ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC KINH TẾ

BÁO CÁO TỔNG KẾT

ĐỀ ÁN THỰC HÀNH 1

SMART ENROLLMENT ASSISTANCE: A PREDICTIVE AI CHATBOT FOR UNIVERSITY ADMISSION INSIGHTS AND DATA VISUALIZATION

Sinh viên thực hiện	Trần Tiến Đạt
	Nguyễn Minh Toàn
	Lê Văn Đức Tiến
Lớp	48K29.2
Giáo viên hướng dẫn	Th.S Trần Văn Lộc

Đà Nẵng, tháng 05 năm 2025

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC NHỮNG TỪ VIẾT TẮT

STT	Ký hiệu	Nguyên Nghĩa
1	AI	Artificial Intelligence – Trí tuệ nhân tạo
2	API	Application Programming Interface – Giao diện lập trình ứng dụng
3	CPU	Central Processing Unit – Bộ xử lý trung tâm
4	DUE	Danang University of Economics – Trường Đại học Kinh tế – Đại học Đà Nẵng
5	GPU	Graphics Processing Unit – Bộ xử lý đồ họa
6	HTML	HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản
7	JSON	JavaScript Object Notation – Định dạng dữ liệu nhẹ để trao đổi dữ liệu
8	LLM	Large Language Model – Mô hình ngôn ngữ lớn
9	LSTM	Long Short-Term Memory – Mạng nơ-ron hồi tiếp bộ nhớ dài hạn
10	MBTI	Myers–Briggs Type Indicator – Trắc nghiệm phân loại tính cách
11	MoE	Mixture of Experts – Cơ chế pha trộn chuyên gia trong mô hình AI
12	NLP	Natural Language Processing – Xử lý ngôn ngữ tự nhiên
13	PDF	Portable Document Format – Định dạng tài liệu di động
14	RAG	Retrieval-Augmented Generation – Mô hình sinh phản hồi có truy xuất dữ liệu
15	RAM	Random Access Memory – Bộ nhớ truy cập ngẫu nhiên
16	RNN	Recurrent Neural Network – Mạng nơ-ron hồi tiếp
17	SSD	Solid State Drive – Ổ cứng thể rắn
18	URL	Uniform Resource Locator – Địa chỉ định danh tài nguyên trên Internet

CHƯƠNG I. MỞ ĐẦU

1.1 Tính cấp thiết của đề tài

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong lĩnh vực giáo dục, nhu cầu tiếp cận thông tin tuyển sinh một cách chính xác, kịp thời và dễ hiểu đã trở thành một yêu cầu cấp thiết đối với học sinh và phụ huynh. Theo Ngân hàng Thế giới (2020), một trong những nguyên nhân khiến tỷ lệ nhập học đại học tại Việt Nam còn hạn chế là do thiếu một hệ thống thông tin tuyển sinh minh bạch, đồng bộ và dễ tiếp cận [1].

Thực tế, hệ thống tuyển sinh đại học tại Việt Nam đang ngày càng trở nên phức tạp với hơn 20 phương thức xét tuyển khác nhau được áp dụng đồng thời, cùng với sự đa dạng về tổ hợp môn thi và chỉ tiêu riêng cho từng ngành, từng trường [2]. Thông tin tuyển sinh hiện vẫn được công bố rời rạc trên các website riêng lẻ, thiếu sự tích hợp, khiến học sinh – đặc biệt là ở vùng sâu vùng xa (nơi chỉ khoảng 62% hộ gia đình có kết nối Internet ổn định – Tổng cục Thống kê, 2022) – gặp nhiều khó khăn trong việc tra cứu và ra quyết định.

Khảo sát của Bộ Giáo dục và Đào tạo (2023) cho thấy, hơn 65% học sinh cuối cấp THPT gặp khó khăn trong việc tiếp cận và so sánh các thông tin như điểm chuẩn, phương thức xét tuyển và chỉ tiêu tuyển sinh [3]. Điều này không chỉ tạo áp lực tâm lý trong mùa thi mà còn làm gia tăng nguy cơ chọn sai ngành, sai trường – ảnh hưởng nghiêm trọng đến định hướng nghề nghiệp lâu dài.

Mặt khác, sự gia tăng liên tục số lượng thí sinh đăng ký xét tuyển đại học cũng cho thấy nhu cầu học lên cao đang ngày càng rõ nét. Năm 2024, có hơn 733.000 thí sinh đăng ký xét tuyển đại học, chiếm 68,5% tổng số thí sinh dự thi tốt nghiệp THPT – tăng so với 65,9% của năm 2023 và 64,1% của năm 2022 [4]. Tuy nhiên, mỗi năm vẫn có khoảng 20% thí sinh trúng tuyển không nhập học và 5–7% sinh viên năm nhất phải đăng ký lại vì chọn sai ngành, sai trường [5]. Đây là minh chứng rõ ràng cho những hệ lụy thực tiễn nếu không có sự hỗ trợ tư vấn chính xác.

Trong bối cảnh đó, sự phát triển của công nghệ trí tuệ nhân tạo (AI), đặc biệt là các hệ thống xử lý ngôn ngữ tự nhiên (NLP) và mô hình ngôn ngữ lớn (LLMs), đã mở ra nhiều cơ hội ứng dụng trong giáo dục. Nổi bật trong số đó là mô hình Retrieval-Augmented Generation (RAG) – giải pháp giúp kết hợp khả năng truy xuất thông tin thực tế với khả năng sinh phản hồi ngôn ngữ tự nhiên, nhằm giảm thiểu hiện tượng "ảo

giác" và tăng độ chính xác trong tương tác [6]. Khác với các chatbot truyền thống vốn chỉ dựa vào văn bản có sẵn, chatbot sử dụng RAG có thể truy vấn cơ sở dữ liệu như điểm chuẩn, tổ hợp môn, chỉ tiêu, v.v... để tạo ra phản hồi đúng ngữ cảnh, cá nhân hóa theo nhu cầu người hỏi.

Thực tế, một số nghiên cứu quốc tế đã bắt đầu ứng dụng RAG trong chatbot giáo dục như "Retrieval-Augmented Generation Chatbots for Education" [7] - một khảo sát tổng hợp 47 nghiên cứu ứng dụng RAG trong giáo dục và đào tạo . Tại Việt Nam, một số mô hình chatbot như NEU-Chatbot (ĐH Kinh tế Quốc dân) hay chatbot Rasa dùng trên Facebook cũng đã được phát triển nhưng chủ yếu mang tính chất trả lời tĩnh, thiếu tính tương tác sâu, chưa tích hợp công nghệ RAG hoặc chưa mở rộng cho toàn ngành giáo dục đại học [8].

Điểm nổi bật và mới của đề tài là việc triển khai chatbot sử dụng mô hình RAG trong giao diện web, với khả năng truy xuất dữ liệu từ nhiều trường, nhiều năm, nhiều phương thức tuyển sinh, cung cấp kết quả có điều kiện lọc (năm – điểm – tổ hợp – phương thức) và phản hồi theo ngữ cảnh từng học sinh. Đặc biệt, hệ thống còn tích hợp bài trắc nghiệm MBTI (Myers–Briggs Type Indicator) – một trong những công cụ được sử dụng phổ biến nhất trong định hướng nghề nghiệp. Nhờ đó, học sinh không chỉ được cung cấp thông tin học thuật phù hợp mà còn được gợi ý nhóm ngành theo tính cách, giúp tăng tính chính xác và đồng bộ giữa năng lực, sở thích và hướng nghiệp.

Do đó, việc nghiên cứu và phát triển một hệ thống chatbot tư vấn tuyển sinh ứng dụng mô hình ngôn ngữ lớn là cần thiết và cấp bách, nhằm cung cấp thông tin chính xác, kịp thời và cá nhân hóa cho học sinh, hỗ trợ họ trong quá trình ra quyết định quan trọng về giáo dục và nghề nghiệp.

1.2 Mục tiêu đề tài.

1.2.1 Mục tiêu về lý thuyết.

Đề tài hướng đến việc nghiên cứu và hệ thống hóa cơ sở lý thuyết liên quan đến mô hình ngôn ngữ lớn và khả năng ứng dụng của nó trong lĩnh vực tư vấn giáo dục. Trên cơ sở đó, nhóm nghiên cứu tập trung làm rõ các nội dung sau:

- Phân tích cấu trúc và cơ chế hoạt động của mô hình RAG trong mối liên hệ với các tác vụ xử lý ngôn ngữ tự nhiên, đặc biệt trong bài toán sinh phản hồi có điều hướng tri thức.

● Nghiên cứu đặc điểm ngôn ngữ, hành vi truy vấn và nhu cầu thông tin của người dùng trong bối cảnh tuyển sinh đại học tại Việt Nam.

● Đề xuất một kiến trúc tổng thể cho hệ thống chatbot tư vấn tuyển sinh dựa trên nền tảng RAG, có khả năng tích hợp dữ liệu từ nhiều nguồn và xử lý ngôn ngữ tiếng Việt một cách chính xác và tự nhiên.

● Góp phần mở rộng nền tảng lý luận về ứng dụng của các mô hình AI hiện đại trong các bài toán giáo dục có tính thực tiễn cao.

1.2.2 Mục tiêu về ứng dụng

Bên cạnh mục tiêu lý thuyết, đề tài còn đặt trọng tâm vào việc triển khai một sản phẩm ứng dụng thực tế, phục vụ công tác tư vấn tuyển sinh cho học sinh trung học phổ thông. Cụ thể:

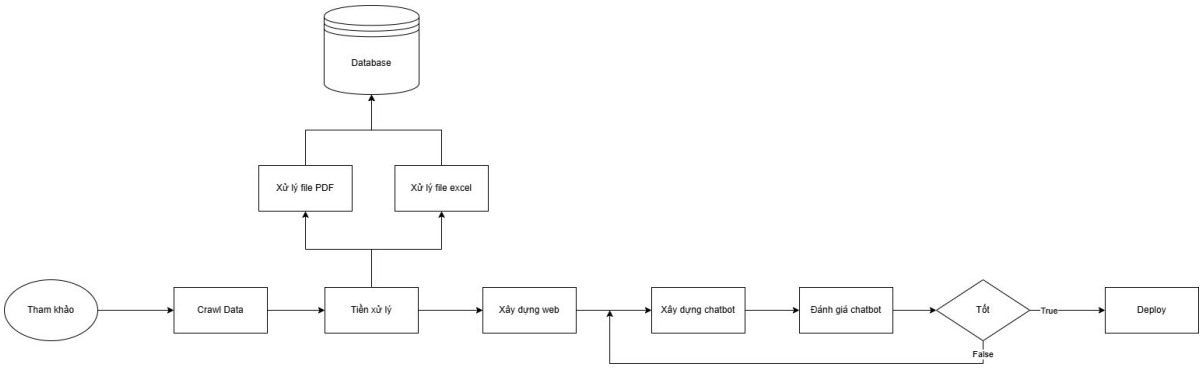
● Xây dựng và triển khai một hệ thống chatbot tư vấn tuyển sinh tự động, ứng dụng mô hình ngôn ngữ lớn (LLM) kết hợp với kiến trúc RAG để truy xuất và cung cấp thông tin về ngành học, phương thức tuyển sinh, điểm chuẩn, chỉ tiêu và học phí của các trường đại học một cách nhanh chóng, chính xác và cá nhân hóa.

● Tích hợp cơ sở dữ liệu tuyển sinh từ các trường thành viên của Đại học Đà Nẵng giai đoạn 2018–2024 vào hệ thống chatbot nhằm đảm bảo độ tin cậy và tính đầy đủ của thông tin cung cấp.

● Phát triển giao diện thân thiện, cho phép người dùng tương tác với chatbot mọi lúc, mọi nơi, đồng thời hỗ trợ gợi ý ngành học phù hợp dựa trên kết quả bài trắc nghiệm tính cách MBTI và dữ liệu điểm thi đầu vào.

● Đánh giá hiệu quả của hệ thống thông qua các chỉ số định lượng như độ chính xác phản hồi, tốc độ truy vấn, mức độ hài lòng của người dùng, cũng như khả năng mở rộng cho các đơn vị giáo dục khác trong tương lai.

1.3 Phương pháp nghiên cứu



Hình 1: Phương pháp nghiên cứu

1.4 Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống chatbot tư vấn tuyển sinh tích hợp trí tuệ nhân tạo, được xây dựng dựa trên mô hình ngôn ngữ lớn (Large Language Model – LLM) kết hợp với kiến trúc Retrieval-Augmented Generation (RAG).

Hệ thống sử dụng LLM để sinh phản hồi bằng ngôn ngữ tự nhiên, đồng thời áp dụng RAG để truy xuất dữ liệu tuyển sinh từ nguồn thông tin thực tế, nhằm đảm bảo độ chính xác và tính cá nhân hóa cao trong câu trả lời.

Ngoài ra, đề tài cũng quan tâm đến hành vi tương tác của người dùng (học sinh và phụ huynh) trong quá trình truy vấn thông tin qua nền tảng chatbot, cũng như khả năng xử lý và sinh phản hồi bằng tiếng Việt, đáp ứng yêu cầu ngữ nghĩa và ngữ cảnh trong giao tiếp.

1.4.2. Phạm vi nghiên cứu

- **Về nội dung**: Đề tài tập trung vào việc thiết kế, xây dựng và đánh giá một hệ thống chatbot tư vấn tuyển sinh ứng dụng mô hình ngôn ngữ lớn (LLM) kết hợp với kiến trúc Retrieval-Augmented Generation (RAG), với khả năng truy xuất và sinh câu trả lời từ cơ sở dữ liệu tuyển sinh của các trường đại học. Hệ thống được thiết kế để hỗ trợ người dùng tìm kiếm thông tin về ngành học, điểm chuẩn, phương thức tuyển sinh, chỉ tiêu, học phí và gợi ý ngành học theo MBTI.
- **Về dữ liệu**: Cơ sở dữ liệu sử dụng trong nghiên cứu bao gồm thông tin tuyển sinh từ năm 2018 đến năm 2024 của các trường thành viên thuộc Đại học Đà Nẵng. Ngoài ra, còn có các dữ liệu nội bộ cũng như thông tin tuyển sinh nội bộ của trường Đại học Kinh tế Đà Nẵng.
- **Về kỹ thuật**: Phạm vi nghiên cứu bao gồm việc triển khai kiến trúc Retrieval-Augmented Generation (RAG) trong môi trường xử lý tiếng Việt, tích hợp mô hình vào giao diện chatbot, và đo lường hiệu quả thông qua các chỉ số định lượng và định tính (chính xác phản hồi, mức độ hài lòng, độ trễ thời gian...).
- **Về không gian – thời gian**: Nghiên cứu được triển khai thử nghiệm tại khu vực miền Trung, với đối tượng sử dụng chính là học sinh lớp 12, phụ huynh học sinh có nhu cầu tham khảo cũng như muốn nắm rõ hơn thông tin về tuyển sinh trước khi đăng kí tuyển sinh vào trường đại học.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1. Kiến trúc Transformer

2.1.1. Giới thiệu kiến trúc Transformer

Trong bối cảnh sự phát triển mạnh mẽ của các mô hình ngôn ngữ lớn (LLM), nhu cầu phát hiện văn bản do AI sinh ra trở nên ngày càng cấp thiết. Gần đây, một công cụ phát hiện văn bản AI dựa trên kiến trúc Transformer đã được phát triển và chứng minh hiệu quả rõ rệt trong việc nâng cao độ chính xác của nhiệm vụ này, đồng thời cung cấp giá trị tham khảo quan trọng cho các nghiên cứu tiếp theo trong lĩnh vực phát hiện nội dung do AI tạo ra [9].

