

SYSTEMATIZATION AND EVALUATION OF PROMPTING TECHNIQUES IN LARGE LANGUAGE MODELS

HỆ THỐNG HÓA VÀ ĐÁNH GIÁ CÁC KỸ THUẬT PROMPT TRONG MÔ HÌNH NGÔN NGỮ LỚN

Võ Hoàng Thành, Hoàng Văn Thành

Trường Đại học Quảng Bình

ABSTRACT: The advancement of Large Language Models (LLMs) has unlocked new possibilities in human-computer interaction. However, the quality of Artificial Intelligence (AI) outputs is directly contingent upon the quality of input prompts. This paper synthesizes and categorizes prompt engineering techniques, ranging from fundamental to advanced methodologies. By analyzing approaches such as Few-Shot prompting, Chain-of-Thought (CoT) reasoning, and the establishment of structured frameworks, this study aims to provide a standardized guideline for optimizing AI performance and accuracy in complex tasks.

Keywords: Large Language Models (LLMs), artificial Intelligence (AI), prompt engineering, frameworks...

TÓM TẮT: Sự phát triển của các mô hình Ngôn ngữ lớn (LLMs) đã mở ra những khả năng mới trong tương tác giữa người và máy. Tuy nhiên, chất lượng đầu ra của AI (Trí tuệ nhân tạo) phụ thuộc trực tiếp vào chất lượng của dữ liệu đầu vào (prompt). Bài viết này tổng hợp và phân loại các kỹ thuật câu lệnh (Prompt) từ cơ bản đến nâng cao. Thông qua việc phân tích các phương pháp như học qua vài ví dụ (Few-Shot), chuỗi suy luận (Chain-of-Thought) và việc thiết lập khung cấu trúc (Frameworks), nghiên cứu nhằm cung cấp một hướng dẫn chuẩn hóa giúp tối ưu hóa hiệu suất và tính chính xác của Trí tuệ nhân tạo trong các tác vụ phức tạp.

Từ khóa: Mô hình Ngôn ngữ lớn (LLMs), trí tuệ nhân tạo (AI), kỹ thuật câu lệnh (Prompt), khung cấu trúc (Frameworks)...

1. MỞ ĐẦU

1.1. Giới thiệu

Trong kiến trúc của các mô hình Ngôn ngữ lớn (LLMs), prompt không chỉ đơn thuần là câu hỏi, mà đóng vai trò như bộ ngữ cảnh và ràng buộc thiết yếu. Khi nhận một prompt thiếu cấu trúc, mô hình Ngôn ngữ lớn phải tự suy diễn ý định của người dùng bằng cách lấy mẫu ngẫu nhiên giữa hàng tỷ tham số xác suất. Điều này dễ dẫn đến “ảo giác” (hallucination) - mô hình tạo ra thông tin sai lệch nhưng văn phong có vẻ hợp lý, hoặc chỉ đưa ra những câu trả lời sáo rỗng, không chính xác.

Vì vậy, Kỹ thuật Prompt phát triển mạnh mẽ nhằm căn chỉnh đầu ra của AI để đạt được ý định mong muốn của chúng ta. Các kỹ thuật tối ưu hóa như Few-shot prompting (cung cấp ví dụ mẫu), Chain-of-Thought (yêu cầu suy luận từng bước) hay Role-playing (gán vai trò) giúp cung cấp một bản đồ tư duy rõ ràng. Việc này giúp thu hẹp không gian dự đoán của mô hình, kích hoạt chính xác các trọng số mạng nơ-ron liên quan, từ đó khai thác tối đa năng lực ẩn và nâng cao độ chính xác, logic của văn bản đầu ra.

Trong tương tác với AI, cấu trúc câu

lệnh (prompt) rất quan trọng để có được kết quả chính xác và phù hợp. Nhiều khung (framework) đã được đề xuất để xây dựng prompt một cách có hệ thống. Có thể kể đến các khung như RTF (Role, Task, Format), CARE (Context, Action, Result, Example), CREATE (Character, Request, Examples, Adjustment, Type, Extras) RACE (Role, Action, Context, Expectation), COAST (Context, Objective, Actions, Scenario, Task), CoT (Chain-of-Thought), ToT (Tree of Thoughts), CRISPE (Capacity, Role, Insight, Statement, Personality, Experiment), AIDA (Attention, Interest, Desire, Action), [1] v.v... Bài báo này giới thiệu các khung phổ biến, thường được khuyến dùng cho các công việc hàng ngày như RTF, CARE, CREATE. Nội dung bao gồm mục tiêu, thành phần chính và ví dụ minh họa cho mỗi khung, đồng thời so sánh cấu trúc các khung.

