

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------------------------

HỌ TÊN SINH VIÊN: NGUYỄN VĂN ĐỨC

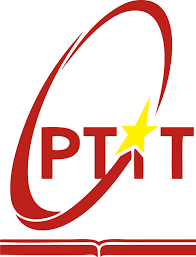
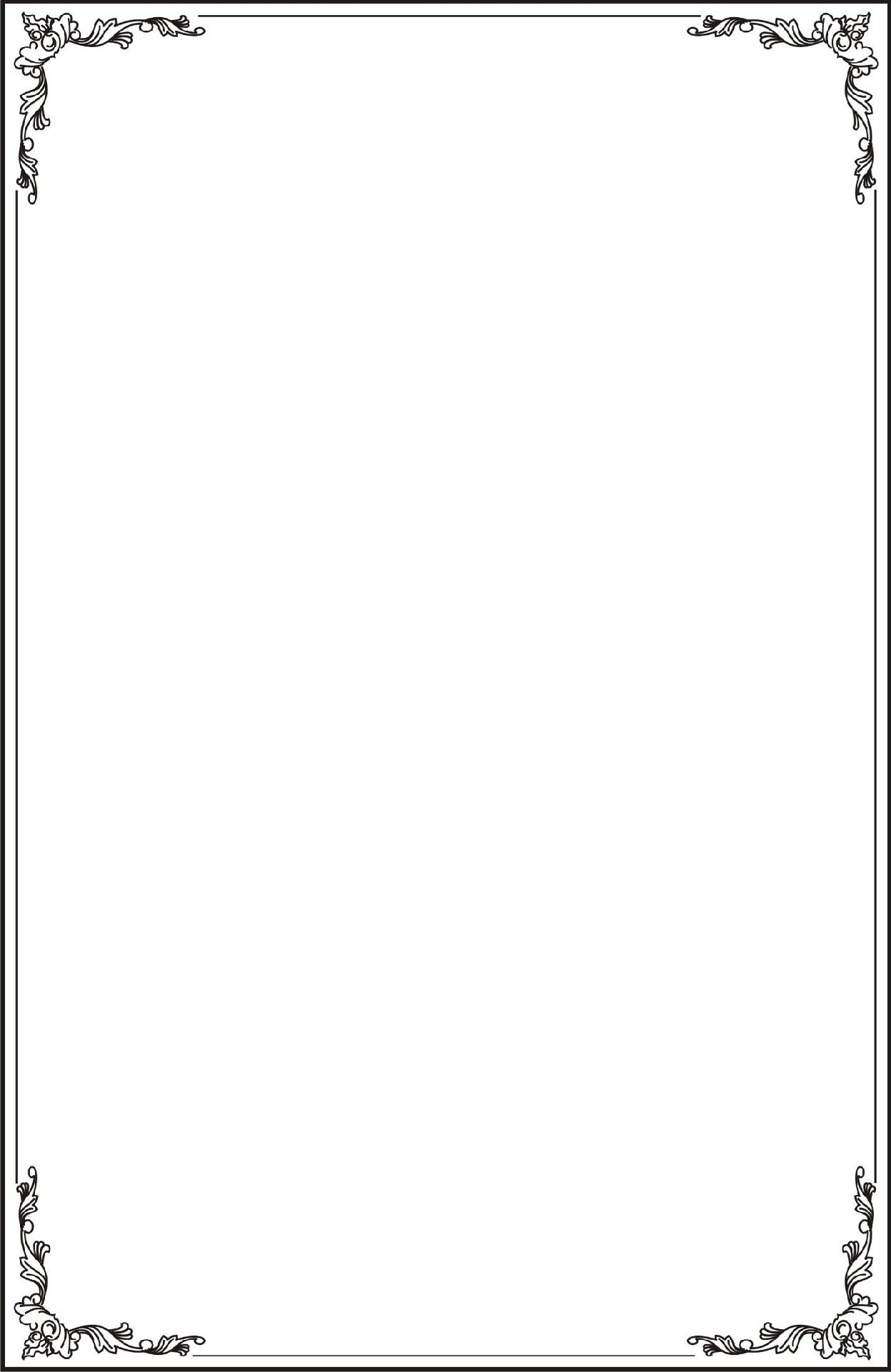
ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG ỨNG DỤNG CHATBOT PHỤC VỤ TUYỂN SINH TẠI HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

LỚP: D20CNPM04

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Vũ Hoài Thư |
| **Sinh viên thực hiện:** | Nguyễn Văn Đức |
| **Mã sinh viên:** | B20DCCN199 |
| **Lớp:** | D20CNPM04 |
| **Niên khóa:** | 2020-2025 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2024**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------------------------

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG ỨNG DỤNG CHATBOT PHỤC VỤ TUYỂN SINH TẠI HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Vũ Hoài Thư |
| **Sinh viên:** | Nguyễn Văn Đức |
| **Mã sinh viên:** | B20DCCN199 |
| **Lớp:** | D20CNPM04 |
| **Niên khóa:** | 2020-2025 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2024**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………

**Điểm: (Bằng chữ: )**

Hà Nội, ngày tháng năm 20…

**Giảng viên hướng dẫn**

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………

**Điểm: (Bằng chữ: )**

Hà Nội, ngày tháng năm 20…

**Giảng viên phản biện**

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô của Học Viện Công Nghệ Bưu Chính Viễn Thông nói chung và các thầy cô trong Bộ môn Công nghệ phần mềm cũng như khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy và truyền đạt những kiến thức chuyên môn nền tảng để làm cơ sở lý luận cho đồ án này.

Em xin bày tỏ lòng biết ơn sâu sắc đến cô ThS.Vũ Hoài Thư đã dìu dắt và hướng dẫn em trong suốt quá trình làm đồ án. Nhờ sự chỉ bảo và định hướng của cô, em đã tự tin hơn trong việc nghiên cứu vấn đề mới và tìm ra giải pháp tối ưu.

Em xin chân thành cảm ơn Ban Giám hiệu Học viện Công Nghệ Bưu Chính Viễn Thông đã tạo điều kiện thuận lợi để em được học tập và làm đồ án một cách thuận lợi.

Mặc dù đã cố gắng rất nhiều, nhưng chắc chắn trong quá trình học tập cũng như làm đồ án không khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và đóng góp của quý thầy cô và các bạn để đồ án hoàn thiện hơn nữa.

Em xin chân thành cảm ơn!

**Hà Nội, tháng 12 năm 2024**

**Sinh viên**

**Nguyễn Văn Đức**

**MỤC LỤC**

[LỜI MỞ ĐẦU 7](#_Toc183444249)

[CHƯƠNG 1: TỔNG QUAN CÁC HỆ THỐNG CHATBOT 9](#_Toc183444250)

[1.1. Khái niệm chatbot 9](#_Toc183444251)

[1.2. Phân loại chatbot 10](#_Toc183444252)

[1.2.1. Rule-based chatbot 10](#_Toc183444253)

[1.2.2. AI-based chatbot 11](#_Toc183444254)

[1.3. Các nền tảng xây dựng chatbot 13](#_Toc183444255)

[1.4. Các thành phần của chatbot mà đồ án nghiên cứu 15](#_Toc183444256)

[1.5. Một số ứng dụng của chatbot 17](#_Toc183444257)

[1.5.1. Dịch vụ khách hàng 17](#_Toc183444258)

[1.5.2. Marketing và quảng cáo 17](#_Toc183444259)

[1.5.3. Tài chính và ngân hàng 18](#_Toc183444260)

[1.5.4. Giáo dục 18](#_Toc183444261)

[1.5.5. Chăm sóc sức khỏe 18](#_Toc183444262)

[1.6. Kết luận chương 18](#_Toc183444263)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 19](#_Toc183444264)

[2.1. Giới thiệu về học sâu 19](#_Toc183444265)

[2.2. Mạng nơ-ron hồi quy 23](#_Toc183444266)

[2.3. Các kỹ thuật word embedding 26](#_Toc183444267)

[2.3.1. TF-IDF 26](#_Toc183444268)

[2.3.2. Word2Vec 27](#_Toc183444269)

[2.3.3. PhoBert 29](#_Toc183444270)

[2.4. Các phương pháp tính độ tương đồng 34](#_Toc183444271)

[2.4.1. Độ tương đồng jaro 34](#_Toc183444272)

[2.4.2. Khoảng cách hamming 35](#_Toc183444273)

[2.4.3. Khoảng cách euclid 36](#_Toc183444274)

[2.4.4. Độ tương đồng jaccard 37](#_Toc183444275)

[2.4.5. Độ tương đồng cosine 39](#_Toc183444276)

[2.5. Xếp hạng lại 41](#_Toc183444277)

[2.6. Kết luận chương 43](#_Toc183444278)

[CHƯƠNG 3: XÂY DỰNG CHATBOT TƯ VẤN TUYỂN SINH 44](#_Toc183444279)

[3.1. Phát biểu bài toán 44](#_Toc183444280)

[3.2. Xây dựng dữ liệu cho chatbot 44](#_Toc183444281)

[3.3. Thực nghiệm và kết quả 47](#_Toc183444282)

[3.4. Cài đặt ứng dụng 47](#_Toc183444283)

[3.4.1. Môi trường phát triển 47](#_Toc183444284)

[3.4.2. Các công nghệ sử dụng 47](#_Toc183444285)

[3.5. Thiết kế giao diện 49](#_Toc183444286)

[3.5.1. Giao diện trang chủ 49](#_Toc183444287)

[3.5.2. Giao diện giới thiệu 49](#_Toc183444288)

[3.5.3. Giao diện đề án tuyển sinh 53](#_Toc183444289)

[3.5.4. Giao diện chatbot 53](#_Toc183444290)

[3.5.5. Giao diện phản hồi 54](#_Toc183444291)

[3.6. Kết luận chương. 54](#_Toc183444292)

[KẾT LUẬN 55](#_Toc183444293)

[1. Kết quả đạt được 55](#_Toc183444294)

[2. Hạn chế của đề tài 55](#_Toc183444295)

[3. Hướng phát triển 55](#_Toc183444296)

[TÀI LIỆU THAM KHẢO 56](#_Toc183444297)

**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Từ chuẩn** | **Diễn giải** |
| 1 | AL | Artificial Intelligence | Trí tuệ nhân tạo |
| 2 | ML | Machine Learning | Học máy |
| 3 | DL | Deep Learning | Học sâu |
| 4 | NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| 5 | RAG | Retrieval-Augmented Generation | Tạo sinh tăng cường bằng truy xuất |
| 6 | LLM | Large Language Model | Mô hình ngôn ngữ lớn |
| 7 | PTIT | Posts and Telecommunications Institute of Technology | Học viện Công nghệ Bưu chính Viễn thông |
| 8 | RNN | Recurrent Neural Network | Mạng nơ-ron hồi quy |
|  | LSTM | Long Short-Term Memory | Bộ nhớ dài-ngắn hạn |
| 9 | TF | Term Frequency | Tần suất xuất hiện của từ |
| 10 | IDF | Inverse Document Frequency | Tần suất ngược tài liệu |
| 11 | CBOW | Continuous bag-of-words | Túi từ liên tục |

