ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

BÙI THỊ THANH PHƯƠNG
TRẦN TUẤN THÁI

# A QUESTION ANSWERING RAG SYSTEM IN EDUCATIONAL DOMAIN

## KHOÁ LUẬN TỐT NGHIỆP

Tp. Hồ Chí Minh - 2024

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**BÙI THỊ THANH PHƯƠNG**
**TRẦN TUẤN THÁI**

# A QUESTION ANSWERING RAG SYSTEM IN EDUCATIONAL DOMAIN

**KHOÁ LUẬN TỐT NGHIỆP**
CHƯƠNG TRÌNH CHÍNH QUY

NGƯỜI HƯỚNG DẪN KHOA HỌC
**TS. KHA TUẤN MINH**

**Tp. Hồ Chí Minh - 2024**

# Lời cảm ơn

Lời đầu tiên, chúng em xin phép gửi lời cảm ơn đến quý thầy cô trường Đại học Khoa học tự nhiên, ĐHQG-HCM nói chung và khoa Toán - Tin học nói riêng vì đã tận tình chỉ dạy và truyền đạt những kiến thức, kinh nghiệm quý báo cho chúng em cũng như tạo điều kiện cho chúng em thực hiện khóa luận tốt nghiệp, nhằm nâng cao khả năng nghiên cứu học thuật, tiếp cận tri thức chuyên môn phục vụ cho con đường sự nghiệp tương lai.

Đặc biệt, chúng em xin chân thành gửi lời cảm ơn sâu sắc đến TS. Kha Tuấn Minh và ThS. Lưu Trung Tín, những người hướng dẫn trực tiếp cho chúng em trong quá trình thực hiện đề tài. Các thầy đã rất tận tâm, nhiệt tình hỗ trợ chúng em vào những lúc chúng em bế tắc trong việc nghiên cứu và tiếp cận các vấn đề mang tính thử thách. Bên cạnh đó, các thầy còn là người truyền cảm hứng rất nhiều cho chúng em trong quá trình tìm hiểu và nghiên cứu về lĩnh vực này.

Cuối cùng, chúng em xin kính chúc quý thầy dồi dào sức khỏe, luôn bình an và hạnh phúc trong cuộc sống.

Chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 26 tháng 7 năm 2024

Nhóm sinh viên thực hiện

Trần Tuấn Thái

Bùi Thị Thanh Phương

# Mục lục

**Lời cảm ơn** **3**

**Bảng thuật ngữ** **7**

**Danh sách hình** **8**

**Danh sách bảng** **8**

**Lời nói đầu** **10**

**1 Giới thiệu** **11**

1.1 Giới thiệu về đề tài . . . . . 11

1.2 Mục tiêu đề tài . . . . . 12

1.3 Phạm vi của đề tài . . . . . 12

1.4 Cách tiếp cận dự kiến . . . . . 12

1.5 Kết quả dự kiến của đề tài . . . . . 13

1.6 Kế hoạch thực hiện: . . . . . 14

**2 Kiến thức nền tảng** **15**

2.1 Kỹ thuật Retrieval-Augmented Generation . . . . . 15

2.1.1 Ingestion . . . . . 16

2.1.2 Retrieval . . . . . 17

2.1.3 Response Generation . . . . . 17

2.2 Kỹ thuật Prompt Engineering . . . . . 18

2.3 Phương pháp Hypothetical Document Embeddings . . . . . 19
2.4 Mô hình ngôn ngữ . . . . . 20
2.4.1 Mô hình GPT-3.5 Turbo . . . . . 20
2.4.2 Mô hình Mixtral 8x7B . . . . . 20

**3 Phương pháp xây dựng hệ thống hỏi đáp 22**
3.1 Xây dựng bộ dữ liệu . . . . . 22
3.1.1 Nguồn dữ liệu và phạm vi thu thập . . . . . 22
3.1.2 Cấu trúc dữ liệu và thông tin chi tiết . . . . . 23
3.1.3 Hình thức lưu trữ và khả năng truy cập . . . . . 24
3.2 Thiết lập các tham số cho quá trình tạo sinh câu trả lời . . . 25
3.2.1 Chunk size . . . . . 25
3.2.2 Chunk overlap . . . . . 26
3.2.3 Top K . . . . . 26

**4 Thực nghiệm và đánh giá 28**
4.1 Những thách thức trong việc đánh giá hệ thống hỏi đáp . . . 28
4.2 Tập dữ liệu đánh giá . . . . . 29
4.3 Metric đánh giá . . . . . 30
4.3.1 Faithfulness . . . . . 31
4.3.2 Answer Relevancy . . . . . 31
4.3.3 Context Precision . . . . . 32
4.3.4 Context Recall . . . . . 32
4.3.5 Vận dụng . . . . . 33
4.4 Thí nghiệm đánh giá . . . . . 36
4.4.1 Thí nghiệm 1: Đánh giá hệ thống sử dụng RAG kết hợp GPT-3.5 Turbo . . . . . 36
4.4.2 Thí nghiệm 2: Đánh giá hệ thống sử dụng RAG kết hợp GPT-3.5 Turbo và Prompt Engineering . . . . . 39

4.4.3 Thí nghiệm 3: Đánh giá hệ thống sử dụng RAG kết hợp GPT-3.5 Turbo và HyDE . . . . . 41
4.4.4 Thí nghiệm 4: Đánh giá hệ thống sử dụng RAG kết hợp Mixral 8x7B và Prompt Engineering . . . . . 43
4.4.5 Thí nghiệm 5: Đánh giá hệ thống sử RAG kết hợp Mixtral 8x7B và HyDE . . . . . 45

**5 Tổng kết và hướng phát triển** **48**
5.1 Tổng kết . . . . . 48
5.2 Hướng phát triển . . . . . 50

**Tài liệu tham khảo** **51**

# Bảng thuật ngữ

| Tiếng anh | Viết tắt | Tiếng việt |
|---|---|---|
| Retrieval-Augmented Generation | RAG | Tạo sinh truy xuất tăng cường |
| Large Language Models | LLMs | Mô hình ngôn ngữ lớn |
| Ingestion | | Thu thập xử lí dữ liệu |
| Chunking | | Tách khối |
| Embeddings | | Nhúng |
| Prompt | | Lời nhắc |
| Pre-trained Language Models | PLMs | Mô hình ngôn ngữ được đào tạo trước |
| Bidirectional Encoder Representation from Transformer | BERT | Mô hình biểu diễn từ theo hai chiều |
| Sentence Transformers | | Mô hình biểu diễn câu |
| Hypothetical Document Embeddings | HyDE | Biểu diễn văn bản giả định |
| Reinforcement Learning from Human Feedback | RLHF | Học tăng cường từ phản hồi của con người |
| Grouped-Query Attention | GQA | Chú ý truy vấn theo nhóm |
| Parameter Sharing | | Chia sẻ tham số |
| Simple | | Đơn giản |
| Reasoning | | Suy luận |
| Multi-Context | | Đa ngữ cảnh |
| Context Precision | | Độ chính xác ngữ cảnh |
| Context Recall | | Khả năng truy xuất ngữ cảnh |
| Faithfulness | | Trung thành |
| Answer Relevancy | | Độ liên quan đến câu trả lời |
| Sentence Window Retrieval | | Truy xuất cửa sổ câu |
| Auto Merging Retriever | | Tự động tìm kiếm |

# Danh sách hình

2.1 Một RAG pipeline cơ bản . . . . . 15
2.2 Hypothetical Document Embeddings . . . . . 19

3.1 Dữ liệu thông tin của khoa Kinh tế, trường Đại học Kinh tế - Luật, ĐHQG-HCM . . . . . 25

4.1 Minh hoạ tập dữ liệu đánh giá . . . . . 29
4.2 So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo và đánh giá trực tiếp bởi con người . . . . . 36
4.3 So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo và Prompt Engineering . . . . . 39
4.4 So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo và Hypothetical Document Embeddings (HyDE) . . . . . 42
4.5 So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo với kỹ thuật RAG kết hợp Mixtral 8x7B và Prompt Engineering . . . . . 44
4.6 So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo với kỹ thuật RAG kết hợp Mixtral 8x7B và HyDE . . . . . 46

# Danh sách bảng

3.1 Bảng Kết quả đánh giá RAG dựa trên Top_k . . . . . 27

4.1 Phân loại câu hỏi . . . . . 30
4.2 Bảng đánh giá phân loại câu hỏi của RAG kết hợp GPT-3.5 Turbo . . . . . 38
4.3 Bảng đánh giá phân loại câu hỏi của RAG kết hợp GPT-3.5 Turbo và Prompt Engineering . . . . . 40
4.4 Bảng đánh giá phân loại câu hỏi của RAG kết hợp GPT-3.5 Turbo và HyDE . . . . . 42
4.5 Bảng đánh giá phân loại câu hỏi của RAG kết hợp Mixtral 8x7B và Prompt Engineering . . . . . 44
4.6 Bảng đánh giá phân loại câu hỏi của RAG kết hợp Mixtral 8x7B và HyDE . . . . . 46

5.1 Kết quả thí nghiệm với các mô hình và kỹ thuật khác nhau . 48

# Lời nói đầu

Sự phát triển mạnh mẽ của công nghệ thông tin trong những năm gần đây đã tác động sâu sắc đến mọi khía cạnh của đời sống xã hội, trong đó có lĩnh vực giáo dục. Việc ứng dụng Trí tuệ nhân tạo (AI) vào lĩnh vực giáo dục đang ngày càng phổ biến, mang đến nhiều giải pháp sáng tạo và hiệu quả để nâng cao chất lượng giáo dục. Một trong những ứng dụng tiềm năng đó là phát triển xây dựng hệ thống hỏi đáp thông tin các trường đại học.

Hệ thống hỏi đáp thông tin các trường đại học có thể giải quyết những hạn chế đang tồn đọng của quy trình tư vấn truyền thống như: tốn thời gian, thiếu hụt thông tin và khó khăn trong việc tiếp cận. Hệ thống sẽ cung cấp cho phụ huynh và học sinh những thông tin cần thiết một cách nhanh chóng, chính xác và dễ hiểu, từ đó giúp học sinh đưa ra lựa chọn phù hợp cho tương lai.

Trong khuôn khổ của khóa luận tốt nghiệp này tập trung vào việc thiết kế và phát triển một hệ thống hỏi đáp thông tin các trường đại học. Hệ thống sẽ sử dụng kỹ thuật *RAG (Retrieval-Augmented Generation)* để xử lý ngôn ngữ tự nhiên và truy xuất thông tin từ cơ sở dữ liệu các trường đại học, nhằm cung cấp cho học sinh các câu trả lời chính xác và đầy đủ cho các câu hỏi liên quan đến các trường đại học.

Nội dung luận văn bao gồm 5 chương:

**Chương 1:** Giới thiệu tổng quan về đề tài.

**Chương 2:** Trình bày kiến thức nền tảng của các kỹ thuật được sử dụng.

**Chương 3:** Mô tả bộ dữ liệu và mô hình ngôn ngữ được đề xuất.

**Chương 4:** Trình bày quá trình thực nghiệm và đánh giá kết quả.

**Chương 5:** Tổng kết và các hướng phát triển thêm của đề tài.

# Chương 1

# Giới thiệu

## 1.1 Giới thiệu về đề tài

Hiện nay, việc tư vấn thông tin các trường đại học đóng vai trò quan trọng trong việc định hướng tương lai cho học sinh. Tuy nhiên, quy trình tư vấn truyền thống thường gặp nhiều hạn chế như: tốn thời gian, thiếu hụt thông tin và khó khăn trong việc tiếp cận. Do đó, việc phát triển một hệ thống hỏi đáp thông tin các trường đại học hiệu quả là vô cùng cần thiết.

Một cách cụ thể, bài toán xây dựng hệ thống hỏi đáp thông tin các trường đại học được phát biểu như sau:

- Cho đầu vào là thông tin của các trường đại học (tên ngành, mã ngành, quy mô đào tạo, chương trình đào tạo,...).

- Yêu cầu: Xây dựng được một hệ thống có thể cung cấp cho người dùng các thông tin cần thiết về các trường đại học dựa trên thông tin truy vấn được cung cấp.

Khó khăn lớn của bài toán này là thông tin các trường đại học có thể không đủ (ví dụ thông tin các trường đại học sẽ thay đổi theo thời gian) để hệ thống có thể truy xuất và cung cấp các thông tin phù hợp cho người dùng.

Một hướng tiếp cận gần đây có thể giúp giải quyết khó khăn ở trên là sử dụng kỹ thuật RAG (Retrieval-Augmented Generation) và đây cũng là hướng tiếp cận mà đề tài tập trung tìm hiểu.

## 1.2 Mục tiêu đề tài

Trong quá trình thực hiện, đề tài hướng đến đạt được những mục tiêu cụ thể sau:

- Nắm được ý tưởng của các hướng tiếp cận đã được đề xuất để giải quyết bài toán xây dựng hệ thống hỏi đáp thông tin các trường đại học, từ đó chọn ra một kỹ thuật tốt để tập trung tìm hiểu sâu.
- Nắm rõ lý thuyết của kỹ thuật đã chọn.
- Cài đặt lại lý thuyết của kỹ thuật đã chọn để có thể đạt được các kết quả như trong tài liệu hướng dẫn, thực hiện thêm các thí nghiệm với dữ liệu của đề tài để thấy rõ ưu nhược điểm của kỹ thuật.
- Rèn luyện những kỹ năng nềm cần thiết khác: Kỹ năng làm việc nhóm, quản lý công việc, thuyết trình,...