Nghiên cứu này nhằm hệ thống hóa và phân loại (taxonomy) các kỹ thuật và khung cấu trúc câu lệnh (Prompting Frameworks) hiện hành. Thay vì tiếp cận theo hướng thử-sai (trial-and-error), bài báo phân tích cơ chế tác động của các kỹ thuật này lên cấu trúc tự chú ý (self-attention mechanism) của mô hình Ngôn ngữ lớn, từ đó cung cấp một cơ sở phương pháp luận để tối ưu hóa đầu ra cho các tác vụ đặc thù.

Qua đó, xây dựng một hệ thống phân loại (taxonomy) rõ ràng cho các kỹ thuật prompt dựa trên nguyên lý hoạt động nội hàm của Ngôn ngữ lớn. Cung cấp khung đánh giá so sánh chi tiết về chi phí tính toán (token cost) và hiệu suất nhận thức (cognitive performance) giữa các khuôn mẫu (RTF, CARE, CREATE), giúp người sử dụng có cơ sở khoa học khi thiết kế hệ thống AI.

1.2. Các nguyên lý cơ sở

Để xây dựng một lệnh hiệu quả, thiết kế cơ sở cần tuân thủ 3 biến số độc lập nhằm giảm thiểu độ nhiễu của mô hình:

- Tính Tường minh (Clarity): Loại bỏ các từ ngữ đa nghĩa. Sử dụng động từ chỉ thị rõ ràng (ví dụ: phân loại, tổng hợp, trích xuất thay vì nói về).

- Bối cảnh hóa (Contextualization): Giới hạn không gian tìm kiếm thông tin của AI bằng cách cung cấp dữ liệu nền tảng (background information).

- Định dạng đầu ra (Output Formatting): Xác định rõ ràng cấu trúc dữ liệu trả về (JSON, bảng biểu, danh sách điểm chuẩn).

2. PHÂN LOẠI VÀ PHÂN TÍCH CÁC KỸ THUẬT CÂU LỆNH (PROMPT)

Để tối ưu hóa đầu ra của các mô hình Ngôn ngữ lớn, việc áp dụng các kỹ thuật câu lệnh cần dựa trên sự hiểu biết về cơ chế nhận thức nội hàm của mô hình. Trong nghiên cứu này, thay vì tiếp cận theo hướng thử-sai (trial-and-error), các kỹ thuật câu lệnh được hệ thống hóa và phân loại (taxonomy) thành ba nhóm chính dựa trên cách thức chúng can thiệp vào quá trình xử lý của mạng nơ-ron: Can thiệp bối cảnh, định hướng suy luận và ràng buộc ngữ nghĩa.

2.1. Nhóm kỹ thuật Can thiệp bối cảnh

Nhóm kỹ thuật này hoạt động dựa trên nguyên lý định hình lại không gian tìm kiếm ngữ nghĩa và cung cấp hệ quy chiếu thông qua Cửa sổ ngữ cảnh (Context Window), từ đó giúp mô hình khởi tạo trạng thái phù hợp trước khi thực thi tác vụ.

2.1.1. Kỹ thuật Cấp phát vai trò

Kỹ thuật Cấp phát vai trò (Persona Pattern) [2] yêu cầu mô hình Ngôn ngữ lớn chấp nhận một nhận dạng cụ thể trước khi

thực thi tác vụ. Về mặt toán học, việc cấp phát vai trò giúp mô hình thu hẹp trọng số phân phối từ vựng, hội tụ xác suất dự đoán từ tiếp theo vào một miền tri thức đặc thù thay vì lấy trung bình trên toàn bộ tập dữ liệu huấn luyện. Điều này đảm bảo văn bản đầu ra phù hợp với giọng điệu và mức độ chuyên môn được yêu cầu.

Cấu trúc: “Đóng vai một [Chuyên gia X] với [Y năm kinh nghiệm]. Bạn có nhiệm vụ [Z]”.