**DANH MỤC CÁC** **HÌNH VẼ**

[Hình 1.1. Phân loại chatbot [1] 11](#_Toc184048066)

[Hình 1.2. Ví dụ về chatbot dựa trên kịch bản. 11](#_Toc184048067)

[Hình 1.3. Kiến trúc của chatbot dựa trên truy xuất. [1] 12](#_Toc184048068)

[Hình 1.4. Các thành phần của hệ thống chatbot. 16](#_Toc184048069)

[Hình 2.1. Mối quan hệ giữa AL, ML và DL. 20](#_Toc184048070)

[Hình 2.2. Kiến trúc mạng nơ-ron. [2] 22](#_Toc184048071)

[Hình 2.3. Quá trình xử lý thông tin trong nơ-ron. 23](#_Toc184048072)

[Hình 2.4. Mạng nơ-ron hồi quy. 24](#_Toc184048073)

[Hình 2.5. Mạng RNN 2 chiều. 26](#_Toc184048074)

[Hình 2.6. Mạng RNN sâu*.* 26](#_Toc184048075)

[Hình 2.7. Mô hình từ nhúng. 29](#_Toc184048076)

[Hình 2.8. Mô hình CBOW và Skip-Ngram. 29](#_Toc184048077)

[Hình 2.9. Chuỗi được mã hóa thành mã thông báo. 30](#_Toc184048078)

[Hình 2.10. Hình biểu diễn vector nhúng của chuỗi qua các bước. 31](#_Toc184048079)

[Hình 2.11. Hình chuỗi được đánh ký hiệu vị trí position. 32](#_Toc184048080)

[Hình 2.12. Hình mô tả lớp nhúng của mô hình BERT. 34](#_Toc184048081)

[Hình 2.13. Hình biểu diễn độ tương đồng jaccard 39](#_Toc184048082)

[Hình 2.14. Hình biểu diễn khoảng cách jaccard 39](#_Toc184048083)

[Hình 2.15. Hình biểu diễn độ tương đồng cosine giữa 2 vector. 41](#_Toc184048084)

[Hình 2.16. Hình biểu diễn các giai đoạn khi xếp hạng lại. 43](#_Toc184048085)

[Hình 3.1. Quá trình xây dựng dữ liệu cho chatbot. 48](#_Toc184048086)

[Hình 3.2. Bảng đánh giá hiệu suất mô hình trên tập dữ liệu huấn luyện. 48](#_Toc184048087)

[Hình 3.3. Bảng đánh giá hiệu suất mô hình trên tập dữ liệu kiểm tra. 49](#_Toc184048088)

[Hình 3.4. Giao diện trang chủ 51](#_Toc184048089)

[Hình 3.5. Tổng quan học viện. 51](#_Toc184048090)

[Hình 3.6. Chính sách học bổng. 52](#_Toc184048091)

[Hình 3.7. Các câu hỏi thường gặp. 52](#_Toc184048092)

[Hình 3.8. Điểm các ngành năm 2024. 53](#_Toc184048093)

[Hình 3.9. Điểm các ngành năm 2023. 53](#_Toc184048094)

[Hình 3.10. Điểm các ngành năm 2022. 54](#_Toc184048095)

[Hình 3.11. Điểm các ngành năm 2021. 54](#_Toc184048096)

[Hình 3.12. Giao diện đề án tuyển sinh. 55](#_Toc184048097)

[Hình 3.13. Giao diện chatbot. 55](#_Toc184048098)

[Hình 3.14. Giao diện phản hồi. 56](#_Toc184048099)

**DANH MỤC CÁC BẢNG**

[Bảng 3.1. Kịch bản về điểm chuẩn ngành công nghệ thông tin. 46](#_Toc183444234)

# LỜI MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, việc áp dụng các giải pháp tự động hóa vào các hoạt động thường ngày, đặc biệt trong lĩnh vực giáo dục, ngày càng trở nên phổ biến. Một trong những ứng dụng nổi bật là chatbot, công cụ hỗ trợ giao tiếp tự động, mang lại sự thuận tiện và hiệu quả trong việc giải đáp thông tin. Chatbot không chỉ giúp giảm tải công việc cho nhân viên mà còn nâng cao trải nghiệm người dùng nhờ khả năng phản hồi nhanh chóng và chính xác.

Đồ án này nhằm phát triển một chatbot hỗ trợ quá trình tư vấn và giải đáp thông tin tuyển sinh tại Học viện Công nghệ bưu chính viễn thông. Nó sẽ cung cấp các thông tin liên quan đến kỳ thi tuyển sinh, điều kiện xét tuyển, chương trình đào tạo, và các thủ tục đăng ký một cách tự động.

Mục tiêu chính của đồ án là xây dựng một chatbot có khả năng hiểu ngôn ngữ tự nhiên, đáp ứng nhu cầu của thí sinh trong việc tra cứu thông tin tuyển sinh, qua đó góp phần cải thiện quy trình tuyển sinh của nhà trường. Đồng thời, đồ án cũng sẽ đề cập đến các kỹ thuật và công nghệ hiện đại được áp dụng trong quá trình xây dựng chatbot.

Đồ án tập trung trình bày một số nội dung chính như sau:

**Chương 1: Tổng quan các hệ thống chatbot:**

Nội dung chương này sẽ trình bày về những kiến thức tổng quan về chatbot những kiến thức tổng quan nhất về hệ thống chatbot, phân tích các ưu nhược điểm của các mô hình chatbot và một số ứng dụng của chatbot phổ biến trong thời đại ngày nay, từ đó định hướng xây dựng mô hình chatbot mà đồ án nghiên cứu và xây dựng.

**Chương 2: Cơ sở lý thuyết:**

Chương này giới thiệu một số kiến thức nền tảng về học sâu, cũng như mạng nơ-ron nhân tạo, mạng nơ ron hồi quy, cung cấp cái nhìn khái quát về nền tảng kỹ thuật sử dụng trong chatbot. Các kỹ thuật nhúng từ, bao gồm TF-IDF, Word2Vec và PhoBERT. Bên cạnh đó, các phương pháp tính độ tương đồng cũng được đề cập để đánh giá mức độ tương đồng giữa câu truy vấn và các câu hỏi tiềm năng. Cuối cùng, xếp hạng lại (reranking) giúp nâng cao độ chính xác trong việc lựa chọn câu trả lời phù hợp nhất cho người dùng. Các kiến thức này là nền tảng quan trọng để phát triển chatbot phục vụ tuyển sinh trong những chương tiếp theo.

**Chương 3: Xây dựng chatbot tuyển sinh:**

Nội dung chương 3 sẽ trình bày về quá trình xây dựng chatbot tư vấn tuyển sinh, bao gồm quá trình thu thập dữ liệu, thử nghiệm và cài đặt ứng dụng. Các công nghệ và môi trường phát triển được lựa chọn phù hợp để đảm bảo tính linh hoạt và hiệu quả của hệ thống. Chương này cũng đề cập chi tiết về thiết kế giao diện nhằm mang lại trải nghiệm tối ưu cho người dùng.

**Kết luận:**

Tổng kết bài toán, tóm tắt những kết quả đã đạt được và hạn chế của đồ án. Từ đó đề xuất mục tiêu hướng tới cũng như hướng nghiên cứu, phát triển tiếp theo.

# CHƯƠNG 1: TỔNG QUAN CÁC HỆ THỐNG CHATBOT

## 1.1. Khái niệm chatbot

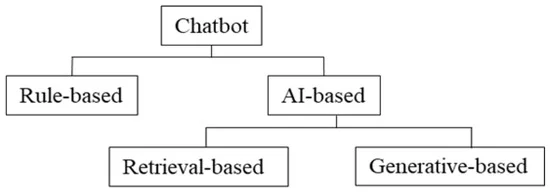
Chatbot là một ứng dụng hoặc chương trình máy tính được tạo ra với mục đích tương tác với con người thông qua các cuộc trò chuyện. Sử dụng công nghệ trí tuệ nhân tạo (Artifical Intelligent - AI) và xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), chatbot có khả năng hiểu và phản hồi lại các câu hỏi từ người dùng một cách tự động. Bằng cách này, chatbot giúp tạo ra trải nghiệm gần như giao tiếp với con người, giải quyết các vấn đề cơ bản và cung cấp thông tin trong nhiều lĩnh vực khác nhau, từ dịch vụ khách hàng đến giáo dục và giải trí... Sự phổ biến của chatbot đã tăng lên trong thời gian gần đây, với việc tích hợp chúng vào các nền tảng trò chuyện trực tuyến, ứng dụng di động và trang web, mang lại lợi ích to lớn cho cả người dùng và doanh nghiệp thông qua việc tiết kiệm thời gian và công sức.

Chatbot có thể được chia thành hai loại chính: Miền mở (Open Domain), miền đóng (Close Domain) dựa trên chủ đề và tính linh hoạt trong việc xử lý thông tin.

Miền mở (Open Domain): Chatbot này có khả năng trò chuyện về nhiều chủ đề khác nhau và không giới hạn trong một lĩnh vực cụ thể, nó dựa vào các mô hình học máy và xử lý ngôn ngữ tự nhiên để hiểu ý nghĩa của câu hỏi và đưa ra phản hồi dựa trên ngữ cảnh. Nó tương tự như cuộc hội thoại trên mạng xã hội khi chat với nhau, có thể nói chuyện về một bộ phim và chuyển sang chủ đề nhạc, rồi lại quay sang chủ đề bóng đá. Cũng vì do số lượng chủ đề là rất lớn, hệ thống cần phải trả về lượng phản hồi tương ứng nên đây trở thành một bài toán khá khó. Các trợ lý ảo như Siri, Google Assistant hoặc ChatGPT đều là ví dụ điển hình của miền mở.

Miền đóng (Close Domain): Chatbot này được thiết kế để hoạt động trong một phạm vi cụ thể và chỉ trả lời các câu hỏi liên quan đến lĩnh vực hoặc chủ đề nhất định, ví dụ như: thời tiết, y tế, giáo dục, mua sắm, tâm lý, du lịch, ngân hàng, …nó thường dựa vào các quy tắc và kịch bản đã được lập trình sẵn, giúp xử lý những yêu cầu cụ thể. Khi nhận được câu hỏi từ người dùng, chatbot sẽ tìm từ khóa và dựa vào kịch bản để đưa ra câu trả lời thích hợp. Hệ thống hỗ trợ kỹ thuật hay các dịch vụ chăm sóc khách hàng là các ứng dụng thuộc miền đóng.

