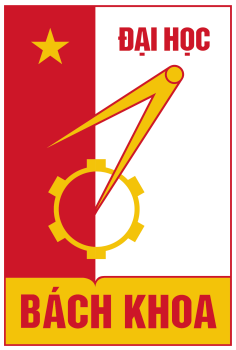
Đại học Bách khoa Hà Nội

Khoa Toán - Tin




# ĐỒ ÁN II

# XÂY DỰNG TRỢ LÝ ẢO CỐ VẤN HỌC TẬP CHO SINH VIÊN BÁCH KHOA HÀ NỘI

Giảng viên hướng dẫn: TS. Trần Ngọc Thăng
Sinh viên thực hiện: Trương Cảnh Dương
Mã số sinh viên: 20216807
Lớp: Toán-Tin 03-K66

Hà Nội, 2025

**Nhận xét của giảng viên hướng dẫn**

**Mục tiêu và nội dung của đồ án:**

.....

.....

.....

.....

**Kết quả đạt được:**

.....

.....

.....

.....

**Ý thức làm việc của sinh viên:**

.....

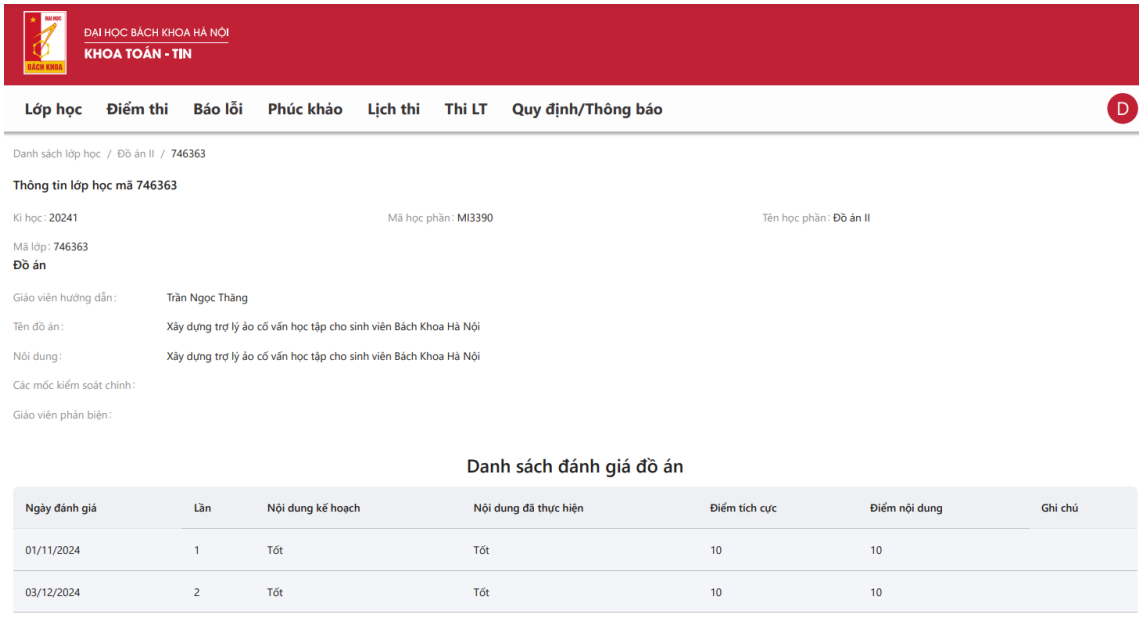
.....

.....

.....

Hà Nội, ngày ... tháng ... năm 2024
Giảng viên hướng dẫn

# PHIẾU BÁO CÁO TIẾN ĐỘ ĐỒ ÁN



ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

Lớp học | Điểm thi | Báo lỗi | Phúc khảo | Lịch thi | Thi LT | Quy định/Thông báo

Danh sách lớp học / Đồ án II / 746363

**Thông tin lớp học mã 746363**

Kì học: 20241 Mã học phần: MI3390 Tên học phần: Đồ án II

Mã lớp: 746363
**Đồ án**

Giáo viên hướng dẫn: Trần Ngọc Thăng

Tên đồ án: Xây dựng trợ lý ảo cố vấn học tập cho sinh viên Bách Khoa Hà Nội

Nội dung: Xây dựng trợ lý ảo cố vấn học tập cho sinh viên Bách Khoa Hà Nội

Các mốc kiểm soát chính:

Giáo viên phản biện:

**Danh sách đánh giá đồ án**

| Ngày đánh giá | Lần | Nội dung kế hoạch | Nội dung đã thực hiện | Điểm tích cực | Điểm nội dung | Ghi chú |
|---|---|---|---|---|---|---|
| 01/11/2024 | 1 | Tốt | Tốt | 10 | 10 | |
| 03/12/2024 | 2 | Tốt | Tốt | 10 | 10 | |

**Lời cảm ơn**

Trước hết, em xin gửi lời cảm ơn chân thành đến Khoa Toán-Tin đã tạo điều kiện thuận lợi và cung cấp môi trường học tập tốt để em có thể hoàn thành đồ án này.

Em xin bày tỏ lòng biết ơn sâu sắc đến thầy Trần Ngọc Thăng, người đã tận tình hướng dẫn, đóng góp ý kiến quý báu và hỗ trợ em trong suốt quá trình thực hiện đồ án. Sự kiên nhẫn, nhiệt tình và kiến thức chuyên môn của thầy đã giúp em vượt qua nhiều khó khăn và hoàn thiện đồ án một cách tốt nhất.

Em cũng xin cảm ơn các bạn bè đã cùng em thảo luận, chia sẻ tài liệu và góp ý trong quá trình thực hiện nghiên cứu này. Những ý kiến và sự hỗ trợ từ các bạn đã giúp em nhìn nhận vấn đề từ nhiều góc độ khác nhau và cải thiện chất lượng đồ án.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình đã luôn động viên, ủng hộ và tạo điều kiện tốt nhất cho em trong suốt quá trình học tập và thực hiện đồ án.

Mặc dù đã cố gắng hết sức, đồ án này không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý và chỉ dẫn của quý thầy cô và các bạn để tôi có thể hoàn thiện hơn trong những nghiên cứu và công việc sau này.

Em xin chân thành cảm ơn!

# Mục lục


**1 Cơ sở lý thuyết** **10**
- 1.1 Tổng quan về xử lý ngôn ngữ tự nhiên . . . 10
- 1.2 Mô hình ngôn ngữ lớn . . . 11
- 1.3 Quy trình RAG . . . 14
  - 1.3.1 Adaptive RAG . . . 15
  - 1.3.2 Self-RAG . . . 15
  - 1.3.3 Corrective RAG . . . 16

**2 Khảo sát và phân tích** **18**
- 2.1 Phát biểu bài toán . . . 18
- 2.2 Mô tả dữ liệu . . . 25
- 2.3 Giải pháp tổng thể . . . 27

**3 Công nghệ sử dụng và phương pháp** **29**
- 3.1 Xử lý dữ liệu . . . 29
- 3.2 Cơ sở dữ liệu vector . . . 31
- 3.3 Web search . . . 33
- 3.4 LangChain . . . 33
  - 3.4.1 LangGraph . . . 34
  - 3.4.2 LangSmith . . . 36
- 3.5 Mô hình Chat . . . 37
  - 3.5.1 ChatGroq . . . 37
  - 3.5.2 Llama . . . 37
- 3.6 Xây dựng phần mềm . . . 38
  - 3.6.1 React . . . 38
  - 3.6.2 FastAPI . . . 38

**4 Cài đặt chương trình và kết quả** **39**
- 4.1 Cách thức cài đặt . . . 39
- 4.2 Đánh giá mô hình . . . 40
- 4.3 Giao diện . . . 41

**Kết luận và hướng phát triển** **43**

**Chỉ mục** **44**

**Tài liệu tham khảo** **45**

# Bảng ký hiệu và chữ viết tắt

| | |
|---|---|
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| CRAG | Corrective Retrieval-Augmented Generation |
| ĐHBKHN | Đại học Bách khoa Hà Nội |
| LLM | Large Language Model |
| NLP | Natural Language Processing |
| RAG | Retrieval-Augmented Generation |
| Self-RAG | Self-Reflective Retrieval-Augmented Generation |
| SHCD | Sinh hoạt công dân |

# Danh sách hình vẽ


1.1 Kiến trúc mô hình Transformer . . . . . 11
1.2 Cây phát triển của LLM[2] . . . . . 14
1.3 Quy trình RAG . . . . . 15

2.1 Quy định về cố vấn học tập của ĐHBKHN . . . . . 18
2.2 Hệ thống thông báo và giải đáp câu hỏi của ĐHBKHN . . . . 19
2.3 Sổ tay dành cho sinh viên trên cổng thông tin đại học . . . . . 19
2.4 Quy chế đào tạo . . . . . 20
2.5 Quy định về ngoại ngữ . . . . . 20
2.6 Quy định về học bổng . . . . . 21
2.7 Quy định về miễn giảm học phí, hỗ trợ sinh viên . . . . . 21
2.8 Quy định về điểm rèn luyện . . . . . 22
2.9 Slide định hướng trong một buổi SHCD . . . . . 22
2.10 Thông tin về chương trình đào tạo trên website của khoa Toán-Tin . . . . . 23
2.11 Các trợ lý ảo với tính năng riêng trên ChatGPT . . . . . 24
2.12 Biểu đồ use case tổng quát . . . . . 24
2.13 Sơ đồ quy trình chi tiết . . . . . 27

3.1 Điều khoản trong quy chế đào tạo 2023 . . . . . 29
3.2 Nội dung phân đoạn điều khoản . . . . . 30
3.3 Nội dung hướng dẫn thư viện . . . . . 30
3.4 Nội dung phân đoạn các mục . . . . . 31
3.5 Hệ sinh thái của LangChain . . . . . 33
3.6 Một đồ thị đơn giản trên LangGraph . . . . . 35
3.7 Kiểm soát lượng sử dụng . . . . . 36
3.8 Giao diện kiểm thử trên LangSmith . . . . . 36

4.1 Kiến trúc hệ thống trợ lý ảo . . . . . 39
4.2 Đồ thị được xây dựng cho trợ lý ảo . . . . . 40
4.3 Đồ thị được xây dựng cho việc đánh giá . . . . . 40
4.4 Giao diện chính . . . . . 41
4.5 Giao diện khi trò chuyện với trợ lý ảo . . . . . 42

# Danh sách bảng

1.1 Bốn kiểu token tự phản ánh . . . . . 16

4.1 Kết quả đánh giá . . . . . 41

## Lời mở đầu

Trong những năm gần đây, việc áp dụng trí tuệ nhân tạo (AI) vào công việc thực tiễn để tự động hoá công việc và hỗ trợ sáng tạo ngày càng trở nên phổ biến. Nhu cầu tự động hoá công việc trong các doanh nghiệp vừa và nhỏ ngày càng tăng. Trợ lý ảo (Virtual Assistant) được tạo ra để hỗ trợ doanh nghiệp trong công việc và giúp khách hàng tìm hiểu về doanh nghiệp và nhận trợ giúp liên quan. Từ nhu cầu đó, đặt vào trường hợp sử dụng của Đại học Bách khoa Hà Nội, việc xây dựng trợ lý ảo có thể giúp sinh viên tìm hiểu thông tin của đại học và trợ giúp trong định hướng học tập là vô cùng cần thiết.

Hiện nay, chúng ta có thể tiếp cận với các mô hình ngôn ngữ lớn (LLM) vô cùng dễ dàng và đa dạng về cả hình thức và số lượng. Ví dụ như ChatGPT, ta có thể hỏi nó các kiến thức chung, hoặc gửi lên các tài liệu để mô hình có thể trả lời theo được. Tuy nhiên, khi số tài liệu lớn và mang tính riêng tư của một tổ chức thì các ứng dụng sẽ không thể trả lời được. Thay vào đó, ta có thể xây dựng một trợ lý ảo riêng có thể sử dụng các tài liệu, quy định của tổ chức để hỗ trợ, giải đáp. Trong đồ án này, trợ lý ảo có thể sử dụng các tài liệu, quy định của đại học và trả lời một cách khá chính xác các câu hỏi liên quan.

Các cách để một mô hình có thể học từ tài liệu đó là fine-tune và RAG (Retrieval Augmented Generation). Trong đó, fine-tune là quá trình huấn luyện lại một LLM trên một tập dữ liệu cụ thể nhằm tinh chỉnh mô hình cho một nhiệm vụ hoặc ngữ cảnh nhất định, mô hình học và lưu trữ kiến thức từ tập dữ liệu trong chính tham số của nó. Sau khi fine-tune, mô hình có thể trả lời mà không cần kết nối cơ sở dữ liệu ngoài. Fine-tune sẽ tốt nhất khi sử dụng cho một lĩnh vực, nhiệm vụ cụ thể ít khi thay đổi về dữ liệu. Tuy nhiên việc Fine-tune sẽ tiêu tốn tài nguyên khi phải train lại khi muốn thêm dữ liệu mới. Còn với RAG, quy trình sẽ linh hoạt hơn khi dữ liệu liên tục thay đổi, thêm mới.

Trợ lý ảo trong đồ án sẽ được xây dựng dựa trên kỹ thuật RAG, tập trung vào việc trả lời câu hỏi dựa trên tài liệu liên quan đến Đại học Bách khoa Hà Nội. Đồ án này sẽ chia làm 4 chương

- **Chương 1:** Cơ sở lý thuyết. Giới thiệu khái quát các khái niệm về xử lý ngôn ngữ tự nhiên, mô hình ngôn ngữ lớn, quy trình RAG và các phương pháp giúp RAG trở nên chính xác hơn.
- **Chương 2:** Phân tích và khảo sát. Thực hiện khảo sát thực tế dẫn đến bài toán cần giải quyết sau đó phân tích và đưa ra giải pháp tổng thể.
- **Chương 3:** Công nghệ sử dụng và phương pháp. Giới thiệu các công nghệ được dùng để xây dựng trợ lý ảo và phương pháp để việc trả lời của trợ lý ảo chính xác hơn.
- **Chương 4:** Cài đặt chương trình và kết quả. Nêu cấu trúc chương trình, sau đó đưa ra một kết quả đánh giá mô hình.

# Chương 1

# Cơ sở lý thuyết

## 1.1 Tổng quan về xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (NLP) là lĩnh vực thuộc một nhánh của trí tuệ nhân tạo. Lĩnh vực này tập trung vào việc giúp máy tính hiểu và diễn giải ngôn ngữ của con người. Trong khi máy tính xử lý dữ liệu có cấu trúc tốt như bảng tính hoặc cơ sở dữ liệu, ngôn ngữ tự nhiên ở dạng phi cấu trúc (văn bản, giọng nói,...) lại là một thách thức lớn. NLP sẽ giúp máy tính hiểu, phân tích, và sinh ngôn ngữ tự nhiên như con người, giúp nó là một công cụ cần thiết cho hệ thống AI hiện đại.

NLP được tạo thành dựa trên các tác vụ liên quan đến ngôn ngữ, bao gồm các bước chính là tiền xử lý ngôn ngữ, phân tích ngữ pháp, cú pháp, ngữ nghĩa.

- Tiền xử lý ngôn ngữ
  - Phân tách câu thành các từ hoặc các đơn vị nhỏ (tokenization).
  - Loại bỏ các từ không quan trọng, các từ dừng.
  - Stemming/Lemmatization: Giảm các từ về dạng gốc. Ví dụ với các từ tiếng Anh như "go", "goes", "went", kỹ thuật này sẽ làm cho các từ đấy có giá trị bằng nhau khi so sánh. Với tiếng Việt, các từ không có biến thể khác nhau nên sẽ không cần bước này.
- Phân tích ngữ pháp, cú pháp, ngữ nghĩa
  - Gán nhãn từ loại (POS tagging) cho từng từ trong các câu như danh từ, động từ, tính từ,...
  - Phân tích phụ thuộc xác định mối quan hệ giữa các từ trong câu, từ đó hiểu được ngữ pháp câu vai trò của từng từ.
  - Xác định ý nghĩa của từ và câu trong ngữ cảnh.
  - Xác định các thực thể được đặt tên (NER) như con người, địa điểm,...