2.1.2. Kỹ thuật Học qua số ít ví dụ

Trái ngược với phương pháp Zero-shot (không cung cấp ví dụ), phương pháp Học qua số ít ví dụ (Few-shot) cung cấp cho mô hình một tập hợp nhỏ các cặp dữ liệu đầu vào - đầu ra (input-output) mẫu trong câu lệnh [3]. Phương pháp này khai thác cơ chế Học nội cảnh (In-context Learning) của kiến trúc Transformer, kích hoạt khả năng nhận diện quy luật (pattern recognition) của mô hình. Few-shot đặc biệt hiệu quả trong việc hướng dẫn AI tuân thủ các định dạng đặc thù hoặc phong cách phản hồi phi truyền thống mà không đòi hỏi chi phí tinh chỉnh trọng số (fine-tuning).

2.2. Nhóm kỹ thuật Định hướng suy luận

Đối với các bài toán phức tạp, việc chỉ cung cấp bối cảnh là không đủ. Nhóm kỹ thuật này can thiệp trực tiếp vào lộ trình tính toán của mô hình Ngôn ngữ lớn, ép buộc mô hình phân bổ ngân sách chú ý (attention budget) cho các bước logic trung gian nhằm tránh các kết luận vội vàng và sai lệch.

2.2.1. Kỹ thuật Chuỗi suy luận

Kỹ thuật “Chuỗi suy luận” (Chain-of-Thought - CoT) thực sự là một cuộc cách mạng trong việc bê khóa khả năng tư duy logic của các mô hình Ngôn ngữ lớn, được giới thiệu bởi Wei et al. (2022), [4] CoT yêu cầu mô hình phân rã quá trình giải quyết vấn

đề thành các bước logic trung gian trước khi đưa ra kết quả cuối cùng. Kỹ thuật này cải thiện đáng kể hiệu suất của AI trong các bài toán suy luận logic, toán học và lập trình.

Biến thể phổ biến nhất là Zero-shot CoT, được kích hoạt đơn giản bằng cách bổ sung tiền tố “Hãy suy nghĩ từng bước một” (Let's think step by step) vào cuối câu lệnh.

2.2.2. Kỹ thuật Phân rã tác vụ (Task Decomposition)

Đối với các truy vấn chứa lượng thông tin lớn và đa mục tiêu, sự phân bổ trọng số của cơ chế chú ý (attention mechanism) thường bị phân tán, dẫn đến hiện tượng suy giảm hiệu năng hoặc bỏ sót yêu cầu. Kỹ thuật Phân rã tác vụ giải quyết hạn chế này bằng cách chia nhỏ một quy trình phức tạp thành các chuỗi lệnh liên tiếp (chained prompts). Trong đó, đầu ra của lệnh trước được sử dụng làm đầu vào có điều kiện cho lệnh tiếp theo, giúp giảm tải nhận thức (cognitive load) cho mô hình tại một thời điểm tính toán cụ thể.

2.3. Nhóm kỹ thuật Ràng buộc ngữ nghĩa

Nhóm kỹ thuật này đóng vai trò như các hệ thống kiểm soát độc lập, giúp thu hẹp không gian sinh từ của AI bằng cách loại trừ các kết quả không mong muốn. Kỹ thuật đại diện cho nhóm này đó là kỹ thuật Ràng buộc tiêu cực (Negative Prompting)[5] ban đầu vốn nổi tiếng trong lĩnh vực AI tạo ảnh, từ cơ chế hướng dẫn không phân loại (Classifier-Free Guidance) trong các mô hình khuếch tán hình ảnh (Diffusion Models), kỹ thuật Ràng buộc tiêu cực hiện được ứng dụng rộng rãi vào mô hình Ngôn ngữ lớn nhằm thiết lập các biên giới an toàn (guardrails) cho văn bản đầu ra.

Thay vì chỉ định hướng các yêu cầu khẳng định, kỹ thuật này bổ sung các tham số loại trừ (Ví dụ: “Không sử dụng thuật

ngữ hàn lâm”, “Không vượt quá 500 từ”, hoặc “Bỏ qua dữ liệu trước năm 2020”). Việc thiết lập các ràng buộc tiêu cực giúp triệt tiêu xác suất sinh ra các token sai lệch, từ đó hạn chế hiệu quả hiện tượng ảo giác (hallucination) và sự dư thừa thông tin.