## 1.2. Phân loại chatbot



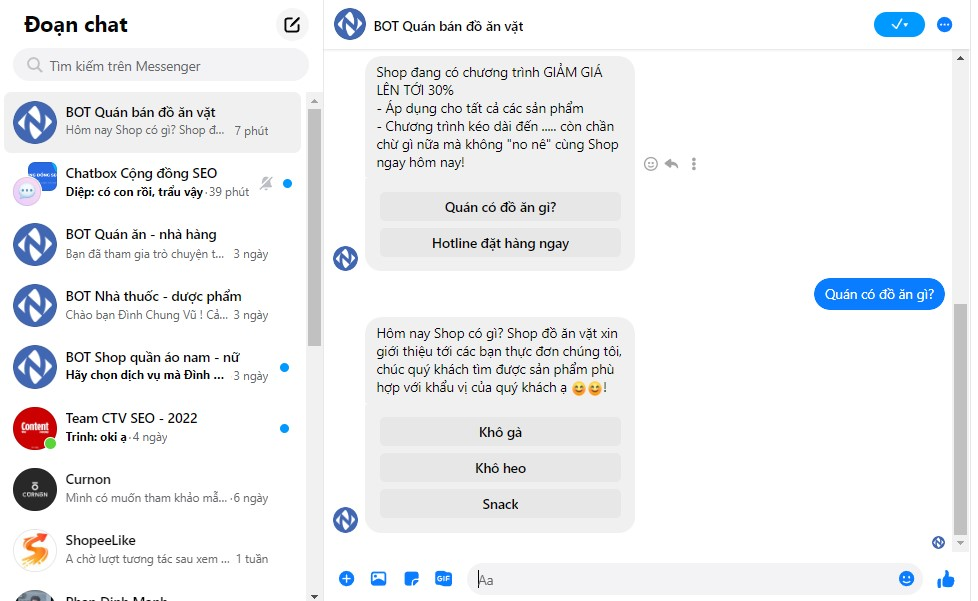
Hình 1.1. Phân loại chatbot [1]

Dựa theo cách thức hoạt động thì chatbot được chia làm 2 loại chính là Rule-based chatbot và Al-based chatbot.

### 1.2.1. Rule-based chatbot

**Chatbot dựa trên kịch bản (Script-Based chatbot)**

Chatbot dựa trên kịch bản là một loại chatbot hoạt động dựa trên các kịch bản hoặc quy trình đã được lập trình sẵn để tương tác với người dùng. Loại chatbot này hoạt động theo một chuỗi các bước được xác định trước, điều này giúp nó cung cấp phản hồi và hướng dẫn người dùng theo một trình tự nhất định. Dưới đây là một vài ví dụ về chatbot này.



Hình 1.2. Ví dụ về chatbot dựa trên kịch bản.

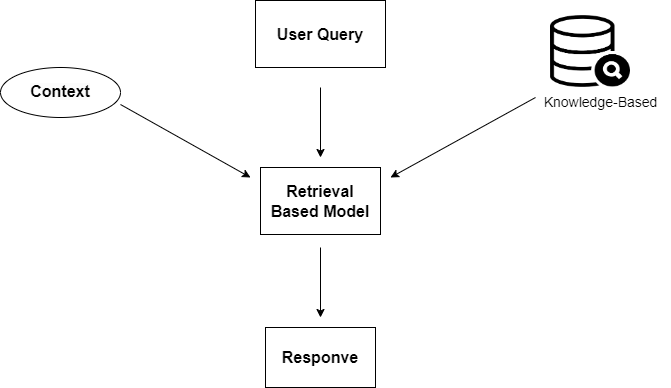
Ở ví dụ trên chatbot đã xây dựng sẵn các nút tương ứng với yêu cầu của người dùng, chỉ cần chọn vào đúng nút theo yêu cầu của mình thì chatbot sẽ phản hồi dựa trên yêu cầu đó.

Ưu điểm của chatbot này là xây dựng rất dễ dàng, độ chính xác cao vì người dùng đưa ra yêu cầu dựa trên kịch bản có trước. Tuy nhiên người dùng rất bị động trước những câu hỏi không có trong kịch bản, mà phụ thuộc hoàn toàn vào các câu hỏi đã được cung cấp sẵn của chatbot.

### 1.2.2. AI-based chatbot

**Mô hình chatbot mà đồ án nghiên cứu - Chatbot dựa trên truy xuất**

Chatbot dựa trên truy xuất là một loại chatbot hoạt động bằng cách tìm kiếm và trả lời các câu hỏi hoặc yêu cầu của người dùng dựa trên một tập hợp các câu hỏi và câu trả lời đã được lưu trữ trước đó. Phương pháp này chủ yếu dựa vào cơ sở dữ liệu hoặc kho lưu trữ thông tin chứa các câu hỏi và câu trả lời mẫu, cùng với các công cụ truy vấn để tìm ra câu trả lời phù hợp nhất cho câu hỏi của người dùng. Khi người dùng gửi một câu hỏi, chatbot sẽ so sánh câu hỏi đó với các câu hỏi trong cơ sở dữ liệu và chọn ra câu trả lời tương ứng mà nó cho là phù hợp nhất.

****

Hình 1.3. Kiến trúc của chatbot dựa trên truy xuất. [1]

Một trong những ưu điểm chính của các chatbot dựa trên truy xuất là sự dễ dàng trong việc triển khai và quản lý. Do chúng hoạt động dựa trên các câu trả lời đã được chuẩn bị sẵn, việc phát triển và duy trì các chatbot này không đòi hỏi phải xây dựng các mô hình học sâu phức tạp. Điều này cũng giúp giảm thiểu nguy cơ phát sinh lỗi trong phản hồi, vì các câu trả lời đã được kiểm chứng trước và được lưu trữ một cách có tổ chức.

Tuy nhiên, hạn chế của các chatbot dựa trên truy xuất là chúng có thể gặp khó khăn khi phải xử lý các câu hỏi hoặc yêu cầu không nằm trong cơ sở dữ liệu đã được lập trình sẵn. Khi người dùng đặt câu hỏi ngoài phạm vi các câu hỏi đã chuẩn bị, chatbot có thể không cung cấp được phản hồi thỏa đáng hoặc có thể trả lời sai. Điều này có thể dẫn đến trải nghiệm người dùng không tốt và hạn chế khả năng của chatbot trong việc đáp ứng các yêu cầu đa dạng và phức tạp.

**Chatbot dựa trên mô hình ngôn ngữ lớn**

Chatbot dựa trên mô hình ngôn ngữ lớn là loại chatbot sử dụng các mô hình học sâu với khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ để tương tác với người dùng, nó cũng là chatbot thuộc loại generative chatbot. Các mô hình như GPT (Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers), T5 (Text-To-Text Transfer Transformer)... có khả năng hiểu và tạo ra ngôn ngữ một cách tự nhiên và chính xác. Nhờ vào việc được huấn luyện trên khối lượng dữ liệu khổng lồ, những mô hình này có thể nắm bắt ngữ nghĩa, ngữ cảnh, và các sắc thái của ngôn ngữ, từ đó tạo ra các phản hồi hợp lý và mạch lạc trong cuộc trò chuyện.

Một trong những ưu điểm nổi bật của các chatbot dựa trên mô hình ngôn ngữ lớn là khả năng hiểu ngữ nghĩa sâu rộng và tạo ra các phản hồi tự nhiên. Điều này cho phép chatbot có thể xử lý các câu hỏi phức tạp, duy trì mạch trò chuyện một cách liên tục, và đáp ứng linh hoạt với các yêu cầu đa dạng của người dùng. Việc sử dụng các mô hình ngôn ngữ lớn cũng giúp giảm thiểu sự cần thiết phải xây dựng và duy trì các quy tắc và kịch bản phức tạp, vì mô hình đã được huấn luyện để hiểu và phản ứng dựa trên ngữ cảnh của cuộc trò chuyện.

Tuy nhiên, việc triển khai chatbot dựa trên mô hình ngôn ngữ lớn cũng đi kèm với một số thách thức. Một trong những vấn đề chính là yêu cầu về tài nguyên tính toán và bộ nhớ lớn, do các mô hình này thường có kích thước rất lớn và đòi hỏi sức mạnh xử lý cao để hoạt động hiệu quả. Điều này có thể dẫn đến chi phí cao cho việc vận hành và duy trì hệ thống. Hơn nữa, mặc dù các mô hình ngôn ngữ lớn có khả năng tạo ra phản hồi chính xác và tự nhiên, nhưng chúng vẫn có thể gặp phải các vấn đề như sinh ra thông tin sai lệch hoặc không liên quan nếu không được huấn luyện và tinh chỉnh đúng cách.

## 1.3. Các nền tảng xây dựng chatbot

**Dialogflow**

Dialogflow là một nền tảng mạnh mẽ được phát triển bởi Google, cho phép xây dựng chatbot và các ứng dụng thoại có khả năng hiểu ngôn ngữ tự nhiên. Với Dialogflow, bạn có thể tạo ra những chatbot thông minh có khả năng hiểu và phản hồi tự nhiên.

Điểm mạnh của Dialogflow là hỗ trợ nhiều ngôn ngữ và tích hợp dễ dàng với Google Cloud, mang lại khả năng mở rộng và tính linh hoạt cao. Nền tảng này cũng hỗ trợ tích hợp với nhiều nền tảng nhắn tin như Facebook Messenger, Slack, Telegram, giúp doanh nghiệp dễ dàng tiếp cận khách hàng trên nhiều kênh giao tiếp.

**Microsoft Bot Framework**

Microsoft Bot Framework được phát triển bởi Microsoft, nó cung cấp công cụ và dịch vụ để xây dựng, thử nghiệm và triển khai chatbot. Nền tảng này hỗ trợ nhiều kênh giao tiếp như Skype, Microsoft Teams, Slack, và Facebook Messenger, mang lại khả năng tiếp cận đa dạng.

**Rasa**

Rasa là một nền tảng mã nguồn mở cho phép xây dựng chatbot AI tùy chỉnh với khả năng hiểu và xử lý ngôn ngữ tự nhiên. Với Rasa, các nhà phát triển có thể hoàn toàn tùy biến chatbot theo nhu cầu cụ thể và kiểm soát toàn bộ mã nguồn. Rasa hỗ trợ tích hợp với nhiều kênh giao tiếp, mang lại sự linh hoạt và khả năng mở rộng cao. Đây là lựa chọn lý tưởng cho những ai muốn xây dựng chatbot chuyên sâu và có khả năng tùy chỉnh tối đa.

**Botpress**

Botpress là một nền tảng chatbot AI đa năng, có thể tùy chỉnh và mở rộng vô tận. Nó luôn được cập nhật với những thông tin mới nhất LLM công cụ, đảm bảo chatbot và tác nhân AI của nó luôn được hỗ trợ bởi công nghệ mới nhất.

Botpress Cung cấp canvas kéo và thả trực quan cho các nhà phát triển, bản dịch tự động cho hơn 100 ngôn ngữ và khả năng tùy chỉnh vô tận. Nền tảng này bao gồm các tích hợp được xây dựng sẵn với phần mềm và kênh phổ biến nhất, nhưng cho phép các nhà phát triển kết nối bot của họ với bất kỳ cơ sở kiến thức hoặc nền tảng nội bộ nào. Khả năng mở rộng vô tận này làm cho Botpress một nền tảng tuyệt vời cho các tác nhân AI chuyên nghiệp, cấp doanh nghiệp.

**IBM watsonx Assistant**

IBM watsonx Assistant là một nền tảng AI đàm thoại được thiết kế để xây dựng trợ lý ảo và giọng nói cho các ứng dụng dịch vụ khách hàng. Nó tận dụng trí tuệ nhân tạo và các mô hình ngôn ngữ lớn để học hỏi từ các tương tác của khách hàng, nhằm cải thiện hiệu quả giải quyết vấn đề và giảm thời gian chờ đợi của khách hàng.

