QUESTION ANSWERING WITH LARGE LANGUAGE MODELS APPLICATION

Hoang Duc Chung, Nguyen Hai Dang

Ngày 20 tháng 1 năm 2025

Abstract

Mục tiêu chính của nghiên cứu này là tối ưu hóa hiệu suất của các mô hình LLM thông qua việc thử nghiệm với nhiều kỹ thuật Prompting, bao gồm zero-shot, one-shot, few-shot và chain-of-thought. Kết quả được đánh giá bằng các chỉ số hiệu năng như Exact Match và F1 Score.

1 Giới thiệu

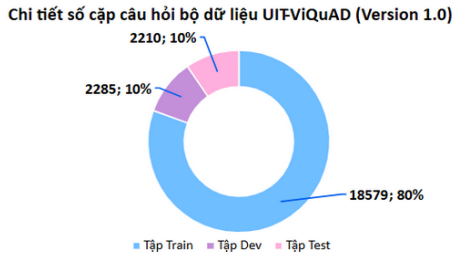
Trong bối cảnh công nghệ trí tuệ nhân tạo (AI) ngày càng phát triển, các mô hình ngôn ngữ lớn (Large Language Models - LLMs) đã chứng minh tiềm năng vượt trội trong nhiều ứng dụng, đặc biệt là xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Một trong những ứng dụng tiêu biểu của LLM là giải quyết các tác vụ trả lời câu hỏi dựa trên ngữ cảnh, giúp nâng cao khả năng truy vấn và xử lý thông tin tự động.

Bài toán trả lời câu hỏi trên văn bản tiếng Việt vẫn đang là một thách thức do các hạn chế về dữ liệu và sự phức tạp trong ngữ nghĩa tiếng Việt. Bộ dữ liệu UITViQuADv1 được xây dựng nhằm hỗ trợ nghiên cứu các mô hình NLP trên nền tảng ngôn ngữ tiếng Việt. Đề tài này hướng đến việc áp dụng các kỹ thuật Prompting với LLM để giải quyết bài toán trả lời câu hỏi tiếng Việt, tối ưu hóa hiệu năng mô hình và đánh giá khả năng ứng dụng thực tế thông qua việc xây dựng một nền tảng hỗ trợ người dùng.

2 Bộ dữ liệu

2.1 Tổng quan

UIT-ViQuAD (version 1.0) - A Vietnamese Dataset for Evaluating Machine Reading Comprehension - một bộ dữ liệu mới với hơn 23.000 cặp câu hỏi và trả lời do con người tạo ra, dựa trên 5.109 đoạn văn từ 174 bài viết tiếng Việt trên Wikipedia. Quá trình tạo dữ liệu được thiết kế đặc biệt để phù hợp với tiếng Việt, yêu cầu các khả năng suy luận phức tạp hơn như kết hợp câu và suy luận nhiều câu. Chi tiết quá trình thiết kế bộ dữ liệu được ghi rõ qua báo cáo khoa học "Kiet Van Nguyen, Duc-Vu Nguyen, Anh Gia-Tuan Nguyen, Ngan Luu-Thuy Nguyen. A Vietnamese Dataset for Evaluating Machine Reading Comprehension. COLING 2020".



2.2 Chú giải dữ liệu

Mỗi tập dữ liệu được phân theo các nhóm. Mỗi nhóm gồm 1 đoạn mô tả thông tin bất kỳ và các cặp câu hỏi liên quan đến đoạn mô tả đó. Chi tiết ở mỗi cặp câu hỏi gồm 1 câu hỏi kèm theo các câu trả lời đúng (vị trí bắt đầu câu trả lời, đoạn văn bản trả lời).

3 Kỹ thuật Prompting

3.1 Zero shot

Zero-shot là một phương pháp áp dụng trong các mô hình ngôn ngữ lớn, trong đó người dùng đưa ra lời nhắc (prompt) mà không cần cung cấp ví dụ cụ thể để mô hình hiểu và thực hiện nhiệm vụ. Phương pháp này dựa vào khả năng của mô hình trong việc suy luận và sử dụng kiến thức đã học để giải quyết các nhiệm vụ mà nó chưa được huấn luyện trực tiếp. Sử dụng thực tế:
"Read the following text and answer the question below. The answer must be within the paragraph and be one continuous phrase or keyword, not separated, without explanation or additional information outside the paragraph.
Văn bản:

<context>

Câu hỏi:

<question>

Trả lời."