Bảng 1. Bảng phân loại các kỹ thuật câu lệnh

Nhóm phân loại	Mục tiêu can thiệp vào LLMs	Các kỹ thuật đại diện
1. Kỹ thuật Can thiệp Bối cảnh (<i>Contextual Intervention</i>)	Định hình lại không gian ngữ nghĩa và cung cấp hệ quy chiếu mẫu thông qua Cửa sổ ngữ cảnh (Context Window).	- Cấp phát vai trò (Persona Pattern) - Học qua số ít ví dụ (Few-shot Learning)
2. Kỹ thuật Định hướng Suy luận (<i>Reasoning-based Intervention</i>)	Can thiệp vào lộ trình tính toán, ép mô hình cấp phát thêm ngân sách chú ý (attention budget) cho các bước logic trung gian.	- Chuỗi suy luận (Chain-of-Thought - CoT) - Phân rã tác vụ (Task Decomposition)
3. Kỹ thuật Ràng buộc Ngữ nghĩa (<i>Constraint-based Intervention</i>)	Thiết lập hàng rào bảo vệ (Guardrails) bằng cách triệt tiêu xác suất sinh ra các token không mong muốn.	- Ràng buộc tiêu cực (Negative Prompting)

3. CÁC KHUNG CẤU TRÚC CÂU LỆNH PHỔ BIẾN (PROMPTING FRAMEWORKS)

Trong kỹ thuật câu lệnh (Prompt), việc dựa vào trực giác để viết câu lệnh thường dẫn đến sự thiếu hụt các tham số quan trọng (missing parameters), làm giảm độ chính xác của đầu ra. Các khung cấu trúc (Frameworks) ra đời như một phương pháp luận chuẩn hóa, giúp người dùng sắp xếp các biến số đầu vào một cách có hệ thống, từ đó tối ưu hóa cơ chế chú ý (attention mechanism) của Trí tuệ nhân tạo. Nó là sự lắp ráp của các Mẫu thiết kế (Prompt Patterns) đã được giới học thuật công nhận. Các khung cấu trúc câu lệnh như RTF, CARE hay CREATE bản chất là các mẫu thiết kế thuật toán (design patterns) nhằm chuẩn hóa các biến số đầu vào.

3.1. Khung RTF (Role - Task - Format):

Mô hình tối giản (Minimalist Model)

Khung RTF là mô hình nền tảng, tập trung vào ba biến số tối thiểu cần thiết để thiết lập một vòng lặp giao tiếp hiệu quả với mô hình Ngôn ngữ lớn.

R - Role (Vai trò): Ghi đè (override) bộ trọng số mặc định của mô hình bằng cách ép buộc nó truy xuất từ vựng và kiến thức của một nhân dạng cụ thể. Đây là Kỹ thuật thu hẹp không gian ngữ nghĩa (semantic search space). Về mặt toán học, mô hình Ngôn ngữ lớn dự đoán từ tiếp theo dựa trên xác suất có điều kiện. Việc đưa một “Nhân dạng” [2] vào đầu prompt sẽ làm thay đổi hoàn toàn phân phối xác suất này.

T - Task (Nhiệm vụ): Kích hoạt mạng nơ-ron chỉ thị. Task [6] không chỉ là việc chúng ta muốn làm gì, mà là việc chúng ta sử dụng đúng “Động từ” để kích hoạt các trọng số đã được huấn luyện riêng biệt của

AI. Động từ chỉ thị cốt lõi xác định phép toán mà mô hình cần thực hiện (ví dụ: tóm tắt, dịch thuật, tạo mã).

F - Format (Định dạng): Quy định cấu trúc liên kết của dữ liệu đầu ra.

Định dạng giúp ép buộc mô hình phải tuân thủ một cấu trúc liên kết nhất định, giảm thiểu tình trạng AI lan man tự sinh ra các câu văn chuyên tiếp không cần thiết. Kỹ thuật chỉ định định dạng được xếp vào nhóm Output Formatting [7]. Khi chúng ta

yêu cầu định dạng rõ ràng (Ví dụ: “Trả về định dạng JSON”, “Lập bảng 3 cột”, “Gạch đầu dòng”), AI sẽ ưu tiên phân bổ token cho các ký tự cấu trúc. Việc này thiết lập một “khuôn đúc” cứng, chặn đứng rủi ro Ảo giác (Hallucination) do AI tự vẽ vời thêm thông tin ngoài lề.

Ưu điểm: Tốc độ triển khai nhanh, tiêu tốn ít token đầu vào (input tokens). Thích hợp cho các tác vụ tuyến tính, đơn giản (ví dụ: viết email, định nghĩa thuật ngữ).

