc prompt khác nhau:** Thử nghiệm các cách khác nhau để tổ chức các thành phần prompt và xem cách nào mang lại kết quả tốt nhất cho nhiệm vụ cụ thể của bạn^[1].
2. **Đánh giá hiệu suất mô hình:** Trong một hệ thống mạnh mẽ, bạn cần thử nghiệm và đánh giá cách mô hình hoạt động trên nhiều ví dụ khác nhau để đảm bảo độ tin cậy^[1].

3. **Cụ thể hóa chỉ dẫn và đầu ra:** Càng cụ thể về nhiệm vụ và loại đầu ra bạn mong muốn, mô hình càng có khả năng đáp ứng chính xác yêu cầu của bạn^[1].
4. **Cân nhắc cấu trúc đầu ra:** Đôi khi bạn có thể muốn đầu ra ở một định dạng cụ thể như JSON hoặc một cấu trúc khác. Trong những trường hợp này, việc chỉ định rõ định dạng đầu ra có thể rất quan trọng^[1].

Những chiến lược này không chỉ giúp cải thiện hiệu suất của mô hình trong nhiệm vụ hiện tại mà còn giúp phát triển kỹ năng thiết kế prompt tốt hơn cho các nhiệm vụ trong tương lai.

Rút ra cấu trúc thực tế từ các Element trên

1. Role (Vai trò)

- Mục đích: Xác định vai trò của AI để định hướng phong cách và kiến thức trả lời.
- Ví dụ:
"Bạn là một giáo viên dạy văn có 10 năm kinh nghiệm. Hãy giải thích ý nghĩa của bài thơ 'Sóng' của Xuân Quỳnh."

2. Task (Nhiệm vụ)

- Mục đích: Nêu rõ yêu cầu cụ thể cần AI thực hiện.
- Ví dụ:
"Viết một email tiếp thị giới thiệu sản phẩm máy lọc nước mới, nhấn mạnh tính năng thân thiện với môi trường."

3. Context (Ngữ cảnh)

- Mục đích: Cung cấp thông tin nền để AI hiểu bối cảnh và giới hạn phạm vi trả lời.
- Ví dụ:
"Công ty bạn đang nhắm đến khách hàng ở nông thôn, thu nhập trung bình. Hãy viết bài quảng cáo phù hợp với đối tượng này."

4. Examples (Ví dụ)

- Mục đích: Đưa ra mẫu đầu vào/đầu ra để AI học theo format và phong cách.
- Ví dụ:
*"Ví dụ:

- Input: 'Dịch câu này sang tiếng Anh: Tôi yêu Hà Nội vì ẩm thực phong phú.'
 - Output: 'I love Hanoi because of its diverse cuisine.'
- Hãy dịch câu sau: 'Việt Nam có nhiều danh lam thắng cảnh đẹp.'**

5. Constraints (Ràng buộc)

- Mục đích: Đặt giới hạn về độ dài, ngôn ngữ, hoặc tiêu chí cụ thể.
- Ví dụ:
"Viết một bài thơ 4 câu về mùa hè, sử dụng thể thơ lục bát và từ ngữ đơn giản."

6. Format (Định dạng)

- Mục đích: Yêu cầu kết quả theo cấu trúc nhất định (danh sách, bảng, JSON, văn bản...).
- Ví dụ:
"Trình bày 5 lợi ích của việc tập thể dục dưới dạng gạch đầu dòng."

Công Thức Kết Hợp Các Element

Để prompt hiệu quả, kết hợp các yếu tố trên theo mẫu:

"Role + Task + Context + Examples + Constraints + Format"

- Ví dụ hoàn chỉnh:
"Bạn là một luật sư chuyên về hợp đồng lao động (Role). Hãy soạn một hợp đồng thử việc cho vị trí nhân viên kinh doanh (Task), áp dụng theo Bộ luật Lao động 2019 (Context). Lưu ý: Không bao gồm điều khoản bảo mật (Constraints). Trình bày dưới dạng các điều khoản được đánh số (Format)."

Kết Luận

Thiết kế prompt hiệu quả là một yếu tố quan trọng trong việc tận dụng tối đa tiềm năng của các mô hình ngôn ngữ lớn. Thông qua video này, chúng ta đã hiểu rõ hơn về bốn thành phần cơ bản của một prompt: instruction, context, input data, và output indicator.

Các thành phần này tạo nên khung cơ bản cho việc giao tiếp với mô hình ngôn ngữ, giúp định hướng mô hình thực hiện nhiệm vụ cụ thể một cách chính xác. Tuy nhiên, điều quan trọng là phải hiểu rằng không phải tất cả các thành phần đều cần thiết cho mọi trường hợp sử dụng. Việc sử dụng linh hoạt các thành phần này tùy thuộc vào nhiệm vụ cụ thể và kết quả mong muốn^[1].

Những kỹ năng thiết kế prompt không chỉ dừng lại ở việc hiểu các thành phần cơ bản mà còn cần thực hành và thử nghiệm để phát triển sự hiểu biết sâu sắc về cách các mô hình phản ứng với các loại prompt khác nhau. Theo tác giả, trong các video tiếp theo sẽ có thêm nhiều thảo luận về cách tận dụng tốt hơn các vai trò khác nhau và cách tối ưu hóa chỉ báo đầu vào và đầu ra để đạt được kết quả đáng tin cậy^[1].

Hiểu rõ và áp dụng hiệu quả các thành phần của prompt là bước đầu tiên hướng tới việc làm chủ kỹ thuật prompt engineering và tối ưu hóa việc sử dụng các mô hình ngôn ngữ lớn trong các ứng dụng thực tế.

**

1. <https://www.youtube.com/watch?v=kgBZhJnh-vk>