3.2 One shot

One-shot là một phương pháp áp dụng trong các mô hình ngôn ngữ lớn, trong đó người dùng cung cấp một ví dụ duy nhất trong lời nhắc (prompt) để mô hình hiểu và thực hiện nhiệm vụ. Phương pháp này giúp mô hình tham khảo ví dụ mẫu, từ đó cải thiện khả năng suy luận và đưa ra kết quả chính xác hơn cho nhiệm vụ được yêu cầuu Sử dụng thực tế:
"Read the following text and answer the question below. The answer must be within the paragraph and be one continuous phrase or keyword, not separated, without explanation or additional information outside the paragraph.
Ví dụ:
Văn bản: "Albert Einstein là nhà vật lý nổi tiếng với thuyết tương đối."
Câu hỏi: "Ai là người phát triển thuyết tương đối?"
Trả lời: "Albert Einstein"

Văn bản:

<context>

Câu hỏi:

<question>

Trả lời."

3.3 Few shot

Few-shot là một phương pháp áp dụng trong các mô hình ngôn ngữ lớn, trong đó người dùng cung cấp một số ít ví dụ cụ thể trong lời nhắc (prompt) để mô hình hiểu và thực hiện nhiệm vụ. Phương pháp này giúp mô hình học từ các ví dụ mẫu, nâng cao khả năng suy luận và cải thiện độ chính xác khi giải quyết các nhiệm vụ yêu cầu. Sử dụng thực tế:
"Read the following text and answer the

question below. The answer must be within the paragraph and be one continuous phrase or keyword, not separated, without explanation or additional information outside the paragraph.
Ví dụ:
Văn bản: "Albert Einstein là nhà vật lý nổi tiếng với thuyết tương đối."
Câu hỏi: "Ai là người phát triển thuyết tương đối?"
Trả lời: "Albert Einstein"
Văn bản: "Kháng sinh được dùng để điều trị các bệnh nhiễm trùng do vi khuẩn gây ra."
Câu hỏi: "Kháng sinh dùng để điều trị cái gì?"
Trả lời: "nhiễm trùng do vi khuẩn" Văn bản: "Con người khám phá ra lửa có thể được tạo ra bằng cách đánh đá vào nhau vì nó tạo ra tia lửa."
Câu hỏi: "Tại sao lửa có thể được tạo ra khi đánh đá vào nhau?"
Trả lời: "tạo ra tia lửa"
Văn bản: "Việc xây dựng cây cầu mất 5 năm do địa hình phức tạp và thời tiết xấu."
Câu hỏi: "Việc xây dựng cây cầu mất bao lâu?"
Trả lời: "5 năm"
Văn bản:

<context>

Câu hỏi:

<question>

Trả lời."

3.4 Chain-of-thought

Chain-of-thought là một phương pháp áp dụng trong các mô hình ngôn ngữ lớn, trong đó người dùng đưa ra lời nhắc (prompt) kèm theo hướng dẫn suy luận từng bước để mô hình có thể hiểu và giải quyết nhiệm vụ một cách logic. Phương pháp này khai thác khả năng suy diễn của mô hình, giúp nó đưa ra các câu trả lời chính xác hơn bằng cách trình bày chi tiết quá trình tư duy trước khi kết luận. Sử dụng thực tế:
"Read the following context and answer the question below. Follow these steps to ensure accuracy:
1. Read and fully understand the provided context.
2. Identify the sentence or sentences in the context that may contain the answer to the question.
3. Analyze the sentence(s) to confirm the exact information that answers the question.
4. Provide the exact phrase or continuous keyword from the sentence(s) that answers the question, ensuring the answer is unbroken and directly addresses the question.
5. Ensure no additional explanation or words are included outside of the exact phrase.
Văn bản:

<context>

Câu hỏi:

<question>

Trả lời."

4 Các mô hình ngôn ngữ lớn (large language models)

4.1 Gemini 1.5 pro (Google)

Gemini 1.5 Pro là một phiên bản tiên tiến của dòng mô hình trí tuệ nhân tạo từ Google DeepMind, tập trung vào việc tối ưu hóa kiến trúc để nâng cao hiệu quả và hiệu suất. Mô hình này được xây dựng dựa trên kiến trúc Mixture-of-Experts (MoE), một cải tiến đột phá cho phép chỉ kích hoạt các thành phần cần thiết trong mạng khi xử lý dữ liệu. Điều này không chỉ giảm thiểu tài nguyên tiêu thụ mà còn tăng khả năng đáp ứng của mô hình trên các tác vụ phức tạp.