Bảng 2. Tóm tắt khung RTF

Thành phần	Câu hỏi cốt lõi	Ví dụ minh họa
R (Role)	AI cần tư duy như ai?	Đóng vai một chuyên gia...
T (Task)	AI cần giải quyết việc gì?	...hãy phân tích 3 điểm yếu trong đoạn văn bản sau...
F (Format)	Đầu ra trông như thế nào?	...và trình bày dưới dạng bảng so sánh (Lỗi / Đề xuất sửa).

3.2. Khung hỗ trợ: CARE (Context - Action - Result - Example)

Khung hỗ trợ (CARE) phù hợp với môi trường “định hướng mục tiêu” (Goal-oriented)[8] là một biến thể thường được ứng dụng mạnh mẽ trong môi trường doanh nghiệp (Corporate Setting).

C - Context (Bối cảnh): Cung cấp thông tin nền tảng về tình huống.

A - Action (Hành động): Hành động cụ thể cần AI thực hiện.

R - Result (Kết quả): Mục tiêu cuối cùng mà hành động này cần đạt được (ví dụ: “Để thuyết phục khách hàng gia hạn hợp đồng”).

E - Example (Ví dụ): Mẫu tham chiếu. Sự khác biệt lớn nhất của CARE nằm ở chữ R (Result) là yếu tố thay đổi. Khi bạn nói rõ kết quả mong muốn (Ví dụ: “Kết quả là bài viết này phải khiến khách hàng click vào link đăng ký”), AI sẽ tự động điều chỉnh thuật toán phân bổ từ vựng, áp dụng các kỹ thuật thuyết phục tâm lý (như AIDA hay FOMO) vào trong câu trả lời mà bạn không cần phải ra lệnh chi tiết.. Bằng cách gieo “Mục tiêu” vào đầu prompt, chúng ta đang định hướng toàn bộ mạng nơ-ron của nó hội tụ về một đích đến duy nhất mang lại giá trị kinh tế.

Bảng 3. Minh họa về bản thiết kế Prompt dựa theo khung CARE

Thành phần	Ý nghĩa	Ví dụ minh họa
C - Context (Bối cảnh):	Tình huống, vấn đề hoặc thử thách mà bạn đã gặp phải là gì?	... do quá tải nên công ty giao hàng trễ cho một khách hàng VIP mất 3 ngày. Khách hàng này đã rất tức giận và dọa sẽ viết bài phản nản lên mạng xã hội.
A - Action (Hành động):	Bạn đã đưa ra giải pháp gì và thực hiện những hành động cụ thể nào để giải quyết vấn đề?	Tôi ngay lập tức gọi điện trực tiếp để lắng nghe khách trút giận mà không ngắt lời. Sau đó, tôi chân thành xin lỗi, giải thích rõ lý do khách quan từ đơn vị vận chuyển. Tôi chủ động đề xuất ...
R - Result (Kết quả):	Kết quả cuối cùng đạt được là gì? (Nên định lượng bằng con số để tăng tính thuyết phục).	Khách hàng đã nguôi giận, đánh giá 5 sao cho thái độ phục vụ của bộ phận CSKH và tháng sau tiếp tục quay lại mua một đơn hàng lớn hơn.
E - Example (Ví dụ minh họa/Bài học áp dụng):	Nêu một điểm nhấn/ví dụ cụ thể trong quá trình đó để làm nổi bật kỹ năng của bạn, hoặc cách bạn đã áp dụng kinh nghiệm này vào một tình huống khác.	Ví dụ điển hình cho bài học rút ra từ sự cố này là: ngay tuần sau đó, khi nhận thấy một đơn hàng khác có nguy cơ trễ hạn, tôi đã áp dụng ngay quy trình “cảnh báo sớm”. Tôi gọi điện thông báo và tặng mã giảm giá cho khách <i>trước khi</i> họ phản nản, giúp công ty tránh được một cuộc khủng hoảng tương tự.

3.3. Khung CREATE: Mô hình toàn diện (Comprehensive Model)

Khung CREATE mở rộng từ RTF bằng cách tích hợp các kỹ thuật nâng cao như Few-Shot (Học qua ví dụ) và Negative Prompting (Ràng buộc tiêu cực). Đây là cấu trúc phù hợp cho các bài toán phi tuyến tính hoặc đòi hỏi độ chính xác tuyệt đối[7, 9].

C - Character (Vai trò): Tương đương với tham số Role trong RTF nhưng yêu cầu

định nghĩa sâu hơn về bối cảnh (ví dụ: “Chuyên gia dữ liệu tài chính với phong cách phân tích thận trọng”).