## 1.3 Phạm vi của đề tài

Đề tài tìm hiểu và cài đặt lại kỹ thuật được đề xuất trong một tài liệu hướng dẫn. Ngoài ra, đề tài còn thực hiện thêm các thí nghiệm bên ngoài tài liệu để thấy rõ hơn về ưu và nhược điểm của kỹ thuật. Đề tài sử dụng bộ dữ liệu thông tin các trường đại học từ các đơn vị thành viên thuộc khối Đại học Quốc gia Thành phố Hồ Chí Minh. Nếu có đủ thời gian thì đề tài có thể mở rộng kỹ thuật với dữ liệu thông tin từ các trường đại học, cao đẳng, học viện trên địa bàn Thành phố Hồ Chí Minh.

## 1.4 Cách tiếp cận dự kiến

Để xây dựng hệ thống hỏi đáp thông tin các trường đại học, chúng ta có thể nghĩ đến các phương pháp truy vấn thông tin đơn giản dựa trên từ khóa. Tuy nhiên, các phương pháp này thường chỉ tập trung khai thác những từ khóa chính xác xuất hiện trong câu hỏi, bỏ qua yếu tố ngữ cảnh và ý nghĩa thực sự của người dùng. Do đó, kết quả trả lời thường không chính xác hoặc không đầy đủ, gây khó khăn cho người dùng trong việc tìm kiếm thông tin mong muốn.

Trong thời gian gần đây, kỹ thuật Retrieval-Augmented Generation (RAG) đã được áp dụng cho thấy những kết quả khả quan trong việc giải quyết các hạn chế của phương pháp truyền thống. Phương pháp này không chỉ đơn thuần tìm kiếm thông tin dựa trên từ khóa, mà còn kết hợp khả năng hiểu ngữ cảnh của câu hỏi và tạo ra câu trả lời tự nhiên, dễ hiểu như con người. Cụ thể, RAG hoạt động dựa trên hai công đoạn chính: *Retrieval* - tìm kiếm và lựa chọn thông tin liên quan từ cơ sở dữ liệu kiến thức dựa trên mức độ tương đồng ngữ nghĩa với câu hỏi và *Generation* - sử dụng mô hình ngôn ngữ sinh văn bản để tạo ra câu trả lời hoàn chỉnh và mạch lạc dựa trên thông tin đã được truy xuất.

Mặc dù kỹ thuật RAG đòi hỏi nguồn lực tính toán lớn hơn và quy trình huấn luyện phức tạp hơn so với các phương pháp truyền thống nhưng với khả năng cung cấp câu trả lời chất lượng cao, chính xác và dễ hiểu, chúng em tin rằng kỹ thuật RAG là một lựa chọn phù hợp để giải quyết bài toán xây dựng hệ thống hỏi đáp thông tin các trường đại học. Trong khuôn khổ của đề tài, với hạn chế về thời gian, nguồn lực và kiến thức, chúng em dự kiến sẽ tập trung nghiên cứu và triển khai kỹ thuật RAG dựa trên kết quả của tài liệu hướng dẫn để xây dựng hệ thống hỏi đáp hiệu quả và tối ưu nhất.

## 1.5 Kết quả dự kiến của đề tài

- Cài đặt lại từ đầu kỹ thuật được đề xuất trong tài liệu hướng dẫn.
- Có được các kết quả thí nghiệm cho thấy mã nguồn tự cài đặt cho các kết quả tương tự với tài liệu hướng dẫn gốc
- Có được các kết quả thí nghiệm với bộ dữ liệu của đề tài để thấy rõ hơn về ưu và nhược điểm của kỹ thuật.
- Nếu còn thời gian thì có thể cài đặt cải tiến phương pháp để khắc phục những hạn chế của kỹ thuật và có được các kết quả thí nghiệm tương ứng.

## 1.6 Kế hoạch thực hiện:

| Thời gian | Công việc thực hiện |
|---|---|
| 12/02 - 25/02/2024 | Tìm hiểu và lựa chọn nội dung đề tài khóa luận |
| 26/02 - 10/03/2024 | Lựa chọn bài báo, tài liệu hướng dẫn theo chủ đề đã chọn |
| 11/03 - 17/03/2024 | Đọc, hiểu nội dung chính của bài báo, tài liệu hướng dẫn |
| 18/03 - 24/03/2024 | Viết đề cương khóa luận |
| 25/03 - 31/03/2024 | Tìm hiểu các kiến thức cần thiết trong tài liệu hướng dẫn |
| 01/04 - 07/04/2024 | Thu thập và xử lý dữ liệu |
| 08/04 - 21/04/2024 | Tìm hiểu, chạy thử code theo kỹ thuật Retrieval-Augmented Generation (RAG) |
| 22/04 - 05/05/2024 | Cài đặt lại kỹ thuật Retrieval-Augmented Generation (RAG) |
| 06/05 - 02/06/2024 | Thực hiện các thí nghiệm và đánh giá |
| 03/06 - 23/06/2024 | Hoàn thành code, viết cuốn |
| 24/06 - 14/07/2024 | Chỉnh sửa cuốn, làm slide báo cáo |

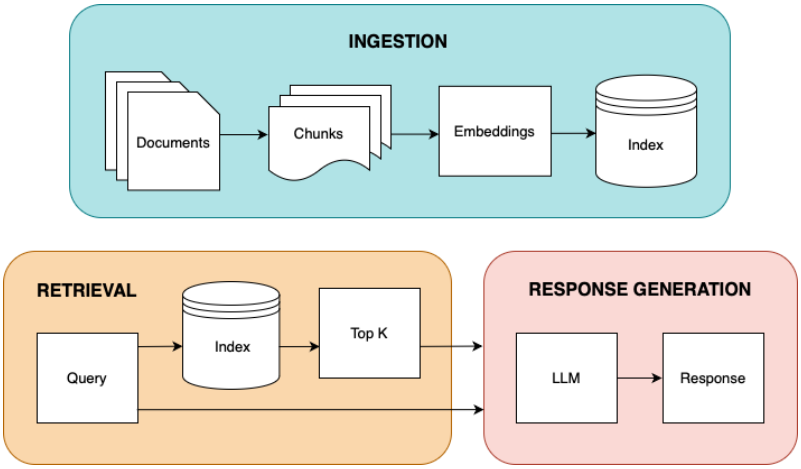
# Chương 2

# Kiến thức nền tảng

## 2.1 Kỹ thuật Retrieval-Augmented Generation

*Retrieval-Augmented Generation (RAG)* là một kỹ thuật kết hợp giữa việc truy xuất thông tin từ nguồn dữ liệu bên ngoài và việc tạo ra nội dung bằng mô hình ngôn ngữ lớn (*Large Language Models - LLMs*). RAG giúp cải thiện độ chính xác và đáng tin cậy của việc tạo ra nội dung, đặc biệt là trong các nhiệm vụ đòi hỏi kiến thức sâu và cụ thể. Bằng cách tích hợp thông tin từ cơ sở dữ liệu bên ngoài, RAG cho phép cập nhật kiến thức liên tục và kết hợp kiến thức bẩm sinh của các mô hình ngôn ngữ lớn với các nguồn dữ liệu đa dạng để tạo ra nội dung phong phú và chính xác hơn.

Một RAG pipeline bao gồm 3 thành phần chính: *Ingestion, Retrieval* và *Response Generation.*




Hình 2.1: Một RAG pipeline cơ bản

### 2.1.1 Ingestion

*Ingestion* là bước đầu tiên của kỹ thuật RAG, đóng vai trò quan trọng trong việc chuẩn bị dữ liệu đầu vào để phục vụ cho các bước tiếp theo. Bao gồm các quá trình:

**Chunking**

Quá trình này liên quan đến việc phân đoạn văn bản đầu vào thành các đơn vị ngắn gọn, có ý nghĩa, tạo điều kiện thuận lợi cho hệ thống truy xuất xác định chính xác các đoạn ngữ cảnh có liên quan để tạo phản hồi. Chất lượng và cấu trúc của các khối này đóng vai trò then chốt đối với hiệu quả của hệ thống, đảm bảo rằng các văn bản được truy xuất được điều chỉnh phù hợp với yêu cầu của người dùng. Các khối này có thể được xác định bằng một kích thước cố định, chẳng hạn như số lượng ký tự, câu hoặc đoạn văn cụ thể. Mỗi đoạn được mã hóa thành một vectơ nhúng để truy xuất. Các phần nhỏ hơn, chính xác hơn sẽ giúp kết quả phù hợp hơn giữa truy vấn của người dùng và nội dung, nâng cao độ chính xác và mức độ liên quan của thông tin được truy xuất. Các khối lớn hơn có thể bao gồm thông tin không liên quan, gây ra nhiễu và có khả năng làm giảm độ chính xác khi truy xuất.

Bằng cách kiểm soát kích thước khối, RAG có thể duy trì sự cân bằng giữa tính toàn diện và độ chính xác. Việc phân đoạn ảnh hưởng đáng kể đến chất lượng của nội dung được tạo ra, chẳng hạn như 100, 128, 512, 1024,... Các phát hiện cho thấy rằng kích thước khối lớn hơn có thể có lợi, nhưng lợi ích sẽ giảm dần sau một thời điểm nhất định, cho thấy rằng quá nhiều ngữ cảnh có thể gây ra nhiễu. Điều đáng chú ý là nhiều mô hình nhúng mã nguồn mở giới hạn ở mức 512 tokens.

**Embeddings**

Sau khi đã phân đoạn một cách thích hợp, bước tiếp theo là Embeddings. Việc tìm kiếm thông tin liên quan trong lượng dữ liệu khổng lồ đòi hỏi một phương pháp hiệu quả và nhanh chóng. Embedding dữ liệu chính là giải pháp cho bài toán này. Bằng cách sử dụng mô hình được đào tạo trước (*Pre-trained language models*) như *BERT* hay *Sentence Transformers*, chúng ta có thể "mã hóa" ý nghĩa của cả dữ liệu và truy vấn thành các vectơ số học (embeddings). Các vectơ này không chỉ nắm bắt được ý nghĩa ngữ nghĩa của văn bản mà còn giúp đơn giản hóa quá trình so sánh và tìm kiếm. Khoảng cách giữa các vectơ đại diện cho mức độ tương đồng về mặt ngữ nghĩa. Từ

đó, ta có thể dễ dàng xác định các phần dữ liệu (khối) top-k có liên quan nhất đến truy vấn. Phương pháp này mang lại hiệu quả cao, đồng thời rất linh hoạt, có thể ứng dụng trong nhiều tác vụ xử lý ngôn ngữ tự nhiên khác nhau, từ tìm kiếm thông tin, phân loại văn bản đến dịch máy và tóm tắt văn bản.

**Index data**

Bây giờ chúng ta đã có các đoạn được embedding, bước tiếp theo là index data để tạo ra một "hệ thống tra cứu" hiệu quả. Thay vì phải so sánh truy vấn với từng đoạn dữ liệu một, việc index data cho phép chúng ta nhanh chóng tìm ra các đoạn liên quan nhất. Chúng ta sẽ sử dụng cơ sở dữ liệu vectơ (vector database) để lưu trữ các embeddings. Loại cơ sở dữ liệu này được thiết kế đặc biệt để xử lý các vector embedding và hỗ trợ việc tìm kiếm vectơ hiệu quả.

Trong đề tài này, chúng em sẽ sử dụng LlamaIndex - một thư viện phổ biến để xây dựng index data và tìm kiếm vector. Cụ thể, chúng em sẽ sử dụng class VectorStoreIndex từ LlamaIndex để xây dựng index data. Khi có một câu hỏi mới, vectơ biểu diễn của câu hỏi sẽ được tính toán và so sánh với các vectơ trong chỉ mục. Các vectơ có khoảng cách gần nhất với vectơ câu hỏi (thường được tính bằng khoảng cách cosine) sẽ được xem là các đoạn văn liên quan nhất.

### 2.1.2 Retrieval

Với các đoạn embedding được index data trong cơ sở dữ liệu vectơ, chúng em đã sẵn sàng thực hiện truy xuất cho một truy vấn nhất định. Để tìm kiếm thông tin liên quan, chúng em sẽ chuyển đổi câu truy vấn thành dạng vector embedding, sử dụng chính mô hình embedding đã được dùng để "mã hóa" các đoạn văn bản trước đó. Sau đó, hệ thống sẽ so sánh vectơ của câu hỏi với tất cả các vectơ trong cơ sở dữ liệu và chọn ra K đoạn văn bản có mức độ "gần" nhất về mặt ngữ nghĩa. Những đoạn văn bản được chọn này chính là kết quả truy xuất (retrieval) của hệ thống.

### 2.1.3 Response Generation

Bây giờ, có thể sử dụng ngữ cảnh để tạo phản hồi từ LLM của mình. Nếu không có ngữ cảnh liên quan đã truy xuất, LLM có thể không đưa ra câu trả

lời chính xác cho câu hỏi. Với sự phát triển của dữ liệu, chúng ta có thể dễ dàng nhúng và lập chỉ mục bất kỳ dữ liệu mới nào, từ đó có thể truy xuất để đáp ứng các câu hỏi.