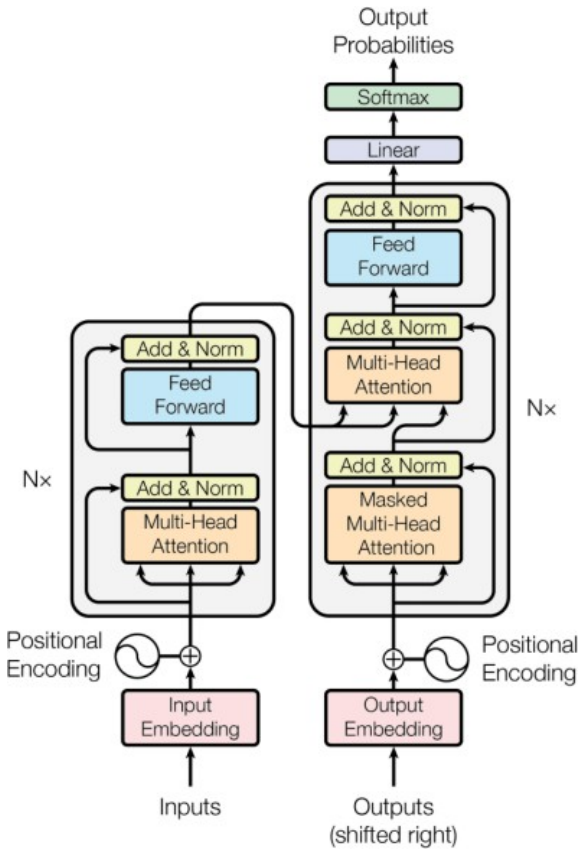
Trước khi Transformer ra đời, các phương pháp mô hình hóa chuỗi như mạng nơ-ron hồi tiếp (RNN), mạng bộ nhớ dài hạn (LSTM) và mạng hồi tiếp có cổng (GRU) đã đạt được thành công đáng kể trong các tác vụ như mô hình ngôn ngữ và dịch máy. Tuy nhiên, những mô hình này gặp phải giới hạn cố hữu do tính chất tính toán tuần tự, làm giảm khả năng song song hóa và ảnh hưởng đến hiệu suất xử lý với các chuỗi dài. Mặc dù cơ chế attention đã phần nào cải thiện vấn đề này, nó vẫn thường được sử dụng kết hợp với mạng hồi tiếp, khiến các rào cản về tính toán vẫn chưa được giải quyết triệt để [10].

Trong nghiên cứu này, chúng tôi giới thiệu Transformer — một kiến trúc mới hoàn toàn loại bỏ thành phần hồi tiếp và chỉ dựa vào cơ chế self-attention để mô hình hóa các phụ thuộc toàn cục trong chuỗi. Thiết kế này không những giúp tăng khả năng song song hóa trong quá trình huấn luyện mà còn cải thiện hiệu quả học các mối quan hệ xa, từ đó mở ra một hướng tiếp cận mới cho các bài toán chuyển đổi chuỗi trong xử lý ngôn ngữ tự nhiên [10].

Kiến trúc của transformer gồm 2 phần chính là encoders (là 1 ngăn xếp gồm 6 khối encoder kiến trúc giống nhau) và decoders (là 1 ngăn xếp gồm 6 khối decoder giống nhau) [10].

- Mỗi khối encoder có 2 layer chính: self-attention và feed forward [10].
- Mỗi khối decoder có 3 layer chính: sef-attention, encoder-decoder attention và feed forward [10].

2.1.2 Kiến trúc của mô hình



Hình 2: Kiến trúc tổng quát của mô hình Transformer gồm Encoder và Decoder

Hầu hết các mô hình chuyển đổi chuỗi bằng mạng nơ-ron hiện đại đều tuân theo cấu trúc encoder-decoder [5, 2, 35]. Trong cấu trúc này, encoder có nhiệm vụ chuyển đổi chuỗi đầu vào gồm các biểu diễn ký hiệu $(x_1, ..., x_n)$ thành một chuỗi biểu diễn liên tục $z = (z_1, ..., z_n)$. Dựa trên chuỗi biểu diễn này, decoder sẽ tạo ra chuỗi đầu ra $(y_1, ..., y_n)$ từng phần tử một.

Mô hình này hoạt động theo nguyên tắc tự hồi quy (auto-regressive) , tức là mỗi bước tạo ra một ký hiệu mới sẽ sử dụng các ký hiệu trước đó làm đầu vào bổ sung

Transformer tuân theo kiến trúc trên, nhưng thay vì dùng mạng hồi tiếp như RNN hay LSTM, mô hình sử dụng các lớp tự chú ý chồng lên nhau (stacked self-attention) và các lớp kết nối đầy đủ theo điểm (point-wise fully connected layers) cho cả phần encoder và decoder, như minh họa trong Hình 1 (bên trái là encoder, bên phải là decoder) [10].

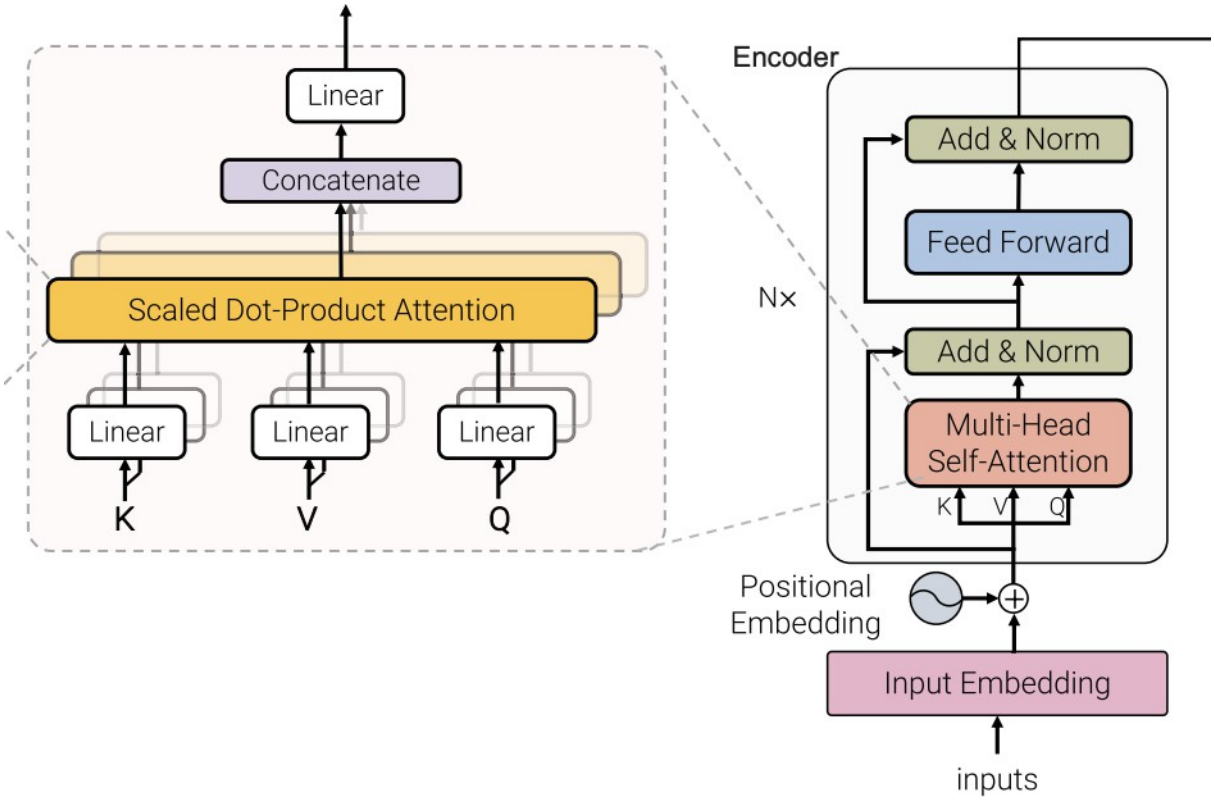
2.1.3. Kiến trúc Khối encoder

2.1.3.1. Input embedding

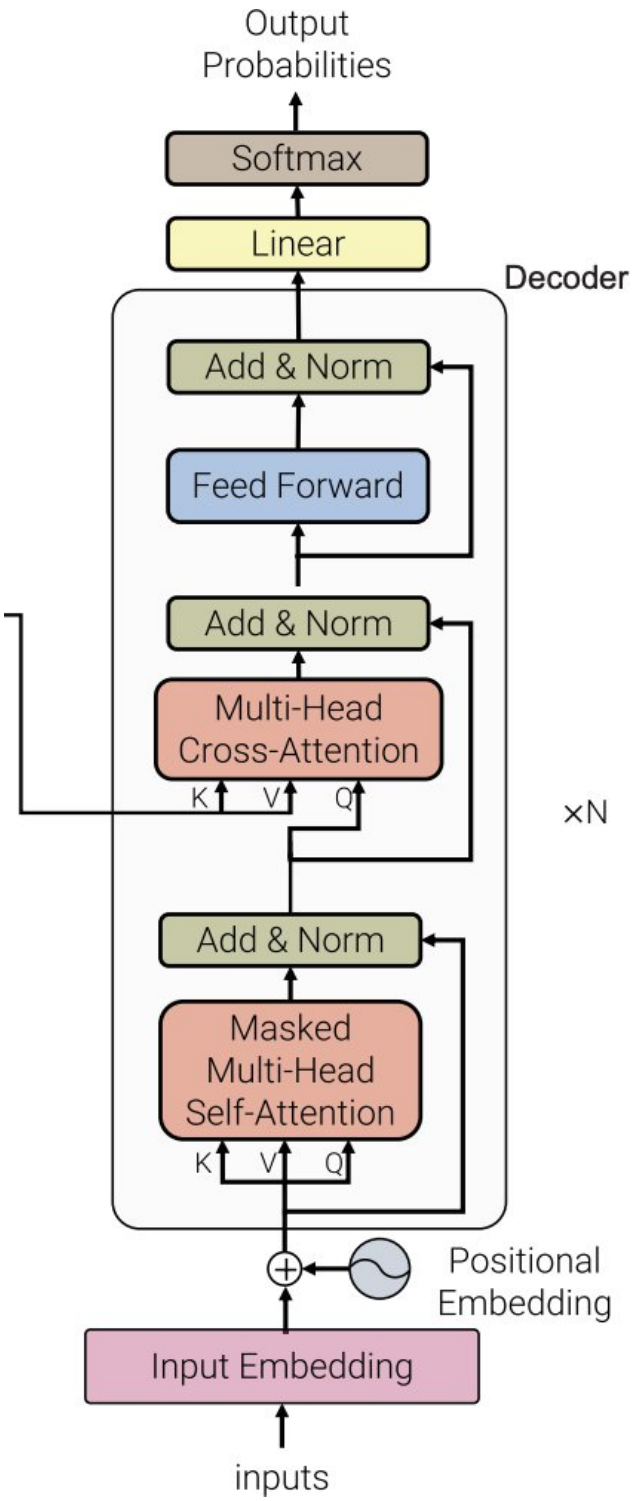
Tương tự như các mô hình biến đổi chuỗi khác, mô hình Transfomer sử dụng các vector embedding học được để chuyển đổi các token đầu vào và đầu ra thành các vector có cùng số chiều là d_{model}. Tiếp đến đầu vào của khối Decorder là đầu ra của khối Encoder. Bên cạnh vector word embedding thông thường thì kiến trúc Transfomer còn kết hợp thêm cả Positional Encoding. Trong đó:

- Positional Encoding: là một kỹ thuật thêm thông tin vị trí vào vector biểu diễn của mỗi token trong chuỗi, giúp mô hình Transformer có khả năng phân biệt vị trí của các token
- Vector word embedding là vector biểu diễn các từ được tạo ra từ các pre-model như word2vec, glove…

2.1.3.2. Multi-head self–attention



Hình 3: Cơ chế Self-Attention trong mô hình Transformer



Hình 4: Cấu trúc Decoder và cơ chế Multi-head Attention trong Transformer

Multi-Head Self-attention là một thành phần then chốt trong các mô hình Transformer. Cơ chế này giúp mỗi token trong chuỗi đầu vào có khả năng học cách tương tác và thu nhận thông tin từ các token khác trong chuỗi. Về bản chất, quá trình này được hiểu như một quá trình học cách liên kết giữa các phần tử trong chuỗi [11].

Giả sử ta có tập embedding đầu vào X với kích thước d_model. Thông qua các phép biến đổi tuyến tính, ta thu được:

- Truy vấn: $Q_S = X * W_S^q$
- Khóa: $K_S = X * W_S^k$
- Giá trị: $V_S = X * W_S^v$

Trong đó, W_S^q, W_S^q, W_S^v là các ma trận trọng số học được tương ứng. Với kích thước d_k của truy vấn và khóa, và d_v của giá trị, attention đơn đầu (single-head) thực hiện như sau:

$$A_S Q_S, K_S, V_S \bullet soft\max(\frac{Q_S K_S^T}{\sqrt{d_k}})V_S$$

Thay vì chỉ sử dụng một phép chiếu duy nhất cho tất cả các đầu vào, attention đa đầu chia input thành h phần. Mỗi phần được biến đổi thông qua một tập trọng số riêng biệt để tạo ra các phiên bản khác nhau của Q, K và V. Các phép attention được thực hiện song song trên từng phần riêng biệt. Sau đó, kết quả từ các đầu được nối lại và chiếu thông qua một ma trận tuyến tính cuối cùng để tạo thành đầu ra:

$A_h(X) = concat(head_1, ..., head_h) * W^o$

Với mỗi head_i được định nghĩa như sau:

$$head_i \bullet A_s(XW_i^q, XW_i^k, XW_i^v)$$

Trong đó:

- $W_i^q \in R^{(d_{model} \times d_k)}$
- $W_i^k \in R^{(d_{model} \times d_k)}$
- $W_i^v \in R^{(d_{model} \times d_k)}$

là các ma trận chiếu của đầu thứ i. $W^O \in R^{h.d_v \times d_{model}}$ là ma trận chiếu đầu ra sau khi các kết quả attention đã được nối lại. Thông thường, giá trị d_k và d_v được chọn sao cho: $d_k = d_v = d_{model} / h$ [11].

2.1.3.3 Position-wise Feed-forward Layers

Đầu ra từ mô-đun attention đa đầu sẽ được đưa qua một mạng lan truyền tiến gồm hai lớp tuyến tính. Trong đó, lớp ẩn sử dụng hàm kích hoạt ReLU. Điểm quan trọng là quá trình tính toán này được thực hiện độc lập tại từng vị trí trong chuỗi đầu vào - vì thế gọi là "position-wise".

Biểu thức của tầng lan truyền tiến như sau:

$FF(X_a) = ReLU(0, X_a \cdot W_1 + b_1) \cdot W_2 + b_2$

Với:

- $W_1 \in R^{d_{model} \times d_{ff}}$ và $W_2 \in R^{d_{model} \times d_{ff}}$ là trọng số của hai lớp tuyến tính
- $b_1 \in R^{d_{ff}}$ và $b_2 \in R^{d_{model}}$ là các vector dịch (bias) tương ứng.

Các tham số trên đóng vai trò học trong quá trình huấn luyện, giúp mạng học được cách chuyển đổi và biểu diễn đặc trưng của từng vị trí trong chuỗi đầu vào một cách hiệu quả [11].

2.1.3.4. Add & Normalize

Ở bước này, các vector đầu ra từ lớp con (multi-head self-attention và feed forward) sau đó qua bước dropout với tỉ lệ 0.1, rồi cộng thêm vector đầu vào (vector trước khi bị biến đổi), cuối cùng được normalized theo một công thức nào đó rồi chuyển vào layer kế tiếp. Ý nghĩa của bước này là để bổ sung thêm thông tin nguyên bản, tránh bị mất mát quá nhiều thông tin sau khi qua các phép biến đổi ở các layer multi-head self-attention và feed forward [10].

2.1.3.5. Đầu ra của khối encoder cuối cùng

Các vector sau khi qua lớp FFN của khối encoder cuối cùng sẽ được nhân với 2 ma trận trọng số K và V để tạo thành các cặp vector {(k1, v1), (k2, v2),...(kn, vn)} ứng với câu có n từ. Các vector này sẽ được dùng để tính vector biểu diễn z trong lớp encoder-decoder attention [10].