R - Request (Yêu cầu): Hệ thống hóa nhiệm vụ chính và các nhiệm vụ phụ theo trình tự logic.

E - Examples (Ví dụ mẫu): Cung cấp các cặp dữ liệu (Input-Output) làm tham chiếu.

Cơ sở khoa học: Kích hoạt khả năng

học nội cảnh (in-context learning) của mô hình, giúp AI sao chép chính xác cấu trúc logic mà không cần cập nhật tham số mạng nơ-ron.

A - Adjustment (Tinh chỉnh): Quy định các hệ số về sắc thái văn bản (Tone of voice) như mức độ trang trọng, tính thuyết phục, hoặc đối tượng độc giả mục tiêu.

T - Type of Output (Định dạng đầu ra): Đặc tả chi tiết cấu trúc (ví dụ: "Bảng markdown gồm 3 cột: STT, Tên khái niệm, Định nghĩa").

E - Extras (Các yếu tố bổ sung/Ràng

buộc): Đặt ra các giới hạn hành vi (guardrails). Ví dụ: "Không sử dụng thuật ngữ tiếng Anh chưa được Việt hóa", "Bỏ qua các dữ liệu trước năm 2020".

Khung CREATE là sự đóng gói (packaging) của những người thực hành AI. Nó lấy các khám phá học thuật phức tạp (như Few-shot learning, Persona adoption, Negative constraints) và biến chúng thành một danh sách kiểm tra (checklist) dễ nhớ, giúp người dùng không bỏ sót bất kỳ tham số nào khi giao việc cho AI.

Bảng 4. Minh họa về bản thiết kế Prompt dựa theo khung CREATE

Thành phần	Ví dụ minh họa
C - Character Vai trò	Đóng vai một Chuyên gia trải nghiệm khách hàng (CX) cấp cao ...
R - Request Yêu cầu logic	Bạn có nhiệm vụ... sau đó thực hiện 3 bước: - Phân loại cảm xúc... - Trích xuất ngắn gọn vấn đề... - Viết một câu phản hồi chuẩn mực...
E - Examples Học qua ví dụ (Few-Shot)	... ví dụ về cách tôi muốn bạn xử lý dữ liệu: Đầu vào: “App dạo này lag quá...” Đầu ra: > - Cảm xúc: Tiêu cực Vấn đề cốt lõi: Lỗi hệ thống... Câu phản hồi: “Dạ, ... Mong anh/chị thông cảm và...”
A - Adjustment Tinh chỉnh sắc thái	Giọng điệu của câu phản hồi phải luôn xưng là “Công ty” hoặc “Dạ”, gọi khách hàng là “anh/chị”. Nếu khách hàng giận dữ, phải ưu tiên lời xin lỗi. Nếu khách hàng khen ngợi, phải thể hiện sự biết ơn.
T - Type of Output Định dạng đầu ra	Trình bày kết quả dưới dạng một Bảng Markdown gồm 4 cột: [Đánh giá gốc] [Cảm xúc] [Vấn đề cốt lõi] [Câu phản hồi].
E - Extras Ràng buộc tiêu cực	Tuyệt đối không giải thích dài dòng... Chỉ in ra bảng kết quả. Không viết câu phản hồi dài quá 35 từ. Bỏ qua (không phân tích) những đánh giá chỉ chứa biểu tượng cảm xúc (emoji) mà không có văn bản.

3.4. Đánh giá khả năng ứng dụng

Để tối ưu hóa chi phí tính toán (token cost) và thời gian, việc lựa chọn Framework cần dựa trên mức độ phức tạp của tác vụ (Task Complexity).

Khi mà các công ty bắt đầu tính phí AI qua từng “token” (đơn vị từ vựng), việc lạm dụng một khung prompt khổng lồ cho một tác vụ vụn vặt không chỉ làm nghẽn băng thông, tăng độ trễ (latency) mà còn gây lãng phí tài nguyên.

Ngược lại, dùng một khung quá sơ sài cho một bài toán phức tạp sẽ khiến AI sinh ra “ảo giác” (hallucination) và thời gian bạn ngồi sửa lại kết quả (re-prompting) còn tốn kém hơn rất nhiều.

Việc đánh giá khả năng ứng dụng của các khung cấu trúc câu lệnh (Prompting Frameworks) cần được đối chiếu dựa trên ba tham số kỹ thuật cốt lõi: Độ phức tạp tác vụ (Task Complexity), Hiệu suất tiêu thụ token (Token Efficiency) và Tần suất hiệu chỉnh (Iteration Overhead). Sự lựa chọn khuôn mẫu phù hợp không chỉ quyết định độ chính xác của văn bản đầu ra mà còn ảnh hưởng trực tiếp đến chi phí tính toán và độ trễ suy luận (inference latency) của hệ thống.