## 2.2 Kỹ thuật Prompt Engineering

*Prompt Engineering* là kỹ thuật thực hành thiết kế và tạo ra các gợi ý đầu vào hoặc ví dụ được cung cấp cho mô hình ngôn ngữ một cách cẩn thận, với mục tiêu gợi ra các đầu ra hoặc hành vi mong muốn. Kỹ thuật này nhận ra rằng cách diễn đạt và cấu trúc lời nhắc có thể ảnh hưởng đáng kể đến các phản hồi được tạo ra bởi mô hình.

Một số kỹ thuật Prompt Engineering phổ biến:

- *Zero-shot prompting*: Đây là kỹ thuật Prompt Engineering trực tiếp và đơn giản nhất, trong đó mô hình chỉ được cung cấp hướng dẫn trực tiếp hoặc được hỏi một câu hỏi mà không có thêm thông tin bổ sung nào. Cách tiếp cận này phù hợp nhất cho các tác vụ tương đối đơn giản hơn là những tác vụ phức tạp.

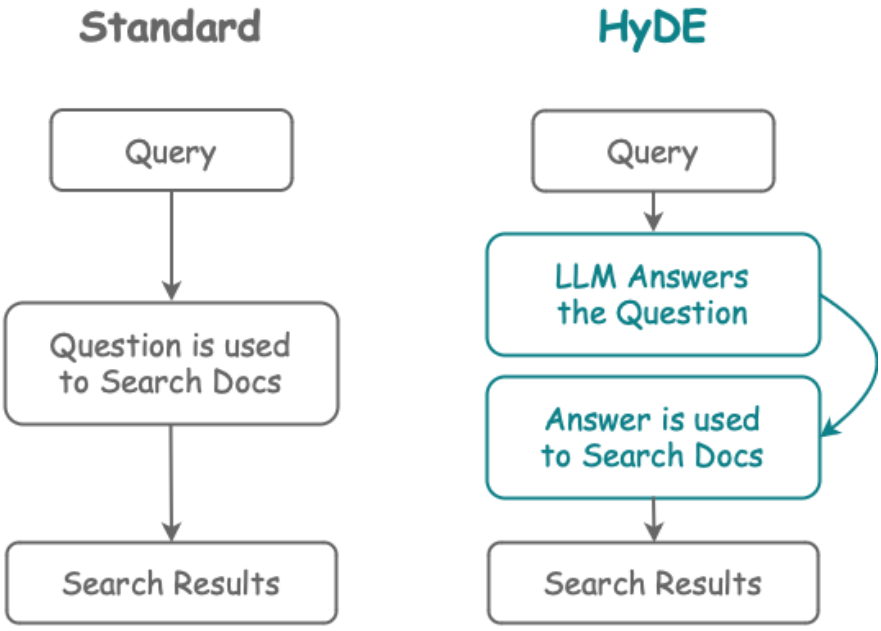
- *Few-shot prompting*: Kỹ thuật này cung cấp một số ví dụ hoặc lời nhắc thể hiện định dạng hoặc mẫu đầu ra mong muốn, cho phép mô hình học hỏi từ những ví dụ đó.

- *Chain-of-thought prompting (CoT)*: Là một kỹ thuật nâng cao cung cấp cho mô hình quy trình lập luận từng bước để tuân theo. Bằng cách chia nhỏ một nhiệm vụ phức tạp thành các bước trung gian, hoặc "chuỗi suy luận", phương pháp này giúp mô hình hiểu được ngữ cảnh và tạo ra kết quả chính xác hơn.

Trong khuôn khổ của khoá luận tốt nghiệp này, chúng em lựa chọn sử dụng kỹ thuật Few-shot prompting cho quá trình xây dựng hệ thống hỏi đáp thông tin các trường đại học bằng cách thiết lập prompt cho quá trình truy xuất và đưa ra ví dụ gần với bộ câu hỏi mà chúng em sử dụng.

## 2.3 Phương pháp Hypothetical Document Embeddings

*Hypothetical Document Embeddings (HyDE)* là phương pháp sử dụng mô hình ngôn ngữ, như ChatGPT, để tạo tài liệu lý thuyết khi trả lời truy vấn, thay vì sử dụng truy vấn và vectơ được tính toán để tìm kiếm trực tiếp trong cơ sở dữ liệu vectơ. HyDE tiến xa hơn một bước nữa bằng cách sử dụng bộ mã hóa không giám sát được học thông qua các phương pháp tương phản. Bộ mã hóa này thay đổi tài liệu lý thuyết thành một vectơ nhúng để định vị các tài liệu tương tự trong cơ sở dữ liệu vectơ. Thay vì tìm kiếm sự tương đồng khi nhúng các câu hỏi hoặc truy vấn, phương pháp này tập trung vào sự tương đồng khi nhúng câu trả lời vào câu trả lời.




Hình 2.2: Hypothetical Document Embeddings

Không chỉ tìm kiếm những câu hỏi và câu trả lời tương tự, HyDE thật sự hiểu nội dung mà nó tạo ra và tìm kiếm câu trả lời dựa trên mức độ tương đồng của chúng với các câu trả lời khác, mang lại hiệu quả trong các tác vụ như tìm kiếm trên web, trả lời câu hỏi và kiểm tra thực tế.

Phương thức hoạt động HyDE bao gồm ba bước chính: Đầu tiên, tạo tài liệu giả định bằng cách sử dụng mô hình ngôn ngữ để trả lời truy vấn của người dùng, trong bài này chúng em lựa chọn sử dụng GPT-3.5 Turbo và Mixtral 8x7B để tạo tài liệu. Sau đó, sử dụng bộ mã hóa tương phản không giám sát

để mã hóa tài liệu giả định được tạo thành một vectơ nhúng. Dựa trên sự tương đồng của vectơ, vectơ này chỉ định một vùng trong không gian nhúng dữ liệu nơi các tài liệu thực tương tự được lấy ra. Cuối cùng, trong quá trình truy xuất, HyDE tìm kiếm các tài liệu thực tế có độ tương đồng cao nhất với tài liệu giả định đã được mã hóa

Tuy nhiên, phương pháp này có nhược điểm là không phải lúc nào cũng mang lại kết quả tốt. Ví dụ, nếu chủ đề đang thảo luận hoàn toàn xa lạ với mô hình ngôn ngữ được chọn sử dụng, phương pháp này sẽ không hiệu quả và có thể dẫn đến việc tạo ra thông tin không chính xác. Điều đó cần lựa chọn mô hình ngôn ngữ phù hợp khi sử dụng HyDE.

## 2.4 Mô hình ngôn ngữ

### 2.4.1 Mô hình GPT-3.5 Turbo

GPT-3.5 Turbo là một trong những mô hình ngôn ngữ lớn (LLM) tiên tiến nhất hiện nay, được phát triển bởi OpenAI. Được huấn luyện trên một tập dữ liệu văn bản khổng lồ bằng phương pháp học máy có giám sát *(Supervised Learning)* và học tăng cường từ phản hồi của con người (*Reinforcement Learning from Human Feedback*), GPT-3.5 Turbo sở hữu khả năng tạo văn bản tiếng người tự nhiên ấn tượng. Giống như các phiên bản trước đó, GPT-3.5 Turbo dựa trên kiến trúc Transformer với cơ chế self-attention, cho phép xử lý thông tin và ngữ cảnh một cách hiệu quả.

GPT-3.5 Turbo có khả năng thực hiện đa dạng các tác vụ liên quan đến ngôn ngữ, từ dịch thuật, tóm tắt, trả lời câu hỏi cho đến sáng tác văn bản. Phiên bản "turbo" được cải tiến về tốc độ xử lý và hiệu suất, giúp việc triển khai ứng dụng thực tế trở nên hiệu quả hơn. Điểm mạnh của GPT-3.5 Turbo nằm ở khả năng "hiểu" ngữ cảnh và tạo ra văn bản liên kết, logic, thậm chí sáng tạo. Mô hình có thể được tinh chỉnh cho các nhiệm vụ cụ thể bằng cách cung cấp thêm dữ liệu huấn luyện, giúp tối ưu hóa hiệu suất cho từng ứng dụng.

### 2.4.2 Mô hình Mixtral 8x7B

Mixtral 8x7B là một mô hình ngôn ngữ lớn mã nguồn mở (open-source), được phát triển bởi Mistral AI, mang đến hiệu suất ấn tượng trong một kích

thước tương đối gọn nhẹ. Với 8 khối, mỗi khối chứa 7 tỷ tham số (8x7B), Mixtral 8x7B thể hiện khả năng tạo văn bản chất lượng cao, đồng thời vẫn đảm bảo khả năng triển khai và tinh chỉnh hiệu quả.

Kiến trúc của Mixtral 8x7B được xây dựng dựa trên Transformer, kết hợp cơ chế phân nhóm trọng số *(Grouped-Query Attention)* và kỹ thuật chia sẻ tham số *(Parameter Sharing)* để tối ưu hóa hiệu suất và giảm thiểu chi phí tính toán. Mô hình được huấn luyện trên một tập dữ liệu văn bản khổng lồ, đa dạng, sử dụng phương pháp học máy có giám sát *(Supervised Learning)* kết hợp với kỹ thuật học tăng cường từ phản hồi của con người (*Reinforcement Learning from Human Feedback*) cho phép Mixtral 8x7B học hỏi từ dữ liệu phong phú, nắm bắt được ngữ cảnh và tạo ra văn bản tự nhiên, sáng tạo.

# Chương 3

# Phương pháp xây dựng hệ thống hỏi đáp

Ở chương này, chúng em sẽ trình bày về phương pháp xây dựng hệ thống hỏi đáp thông tin các trường đại học, tập trung vào hai yếu tố chính: bộ dữ liệu và các tham số thiết lập hệ thống. Đầu tiên, chúng em sẽ giới thiệu chi tiết về bộ dữ liệu được thu thập từ các nguồn chính thống, đảm bảo tính tin cậy và cập nhật cho hệ thống. Tiếp theo, chúng em sẽ phân tích các tham số quan trọng được thiết lập trong quá trình xử lý ngôn ngữ tự nhiên và tạo sinh câu trả lời, nhằm tối ưu hóa hiệu suất và độ chính xác của hệ thống. Việc kết hợp hài hòa giữa nguồn dữ liệu đáng tin cậy và các tham số được tinh chỉnh sẽ tạo nên nền tảng vững chắc, giúp hệ thống hoạt động hiệu quả và chính xác.

## 3.1 Xây dựng bộ dữ liệu

### 3.1.1 Nguồn dữ liệu và phạm vi thu thập

Dữ liệu được sử dụng để xây dựng hệ thống hỏi đáp thông tin các trường đại học được thu thập trực tiếp từ trang web chính thức của 07 trường thành viên và 01 khoa trực thuộc Đại học Quốc gia Thành phố Hồ Chí Minh. Việc lựa chọn nguồn dữ liệu trực tiếp từ các website chính thức đảm bảo tính chính xác và cập nhật mới nhất.

Bộ dữ liệu bao gồm thông tin từ:

- Trường Đại học Bách khoa, ĐHQG-HCM
- Trường Đại học Khoa học tự nhiên, ĐHQG-HCM
- Trường Đại học Khoa học Xã hội và Nhân văn, ĐHQG-HCM
- Trường Đại học Quốc tế, ĐHQG-HCM
- Trường Đại học Công nghệ Thông tin, ĐHQG-HCM
- Trường Đại học Kinh tế - Luật, ĐHQG-HCM
- Trường Đại học An Giang, ĐHQG-HCM
- Khoa Y, ĐHQG-HCM

### 3.1.2 Cấu trúc dữ liệu và thông tin chi tiết

Dữ liệu sau khi thu thập dưới dạng văn bản thô từ các trang web, sau đó được xử lý, phân loại và tổ chức theo cấu trúc rõ ràng, bao gồm các thông tin thiết yếu hỗ trợ tối đa trong quá trình truy vấn thông tin:

**Thông tin chung về khoa:**

- Tên khoa: Tên đầy đủ và tên viết tắt của khoa.
- Giới thiệu sơ lược về khoa: Lịch sử hình thành, sứ mệnh, tầm nhìn, thành tựu nổi bật,...
- Cơ sở vật chất: Hệ thống phòng học, phòng thí nghiệm, thư viện, ký túc xá,...
- Thông tin liên hệ: Địa chỉ, số điện thoại, email, website,...

**Danh sách các ngành đào tạo:**

- Tên ngành: Tên đầy đủ của ngành đào tạo.
- Mã ngành: Mã số của ngành theo quy định của Bộ Giáo dục và Đào tạo.

- Chỉ tiêu tuyển sinh: Số lượng thí sinh dự kiến được tuyển vào ngành trong năm tuyển sinh

- Chuyên ngành: Các chuyên ngành đào tạo thuộc ngành (nếu có).

- Chương trình đào tạo: Mô tả chi tiết chương trình đào tạo của ngành (đại trà, chất lượng cao, chương trình tiên tiến, chương trình liên kết quốc tế,...).

- Thời gian đào tạo: Số năm đào tạo của ngành (đặc biệt quan trọng đối với khối ngành sức khỏe)

**Thông tin về cơ chế, chính sách:**

- Học phí: Mức học phí dự kiến của ngành trong năm tuyển sinh.

- Học bổng: Các chương trình học bổng, hỗ trợ tài chính dành cho sinh viên (nếu có).