Không giống như các chatbot truyền thống, Watsonx Assistant có thể truy vấn các cơ sở kiến thức, tìm kiếm sự làm rõ hoặc báo cáo lên một tác nhân con người khi cần thiết. Nó có thể thích ứng để sử dụng trong các môi trường khác nhau, bao gồm thiết lập đám mây và tại chỗ.

Nền tảng này cũng cung cấp khả năng thoại, cho phép tích hợp vào các hệ thống hỗ trợ khách hàng qua điện thoại. IBM quảng bá Watsonx Assistant như một công cụ để nâng cao hiệu quả và hiệu quả của các tương tác dịch vụ khách hàng.

## 1.4. Các thành phần của chatbot mà đồ án nghiên cứu

Hình 1.4. Các thành phần của hệ thống chatbot.

* Web + Selenium:

Dữ liệu từ trang web liên quan đến tuyển sinh PTIT(Học viện Công nghệ Bưu chính Viễn thông) được thu thập thông qua quá trình crawl dữ liệu sử dụng Selenium, một công cụ tự động hóa trình duyệt.

* Dữ liệu:

Sau khi crawl data, dữ liệu được lưu lại dưới dạng json để thuận lợi cho việc xử lý.

* Dữ liệu hỏi và đáp:

Dữ liệu sẽ được xử lý để chuyển đổi thành các cặp câu hỏi và câu trả lời (Q&A) và được lưu trữ trong file CSV cho mục đích truy xuất sau này. Các câu hỏi trong tập Q&A cũng được xử lý và được nhúng vào mô hình embedding.

* Câu hỏi người dùng:

Khi người dùng nhập câu hỏi cho chatbot, câu hỏi sẽ được xử lý và được nhúng vào mô hình embedding.

* Mô hình nhúng:

Sử dụng mô hình PhoBERT embedding để chuyển câu hỏi của người dùng và câu hỏi trong tập dữ liệu Q&A đã được xử lý thành các vector. Các vector này là các biểu diễn số của câu hỏi và dữ liệu, giúp hệ thống hiểu được ý nghĩa ngữ nghĩa của chúng.

* Lưu trữ và bảo quản:

Dữ liệu sau khi được chuyển thành embedding sẽ được lưu trữ trong cơ sở dữ liệu MongoDb và có thể được truy xuất sau này để so khớp với câu hỏi của người dùng. Khi có một truy vấn từ người dùng, hệ thống sẽ thực hiện tìm kiếm các tài liệu tương đồng nhất trong cơ sở dữ liệu và trả về top k tài liệu liên quan nhất.

* Xếp hạng lại:

Một mô hình xếp hạng lại kết hợp cả truy vấn của người dùng và top k tài liệu để tìm ra kết quả tốt nhất và phù hợp nhất.

* Câu trả lời:

Sau khi reranker xác định được tài liệu phù hợp nhất, câu trả lời cuối cùng sẽ được server gửi tới client.

* Client Web

Client Web là giao diện mà người dùng sẽ tương tác với chatbot. Các câu hỏi được nhập vào đây và câu trả lời sẽ được hiển thị lại cho người dùng thông qua giao diện này.

Trên đây là kiến trúc của Retrieval-Based Chatbot cũng như là kiến trúc của đồ án chatbot được nghiên cứu. Kiến trúc này tối ưu hóa việc xử lý câu hỏi người dùng, sau đó chuyển câu hỏi đấy thành vector thông qua mô hình PhoBERT embedding. Bằng cách đấy, hệ thống có thể hiểu được ngữ nghĩa của chúng và tìm kiếm các tài liệu liên quan nhất thông qua các phép toán so sánh độ tương đồng. Hơn nữa, quy trình reranking cho phép cải thiện độ chính xác của câu trả lời cuối cùng bằng cách kết hợp giữa truy vấn của người dùng và các tài liệu liên quan nhất. Toàn bộ quá trình này được tích hợp trên giao diện web, giúp người dùng tương tác một cách dễ dàng và tiện lợi. Kiến trúc này không chỉ phù hợp với hệ thống chatbot hỗ trợ tuyển sinh mà còn có tiềm năng ứng dụng trong nhiều lĩnh vực khác, mang lại hiệu quả cao trong việc truy xuất thông tin và tương tác với người dùng.

## 1.5. Một số ứng dụng của chatbot

Trong thời đại công nghệ số, sự phát triển mạnh mẽ của trí tuệ nhân tạo đã mở ra nhiều cơ hội mới cho các doanh nghiệp và tổ chức, trong đó nổi bật là chatbot – một trong những ứng dụng quan trọng và phổ biến nhất của trí tuệ nhân tạo. Chatbot thường được tích hợp trong các ứng dụng nhắn tin, website, hoặc các nền tảng thương mại điện tử. Với khả năng tương tác linh hoạt và không giới hạn thời gian, chatbot đã và đang trở thành một công cụ hỗ trợ đắc lực trong nhiều lĩnh vực, từ chăm sóc khách hàng, tư vấn sản phẩm, đến marketing và giáo dục. Việc sử dụng chatbot không chỉ giúp tối ưu hóa hiệu quả vận hành mà còn mang lại trải nghiệm người dùng tốt hơn, đáp ứng nhanh chóng nhu cầu của khách hàng.

### 1.5.1. Dịch vụ khách hàng

Trợ lý 24/7: Chatbot có thể hoạt động suốt ngày đêm để giải đáp các câu hỏi thường gặp của khách hàng, từ việc tra cứu thông tin sản phẩm, trạng thái đơn hàng, đến việc hỗ trợ xử lý các vấn đề kỹ thuật cơ bản.

Tự động hóa quy trình: Chatbot giúp tự động hóa các quy trình hỗ trợ khách hàng như đặt lịch hẹn, xử lý yêu cầu bảo hành, hoặc thay đổi thông tin tài khoản mà không cần sự can thiệp của con người.

### 1.5.2. Marketing và quảng cáo

Chatbot quảng bá sản phẩm: Thông qua các nền tảng như Facebook Messenger hoặc WhatsApp, chatbot có thể gửi thông tin sản phẩm mới, chương trình khuyến mãi và nội dung quảng cáo tới khách hàng tiềm năng.

Thu thập dữ liệu khách hàng: Chatbot giúp thu thập thông tin từ người dùng thông qua các cuộc trò chuyện, từ đó hỗ trợ doanh nghiệp phân tích nhu cầu và tối ưu hóa chiến lược tiếp thị.

### 1.5.3. Tài chính và ngân hàng

Trợ lý tài chính cá nhân: Chatbot có thể giúp người dùng quản lý tài chính, đưa ra lời khuyên về chi tiêu, và nhắc nhở về hóa đơn, thanh toán sắp đến hạn.

Hỗ trợ giao dịch: Các ngân hàng sử dụng chatbot để hỗ trợ người dùng kiểm tra số dư, lịch sử giao dịch, và xử lý các lệnh chuyển tiền.

### 1.5.4. Giáo dục

Trợ giảng ảo: Chatbot có thể hỗ trợ học sinh, sinh viên trong việc giải đáp thắc mắc về bài học, cung cấp tài liệu tham khảo, hoặc đưa ra các bài kiểm tra và bài tập luyện tập.

Tư vấn tuyển sinh: Nhiều trường học sử dụng chatbot để trả lời câu hỏi về quy trình tuyển sinh, học bổng, và các thông tin cần thiết cho sinh viên mới.

### 1.5.5. Chăm sóc sức khỏe

Tư vấn sức khỏe ban đầu: Chatbot giúp người dùng đánh giá tình trạng sức khỏe cơ bản bằng cách hỏi các triệu chứng và đưa ra khuyến nghị, bao gồm việc đến gặp bác sĩ hoặc tự chăm sóc tại nhà.

Đặt lịch hẹn và nhắc nhở: Chatbot có thể giúp bệnh nhân đặt lịch hẹn khám bệnh và nhắc nhở về các cuộc hẹn, cũng như hướng dẫn sử dụng thuốc.

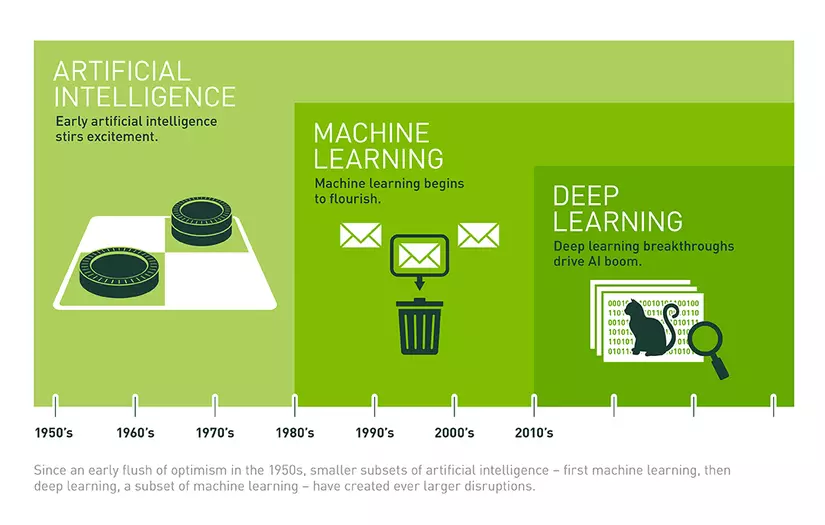
## 1.6. Kết luận chương

Nội dung chương này đã trình bày và giới thiệu những kiến thức tổng quan nhất về hệ thống chatbot, phân tích các ưu nhược điểm của các mô hình chatbot và một số ứng dụng của chatbot phổ biến trong thời đại ngày nay, từ đó định hướng xây dựng mô hình chatbot mà đồ án nghiên cứu và xây dựng.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Giới thiệu về học sâu

Trí tuệ nhân tạo đang len lỏi vào trong cuộc sống và ảnh hưởng sâu rộng tới mọi khía cạnh, các cụm từ "Artificial Intelligence", "Machine Learing", "Deep Learning" đã không còn quá xa lạ gì. Dưới dây là hình vẽ để mô tả lại mối quan hệ giữa trí tuệ nhân tạo, học máy và học sâu:



Hình 2.1. Mối quan hệ giữa AL, ML và DL.

Học sâu đã và đang là một chủ đề Al được bàn luận sôi nổi. Là một phạm trù nhỏ của ML, DL tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, thị giác máy tính và xử lý ngôn ngữ tự nhiên.

**Nhận dạng giọng nói**

Các mô hình học sâu có thể phân tích giọng nói con người, bất kể mẫu giọng, cao độ, tông, ngôn ngữ và giọng vùng miền khác nhau. Trợ lý ảo như Amazon Alexa và phần mềm phiên âm tự động sử dụng nhận dạng giọng nói để thực hiện các tác vụ sau:

* Hỗ trợ các nhân viên trực tổng đài và tự động phân loại cuộc gọi.
* Chuyển đổi các cuộc trò chuyện về y khoa thành văn bản trong thời gian thực.
* Tạo phụ đề chính xác cho video và bản ghi âm cuộc họp để mở rộng phạm vi tiếp cận nội dung.