4.2 GPT 4.o mini (OpenAI)

GPT-4.o Mini là một phiên bản thu gọn của mô hình ngôn ngữ GPT-4.o, được thiết kế để duy trì hiệu suất cao trong khi giảm thiểu yêu cầu về tài nguyên tính toán. Mô hình này sử dụng kiến trúc Transformer cải tiến, cho phép xử lý hiệu quả các tác vụ ngôn ngữ tự nhiên phức tạp. Bằng cách tối ưu hóa số lượng tham số và cấu trúc lớp, GPT-4.o Mini đạt được sự cân bằng giữa khả năng xử lý mạnh mẽ và hiệu quả sử dụng tài nguyên, phù hợp cho các ứng dụng yêu cầu mô hình nhỏ gọn nhưng vẫn đảm bảo chất lượng đầu ra cao.

4.3 Llama 3.2-3b Instruct (Meta)

Llama 3.2-3B Instruct là một mô hình trí tuệ nhân tạo được phát triển bởi Meta, hướng tới việc cải thiện khả năng hiểu và đáp ứng yêu cầu từ người dùng thông qua các lệnh (instructions). Mô hình này sử dụng kiến trúc tiên tiến, tối ưu hóa việc xử lý các nhiệm vụ ngôn ngữ tự nhiên, nhưng trọng tâm chính là sự linh hoạt và hiệu quả trong việc điều chỉnh mô hình dựa trên các yêu cầu cụ thể.
Llama 3.2-3B Instruct được thiết kế để hoạt động với một lượng tham số vừa phải, giúp mô hình trở nên nhẹ nhàng nhưng vẫn duy trì hiệu suất cao. Nó đặc biệt phù hợp cho các ứng dụng yêu cầu xử lý lệnh với độ chính xác cao và có khả năng thích ứng linh hoạt với nhiều dạng yêu cầu khác nhau từ người dùng.

5 Finetune XLM-R-Base

XLM-RoBERTa là một phiên bản đa ngữ của RoBERTa, được huấn luyện trước trên 2,5TB dữ liệu CommonCrawl đã qua lọc, bao gồm 100 ngôn ngữ. RoBERTa là một mô hình dựa trên Transformer, được huấn luyện trên một tập dữ liệu lớn thông qua học tự giám sát. Phương pháp này sử dụng dữ liệu văn bản thô mà không cần sự gán nhãn của con người, dựa vào các quy trình tự động để tạo đầu vào và nhãn, từ đó tận dụng các tập dữ liệu công khai lớn.

Nhóm em sử dụng GPU P100 trên Kaggle để finetune xlm-r-base. Số epoch nhóm train là 25 epochs với batch sizes là 32 và một số hyper-para như lr là 2e-5, weight decay là 0.01. Chúng em sử dụng max length là 384 và doc stride là 128 để xử lý dữ liệu để tokenize. Kết quả tốt nhất nhóm em đạt được là ở epoch 5.

6 Độ đo

6.1 Exact Match

Exact Match (EM) dùng để đo độ chính xác của câu trả lời được sinh ra so với câu trả lời đúng. Cụ thể, EM đo lường tỷ lệ phần trăm các câu trả lời của mô hình khớp chính xác với câu trả lời trong bộ dữ liệu tham chiếu.

$$EM = \frac{\text{Số câu trả lời chính xác}}{\text{Tổng số câu trả lời}}$$

6.2 Precision

Precision được định nghĩa là tỷ lệ mẫu True Positive (TP) trong số các mẫu được dự đoán là Positive. Nó đánh giá khả năng của mô hình trong việc không gán nhầm nhãn Positive cho các mẫu không thuộc lớp đó. Precision phù hợp khi lỗi dương tính giả (False Positive) cần được hạn chế.
Trong bài toán này, Precision là tỷ lệ số từ đúng trong câu trả lời của mô hình so với tổng số từ trong câu trả lời mà mô hình tạo ra.

$$P = \frac{\text{Số từ đúng}}{\text{Số từ trong câu trả lời của mô hình}}$$

6.3 Recall

Recall đo lường tỷ lệ mẫu True Positive (TP) trong số các mẫu thực sự thuộc lớp Positive.

Nó phản ánh khả năng phát hiện đầy đủ các mẫu thuộc lớp mục tiêu của mô hình. Recall hữu ích trong các trường hợp cần hạn chế bỏ sót các mẫu dương tính (False Negative).
Trong bài toán này, Recall đo lường tỷ lệ số từ đúng trong câu trả lời của mô hình so với tổng số từ trong câu trả lời đúng.

$$Recall = \frac{\text{Số từ đúng}}{\text{Số từ trong câu trả lời đúng}}$$

6.4 F1 Score

F1-score là trung bình điều hòa của precision và recall, giúp cân bằng giữa hai độ đo này. F1-score hữu ích khi cần một chỉ số duy nhất để đánh giá mô hình, đặc biệt trong trường hợp dữ liệu không cân bằng.