NLP có nhiều phương pháp tiếp cận từ truyền thống đến hiện đại

- Phương pháp truyền thống

  - Quy tắc hóa: Sử dụng các quy tắc ngôn ngữ do con người xây dựng.
  - Xử lý thống kê: Dựa vào các mô hình thống kê như mô hình Markov ẩn hay trường điều kiện ngẫu nhiên.

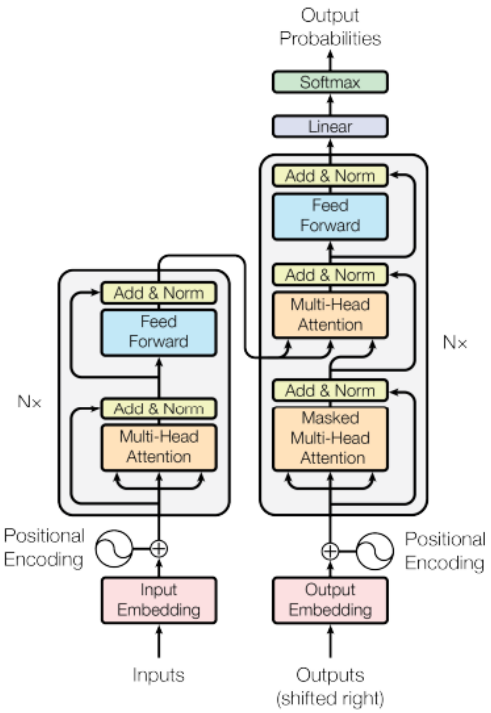
- Phương pháp hiện đại
  - Học máy: Sử dụng các thuật toán như Naïve Bayes, SVM.
  - Học sâu: Sử dụng các mô hình như RNN, LSTM, Transformer.

Trong thời gian gần đây, các phương pháp hiện đại sử dụng Transformer ngày càng trở nên phổ biến. Các mô hình ngôn ngữ như GPT, BERT, T5,... được tạo ra các bước nhảy vọt trong NLP. Chúng có khả năng hiểu được ngữ cảnh phức tạp, sinh ngôn ngữ chính xác hơn, và được học từ dữ liệu lớn.

NLP được ứng dụng trong rất nhiều tác vụ như nhận dạng giọng nói, dịch máy, chatbot và trợ lý ảo, tóm tắt văn bản, phân tích tình cảm, truy xuất thông tin,...

## 1.2 Mô hình ngôn ngữ lớn

Xử lý ngôn ngữ hiện đại có được sự bứt phá lớn khi có sự xuất hiện của các mô hình ngôn ngữ lớn (LLM). Các mô hình ngôn ngữ lớn có đặc điểm chính là các mô hình có kích thước lớn có hàng tỷ đến trăm tỷ tham số, có khả năng đa nhiệm và hiểu ngữ cảnh sâu sắc. Hiện nay các mô hình đều được xây dựng dựa trên kiến trúc Transformer.




**Hình 1.1:** Kiến trúc mô hình Transformer

Transformer là một kiến trúc mạng học sâu được giới thiệu bởi Vaswani et al. vào năm 2017 trong bài báo "Attention is All You Need"[1]. Nó đã loại bỏ

sự phụ thuộc vào mô hình tuần tự như mạng nơ-ron hồi tiếp (RNN) và mạng LSTM (Long Short-Term Memory) và tập trung vào cơ chế Attention để xử lý dữ liệu đồng thời, tăng tốc độ tính toán và cải thiện hiệu suất.

Mô hình Transformer gồm hai thành phần chính là encoder và decoder làm việc cùng nhau để xử lý và tạo ra thông tin. Encoder mã hóa câu nguồn thành một không gian biểu diễn (embedding) chung, còn Decoder sử dụng biểu diễn này để tạo ra chuỗi đầu ra, chẳng hạn như một câu dịch.

Encoder bao gồm các thành phần là Self-Attention và Feedforward neural networks. Trong đó, Self-Attention là một trong những cơ chế quan trọng nhất trong Transformer. Self-Attention là cơ chế cho phép mỗi từ trong câu "chú ý" đến tất cả các từ khác trong câu để học được các mối liên hệ giữa chúng. Self-Attention có trọng số được tính theo công thức Scaled Dot-Product Attention:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Trong đó:

- $Q$ (Query) là vector đại diện của từ hiện tại.
- $K$ (Key là vector đại diện của tất cả từ trong chuỗi.
- $V$ (Value) là vector giá trị được sử dụng để tổng hợp thông tin.
- $d_k$ là kích thước của vector Key (dùng để chuẩn hóa).

Thay vì thực hiện một hàm *Attention* thì ta có thể thực hiện Multi-Head Attention. Từ cho phép mô hình học được nhiều biểu diễn của Self-Attention, giúp mô hình có thể học các mối quan hệ phức tạp giữa các từ.

Công thức cho Multi-Head Attention là:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

Trong đó,

- $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$
  Với ma trận trọng số $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$
- Ma trận trọng số $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

Sau khi mỗi từ đã qua quá trình Self-Attention, chúng sẽ được đưa qua Feedforward Neural Networks (FNN) để tạo ra một biểu diễn cuối cùng của từ đó. Mạng Feedforward có thể coi là một lớp xử lý đơn giản gồm các tính toán tuyến tính và hàm kích hoạt.

Trong Decoder, có sự khác biệt khi Self-Attention được cộng thêm giá trị masked để tránh việc nhìn trước tương lai:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V$$

Để kết nối Enocder và Decoder, ta có công thức cho Encoder-Decoder Attention là:

$$Attention(Q_{decoder}, K_{encoder}, V_{encoder}) = softmax\left(\frac{Q_{decoder}K_{encoder}^{T}}{\sqrt{d_k}}\right)V_{encoder}$$

Mỗi lớp trong encoder và decoder còn có các bước Layer Normalization và Residual Connections để đảm bảo tính ổn định khi lan truyền gradient.

Các từ khi đi vào mô hình đều sẽ trải qua quá trình chuyển đổi các từ hoặc ký tự thành các vector số có độ dài cố định hay là vector embedding. Các vector thường có số chiều lớn và có khả năng miêu tả được mối liên hệ, sự tương đồng về mặt ngữ nghĩa, ngữ cảnh của dữ liệu. Sau đó các vector embedding được cộng thêm vector mã hoá vị trí (Positional Encoding) nhằm cung cấp thông tin về vị trí của từ trong câu. Cách tính Positional Encoding trong Transformer thường dùng công thức sau:

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d}}\right)$$

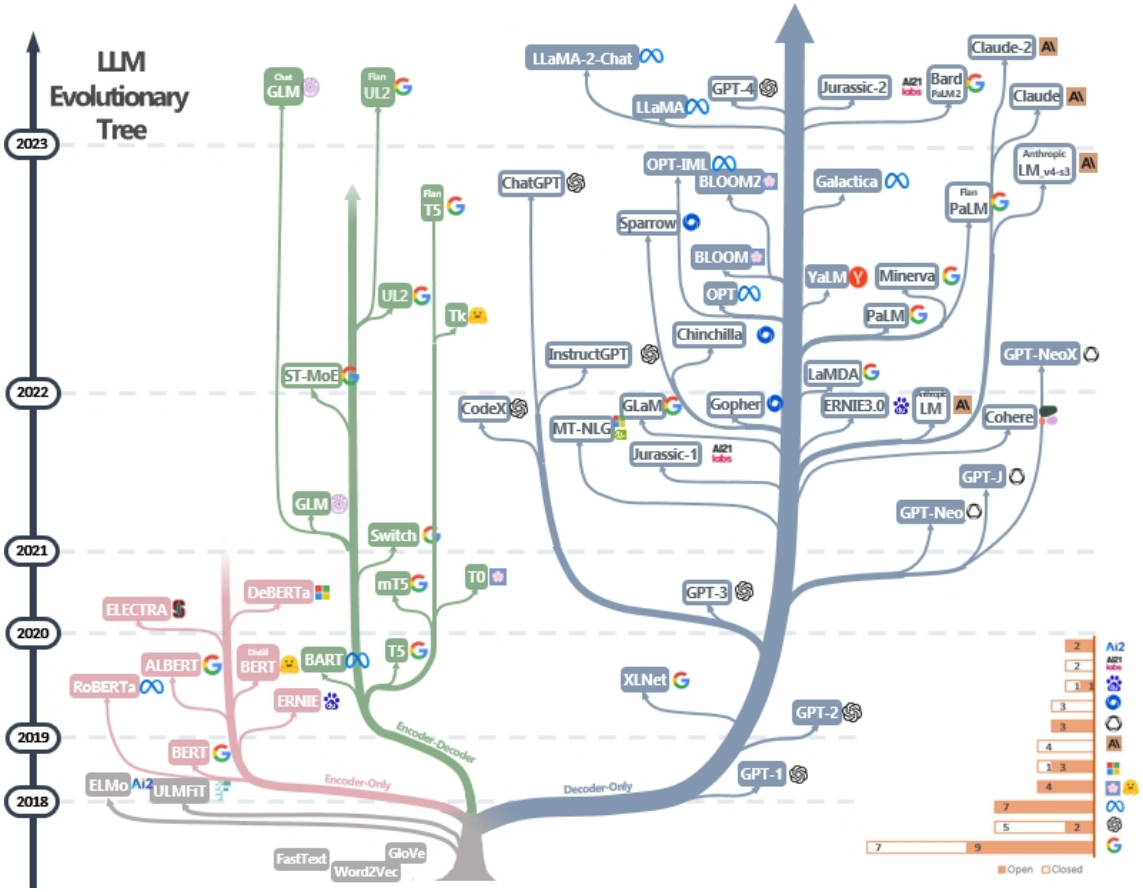
$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d}}\right)$$

Trong đó:

- $pos$ là chỉ số vị trí của từ trong câu.
- $i$ là chỉ số chiều của vector embedding.
- $d$ là số chiều của vector embedding trong mô hình embedding.

Các mô hình ngôn ngữ lớn sẽ được xây dựng dựa trên mô hình Transformer với đủ hai khối encoder-decoder hoặc một trong hai khối trên. Một số LLM nổi bật có thể kể đến là:

- BERT (Bidirectional Encoder Representation from Transformer) là mô hình được huấn luyện để hiểu ngữ cảnh từ cả hai chiều của văn bản.
- GPT (Generative Pre-trained Transformer) là mô hình tập trung vào việc sinh ngôn ngữ tự nhiên.
- T5 (Text-to-Text Transfer Transformer): Chuyển đổi mọi nhiệm vụ NLP về định dạng đầu vào - đầu ra dạng văn bản, đơn giản hóa quá trình xử lý.
- Llama (Large Language Model Meta AI) được phát triển bởi Meta để nghiên cứu và phát triển các mô hình ngôn ngữ lớn, với mục tiêu cung cấp các công cụ mở giúp cộng đồng dễ tiếp cận và nghiên cứu.




**Hình 1.2:** Cây phát triển của LLM[2]

Mô hình ngôn ngữ lớn có thể đảm nhiệm các ứng dụng chính của NLP nhưng cũng có các thách thức về tài nguyên tính toán, khả năng kiểm soát và tính bảo mật.

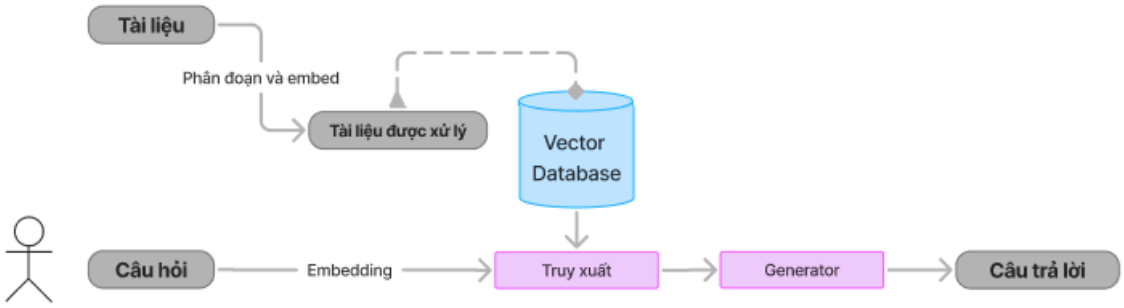
## 1.3 Quy trình RAG

Các mô hình ngôn ngữ lớn mặc dù được huấn luyện trên một tập dữ liệu lớn nhưng không thể tự cập nhật thêm thông tin. Dẫn đến thông tin mà chúng sinh ra có thể bị sai hoặc lỗi thời. Với các dữ liệu riêng của một tổ chức, doanh nghiệp, vì tính bảo mật hoặc chuyên biệt mà các LLM sẽ không thể có thông tin dẫn đến các câu trả lời sai, phản hồi chung, hoặc tạo ra ảo giác (hallucination), tức là tự tin đưa ra phản hồi sai dựa trên các sự kiện không có thật. Do vậy, kỹ thuật Retrieval-Augmented Generation được sử dụng để cải thiện và chuyên biệt hoá các mô hình ngôn ngữ lớn.

Retrieval-Augmented Generation (RAG) là phương pháp kết hợp việc truy xuất thông tin từ các tài liệu, văn bản (có thể nằm trong một cơ sở dữ liệu) và việc sinh văn bản của một LLM với mục đích cải thiện hiệu suất và độ chính xác[3].

Đầu vào của quy trình sẽ là câu hỏi (truy vấn) của người dùng. Câu hỏi đó sẽ được chuyển về dạng vector embedding. Từ vector embedding của câu hỏi, ta sẽ tìm kiếm trong một cơ sở dữ liệu chứa các tài liệu văn bản đã được xử lý bằng nhiều công cụ tìm kiếm. Sau đó, một mô hình ngôn ngữ lớn được

sử dụng để sinh ra câu trả lời dựa trên các tài liệu đã tìm được và trả về cho người hỏi.




**Hình 1.3:** Quy trình RAG

Kỹ thuật RAG sẽ cho ta khả năng cập nhật linh hoạt khi chỉ cần cập nhật tài liệu trong cơ sở dữ liệu khi có thông tin mới mà không cần huấn luyện lại mô hình, mô hình cũng không cần lưu trữ toàn bộ kiến thức trong tham số, do đó kích thước mô hình có thể giữ nguyên, không cần tốn tài nguyên để huấn luyện lại mô hình.

Tuy nhiên RAG cũng có một số nhược điểm khi phải phụ thuộc vào chất lượng truy xuất: Nếu hệ thống tìm kiếm không tốt hoặc cơ sở dữ liệu không đầy đủ, mô hình có thể sinh ra thông tin không chính xác. Việc quản lý và tìm kiếm trong các cơ sở dữ liệu lớn cũng là một vấn đề lớn, đặc biệt là khi kích thước, số lượng và kiểu tài liệu tăng lên. Ngoài ra, do có thêm bước truy xuất dữ liệu nên hiệu suất cũng giảm đi.

### 1.3.1 Adaptive RAG

Adaptive RAG là phương pháp giúp mô hình RAG thích nghi linh hoạt với độ phức tạp của câu hỏi[4], phương pháp chọn chiến lược tối ưu giữa:

- Không truy xuất (Non-Retrieval): Dùng kiến thức có sẵn của mô hình cho câu hỏi đơn giản.
- Tiếp cận một bước (Single-step approach): Kết hợp tài liệu ngoài cho câu hỏi trung bình và câu trả lời không bị .
- Tiếp cận đa bước (Multi-step approach): Sử dụng nhiều tài liệu và logic suy luận phức tạp được sử dụng cho câu hỏi khó hoặc khi câu trả lời bị sai.