Thứ nhất, đối với Khung tối giản RTF (Role - Task - Format), mô hình này thể hiện tính ưu việt trong các tác vụ có độ phức tạp tuyến tính và quy trình đơn bước. Do cấu trúc chỉ yêu cầu ba biến số nền tảng, RTF tối ưu hóa hiệu suất tiêu thụ token ở mức cao nhất, qua đó giảm thiểu tối đa độ trễ suy luận của các mô hình Ngôn ngữ lớn. Khung cấu trúc này đòi hỏi chi phí thiết lập câu lệnh (prompt engineering overhead) thấp, lý tưởng cho các hệ thống yêu cầu phản hồi theo thời gian thực (real-time) hoặc các tác vụ xử lý ngôn ngữ tự nhiên cơ

bản mà dung sai lỗi (error tolerance) có thể chấp nhận được ở mức độ nhất định.

Thứ hai, Khung hỗ trợ CARE (Context - Action - Result - Example) được ứng dụng hiệu quả trong các bài toán phụ thuộc bối cảnh (context-aware tasks) và mang tính định hướng mục tiêu (goal-oriented). Bằng cách bổ sung biến số mục tiêu/kết quả (Result) và ví dụ tham chiếu (Example), CARE cung cấp một hệ quy chiếu rõ ràng giúp mô hình Ngôn ngữ lớn tự động điều chỉnh thuật toán phân bổ trọng số từ vựng. Mặc dù làm tăng lượng token đầu vào so với RTF, khung CARE giúp giảm thiểu đáng kể tần suất hiệu chỉnh (iteration overhead) từ phía người sử dụng. Nguyên nhân là do mô hình đã được cung cấp đủ cơ sở ngữ nghĩa để tự tối ưu hóa hướng tiếp cận nhằm đạt được mục tiêu định trước.

Cuối cùng, Khung toàn diện CREATE được thiết kế đặc thù cho các tác vụ phi tuyến tính, đòi hỏi khả năng phân tích logic sâu và có dung sai lỗi tiệm cận mức không (zero error tolerance). Bằng việc tích hợp đồng thời kỹ thuật Học qua số ít ví dụ (Few-shot Learning) và Ràng buộc tiêu cực (Negative Prompting), CREATE tạo ra một hệ thống kiểm soát khép kín. Việc áp dụng khuôn mẫu này đặt ra tải trọng tính toán (computational load) lớn nhất lên Cửa sổ ngữ cảnh (Context Window) của mô hình. Tuy nhiên, sự đánh đổi về chi phí token và thời gian xử lý mang lại độ chính xác tối ưu, loại trừ sự cần thiết của các can thiệp thủ công ở khâu hậu kiểm (post-editing). Khung này đặc biệt phù hợp cho việc tự động hóa các quy trình thao tác chuẩn (Standard Operating Procedures) phức tạp trong môi trường doanh nghiệp và nghiên cứu.

Như vậy, việc ứng dụng các khung cấu

trúc câu lệnh cần được xem xét như một bài toán tối ưu hóa đa mục tiêu. Tùy thuộc vào bản chất dữ liệu đầu vào và các tiêu chuẩn khắt khe của đầu ra, việc linh hoạt chuyển đổi giữa RTF, CARE và CREATE là cơ sở phương pháp luận quan trọng để khai thác tối đa năng lực nhận thức của các mô hình AI tạo sinh hiện hành.

Bảng 5. Đánh giá và so sánh các Khung cấu trúc câu lệnh

Khung cấu trúc	Độ phức tạp tác vụ	Hiệu suất tiêu thụ Token	Tần suất hiệu chỉnh (Overhead)	Dung sai lỗi (Error Tolerance)	Khả năng ứng dụng đặc thù
RTF	Thấp (Tuyến tính)	Tối ưu (Chi phí thấp)	Cao (Cần kiểm tra lại)	Chấp nhận mức biên	Tác vụ đơn bước, yêu cầu phản hồi theo thời gian thực.
CARE	Trung bình (Định hướng)	Trung bình	Thấp	Khắt khe vừa phải	Tác vụ xử lý ngôn ngữ giao tiếp, định hướng mục tiêu doanh nghiệp.
CREATE	Cao (Phi tuyến tính)	Tiêu hao lớn (Chi phí cao)	Tiệm cận 0 (Zero-shot output)	Tiệm cận 0	Tác vụ phân tích logic phức tạp, tự động hóa quy trình SOP.