**Thị giác máy tính**

Thị giác máy tính là khả năng của máy tính thực hiện trích xuất dữ liệu cũng như thông tin chuyên sâu từ hình ảnh và video. Máy tính có thể sử dụng các kỹ thuật học sâu để hiểu hình ảnh theo cách giống như con người. Thị giác máy tính được ứng dụng trong nhiều trường hợp, chẳng hạn như:

* Kiểm duyệt nội dung để tự động loại bỏ nội dung không an toàn hoặc không phù hợp khỏi kho lưu trữ hình ảnh và video.
* Nhận diện khuôn mặt để xác định khuôn mặt cũng như các đặc điểm như mở mắt, đeo kính và để râu.
* Phân loại hình ảnh để xác định logo thương hiệu, quần áo, đồ bảo hộ và các chi tiết hình ảnh khác.

**Xử lý ngôn ngữ tự nhiên**

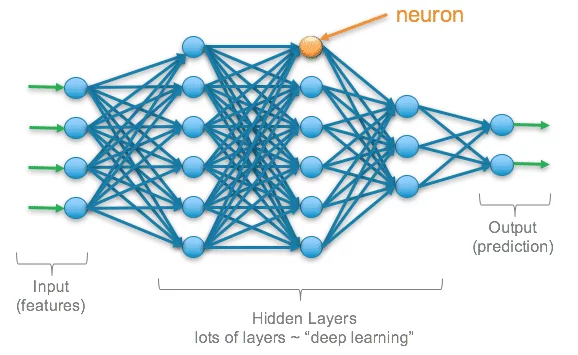
Máy tính sử dụng các thuật toán học sâu để thu thập thông tin chuyên sâu và ý nghĩa từ dữ liệu văn bản và tài liệu. Khả năng xử lý văn bản tự nhiên, do con người tạo ra này có một số trường hợp sử dụng, kể cả trong các chức năng sau:

* Tổng đài viên ảo và chatbot tự động.
* Tự động tóm tắt tài liệu hoặc bài viết tin tức.
* Phân tích nghiệp vụ thông minh các tài liệu dài, chẳng hạn như các email và biểu mẫu.
* Lập chỉ mục các cụm từ quan trọng thể hiện cảm xúc, chẳng hạn như những bình luận tích cực và tiêu cực trên mạng xã hội.

**Mô hình học sâu hoạt động như thế nào?**

Các thuật toán học sâu là các mạng nơ-ron được lập mô hình theo bộ não con người. Ví dụ: một bộ não con người chứa hàng triệu nơ-ron được kết nối với nhau, làm việc cùng nhau để tìm hiểu và xử lý thông tin. Tương tự, các mạng nơ-ron học sâu, hay mạng nơ-ron nhân tạo, được tạo thành từ nhiều lớp nơ-ron nhân tạo hoạt động cùng nhau bên trong máy tính.

Các nơ-ron nhân tạo là những mô-đun phần mềm được gọi là nút, sử dụng các phép toán để xử lý dữ liệu. Các mạng nơ-ron nhân tạo là những thuật toán học sâu sử dụng các nút này để giải quyết các vấn đề phức tạp.

**

Hình 2.2. Kiến trúc mạng nơ-ron. [2]

Kiến trúc chung của mạng nơ-ron bao gồm 3 thành phần đó là lớp đầu vào, lớp ẩn và lớp đầu ra.

**Lớp đầu vào**

Một mạng nơ-ron nhân tạo sẽ có một số nút để nhập dữ liệu đầu vào. Các nút này tạo nên lớp đầu vào của hệ thống.

**Lớp ẩn**

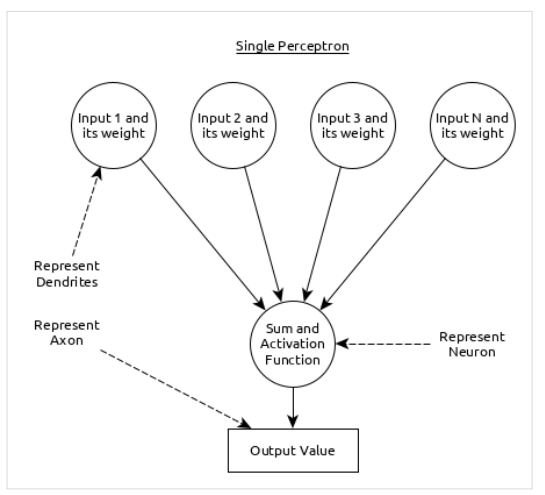
Lớp đầu vào xử lý và chuyển dữ liệu đến các lớp sâu hơn trong mạng nơ-ron. Các lớp ẩn này xử lý thông tin ở các cấp độ khác nhau, thích ứng với hành vi của mình khi nhận được thông tin mới. Các mạng học sâu có hàng trăm lớp ẩn có thể được dùng để phân tích một vấn đề từ nhiều góc độ khác nhau.

Các lớp ẩn trong mạng nơ-ron chuyên sâu hoạt động theo cùng một cách. Nếu một thuật toán học sâu đang cố gắng phân loại một hình ảnh động vật, mỗi lớp ẩn của thuật toán này sẽ xử lý một đặc điểm khác nhau của con vật và cố gắng phân loại chính xác nó.

**Lớp đầu ra**

Lớp đầu ra bao gồm các nút xuất dữ liệu. Các mô hình học sâu xuất ra đáp án "có" hoặc "không" chỉ có hai nút trong lớp đầu ra. Mặt khác, các mô hình xuất ra nhiều đáp án hơn sẽ có nhiều nút hơn.

Trên đây là kiến trúc về mạng nơ-ron. Còn quá trình xử lý thông tin trong một nơ-ron hay còn gọi là một nút thì được biểu diễn như dưới đây.

**

Hình 2.3. Quá trình xử lý thông tin trong nơ-ron.

Mỗi input tương ứng với 1 đặc trưng của dữ liệu. Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi input là 1 đặc trưng của khách hàng như thu nhập, nghề nghiệp, tuổi… Output là kết quả, với bài toán trên, nếu chấp nhận cho khách hàng vay tiền thì output là yes - cho vay hoặc no - không cho vay. Trọng số (weights) là thành phần rất quan trọng, nó thể hiện mực độ quan trọng của dữ liệu đầu vào đối với quá trình xử lý thông tin, chuyển đổi dữ liệu từ layer này qua layer khác. Quá trình học của mạng nơ-ron thực chất là quá trình điều chỉnh các trọng số (weights) của các input để có được kết quả mong muốn. Hàm tính tổng (Sum Function) cho phép tính tổng trọng số của tất cả các input và độ lệch bias. Phép tính này được biểu diễn dưới dạng:

|  |  |  |
| --- | --- | --- |
|  |  |  |

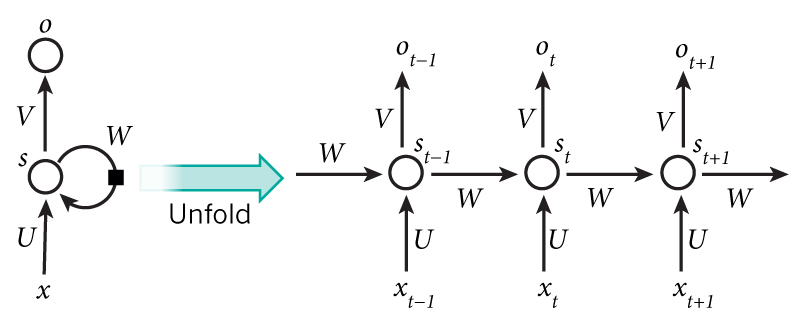
Nó xác định tổng trọng số được truyền như một đầu vào cho hàm kích hoạt để tạo ra đầu ra. Các hàm kích hoạt chọn xem nút đó có nên kích hoạt hay không. Chỉ những nút được kích hoạt mới được đưa vào lớp đầu ra. Các nhiều hàm kích hoạt riêng biệt có thể áp dụng riêng cho từng bài toán, một trong những hàm kích hoạt phổ biến là sigmoid function.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Kết quả của sigmoid function thuộc khoảng [0,1] nên còn gọi là hàm chuẩn hóa (Normalized Function).

## 2.2. Mạng nơ-ron hồi quy

Mạng nơ-ron hồi quy (RNN - Recurrent Neural Network) là một trong những mô hình học sâu được đánh giá là có nhiều ưu điểm trong các tác vụ xử lý ngôn ngữ tự nhiên. Ý tưởng chính của RNN là thiết kế một mạng lưới thần kinh có khả năng sử dụng chuỗi thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Nên chúng ta có mạng nơ-ron hồi quy bởi chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào kết quả của các phép tính trước đó. Hay nói cách khác, RNN là một mô hình có bộ nhớ, có thể nhớ các thông tin được tính toán trước đó. Về mặt lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó. Về cơ bản một mạng RNN có dạng như sau:



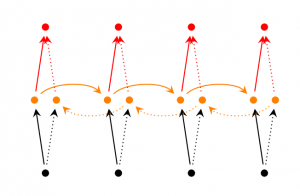
Hình 2.4. Mạng nơ-ron hồi quy.

RNN có khả năng biểu diễn mối quan hệ phụ thuộc giữa các thành phần trong chuỗi. Ví dụ, nếu chuỗi đầu vào là một câu có 5 từ thì RNN này sẽ dàn ra (unfold) thành RNN có 5 layer, mỗi layer tương ứng với 1 từ, chỉ số của các từ được đánh từ 0 đến 4. Lúc đó việc tính toán bên trong RNN được thực hiện như sau:

* xt là đầu vào tại bước t. Ví dụ x2 là 1 vector one-hot tương ứng với từ thứ 3 của câu.
* st là trạng thái ẩn tại bước t. Nó chính là bộ nhớ của mạng. st được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó: st  = f(Uxt  + Wst-1). Hàm f thường là một hàm phi tuyến tính như tang hyperbolic (tanh) hay ReLu.
* ot là đầu vào tại bước t. Ví dụ, ta muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì ot chính là một vec-tơ xác xuất các từ trong danh sách từ vựng của ta: ot = softmax(Vst).

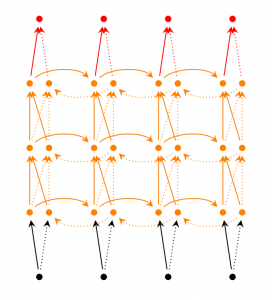
Không giống như mạng nơ-ron truyền thống, ở mỗi lớp cần sử dụng một tham số khác, RNN chỉ sử dụng một bộ tham số (U, V, W) cho tất cả các bước. Về mặt lý thuyết, RNN có thể xử lý và lưu trữ thông tin của một chuỗi dữ liệu có độ dài bất kỳ. Tuy nhiên, trong thực tế, RNN chỉ hiểu quả đối với chuỗi dữ liệu có độ dài không quá dài. Nguyên nhân của vấn đề này là do độ dốc biến mất (gradient được sử dụng để cập nhật giá trị của weight maxtrix trong RNN và nó có giá trị nhỏ dần theo từng layer khi thực hiện back propagation). Khi độ dốc trở nên rất nhỏ (với giá trị gần bằng 0), giá trị của ma trận trọng số sẽ không được cập nhật nữa, do đó mạng nơ-ron sẽ ngừng học ở lớp đấy. Trong vài năm qua, các nhà khoa học đã nghiên cứu và phát triển mở rộng nhiều RNN để giải quyết các hạn chế của RNN.