### 1.3.2 Self-RAG

Các LLM thường tạo ra các phản hồi chưa chính xác do chỉ dựa vào các kiến thức mà chúng bao hàm. Việc truy xuất và kết hợp một số lượng cố định các đoạn văn đã truy xuất qua RAG, bất kể việc truy xuất có cần thiết hay các đoạn văn có liên quan hay không, sẽ làm giảm tính linh hoạt của LLM hoặc có thể dẫn đến việc tạo phản hồi không hữu ích nên Self-RAG nâng cao chất lượng và tính thực tế của LLM qua việc truy xuất và tự phản ánh.

Mô hình sử dụng token phản ánh để đánh giá tính hữu ích của truy xuất và chất lượng đầu ra. Các token phản ánh gồm:

| Token | Định nghĩa | Đầu ra |
|---|---|---|
| **Retrieve** | Quyết định khi nào cần truy xuất | {yes, no, continue} |
| **IsRel** | Đánh giá độ liên quan của tài liệu truy xuất | {relevant, irrelevant} |
| **IsSup** | Xác định liệu đầu ra có được hỗ trợ bởi tài liệu truy xuất không? | {fully supported, partially supported, no support} |
| **IsUse** | Đánh giá tổng quan tính hữu ích của đầu ra | {1, 2, 3,...} |

**Bảng 1.1:** Bốn kiểu token tự phản ánh

Dưới đây là thuật toán đề xuất từ bài báo về Self-RAG

**Thuật toán 1: Self-RAG[5]**

**Yêu cầu:** Mô hình sinh $\mathcal{G}$, truy xuất $\mathcal{R}$, Bộ tài liệu $\{d_1, \ldots, d_n\}$
**Input:** Promt đầu vào $x$, văn bản sinh trước đó $y_{<t}$
**Output:** Phân đoạn đầu ra tiếp theo $y_t$
$\mathcal{G}$ dự đoán **Retrieve** với $(x, y_{<t})$ cho trước
**if** ***Retrieve*** $==$ *Yes* **then**
    truy xuất văn bản phù hợp **D** sử dụng $\mathcal{R}$ với $(x, y_{t-1})$
    $\mathcal{G}$ dự đoán **IsRel** với $x, d$, và $y_{<t}$ với mỗi $d \in \mathbf{D}$
    $\mathcal{G}$ dự đoán **IsSup** và **IsUse** với $x, y_t, d$ với mỗi $d \in \mathbf{D}$
    Xếp hạng $y_t$ dựa trên **IsRel**, **IsSup**, **IsUse**
**else if** ***Retrieve*** $==$ *No* **then**
    $\mathcal{G}_{\text{gen}}$ dự đoán $y_t$ với $x$ cho trước
    $\mathcal{G}_{\text{gen}}$ dự đoán **IsUse** với $x, y_t$ cho trước

Self-RAG sẽ kích hoạt truy xuất khi cần thiết, giảm thiểu tài nguyên tính toán so với phương pháp RAG truyền thống. Khi truy xuất, Self-RAG phân tích và chọn lọc tài liệu, sau đó tạo đầu ra từ các tài liệu phù hợp. Mô hình sẽ đánh giá các phân đoạn đầu ra dựa trên tính đúng đắn và sự hỗ trợ từ tài liệu. Điều này giúp tối ưu hóa quá trình sinh văn bản trong thời gian suy luận.

### 1.3.3 Corrective RAG

CRAG sẽ đánh giá mức độ liên quan của tài liệu, từ đó kích hoạt các hành động:

- Correct (Đúng): Tinh chỉnh thông tin từ tài liệu hiện tại.
- Incorrect (Sai): Bỏ tài liệu sai và thay thế bằng tài liệu từ việc web search.
- Ambiguous (Mơ hồ): Kết hợp cả hai loại thông tin trên.

Dưới đây là thuật toán đề xuất từ bài báo về CRAG

**Thuật toán 2: Corrective-RAG[6]**

**Yêu cầu:** Mô hình sinh $\mathcal{M}$, Đánh giá truy xuất $\mathcal{E}$, thành phần viết lại truy vấn $\mathcal{W}$
**Input:** Câu hỏi đầu vào $x$, bộ tài tài liệu đã truy xuất
$\mathbf{D} = \{d_1, d_2, ..., d_k\}$
**Output:** $y$ là câu trả lời được sinh ra
$score_i = E$ đánh giá độ phù hợp của từng cặp $(x, d_i), d_i \in \mathbf{D}$
Đặt **Confidence** $= \{score_1, score_2, ..., score_k\}$
// **Confidence** có thể mang các giá trị: [Correct], [Incorrect] hoặc [Ambiguous]
**if** ***Confidence*** $==$ *Correct* **then**
    InternalKnowledge = KnowledgeRefine$(x, \mathbf{D})$
    k = InternalKnowledge
**else if** ***Confidence*** $==$ *Incorrect* **then**
    ExternalKnowledge = WebSearch ($\mathcal{W}$ viết lại $x$ để tìm kiếm)
    k = ExternalKnowledge
**else if** ***Confidence*** $==$ *Ambiguous* **then**
    InternalKnowledge = KnowledgeRefine$(x, \mathbf{D})$
    ExternalKnowledge = WebSearch ($\mathcal{W}$ viết lại $x$ để tìm kiếm)
    k = InternalKnowledge + ExternalKnowledge
$\mathcal{G}$ dự đoán $y$ dựa vào $x$ và $k$

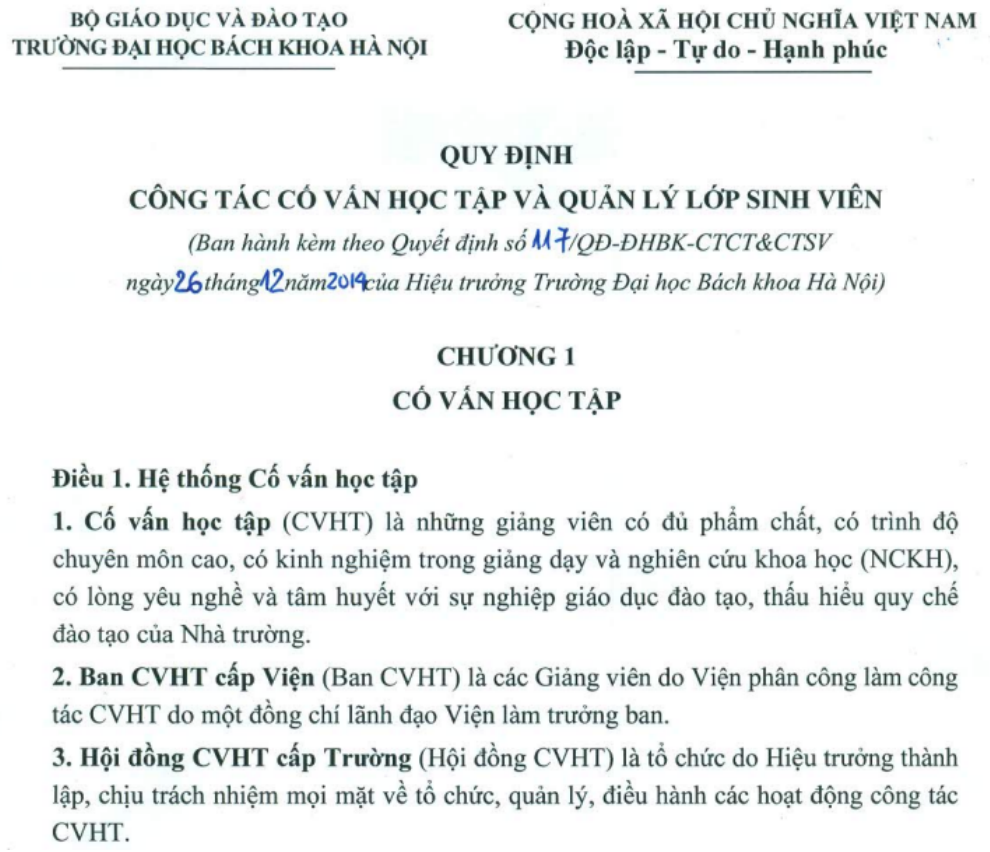
Quy trình sẽ bắt đầu với câu hỏi ban đầu và tài liệu được truy xuất. Khi tài liệu truy xuất không đủ chính xác, CRAG sử dụng tìm kiếm web để bổ sung nguồn tri thức đa dạng.

# Chương 2

# Khảo sát và phân tích

## 2.1 Phát biểu bài toán

Khi sinh viên trúng tuyển Đại học Bách khoa Hà Nội (ĐHBKHN) sẽ được chia vào các lớp, mỗi lớp sẽ có một cố vấn học. Cố vấn học tập sẽ là giảng viên có nhiệm vụ tư vấn về công tác học tập, nghiên cứu khoa học, và định hướng nghề nghiệp. Cố vấn sẽ là người đồng hành với sinh viên suốt quá trình học tập trên trường, luôn quan tâm đến lợi ích sinh viên và hết lòng giúp đỡ sinh viên.



BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

QUY ĐỊNH
CÔNG TÁC CỐ VẤN HỌC TẬP VÀ QUẢN LÝ LỚP SINH VIÊN
*(Ban hành kèm theo Quyết định số 117/QĐ-ĐHBK-CTCT&CTSV ngày 26 tháng 12 năm 2014 của Hiệu trưởng Trường Đại học Bách khoa Hà Nội)*

CHƯƠNG 1
CỐ VẤN HỌC TẬP

**Điều 1. Hệ thống Cố vấn học tập**

**1. Cố vấn học tập** (CVHT) là những giảng viên có đủ phẩm chất, có trình độ chuyên môn cao, có kinh nghiệm trong giảng dạy và nghiên cứu khoa học (NCKH), có lòng yêu nghề và tâm huyết với sự nghiệp giáo dục đào tạo, thấu hiểu quy chế đào tạo của Nhà trường.

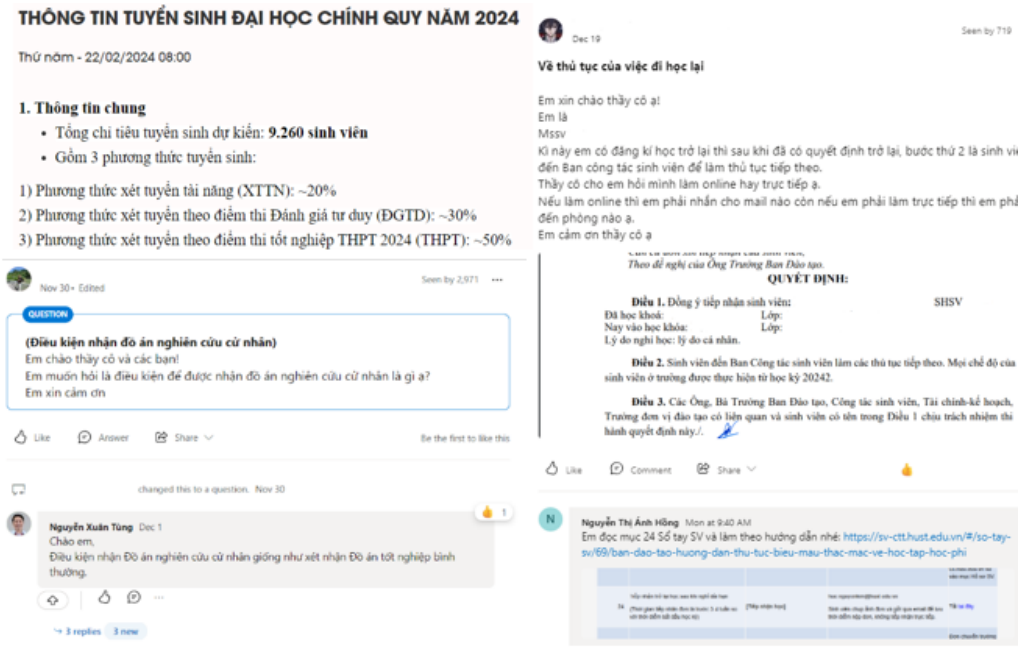
**2. Ban CVHT cấp Viện** (Ban CVHT) là các Giảng viên do Viện phân công làm công tác CVHT do một đồng chí lãnh đạo Viện làm trưởng ban.

**3. Hội đồng CVHT cấp Trường** (Hội đồng CVHT) là tổ chức do Hiệu trưởng thành lập, chịu trách nhiệm mọi mặt về tổ chức, quản lý, điều hành các hoạt động công tác CVHT.

**Hình 2.1:** Quy định về cố vấn học tập của ĐHBKHN

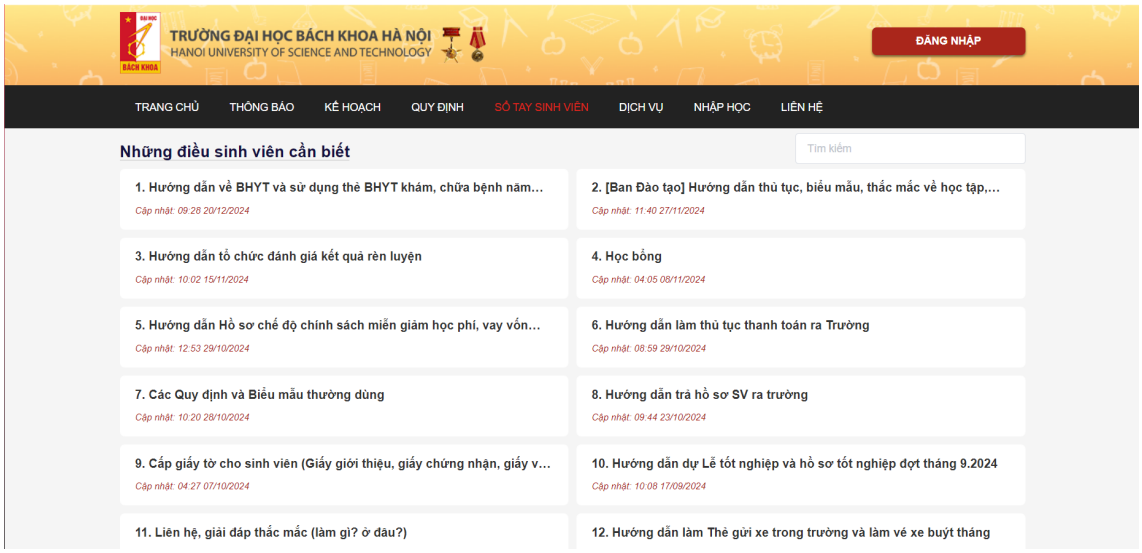
Không chỉ có cố vấn học tập riêng cho từng lớp, trong hệ thống thông tin

của ĐHBKHN có những diễn đàn cho việc giải đáp như Teams, Viva Engage (ngày trước là Yammer).




**Hình 2.2:** Hệ thống thông báo và giải đáp câu hỏi của ĐHBKHN

Ngoài những kênh hỏi đáp, ĐHBKHN còn có cổng thông tin riêng để sinh viên truy cập tìm hiểu về các quy định, quy chế và sử dụng các dịch vụ online của đại học. Sinh viên có thắc mắc có thể tra cứu thông tin trên trang web https://ctt.hust.edu.vn.




**Hình 2.3:** Sổ tay dành cho sinh viên trên cổng thông tin đại học

Trên cổng thông tin đại học có các tài liệu quy định thường dùng của ĐHBKHN như quy chế đào tạo, quy định về ngoại ngữ, quy định về học bổng, quy định về miễn giảm học phí, hỗ trợ sinh viên và quy định về điểm rèn luyện. Trong các quy chế/quy định sẽ có đầy đủ các thông tin về học tập, rèn luyện cũng như các hỗ trợ của trường cho sinh viên.