- Cơ hội nghề nghiệp sau khi tốt nghiệp: Định hướng nghề nghiệp, nhu cầu tuyển dụng, các lĩnh vực có thể làm việc sau khi tốt nghiệp.

### 3.1.3 Hình thức lưu trữ và khả năng truy cập

Để đảm bảo tính tiện dụng và khả năng truy cập dễ dàng, dữ liệu sau khi được tổng hợp và xử lý được lưu trữ thành các file PDF riêng biệt cho từng trường đại học. Hình thức lưu trữ này cho phép người dùng dễ dàng tìm kiếm, tải xuống và chia sẻ thông tin.

**PHẦN 1: KHOA KINH TẾ, TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT, ĐHQG-HCM**

**I. GIỚI THIỆU CHUNG:**

Khoa Kinh tế là cơ sở đào tạo các cán bộ có trình độ đại học và sau đại học, nghiên cứu khoa học và chuyển giao công nghệ theo tiêu chí chất lượng cao, đạt trình độ tiên tiến, đáp ứng nhu cầu nguồn nhân lực hội tụ cả đức và tài phục vụ quá trình công nghiệp hóa, hiện đại hóa đất nước và hội nhập quốc tế. Đội ngũ cán bộ của Khoa Kinh tế hiện nay gồm 24 người bao gồm 8 Tiến sĩ (5 Phó Giáo sư), 15 Thạc sĩ (14 Nghiên cứu sinh và 1 Thư ký).

**II. CÁC NGÀNH ĐÀO TẠO:**

**1. Ngành Kinh tế (Chuyên ngành Kinh tế học):**

- Quy mô đào tạo: 260 sinh viên
- Mã số: 7310101_401
- Mục tiêu đào tạo: Đào tạo đội ngũ những cử nhân kinh tế có kiến thức chuyên sâu, vững vàng trong lĩnh vực kinh tế học; Đào tạo những cử nhân kinh tế có khả năng tổ chức, quản lý, thực thi các hoạt động kinh tế ở khu vực doanh nghiệp; Đào tạo đội ngũ những nhà nghiên cứu, chuyên gia kinh tế có khả năng hoạch định, tham mưu, tư vấn các vấn đề kinh tế cho doanh nghiệp, cơ quan nhà nước các cấp, các tổ chức quốc tế và các tổ chức phi chính phủ; Đào tạo những chuyên gia kinh tế có khả năng nghiên cứu độc lap trong lĩnh vực kinh tế học; Đào tạo những cử nhân kinh tế có sức khỏe tốt, có phẩm chất đạo đức nghề nghiệp.

**2. Ngành Kinh tế (Chuyên ngành Kinh tế và Quản lý công):**

- Quy mô đào tạo: 260 sinh viên
- Mã số: 7310101_403
- Mục tiêu đào tạo: Đào tạo những cử nhân kinh tế có có kiến thức chuyên sâu và kỹ năng quản lý nhà nước về kinh tế tại khu vực hành chính, đơn vị sự nghiệp; Đào tạo đội ngũ những nhà nghiên cứu, chuyên gia kinh tế có khả năng phân tích, hoạch định, tham mưu, tư vấn, tổ chức thực hiện các chính sách kinh tế- xã hội, các kế hoạch, chương trình, dự án đầu tư công và cung ứng dịch vụ công; Đào tạo những cử nhân kinh tế có khả năng tổ chức, quản lý, vận hành các hoạt động kinh tế ở khu vực doanh nghiệp công và tư; Đào tạo những cử nhân kinh tế có tư duy nghiên cứu độc lập trong lĩnh vực kinh tế và quản lý công, có khả năng học lên ở bậc cao hơn.

**III. HỌC PHÍ:**

**1. Ngành Kinh tế (Chuyên ngành Kinh tế học):** 25.900.000 đồng

**2. Ngành Kinh tế (Chuyên ngành Kinh tế và Quản lý công):** 25.900.000 đồng

Hình 3.1: Dữ liệu thông tin của khoa Kinh tế, trường Đại học Kinh tế - Luật, ĐHQG-HCM

## 3.2 Thiết lập các tham số cho quá trình tạo sinh câu trả lời

Trong quá trình thực hiện khoá luận tốt nghiệp, chúng em nhận thấy rằng việc điều chỉnh một số thông số nhất định có thể ảnh hưởng đáng kể đến kết quả tạo sinh câu trả lời.

### 3.2.1 Chunk size

Như đã nói ở trên việc phân đoạn ảnh hưởng đáng kể đến chất lượng của nội dung được tạo ra, bao gồm 100, 128, 512, 1024,... Nên việc lựa chọn các

kích thước chunk_size thực sự quan trọng.

Qua quá trình tìm hiểu đánh giá và khảo sát để tương thích với mô hình ngôn ngữ và bộ dữ liệu của chúng em. Chúng em nhận thấy với các thông số chunk_size nhỏ tạo ra các khối chi tiết, tuy nhiên một số thông tin quan trọng không nằm trong các khối đó. Đối với các chunk_size lớn, nó đảm bảo toàn diện các thông tin trên các khối, tuy nhiên nó cũng làm chậm quá trình tạo sinh câu trả lời.

Chúng em lựa chọn chunk_size = 512, sử dụng trong các mô hình cần ngữ cảnh lớn hơn, đảm bảo các thông tin nằm trong các khối và quá trình tạo sinh nhanh chóng.

### 3.2.2 Chunk overlap

Khi chia thành các khối thông tin như vậy, sẽ có trường hợp các đoạn thông tin không chứa đầy đủ ngữ cảnh, vì các khối này được chia theo số lượng token chứ không phải theo câu, dẫn đến việc một câu có thể bị cắt thành hai phần, trong đó một phần nằm ở cuối khối này và phần còn lại nằm ở đầu khối tiếp theo. Điều này khiến hệ thống gặp khó trong việc sinh ra được câu trả lời chính xác. Để giải quyết vấn đề này, chúng em sử dụng thêm một thông số là chunk_overlap. Chunk_overlap được sử dụng để xác định số lượng token chồng chéo giữa các khối liên tiếp nhau, cụ thể là số token ở cuối khối này sẽ là số token ở đầu khối kia. Điều này rất hữu ích khi tách ra các đoạn văn bản vì nó giúp duy trì được tính ngữ cảnh giữa các khối với nhau, đảm bảo cho tính toàn vẹn của thông tin, giữ được ý nghĩa và sự mạch lạc khi xử lý văn bản bằng hệ thống.

### 3.2.3 Top K

Một tham số quan trọng khác ảnh hưởng đến quá trình tạo sinh câu trả lời là Top_K. Phương pháp lấy mẫu Top_K bao gồm việc chọn K từ có xác suất cao nhất từ phân bố xác suất, sau đó chỉ lấy mẫu từ ngẫu nhiên từ tập con này để tạo ra từ tiếp theo. Ví dụ: Hãy tưởng tượng một kịch bản trong đó mô hình ngôn ngữ lớn (LLM) được cung cấp đoạn đầu vào "Thủ đô của Pháp là...". Mô hình có thể gán xác suất cao nhất cho các từ như "Paris", "một", "xinh đẹp",... Với lấy mẫu top_k = 3, mô hình sẽ chỉ xem xét 3 từ

có xác suất cao nhất cho đầu ra tiếp theo.

Việc lựa chọn Top_K với giá trị quá cao có thể khiến kết quả đầu ra chung chung và kém sáng tạo. Mặt khác, các giá trị quá thấp có thể khiến kết quả đầu ra trở nên quá khó đoán và vô nghĩa. Do đó, việc lựa chọn K thường là sự thử.

Dưới đây là bảng kết quả đánh giá sử dụng giá trị độ đo nằm trong khoảng (0; 1) của hệ thống sử dụng top_k = 3 và top_k = 5

| **Độ đo** | **Top_K = 3** | **Top_K = 5** |
|---|---|---|
| Faithfulness | 0.62 | 0.74 |
| Answer Relevance | 0.73 | 0.79 |
| Context Precision | 0.56 | 0.73 |
| Context Recall | 0.56 | 0.79 |

Bảng 3.1: Bảng Kết quả đánh giá RAG dựa trên Top_k

Kết quả đánh giá trên cho thấy với bộ dữ liệu của chúng em khi đưa vào mô hình thì với Top_K = 5 đưa ra kết quả tốt hơn . Có một điều lưu ý là chúng em đã thực hiện thí nghiệm Top_K với các giá trị $K$ từ 1 đến 12 và chọn ra giá trị Top K phù hợp nhất trong bài thí nghiệm là Top_K = 5.

# Chương 4

# Thực nghiệm và đánh giá

## 4.1 Những thách thức trong việc đánh giá hệ thống hỏi đáp

Retrieval-Augmented Generation (RAG) là một kỹ thuật phức tạp, gắn bó chặt chẽ với các yêu cầu và mô hình ngôn ngữ cụ thể. Điều này dẫn đến sự đa dạng trong các phương pháp và công cụ đánh giá, đặc biệt đối với các mô hình ngôn ngữ lớn (LLM) vốn thiếu minh bạch và khó dự đoán. Việc phát triển các tiêu chí đánh giá toàn diện, có khả năng nắm bắt hiệu quả sự tương tác giữa độ chính xác truy xuất và chất lượng tổng hợp là điều cần thiết. Nhằm làm sáng tỏ các yếu tố quan trọng, chúng em đã tiến hành đánh giá hệ thống để xác định tính khả thi của kết quả. Trong quá trình này, chúng em cũng đã phát hiện ra nhiều thách thức cần được chú ý và giải quyết.

- *Đối với Retrieval:* Thách thức chính trong việc đánh giá thành phần này là dữ liệu động và rộng lớn của các cơ sở tri thức tiềm năng, từ dữ liệu có cấu trúc đến các trang web. Các thông tin đó chỉ mang tính chất của hiện tại và có khả năng thay đổi trong tương lai, dẫn đến độ chính xác của dữ liệu có thể thay đổi theo thời gian, làm tăng thêm độ phức tạp cho quá trình đánh giá. Tài liệu đưa vào quá lớn để có thể tìm và truy xuất đưa vào ngữ cảnh đòi hỏi cần lựa chọn các tham số thích hợp.

- *Đối với Generation:* Hệ thống không chỉ cần cung cấp thông tin chính xác mà còn phải thể hiện khả năng suy luận logic để trả lời những câu hỏi phức tạp, đảm bảo câu trả lời thể hiện được quá trình lập luận chặt chẽ, hợp lý chứ không đơn thuần đưa ra thông tin đúng sai. Bên cạnh

đó, tính mạch lạc và dễ hiểu của câu trả lời cũng là một yếu tố quan trọng cần được xem xét, bao gồm sự liên kết ý, cách sử dụng từ ngữ, ngữ pháp, và cách thức triển khai ý tưởng sao cho tự nhiên như một đoạn văn bản do con người viết.

- *Trên toàn hệ thống:* Toàn bộ hệ thống không thể được hiểu hoàn toàn chỉ bằng cách đánh giá từng thành phần của nó riêng lẻ. Thay vào đó, hệ thống cần được đánh giá về khả năng tận dụng thông tin được truy xuất một cách hiệu quả để cải thiện chất lượng phản hồi, bao gồm việc đo lường giá trị gia tăng của thành phần truy xuất đối với quy trình tổng hợp.

## 4.2 Tập dữ liệu đánh giá

Chúng em đã tạo tập dữ liệu bao gồm 40 cặp question và ground_truth từ tập dữ liệu thông tin các trường thuộc khối Đại học Quốc gia Thành phố Hồ Chí Minh.