**RNN 2 chiều (Bidirectional RNN) :** dựa trên ý tưởng rằng đầu ra tại thời điểm t không chỉ phụ thuộc vào các thành phần phía trước mà còn phụ thuộc cả các thành phần phía sau. Ví dụ, để dự đoán từ còn thiếu trong câu, thì việc xem xét cả phần trước và phần sau của câu là cần thiết. Vì vậy, ta có thể coi mô hình là việc chồng 2 mạng RNN ngược hướng nhau lên nhau. Lúc này đầu ra được tính toán dựa vào cả 2 trạng thái ẩn của 2 mạng RNN ngược hướng này.



Hình 2.5. Mạng RNN 2 chiều.

**RNN sâu (Deep RNN) :** tương tự như RNN 2 chiều, sự khác biệt là mô hình này bao gồm nhiều lớp RNN 2 chiều tại cùng 1 thời điểm. Mô hình này giúp cho việc học ở mức độ cao hơn, tuy nhiên cũng cần phải có nhiều dữ liệu huấn luyện hơn.



Hình 2.6. Mạng RNN sâu*.*

**Long Short-Term Memory (LSTM):** Về cơ bản mô hình của LSTM không khác mô hình truyền thống của RNN, nhưng chúng sử dụng hàm tính toán khác ở các trạng thái ẩn. Bộ nhớ của LSTM được gọi là tế bào. Chúng ta có thể thấy đây là một hộp đen nhận thông tin đầu vào bao gồm cả trạng thái và giá trị ẩn. Trong hộp đen này sẽ xác định thông tin nào cần lưu trữ và thông tin nào cần xóa, để mô hình có thể lưu trữ thông tin dài hạn.

## 2.3. Các kỹ thuật word embedding

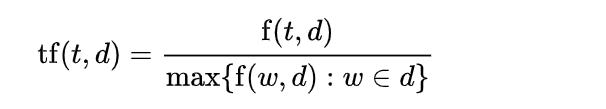
Word embedding là một kỹ thuật trong xử lý ngôn ngữ tự nhiên để biểu diễn các từ dưới dạng các vector số trong một không gian nhiều chiều. Không gian này bao gồm nhiều chiều và các từ trong không gian đó mà có cùng văn cảnh hoặc ngữ nghĩa sẽ có vị trí gần nhau. Ví dụ như ta có hai câu : "Hôm nay ăn táo " và "Hôm nay ăn xoài ". Khi ta thực hiện word embedding, "táo" và "xoài" sẽ có vị trí gần nhau trong không gian chúng ta biễu diễn do chúng có vị trị giống nhau trong một câu và đều có ý nghĩa là hoa quả.

### 2.3.1. TF-IDF

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kỹ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu.

* **TF là gì?**

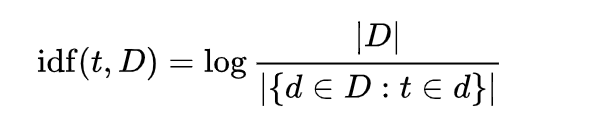
TF: Term Frequency(Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản( tổng số từ trong một văn bản).



Trong đó:

* tf(t, d): tần suất xuất hiện của từ t trong văn bản d
* f(t, d): Số lần xuất hiện của từ t trong văn bản d
* max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d
* IDF là gì?

IDF: Inverse Document Frequency(Tần suất ngược tài liệu), giúp đánh giá tầm quan trọng của một từ. Khi tính toán TF, tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như "và", "của", "là" thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.



Trong đó:

* idf(t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

Cơ số logarit trong công thức này không thay đổi giá trị của 1 từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Tuy nhiên việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng để dùng cho công thức TF-IDF như bên dưới.

Giá trị TF-IDF:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

IDF có ứng dụng quan trọng trong việc tìm kiếm. Ví dụ, khi người dùng gửi một truy vấn, hệ thống cần biết từ nào là từ người dùng quan tâm nhất. Chẳng hạn: truy vấn của người dùng là "làm thế nào để sửa máy ủi". Sau khi tách từ, chúng ta sẽ có tập các từ: làm, thế nào, để, sửa, máy ủi. Trong các từ này, "máy ủi" sẽ có IDF cao nhất. Hệ thống sẽ lấy ra tất cả các văn bản có chứa từ máy ủi và sau đó mới thực hiện việc đánh giá và so sánh dựa trên toàn bộ câu truy vấn.

### 2.3.2. Word2Vec

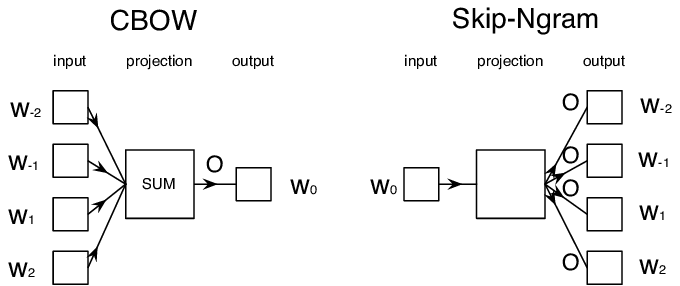
Word2vec là một kỹ thuật trong xử lý ngôn ngữ tự nhiên để thu thập các biểu diễn vectơ của từ. Các vectơ này nắm bắt thông tin về nghĩa của từ dựa trên các từ xung quanh. Thuật toán word2vec ước tính các biểu diễn này bằng cách mô hình hóa văn bản trong một ngữ liệu lớn. Sau khi được đào tạo, một mô hình như vậy có thể phát hiện các từ đồng nghĩa hoặc gợi ý các từ bổ sung cho một câu không trọn vẹn. Word2vec được phát triển bởi Tomáš Mikolov và các đồng nghiệp tại Google và được xuất bản vào năm 2013

.

Hình 2.7. Mô hình từ nhúng.

Có hai kiến ​​trúc Word2Vec chính được sử dụng để biểu diễn phân tán các từ:

* Continuous bag-of-words (CBOW) – Thứ tự của các từ ngữ cảnh không ảnh hưởng đến dự đoán (giả định bag-of-words) (theo Aaron (Ari) Bornstein trên Towards Data Science). Trong kiến ​​trúc continuous skip-gram, mô hình sử dụng từ hiện tại để dự đoán phạm vi xung quanh của các từ ngữ cảnh.
* Continuous skip-gram chú trọng các từ ngữ cảnh ở phạm vi gần. Mỗi vectơ ngữ cảnh được cân nhắc và so sánh độc lập với CBOW.



Hình 2.8. Mô hình CBOW và Skip-Ngram.

**CBOW:** Dự đoán từ ngữ hiện tại dựa trên ngữ cảnh của các từ trước đó.

* Cho các từ ngữ cảnh
* Đoán xác suất của 1 từ đích

**Skip-Ngram:** Dự đoán từ các từ xung quanh dựa trên từ ngữ hiện tại.

* Cho từ đích
* Đoán xác suất của các từ ngữ cảnh

### 2.3.3. PhoBert

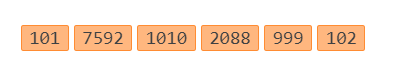
Các mô hình PhoBERT được đào tạo trước là mô hình ngôn ngữ tiên tiến dành cho người Việt Nam (Pho, tức là "Phở", là một món ăn phổ biến ở Việt Nam). Hai phiên bản PhoBERT “base” và “large” là mô hình ngôn ngữ đơn ngữ quy mô lớn đầu tiên được đào tạo trước cho tiếng Việt, nó được dựa trên kiến trúc của BERT “base” và BERT ”large”. Phương pháp đào tạo trước PhoBERT dựa trên RoBERTa giúp tối ưu hóa quy trình đào tạo trước BERT để có hiệu suất mạnh mẽ hơn. [3]

Để biểu diễn dữ liệu đầu vào dạng văn bản, BERT dựa vào 3 loại nhúng riêng biệt: Nhúng mã thông báo (Token Embeddings), Nhúng vị trí (Position Embeddings) và Nhúng loại mã thông báo (Token Type Embeddings).

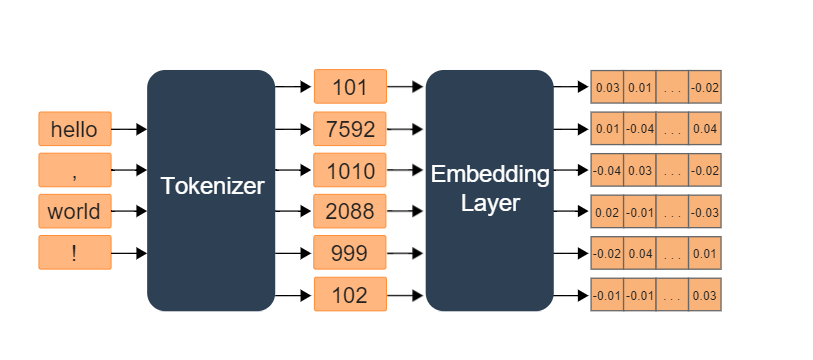
* **Token Embeddings**

Trước khi một chuỗi văn bản được chuyển đến mô hình BERT, BERT Tokenizer được sử dụng để chuyển đổi dữ liệu đầu vào từ một chuỗi thành danh sách các ID mã thông báo số nguyên, trong đó mỗi ID được ánh xạ trực tiếp đến một từ hoặc một phần của từ trong chuỗi gốc.

Ví dụ: chuỗi "hello, world!" được tokenizer chuyển đổi thành các ID mã thông báo sau:



Hình 2.9. Chuỗi được mã hóa thành mã thông báo.

Đối với mỗi ID mã thông báo duy nhất (tức là đối với mỗi trong số 30.522 từ và từ phụ trong từ vựng của trình mã thông báo BERT), mô hình BERT chứa phần nhúng được đào tạo để thể hiện mã thông báo cụ thể đó. Lớp nhúng trong mô hình chịu trách nhiệm ánh xạ mã thông báo tới các phần nhúng tương ứng của chúng.Hình 2.10. Hình biểu diễn vector nhúng của chuỗi qua các bước.

Thuộc tính bert.embeddings của đối tượng mô hình có thể được sử dụng để xem và thực thi lớp nhúng của chính nó. Thuộc tính weight của lớp chứa các phần nhúng mã thông báo (tức là các phần nhúng cho mỗi mã thông báo trong từ vựng của trình mã thông báo BERT).