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC BÁCH KHOA HÀ NỘI

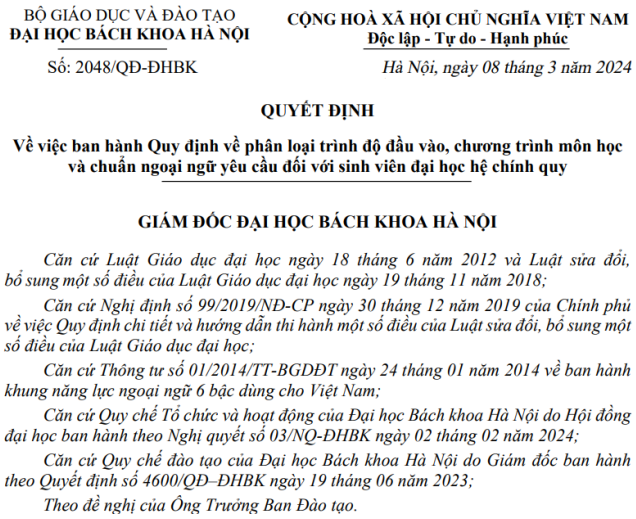
QUY CHẾ ĐÀO TẠO

*(Ban hành kèm theo Quyết định số 4600/QĐ–ĐHBK ngày 09 tháng 6 năm 2023 của Giám đốc Đại học Bách khoa Hà Nội)*

HÀ NỘI, 06-2023

**Hình 2.4:** Quy chế đào tạo

Quy chế đào tạo là là văn bản quy định đầy đủ về các mảng trong học tập của đại học. Một số mảng quan trọng mà nhiều sinh viên quan tâm được nêu trong quy chế như quy định về tín chỉ, điểm học phần, chương trình đào tạo, đăng ký học tập,...



BỘ GIÁO DỤC VÀ ĐÀO TẠO
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

Số: 2048/QĐ-ĐHBK

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập - Tự do - Hạnh phúc**

*Hà Nội, ngày 08 tháng 3 năm 2024*

**QUYẾT ĐỊNH**

**Về việc ban hành Quy định về phân loại trình độ đầu vào, chương trình môn học và chuẩn ngoại ngữ yêu cầu đối với sinh viên đại học hệ chính quy**

**GIÁM ĐỐC ĐẠI HỌC BÁCH KHOA HÀ NỘI**

*Căn cứ Luật Giáo dục đại học ngày 18 tháng 6 năm 2012 và Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học ngày 19 tháng 11 năm 2018;*

*Căn cứ Nghị định số 99/2019/NĐ-CP ngày 30 tháng 12 năm 2019 của Chính phủ về việc Quy định chi tiết và hướng dẫn thi hành một số điều của Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học;*

*Căn cứ Thông tư số 01/2014/TT-BGDĐT ngày 24 tháng 01 năm 2014 về ban hành khung năng lực ngoại ngữ 6 bậc dùng cho Việt Nam;*

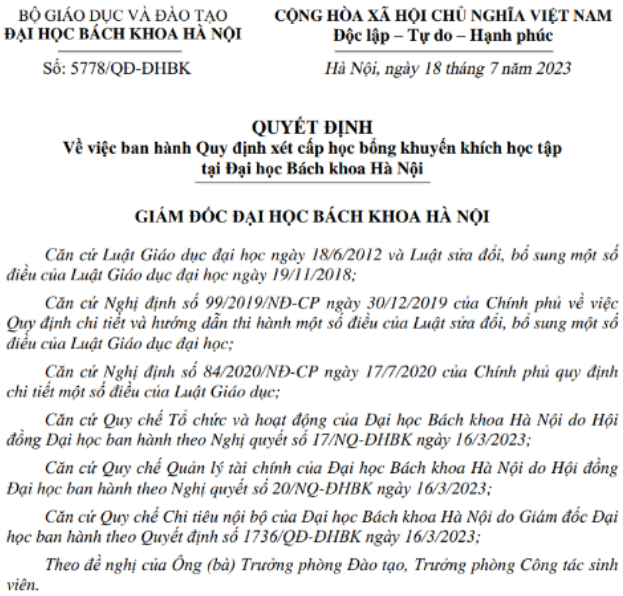
*Căn cứ Quy chế Tổ chức và hoạt động của Đại học Bách khoa Hà Nội do Hội đồng đại học ban hành theo Nghị quyết số 03/NQ-ĐHBK ngày 02 tháng 02 năm 2024;*

*Căn cứ Quy chế đào tạo của Đại học Bách khoa Hà Nội do Giám đốc ban hành theo Quyết định số 4600/QĐ–ĐHBK ngày 19 tháng 06 năm 2023;*

*Theo đề nghị của Ông Trưởng Ban Đào tạo.*

**Hình 2.5:** Quy định về ngoại ngữ

Quy định ngoại ngữ giúp sinh viên biết được các ngoại ngữ được dạy trong đại học, yêu cầu về ngoại ngữ trong các giai đoạn học tập trên đại học, cách chuyển đổi điểm giữa các bằng trong một ngoại ngữ.



BỘ GIÁO DỤC VÀ ĐÀO TẠO
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

Số: 5778/QĐ-ĐHBK

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập – Tự do – Hạnh phúc**

*Hà Nội, ngày 18 tháng 7 năm 2023*

**QUYẾT ĐỊNH**
**Về việc ban hành Quy định xét cấp học bổng khuyến khích học tập tại Đại học Bách khoa Hà Nội**

**GIÁM ĐỐC ĐẠI HỌC BÁCH KHOA HÀ NỘI**

*Căn cứ Luật Giáo dục đại học ngày 18/6/2012 và Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học ngày 19/11/2018;*

*Căn cứ Nghị định số 99/2019/NĐ-CP ngày 30/12/2019 của Chính phủ về việc Quy định chi tiết và hướng dẫn thi hành một số điều của Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học;*

*Căn cứ Nghị định số 84/2020/NĐ-CP ngày 17/7/2020 của Chính phủ quy định chi tiết một số điều của Luật Giáo dục;*

*Căn cứ Quy chế Tổ chức và hoạt động của Đại học Bách khoa Hà Nội do Hội đồng Đại học ban hành theo Nghị quyết số 17/NQ-ĐHBK ngày 16/3/2023;*

*Căn cứ Quy chế Quản lý tài chính của Đại học Bách khoa Hà Nội do Hội đồng Đại học ban hành theo Nghị quyết số 20/NQ-ĐHBK ngày 16/3/2023;*

*Căn cứ Quy chế Chi tiêu nội bộ của Đại học Bách khoa Hà Nội do Giám đốc Đại học ban hành theo Quyết định số 1736/QĐ-ĐHBK ngày 16/3/2023;*

*Theo đề nghị của Ông (bà) Trưởng phòng Đào tạo, Trưởng phòng Công tác sinh viên.*

**Hình 2.6:** Quy định về học bổng

ĐHBKHN thường xuyên có các học bổng để hỗ trợ sinh viên khó khăn và khuyến khích cho sinh viên có thành tích xuất sắc. Đại học hiện có 5 loại học bổng là học bổng khuyến khích học tập, học bổng Trần Đại Nghĩa, học bổng tài trợ, học bổng trao đổi sinh viên quốc tế và học bổng gắn kết quê hương. Với nhiều loại học bổng như vậy thì trường cũng có các quy định riêng cho học bổng.

BỘ GIÁO DỤC VÀ ĐÀO TẠO
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

Số: 5776/QĐ-ĐHBK

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập – Tự do – Hạnh phúc**

*Hà Nội, ngày 18 tháng 7 năm 2023*

**QUYẾT ĐỊNH**
**Ban hành Quy định về việc miễn, giảm học phí, hỗ trợ chi phí học tập, hỗ trợ học tập cho sinh viên Đại học Bách khoa Hà Nội**

**GIÁM ĐỐC ĐẠI HỌC BÁCH KHOA HÀ NỘI**

*Căn cứ Luật Giáo dục đại học ngày 18/6/2012 và Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học ngày 19/11/2018;*

*Căn cứ Nghị định số 99/2019/NĐ-CP ngày 30/12/2019 của Chính phủ về việc Quy định chi tiết và hướng dẫn thi hành một số điều của Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học;*

*Căn cứ Nghị định số 81/2021/NĐ-CP ngày 27/8/2021 của Chính phủ quy định về cơ chế thu, quản lý học phí đối với cơ sở giáo dục thuộc hệ thống giáo dục quốc dân và chính sách miễn, giảm học phí, hỗ trợ chi phí học tập; giá dịch vụ trong lĩnh vực giáo dục, đào tạo;*

*Căn cứ Nghị định 75/2021/NĐ-CP mức trợ cấp, phụ cấp và các chế độ ưu đãi người có công với cách mạng*

*Căn cứ Quyết định số 66/2013/QĐ-TTg ngày 11/11/2013 của Thủ tướng Chính phủ quy định chính sách hỗ trợ chi phí học tập đối với sinh viên là người dân tộc thiểu số học tại các cơ sở giáo dục đại học;*

*Căn cứ Nghị định số 57/2017/NĐ-CP ngày 09/5/2017 của Chính phủ quy định chính sách ưu tiên tuyển sinh và hỗ trợ học tập đối với trẻ mẫu giáo, học sinh, sinh viên dân tộc thiểu số rất ít người;*

*Căn cứ Nghị định 20/2021/NĐ-CP của Chính phủ về việc quy định chính sách trợ giúp xã hội đối với đối tượng bảo trợ xã hội;*

*Căn cứ Thông tư liên tịch số 35/2014/TTLT-BGDĐT-BTC ngày 15/10/2015 giữa Bộ Giáo dục và Đào tạo, Bộ Tài chính hướng dẫn thực hiện Quyết định số 66/2013/QĐ-TTg ngày 11/11/2013 của Thủ tướng Chính phủ quy định chính sách hỗ trợ chi phí học tập đối với sinh viên là người dân tộc thiểu số học tại các cơ sở giáo dục đại học;*

*Căn cứ Quy chế Tổ chức và hoạt động của Đại học Bách khoa Hà Nội do Hội đồng Đại học ban hành theo Nghị quyết số 17/NQ-ĐHBK ngày 16/3/2023;*

*Căn cứ Quy chế Quản lý tài chính của Đại học Bách khoa Hà Nội do Hội đồng Đại học ban hành theo Nghị quyết số 20/NQ-ĐHBK ngày 16/3/2023;*

*Căn cứ Quy chế Chi tiêu nội bộ của Đại học Bách khoa Hà Nội do Giám đốc Đại học ban hành theo Quyết định số 1736/QĐ-ĐHBK ngày 16/3/2023;*

*Theo đề nghị của Ông (bà) Trưởng phòng Công tác sinh viên.*

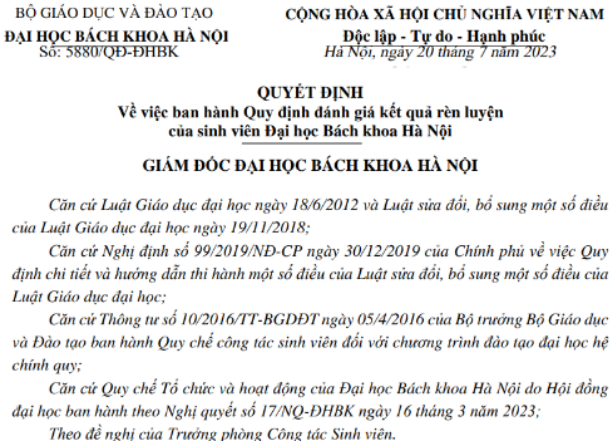
**QUYẾT ĐỊNH:**

**Điều 1.** Ban hành Quy định về việc miễn, giảm học phí, hỗ trợ chi phí học tập, hỗ trợ học tập cho sinh viên Đại học Bách khoa Hà Nội.

**Điều 2.** Quyết định này có hiệu lực kể từ ngày ký và được áp dụng từ năm học 2023-2024.

**Hình 2.7:** Quy định về miễn giảm học phí, hỗ trợ sinh viên

Để hỗ trợ sinh viên, ngoài các loại học bổng thì đại học cũng có các chính sách nhằm hỗ trợ sinh viên có hoàn cảnh khó khăn. Các hình thức hỗ trợ có miễn giảm học phí, hỗ trợ chi phí học tập cho sinh viên. Nhà trường cũng có các quy định rõ về việc miễn giảm và phương thức chi trả,...



BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC BÁCH KHOA HÀ NỘI
Số: 5880/QĐ-ĐHBK

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
*Hà Nội, ngày 20 tháng 7 năm 2023*

**QUYẾT ĐỊNH**
**Về việc ban hành Quy định đánh giá kết quả rèn luyện của sinh viên Đại học Bách khoa Hà Nội**

**GIÁM ĐỐC ĐẠI HỌC BÁCH KHOA HÀ NỘI**

*Căn cứ Luật Giáo dục đại học ngày 18/6/2012 và Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học ngày 19/11/2018;*

*Căn cứ Nghị định số 99/2019/NĐ-CP ngày 30/12/2019 của Chính phủ về việc Quy định chi tiết và hướng dẫn thi hành một số điều của Luật sửa đổi, bổ sung một số điều của Luật Giáo dục đại học;*

*Căn cứ Thông tư số 10/2016/TT-BGDĐT ngày 05/4/2016 của Bộ trưởng Bộ Giáo dục và Đào tạo ban hành Quy chế công tác sinh viên đối với chương trình đào tạo đại học hệ chính quy;*

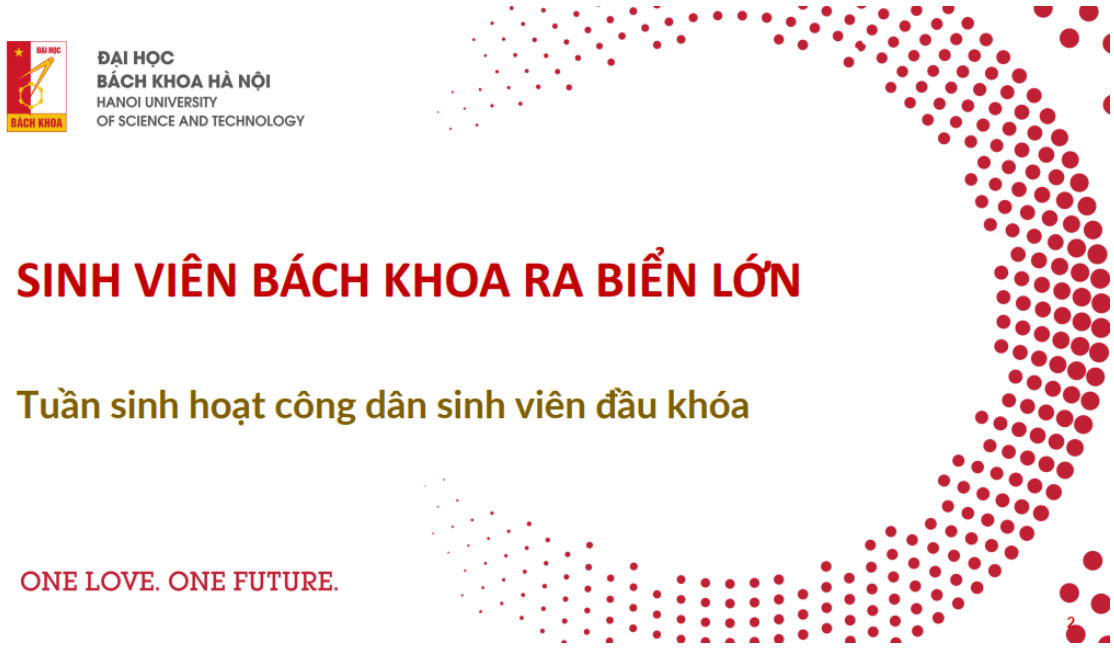
*Căn cứ Quy chế Tổ chức và hoạt động của Đại học Bách khoa Hà Nội do Hội đồng đại học ban hành theo Nghị quyết số 17/NQ-ĐHBK ngày 16 tháng 3 năm 2023;*

*Theo đề nghị của Trưởng phòng Công tác Sinh viên.*

**Hình 2.8:** Quy định về điểm rèn luyện

Điểm rèn luyện dùng để đánh giá mức độ tham gia các phong trào, thái độ học tập, tính chuyên cần của sinh viên trong suốt quá trình học. Để dễ dàng quản lý cách thức và quy trình đánh giá điểm rèn luyện.