$$\text{F1-score} = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

7 Kết quả thí nghiệm

Model	Prompt	Exact Match (EM)		F1 Score	
		Dev	Test	Dev	Test
xlm-r-finetune	x	50.02%	50.14%	75.34%	72.83%
gemini-1.5-pro	Zero_shot	63.81%	62.99%	86.07%	85.78%
	One_shot	65.82%	64.39%	87.08%	86.50%
	Few_shot	64.64%	62.90%	87.63%	86.83%
	Chain_of_Thought	70.42%	69.14%	89.63%	89.38%
gpt-4o-mini	Zero_shot	58.38%	57.74%	82.91%	83.30%
	One_shot	59.12%	59.05%	84.01%	83.33%
	Few_shot	58.86%	57.51%	84.14%	83.58%
	Chain_of_Thought	58.12%	58.82%	84.81%	84.78%
llama-3.2-3b-instruct	Zero_shot	46.52%	46.02%	74.60%	73.45%
	One_shot	43.59%	44.62%	71.21%	71.47%
	Few_shot	41.44%	40.23%	68.76%	68.57%
	Chain_of_Thought	33.04%	35.43%	66.34%	66.94%

Bảng 1: Bảng kết quả các mô hình.

Bảng so sánh cho thấy mô hình **gemini-1.5-pro** có hiệu suất vượt trội, đặc biệt khi sử dụng *Chain_of_Thought*, đạt EM cao nhất (70.42% trên Dev và 69.14% trên Test) cùng F1 Score gần 90%. Điều này chứng tỏ mô hình có khả năng tận dụng tốt các gợi ý phức tạp để cải thiện hiệu quả. Các kỹ thuật *One_shot* và *Few_shot* cũng mang lại kết quả cao, mặc dù thấp hơn so với *Chain_of_Thought*.

Mô hình **gpt-4o-mini** đạt hiệu suất trung bình, với EM và F1 Score ổn định quanh mức 58-59% và 84%, nhưng không cải thiện đáng kể với các loại gợi ý, thậm chí gợi ý *Chain_of_Thought* hoạt động kém nhất, cho thấy hạn chế trong khả năng xử lý các gợi ý phức tạp.

llama-3.2-3b-instruct có hiệu suất thấp nhất, đặc biệt với gợi ý *Chain_of_Thought*, khi EM chỉ đạt 33.04% trên Dev và 35.43% trên Test, phản ánh việc mô hình này chưa đủ mạnh để tận dụng gợi ý phức tạp. Kết quả với các gợi ý *Zero_shot* và *One_shot* tốt hơn một chút nhưng vẫn không vượt quá 46% EM.

Cuối cùng, **xlm-r-finetune** hoạt động ổn

định với EM khoảng 50% và F1 Score trên 72%

Nhìn chung, **gemini-1.5-pro** là mô hình tốt nhất trong bảng, đặc biệt vượt trội khi sử dụng gợi ý *Chain_ of_ Thought*, trong khi các mô hình còn lại cần cải thiện khả năng tận dụng các loại gợi ý để đạt hiệu suất cao hơn.

8 Kết luận

Nghiên cứu này đã đánh giá hiệu quả của các kỹ thuật Prompting (zero-shot, one-shot, few-shot, chain-of-thought) trong việc tối ưu hóa hiệu suất của các mô hình ngôn ngữ lớn (LLMs) khi trả lời câu hỏi trên văn bản tiếng Việt. Kết quả cho thấy mô hình Gemini 1.5 Pro đạt hiệu suất cao nhất, đặc biệt với kỹ thuật Chain-of-Thought, giúp cải thiện đáng kể các chỉ số EM và F1 Score. Các mô hình khác như GPT-4o Mini và XLM-R Finetune thể hiện hiệu quả ổn định nhưng không vượt trội, trong khi Llama 3.2-3B Instruct có hiệu suất kém hơn. Tổng quan, việc áp dụng các kỹ thuật Prompting nâng cao, đặc biệt là Chain-of-Thought, giúp cải thiện đáng kể hiệu quả xử lý câu hỏi phức tạp trong tiếng Việt.

9 References

[1] Nguyen, K. V., Nguyen, D.-V., Nguyen, A. G.-T., Nguyen, N. L.-T. "A Vietnamese dataset for evaluating machine reading comprehension", *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*. Available at: https://aclanthology.org/2020.coling-main.233/

[2] "Question-Answering using Bert" Available at: https://www.kaggle.com/code/arunmohan003/question-answering-using-bert

[3] "What is chain of thoughts (CoT)?" Available at: https://www.ibm.com/think/topics/chain-of-thoughts

[4] "Shot-Based Prompting" Available at: https://learnprompting.org/docs/basics/few_shot