**>>>** embedding\_layer **=** model**.**bert**.**embeddings

**>>>** print**(**embedding\_layer**.**weight**)**

**<**tf**.**Variable 'tf\_bert\_model/bert/embeddings/word\_embeddings/weight:0' shape**=(30522,** **768)** dtype**=**float32**,** numpy**=**

array**([[-0.01018257,** **-0.06154883,** **-0.02649689,** **...,** **-0.01985357,**

**-0.03720997,** **-0.00975152],**

**[-0.01170495,** **-0.06002603,** **-0.03233192,** **...,** **-0.01681456,**

**-0.04009988,** **-0.0106634** **],**

**[-0.01975381,** **-0.06273633,** **-0.03262176,** **...,** **-0.01650258,**

**-0.04198876,** **-0.00323178],**

**...,**

**[-0.02176224,** **-0.0556396** **,** **-0.01346345,** **...,** **-0.00432698,**

**-0.0151355** **,** **-0.02489496],**

**[-0.04617237,** **-0.05647721,** **-0.00192082,** **...,** **0.01568751,**

**-0.01387033,** **-0.00945213],**

**[** **0.00145601,** **-0.08208051,** **-0.01597912,** **...,** **-0.00811687,**

**-0.04746607,** **0.07527421]],** dtype**=**float32**)>**

Các phần nhúng được trả về dưới dạng ma trận 30522 x 768 hoặc tensor 2 chiều:

* Kích thước đầu tiên của tensor này là kích thước từ vựng của mã thông báo BERT: 30.522
* Số thứ hai là kích thước nhúng, còn được gọi là kích thước ẩn. Đây là số lượng trọng số có thể huấn luyện cho mỗi mã thông báo trong từ vựng. Mô hình BERT ban đầu có kích thước ẩn là 768, nhưng các biến thể khác của BERT đã được huấn luyện với các giá trị nhỏ hơn và lớn hơn của kích thước ẩn.
* **Position Embeddings**

Ngoài các phần nhúng mã thông báo được mô tả cho đến nay, BERT còn dựa vào các phần nhúng vị trí. Trong khi nhúng mã thông báo được sử dụng để thể hiện từng từ hoặc từ phụ có thể được cung cấp cho mô hình, thì nhúng vị trí thể hiện vị trí của từng mã thông báo trong chuỗi đầu vào.

****

Hình 2.11. Hình chuỗi được đánh ký hiệu vị trí position.

Thuộc tính position\_embeddings của lớp nhúng được sử dụng để truy cập vào Vị trí nhúng:

**>>>** print**(**embedding\_layer**.**position\_embeddings**)**

**<**tf**.**Variable 'tf\_bert\_model/bert/embeddings/position\_embeddings/embeddings:0' shape**=(512,** **768)** dtype**=**float32**,** numpy**=**

array**([[** **1.7505383e-02,** **-2.5631009e-02,** **-3.6641564e-02,** **...,**

**3.3437202e-05,** **6.8312453e-04,** **1.5441139e-02],**

**[** **7.7580423e-03,** **2.2613001e-03,** **-1.9444324e-02,** **...,**

**2.8909724e-02,** **2.9752752e-02,** **-5.3246655e-03],**

**[-1.1287465e-02,** **-1.9644140e-03,** **-1.1572698e-02,** **...,**

**1.4907907e-02,** **1.8740905e-02,** **-7.3139993e-03],**

**...,**

**[** **1.7417932e-02,** **3.4902694e-03,** **-9.5621375e-03,** **...,**

**2.9599327e-03,** **4.3434653e-04,** **-2.6948910e-02],**

**[** **2.1686664e-02,** **-6.0216337e-03,** **1.4735642e-02,** **...,**

**-5.6118402e-03,** **-1.2589799e-02,** **-2.8084971e-02],**

**[** **2.6412839e-03,** **-2.3297865e-02,** **5.4921862e-03,** **...,**

**1.7536936e-02,** **2.7549751e-02,** **-7.7655964e-02]],** dtype**=**float32**)>**

Trong khi có 30.522 Token Embedding khác nhau, chỉ có 512 Position Embedding khác nhau. Điều này là do chuỗi đầu vào lớn nhất được mô hình BERT chấp nhận dài 512 token.

* **Token Type Embeddings**

Kiểu nhúng cuối cùng được BERT sử dụng là nhúng loại mã thông báo, còn được gọi là nhúng phân đoạn (Segment Embedding) trong BERT paper. Một trong những nhiệm vụ mà BERT ban đầu được đào tạo để giải quyết là dự đoán câu tiếp theo. Nghĩa là, cho hai câu A và B, BERT được huấn luyện để xác định xem B có tuân theo A một cách logic hay không.

Thuộc tính token\_type\_embeddings của đối tượng lớp nhúng được sử dụng để truy cập vào phần nhúng loại mã thông báo:

**>>>** print**(**embedding\_layer**.**token\_type\_embeddings**)**

**>>>** **<**tf**.**Variable 'tf\_bert\_model/bert/embeddings/token\_type\_embeddings/embeddings:0' shape**=(2,** **768)** dtype**=**float32**,** numpy**=**

array**([[** **0.00043164,** **0.01098826,** **0.00370439,** **...,** **-0.00661185,**

**-0.00336983,** **-0.00864201],**

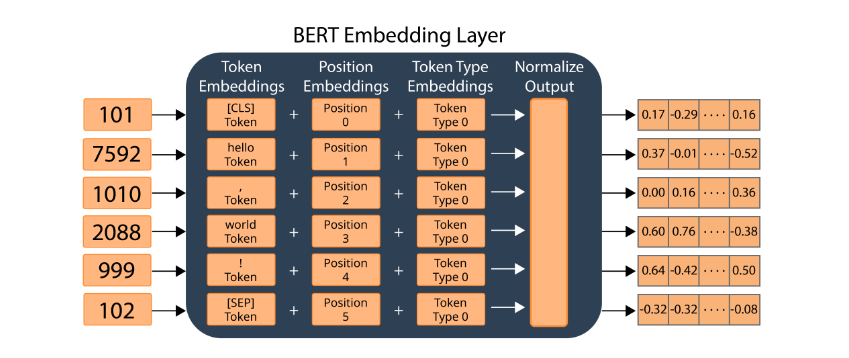
**[** **0.00111319,** **-0.00299169,** **-0.00317028,** **...,** **0.00474542,**

**-0.0052443** **,** **-0.01121742]],** dtype**=**float32**)>**

Chỉ có hai Nhúng loại mã thông báo khác nhau: một được sử dụng để biểu diễn các mã thông báo trong câu A và một được sử dụng để biểu diễn các mã thông báo trong câu B.

**BERT Embedding Layer**

Ba vectơ nhúng được thêm vào cùng nhau biểu thị biểu diễn mã thông báo ban đầu dưới dạng hàm của ba phần thông tin này. Sau khi nhúng, biểu diễn vectơ được chuẩn hóa bằng thao tác LayerNorm, xuất ra vectơ 768 chiều cho mỗi mã thông báo đầu vào. Sau đó, các vectơ biểu diễn được truyền tiếp qua 12 khối mã hóa Transformer và được giải mã trở lại không gian từ vựng 30.000 chiều bằng cách sử dụng lớp biến đổi affine cơ bản. Hình minh họa bên dưới cho thấy cách Lớp nhúng BERT tính toán việc nhúng cho chuỗi "hello, world!"



Hình 2.12. Hình mô tả lớp nhúng của mô hình BERT.

Ngoài ra, BERT còn được trang bị cơ chế attention, nó có vai trò rất quan trọng trong việc nắm bắt mối quan hệ giữa các từ trong câu mà không phụ thuộc vào khoảng cách của chúng trong chuỗi. Nó còn giúp BERT hiểu được ngữ nghĩa phức tạp cũng như nắm bắt bối cảnh toàn cục. Cơ chế này là một phần quan trọng giúp BERT đạt được hiệu quả cao trong việc xử lý ngôn ngữ tự nhiên, từ các tác vụ như phân loại văn bản, trả lời câu hỏi cho đến phân tích ngữ nghĩa.

PhoBERT có kiến trúc tương tự như BERT, nhưng được điều chỉnh đặc biệt để phù hợp với ngôn ngữ tiếng Việt. PhoBERT được huấn luyện trên một tập dữ liệu lớn gồm 20GB văn bản tiếng Việt, bao gồm các bài viết từ Wikipedia tiếng Việt, các bài báo, và nhiều nguồn dữ liệu phong phú khác. Sự khác biệt chính giữa PhoBERT và BERT là PhoBERT tối ưu hóa cho các đặc trưng ngôn ngữ tiếng Việt, giúp mô hình xử lý tốt hơn các tác vụ ngôn ngữ tự nhiên liên quan đến tiếng Việt.

## 2.4. Các phương pháp tính độ tương đồng

### 2.4.1. Độ tương đồng jaro

**Độ tương đồng jaro là gì?**

Độ tương đồng jaro là một phép đo khoảng cách giữa hai chuỗi ký tự, thường được sử dụng trong các bài toán so khớp văn bản. Nó đặc biệt hiệu quả với các chuỗi có độ dài ngắn, như tên hoặc từ vựng, và có thể được dùng để xác định sự tương đồng giữa các chuỗi có chứa lỗi chính tả hoặc sai lệch nhỏ.

Độ tương đồng jaro được tính bằng công thức sau:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Trong đó:

* m: là số ký tự khớp nhau
* t: là một nửa số lần chuyển vị để các ký tự khớp nhau
* |s1| và |s2| lần lượt là độ dài của chuỗi

Ưu điểm:

* Hiệu quả với chuỗi ngắn: Độ tương đồng jaro được thiết kế để xử lý tốt với các chuỗi ngắn, đặc biệt là trong so sánh tên hoặc từ có lỗi nhỏ.
* Chấp nhận sai lệch: Phương pháp này có thể xử lý tốt khi có sai sót nhỏ về thứ tự hoặc sai chính tả, điều này phù hợp trong các bài toán so sánh văn bản thực tế.

Nhược điểm:

* Không hiệu quả với chuỗi dài: Khi chuỗi trở nên dài hơn, độ tương đồng Jaro không hoạt động hiệu quả vì phương pháp này nhấn mạnh vào sự tương đồng giữa các ký tự trong phạm vi ngắn.
* Không quan tâm đến ngữ nghĩa: Jaro chỉ dựa trên sự giống nhau về ký tự mà không quan tâm đến ngữ nghĩa hoặc cấu trúc của văn bản, điều này có thể dẫn đến các kết quả sai lệch trong các văn bản phức tạp.

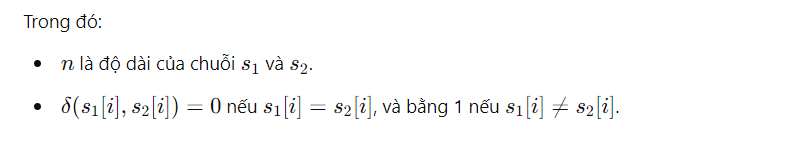
### 2.4.2. Khoảng cách hamming

**Khoảng cách hamming là gì?**

Khoảng cách hamming (Hamming distance) là một phép đo sự khác biệt giữa hai chuỗi ký tự có cùng độ dài, bằng cách đếm số lượng vị trí mà các ký tự tại đó khác nhau. Khoảng cách Hamming được ứng dụng rộng rãi trong các lĩnh vực như mã hóa, truyền dữ liệu và nhận diện lỗi.