Ngoài những quy định, quy chế các thông tin hướng dẫn sinh viên trong cấc buổi sinh hoạt công dân (SHCD) cũng nguồn tài liệu giúp sinh viên tìm hiểu thêm để tự giải đáp thắc mắc.




**Hình 2.9:** Slide định hướng trong một buổi SHCD

Các tài liệu về chương trình đào tạo của tất cả ngành học của trường cũng được công khai trên các website riêng của trường, khoa, viện. Sinh viên cũng có thể truy cập các website riêng để tìm hiểu rõ hơn về từng trường, khoa, viện.


**Hình 2.10:** Thông tin về chương trình đào tạo trên website của khoa Toán-Tin

Chương trình đào tạo của các ngành đều ghi rõ tín chỉ, khối lượng, thời gian học của từng học phần. Sinh viên có thể lập ra kế hoạch học tập trong tương lai. Tuy nhiên sẽ mất thời gian để xem hết toàn bộ chương trình đào tạo và tìm ra kế hoạch phù hợp.

Không chỉ thế, hệ thống cố vấn học hiện tại của ĐHBKHN cũng có một số vấn đề như:

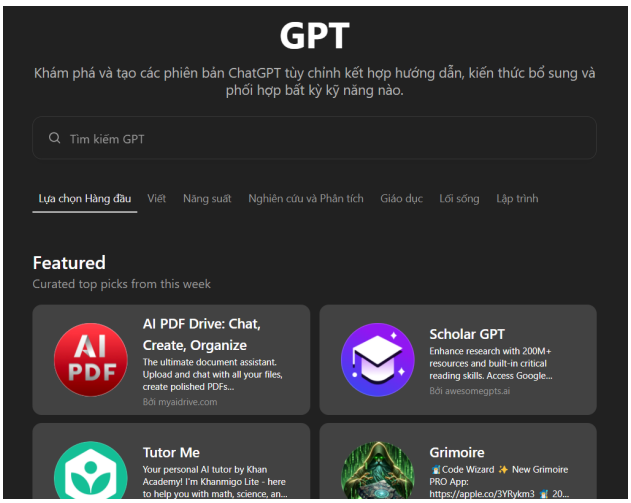
- Trong học kỳ 2024.1, đại học có chỉ tiêu tuyển sinh lên đến 9.260 sinh viên, một con số khá lớn và có thể tăng lên.

  Việc số lương sinh viên tăng lên cũng làm số thắc mắc tăng lên từ khó khăn trong việc giải đáp.

- Việc cố vấn hay cán bộ trả lời thắc mắc có thể gặp phải các nhược điểm như trả lời quá muộn, bỏ sót câu hỏi hay phải liên lạc qua trung gian nhiều bên.

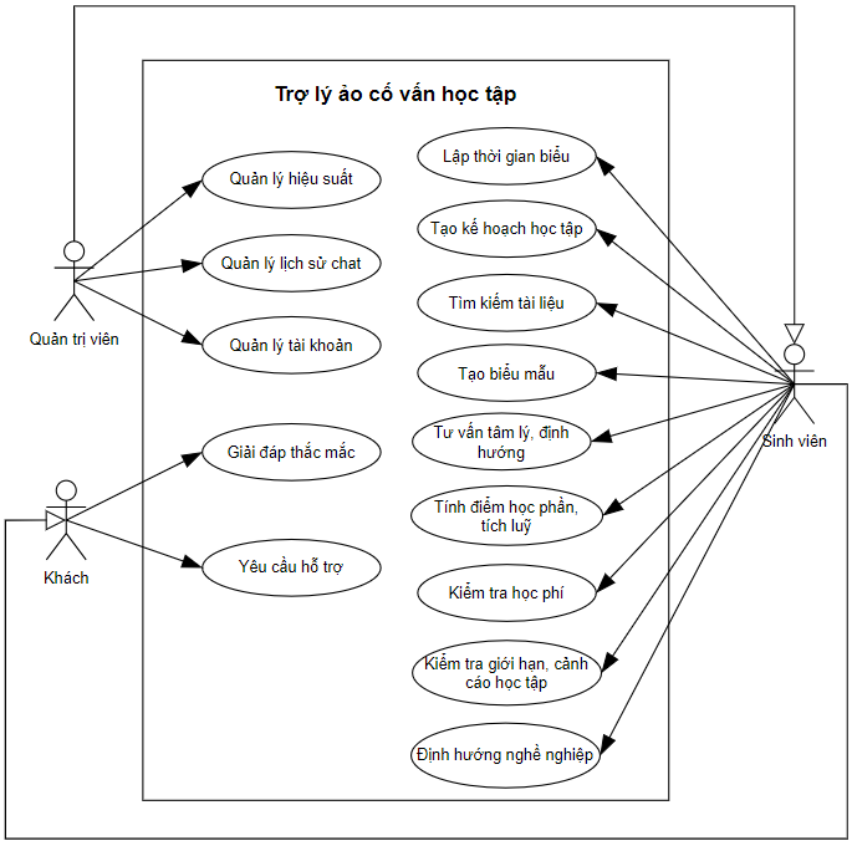
- Mặc dù có nhiều tài liệu với đầy đủ nội dung hỗ trợ nhưng có thể có những sinh viên không thể tìm thấy tài liệu cần thiết hoặc văn bản quá dài khiến sinh viên khó có được thông tin cần.

Từ những khó khăn trên, việc trợ giúp của trợ lý ảo là điều cần thiết. Trợ lý ảo là một ứng dụng phần mềm sử dụng trí tuệ nhân tạo (AI) để thực hiện các tác vụ thay cho người dùng thông qua các lệnh hoặc yêu cầu bằng giọng nói hoặc văn bản. Trợ lý ảo có khả năng nhận diện và xử lý ngôn ngữ tự nhiên, học hỏi từ các tương tác trước đó và cung cấp các phản hồi, giải pháp dựa trên yêu cầu của người dùng. Trợ lý ảo có mặt trong nhiều thiết bị và nền tảng, giúp người dùng tối ưu hóa công việc và cải thiện chất lượng cuộc sống. Các trợ lý ảo có đa dạng tính năng như quản lý thời gian và lịch trình, tìm kiếm thông tin, tích hợp các dịch vụ bên ngoài, trò chuyện, giải đáp,... Trợ lý ảo trở thành một yêu cầu phổ biến trong mỗi doanh nghiệp. Hiện nay trên thế giới cũng đã có các trợ lý ảo phổ biến như Siri, Google Assistant, Amazon Alexa,... Trợ lý ảo được sử dụng cho một chức năng là tự động hóa giao tiếp, trả lời câu hỏi hoặc xử lý yêu cầu của người dùng như một chatbot cũng trở nên phổ biến hơn. Ví dụ như ChatGPT, Gemini,...




**Hình 2.11:** Các trợ lý ảo với tính năng riêng trên ChatGPT

Một trợ lý ảo cho đại học phải đảm bảo nhiều chức năng để hỗ trợ được các thực thể tham gia vào hệ thống. Với khách hay là người muốn tìm hiểu về đại học sẽ được trợ lý ảo giải đáp các thắc mắc, tiếp nhận các yêu cầu hỗ trợ liên quan đến hệ thống. Với đối tượng sinh viên, trợ lý ảo cũng có thể giải đáp các thắc mắc, ngoài ra còn hỗ trợ việc lập thời gian biểu học tập, tạo lộ trình học dựa vào chương trình đào tạo và tình hình học tập của sinh viên, tìm kiếm tài liệu học tập, tính điểm học phần, điểm tích luỹ, tạo biểu mẫu, tư vấn tâm lý cho sinh viên.




**Hình 2.12:** Biểu đồ use case tổng quát

Trong đồ án này, phạm vi của trợ lý ảo là giải đáp được các câu hỏi, thắc mắc của sinh viên và khách. Trợ lý ảo sẽ trả lời được câu hỏi từ khách hoặc sinh viên về thông tin trường, các hướng dẫn dịch vụ trong trường, các quy chế/quy định.

## 2.2 Mô tả dữ liệu

Các tài liệu sử dụng sẽ dưới dạng văn bản. Các văn bản đã thu thập và được sử dụng là các quy định, quy chế của trường và các hướng dẫn sinh viên trong buổi sinh hoạt công dân (SHCD) và trên sổ tay sinh viên.
Các quy định, quy chế sử dụng bao gồm

- Quy chế đào tạo 2023
- Quy chế đào tạo vừa học vừa làm
- Quy định đánh giá quốc phòng - an ninh 2015
- Quy định học bổng Khuyến khích học tập 2023
- Quy định học bổng trao đổi
- Quy định học bổng Trần Đại Nghĩa 2023
- Quy định miễn giảm học phí 2023
- Quy định quản lý câu lạc bộ sinh viên 2023
- Quy định quản lý sinh viên nước ngoài 2023
- Quy định thi Olympic và đổi mới, sáng tạo 2023
- Quy định xét cấp học bổng tài trợ 2024

Các hướng dẫn SHCD và trên sổ tay sinh viên bao gồm

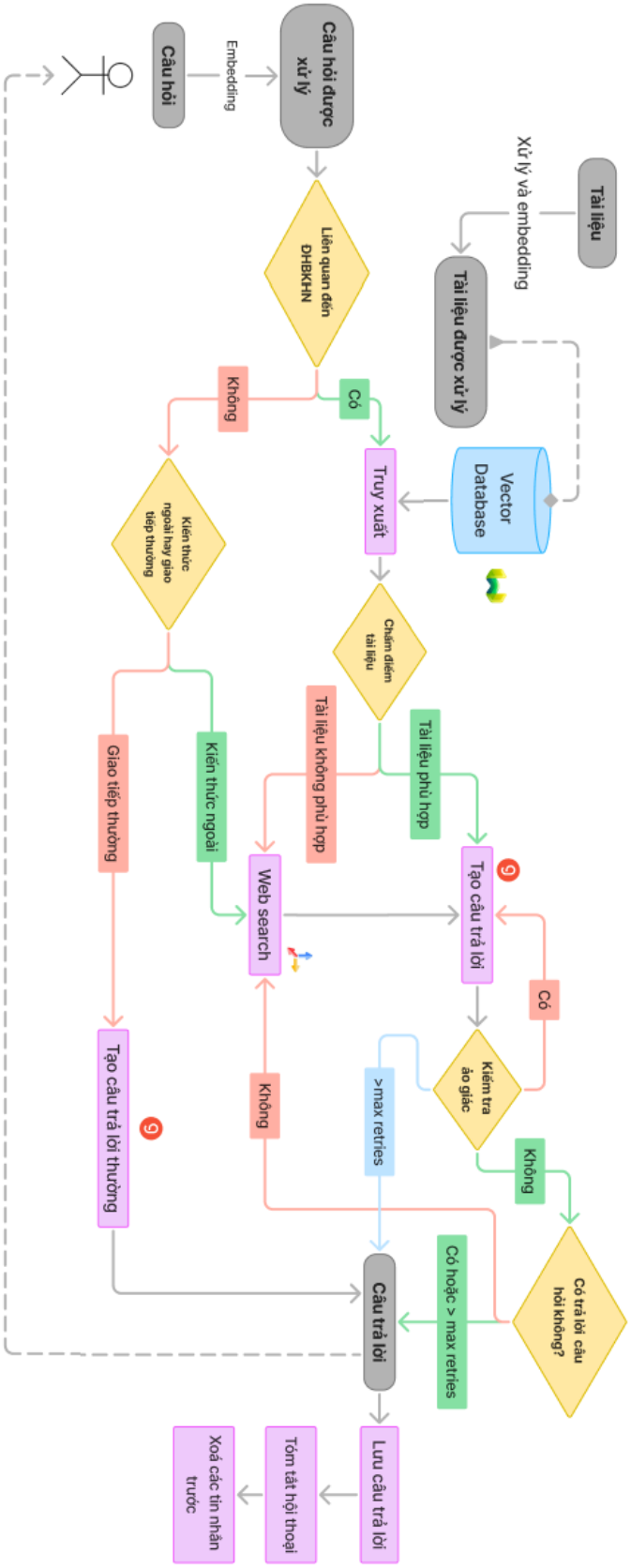
- Định hướng sinh viên
- Giới thiệu công tác điểm rèn luyện
- Giới thiệu thư viện ĐHBK Hà Nội
- Hướng dẫn các nền tảng công nghệ
- Hướng dẫn hồ sơ chế độ chính sách miễn giảm học phí
- Hướng dẫn làm thẻ gửi xe trong trường và làm vé xe buýt tháng
- Hướng dẫn làm thủ tục thanh toán ra trường
- Hướng dẫn nghiệp vụ chương trình tín dụng đối với học sinh, sinh viên
- Hướng dẫn thủ tục chuyển trường dành du học sinh và sinh viên quốc tế

- Kỹ năng viết mail
- Làm thêm cho sinh viên
- Lịch sử Bách khoa
- Những điều sinh viên cần biết
- Phương pháp lập kế hoạch học tập

Với các dữ liệu nêu trên đã chứa phần lớn các thông tin về quy định của đại học. Tuy nhiên các dữ liệu về chương trình đào tạo của các ngành chưa được sử dụng vì giới hạn trong việc phân đoạn dữ liệu dạng bảng, đặc biệt là các bảng dài trong dữ liệu.

Bộ dữ liệu sử dụng để đánh giá trợ lý là bộ câu hỏi với đáp án A, B, C, D. Bộ câu hỏi được lọc ra các câu hỏi quá dài ảnh hưởng đến việc web search và ta có được 140 câu hỏi. Các câu hỏi sẽ liên quan về Trường, về quy chế, quy định và các nội dung liên quan, đảm bảo chất lượng cho việc đánh giá trợ lý ảo được xây dựng.

## 2.3 Giải pháp tổng thể




**Hình 2.13:** Sơ đồ quy trình chi tiết

Quy trình sẽ được diễn tả chi tiết như sau:

**Bước 1.** Các tài liệu sẽ được xử lý, embedding và lưu vào một cơ sở dữ liệu vector.

**Bước 2.** Khi có người dùng đặt câu hỏi, câu hỏi được embedding và được phân loại:

- Nếu câu hỏi liên quan đến ĐHBKHN, sang **Bước 4**.
- Nếu câu hỏi liên quan đến kiến thức bên ngoài, sang **Bước 5**.
- Nếu câu hỏi chỉ là giao tiếp thông thường, sang **Bước 3**.

**Bước 3.** Sử dụng LLM đưa ra câu trả lời bình thường và sang **Bước 8**.

**Bước 4.** Truy xuất tài liệu dựa trên câu hỏi người dùng và kiểm tra tài liệu:

- Nếu tất cả tài liệu liên quan đến câu hỏi, sang **Bước 6**.
- Nếu có tài liệu không liên quan, sang **Bước 5**.