2.1.4. Kiến trúc của mỗi khối decoder

Các layer trong khối decoder được thiết kế tương tự khối encoder tuy nhiên có 1 số điểm khác biệt sau:

- Đầu vào của lớp sef-attention ở lần đầu tiên được là vector được tạo thành bởi embedding của 1 ký tự [start] + vector positonal embedding. Vector đầu vào ở các lần kế tiếp được tạo thành bởi vector output của layer FFN của khối decoder cuối cùng + vector positonal embedding.
- Lớp self-attention chỉ kết hợp thông tin từ các từ trước nó.
- Lớp encoder-decoder attention chỉ tính toán vector q dựa trên đầu ra của self-attention, vector k và v được lấy từ output của khối encoder.

- Việc tính toán được thực hiện cho đến khi decoder dự đoán được ký tự kết thúc [end]
- Đầu ra của lớp FFN cuối cùng sẽ được đi qua lớp Linear để biến đổi các vector này thành một vector có số chiều bằng số từ trong bộ vocabulary. Mỗi giá trị của 1 phần tử trong vector thể hiện điểm số cho 1 từ trong bộ từ vựng. Vector này sau đó được cho qua 1 hàm softmax để biến chúng thành một phân phối xác suất (tất cả phần tử >0 và tổng =1). Từ mà có xác suất cao nhất sẽ là từ được chọn [10].

2.1.5. Ứng dụng của kiến trúc Transformer vào mô hình GPT-4:

Sự phát triển của các mô hình ngôn ngữ lớn (Large Language Models – LLMs) trong những năm gần đây đã mở ra một kỷ nguyên mới cho các ứng dụng trí tuệ nhân tạo, đặc biệt trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Trong bối cảnh đó, GPT-4, được giới thiệu bởi OpenAI vào tháng 3 năm 2023, đánh dấu một bước tiến quan trọng về khả năng hiểu ngữ cảnh, lý luận đa chiều và tương tác ngôn ngữ tự nhiên có chiều sâu. Đây là mô hình thuộc họ Generative Pre-trained Transformer, được huấn luyện trên một tập dữ liệu quy mô lớn với cấu trúc Transformer mở rộng nhằm tối ưu khả năng tạo sinh văn bản, trả lời câu hỏi, viết mã, phân tích logic và xử lý đầu vào đa phương thức [12].

Không giống như các phiên bản tiền nhiệm như GPT-2 hay GPT-3, GPT-4 là một hệ thống đa phương thức (multimodal), có khả năng xử lý cả văn bản và hình ảnh làm đầu vào (input), đồng thời tạo ra phản hồi ở dạng văn bản (text output). Điều này cho phép mô hình không chỉ hiểu và phản hồi các câu hỏi dựa trên ngữ cảnh văn bản, mà còn có thể mô tả, phân tích hoặc diễn giải nội dung từ hình ảnh – một tính năng mở rộng đáng kể phạm vi ứng dụng trong giáo dục, khoa học dữ liệu, y tế, và trợ lý ảo cá nhân [12].

Một trong những điểm nổi bật của GPT-4 là hiệu suất vượt trội trên các bài kiểm tra tiêu chuẩn chuyên môn và học thuật, chẳng hạn như Uniform Bar Exam (kỳ thi luật Hoa Kỳ), GRE, SAT, hoặc các bộ tiêu chuẩn hiểu biết đa nhiệm như MMLU (Massive Multitask Language Understanding). Trong báo cáo của OpenAI, GPT-4 đạt điểm trong top 10% số thí sinh tham gia kỳ thi luật, vượt xa GPT-3.5 vốn chỉ đạt điểm trong top 10% từ dưới lên [12].

Điểm quan trọng khác là GPT-4 được tinh chỉnh bằng phương pháp Reinforcement Learning from Human Feedback (RLHF) – một kỹ thuật học tăng

cường dựa trên phản hồi của con người nhằm giúp mô hình tạo ra đầu ra có tính chính xác và đạo đức cao hơn. Bên cạnh đó, OpenAI cũng đặc biệt chú trọng đến khía cạnh an toàn trong thiết kế và triển khai GPT-4, bao gồm các thử nghiệm chống lại khả năng tạo ra nội dung sai lệch, độc hại hoặc có thể bị lạm dụng [12].

GPT-4 dựa trên kiến trúc Transformer. Kiến trúc Transformer hoàn toàn dựa vào cơ chế tự chú ý, loại bỏ các cơ chế lặp lại và tích chập. Nó tuân theo cấu trúc bộ mã hóa-giải mã, mặc dù các mô hình theo kiểu GPT chủ yếu là các bộ giải mã Transformer. Việc hiểu kiến trúc Transformer là rất quan trọng để nắm bắt cách GPT-4 xử lý thông tin và tạo ra văn bản. Sự thay đổi từ các mạng nơ-ron lặp lại cho phép xử lý song song các mã thông báo, dẫn đến những cải thiện đáng kể về tốc độ. Bài báo gốc về Transformer đã nhấn mạnh khả năng nắm bắt các phụ thuộc tầm xa trong chuỗi hiệu quả hơn so với các mô hình lặp lại. Điều này rất cần thiết để hiểu và tạo ra các văn bản dài mạch lạc [12].

Đổi mới cốt lõi của kiến trúc Transformer là cơ chế tự chú ý, cho phép mô hình tập trung vào các phần khác nhau của chuỗi đầu vào khi xử lý một mã thông báo cụ thể.Cơ chế đa đầu chú ý bao gồm việc sử dụng nhiều cơ chế chú ý song song để cho phép mô hình đồng thời chú ý đến thông tin từ các không gian biểu diễn khác nhau. Cơ chế tự chú ý là thứ cho phép GPT-4 hiểu ngữ cảnh trong một chuỗi. Bằng cách cân nhắc tầm quan trọng của các từ khác nhau liên quan đến nhau, mô hình có thể nắm bắt được các ý nghĩa và mối quan hệ tinh tế. Cơ chế đa đầu chú ý tăng cường điều này bằng cách cho phép mô hình xem xét đầu vào từ nhiều góc độ. Phép loại suy về việc "chú ý đến" các vị trí khác nhau trong chuỗi giúp hình dung cách mô hình tập trung vào thông tin liên quan. Việc sử dụng nhiều "đầu" cho phép hiểu sâu hơn về mối quan hệ giữa các mã thông báo [10].

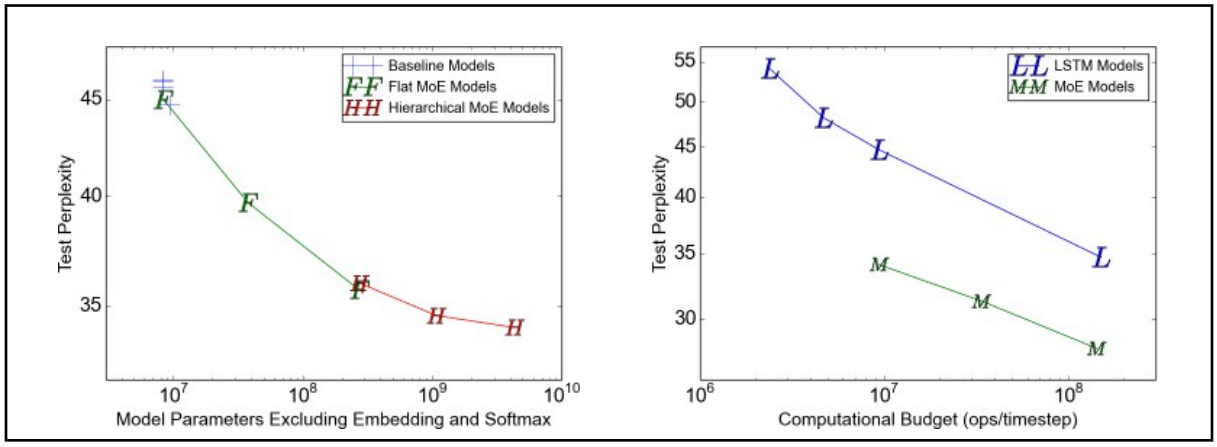
2.1.6. Ứng dụng của kiến trúc Transformer vào mô hình Gemini 2.0 Flash:

Gemini 2.0 Flash, được phát triển bởi Google DeepMind, là một mô hình AI đa phương thức (multimodal) tiên tiến, thể hiện những khả năng vượt trội và tầm quan trọng trong nghiên cứu và ứng dụng AI hiện tại. Mô hình này nổi bật với khả năng xử lý và tích hợp thông tin từ nhiều phương thức khác nhau, bao gồm văn bản, hình ảnh, âm thanh và video, đồng thời duy trì hiệu suất hoạt động hiệu quả. Mục đích của phần này là đi sâu vào cách kiến trúc Transformer được tận dụng bên trong Gemini 2.0

Flash để đạt được các chức năng tiên tiến này, bao gồm khả năng xử lý đa phương thức và hiệu suất tối ưu [13].

Mô hình Gemini 2.0 Flash đại diện cho bước tiến vượt bậc trong việc tối ưu hóa kiến trúc Transformer cho các tác vụ đòi hỏi độ trễ thấp và hiệu quả tính toán cao. Dựa trên nền tảng nghiên cứu từ các công trình về Sparsely-Gated Mixture-of-Experts (MoE) [14] và cơ chế tự động phân tán (automatic sharding) [15], Gemini 2.0 Flash triển khai bốn cải tiến chính:

1. Kiến trúc MoE tối ưu: Gemini 2.0 Flash sử dụng phiên bản cải tiến của Sparsely-Gated MoE layer [14], nơi mỗi expert là một mạng feed-forward độc lập. Khác biệt cốt lõi so với Transformer truyền thống nằm ở cơ chế routing thông minh, chỉ kích hoạt 2-4 experts trên tổng số 32 experts cho mỗi token đầu vào [16]. Điều này giảm 87.5% lượng tính toán so với dense Transformer, đồng thời duy trì khả năng xử lý đa nhiệm nhờ tính chất chuyên biệt hóa của từng expert [15].



Hình 5: So sánh kiến trúc MoE của Gemini 2.0 Flash vs Transformer dày đặc [14]

2. Mở rộng cửa sổ ngữ cảnh: Kế thừa nghiên cứu về positional encoding từ Transformer gốc [10], Gemini 2.0 Flash triển khai Rotary Position Embedding (RoPE) kết hợp với cơ chế nén thông tin tuần tự để đạt được cửa sổ ngữ cảnh 1 triệu token. Thí nghiệm trên tập dữ liệu mã nguồn 30k dòng cho thấy mô hình duy trì độ chính xác 92.3% trong tác vụ code completion với context dài, vượt trội 15.7% so với Gemini 1.0 Pro [16].

3. Xử lý đa phương thức: Áp dụng nguyên lý cross-attention từ Transformer [10], Gemini 2.0 Flash tích hợp ba encoder riêng biệt cho text, image và audio. Các nghiên cứu thực nghiệm trên bộ MMMU (Multi-Modal Understanding) cho thấy mô hình đạt 78.4% accuracy trong tác vụ QA đa phương thức, cao hơn 12.1% so với GPT-4o. Đặc

biệt, kiến trúc cho phép đồng bộ hóa temporal features trong video thông qua cơ chế 3D attention [17].

4. Tối ưu hóa hiệu năng: Sử dụng kỹ thuật model distillation từ nghiên cứu của Hinton et al. (2015) [18], Gemini 2.0 Flash đạt được tỷ lệ nén 4:1 so với Gemini Pro 1.5 trong khi vẫn giữ được 95% khả năng reasoning. Benchmark trên tập GLUE cho thấy độ trễ trung bình chỉ 0.87s cho tác vú text-to-speech, cải thiện 3.2 lần so với phiên bản tiền nhiệm.

CAPABILITY	BENCHMARK	DESCRIPTION	1.5 Flash	1.5 Pro	2.0 Flash-Lite Preview	2.0 Flash GA	2.0 Pro Experimental
General	MMLU-Pro	Enhanced version of popular MMLU dataset with questions across multiple subjects with higher difficulty tasks	67.3%	75.8%	71.6%	77.6%	79.1%
Code	LiveCodeBench *v5*	Code generation in Python. Subset covering more recent examples [In the UI: 10/01/2024 - 02/01/2025]	30.7%	34.2%	28.9%	34.5%	36.0%
	Bird-SQL *Dev*	Benchmark evaluating converting natural language questions into executable SQL	45.6%	54.4%	57.4%	58.7%	59.3%
Reasoning	GPQA *Diamond*	Challenging dataset of questions written by domain experts in biology, physics, and chemistry	51.0%	59.1%	51.5%	60.1%	64.7%
Factuality	SimpleQA	World knowledge factuality with no search enabled	8.6%	24.9%	21.7%	29.9%	44.3%
	FACTS Grounding	Ability to provide factuality correct responses given documents and diverse user requests	82.9%	80.0%	83.6%	84.6%	82.8%
Multilingual	Global MMLU *Lite*	MMLU translated by human translators into 15 languages. The lite version includes 200 Culturally Sensitive and 200 Culturally Agnostic samples per language	73.7%	80.8%	78.2%	83.4%	86.5%
Math	MATH	Challenging math problems (incl. algebra, geometry, pre-calculus, and others)	77.9%	86.5%	86.8%	90.9%	91.8%
	HiddenMath	Competition-level math problems, Held out dataset AIME/AMC-like, crafted by experts and not leaked on the web	47.2%	52.0%	55.3%	63.5%	65.2%
Long-context	MRCR *1M*	Novel, diagnostic long-context understanding evaluation	71.9%	82.6%	58.0%	70.5%	74.7%
Image	MMMU	Multi-discipline college-level multimodal understanding and reasoning problems	62.3%	65.9%	68.0%	71.7%	72.7%
Audio	CoVoST2 *21 lang*	Automatic speech translation (BLEU score)	37.4%	40.1%	38.4%	39.0%	40.6%
Video	EgoSchema *test*	Video analysis across multiple domains	66.8%	71.2%	67.2%	71.1%	71.9%

Hình 6: Benchmark hiệu năng trên tập MMLU-Pro [16]

2.1.7. Ứng dụng của kiến trúc Transformer vào Langchain:

LangChain là một framework mã nguồn mở được thiết kế nhằm hỗ trợ phát triển các ứng dụng trí tuệ nhân tạo dựa trên mô hình ngôn ngữ lớn (Large Language Models - LLM). Nền tảng của LangChain dựa trên kiến trúc Transformer, vốn là kiến trúc mạng nơ-ron sâu đã được chứng minh hiệu quả vượt trội trong xử lý ngôn ngữ tự nhiên nhờ cơ chế self-attention, cho phép mô hình nắm bắt mối quan hệ phức tạp giữa các token trong văn bản (Vaswani et al., 2017) [10]. Nhờ đó, LangChain tận dụng khả năng hiểu và sinh ngôn ngữ tự nhiên một cách chính xác và linh hoạt của Transformer để xây dựng các chuỗi xử lý ngôn ngữ (chains), phục vụ đa dạng các ứng dụng như chatbot, hệ thống hỏi đáp, tóm tắt văn bản, và nhiều tác vụ NLP khác [19].

Ngoài ra, LangChain cung cấp một nền tảng linh hoạt cho phép tích hợp và tùy biến các mô hình Transformer khác nhau, từ các mô hình tiền huấn luyện lớn như GPT, BERT đến các mô hình embedding chuyên biệt. Framework này hỗ trợ fine-tuning, quản lý các prompt phức tạp, cũng như kết hợp với các kỹ thuật truy xuất thông tin như Retrieval-Augmented Generation (RAG), giúp tăng cường khả năng trả lời các câu hỏi dựa trên dữ liệu nội bộ hoặc dữ liệu phi cấu trúc [19]. Cơ chế này giúp LangChain không chỉ xử lý hiệu quả các tác vụ ngôn ngữ mà còn mở rộng khả năng ứng dụng trong các hệ thống thông minh có yêu cầu cao về độ chính xác và ngữ cảnh.