Các question liên quan đến các chủ đề như: ngành học, chỉ tiêu, học phí, cơ hội nghề nghiệp,... đều gần với thực tế nhu cầu truy vấn của học sinh. Các ground_truth được lấy từ bộ dữ liệu thông tin các trường để đảm bảo thông tin chính xác được cập nhật.

| Question | Ground_truth |
|---|---|
| Nhóm ngành Toán của trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm những ngành nào? | Ngành Toán học, ngành Toán tin và ngành Toán ứng dụng |
| Khoa Y, ĐHQG-HCM đào tạo các ngành nào? | Ngành Điều dưỡng, ngành Y học cổ truyền, ngành Răng hàm mặt, ngành Dược học, ngành Y Khoa |
| Trường Đại học An Giang, ĐHQG-HCM có đào tạo ngành Công nghệ sinh học không? | Trường Đại học An Giang, ĐHQG-HCM có đào tạo ngành Công nghệ sinh học |
| Kể tên các loại học bổng của Khoa Báo chí và truyền thông, trường Đại học Khoa học Xã hội và Nhân văn, ĐHQG-HCM? | Học bổng khuyến khích, học bổng thường niên, học bổng từ các doanh nghiệp đối tác, học bổng tham gia các khoá học ngắn hạn |
| Trường Đại học Khoa học tự nhiên, ĐHQG-HCM đào tạo ngành nào nhiều sinh viên nhất? | Ngành Công nghệ thông tin Chương trình Chất lượng cao |
| Cơ hội nghề nghiệp tiềm năng cho sinh viên tốt nghiệp ngành Kỹ thuật Điện tử - Viễn thông trường Đại học Khoa học tự nhiên, ĐHQG-HCM là gì? | Làm việc ở các công ty thiết kế mạch, làm việc tại các công ty viễn thông hoặc làm việc trong các công ty y tế và thiết bị y tế |
| Học phí ngành Hoá học Chương trình Chất lượng cao trường Đại học Khoa học tự nhiên, ĐHQG-HCM là bao nhiêu? | 40.000.000 đồng/năm |

Hình 4.1: Minh hoạ tập dữ liệu đánh giá

Các câu hỏi từ bộ dữ liệu được phân loại thành 2 nhóm chính, dựa trên các tiêu chí như mức độ phức tạp của truy vấn và tổng hợp thông tin. Thông

qua việc phân loại, ta có thể đánh giá chính xác hơn về độ hiệu quả của hệ thống với từng mức độ câu hỏi. Thông tin về phân loại và tiêu chí cụ thể ở bảng sau:

| **Phân loại** | **Tiêu chí** | **Số lượng** | **Tỷ lệ** | **Ví dụ** |
|---|---|---|---|---|
| Single-Context | Câu hỏi chỉ cần tra cứu thông tin trực tiếp từ một nguồn duy nhất | 30 | 75% | Thời gian đào tạo ngành Y học cổ truyền của Khoa Y, ĐHQG-HCM là bao nhiêu năm? |
| Multi-Context | Câu hỏi đòi hỏi kết hợp thông tin từ nhiều phần hoặc nguồn dữ liệu khác nhau | 10 | 25% | Kể tên các trường trong khối Đại học Quốc gia TP.HCM có đào tạo ngành Quản trị kinh doanh? |

Bảng 4.1: Phân loại câu hỏi

## 4.3 Metric đánh giá

Việc đánh giá hệ thống đòi hỏi phải xem xét hiệu suất của cả hai thành phần là truy xuất (Retrieval) và tạo sinh văn bản (Generation), cũng như sự phối hợp nhịp nhàng giữa chúng. Để đánh giá thành phần truy xuất, các độ đo được chúng em sử dụng trong bài bao gồm: *Context Precision* và *Context Recall* - các độ đo này đánh giá khả năng của hệ thống trong việc tìm kiếm và xếp hạng các tài liệu có liên quan đến truy vấn. Đối với thành phần tạo sinh văn bản, các độ đo như *Faithfulness* và *Answer Relevancy* được sử dụng để so sánh văn bản được tạo với văn bản tham chiếu, đánh giá mức độ chồng chéo nội dung và tương đồng ngữ nghĩa.

Các độ đo được sử dụng để đánh giá hệ thống dưới đây đến từ RAGAS framework với các giá trị độ đo nằm trong khoảng [0, 1]. Các giá trị càng gần 1 thì sinh ra kết quả tốt, ngược lại các giá trị càng gần 0 sinh ra giá trị kém.

### 4.3.1 Faithfulness

Faithfulness đo lường tính nhất quán thực tế của câu trả lời được tạo ra so với ngữ cảnh nhất định. Nó được tính toán từ answer và context được truy xuất và có giá trị nằm trong phạm vi [0, 1].

Câu trả lời tạo ra được coi có tính trung thực nếu tất các các ý trong câu trả lời đưa ra đều được lấy trong phần ngữ cảnh được cho. Để tính toán giá trị faithfulness, trước tiên tập hợp các ý của câu trả lời được xác định xem xét xem có trong phần ngữ cảnh hay không. Điểm faithfulness tính bằng:

$$\text{Faithfulness score} = \frac{a}{b}$$

Với:

$a$: Tổng số các ý của answer có trong context

$b$: Tổng số các ý của answer

### 4.3.2 Answer Relevancy

Chỉ số đánh giá, Answer Relevancy, tập trung vào việc đánh giá mức độ liên quan của câu trả lời được tạo ra với lời nhắc đã cho. Điểm thấp hơn được gán cho các câu trả lời không đầy đủ hoặc chứa thông tin thừa và điểm cao hơn cho thấy mức độ liên quan tốt hơn.
Tính liên quan của câu trả lời được định nghĩa là độ tương đồng cosin trung bình của câu hỏi với câu hỏi nhân tạo được tạo ra từ câu trả lời.

$$\text{Answer Relevancy} = \frac{1}{N}\sum_{i=1}^{N}\cos(E_{g_i}, E_0) = \frac{1}{N}\sum_{i=1}^{N}\frac{E_{g_i}\cdot E_0}{\|E_{g_i}\|\cdot\|E_0\|}$$

Với:

$E_{gi}$: Sử dụng câu trả lời để đưa ngược qua mô hình ngôn ngữ tạo ra các câu hỏi liên quan

$E_0$: Câu hỏi ban đầu

N: Số câu hỏi được tạo (Mặc định là 3 câu)

### 4.3.3 Context Precision

Context Precision là một số liệu đánh giá xem tất cả các mục có liên quan đến *ground_truth* có trong *context*, được xếp hạng cao hay không. Lý tưởng nhất tất cả các khối có liên quan phải xuất hiện ở thứ hạng cao nhất *ground_truth* và *context*, với các giá trị nằm trong khoảng từ 0 đến 1, trong đó điểm cao hơn cho thấy độ chính xác tốt hơn.

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} \text{precision@k} * v_k}{n}$$

Với

$$\text{precision@k} = \frac{\text{True precision@k}}{\text{True precision@k} + \text{False precision@k}}$$

$n$: Tổng số mục $k$ có chứa thông tin liên quan trên $K$

$K$: Tổng số đoạn $k$ trong context

$v_k = 1$: Nếu các đoạn nhỏ của ground_truth có trong context

### 4.3.4 Context Recall

Context Recall đo lường mức độ mà ngữ cảnh được truy xuất phù hợp với grouth_truth. Nó được tính toán dựa trên *ground_truth* và *context* với các giá trị nằm trong khoảng [0, 1].

Để ước tính context_recall từ câu trả lời có căn cứ, mỗi grouth_truth sẽ được phân tích để xác định xem liệu nó có thể được quy cho bối cảnh được truy xuất hay không. Trong một tình huống lý tưởng, tất cả các *ground_truth* đều có trong *context*.

$$\text{Context Recall} = \frac{a}{b}$$

Với:

$a$: Tổng số ý trong ground_truth có trong context

$b$: Tổng số ý ban đầu của ground_truth

### 4.3.5 Vận dụng

Để hiểu rõ hơn về các độ đo đánh giá, chúng em có một ví dụ sau đây với $top\_k = 3$.

**Question:** Nhóm ngành Toán của Trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm những ngành nào?

**Context:**

- $k = 1$: Khoa Toán - Tin học trường Đại học Khoa học tự nhiên, ĐHQG-HCM có mã số ngành là: $7460101\_NN$ và bao gồm các chuyên ngành: Ngành Toán học, Ngành Toán tin, Ngành Toán ứng dụng.
- $k = 2$: Khoa Toán - Tin học là một trong những khoa có bề dày truyền thống của Trường Đại học Khoa học tự nhiên, ĐHQG-HCM. Với đội ngũ giảng viên Uy tín - Tận tâm - Trình độ chuyên môn cao, Khoa là đơn vị duy nhất tổ chức đào tạo từ bậc Cử nhân đến bậc Tiến sĩ các ngành về Toán trong Đại học Quốc gia Thành phố Hồ Chí Minh.
- $k = 3$: Khoa Toán - Tin học trường Đại học Khoa học tự nhiên, ĐHQG-HCM có chuyên ngành bao gồm: Ngành Toán học với các chuyên ngành: Đại số, Giải tích, Giải tích số, Xác suất thống kê; Ngành Toán tin với các chuyên ngành: Khoa học dữ liệu, Toán tin ứng dụng, Phương pháp Toán trong Tin học; Ngành Toán ứng dụng với các chuyên ngành: Cơ học, Toán tài chính, Tối ưu và hệ thống, Lý luận và phương pháp giảng dạy môn Toán.

**Answer:** Nhóm ngành của khoa Toán - Tin học trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm các ngành: Toán học, Toán tin và Toán ứng dụng.

**Ground_truth:** Ngành Toán học, ngành Toán tin và ngành Toán ứng dụng

**Khi đó, Faithfulness được tính như sau:**
Bước 1: Chia các câu trả lời thành các ý riêng lẻ:

- Câu 1: Nhóm ngành Toán của trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm ngành Toán học.
- Câu 2: Nhóm ngành Toán của trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm ngành Toán tin.

- Câu 3: Nhóm ngành Toán của trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm ngành Toán ứng dụng.

Bước 2: Đối với mỗi câu lệnh được tạo ra, hãy xác minh xem câu lệnh đó có thể được suy ra từ ngữ cảnh đã cho hay không.

- Câu 1: Có
- Câu 2: Có
- Câu 3: Có

Bước 3: Sử dụng công thức mô tả ở trên để tính Faithfulness

$$\text{Faithfulness score} = \frac{3}{3} = 1$$

**Answer Relevancy được tính như sau:**
Bước 1: Thiết kế ngược "n" biến thể của câu hỏi từ câu trả lời được tạo ra bằng cách sử dụng ChatGPT. Ví dụ: đối với câu trả lời "Nhóm ngành Toán của trường Đại học Khoa học tự nhiên, ĐHQG-HCM bao gồm các ngành: Toán học, Toán tin, và Toán ứng dụng", mô hình ngôn ngữ lớn có thể tạo ra các câu hỏi khả thi sau:

- Câu 1: Nhóm ngành Toán của trường Đại học Khoa học Tự nhiên, ĐHQG-HCM bao gồm các ngành nào? Vậy $\cos(E_g i, E_0) = 1$
- Câu 2: Kể tên các ngành Toán của trường Đại học Khoa học tự nhiên, ĐHQG-HCM? Vậy $\cos(E_g i, E_0) = 1$
- Câu 3: Các ngành: Toán học, Toán tin và Toán ứng dụng thuộc nhóm ngành nào của trường Đại học Khoa học tự nhiên, ĐHQG-HCM? Vậy $\cos(E_g i, E_0) = 0.8$

Bước 2: Tính độ tương đồng cosin trung bình giữa các câu hỏi được tạo ra và câu hỏi thực tế.

$$\begin{aligned}\text{Answer Relevancy} &= \frac{1}{N}\sum_{i=1}^{N}\cos(E_{gi}, E_0) \\ &= \frac{1}{3}(1 + 1 + 0.8) = 0.933\end{aligned}$$

**Context Precision được tính như sau:**
Bước 1: Đối với mỗi k trong ngữ cảnh được lấy ra, hãy kiểm tra xem nó có liên quan đến ground_truth hay không.

- k=1: context tạo ra có liên quan đến ground_truth
- k=2: context tạo ra không liên quan đến ground_truth
- k=3: context tạo ra có liên quan đến ground_truth

Bước 2: Tính độ precision@k cho mỗi khối trong ngữ cảnh.

$$\text{precision@1} = \frac{1}{1}$$

$$\text{precision@2} = \frac{1}{2}$$

$$\text{precision@1} = \frac{2}{3}$$

Bước 3 : Tính toán giá trị trung bình của precision@k để đưa ra kết quả context_precision

$$\text{Context Precision} = \frac{(1 * 1 + 0.5 * 0 + 0.67 * 1)}{2} = 0.833$$

**Context Recall được tính như sau:**
Bước 1: Xác định các ý trong ground_truth có trong context hay không: Ngành Toán học, ngành Toán tin và ngành Toán ứng dụng

- Câu 1: Ngành Toán học
- Câu 2: Ngành Toán tin
- Câu 3: Ngành Toán ứng dụng

Bước 2: Ước tính giá trị của $S$ ?

$$S = 1 + 1 + 1 = 3$$

Bước 3 tính giá trị Context Recall

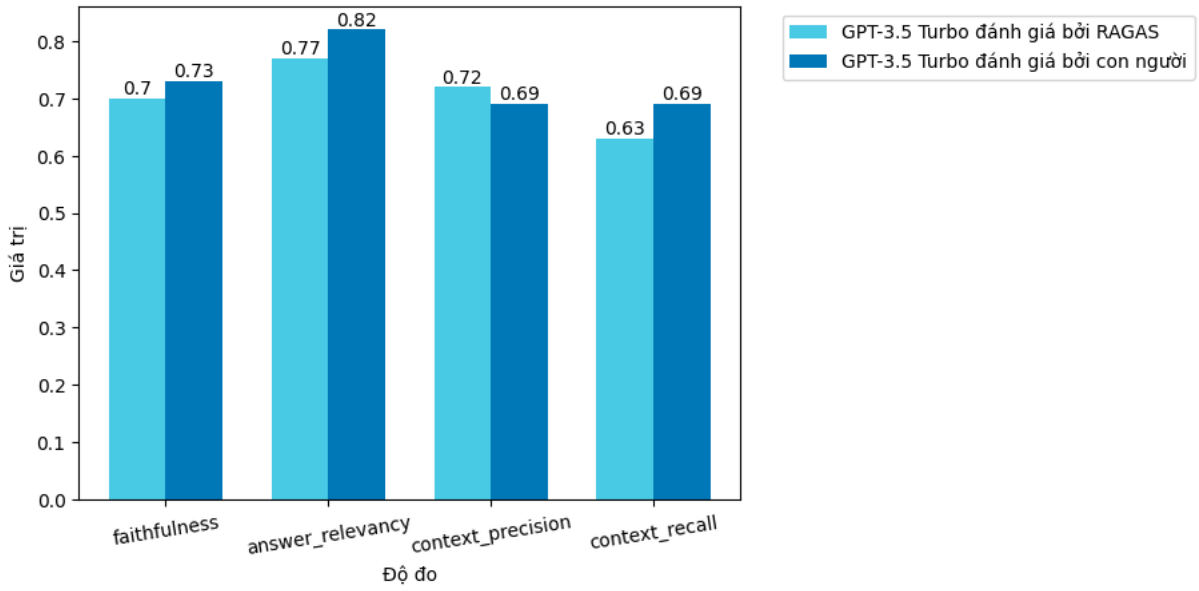
$$\text{Context Recall} = \frac{3}{3} = 1$$

Từ trên ta có giá trị độ đo metric là: Faithfulness = 1, Answer Relevancy = 0.933, Context Precision = 0.833, Context Recall = 1.