Không có một câu lệnh hoàn hảo duy nhất cho mọi tình huống. Các khung cấu trúc (Frameworks) hoạt động như các bản thiết kế kiến trúc (blueprints). Người dùng (câu lệnh) cần linh hoạt lựa chọn giữa RTF, CARE hoặc CREATE dựa trên mức độ nghiêm ngặt của đầu ra mong muốn và bản chất của dữ liệu đầu vào.

4. KẾT LUẬN

Kỹ thuật câu lệnh (Prompt Engineering) đã chuyển đổi từ việc đặt câu hỏi đơn thuần sang hình thức thiết kế thuật toán bằng ngôn ngữ tự nhiên. Trong hệ sinh thái mô hình Ngôn ngữ lớn (LLMs), một prompt tối ưu đóng vai trò là một “chương trình” hoàn chỉnh, bao gồm dữ liệu đầu vào,

các bước xử lý logic và các hàm ràng buộc định dạng đầu ra.

Thay vì dựa vào sự ngẫu hứng, việc áp dụng hệ thống các khung cấu trúc (như CREATE, RACE, RTF) kết hợp với các kỹ thuật module hóa như Role-playing, Few-Shot và Chain-of-Thought cho phép người dùng kiểm soát hành vi của AI một cách chính xác. Cách tiếp cận này mang lại ba giá trị cốt lõi: 1) Tính nhất quán: Đảm bảo đầu ra đáp ứng các tiêu chuẩn khắt khe trong môi trường chuyên nghiệp. 2) Hạn chế ảo giác: Thiết lập các hàng rào logic buộc AI bám sát dữ liệu thực tế. 3) Tính lặp lại: Tạo ra các quy trình làm việc tự động có chất lượng đồng đều, không phụ thuộc vào

biến số ngẫu nhiên của mô hình.

Việc làm chủ kỹ thuật prompt không chỉ là nâng cao hiệu suất, mà còn là kỹ năng then chốt để con người duy trì vai trò điều

phối, biến AI từ một công cụ trò chuyện thành một cộng sự giải quyết vấn đề phức tạp một cách có hệ thống.

TÀI LIỆU THAM KHẢO

- [1] Maab Saleem, 11 ChatGPT Prompt Frameworks Every Marketer Should Know, ButterCMS, . Xem tại: <https://buttercms.com/blog/chatgpt-prompt-frameworks/> Kiểm tra vào: (03/03/2026).
- [2] J. White et al. (2023), “A prompt pattern catalog to enhance prompt engineering with chatgpt”, arXiv preprint arXiv:2302.11382.
- [3] T. Brown et al. (2020), “Language models are few-shot learners”, Advances in neural information processing systems, vol. 33, pp. 1877-1901.
- [4] J. Wei et al. (2022), “Chain-of-thought prompting elicits reasoning in large language models”, Advances in neural information processing systems, vol. 35, pp. 24824-24837.
- [5] G. Sanchez, H. Fan, A. Spangher, E. Levi, P. S. Ammanamanchi, and S. Biderman (2023), “Stay on topic with classifier-free guidance”, arXiv preprint arXiv:2306.17806.
- [6] L. Ouyang et al. (2022), “Training language models to follow instructions with human feedback”, Advances in neural information processing systems, vol. 35, pp. 27730-27744.
- [7] S. Schulhoff et al. (2024), “The prompt report: A systematic survey of prompt engineering techniques”, arXiv preprint arXiv:2406.06608.
- [8] J. Dannhoferová and P. Jedlička (2024), Optimizing Prompts Through AI Frameworks: A Path to More Relevant Responses
- [9] S. M. Bsharat, A. Myrzakhan, and Z. Shen (2023), “Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4”, arXiv preprint arXiv:2312.16171.

Liên hệ:

ThS. Võ Hoàng Thành

Phòng Quản Trị, Trường Đại học Quảng Bình

Địa chỉ: 18 Nguyễn Văn Linh, phường Đồng Thuận, tỉnh Quảng Trị

Email: vohoangthanh@gmail.com

Ngày nhận bài: 27/02/2026

Ngày gửi phản biện: 27/02/2026

Ngày duyệt đăng: 15/3/2026