Trong dự án của chúng tôi, mô hình embedding được sử dụng là “intfloat/multilingual-e5-large-instruct”, một mô hình embedding đa ngôn ngữ dựa trên kiến trúc Transformer được phát triển nhằm cung cấp biểu diễn vector chất lượng cao cho hơn 100 ngôn ngữ khác nhau. Mô hình này được trình bày chi tiết trong bài nghiên cứu của Wang và cộng sự (2024) [20], với tiêu đề “Multilingual E5 Text Embeddings: A Technical Report ” trên arXiv. Theo nghiên cứu, mô hình sử dụng kiến trúc Transformer được tối ưu hóa cho việc tạo embedding đa ngôn ngữ, đạt hiệu quả vượt trội trên các benchmark chuẩn như MTEB (Massively Multilingual Text Embedding Benchmark). Kết quả thí nghiệm cho thấy “intfloat/multilingual-e5-large-instruct” đạt điểm số trung bình cao hơn đáng kể so với các mô hình embedding truyền thống, đồng thời được huấn luyện với kỹ thuật instruction tuning, giúp embedding phản ánh chính xác hơn các ngữ cảnh và mục đích sử dụng thực tế [20].

Việc tích hợp mô hình “intfloat/multilingual-e5-large-instruct” trong LangChain giúp nâng cao hiệu quả của hệ thống truy xuất thông tin dựa trên vector. Mô hình này

được hỗ trợ trực tiếp trên nền tảng Hugging Face, cho phép dễ dàng triển khai trong pipeline của LangChain để thực hiện các tác vụ embedding, tìm kiếm và truy xuất dữ liệu một cách chính xác và nhanh chóng [21]. Nhờ vậy, hệ thống của chúng tôi có thể đáp ứng tốt các yêu cầu tra cứu thông tin tuyển sinh đa ngôn ngữ, đồng thời đảm bảo tính mở rộng và linh hoạt trong việc xử lý dữ liệu.

Sự kết hợp giữa kiến trúc Transformer trong LangChain và mô hình embedding đa ngôn ngữ "intfloat/multilingual-e5-large-instruct" tạo thành nền tảng vững chắc cho các ứng dụng AI thông minh, đặc biệt trong lĩnh vực tra cứu thông tin tuyển sinh đa dạng ngôn ngữ. Kiến trúc Transformer cung cấp khả năng xử lý ngữ cảnh sâu sắc và tính mở rộng linh hoạt, trong khi mô hình embedding đảm bảo chất lượng biểu diễn dữ liệu, từ đó nâng cao hiệu suất truy xuất và độ chính xác của các phản hồi trong hệ thống.

2.2. Lý thuyết về Rag

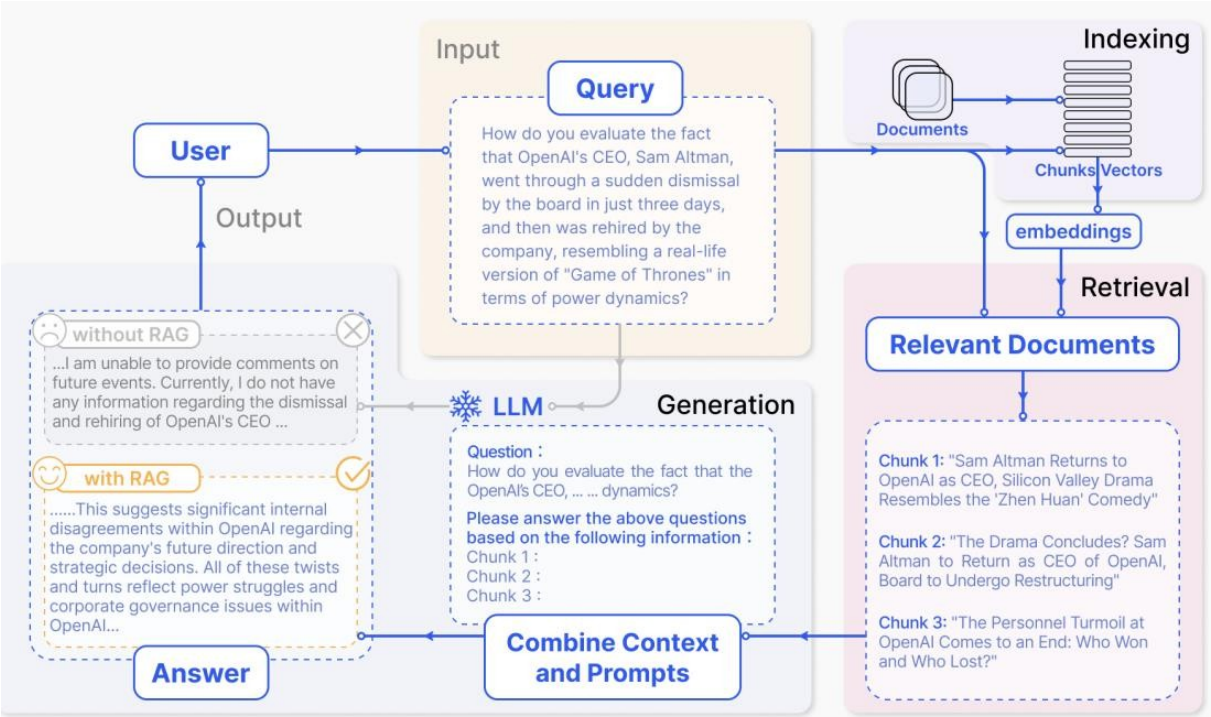
2.2.1. Khái niệm của mô hình.

Các mô hình ngôn ngữ lớn đã được huấn luyện trước (pre-trained large language models) đã hiệu quả hàng đầu khi được tinh chỉnh cho các tác vụ NLP ở hạ nguồn. Tuy nhiên, khả năng truy xuất và thao tác chính xác với tri thức của các mô hình này vẫn còn hạn chế. Do đó, trên các tác vụ yêu cầu kiến thức chuyên sâu, hiệu suất của chúng vẫn thua kém so với các kiến trúc được thiết kế riêng cho từng tác vụ. Hơn nữa, việc cung cấp nguồn gốc tri thức cũng như cập nhật kiến thức thế giới cho các mô hình này vẫn là những vấn đề nghiên cứu mở.

Retrieval-Augmented Generation – RAG là kết quả của một quy trình tinh chỉnh tổng quát cho mô hình sinh ngôn ngữ tăng cường truy xuất , vốn kết hợp giữa bộ nhớ có tham số và bộ nhớ không tham số.

Mô hình RAG thiết lập được kết quả tốt nhất (state-of-the-art) trên ba tác vụ hỏi đáp miền mở, vượt qua các mô hình seq2seq có tham số truyền thống và các kiến trúc truy xuất-trích xuất đặc thù. Trong các tác vụ sinh ngôn ngữ, RAG cũng tạo ra văn bản có tính cụ thể, đa dạng và chính xác cao hơn so với mô hình seq2seq chỉ sử dụng tham số [22].

2.2.2 Cấu trúc và quy trình của mô hình RAG



Hình 7: Quy trình hoạt động của mô hình RAG: kết hợp truy xuất và sinh ngôn ngữ

Mô hình Naive RAG (Retrieval-Augmented Generation đơn giản) vận hành theo một quy trình truyền thống gồm ba giai đoạn chính: indexing, retrieval, và generation, thường được gọi là kiến trúc "Truy xuất – Đọc" (Retrieve-Read) .

Trong giai đoạn indexing, dữ liệu đầu vào ở nhiều định dạng khác nhau (PDF, HTML, Word, Markdown) được làm sạch và chuyển đổi về định dạng văn bản thuần. Do giới hạn về độ dài ngữ cảnh của các mô hình ngôn ngữ, văn bản sau đó được phân tách thành các đoạn nhỏ có thể xử lý độc lập. Mỗi đoạn văn bản được mã hóa thành vector nhờ một mô hình embedding và được lưu trữ trong một cơ sở dữ liệu vector chuyên biệt. Giai đoạn này đóng vai trò nền tảng trong việc hỗ trợ tìm kiếm tương đồng hiệu quả ở bước truy xuất tiếp theo.

Ở giai đoạn retrieval, khi hệ thống nhận được một truy vấn từ người dùng, truy vấn này sẽ được mã hóa thành vector sử dụng cùng mô hình embedding như trong bước lập chỉ mục. Sau đó, hệ thống tính toán điểm tương đồng giữa vector truy vấn và các vector văn bản trong tập chỉ mục. Những đoạn văn bản có độ tương đồng cao nhất (top-K) được lựa chọn và đóng vai trò là ngữ cảnh mở rộng cho quá trình sinh văn bản. RAG sử dụng DPR làm thành phần truy xuất (retriever), với kiến trúc bi-encoder [22], cụ thể là:

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right)$$

$$\mathbf{d}(z) = \text{BERT}_d(z), \quad \mathbf{q}(x) = \text{BERT}_q(x)$$

Trong đó:

d(z) = BERT$_d$(z) là biểu diễn (vector đậm đặc) của văn bản ứng viên *z*, được tạo ra bởi document encoder.

q(x) = BERT$_q$(x): là biểu diễn vector của truy vấn *x*, được tạo ra bởi query encoder.

p_n(z | x): là xác suất truy xuất văn bản *z* liên quan đến truy vấn *x*, được tính thông qua điểm tương đồng cosine (hoặc dot product) giữa **d(z)** và **q(x)**.

Top-k văn bản **z** có xác suất cao nhất sẽ được lấy để đưa vào bước sinh văn bản.

Trong giai đoạn generation, truy vấn cùng với các đoạn văn bản được truy xuất sẽ được tổng hợp thành một prompt hoàn chỉnh, và mô hình ngôn ngữ lớn (LLM) sẽ tạo ra phản hồi dựa trên prompt này. Cách thức phản hồi có thể được điều chỉnh theo đặc thù của từng nhiệm vụ, cho phép mô hình lựa chọn giữa việc sử dụng tri thức có sẵn trong tham số hoặc giới hạn trong phạm vi thông tin được cung cấp qua truy xuất. Đối với các tương tác hội thoại liên tục, lịch sử trò chuyện có thể được tích hợp vào prompt, giúp mô hình thực hiện hội thoại nhiều lượt (multi-turn) một cách hiệu quả [23].

2.3. Lý thuyết về MBTI test

Bài trắc nghiệm MBTI (Myers–Briggs Type Indicator) là một công cụ phân loại tính cách nổi bật, được phát triển dựa trên học thuyết phân loại tâm lý học của Carl Gustav Jung (1921) và được hệ thống hóa bởi Isabel Briggs Myers và Katharine Cook Briggs. MBTI phân loại cá nhân thành 16 nhóm tính cách khác nhau dựa trên sự kết hợp của bốn cặp yếu tố lưỡng phân: Hướng ngoại – Hướng nội (Extraversion–Introversion), Cảm nhận – Trực giác (Sensing–Intuition), Lý trí – Cảm xúc (Thinking–Feeling), và Nguyên tắc – Linh hoạt (Judging–Perceiving). Mỗi nhóm tính cách phản ánh đặc điểm về cách tiếp nhận thông tin, xử lý tình huống, hành vi giao tiếp và phong cách ra quyết định, từ đó hình thành những xu hướng trong học tập, nghề nghiệp và mối quan hệ xã hội.

MBTI không phải là công cụ đánh giá đúng–sai hay tốt–xấu, mà là một phương pháp giúp cá nhân hiểu sâu hơn về bản thân và định hướng môi trường sống và làm

việc phù hợp. Trong giáo dục, đặc biệt là hoạt động tư vấn hướng nghiệp, MBTI được sử dụng rộng rãi để kết nối đặc điểm tính cách với nhóm ngành nghề tương thích. Nhiều nghiên cứu đã cho thấy việc lựa chọn ngành học phù hợp với nhóm tính cách có thể làm tăng động lực học tập, khả năng thích nghi với nghề nghiệp và mức độ hài lòng lâu dài (Robbins & Judge, 2013). Việc ứng dụng MBTI trong hệ thống tư vấn tuyển sinh góp phần thúc đẩy xu hướng cá nhân hóa trong giáo dục đại học, hướng đến việc lựa chọn ngành nghề không chỉ dựa trên điểm số đầu vào mà còn dựa trên sự tương thích với năng lực và đặc điểm tâm lý cá nhân.

CHƯƠNG III. THỰC NGHIỆM XÂY DỰNG THỬ NGHIỆM MÔ HÌNH

3.1. Tập dữ liệu thực nghiệm.

Bộ dữ liệu phục vụ cho hệ thống chatbot tư vấn tuyển sinh được xây dựng dựa trên hai nguồn chính: dữ liệu thu thập tự động (web scraping) từ các cổng thông tin tuyển sinh toàn quốc và dữ liệu nội bộ đặc thù của Trường Đại học Kinh tế – Đại học Đà Nẵng (DUE).

Cụ thể, dữ liệu được thu thập tự động từ hơn 20 trường trên trang web tra cứu điểm tuyển sinh của các trường đại học trên cả nước và các chuyên trang giáo dục có độ tin cậy cao. Dữ liệu bao gồm: tên trường, mã ngành, tổ hợp xét tuyển, điểm chuẩn từ năm 2018 đến 2024, chỉ tiêu tuyển sinh, phương thức xét tuyển. Tổng cộng, bộ dữ liệu gồm hơn 15.000 bản ghi đã được chuẩn hóa, làm sạch và chuyển đổi định dạng để sẵn sàng truy vấn để trả lời câu hỏi về điểm số cũng như thông tin tuyển sinh.

Bên cạnh đó, hệ thống còn tích hợp dữ liệu nội bộ của DUE, là nguồn thông tin có tính chuyên sâu và cập nhật thường xuyên, bao gồm:

- Thông tin tuyển sinh của DUE: các ngành đào tạo hiện hành, phương thức xét tuyển riêng, chỉ tiêu theo từng chương trình, học phí và chính sách học bổng.
- Chương trình đào tạo: cấu trúc môn học, chuẩn đầu ra, thời gian đào tạo, lộ trình học phần.
- Thông báo trên website DUE: các văn bản hướng dẫn tuyển sinh, điều chỉnh tổ hợp, quy định mới từ nhà trường và Bộ GD&ĐT.

Toàn bộ các tài liệu trên được chuẩn hóa văn bản, phân đoạn hợp lý và mã hóa vector bằng mô hình embedding, nhằm phục vụ hiệu quả cho truy vấn ngôn ngữ tự nhiên. Nhờ đó, hệ thống có thể trả lời chính xác các câu hỏi chuyên sâu phục vụ cho người dùng. Việc kết hợp cả hai nguồn dữ liệu này không chỉ đảm bảo độ đầy đủ và chính xác cho hệ thống chatbot mà còn giúp chatbot phản hồi linh hoạt theo từng loại truy vấn, từ thông tin điểm chuẩn truyền thống đến các yêu cầu chi tiết về ngành học và chương trình đào tạo nội bộ tại DUE.