## 4.4 Thí nghiệm đánh giá

### 4.4.1 Thí nghiệm 1: Đánh giá hệ thống sử dụng RAG kết hợp GPT-3.5 Turbo

Từ các phương pháp đánh giá đã nêu ở trên, chúng em tiến hành thí nghiệm đánh giá hiệu suất của kỹ thuật RAG kết hợp với mô hình ngôn ngữ GPT-3.5 Turbo trên tập đánh giá sẵn có. Kết quả của thí nghiệm này được thể hiện ở biểu đồ dưới đây:




Hình 4.2: So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo và đánh giá trực tiếp bởi con người

- Độ đo faithfulness đạt được giá trị ở mức 0.70. Cho thấy mô hình có khả năng sinh ra những câu trả lời đáng tin cậy liên quan đến ngữ cảnh được truy xuất, tuy nhiên vẫn chưa thực sự tốt để có thể tin tưởng hoàn toàn vào câu trả lời.

- Độ đo answer_relevancy có giá trị cao nhất là hơn 0.75, cho thấy mô hình rất thành công trong việc tạo ra các câu trả lời liên quan trực tiếp đến câu hỏi đặt ra. Điều này cho thấy khả năng hiểu và xử lý ngôn ngữ tự nhiên của mô hình GPT-3.5 Turbo là rất tốt.

- Độ đo context_precision đạt giá trị hơn 0.70, đây là một giá trị ở mức khá. Chứng tỏ mô hình khi tìm ra các đoạn ngữ cảnh thì những đoạn

ngữ cảnh có liên quan đến ground_truth sẽ thường nằm ở những vị trí đầu tiên được truy xuất.

- Còn với context_recall xấp xỉ khoảng 0.63 thấp hơn so với kết quả đánh giá của nhóm. Chứng tỏ mô hình chưa làm tốt trong quá trình truy xuất, cụ thể ở đây là chưa tìm được ngữ cảnh thỏa mãn mong muốn với ground_truth. Ngữ cảnh được truy xuất không chứa nhiều thông tin về ground_truth dẫn đến việc sinh ra câu trả lời chưa được chính xác. Nguyên nhân có thể câu hỏi được đưa vào chưa cung cấp đầy đủ hoặc không rõ ràng về yêu cầu, cũng có thể do mô hình chưa xử lý tốt để truy xuất thông tin tương ứng.

Trong quá trình thực hiện đề tài để hiểu rõ chi tiết hơn các độ đo và phương pháp đánh giá, chúng em đã tiến hành tự đánh giá trực tiếp trên kết quả có sắn được sinh ra từ kỹ thuật RAG kết hợp với mô hình GPT-3.5 Turbo.

Biểu đồ cho thấy kết quả sinh ra từ mô hình đánh giá nhờ vào GPT-3.5 Turbo và từ quá trình nghiên cứu được đánh giá trực tiếp bởi nhóm không có sự chênh lệch quá lớn. Điều này cho thấy sự hiểu biết về các độ đo và phương pháp đánh giá của chúng em được chính xác hơn, cụ thể như sau:

- Tương quan về xu hướng: Kết quả đánh giá bởi RAGAS và con người có xu hướng tương tự nhau ở cả 4 metric:
  - faithfulness: Cả hai phương pháp đánh giá đều cho kết quả cao, cho thấy hệ thống tạo ra câu trả lời đáng tin cậy, bám sát nguồn dữ liệu.
  - answer_relevancy: Kết quả đánh giá bởi RAGAS cao hơn một chút so với con người, cho thấy RAGAS có thể đánh giá cao hơn về độ liên quan của câu trả lời so với cảm nhận chủ quan của con người.
  - context_precision: Kết quả đánh giá tương đồng, cho thấy cả RAGAS và con người đều đánh giá cao khả năng trích dẫn thông tin chính xác của hệ thống.
  - context_recall: Kết quả đánh giá của con người cao hơn RAGAS, có thể do con người có khả năng nhận biết thông tin ngữ cảnh bị thiếu sót tốt hơn so với framework tự động.
- Chênh lệch không đáng kể: Mặc dù có sự khác biệt nhỏ về giá trị tuyệt đối ở một số metric, nhưng chênh lệch giữa kết quả đánh giá bởi RAGAS

và con người không đáng kể. Điều này cho thấy RAGAS có khả năng đánh giá chất lượng hệ thống một cách khách quan và tin cậy, tương đương với đánh giá của con người.

Bảng dưới đây là kết quả đánh giá của từng loại câu hỏi:

| **Phân loại** | **Faithfulness** | **Answer Relevancy** | **Context Precision** | **Context Recall** | **Tổng** |
|---|---|---|---|---|---|
| Single-Context | 0.74 | 0.74 | 0.70 | 0.67 | 2.84 |
| Multi-Context | 0.54 | 0.93 | 0.77 | 0.51 | 2.75 |
| Tổng thể | 0.70 | 0.77 | 0.72 | 0.63 | 2.82 |

Bảng 4.2: Bảng đánh giá phân loại câu hỏi của RAG kết hợp GPT-3.5 Turbo

Tổng quan về phân loại các câu hỏi:

- Nhìn chung, hệ thống thể hiện hiệu suất khá tốt trên cả hai loại câu hỏi, với điểm tổng thể dao động từ 2.75 đến 2.84.
- Hệ thống xử lý tốt nhất với câu hỏi loại Single-Context (2.84 điểm), cho thấy GPT-3.5 Turbo có khả năng truy xuất và kết nối thông tin từ ngữ cảnh tốt.
- Câu hỏi loại Multi-Context (2.75 điểm) là thử thách nhất đối với hệ thống, cho thấy còn hạn chế trong việc kết hợp thông tin từ nhiều nguồn khác nhau.

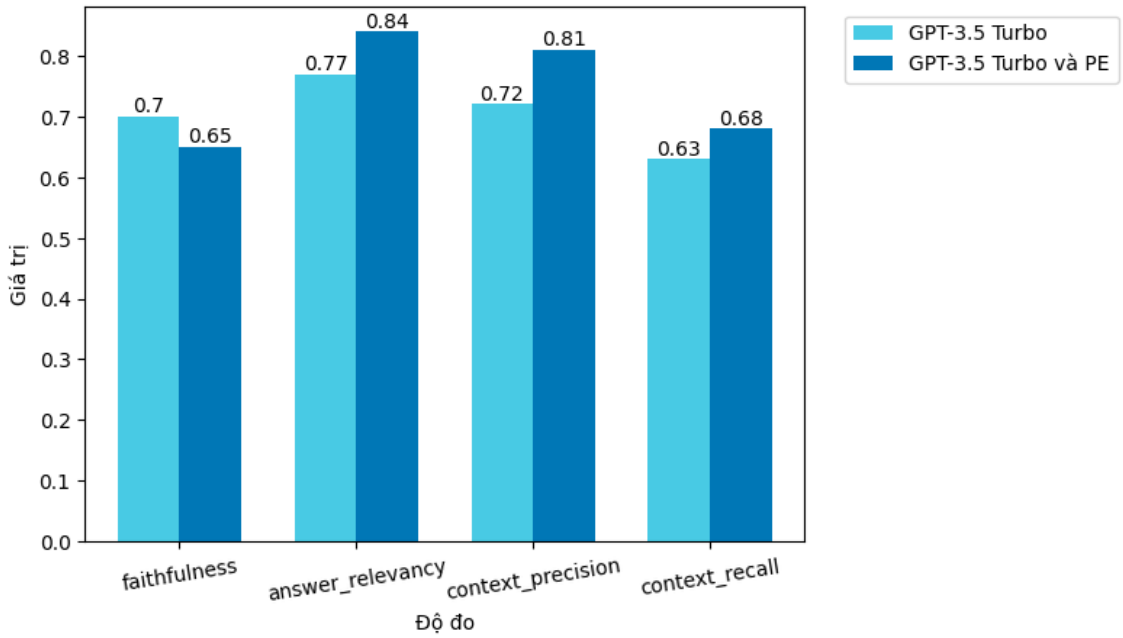
Phân tích chi tiết theo từng loại câu hỏi:

- Single-Context:
  - Điểm mạnh: Hệ thống đạt điểm cao nhất ở metric faithfulness và answer_relevancy(0.74), cho thấy khả năng tạo ra câu trả lời chính xác, bám sát thông tin trong nguồn dữ liệu.
  - Hạn chế: Điểm context_recall (0.67) ở mức trung bình cho thấy hệ thống đôi khi còn gặp khó khăn trong việc lựa chọn thông tin thực sự liên quan và đầy đủ để trả lời cho dù câu hỏi đơn giản.

- Multi-Context:
  - Điểm mạnh: Điểm answer_relevancy (0.93) cao cho thấy hệ thống có khả năng xác định được các thông tin liên quan đến câu hỏi, ngay cả khi câu hỏi yêu cầu kết hợp thông tin từ nhiều nguồn khác nhau.
  - Hạn chế: Độ đo faithfulness (0.54) và context_recall (0.51) vẫn còn thấp cho thấy hệ thống còn gặp nhiều khó khăn trong việc tổng hợp, kết nối thông tin từ nhiều nguồn khác nhau để tạo ra câu trả lời chính xác, đầy đủ và logic.

Chúng em đã tìm hiểu một số phương pháp cải thiện hiệu suất của mô hình. Trong đó đề xuất kỹ thuật Prompt Engineering và phương pháp Hypothetical Document Embeddings (HyDE) được sử dụng nhằm cải thiện quá trình truy xuất. Đồng thời thử nghiệm thêm trên mô hình ngôn ngữ Mixtral 8x7B để xem xét liệu rằng có thể làm tăng giá trị của các chỉ số đánh giá so với các mô hình GPT-3.5 Turbo.

### 4.4.2 Thí nghiệm 2: Đánh giá hệ thống sử dụng RAG kết hợp GPT-3.5 Turbo và Prompt Engineering




Hình 4.3: So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo và Prompt Engineering

Khi đánh giá hệ thống kết hợp mô hình GPT-3.5 Turbo với kỹ thuật Prompt Engineer cho ra các độ đo context_recall = 0.68, answer_relevancy = 0.84 và context_precision = 0.81, vượt trội so với thời điểm chưa cải tiến. Có thể thấy rằng mô hình RAG kết hợp với GPT-3.5 Turbo làm khả năng xử hiểu và xử lý mô hình ngôn ngữ tốt và sử dụng kỹ thuật cải tiến Prompt Engineering làm tăng độ chính xác của ngữ cảnh, mức độ đầy đủ của thông tin khi đưa vào ngữ cảnh đầy đủ. Mặt khác, độ đo faithfulness lại thấp hơn, nguyên nhân có thể do việc cung cấp thêm lời nhắc khiến mô hình bị loạn trong việc sinh ra câu trả lời.

Dưới đây là kết quả đánh giá của từng loại câu hỏi:

| **Phân loại** | **Faithfulness** | **Answer Relevancy** | **Context Precision** | **Context Recall** | **Tổng** |
|---|---|---|---|---|---|
| Single-Context | 0.68 | 0.82 | 0.82 | 0.72 | 3.03 |
| Multi-Context | 0.54 | 0.91 | 0.80 | 0.56 | 2.81 |
| Tổng thể | 0.65 | 0.84 | 0.81 | 0.68 | 2.98 |

Bảng 4.3: Bảng đánh giá phân loại câu hỏi của RAG kết hợp GPT-3.5 Turbo và Prompt Engineering

Tổng quan về phân loại các câu hỏi:

- Kỹ thuật Prompt Engineering mang lại hiệu quả tích cực, giúp hệ thống cải thiện hiệu suất trên cả hai loại câu hỏi so với mô hình GPT-3.5 Turbo đơn lẻ.
- Hệ thống đạt điểm tổng thể cao nhất với câu hỏi Single-Context (3.03 điểm), cho thấy Prompt Engineering giúp tối ưu hóa khả năng trả lời câu hỏi trực tiếp, và GPT-3.5 Turbo cho ra kết quả hiệu quả tốt đối với loại câu hỏi mang tính truy vấn trực tiếp.
- Mặc dù điểm số cho câu hỏi Multi-Context (2.81 điểm) thấp hơn nhưng đã có sự cải thiện đáng kể so với khi chưa sử dụng Prompt Engineering.