**Bước 5.** Thực hiện tìm kiếm trên web, thêm thông tin tìm được vào tài liệu truy xuất, sang **Bước 6**.

**Bước 6.** Tạo câu trả lời bằng LLM dự trên tài liệu truy xuất và kiểm tra ảo giác:

- Nếu xuất hiện ảo giác và chưa vượt số lần thử lại, quay lại **Bước 4**.
- Nếu không xuất hiện ảo giác, sang **Bước 7**.
- Nếu vượt số lần thử lại, sang **Bước 8**.

**Bước 7.** Kiểm tra câu trả lời có trả lời được câu hỏi không:

- Nếu trả lời được hoặc vượt số lần thử lại, sang **Bước 8**.
- Nếu không trả lời được và chưa vượt số lần thử lại sang **Bước 5**.

**Bước 8.** Câu trả lời được gửi lại cho người dùng.

**Bước 9.** Lưu câu trả lời nếu cần thiết, tóm tắt lại hội thoại cho luồng hỏi đáp, xóa các dữ liệu hộ thoại trước.

# Chương 3

# Công nghệ sử dụng và phương pháp

## 3.1 Xử lý dữ liệu

Dữ liệu sau khi thu thập sẽ được chia thành các đoạn nhỏ để có thể chuyển sang vector embedding.

Các tài liệu thu thập được chia làm hai loại là các quy định, quy chế và các hướng dẫn sinh viên. Hai loại văn bản đều có cấu trúc, với quy định, quy chế có được chia thành các điều khoản, các hướng dẫn sinh viên được chia thành các mục lớn A.,B.,... và 1., 2.,..., các mục con 1.1.,1.2,... và 1.1.1, 1.1.2,...

Mỗi đơn vị phân đoạn sẽ bao gồm tên file, tên mục và nội dung của mục đấy. Ví dụ với một điều khoản trong quy định sau

**Điều 1. Phạm vi điều chỉnh và đối tượng áp dụng**

1. Quy chế này quy định về công tác đào tạo đối với các khóa đào tạo theo hình thức chính quy, vừa làm vừa học và cấp văn bằng cử nhân, kỹ sư, thạc sĩ hoặc tiến sĩ của Đại học Bách khoa Hà Nội (sau đây gọi tắt là ĐHBK Hà Nội). Những vấn đề không được đề cập đến trong Quy chế này sẽ được áp dụng theo các quy chế đào tạo do Bộ Giáo dục và Đào tạo (Bộ GDĐT) ban hành [1] [2] [3].

2. Quy chế này áp dụng cho sinh viên đại học, học viên của chương trình thạc sĩ và nghiên cứu sinh (sau đây gọi chung là người học) của ĐHBK Hà Nội.

3. Các đơn vị cấp 2 thuộc ĐHBK Hà Nội được giao nhiệm vụ thực hiện công tác đào tạo được gọi tắt là trường/khoa.

**Hình 3.1:** Điều khoản trong quy chế đào tạo 2023

Sau khi phân đoạn, ta thu được kết quả như sau

Văn bản trong quy chế đào tạo 2023 có nội dung như sau:

Điều 1. Phạm vi điều chỉnh và đối tượng áp dụng

1.Quy chế này quy định về công tác đào tạo đối với các khóa đào tạo theo hình thức chính quy, vừa làm vừa học và cấp văn bằng cử nhân, kỹ sư, thạc sĩ hoặc tiến sĩ của Đại học Bách khoa Hà Nội (sau đây gọi tắt là ĐHBK Hà Nội).

Văn bản trong quy chế đào tạo 2023 có nội dung như sau:

Điều 1. Phạm vi điều chỉnh và đối tượng áp dụng

Những vấn đề không được đề cập đến trong Quy chế này sẽ được áp dụng theo các quy chế đào tạo do Bộ Giáo dục và Đào tạo (Bộ GDĐT) ban hành 1 2 3.2.

Văn bản trong quy chế đào tạo 2023 có nội dung như sau:

Điều 1. Phạm vi điều chỉnh và đối tượng áp dụng

Quy chế này áp dụng cho sinh viên đại học, học viên của chương trình thạc sĩ và nghiên cứu sinh (sau đây gọi chung là người học) của ĐHBK Hà Nội.3.

Văn bản trong quy chế đào tạo 2023 có nội dung như sau:

Điều 1. Phạm vi điều chỉnh và đối tượng áp dụng

Các đơn vị cấp 2 thuộc ĐHBK Hà Nội được giao nhiệm vụ thực hiện công tác đào tạo được gọi tắt là trường/khoa.

**Hình 3.2:** Nội dung phân đoạn điều khoản

Còn với các hướng dẫn trong SHCD là các slide powerpoint nên ta cần lấy nội dung văn bản trong đó và chuyển về định dạng của văn bản. Ví dụ với hướng dẫn thư viện

**A. GIỚI THIỆU CHUNG VÀ NỘI QUY THƯ VIỆN**

**1. Lịch sử hình thành**

• Thư viện thành lập từ năm 1956, cùng năm thành lập trường Đại học Bách Khoa Hà Nội.
• Năm 2006, tòa nhà thư viện khánh thành và đi vào hoạt động, lấy tên là Thư viện Tạ Quang Bửu.

**Tòa nhà thư viện Tạ Quang Bửu**

Tòa nhà gồm 10 tầng:
•**Tầng 1 - 5: Thư viện**
•Tầng 6 - 10: Hội trường, Phòng thí nghiệm và một số Khoa, Viện Phòng ban khác...
***Chú ý**: Không có nhiệm vụ đề nghị bạn đọc không lên các tầng 6-10 để đảm bảo an ninh tòa nhà.*

**2. Cơ cấu tổ chức**

**Ban giám đốc:**

- **Phòng xử lý thông tin**
  **+ Bổ sung**
  **+ Biên mục**
  **+ Kỹ thuật**
  **+ Nghiên cứu phát triển.**
- **Phòng dịch vụ thông tin**
  **+ Đọc tại chỗ**
  **+ Mượn về nhà**
  **+ Đa phương tiện**
  **+ Bảo quản tài liệu.**
- **Phòng thông tin thư mục**
  **+ Văn phòng**
  **+ Tư vấn thông tin**
  **+ Quản trị thông tin số**
  **+ Quản lý thiết bị.**

**Hình 3.3:** Nội dung hướng dẫn thư viện

Ta sẽ phân đoạn thành thành các đơn vị gồm tên file, các mục lớn, mục con như sau

A. GIỚI THIỆU CHUNG VÀ NỘI QUY THƯ VIỆN

Văn bản trong Giới thiệu Thư viện Trường ĐHBK Hà Nội có nội dung như sau:

1. Lịch sử hình thành

• Thư viện thành lập từ năm 1956, cùng năm thành lập trường Đại học Bách Khoa Hà Nội. •Năm 2006, tòa nhà thư viện khánh thành và đi vào hoạt động, lấy tên là Thư viện Tạ Quang Bửu.

Văn bản trong Giới thiệu Thư viện Trường ĐHBK Hà Nội có nội dung như sau:

1. Lịch sử hình thành

Tòa nhà thư viện Tạ Quang Bửu Tòa nhà gồm 10 tầng • Tầng 1 - 5 Thư viện •Tầng 6 - 10 Hội trường, Phòng thí nghiệm và một số Khoa, Viện Phòng ban khác… Chú ý Không có nhiệm vụ đề nghị bạn đọc không lên các tầng 6-10 để đảm bảo an ninh tòa nhà.

Văn bản trong Giới thiệu Thư viện Trường ĐHBK Hà Nội có nội dung như sau:

2. Cơ cấu tổ chức

Ban giám đốc - Phòng xử lý thông tin + Bổ sung + Biên mục + Kỹ thuật + Nghiên cứu phát triển. - Phòng dịch vụ thông tin + Đọc tại chỗ + Mượn về nhà + Đa phương tiện + Bảo quản tài liệu. - Phòng thông tin thư mục + Văn phòng + Tư vấn thông tin + Quản trị thông tin số + Quản lý thiết bị.

**Hình 3.4:** Nội dung phân đoạn các mục

Sau khi phân đoạn các văn bản, ta sẽ chuyển các đoạn thành vector embeding. Các mô hình embedding có rất nhiều và có thể tìm thấy trên Hugging Face. Hugging Face cung cấp nhiều cho cộng đồng mã nguồn mở thông qua việc phát triển các công cụ, thư viện và dịch vụ hỗ trợ. Trên Hugging Face cũng có những công cụ được đóng góp bởi nhiều người dùng. Thông thường, các mô hình embedding được sử dụng cho tiếng Anh. Đối với tiếng Việt cũng có đóng góp từ nhiều người dùng, tổ chức như VinAI, Đặng Văn Tuấn,... Và trong đồ án này, mô hình *vietnamese – embedding* của Đặng Văn Tuấn được sử dụng cho việc embedding. Mô hình này chuyên biệt được đào tạo riêng cho tiếng Việt, tận dụng khả năng của PhoBERT (tạo từ VinAI), mô hình mã hóa các câu tiếng Việt thành vector 768 chiều.

## 3.2 Cơ sở dữ liệu vector

Khi các văn bản được phân đoạn và nhúng thành vector embedding, các vector sẽ thường có không gian lớn. Với mô hình embedding đang sử dụng, các vector embedding có số chiều 768 và với số lượng tài liệu nhiều, ta cần lưu trữ các vector và nội dung tài liệu trong một cơ sở dữ liệu chuyên biệt cho lưu trữ và tìm kiếm theo vector. Việc sử dụng một cơ sở dữ liệu vector sẽ giúp cho việc xử lý, tìm kiếm thông tin nhanh chóng.

Trong đồ án này, cơ sở dữ liệu được sử dụng là Weaviate. Weaviate chủ yếu lưu trữ và truy vấn các vector embedding thay vì các bản ghi hoặc bảng[7]. Nó là một cơ sở dữ liệu tối ưu cho các ứng dụng machine learning và tìm kiếm theo nghĩa (semantic search). Weaviate có thể lưu trữ các đối tượng dữ liệu dưới dạng vector. Mỗi đối tượng (như văn bản, hình ảnh,...) sẽ được chuyển thành một vector và lưu trữ trong cơ sở dữ liệu này. Ngoài vector, Weaviate cũng lưu trữ các metadata liên quan đến các đối tượng đó.

Weaviate còn hỗ trợ Weaviate Cloud là một cơ sở dữ liệu vector được quản lý hoàn toàn trên đám mây. Weaviate Cloud được xây dựng trên lõi Weaviate, cùng một công nghệ và cung cấp cùng các tính năng mà không cần kiến thức

chuyên sâu về DevOps. Ta chỉ cần tạo API key trên Weaviate Cloud là có thể sử dụng các dịch vụ có sẵn.

Dữ liệu trong Weaviate được tìm kiếm bằng từ khóa (Keyword search), tìm kiếm bằng vector (Vector search) hoặc tìm kiếm kết hợp cả hai (Hybrid search).

**Keyword search trong Weaviate**

Weaviate sử dụng mô hình BM25 để đánh giá mức độ liên quan giữa truy vấn và các tài liệu dựa trên tần suất xuất hiện của từ khóa. Phương pháp này tập trung vào việc khớp chính xác các từ hoặc cụm từ trong truy vấn với nội dung trong cơ sở dữ liệu. Hiệu quả trong các trường hợp cần tìm kiếm chính xác dựa trên từ khóa.

**Vector search trong Weaviate**

Vector search trong Weaviate chủ yếu dựa vào các thuật toán tìm kiếm gần nhất (Nearest neighbor search) và đo lường độ tương đồng giữa các vector. Các thuật toán chính bao gồm:

- **Tìm kiếm gần nhất xấp xỉ (ANN)**: Weaviate sử dụng thuật toán Hierarchical Navigable Small World (HNSW) để thực hiện vector search. HNSW tổ chức các vector trong một cấu trúc đồ thị phân cấp nhiều lớp, cho phép điều hướng nhanh qua tập dữ liệu trong quá trình tìm kiếm. Cấu trúc này cân bằng giữa việc tìm kiếm nhanh ở các lớp trên với khoảng cách dài hơn và tìm kiếm chính xác ở các lớp dưới với khoảng cách ngắn hơn.

- **Đo lường độ tương đồng:** Weaviate hỗ trợ các phép đo khoảng cách khác nhau như Cosine Similarity, Dot Product, L2-Squared và Manhattan Distance để tính toán độ tương đồng giữa các vector, tùy thuộc vào yêu cầu cụ thể của ứng dụng.

Weaviate chủ yếu sử dụng các thuật toán k-NN HNSW kết hợp với đo lường độ tương đồng để tìm kiếm vector gần nhất, từ đó tìm ra các đối tượng dữ liệu có ý nghĩa tương tự với truy vấn.

**Hybrid search trong Weaviate**

Hybrid search là sự kết hợp của keyword search và vector search. Việc kết hợp nhằm tận dụng cả khả năng khớp chính xác từ khóa và hiểu ngữ nghĩa của văn bản. Kết quả đánh giá hiệu quả tìm kiếm Weaviate hỗ trợ hai phương pháp kết hợp kết quả:

- *relativeScoreFusion*: Chuẩn hóa điểm số từ cả hai tìm kiếm và tính tổng có trọng số để xếp hạng kết quả cuối cùng.

- *rankedFusion*: Xếp hạng các kết quả dựa trên vị trí của chúng trong từng tìm kiếm và kết hợp thứ hạng này để tạo ra xếp hạng tổng hợp.

Mục đích sử dụng Hybrid search là khi ta cần tận dụng cả khả năng hiểu ngữ nghĩa của tìm kiếm vector và khả năng khớp chính xác của tìm kiếm từ khóa, việc tìm kiếm vector hoặc từ khóa riêng lẻ không mang lại kết quả mong muốn. Từ sự mạnh mẽ của hybrid search, đây là phương pháp được sử dụng cho đồ án này.

## 3.3 Web search

Tavily là một công cụ tìm kiếm tiên tiến được thiết kế đặc biệt cho các mô hình ngôn ngữ lớn và các ứng dụng tạo nội dung dựa trên RAG. Nó cung cấp API tìm kiếm, kết nối các agent với các nguồn kiến thức đáng tin cậy và theo thời gian thực, giúp họ cung cấp kết quả chính xác và thực tế một cách nhanh chóng.

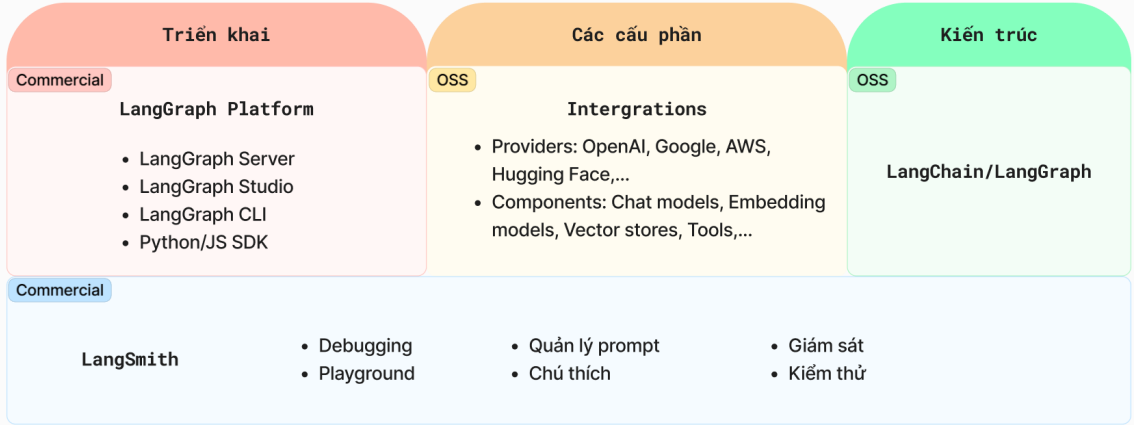
Tavily cung cấp 3 cách đơn giản để tìm kiếm:

- *search*: Thực hiện tìm kiếm và Tavily trả về phản hồi dưới dạng dict có cấu trúc tốt.
- *get_search_context*: thực hiện truy vấn tìm kiếm và Tavily trả về chuỗi nội dung và nguồn trong giới hạn mã thông báo được cung cấp. Nó hữu ích để lấy nội dung liên quan từ các trang web đã truy xuất mà không phải xử lý việc trích xuất ngữ cảnh và quản lý mã thông báo.
- *qna_search*: thực hiện tìm kiếm và trả về chuỗi chứa câu trả lời cho truy vấn gốc. Điều này tối ưu để được sử dụng như một công cụ cho các agent.