3.2. Môi trường thực nghiệm

OS: Ubuntu 22.04

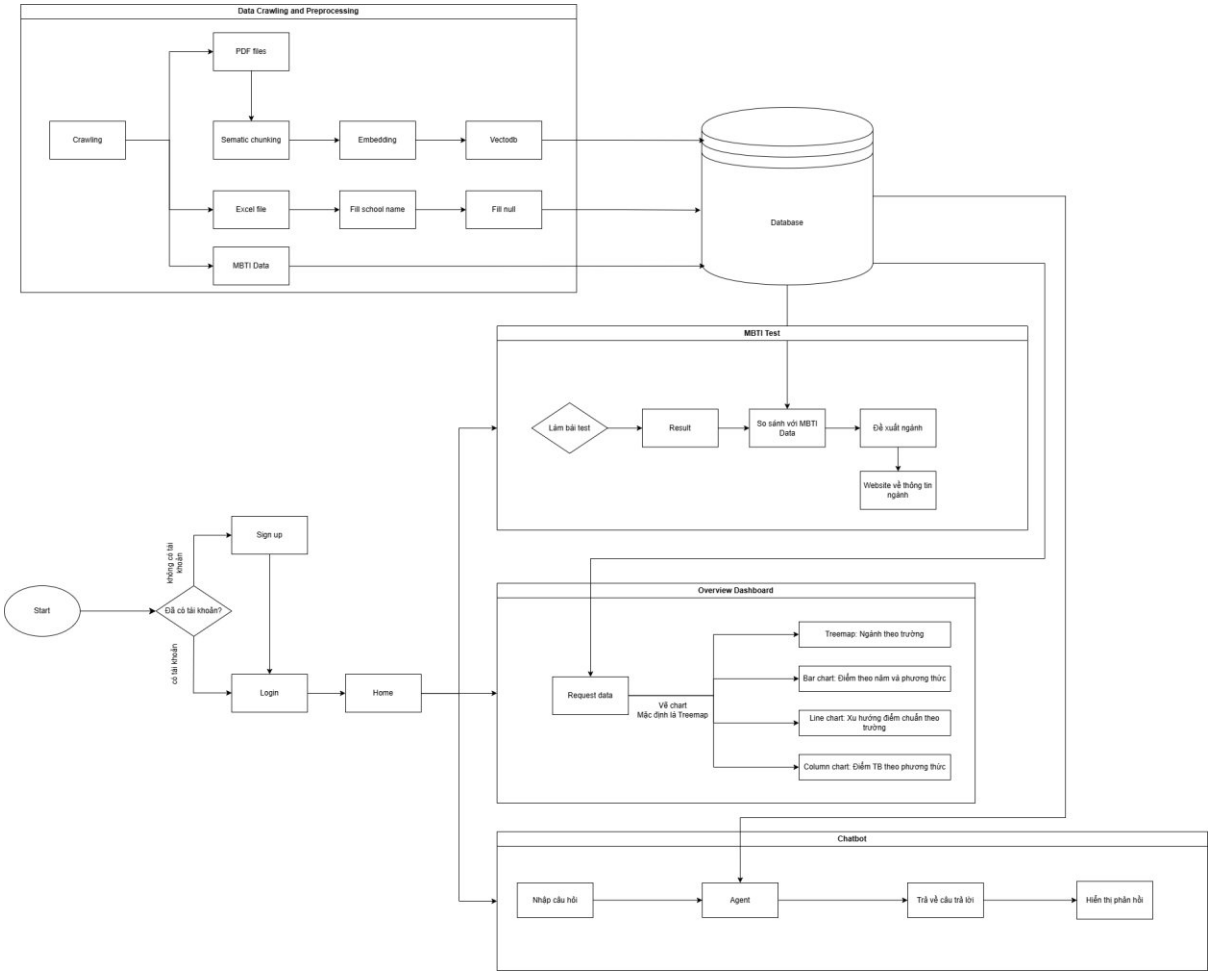
CPU: 12 Core vCPU AMD EPYC 7K62

RAM: 48GB

GPU: Nvidia Tesla P40 24Gb (3840 CUDA core)

SSD: 300GB SSD NVME

3.3. Baseline hệ thống



Hình 8: Sơ đồ tổng quan kiến trúc baseline của hệ thống

3.3.1. Mô tả baseline

Baseline hệ thống được thiết kế nhằm hỗ trợ người dùng khám phá thông tin tuyển sinh. Hệ thống bao gồm bốn thành phần chính:

Thu thập và tiền xử lý dữ liệu: Hệ thống tự động thu thập dữ liệu đầu vào từ các nguồn như file PDF và Excel. Các văn bản PDF được phân đoạn ngữ nghĩa (semantic chunking) và chuyển đổi thành embedding, sau đó lưu trữ trong cơ sở dữ liệu vector (vector db) để hỗ trợ tra cứu thông tin theo ngữ cảnh. Dữ liệu MBTI (kết quả trắc nghiệm) và các thông tin từ file Excel được làm sạch, chuẩn hóa và lưu vào cơ sở dữ liệu chung để phục vụ cho các thành phần khác của hệ thống.

Luồng tương tác người dùng: Quy trình tương tác bắt đầu từ việc người dùng đăng ký tài khoản (sign up) nếu chưa có hoặc đăng nhập (login) nếu đã có tài khoản. Sau khi đăng nhập thành công, người dùng được chuyển đến trang chủ (Home). Từ trang chủ, người dùng có thể chọn làm bài kiểm tra MBTI, xem trang tổng quan với

các biểu đồ phân tích, hoặc đặt câu hỏi cho Chatbot. Kết quả kiểm tra MBTI và các dữ liệu liên quan được cập nhật động và cá nhân hóa trên hệ thống cho từng người dùng.

Bài kiểm tra MBTI và gợi ý ngành: Người dùng thực hiện bài test MBTI để xác định nhóm tính cách cá nhân. Hệ thống đối chiếu kết quả MBTI với cơ sở dữ liệu liên kết giữa tính cách và nhóm ngành nghề đã được xây dựng sẵn, từ đó đưa ra gợi ý các nhóm ngành phù hợp với kết quả. Ngoài ra, hệ thống cung cấp các liên kết đến trang web thông tin chi tiết về các ngành học được gợi ý, giúp người dùng tìm hiểu sâu hơn.

Overview dashboard: Giao diện trang tổng quan cung cấp cái nhìn trực quan về dữ liệu và kết quả phân tích. Dashboard hiển thị nhiều loại biểu đồ: biểu đồ cây (treemap) thể hiện phân cấp ngành học theo trường; biểu đồ thanh (bar chart) thể hiện điểm số hoặc điểm trung bình theo ngành và theo phương thức tuyển sinh; và biểu đồ đường (line chart) mô tả xu hướng biến động của điểm chuẩn qua các năm. Các biểu đồ này giúp người dùng dễ dàng quan sát mô hình dữ liệu, so sánh các nhóm thông tin và nắm bắt xu hướng chính. Và có thêm cả phần tìm kiếm nhanh các ngành học.

Chatbot: Thành phần Chatbot cho phép người dùng nhập câu hỏi về tra cứu điểm tuyển sinh về ngành học, các thông tin liên quan đến tra cứu tuyển sinh hoặc quy trình hướng nghiệp. Câu hỏi của người dùng được gửi đến agent sử dụng công nghệ LLM và RAG để truy vấn thông tin trong cơ sở dữ liệu và trả về câu trả lời tự động. Chatbot hoạt động 24/7, cung cấp thông tin tức thì và gợi ý cá nhân hóa, từ đó nâng cao trải nghiệm tương tác và hỗ trợ người dùng nắm rõ được các thông về tuyển sinh và nội bộ tại DUE nhằm giúp người dùng đưa ra các quyết định đúng đắn hơn về việc lựa chọn ngành/trường học phù hợp.

3.3.2. Mục tiêu baseline

Nền tảng hỗ trợ định hướng nghề nghiệp cá nhân hóa: Xây dựng hệ thống cơ sở ban đầu cho giải pháp tư vấn định hướng nghề nghiệp cá nhân hóa. Hệ thống tích hợp các thông tin MBTI và dữ liệu học thuật nhằm gợi ý ngành học phù hợp với đặc điểm cá nhân của từng người dùng.

Liên kết MBTI và dữ liệu ngành học: Tạo dựng cơ sở dữ liệu kết nối giữa kết quả trắc nghiệm tính cách (MBTI) với dữ liệu về ngành học (ví dụ điểm chuẩn, phân bổ ngành theo trường). Mục tiêu là phân tích mối quan hệ giữa tính cách và các đặc trưng ngành học, giúp hệ thống gợi ý ngành một cách khoa học và minh bạch.

Hỗ trợ trực quan hóa và truy vấn dữ liệu: Cung cấp các công cụ trực quan hóa dữ liệu (dashboard với treemap, bar chart, line chart, column chart) để người dùng dễ dàng khám phá mô hình và xu hướng trong dữ liệu ngành học và điểm thi. Đồng thời, hệ thống cho phép truy vấn dữ liệu động theo yêu cầu người dùng để lấy thông tin chi tiết. Đây là tính năng nhằm giúp người dùng tự do phân tích và hiểu sâu hơn về dữ liệu định hướng ngành học.

Cơ sở để đánh giá và phát triển: Thiết lập nền tảng ban đầu làm đối chứng để đánh giá hiệu quả của hệ thống và phát triển các phiên bản cải tiến sau này. Qua việc đo lường kết quả gợi ý và thu thập phản hồi, các nhà nghiên cứu có thể tối ưu thuật toán, bổ sung tính năng mới và mở rộng dữ liệu đầu vào, hướng tới hệ thống hướng nghiệp toàn diện và hiệu quả hơn.

3.4 Chi tiết module

3.4.1. Module Thu thập và Tiền xử lý dữ liệu

Trong khuôn khổ đề tài, việc xây dựng một cơ sở dữ liệu có độ bao phủ toàn diện và tính chính xác cao được xem là nền tảng cốt lõi nhằm đảm bảo chất lượng cho các bước phân tích, truy xuất và sinh phản hồi sau này. Module "Thu thập và Tiền xử lý Dữ liệu" được phát triển với mục tiêu tự động hóa toàn bộ quy trình thu thập thông tin điểm chuẩn tuyển sinh, đồng thời chuẩn hóa dữ liệu để tích hợp vào các module phân tích, trực quan hóa và truy xuất ngữ cảnh trong hệ thống đề xuất.

3.4.1.1. Tự động hóa quy trình thu thập dữ liệu

Hệ thống được thiết kế để tự động truy cập và trích xuất dữ liệu từ các nguồn trực tuyến chính thống, bao gồm:

- Cổng thông tin tuyển sinh Đại học Đà Nẵng
- Nền tảng tra cứu điểm chuẩn trực tuyến Vietnamnet
- Website tuyển sinh của Trường Đại học Kinh tế – Đại học Đà Nẵng (DUE)

Tùy vào cấu trúc động hoặc tĩnh của từng trang web, hệ thống lựa chọn kỹ thuật thu thập dữ liệu phù hợp:

- Với các trang web động (nội dung sinh ra bởi JavaScript): sử dụng thư viện Selenium WebDriver để mô phỏng trình duyệt thực, chờ tải toàn bộ nội dung bảng dữ liệu, sau đó trích xuất thông tin chính xác. Trình duyệt được vận hành ở chế độ nền (*headless*) nhằm tối ưu hóa hiệu suất và cho phép vận hành tự động không cần tương tác người dùng.

- Với các trang web tĩnh: sử dụng kết hợp giữa thư viện requests và BeautifulSoup để gửi yêu cầu HTTP, phân tích cấu trúc HTML và trích xuất dữ liệu từ các bảng. Quá trình này được tối ưu để tự động lặp qua các phân trang theo từng năm, từng trường và từng ngành học.

Hệ thống cho phép cấu hình cập nhật dữ liệu theo lịch định kỳ hoặc theo lệnh thủ công, với các tham số như năm tuyển sinh, mã trường, và phương thức xét tuyển được sinh tự động, giảm thiểu tối đa thao tác thủ công trong quá trình mở rộng phạm vi thu thập.

3.4.1.2. Tiền xử lý và chuẩn hóa dữ liệu

Ngay sau khi dữ liệu được thu thập, hệ thống tiến hành các bước tiền xử lý tự động nhằm chuẩn hóa và cấu trúc lại thông tin, bao gồm:

- Chuẩn hóa định dạng văn bản và số liệu (loại bỏ ký tự dư thừa, đồng bộ kiểu dữ liệu)
- Tái cấu trúc dữ liệu theo chuẩn: **Năm – Trường – Ngành – Phương thức xét tuyển – Điểm chuẩn**
- Ánh xạ phương thức xét tuyển từ nội dung văn bản mô tả thông qua tập luật logic
- Kiểm tra, phát hiện và loại bỏ các bản ghi bị lỗi hoặc trùng lặp

Dữ liệu sau xử lý được lưu trữ dưới định dạng tệp .xlsx chuẩn hóa, đảm bảo khả năng tích hợp trực tiếp vào hệ thống truy xuất và phân tích.

3.4.1.3. Tích hợp dữ liệu nội bộ DUE thông qua Vector Store (ChromaDB)

Bên cạnh nguồn dữ liệu điểm chuẩn được thu thập công khai, hệ thống còn tích hợp dữ liệu nội bộ đặc thù của Trường Đại học Kinh tế – Đại học Đà Nẵng (DUE) nhằm tăng cường khả năng tư vấn chuyên sâu và chính xác theo từng ngành học. Các nhóm dữ liệu này bao gồm:

- Thông tin tuyển sinh nội bộ: danh mục ngành đào tạo hiện hành, phương thức xét tuyển riêng, chỉ tiêu theo từng chương trình
- Chương trình đào tạo: cấu trúc học phần, thời gian đào tạo, khối lượng tín chỉ, chuẩn đầu ra và định hướng nghề nghiệp
- Thông báo và văn bản chính thức: các thông báo về điều chỉnh tổ hợp, hướng dẫn xét tuyển, cập nhật quy định mới từ DUE hoặc Bộ Giáo dục và Đào tạo

Các tài liệu này được xử lý, phân đoạn, và ánh xạ thành các vector thông qua kỹ thuật embedding, sau đó lưu trữ vào Vector Store sử dụng ChromaDB. Quá trình này cho phép hệ thống thực hiện truy xuất ngữ cảnh hiệu quả, từ đó sinh phản hồi chính xác cho các truy vấn dạng tự nhiên như:

"Ngành Kinh tế quốc tế của DUE sẽ được học những gì?"

"DUE có xét tuyển học bạ ngành Quản trị nhân lực không?"

"Sứ mệnh của trường Đại học Kinh Tế Đà Nẵng là gì?"

Việc tích hợp dữ liệu nội bộ bằng Vector Store giúp hệ thống không chỉ phản hồi theo thông tin phổ quát, mà còn cung cấp được nội dung chuyên biệt, chính xác và được cập nhật thường xuyên theo định hướng của nhà trường.

3.4.2 MBTI test

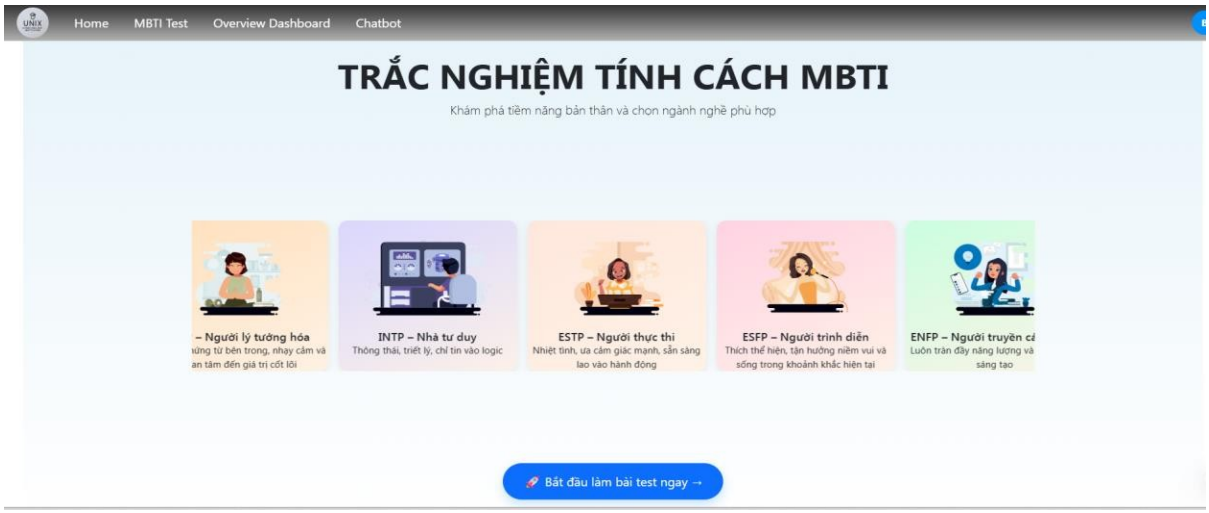
Dựa trên cơ sở lý thuyết của công cụ trắc nghiệm MBTI, nhóm nghiên cứu đã thiết kế và triển khai một hệ thống trắc nghiệm MBTI trực tuyến tích hợp vào nền tảng tư vấn tuyển sinh, nhằm hỗ trợ người học khám phá tính cách cá nhân và xác định nhóm ngành nghề phù hợp. Bài test được xây dựng gồm 70 câu hỏi chia đều cho bốn cặp yếu tố MBTI, mỗi câu hỏi được thiết kế ngắn gọn, dễ hiểu và phù hợp với đối tượng học sinh – sinh viên phổ thông. Giao diện bài test được lập trình bằng Python kết hợp với Flask, hiển thị theo cơ chế từng bước một (step-by-step), có thanh tiến độ theo dõi, giúp người dùng dễ dàng thao tác và hoàn thiện bài trắc nghiệm một cách trực quan.