Phân tích chi tiết theo từng loại câu hỏi:

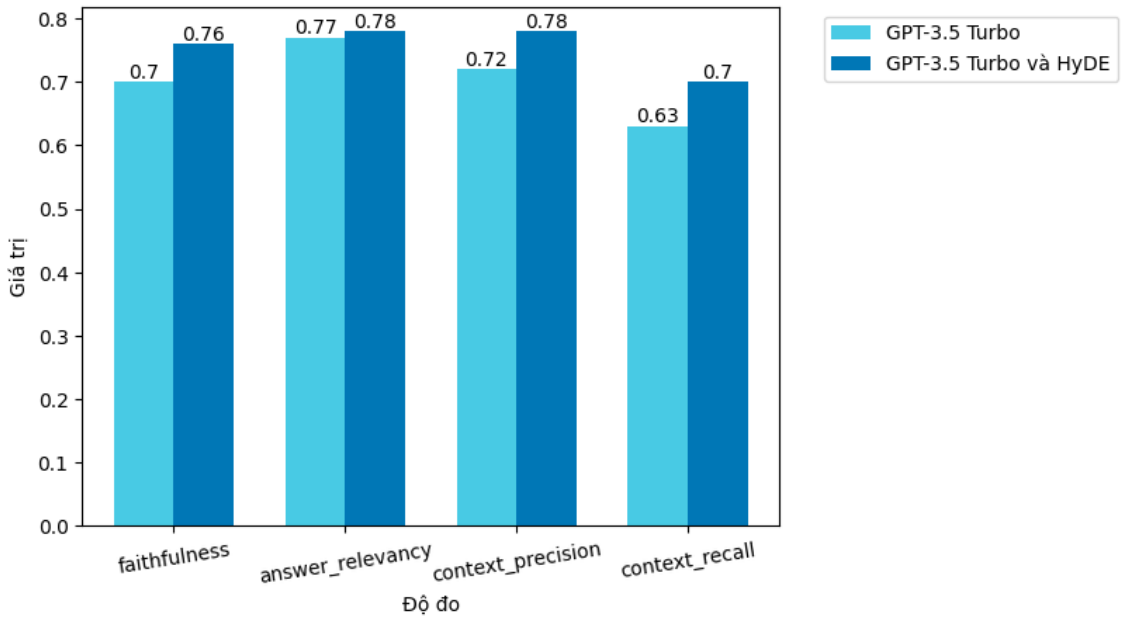
- Single-Context:

  - Điểm mạnh: Hệ thống đạt điểm cao ở hầu hết các metric, đặc biệt là answer_relevancy (0.82) và context_precision (0.82), cho thấy Prompt Engineering giúp GPT-3.5 Turbo hiểu rõ hơn yêu cầu của câu hỏi đơn giản, từ đó trích xuất và sử dụng thông tin chính xác hơn.
  - Hạn chế: Điểm faithfulness (0.68) giảm nhẹ so với khi chưa sử dụng Prompt Engineering, cho thấy kỹ thuật này có thể khiến câu trả lời đôi khi đi lệch một chút so với thông tin gốc, mặc dù vẫn đảm bảo liên quan.

- Multi-Context:
  - Điểm mạnh: Mặc dù vẫn là loại câu hỏi khó nhất nhưng hệ thống đã có sự tiến bộ rõ rệt so với khi chưa sử dụng Prompt Engineering, đặc biệt là answer_relevancy (0.91) và context_precision (0.8) tăng lên đáng kể. Điều này chứng tỏ Prompt Engineering giúp hệ thống xác định và kết nối thông tin liên quan từ nhiều nguồn hiệu quả hơn.
  - Hạn chế: Điểm faithfulness (0.54) và context_recall (0.56) vẫn còn thấp, cho thấy việc kết hợp thông tin từ nhiều nguồn một cách chính xác và đầy đủ vẫn là một thách thức đối với hệ thống, ngay cả khi đã sử dụng Prompt Engineering.

### 4.4.3 Thí nghiệm 3: Đánh giá hệ thống sử dụng RAG kết hợp GPT-3.5 Turbo và HyDE

Khi đánh giá hệ thống kết hợp mô hình GPT-3.5 Turbo với phương pháp HyDE cho ra các giá trị độ đo faithfulness = 0.76, answer_relevancy = 0.78, context_precision = 0.79 và context_recall = 0.69. Các giá trị đồ đo đều được cải thiện tốt hơn so với thời điểm ban đầu. Việc sử dụng HyDE đã cung cấp thêm ngữ cảnh và thông tin cho hệ thống bằng việc sinh ra một câu trả lời giả định từ câu hỏi. Điều này cho thấy việc kết hợp mô hình GPT-3.5 Turbo và phương pháp cải tiến HyDE làm tăng độ chính xác ngữ cảnh, tăng khả năng truy xuất của ngữ cảnh, mức độ đầy đủ của thông tin khi đưa vào ngữ cảnh được đầy đủ hơn. Các câu trả lời được sinh ra có độ tin cây cao.




Hình 4.4: So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo và Hypothetical Document Embeddings (HyDE)

Dưới đây là kết quả đánh giá của từng loại câu hỏi:

| **Phân loại** | **Faithfulness** | **Answer Relevancy** | **Context Precision** | **Context Recall** | **Tổng** |
|---|---|---|---|---|---|
| Single-Conext | 0.81 | 0.74 | 0.81 | 0.67 | 3.04 |
| Multi-Context | 0.58 | 0.89 | 0.69 | 0.79 | 2.95 |
| Tổng thể | 0.76 | 0.78 | 0.78 | 0.70 | 3.02 |

Bảng 4.4: Bảng đánh giá phân loại câu hỏi của RAG kết hợp GPT-3.5 Turbo và HyDE

Tổng quan về phân loại các câu hỏi:

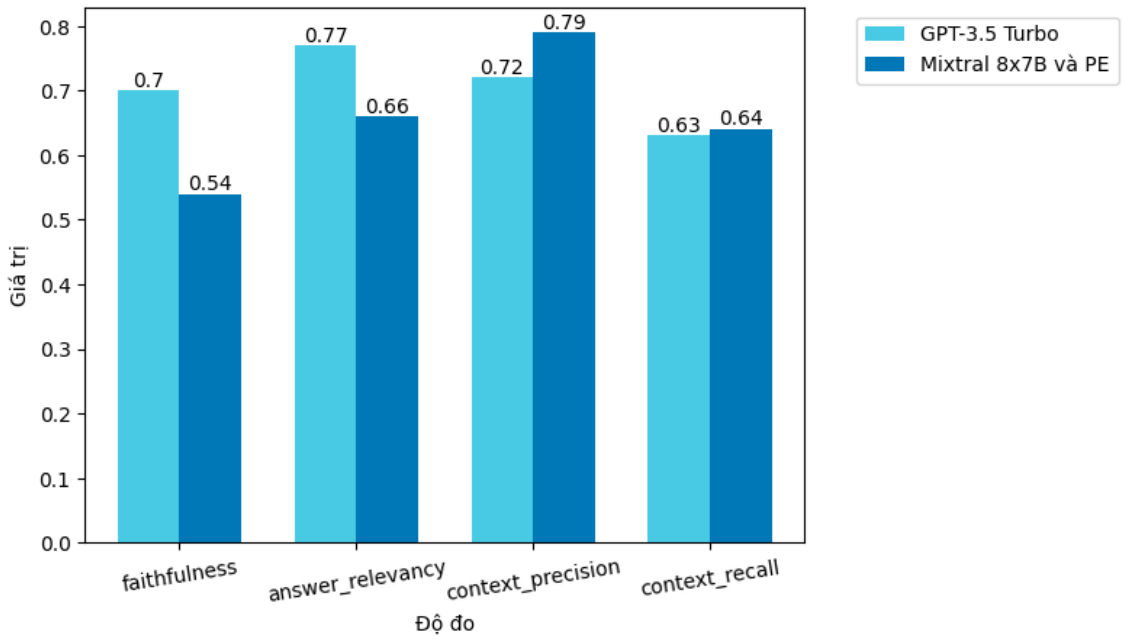
- HyDE chứng minh được hiệu quả rõ rệt khi kết hợp với GPT-3.5 Turbo, giúp hệ thống đạt điểm tổng thể cao nhất (3.02) so với các thí nghiệm trước đó.
- Hệ thống xử lý tốt nhất với câu hỏi Single-Context (3.04), cho thấy HyDE rất hiệu quả trong việc truy xuất thông tin chính xác và đầy đủ cho các câu hỏi trực tiếp. đặc biệt có thể khẳng định rằng GPT-3.5 Turbo cho ra kết quả hiệu quả đối với loại câu hỏi mang tính truy vấn trực tiếp.

Phân tích chi tiết theo từng loại câu hỏi:

- Single-Context:
  - Điểm mạnh: Hệ thống đạt điểm cao nhất ở hầu hết các metric, đặc biệt faithfulness (0.81) và context_precision (0.81), cho thấy HyDE giúp GPT-3.5 Turbo tạo ra câu trả lời đáng tin cậy, bám sát nguồn dữ liệu và trích dẫn thông tin chính xác.
  - Hạn chế: Điểm answer_relevancy (0.74) thấp hơn một chút so với hai phương pháp trước đó, cho thấy HyDE đôi khi có thể tập trung quá nhiều vào độ chính xác của thông tin mà chưa tối ưu hóa hoàn toàn cho độ liên quan đến câu hỏi.
- Multi-Context:
  - Điểm mạnh: Hệ thống đạt điểm context_recall (0.79) cao, cho thấy HyDE hỗ trợ hiệu quả cho GPT-3.5 Turbo trong việc truy xuất đầy đủ thông tin từ nhiều nguồn khác nhau.
  - Hạn chế: Mặc dù answer_relevancy là độ đo có giá trị cao nhất tuy nhiên kết quả cho thấy nó đã xuất hiện trạng thái giảm so với các thí nghiệm trước. Nguyên nhân có thể đến từ việc các câu trả lời được sinh ra dư thừa so với yêu cầu.

### 4.4.4 Thí nghiệm 4: Đánh giá hệ thống sử dụng RAG kết hợp Mixral 8x7B và Prompt Engineering

Khi đánh giá hệ thống kết hợp mô hình Mixtral 8x7B với kỹ thuật Prompt Engineering cho ra các giá trị độ đo: context_recall = 0.64 và context_precision = 0.79 đạt kết quả khả quan hơn so với thời điểm chưa cải tiến, có thể thấy việc đưa thêm lời nhắc vào mô hình đã khiến cho việc truy xuất được ngữ cảnh liên quan đến ground_truth được cải thiện. Mô hình đã tăng khả năng truy xuất chính xác ngữ cảnh. Mặc khác các giá trị độ đo faithfulness xấp xỉ 0.54 cho thấy dù truy xuất được ngữ cảnh tốt hơn nhưng việc sinh ra câu trả lời tin cậy lại tệ hơn. Độ đo answer_relevancy có giá trị là 0.66 lại thấp hơn khi chưa cải tiến. Điều này có thể do Mixtral 8x7B là một mô hình không có quá nhiều trọng số so với GPT-3.5 Turbo, nên khả năng xử lý ngôn ngữ của nó còn nhiều hạn chế.




Hình 4.5: So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo với kỹ thuật RAG kết hợp Mixtral 8x7B và Prompt Engineering

Bảng dưới đây là kết quả đánh giá của từng loại câu hỏi:

| **Phân loại** | **Faithfulness** | **Answer Relevancy** | **Context Precision** | **Context Recall** | **Tổng** |
|---|---|---|---|---|---|
| Single-Context | 0.51 | 0.61 | 0.76 | 0.62 | 2.50 |
| Multi-Context | 0.64 | 0.81 | 0.91 | 0.64 | 3.00 |
| Tổng thể | 0.54 | 0.66 | 0.79 | 0.64 | 2.63 |

Bảng 4.5: Bảng đánh giá phân loại câu hỏi của RAG kết hợp Mixtral 8x7B và Prompt Engineering

Tổng quan về phân loại các câu hỏi:

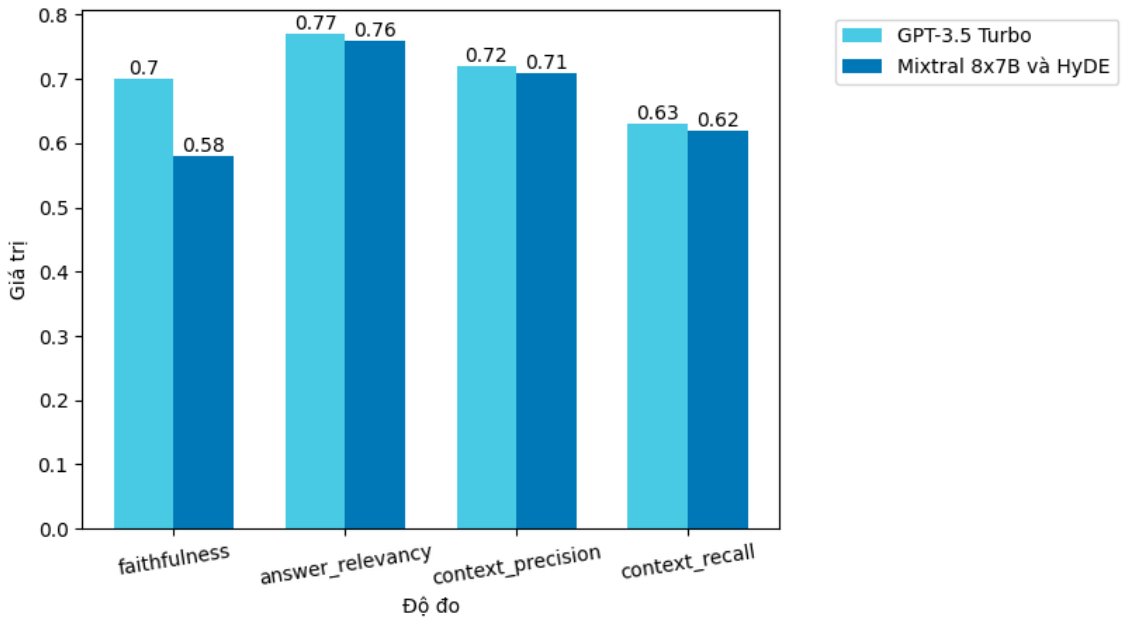
- Hệ thống kết hợp mô hình Mixtral 8x7B và kỹ thuật Prompt Engineering cho thấy hiệu suất tổng thể (2.63 điểm) thấp hơn so với GPT-3.5 Turbo trong các thí nghiệm trước.
- Hệ thống xử lý tốt nhất với câu hỏi Multi-Context (3.0 điểm), cho thấy tiềm năng của Mixtral 8x7B trong việc kết hợp thông tin từ nhiều nguồn khi được hỗ trợ bởi Prompt Engineering.