Tavily là một công cụ quan trọng cho các nhà phát triển xây dựng nên các LLM và hệ thống RAG mạnh mẽ và đáng tin cậy hơn. Tavily giúp giảm thiểu các thách thức của LLM như ảo giác và kiến thức bị lỗi thời, giúp các ứng dụng AI đáng tin cậy và sâu sắc hơn.

## 3.4 LangChain

LangChain là framework mã nguồn mở dùng để phát triển ứng dụng sử dụng mô hình ngôn ngữ lớn[8]. Framework cung cấp các công cụ và thư viện giúp dễ dàng tích hợp LLM vào ứng dụng của mình và đơn giản hoá các bước xây dựng ứng dụng như phát triển, sản phẩm hoá, triển khai.




**Hình 3.5:** Hệ sinh thái của LangChain

Rất nhiều tính năng mã nguồn mở được LangChain hỗ trợ, tích hợp sãn như các mô hình chat, bộ truy xuất (retriver), các công cụ hỗ trợ (như web search, biên dịch mã), mô hình embedding, lưu trữ lịch sử chat,...

Về mặt triển khai, LangChain đưa ra LangGraph platform là giải pháp thương mại để triển khai các ứng dụng agent trong sản xuất, được xây dựng trên nền tảng LangGraph nguồn mở. LangGraph platform bao gồm một số thành phần hoạt động cùng nhau để hỗ trợ triển khai và quản lý các ứng dụng LangGraph:

- LangGraph Server: Được thiết kế để hỗ trợ nhiều trường hợp sử dụng ứng dụng agentic, từ xử lý nền đến tương tác thời gian thực.
- LangGraph Studio: Là một IDE chuyên dụng có thể kết nối với LangGraph Server để cho phép trực quan hóa, tương tác và gỡ lỗi ứng dụng cục bộ.
- LangGraph CLI: Là giao diện dòng lệnh giúp tương tác với LangGraph cục bộ
- Python/JS SDK: Cung cấp một cách lập trình để tương tác với các ứng dụng LangGraph đã triển khai.

Và để xây dựng trợ lý ảo, LangGraph và LangSmith là hai thành phần của LangChain được sử dụng trong đồ án này.

### 3.4.1 LangGraph

Langgraph là một thư viện chuyên biệt trong hệ sinh thái LangChain sử dụng để xây dựng và quản lý các hệ thống agent phức tạp với LLM. Nó có khả năng cách cung cấp kiểm soát chi tiết đối với quy trình làm việc của agent, từ việc xây dựng các chuỗi đơn giản đến thiết kế các cấu trúc nhiều agent phức tạp hơn.

Các tác vụ lớn trong quy trình làm việc của agent được chia thành các tác vụ con và được gọi là nút của đồ thị. Và sự chuyển tiếp giữa các tác vụ được gọi là cạnh của đồ thị. Một khái niệm cơ bản nữa trong Langgraph là State (trạng thái), thành phần này sử dụng để duy trì các tác vụ con. Tất cả các nút có thể truy cập và sửa đổi State cho phép tương tác trạng thái theo ngữ cảnh.

Đặc điểm nổi bật:

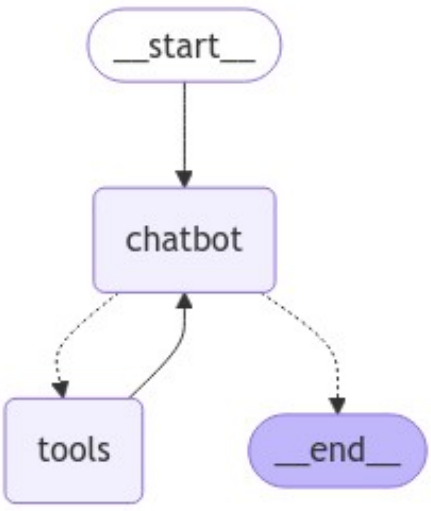
- Kiến trúc nhận thức (Cognitive architectures): LangGraph hỗ trợ việc xây dựng các "state machines" - kiến trúc nhận thức có dạng đồ thị, nơi các trạng thái đại diện cho các agent độc lập. Các kết nối (cạnh) giữa các trạng thái xác định luồng công việc, cho phép thêm các nhánh điều kiện và vòng lặp.
- Thiết kế theo mô-đun:
  - StateGraph: Là nền tảng cốt lõi để định nghĩa trạng thái và quản lý cách trạng thái thay đổi theo thời gian qua các node.
  - Nút (Node): Từng nút trong đồ thị đại diện cho một tác vụ cụ thể hoặc một agent, với khả năng sử dụng công cụ, gọi mô hình ngôn ngữ, hoặc thực hiện logic tùy chỉnh.

  - Cạnh điều kiện và cạnh vòng lặp (Conditional and looping edges): Hỗ trợ điều hướng phức tạp, như quyết định bước tiếp theo dựa trên đầu ra hoặc điều kiện cụ thể.

- Quy trình làm việc Multi-Agent: LangGraph cho phép kết nối nhiều agent độc lập để làm việc trên cùng một nhiệm vụ hoặc hợp tác theo các mô hình khác nhau như:
  - Cho phép chia sẻ dữ liệu giữa các agent.
  - Agent chính sẽ giám sát và điều phối các agent con.
  - Cấu trúc phân cấp cho phép các nhóm agent lồng ghép với nhau để xử lý nhiệm vụ lớn hơn.
- LangGraph Cloud: Đây là nền tảng triển khai LangGraph trên quy mô lớn với các tính năng như quản lý trạng thái lâu dài, xử lý nhiều người dùng đồng thời, và tích hợp kiểm tra chất lượng. Các tính năng khác như double-texting (quản lý đầu vào mới trong luồng công việc đang chạy) và hỗ trợ tương tác giữa con người và agent cũng được tích hợp.

LangGraph hữu ích cho các trường hợp cần sự phối hợp của nhiều agent, như tạo báo cáo, lập trình tự động, và quản lý dữ liệu lớn. Làm việc với LangGraph ta có thể kiểm soát được luồng thực thi của agent qua các đồ thị từ đơn giản đến phức tạp. Một đồ thị có thể trực quan hoá như hình dưới đây.

Chương trình sẽ thực thi theo một luồng từ start cho đến end. Trong ví dụ về một đồ thị đơn giản trên có các nút là chatbot để một LLM sinh ra câu trả lời và nút tools để gọi tool hỗ trợ, có thể là truy xuất dữ liệu, tính toán số học, gửi email, lên lịch,...




**Hình 3.6:** Một đồ thị đơn giản trên LangGraph

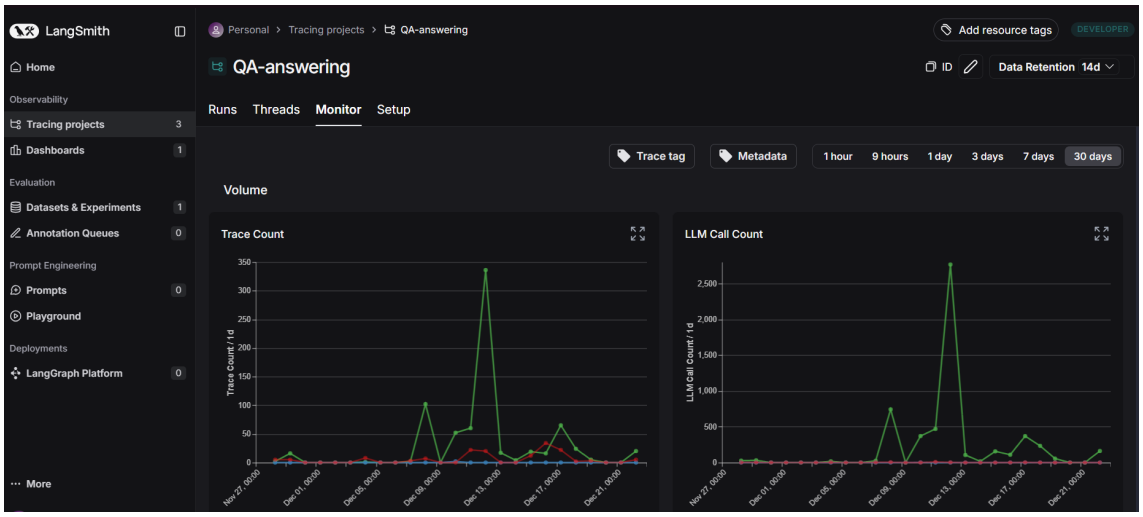
Khi chạy chương trình, ta sẽ chạy từ start đến chatbot. Sau đó, chương trình sẽ đi qua cạnh có điều kiện, nút tools được gọi tiếp theo nếu thoả mãn điều kiện nào đó và sẽ trả lại kết quả cho chatbot hoặc có thể đi tiếp sang nút end để kết thúc một lần chạy. Kết quả trả về là câu trả lời của LLM.

### 3.4.2 LangSmith

LangSmith là công cụ được phát triển nhằm hỗ trợ các nhà phát triển trong việc giám sát, kiểm thử, và cải thiện các ứng dụng sử dụng mô hình ngôn ngữ lớn. Đây là một dịch vụ quản lý và phân tích chuyên sâu cho các ứng dụng AI, nhất là các ứng dụng xây dựng trên nền tảng LangChain.

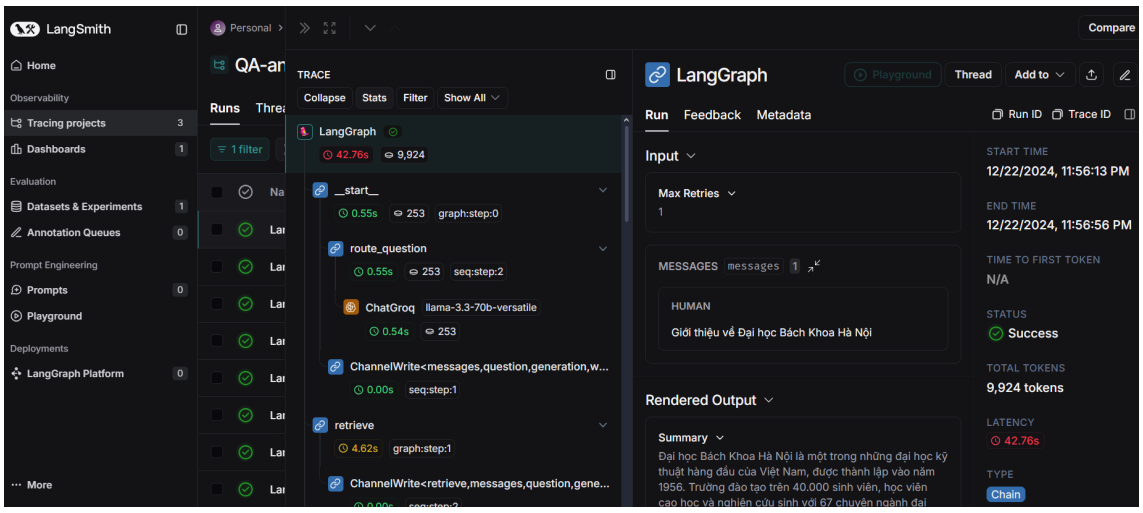
Chức năng chính của LangSmith

- Giám sát ứng dụng, theo dõi chi tiết mọi tương tác giữa người dùng và hệ thống, bao gồm input và output của mô hình ngôn ngữ lớn. Ghi lại thông tin về mức sử dụng API, thời gian phản hồi, và hiệu suất của từng thành phần trong ứng dụng.




**Hình 3.7:** Kiểm soát lượng sử dụng

- Hỗ trợ nhà phát triển kiểm tra tính chính xác và hiệu quả của ứng dụng. Cung cấp công cụ để chạy thử các agent, giúp phát hiện lỗi hoặc điểm cần cải thiện.




**Hình 3.8:** Giao diện kiểm thử trên LangSmith

- Ghi nhận và phân tích hiệu quả của các prompts. Cung cấp công cụ để thử nghiệm các biến thể khác nhau, từ đó tối ưu hóa kết quả đầu ra của mô hình.
- Lưu trữ và tổ chức các tập dữ liệu (datasets) phục vụ cho việc huấn luyện, kiểm tra, và phân tích. Hỗ trợ truy cập và sử dụng lại dữ liệu một cách dễ dàng trong các lần thử nghiệm hoặc triển khai tiếp theo.
- Cho phép các thành viên trong nhóm làm việc cùng nhau trên cùng một dự án.

LangSmith giúp cải thiện trải nghiệm người dùng, theo dõi và phân tích hiệu suất. Khi ta theo dõi được mức độ sử dụng tài nguyên (API, mô hình) trên LangSmith, khi đó ta có thể tối ưu hóa chi phí vận hành. Nhà phát triển còn có thể xây dựng và cải tiến ứng dụng nhanh hơn với các công cụ kiểm thử và phân tích tự động.

LangSmith có thể tích hợp với LangChain, ta có thể dễ dàng chuyển từ việc thiết kế ứng dụng trong LangChain sang kiểm thử và giám sát trong LangSmith. Tạo ra một quy trình làm việc từ phát triển đến triển khai và cải tiến.

## 3.5 Mô hình Chat

### 3.5.1 ChatGroq

Groq là một nền tảng phần cứng và phần mềm tập trung vào việc cung cấp tốc độ tính toán cao, chất lượng và hiệu quả năng lượng cho các ứng dụng AI. ChatGroq là một tích hợp của mô hình ngôn ngữ lớn từ Groq, được thiết kế để cung cấp khả năng suy luận AI nhanh chóng và hiệu quả.

ChatGroq được tích hợp trong thư viện LangChain, cho phép các nhà phát triển dễ dàng sử dụng API của Groq trong ứng dụng. ChatGroq là mô hình hỗ trợ các tính năng như tool calling, JSON mode (trả về câu trả lời dạng JSON),... ChatGroq còn hỗ trợ tuỳ chỉnh các LLM, một số mô hình được hỗ trợ API như Llama, Mixtral, Gemma với giới hạn lượng sử dụng.

### 3.5.2 Llama

Llama (Large Language Model Meta AI) là một bộ mô hình ngôn ngữ lớn được phát triển bởi Meta (trước đây là Facebook) và công bố vào năm 2023. Mô hình này được thiết kế để cung cấp khả năng hiểu và sinh ngôn ngữ tự nhiên với hiệu suất cao, đồng thời tối ưu hóa việc sử dụng tài nguyên tính toán, giúp dễ dàng triển khai trong nhiều ứng dụng khác nhau. Llama được phát triển nhằm cung cấp một mô hình ngôn ngữ có khả năng nghiên cứu cao mà không yêu cầu quá nhiều tài nguyên tính toán như các mô hình lớn khác. Điều này giúp giảm chi phí và tài nguyên cần thiết để huấn luyện và triển khai mô hình, đồng thời giúp Llama tiếp cận dễ dàng với nhiều tổ chức nghiên cứu và doanh nghiệp.