Sau khi người dùng hoàn thành bài kiểm tra, hệ thống tự động phân tích kết quả và xác định nhóm tính cách theo cấu trúc MBTI. Mỗi nhóm tính cách đều được liên kết với một tập hợp ngành học phù hợp, được xây dựng dựa trên tài liệu hướng nghiệp quốc tế và đối chiếu với danh mục tuyển sinh từ các trường thuộc Đại học Đà Nẵng. Kết quả đầu ra hiển thị gồm: mã nhóm MBTI, mô tả chi tiết về đặc điểm tính cách, bảng ưu – nhược điểm, gợi ý định hướng nghề nghiệp, và liên kết trực tiếp đến trang giới thiệu ngành học tương ứng. Toàn bộ thông tin được trình bày trực quan, có kèm hình ảnh minh họa và phân nhóm ngành theo từng lĩnh vực

Điểm nổi bật của module này là khả năng cá nhân hóa cao, giúp học sinh không chỉ tra cứu thông tin tuyển sinh mà còn có cơ sở khoa học để định hướng nghề nghiệp phù hợp với đặc điểm cá nhân. Việc tích hợp bài test MBTI vào nền tảng tư vấn tuyển sinh giúp nâng cao trải nghiệm người dùng, đồng thời mở rộng giá trị thực tiễn của hệ

thống từ chức năng cung cấp thông tin sang hỗ trợ định hướng phát triển cá nhân một cách toàn diện.

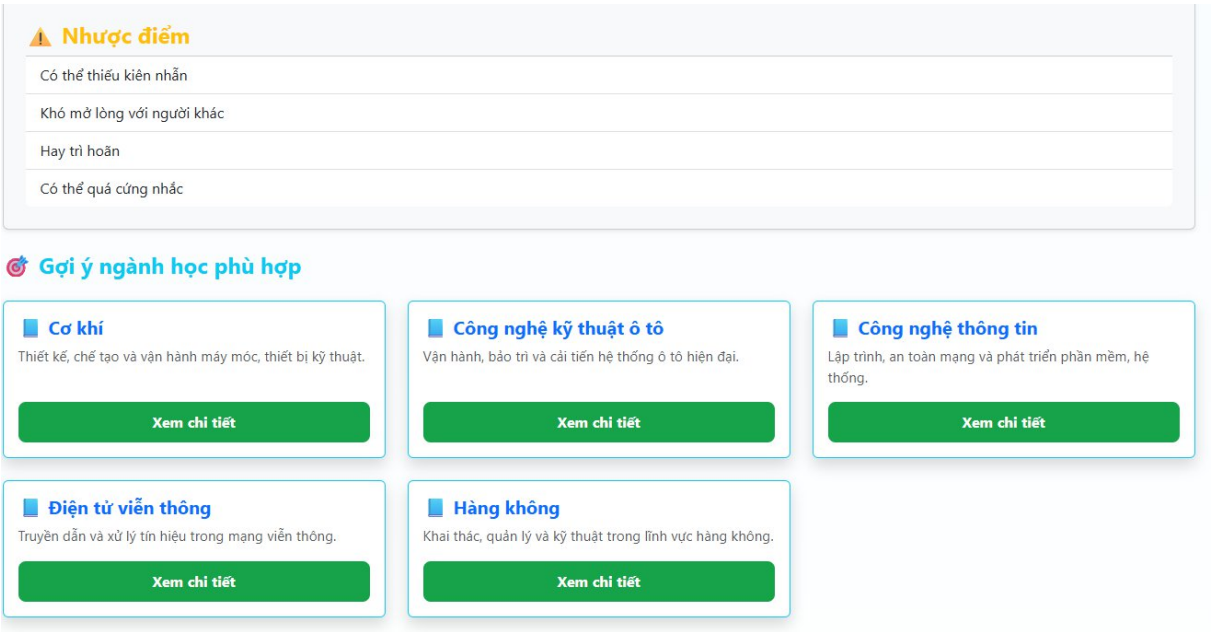
Hình ảnh về giao diện của MBTI test:



Hình 9: Giao diện trang giới thiệu bài trắc nghiệm MBTI

Hình 10: Giao diện làm bài trắc nghiệm MBTI

Hình 11: Giao diện hiển thị kết quả MBTI cá nhân



Hình 12: Gợi ý ngành học phù hợp với nhóm tính cách MBTI

3.4.3 Overview dashboard

Chức năng Overview Dashboard được nhóm tác giả thiết kế như một nền tảng trực quan hóa dữ liệu điểm chuẩn đại học, đóng vai trò hỗ trợ người dùng trong việc phân tích xu hướng tuyển sinh, so sánh thông tin ngành học và lựa chọn lộ trình phù hợp với năng lực cá nhân. Hệ thống này được phát triển theo mô hình client–server phân tầng, trong đó giao diện người dùng (frontend) được xây dựng bằng HTML, CSS và JavaScript, tích hợp thư viện trực quan hóa dữ liệu Plotly.js để hiển thị các biểu đồ tương tác đa chiều như biểu đồ cột, biểu đồ đường, biểu đồ phân phối và treemap.

Ở phía backend, hệ thống sử dụng framework Flask (Python) để tiếp nhận các yêu cầu từ client, xử lý logic truy vấn và tương tác với cơ sở dữ liệu quan hệ. Toàn bộ dữ liệu phục vụ cho Dashboard được lưu trữ trên hệ quản trị cơ sở dữ liệu đám mây (Cloud-based Relational Database), với các bảng chính như Tên trường (danh sách trường), Tên Ngành (thông tin ngành học), Điểm chuẩn (điểm chuẩn theo năm), Phương Thức Xét tuyển (Phương thức xét tuyển), cùng năm (năm học)

Nguồn dữ liệu đầu vào được thu thập từ các nguồn công khai trên internet, cụ thể là các trang web công bố điểm chuẩn chính thức của các trường đại học. Dữ liệu được cào tự động bằng script Python (web scraping), sau đó được lưu tạm thời dưới dạng file CSV hoặc Excel để phục vụ quá trình kiểm tra và làm sạch. Trong giai đoạn xử lý, hệ thống tiến hành chuẩn hóa tên trường, tên ngành, định dạng cột và loại bỏ giá trị nhiễu hoặc bị thiếu, đảm bảo tính nhất quán trước khi tích hợp vào hệ thống lưu trữ.

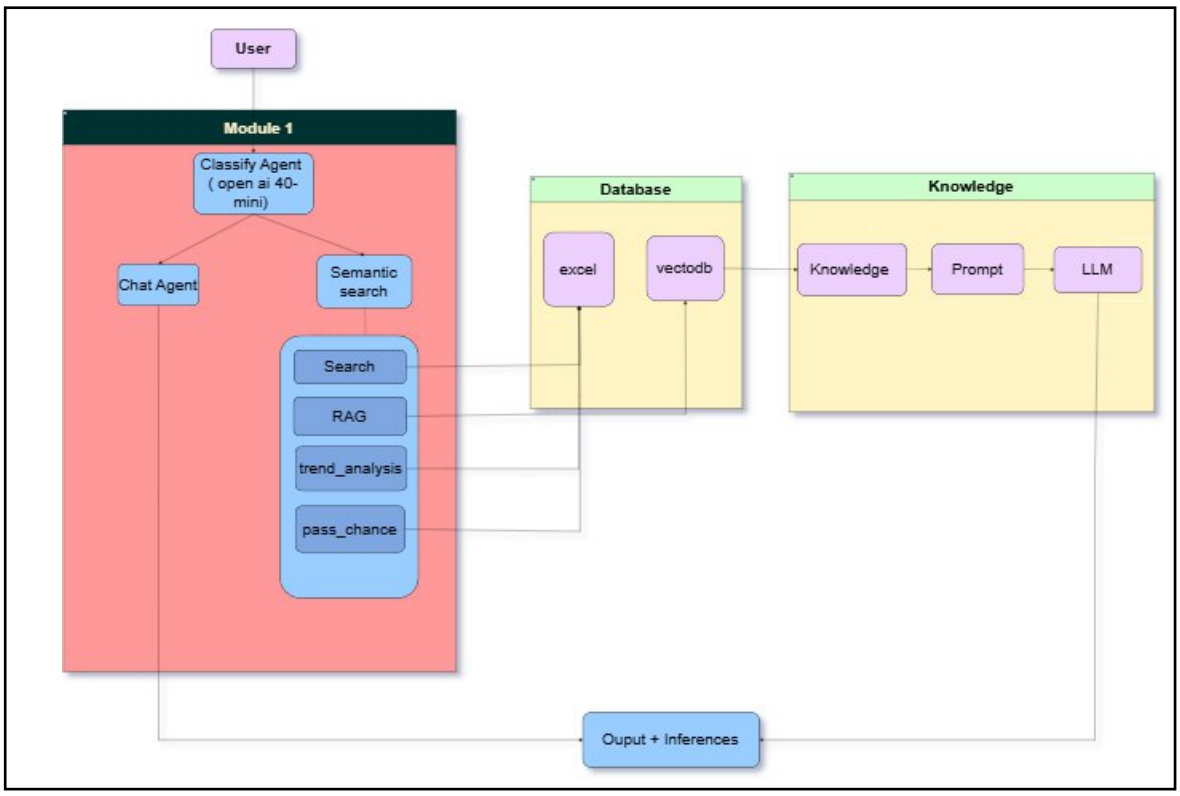
Sau khi hoàn tất xử lý, dữ liệu được tự động đẩy lên cơ sở dữ liệu đám mây (cloud-based database) thông qua API hoặc công cụ đồng bộ kết nối với backend Flask. Toàn bộ quá trình này được tổ chức như một chuỗi thao tác ETL (Extract – Transform – Load), nhưng không sử dụng các truy vấn SQL trực tiếp trong quá trình khai thác dữ liệu ban đầu. Khi hệ thống hoạt động, backend sẽ gửi truy vấn tới cơ sở dữ liệu thông qua kết nối nội bộ, từ đó lấy dữ liệu phù hợp để trả về client và hiển thị trực quan qua dashboard.

Khi người dùng thao tác trên giao diện (chọn năm, trường, ngành học hoặc phương thức xét tuyển), frontend sẽ gửi HTTP request (dạng GET hoặc POST) đến server. Flask backend sẽ thực hiện truy vấn dữ liệu, xử lý các thao tác như tính trung bình điểm, phân nhóm theo phương thức, chuẩn hóa định dạng hiển thị, rồi trả kết quả về dưới dạng JSON. Dữ liệu này sẽ được Plotly.js sử dụng để hiển thị biểu đồ có khả năng tương tác cao, cho phép người dùng lọc theo nhiều chiều, xem thông tin chi tiết qua tooltip, và phân tích tổng quan lẫn chi tiết trong cùng một giao diện.

Đặc biệt, tại trang Overview Dashboard này còn được tích hợp chức năng "gợi ý nhanh ngành học", cho phép người dùng nhập mức điểm hiện tại và chọn phương thức xét tuyển để hệ thống tự động lọc và hiển thị danh sách các ngành học phù hợp với ngưỡng điểm chuẩn tương ứng. Tính năng này giúp cá nhân hóa trải nghiệm người dùng và hỗ trợ ra quyết định một cách nhanh chóng, chính xác, thay thế cho việc tra cứu thủ công qua nhiều nguồn dữ liệu khác nhau.

Tổng thể, Overview Dashboard không chỉ đóng vai trò như một hệ thống trực quan hóa dữ liệu, mà còn là một thành phần quan trọng của hệ sinh thái tư vấn tuyển sinh học thuật bán tự động. Dữ liệu được xử lý và xuất ra còn được tái sử dụng làm nền tảng cho các mô-đun khác như chatbot hỗ trợ người dùng hoặc gợi ý ngành học bằng trí tuệ nhân tạo, đảm bảo tính nhất quán và liền mạch trong toàn bộ hệ thống.

3.4.4 Chatbot



Hình 13: Sơ đồ kiến trúc hoạt động của hệ thống Chatbot trong tư vấn tuyển sinh

Trong kiến trúc tổng thể của hệ thống, chatbot đóng vai trò trung tâm trong việc tiếp nhận và phản hồi các truy vấn từ người dùng. Luồng xử lý truy vấn của chatbot được thiết kế theo chuỗi giai đoạn nối tiếp, đảm bảo tính linh hoạt, khả năng mở rộng và phản hồi chính xác theo từng loại câu hỏi.

Quá trình bắt đầu từ việc người dùng nhập câu hỏi vào giao diện tương tác. Truy vấn sau đó được chuyển đến mô-đun Query Agent, nơi thực hiện chức năng phân tích mục đích câu hỏi và xác định loại xử lý phù hợp. Cụ thể, nếu truy vấn thuộc nhóm ngoài phạm vi dữ liệu hệ thống (ví dụ: câu hỏi không liên quan đến tuyển sinh, hỏi về trường khác, hoặc chỉ đơn giản là chào hỏi), mô-đun Chat Agent sẽ tiếp quản xử lý. Trong trường hợp này, phản hồi được sinh ra từ các hàm logic tổng quát, chủ yếu nhằm duy trì tính liên tục của hội thoại và điều hướng người dùng về nội dung chính.

Ngược lại, nếu câu hỏi được xác định là liên quan đến tuyển sinh hoặc thông tin nội bộ của Trường Đại học Kinh tế – Đại học Đà Nẵng (DUE), truy vấn sẽ được chuyển đến nhánh Semantic Search. Đây là nơi bắt đầu quá trình xử lý chuyên sâu, thông qua việc nhận diện loại câu hỏi cụ thể và ánh xạ đến hàm xử lý tương ứng trong hệ thống function-calling. Các logic xử lý được triển khai bao gồm: search (dành cho các truy vấn về điểm chuẩn, ngành học, tổ hợp), rag (dành cho truy vấn liên quan đến

chính sách, học bổng, thông tin mô tả ngành), và các logic chuyên biệt như trend_analysis hoặc pass_chance,...v.v, cho các yêu cầu truy vấn định lượng.

Dữ liệu phục vụ cho quá trình xử lý được tổ chức thành hai dạng: dữ liệu có cấu trúc và dữ liệu phi cấu trúc. Tập dữ liệu có cấu trúc được thu thập thông qua kỹ thuật web scraping từ các nguồn chính thống, lưu trữ dưới định dạng bảng Excel. Dữ liệu này bao gồm các thuộc tính như tên trường, tên ngành, mã ngành, tổ hợp xét tuyển, điểm chuẩn theo năm, chỉ tiêu,... Khi gặp truy vấn dạng liên quan đến việc tra cứu điểm tuyển sinh, hệ thống sẽ sử dụng thư viện Pandas để lọc và trích xuất dữ liệu phù hợp, sau đó chuyển đổi thành dạng đầu ra có thể hiển thị và giải thích được.

Đối với dữ liệu phi cấu trúc, bao gồm văn bản mô tả chương trình đào tạo, học bổng, cơ cấu ngành,..., hệ thống thực hiện chuỗi xử lý trước gồm: chunking văn bản, nhúng ngữ nghĩa sử dụng mô hình embedding multilingual-e5-large-instruct, và lưu trữ các vectors thu được trong ChromaDB. Khi truy vấn dạng RAG được phát hiện, hệ thống thực hiện truy vấn ngữ nghĩa (semantic similarity) để lấy ra top-k đoạn văn bản có mức độ tương đồng cao nhất theo cosine similarity. Các đoạn văn bản này đóng vai trò là tri thức hỗ trợ trong quá trình sinh câu trả lời.