Phân tích chi tiết theo từng loại câu hỏi:

- Single-Context:
  - Điểm mạnh: Prompt Engineering giúp cải thiện đáng kể context_precision (0.76) cho thấy hệ thống đã trích dẫn thông tin chính xác hơn.
  - Hạn chế: Điểm faithfulness (0.51) và answer_relevancy (0.62) thấp cho thấy hệ thống vẫn còn hạn chế trong việc tạo ra câu trả lời đáng tin cậy và liên quan đến câu hỏi đơn giản, mặc dù đã sử dụng Prompt Engineering.
- Multi-Context:
  - Điểm mạnh: Hệ thống đạt điểm cao nhất ở context_precision (0.91), cho thấy khả năng trích dẫn thông tin chính xác từ nhiều nguồn khi được hỗ trợ bởi Prompt Engineering.
  - Hạn chế: Điểm faithfulness (0.64) và context_recall (0.64) tuy đã được cải thiện nhưng vẫn ở mức trung bình. Điều này cho thấy hệ thống vẫn cần cải thiện khả năng tổng hợp thông tin chính xác và đầy đủ từ nhiều nguồn khác nhau.

### 4.4.5 Thí nghiệm 5: Đánh giá hệ thống sử RAG kết hợp Mixtral 8x7B và HyDE

Khi đánh giá hệ thống kết hợp mô hình Mixtral 8x7B với sử kỹ thuật HyDE cho ra các giá trị độ đo kém so với thời điểm chưa cải tiến. Kể cả khi có áp dụng thêm HyDE để làm tăng hiệu suất thì nó cũng không đạt được mức độ tương đương so với GPT-3.5 Turbo. Điều này càng chắc chắn mô hình ngôn ngữ GPT-3.5 Turbo có khả năng xử lý và hiểu tốt hơn so với mô hình Mixtral 8x7B.




Hình 4.6: So sánh kỹ thuật RAG kết hợp GPT-3.5 Turbo với kỹ thuật RAG kết hợp Mixtral 8x7B và HyDE

Bảng dưới đây là kết quả đánh giá của từng loại câu hỏi:

| **Phân loại** | **Faithfulness** | **Answer Relevancy** | **Context Precision** | **Context Recall** | **Tổng** |
|---|---|---|---|---|---|
| Single-Context | 0.56 | 0.75 | 0.67 | 0.64 | 2.62 |
| Multi-Context | 0.67 | 0.80 | 0.80 | 0.52 | 2.79 |
| Tổng thể | 0.58 | 0.76 | 0.71 | 0.62 | 2.67 |

Bảng 4.6: Bảng đánh giá phân loại câu hỏi của RAG kết hợp Mixtral 8x7B và HyDE

Tổng quan về phân loại các câu hỏi:

- Tương tự như kết quả khi kết hợp với Prompt Engineering, Mixtral 8x7B kết hợp HyDE cho thấy hiệu suất tổng thể (2.67) vẫn chưa đạt được mức lý tưởng so với các thí nghiệm sử dụng GPT-3.5 Turbo.
- Hệ thống xử lý tốt nhất với câu hỏi Multi-Context (2.79)điều này càng chắc chắn rằng mô hình ngôn ngữ Mixtral 8x7B có tiềm năng trong việc kết hợp thông tin từ nhiều nguồn

Phân tích chi tiết theo từng loại câu hỏi:

- Single-Context:
  - Điểm mạnh: HyDE giúp cải thiện đáng kể answer_relevancy (0.75) so với Mixtral 8x7B kết hợp với Prompt Engineering. Điều này cho thấy HyDE hỗ trợ mô hình trong việc tạo ra câu trả lời đáng tin cậy hơn.
  - Hạn chế: Điểm faithfulness (0.56) ở mức trung bình, cho thấy hệ thống vẫn còn hạn chế trong việc đảm bảo độ tin cậy cho câu trả lời, mặc dù đã sử dụng HyDE.
- Multi-Context:
  - Điểm mạnh: Điểm answer_relevancy (0.80) và context_precision (0.80) cao cho thấy HyDE hỗ trợ Mixtral 8x7B trích dẫn thông tin chính xác từ nhiều nguồn khác nhau và câu trả lời có độ liên quan cao.
  - Hạn chế: Điểm context_recall (0.52) chỉ ở mức trung bình, cho thấy hệ thống vẫn còn hạn chế trong việc tổng hợp thông tin một cách chính xác và đầy đủ từ nhiều nguồn dữ liệu.

# Chương 5

# Tổng kết và hướng phát triển

## 5.1 Tổng kết

Dưới đây là bảng tổng hợp kết quả của các thí nghiệm đánh giá hệ thống hỏi đáp thông tin các trường đại học:

| **Thí nghiệm** | **Faithfulness** | **Answer Relevancy** | **Context Precision** | **Context Recall** | **Tổng** |
|---|---|---|---|---|---|
| RAG với GPT-3.5 Turbo | 0.7 | 0.77 | 0.72 | 0.63 | 2.82 |
| RAG với GPT-3.5 Turbo và Prompt Engineering | 0.65 | **0.84** | **0.81** | 0.68 | 2.98 |
| RAG với GPT-3.5 Turbo và HyDE | **0.76** | 0.78 | 0.78 | **0.7** | **3.02** |
| RAG với Mixtral 8x7B và Prompt Engineering | 0.54 | 0.66 | 0.79 | 0.64 | 2.63 |
| RAG với Mixtral 8x7B và HyDE | 0.58 | 0.76 | 0.71 | 0.62 | 2.67 |

Bảng 5.1: Kết quả thí nghiệm với các mô hình và kỹ thuật khác nhau

Từ bảng kết quả trên, chúng em nhận thấy rằng mô hình GPT-3.5 Turbo có hiệu quả vượt trội so với Mixtral 8x7B. Điều này dễ hiểu khi GPT-3.5 Turbo là một mô hình lớn hơn rất nhiều so với Mixtral 8x7B nên xử lý thông tin và ngữ cảnh tốt hơn. Bên cạnh đó, Prompt Engineering và phương pháp HyDE đều cho thấy hiệu quả rõ rệt trong việc cải thiện chất lượng truy xuất của hệ thống. Với Prompt Engineering, việc cung cấp lời nhắc cụ thể và rõ ràng giúp cho mô hình hiểu chính xác hơn khi tìm ngữ cảnh liên quan từ đó tạo sinh ra câu trả lời đầy đủ và chi tiết giúp độ đo Context Precision và Answer Relevancy được cải thiện nâng cao đáng kể. Còn với phương pháp HyDE, việc có câu trả lời giả định giúp hệ thống xác định đúng hơn các đoạn ngữ cảnh liên quan cũng như dễ dàng trong việc tìm kiếm câu trả lời tương đồng từ đoạn ngữ cảnh được truy xuất qua đó làm tăng độ đo Context Recall và Faithfulness.

Như vậy, trong khóa luận này, chúng em đã tìm hiểu về kỹ thuật Retrieval-Augmented Generation (RAG) để giải quyết bài toán xây dựng hệ thống hỏi đáp thông tin các trường đại học. Kỹ thuật này có nhiều ưu điểm như:

- **Hiệu quả cao:** Qua các thí nghiệm mà chúng em đã trình bày ở Chương 4, bằng việc kết hợp với các kỹ thuật cải tiến khả năng truy xuất như Prompt Engineering, HyDE,... có thể thấy kỹ thuật RAG không chỉ mang lại hiệu quả cao trong việc truy vấn thông tin mà các câu trả lời còn được cá nhân hóa với từng người dùng.

- **Khả năng xử lý câu hỏi mở:** Khác với các hệ thống Rule-Based truyền thống thường gặp khó khăn trong việc xử lý các câu hỏi phức tạp, kỹ thuật RAG thể hiện sự linh hoạt trong việc hiểu và trả lời các câu hỏi mở, không được định nghĩa trước. Kỹ thuật có khả năng phân tích ngữ nghĩa, kết nối thông tin từ nhiều nguồn khác nhau để đưa ra câu trả lời chính xác cho những câu hỏi đòi hỏi suy luận hoặc tổng hợp kiến thức.

- **Dễ dàng cài đặt và thực hiện:** Kỹ thuật RAG có thể được cài đặt một cách dễ dàng bằng cách sử dụng các thư viện và công cụ mã nguồn mở có sẵn, giúp rút ngắn thời gian phát triển hệ thống.

Ngoài những ưu điểm trên, kỹ thuật RAG cũng tồn tại những hạn chế nhất định:

- **Khó khăn trong việc xử lý tri thức ngầm:** Do phụ thuộc chủ yếu vào thông tin được cung cấp rõ ràng trong cơ sở dữ liệu, RAG gặp khó khăn trong việc suy luận, kết nối các thông tin ẩn ý, điều mà con người thực hiện một cách tự nhiên. Ví dụ, RAG có thể không tự suy luận được rằng "ứng viên có điểm trung bình cao thường có cơ hội trúng tuyển lớn hơn", hoặc "ngành Khoa học Máy tính thường có điểm chuẩn cao hơn ngành Ngôn ngữ Anh", nếu những thông tin này không được đề cập trực tiếp trong cơ sở dữ liệu. Điều này đặt ra thách thức cho việc xây dựng hệ thống hỏi đáp có khả năng "hiểu" và trả lời linh hoạt như con người.
- **Yêu cầu cao về tài nguyên tính toán:** Việc xử lý các mô hình ngôn ngữ lớn (LLM) cho cả giai đoạn truy xuất và tạo sinh văn bản đòi hỏi hệ thống có dung lượng bộ nhớ lớn để lưu trữ mô hình, cũng như khả năng xử lý mạnh mẽ để thực hiện các phép tính phức tạp trong thời gian thực. Điều này có thể gây khó khăn cho việc triển khai và duy trì hệ thống, đặc biệt là trên các thiết bị có tài nguyên hạn chế hoặc trong môi trường yêu cầu tốc độ phản hồi nhanh.

## 5.2 Hướng phát triển

Đối với hướng phát triển trong tương lai, chúng em dự định ứng dụng các kỹ thuật nâng cao giúp cải thiện thêm khả năng truy xuất của hệ thống như *Sentence Window Retrieval, Auto Merging Retriever,...* Ngoài ra, một hướng phát triển khác là mở rộng tập dữ liệu với thông tin các trường đại học, cao đẳng, học viện trên địa bàn Thành phố Hồ Chí Minh. Từ đó, phát triển hoàn thiện hệ thống hỏi đáp thông tin các trường đại học nhằm cung cấp thông tin cần thiết cho học sinh đưa ra lựa chọn phù hợp cho tương lai.

# Tài liệu tham khảo


[1] Quinn Leng, Kasey Uhlenhuth, Alkis Polyzotis. *Best Practices for LLM Evaluation of RAG Applications.* [September 12, 2023 in Machine Learning]

[2] Leonie Monigatti. *Evaluating RAG Applications with RAGAs.* [ Dec 13, 2023]

[3] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, Zhaofeng Liu. *Evaluation of Retrieval-Augmented Generation: A Survey.* [Submitted on 13 May 2024 (v1), last revised 3 Jul 2024 (this version, v2)]

[4] Apoorva Joshiu. *How to Choose the Right Chunking Strategy for Your LLM Application.* [Jun 18, 2024]

[5] Ravi Theja. *Evaluate RAG with LlamaIndex.* [Nov 6, 2023]

[6] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang. *Retrieval-Augmented Generation for Large Language Models: A Survey.* [Submitted on 18 Dec 2023 (v1), last revised 27 Mar 2024 (this version, v5)]

[7] Plaban Nayak. *Advanced RAG — Improving retrieval using Hypothetical Document Embeddings(HyDE).* [Nov 4, 2023]

[8] Coursera Staff. *What Is Prompt Engineering? Definition and Examples.* [Mar 20, 2024]

[9] `https://docs.ragas.io/en/latest/concepts/metrics/index.html`

[10] `https://myscale.com/blog/prompt-engineering-vs-finetuning-vs-rag/`

[11] `https://2ocs.haystack.deepset.ai/docs/hypothetical-document-embeddings-hyde`

[12] `https://www.aporia.com/learn/enhance-rags-hyde/`