Llama đã trải qua nhiều phiên bản khác nhau. Hiện tại phiên bản Llama 3.3, được Meta phát hành vào ngày 6 tháng 12 năm 2024, là phiên bản mới nhất trong các mô hình ngôn ngữ lớn của Meta, mang đến nhiều cải tiến đáng kể so với các phiên bản trước đó.

Llama 3.3 có những cải tiến về lý luận, hiểu biết toán học, kiến thức chung và làm theo hướng dẫn. Ta có thể sử dụng Llama 3.3 cho các ứng dụng doanh nghiệp, sáng tạo nội dung và các nghiên cứu nâng cao. Mô hình này hỗ trợ nhiều ngôn ngữ và vượt trội hơn nhiều mô hình LLM đã có. Llama 3.3 với kích thước 70B cung cấp hiệu suất tương đương với LLaMA 3.1 405B[9], nhưng với kích thước nhỏ hơn đáng kể, giúp giảm chi phí và tài nguyên tính toán cần thiết. Llama 3.3 còn là mô hình hỗ trợ đa ngôn ngữ, được cải thiện khả năng theo dõi hướng dẫn, phù hợp với việc ứng dụng cho chatbot và trợ lý ảo.

## 3.6 Xây dựng phần mềm

### 3.6.1 React

Để xây dựng phần frontend cho trợ lý ảo để người dùng có thể truy cập và gửi câu hỏi, ta cso thể sử dụng React. React, một thư viện JavaScript do Facebook phát triển, giúp xây dựng giao diện người dùng. Thư viện này giúp phát triển các thành phần giao diện người dùng có thể tái sử dụng. Thay vì viết lại mã cho mỗi phần của UI, React cho phép bạn chia nhỏ giao diện thành các phần nhỏ hơn, gọi là "components", mỗi component này có thể độc lập và dễ dàng tái sử dụng. React được sử dụng rộng rãi trong phát triển frontend vì khả năng tối ưu hóa hiệu suất và khả năng mở rộng dễ dàng.

### 3.6.2 FastAPI

FastAPI là một framework web hiện đại, nhanh chóng và dễ sử dụng để phát triển các API trong Python. FastAPI được thiết kế để giúp lập trình viên phát triển các ứng dụng web RESTful APIs một cách nhanh chóng, hiệu quả, và an toàn.

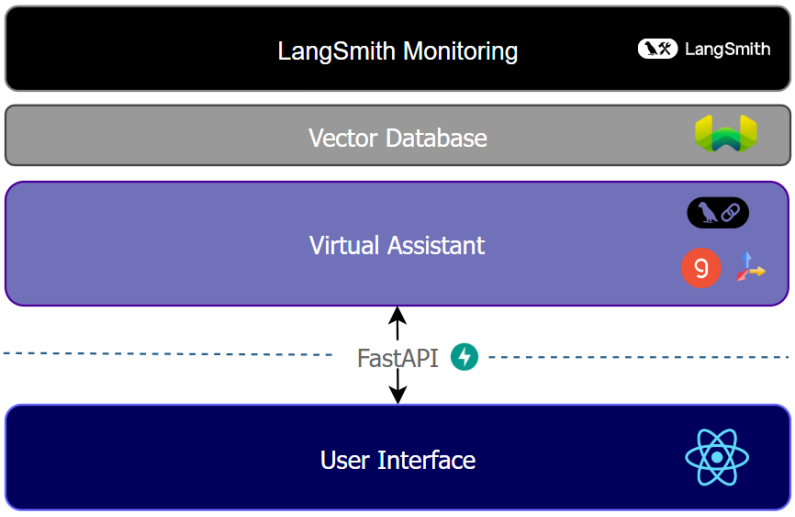
Các đặc điểm của FastAPI bao gồm tốc độ, hiệu suất cao, dễ học và sử dụng, tương thích với các tiêu chuẩn web. Việc sử dụng FastAPI để kết nối trợ lý ảo với giao diện người dùng.

# Chương 4

# Cài đặt chương trình và kết quả

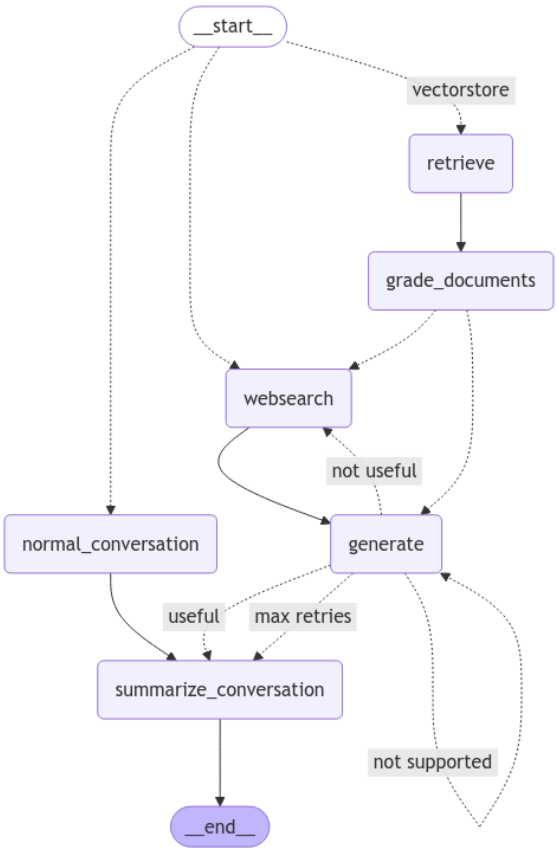
## 4.1 Cách thức cài đặt

Kiến trúc hệ thống trợ lý ảo trợ lý bao gồm LangSmith để sử dụng cho việc kiểm thử, quản lý hiệu suất, mức độ sử dụng tài nguyên của hệ thống, cơ sở dữ liệu vector Weaviate để lưu trữ các vector embedding và các metadata của tài liệu, trợ lý ảo kết nối với giao diện người dùng thông qua API để trao đổi thông tin về câu hỏi và câu trả lời.




**Hình 4.1:** Kiến trúc hệ thống trợ lý ảo

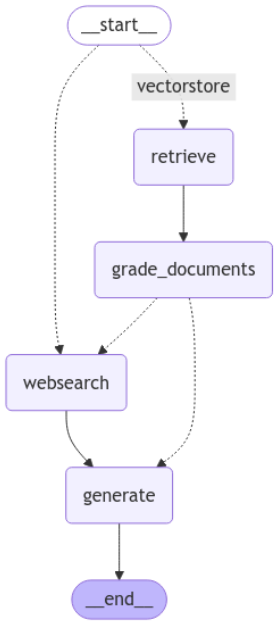
Dựa trên quy trình tổng thể được trình bày ở trên, ta chỉ cần xây dựng các nút của đồ thị và gắn chúng với nhau để trở thành đồ thị ở hình 4.2. Các nút trong đồ thị được hoạt động bằng các prompt chỉ dẫn cho LLM đưa ra quyết định theo nhánh nào.




**Hình 4.2:** Đồ thị được xây dựng cho trợ lý ảo

## 4.2 Đánh giá mô hình

Để đánh giá tính chính xác của trợ lý ảo, ta sẽ kiểm tra trên bộ Q&A kiểm tra quy chế. Trợ lý ảo sẽ đưa ra đáp án cho các câu hỏi hoặc trả lời không biết khi không có hoặc không đủ thông tin để đưa ra đáp án.




**Hình 4.3:** Đồ thị được xây dựng cho việc đánh giá

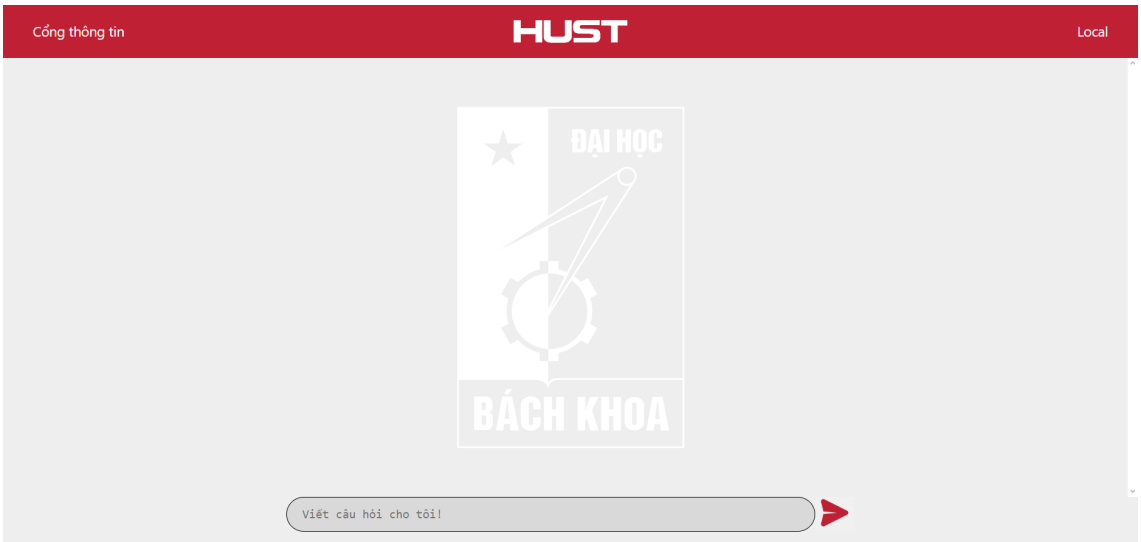
Luồng chạy cho việc trả lời các câu hỏi quy chế là một luồng khác vì câu trả lời chỉ bao gồm các đáp án A, B, C, D và "không biết". Việc kiểm tra ảo giác và lưu trữ hội thoại được lược bớt và sẽ chỉ còn việc truy xuất thông tin từ văn bản hoặc từ web search và đưa ra câu trả lời.

| | **Số câu** | **Phần trăm** |
|---|---|---|
| **Đúng** | 104 | 74% |
| **Không biết** | 21 | 15% |
| **Sai** | 15 | 11% |

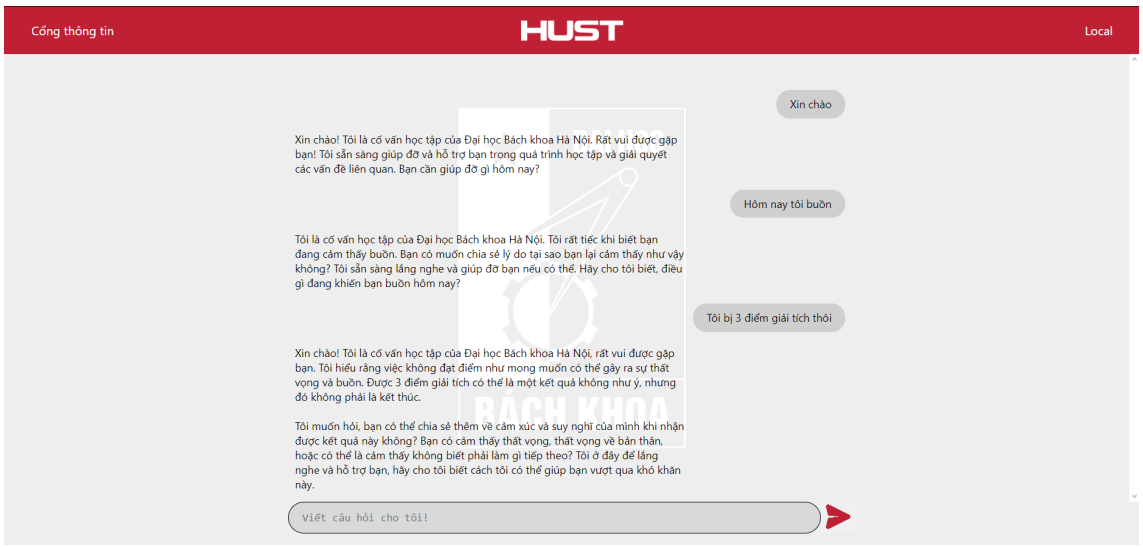
**Bảng 4.1:** Kết quả đánh giá

Thông qua kết quả trên, cho ta thấy trợ lý ảo trả lời bộ câu hỏi khá tốt khi chỉ sai 11% số câu hỏi và biết đưa ra câu trả lời không biết khi không có hoặc không đủ thông tin. Tuy nhiên, trợ lý ảo còn có thể chính xác hơn nữa nếu ta cải thiện được chiến lược phân đoạn, chiến lược tìm kiếm và bổ sung được các tài liệu dạng bảng mà mô hình còn thiếu.

## 4.3 Giao diện




**Hình 4.4:** Giao diện chính




**Hình 4.5:** Giao diện khi trò chuyện với trợ lý ảo

# Kết luận và hướng phát triển

## Kết luận

Qua quá trình tìm hiểu và xây dựng trợ lý ảo cố vấn học tập, đồ án đã thực hiện các công việc sau:

- Tìm hiểu về mô hình ngôn ngữ lớn và cách hoạt động, phương pháp RAG và cách cải thiện độ chính xác cho nó.
- Khảo sát và phân tích hệ thống tài liệu và hỗ trợ của ĐHBKHN.
- Cài đặt phân đoạn cho các tài liệu để phục vụ cho việc truy xuất.
- Áp dụng thư viện mạnh mẽ của hệ sinh thái LangChain để xây dựng trợ lý ảo để trả lời các câu hỏi liên quan đến ĐHBKHN.
- Cài đặt giao diện người dùng và kết nối được với trợ lý ảo để hỏi đáp.

Kết quả thu được sau thực hiện xây dựng trợ lý ảo ta có được đánh giá khá tốt về trợ lý ảo với tỷ lệ trả lời đúng khá cao và biết kiến thức nào không có trong bộ tài liệu.

## Hướng phát triển trong tương lai

- Cải thiện khả năng phân đoạn dữ liệu, nhất là dữ liệu dạng bảng để mô hình sử dụng được với nhiều loại văn bản hơn.
- Tích hợp thêm các chức năng liên quan đến chương trình đào tạo của các ngành, chức năng tạo thời gian biểu, kế hoạch học tập.
- Khai thác nhiều hơn các tính năng của thư viện LangGraph.
- Triển khai được hệ thống lên môi trường cloud để trợ lý ảo luôn hoạt động.

# Chỉ mục

adaptive RAG, 15
agent, 33

corrective RAG, 16
cơ sở dữ liệu vector, 31

decoder, 12

encoder, 12

framework, 33

hybrid search, 32
học máy, 11
học sâu, 11

keyword search, 32

LangChain, 33
LangGraph, 34
LangSmith, 36

mô hình ngôn ngữ lớn, 11

RAG, 14

Self-Attention, 12
Self-RAG, 15
semantic search, 31

Transformer, 11
trí tuệ nhân tạo, 10
trợ lý ảo, 23

vector embedding, 13

web search, 33

xử lý ngôn ngữ tự nhiên, 10

ảo giác, 14

# Tài liệu tham khảo


[1] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[2] J. Yang, H. Jin, R. Tang, *et al.*, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," 2023. arXiv: 2304.13712 [cs.CL].

[3] P. Lewis, E. Perez, A. Piktus, *et al.*, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021. arXiv: 2005.11401 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.11401.

[4] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, *Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity*, 2024. arXiv: 2403.14403 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2403.14403.

[5] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, *Self-rag: Learning to retrieve, generate, and critique through self-reflection*, 2023. arXiv: 2310.11511 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.11511.

[6] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, *Corrective retrieval augmented generation*, 2024. arXiv: 2401.15884 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2401.15884.

[7] *Weaviate documentation*, 2025. [Online]. Available: https://weaviate.io/developers/weaviate.

[8] *Langchain introduction*, 2025. [Online]. Available: https://python.langchain.com/docs/introduction/.

[9] MetaAI, *Model cards prompt formats llama 3.3*, 2025. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/.