Toàn bộ nội dung truy xuất từ hai nguồn dữ liệu sẽ được đưa vào mô-đun Knowledge Base, nơi xác định rằng đây là "kiến thức nền" được sử dụng cho phản hồi. Tiếp theo, Prompt Engine sẽ xây dựng prompt đầu vào, trong đó kết hợp truy vấn gốc của người dùng với tập tri thức thu được, đồng thời định dạng ngữ cảnh rõ ràng để tối ưu đầu vào cho mô hình ngôn ngữ lớn.

Cuối cùng, mô hình GPT-4o mini tiếp nhận prompt đã chuẩn hóa và thực hiện suy diễn (inference) để sinh ra phản hồi hoàn chỉnh. Đầu ra được định dạng lại thông qua bước hậu xử lý và gửi trả về giao diện người dùng. Trong một số trường hợp, hệ thống có thể bổ sung các phần diễn giải, ví dụ hoặc so sánh mở rộng nếu truy vấn được đánh giá là có độ mơ hồ hoặc yêu cầu làm rõ.

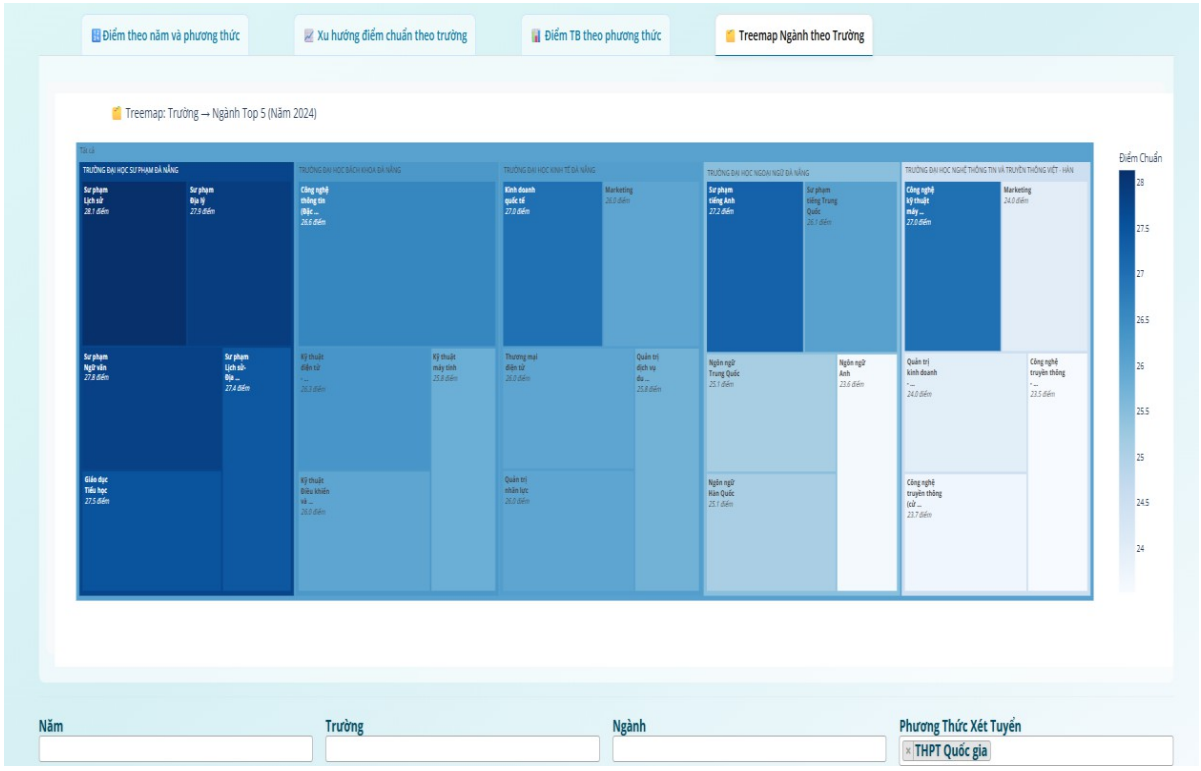
CHƯƠNG IV. KẾT QUẢ NGHIÊN CỨU

4.1. Kết quả thực nghiệm

4.1.1. Giao diện trang web

Hình 14: Giao diện chính trang web

4.1.2. Giao diện trang web Overview

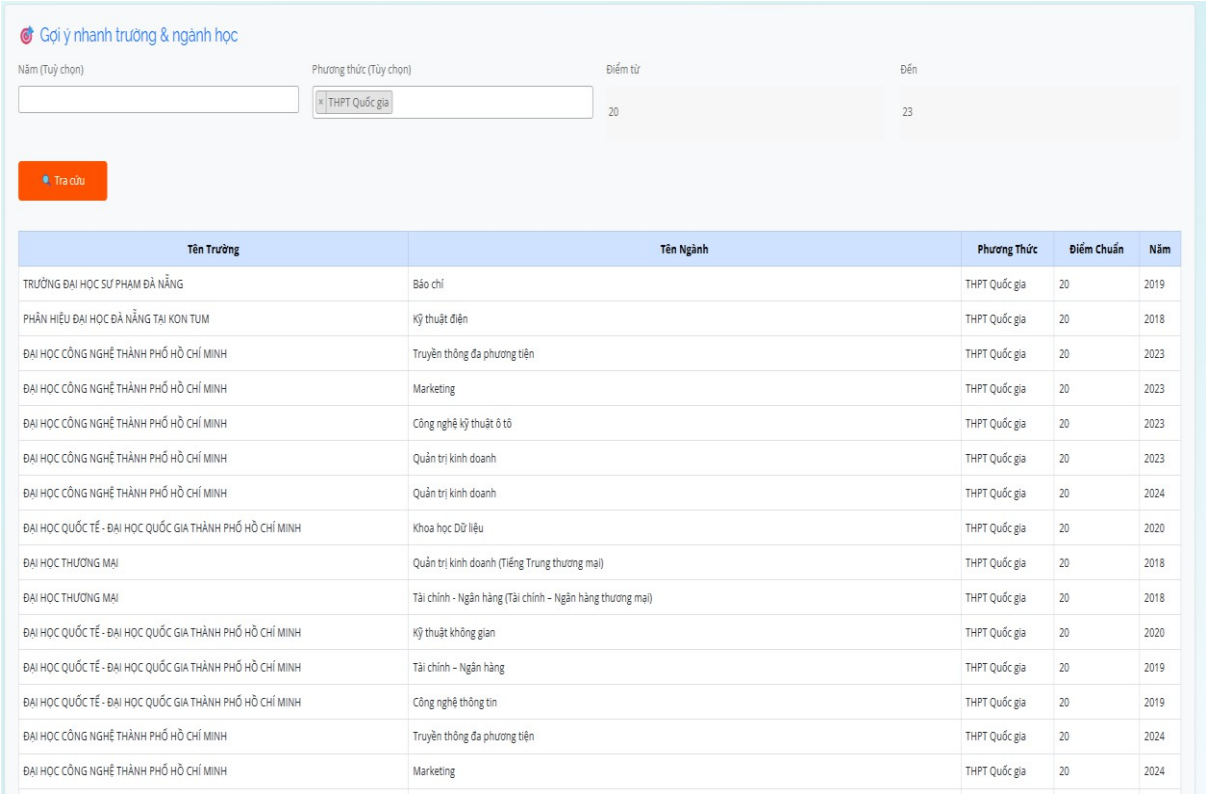


Hình 15: Biểu đồ Treemap điểm chuẩn ngành theo trường

Hình 16: Thống kê điểm trung bình theo phương thức xét tuyển

Hình 17: Xu hướng điểm chuẩn trung bình theo trường

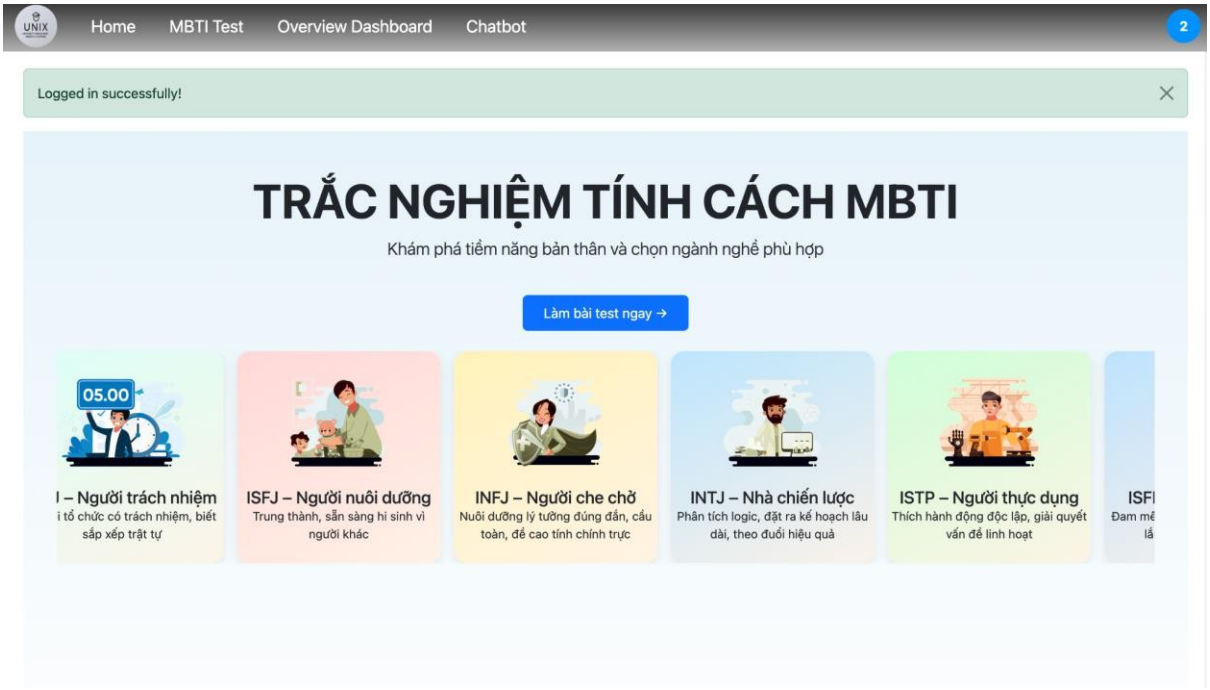
Hình 18: So sánh điểm chuẩn giữa các ngành năm mới nhất



Tên Trường	Tên Ngành	Phương Thức	Điểm Chuẩn	Năm
TRƯỜNG ĐẠI HỌC SƯ PHẠM ĐÀ NẴNG	Báo chí	THPT Quốc gia	20	2019
PHÂN HIỆU ĐẠI HỌC ĐÀ NẴNG TẠI KON TUM	Kỹ thuật điện	THPT Quốc gia	20	2018
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Truyền thông đa phương tiện	THPT Quốc gia	20	2023
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Marketing	THPT Quốc gia	20	2023
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Công nghệ kỹ thuật ô tô	THPT Quốc gia	20	2023
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Quản trị kinh doanh	THPT Quốc gia	20	2023
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Quản trị kinh doanh	THPT Quốc gia	20	2024
ĐẠI HỌC QUỐC TẾ - ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH	Khoa học Dữ liệu	THPT Quốc gia	20	2020
ĐẠI HỌC THƯƠNG MẠI	Quản trị kinh doanh (Tiếng Trung thương mại)	THPT Quốc gia	20	2018
ĐẠI HỌC THƯƠNG MẠI	Tài chính - Ngân hàng (Tài chính – Ngân hàng thương mại)	THPT Quốc gia	20	2018
ĐẠI HỌC QUỐC TẾ - ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH	Kỹ thuật không gian	THPT Quốc gia	20	2020
ĐẠI HỌC QUỐC TẾ - ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH	Tài chính – Ngân hàng	THPT Quốc gia	20	2019
ĐẠI HỌC QUỐC TẾ - ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH	Công nghệ thông tin	THPT Quốc gia	20	2019
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Truyền thông đa phương tiện	THPT Quốc gia	20	2024
ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH	Marketing	THPT Quốc gia	20	2024

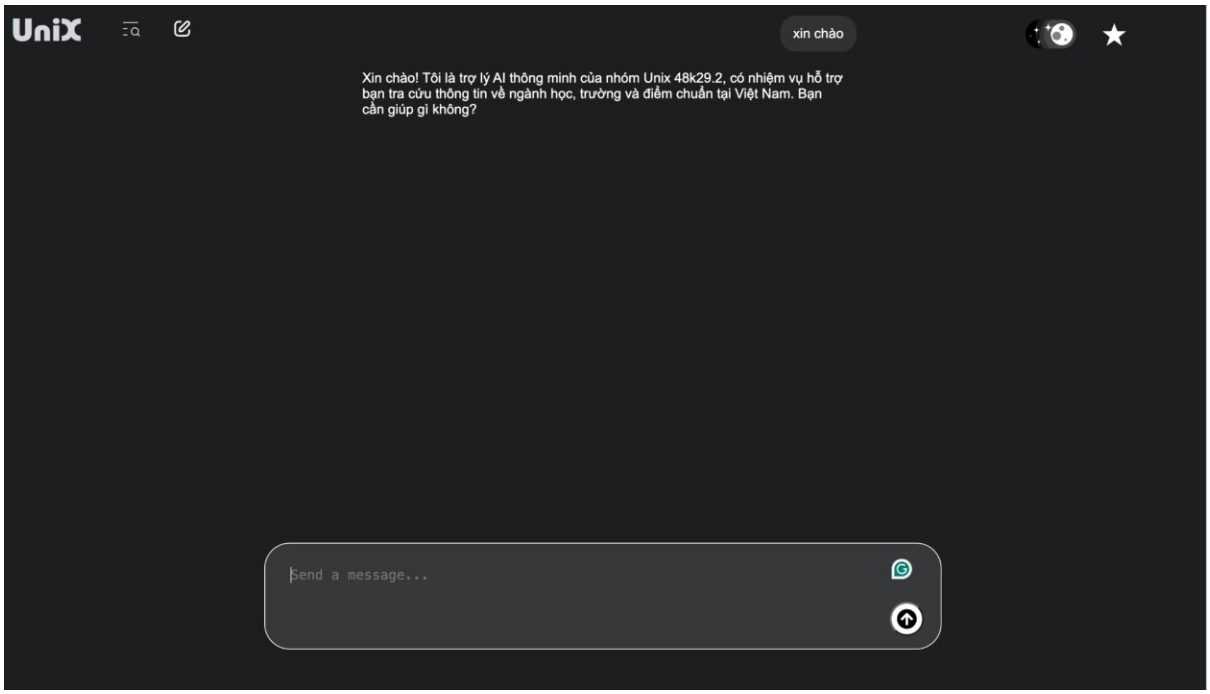
Hình 19: Gợi ý nhanh ngành và trường học theo khoảng điểm

4.1.3. Giao diện trang MBTI test



Hình 20: Giao diện MBTI test

4.1.4. Kết quả xây dựng chatbot

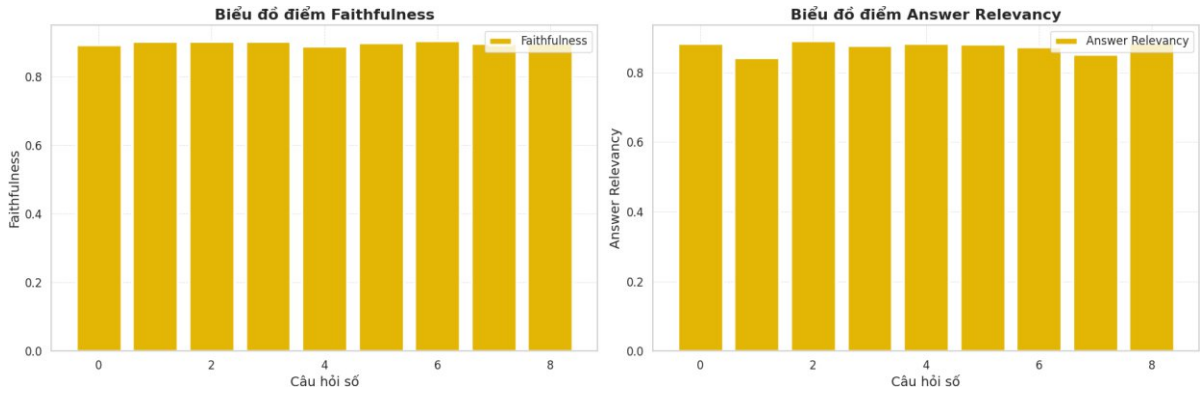


Hình 21: Giao diện của chatbot

Hình 22: Kết quả của chatbot khi trả lời người dùng

4.2. Đánh giá kết quả

4.2.1. Đánh giá mô hình RAG



Hình 23: Biểu đồ đánh giá độ tin cậy và mức độ liên quan của câu trả lời từ Chatbot

Lần thực nghiệm	Chỉ số đánh giá	
	Relevant	**Faithfulness**
1	0.8829	0.8934
2	0.8425	0.9043
3	0.8908	0.9025
4	0.8783	0.903
5	0.8833	0.89
6	0.8811	0.8995
7	0.8731	0.9051
8	0.8518	0.8971
9	0..8859	0.8972

Bảng 1: Bảng kết quả đánh giá Chatbot

1. Chỉ số Faithfulness

- Chỉ số Faithfulness dao động trong khoảng 0.89 đến 0.9051, cho thấy độ tin cậy cao giữa nội dung câu trả lời và thông tin từ nguồn dữ liệu gốc.

● Giá trị cao nhất đạt được là 0.9051 ở lần thử nghiệm thứ 7, chứng minh rằng hệ thống có khả năng giữ đúng nội dung và logic ban đầu của tài liệu tham chiếu trong quá trình tạo câu trả lời.

● Không có thử nghiệm nào có điểm dưới 0.89, điều này phản ánh tính ổn định và đáng tin cậy của mô hình trong việc duy trì tính nhất quán ngữ nghĩa.

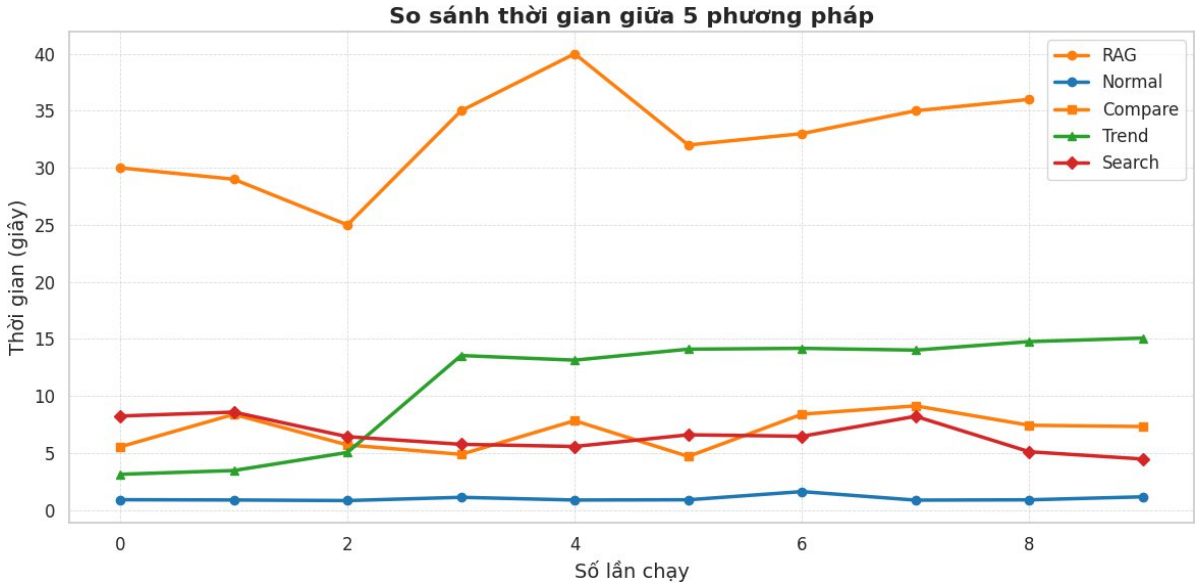
2. Chỉ số Answer Relevancy (Relevant)

● Chỉ số này dao động trong khoảng 0.8425 đến 0.8908, phản ánh mức độ phù hợp giữa câu hỏi và câu trả lời sinh ra.

● Mức cao nhất là 0.8908 ở lần thứ 3, cho thấy hệ thống hiểu và phản hồi tốt nội dung câu hỏi.

● Tuy nhiên, có một số lần (như lần 2 và lần 8) có điểm tương đối thấp (0.8425 và 0.8518), có thể do câu hỏi có tính mơ hồ hoặc mô hình chưa tối ưu hóa đủ trong việc lựa chọn thông tin liên quan

4.2.2. Đánh giá hệ thống chatbot



Hình 24: So sánh thời gian giữa các phương pháp

Lần thực nghiệm	Thời gian thực hiện				
	Câu hỏi thông thường	**Câu hỏi so sánh điểm**	**Phân tích xu hướng**	**Tra cứu điểm**	**RAG**
1	0.905571	5.540076	3.132601023	8.24146485	30
2	0.879929	8.386319	3.463269472	8.58456897	29
3	0.827968	5.708064	5.048968315	6.42977905	25
4	1.114137	4.878826	13.5444312	5.75947403	35
5	0.876013	7.833916	13.14015412	5.571167	40
6	0.898237	4.696879	14.10098457	6.59528875	32
7	1.615262	8.401493	14.17540884	6.46516180	33
8	0.867216	9.131405	14.01899099	8.21744561	35
9	0.894807	7.431742	14.7643497	5.11007213	36
10	1.158077	7.31855	15.07257891	4.47127103	

Bảng 2: Kết quả thực nghiệm của những lần thực nghiệm

Kết quả thực nghiệm cho thấy mô hình chatbot có khả năng xử lý đa dạng các loại câu hỏi với mức độ hiệu quả khác nhau về mặt thời gian:

- Hiệu năng tổng thể được đánh giá ổn định, đặc biệt đối với các loại câu hỏi phổ biến như loại câu hỏi thông thường và so sánh điểm của ngành học từ 2 trường, thời gian phản hồi nhanh và phù hợp với các ứng dụng thời gian thực.
- Khả năng phân tích nâng cao được thể hiện rõ qua việc mô hình xử lý được các câu hỏi phức tạp như phân tích xu hướng điểm và sử dụng RAG truy vấn những thông tin dưới dạng pdf, dù thời gian phản hồi cao hơn. Điều này cho thấy hệ thống có

năng lực suy luận và truy xuất thông tin chuyên sâu, phù hợp với các tình huống cần giải thích hoặc tư vấn phức tạp.

- Tuy nhiên, chi phí thời gian của phương pháp RAG còn khá lớn (trên 30 giây ở nhiều lần chạy), điều này cần được tối ưu nếu mô hình được triển khai ở quy mô lớn hoặc trong môi trường yêu cầu tốc độ phản hồi nhanh.

KẾT LUẬN VÀ KIẾN NGHỊ

1 Kết luận.

Trong nghiên cứu này, nhóm đã đề xuất và triển khai một mô hình chatbot tư vấn tuyển sinh có khả năng xử lý ngôn ngữ tự nhiên, kết hợp giữa truy xuất ngữ nghĩa và sinh văn bản (Retrieval-Augmented Generation – RAG). Mô hình được thiết kế nhằm hỗ trợ người dùng tra cứu thông tin tuyển sinh một cách chủ động, thông minh và linh hoạt theo ngữ cảnh.

Thông qua quá trình thực nghiệm, hệ thống đã thể hiện năng lực phản hồi hiệu quả với nhiều loại câu hỏi từ người dùng, bao gồm: tra cứu thông tin, so sánh giữa các trường/ngành, phân tích xu hướng điểm chuẩn, và tìm kiếm theo điều kiện cụ thể. Hai chỉ số đánh giá chất lượng câu trả lời là Answer Relevancy và Faithfulness lần lượt đạt giá trị trung bình 0.8717 và 0.8996, phản ánh mức độ chính xác và độ phù hợp cao trong quá trình sinh phản hồi.

Bên cạnh đó, mô hình cũng được đánh giá theo thời gian xử lý. Kết quả cho thấy thời gian phản hồi có sự phân hóa rõ rệt theo từng loại câu hỏi: các truy vấn thông thường (Normal) chỉ mất khoảng 1–2 giây, trong khi các truy vấn phức tạp sử dụng RAG có thể mất tới 30–40 giây. Dù vậy, thời gian xử lý vẫn nằm trong phạm vi chấp nhận được đối với các hệ thống tư vấn bán thời gian hoặc có độ ưu tiên cao về độ chính xác.

Nhìn chung, mô hình chatbot đề xuất đã đạt được các mục tiêu nghiên cứu ban đầu, thể hiện tiềm năng ứng dụng trong thực tế để hỗ trợ người học tiếp cận thông tin một cách chính xác, kịp thời và mang tính cá nhân hóa. Nghiên cứu mở ra định hướng phát triển các hệ thống hỏi – đáp chuyên biệt trong lĩnh vực giáo dục, đặc biệt trong bối cảnh chuyển đổi số ngành tuyển sinh đại học.

Tuy đạt được nhiều kết quả khả quan, mô hình vẫn tồn tại một số hạn chế nhỏ:

- Chi phí còn khá cao
- Chưa xử lý sâu các câu hỏi mơ hồ hoặc đa nghĩa, đôi khi dẫn đến phản hồi chưa sát với nhu cầu thực tế của người dùng.

2 Hướng phát triển đề tài.

Đề tài có tiềm năng phát triển mạnh mẽ trong giai đoạn tiếp theo thông qua việc mở rộng quy mô dữ liệu, tối ưu chi phí vận hành hệ thống và nâng cấp trải nghiệm người dùng. Trước tiên, một trong những định hướng trọng tâm là mở rộng bộ dữ liệu

điểm chuẩn theo cả chiều rộng và chiều sâu. Về chiều rộng, hệ thống cần tích hợp dữ liệu tuyển sinh từ hàng trăm trường đại học trên toàn quốc, bao gồm cả các chương trình đào tạo đặc thù, liên kết quốc tế và phân hiệu khu vực. Về chiều sâu, dữ liệu sẽ được bổ sung thêm các trường thông tin như: tỷ lệ chọi, điểm học bạ trung bình, điểm thi thành phần theo tổ hợp, học phí, chỉ tiêu từng năm, chính sách ưu tiên, tỷ lệ việc làm sau tốt nghiệp,... Việc xây dựng một cơ sở dữ liệu học thuật đa chiều, chuẩn hóa và kết nối liên bảng, là tiền đề cho các phân tích nâng cao và mô hình khuyến nghị chuyên sâu trong tương lai.

Một hướng đi quan trọng khác nhằm tối ưu chi phí triển khai là chuyển dịch hệ thống chatbot từ việc phụ thuộc vào các mô hình ngôn ngữ lớn thương mại (LLM-as-a-service) sang sử dụng các mô hình LLM mã nguồn mở như LLaMA, Mistral, Falcon hoặc OpenChat, kết hợp với phương pháp fine-tuning trên tập dữ liệu ngành học đặc thù. Điều này không chỉ giúp tiết kiệm đáng kể chi phí API token, mà còn cho phép kiểm soát tốt hơn quá trình tùy biến mô hình theo ngữ cảnh tiếng Việt và đặc thù tuyển sinh trong nước.

Song song đó, đề tài cũng hướng tới việc tối ưu hiệu suất hệ thống để phục vụ số lượng lớn người dùng truy cập đồng thời. Việc nâng giới hạn token request và cải tiến cơ chế xử lý truy vấn sẽ giúp chatbot có thể phản hồi nhanh, chính xác và mượt mà trong môi trường đa người dùng. Hệ thống cũng có thể ứng dụng các kỹ thuật như caching, batching request, và phân tầng truy vấn theo mức ưu tiên để tối ưu hiệu suất tổng thể.

Ngoài ra, hệ thống có thể được mở rộng để tích hợp thêm các module phân tích dự đoán khả năng trúng tuyển theo hồ sơ cá nhân, khuyến nghị lộ trình học tập, hoặc cảnh báo ngành học có xu hướng giảm điểm qua từng năm (early warning system). Một ý tưởng triển khai khác là phát triển dashboard quản trị dành riêng cho giáo viên hướng nghiệp, nơi họ có thể theo dõi xu hướng chọn ngành, hiệu suất tương tác của học sinh, và phân tích nhóm đối tượng tiềm năng theo khu vực hoặc khối thi.

TÀI LIỆU THAM KHẢO

[1] World Bank (2020). Vietnam Higher Education Development Report

[2] Zing News (2022). *Thí sinh rối như tơ vò với 20 phương thức xét tuyển*

[3] Bộ Giáo dục và Đào tạo (MOET). (2023). Khảo sát về tiếp cận thông tin tuyển sinh của học sinh THPT.

[4] Hơn 733.000 thí sinh đăng ký xét tuyển trên Hệ thống của Bộ GDĐT. Retrieved May 15, 2025 from https://moet.gov.vn/tintuc/Pages/tin-tong-hop.aspx?ItemID=9680

[5] daibieunhandan.vn. 2024. Nhiều thí sinh vẫn chọn sai ngành, sai trường, cần đẩy mạnh tư vấn hướng nghiệp năm 2024. daibieunhandan.vn. Retrieved May 15, 2025 from https://daibieunhandan.vn/nhieu-thi-sinh-van-chon-sai-nganh-sai-truong-can-day-manh-tu-van-huong-nghiep-nam-2024-10333055.html

[6] Lewis, P., Oguz, B., Rinott, R., Riedel, S., & Stoyanov, V. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv preprint arXiv:2005.11401.

[7] Jakub Swacha and Michał Gracel. 2025. Retrieval-Augmented Generation (RAG) Chatbots for Education: A Survey of Applications. Appl. Sci. 15, (April 2025), 4234. https://doi.org/10.3390/app15084234

[8] Trung Thanh Nguyen, Anh Duc Le, Ha Thanh Hoang, and Tuan Nguyen. 2021. NEU-chatbot: Chatbot for admission of National Economics University. *Comput. Educ. Artif. Intell.* 2, (January 2021), 100036. https://doi.org/10.1016/j.caeai.2021.100036

[9] Yuhong Mo, Hao Qin, Yushan Dong, Ziyi Zhu, and Zhenglin Li. 2024. Large Language Model (LLM) AI text generation detection based on transformer deep learning algorithm. https://doi.org/10.48550/arXiv.2405.06652

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762

[11] Yifei Ding, Minping Jia, Qiuhua Miao, and Yudong Cao. 2022. A novel time-frequency Transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings. Mech. Syst. Signal Process. 168, (April 2022), 108616. https://doi.org/10.1016/j.ymssp.2021.108616

[12] OpenAI 2024. G P T - 4 Technical Report. https://doi.org/10.48550/arXiv.2303.08774

[13] Gemini (language model - Wikipedia. Retrieved May 15, 2025 from https://en.wikipedia.org/wiki/Gemini_(language_model

[14] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. https://doi.org/10.48550/arXiv.1701.06538

[15] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. https://doi.org/10.48550/arXiv.2006.16668

[16] Google Developers Blog. 2024. Gemini 1.5: The Gemini 2 family expands. Google Developers Blog. Retrieved May 15, 2025 from https://developers.googleblog.com/en/gemini-2-family-expands/

[17] Gemini Robotics: Bringing AI into the Physical World. Retrieved May 15, 2025 from https://arxiv.org/html/2503.20020v1

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. https://doi.org/10.48550/arXiv.1503.02531

[19] Introduction | LangChain. Retrieved May 15, 2025 from https://python.langchain.com/docs/introduction/

[20] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. https://doi.org/10.48550/arXiv.2402.05672

[21]2025. intfloat/multilingual-e5-large-instruct · Hugging Face. Retrieved May 15, 2025 from https://huggingface.co/intfloat/multilingual-e5-large-instruct

[22]Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://doi.org/10.48550/arXiv.2005.11401

[23]Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. https://doi.org/10.48550/arXiv.2312.10997