Công thức tính khoảng cách Hamming giữa hai chuỗi ký tự s1​ và s2​:

|  |  |  |
| --- | --- | --- |
|  |  |  |



Ưu điểm:

* Tính đơn giản: Cách tính toán của khoảng cách Hamming rất đơn giản, chỉ cần đếm số lượng ký tự khác nhau giữa hai chuỗi.
* Hiệu quả tính toán: Do không cần tính toán phức tạp, khoảng cách hamming có tốc độ xử lý nhanh, phù hợp với các bài toán yêu cầu thời gian thực hoặc khối lượng tính toán lớn.
* Ứng dụng hiệu quả trong mã hóa và truyền dữ liệu: Khoảng cách hamming được sử dụng nhiều trong lĩnh vực phát hiện và sửa lỗi trong truyền thông dữ liệu và mã hóa.

Nhược điểm:

* Chỉ áp dụng cho chuỗi có cùng độ dài: Khoảng cách Hamming chỉ có thể được tính khi hai chuỗi có độ dài bằng nhau. Điều này làm hạn chế ứng dụng của nó trong các bài toán mà các chuỗi so sánh có thể có độ dài khác nhau.
* Không hiệu quả với sự khác biệt lớn: Khoảng cách Hamming không phản ánh tốt mức độ tương đồng của các chuỗi có nhiều thay đổi nhỏ về vị trí, như việc hoán đổi thứ tự ký tự.
* Không quan tâm đến ngữ nghĩa: Giống như jaro, hamming cũng không xem xét đến ý nghĩa hoặc ngữ cảnh của văn bản, chỉ tập trung vào sự khác biệt vị trí của các ký tự.

### 2.4.3. Khoảng cách euclid

**Khoảng cách euclid là gì?**

Khoảng cách euclid là thước đo cho biết khoảng cách giữa hai điểm bất kỳ trong mặt phẳng n chiều. Đây là thước đo khoảng cách phổ biến và quen thuộc nhất.

Công thức tính khoảng cách euclid giữa 2 điểm trong không gian n chiều:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Trong đó:

* d(A,B) là khoảng cách euclid
* i chạy từ 1 đến n
* ai là tọa độ tương ứng của điểm A
* bi là tọa độ tương ứng của điểm B

Ưu điểm:

* Tính toán đơn giản: Công thức của khoảng cách Euclid đơn giản và dễ thực hiện, đặc biệt với dữ liệu đã được biểu diễn dưới dạng vector số.
* Hiệu quả khi dữ liệu được chuẩn hóa: Khoảng cách Euclid có thể hoạt động tốt khi dữ liệu được chuẩn hóa về cùng một không gian, ví dụ các vector nhúng có độ dài đồng đều.
* Phổ biến trong các mô hình vector: Khi văn bản được biểu diễn dưới dạng vector như trong mô hình bag-of-words, TF-IDF hoặc vector nhúng từ (word embeddings), khoảng cách Euclid là một trong những cách phổ biến để tính toán sự tương đồng giữa các tài liệu.

Nhược điểm:

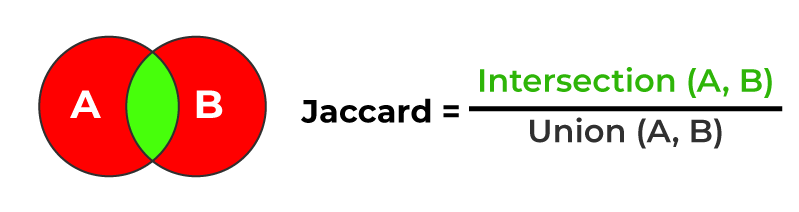
* Không hiệu quả khi số chiều lớn (curse of dimensionality): Trong không gian có số chiều lớn, chẳng hạn như khi biểu diễn văn bản bằng các vector có nhiều thành phần (như trong TF-IDF), khoảng cách Euclid có thể mất hiệu quả vì các điểm trở nên rất xa nhau, gây khó khăn cho việc phân biệt sự tương đồng.
* Nhạy cảm với độ lớn của vector: Nếu hai vector không được chuẩn hóa, khoảng cách Euclid sẽ bị ảnh hưởng bởi độ lớn của các vector, dẫn đến việc các văn bản có nhiều từ xuất hiện sẽ có khoảng cách lớn với các văn bản ngắn hơn, mặc dù ngữ nghĩa có thể tương đồng.

### 2.4.4. Độ tương đồng jaccard

**Độ tương đồng jaccard là gì?**

Độ tương đồng Jaccard hay còn gọi là chỉ số Jaccard, là một thống kê để đo lường mức độ tương tự giữa hai bộ dữ liệu. Nó được đo bằng kích thước giao (intersection) của hai tập hợp chia cho kích thước hợp (union) của chúng.

Ví dụ: Cho hai bộ A và B, Độ tương đồng jaccard được tính bằng



Hình 2.13. Hình biểu diễn độ tương đồng jaccard

Hay công thức:

|  |  |  |
| --- | --- | --- |
|  |  |  |

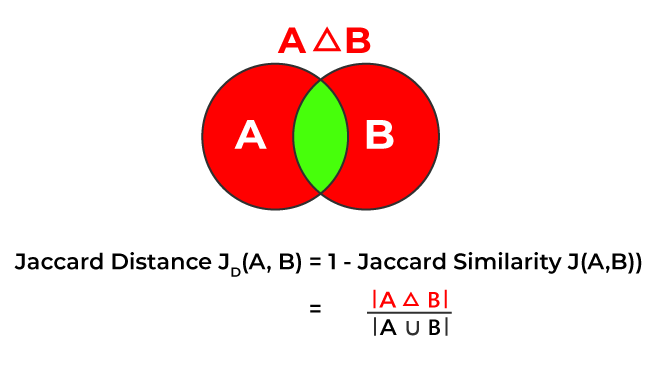
Trong đó:

* là số lượng phần tử (kích thước) của giao hai tập hợp A và B.
* là số lượng phần tử (kích thước) của hợp hai tập hợp A và B.

Các giá trị của nó nằm trong khoảng từ 0 đến 1. Trong đó 0 nghĩa là không có sự tương đồng và các giá trị càng gần 1 nghĩa là độ tương tự càng tăng. 1 nghĩa là các tập dữ liệu giống nhau.

**Khoảng cách Jaccard**

Khoảng cách jaccard là thước đo mức độ khác nhau của hai bộ. Tức là không giống như hệ số jaccard xác định mức độ giống nhau của hai bộ. Khoảng cách jaccard được tính bằng cách lấy 1 trừ hệ số jaccard.



Hình 2.14. Hình biểu diễn khoảng cách jaccard

Ưu điểm:

* Đơn giản và dễ tính toán: Độ tương đồng jaccard sử dụng các phép toán cơ bản trên tập hợp, giúp dễ dàng hiểu và triển khai, đặc biệt trong việc so sánh các tập hợp từ.
* Không nhạy cảm với số lượng từ trong chuỗi: Vì chỉ dựa trên số lượng phần tử chung và tổng số phần tử, độ tương đồng jaccard không bị ảnh hưởng bởi chiều dài của văn bản, mà chỉ tập trung vào sự tương quan giữa các từ.

Nhược điểm:

* Không tốt với dữ liệu nhiều chiều: Khi áp dụng trên các vector có nhiều chiều (chẳng hạn văn bản dài hoặc nhiều thuộc tính)
* Thiếu khả năng xử lý văn bản ngữ nghĩa: Hệ số Jaccard chỉ xét các từ chung giữa các chuỗi, không tính đến mối liên hệ ngữ nghĩa
* Nhạy cảm với từ đồng nghĩa: Các chuỗi sử dụng từ đồng nghĩa sẽ không được tính là tương tự nhau mặc dù về nghĩa có thể giống nhau.

### 2.4.5. Độ tương đồng cosine

Độ tương đồng cosine là thước đo độ tương tự giữa hai vectơ khác 0 được áp dụng rộng rãi trong nhiều ứng dụng phân tích dữ liệu và học máy. Nó thực sự đo cosin của góc giữa hai vectơ. Kết quả là, một ý tưởng được đưa ra về khoảng cách hai vectơ bất kể độ lớn của chúng. Nó có thể được sử dụng phổ biến trong các nhiệm vụ phân tích văn bản, chẳng hạn như so sánh sự giống nhau giữa các tài liệu, truy vấn tìm kiếm và thậm chí cả hệ thống đề xuất để có thể khớp với sở thích của người dùng.

**Độ tương đồng cosine là gì?**

Độ tương đồng cosine là một thước đo hữu ích trong việc xác định mức độ giống nhau của các đối tượng dữ liệu bất kể kích thước của chúng. Trong độ tương đồng cosine, các đối tượng dữ liệu trong tập dữ liệu được coi là một vectơ.

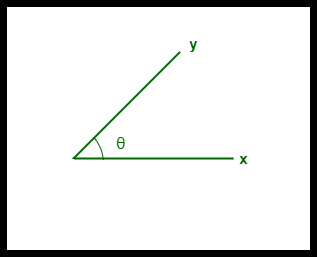
Công thức tìm độ tương đồng cosin giữa hai vectơ là

|  |  |  |
| --- | --- | --- |
|  |  |  |

Trong đó 𝐴i và 𝐵i lần lượt là các 𝑖 thành phần thứ của vectơ 𝐴 và 𝐵 .

Độ tương đồng có kết quả được dao động từ -1 có nghĩa là hoàn toàn đối lập, đến 1 có nghĩa là hoàn toàn giống nhau, với 0 biểu thị tính trực giao hoặc không tương quan, trong khi các giá trị ở giữa biểu thị độ tương tự hoặc độ khác nhau trung gian.

Trong so khớp văn bản, các vectơ thuộc tính A và B thường là các vectơ tần số thuật ngữ của tài liệu. Độ tương tự cosine có thể được coi là một phương pháp chuẩn hóa độ dài tài liệu trong quá trình so sánh. Trong trường hợp truy xuất thông tin, độ tương tự cosine của hai tài liệu sẽ nằm trong khoảng từ [0,1], vì tần số thuật ngữ không thể âm. Điều này vẫn đúng khi sử dụng trọng số TF-IDF. Góc giữa hai vectơ tần số không được lớn hơn 90°.



Hình 2.15. Hình biểu diễn độ tương đồng cosine giữa 2 vector.

* Độ tương tự cosine giữa hai vectơ được đo bằng góc 'θ'.
* Nếu θ = 0°, vectơ ‘x’ và ‘y’ trùng nhau, do đó chứng tỏ chúng giống nhau.
* Nếu θ = 90°, vectơ ‘x’ và ‘y’ là không giống nhau.

Ưu điểm:

* Độ tương đồng cosin có lợi vì ngay cả khi hai đối tượng dữ liệu tương tự cách xa nhau theo khoảng cách Euclidean do kích thước, chúng vẫn có thể có góc nhỏ hơn giữa chúng. Góc càng nhỏ, độ tương đồng càng cao.
* Khi biểu diễn trên không gian đa chiều, độ tương đồng cosin sẽ nắm bắt được hướng (góc) của các đối tượng dữ liệu chứ không phải độ lớn.

Nhược điểm:

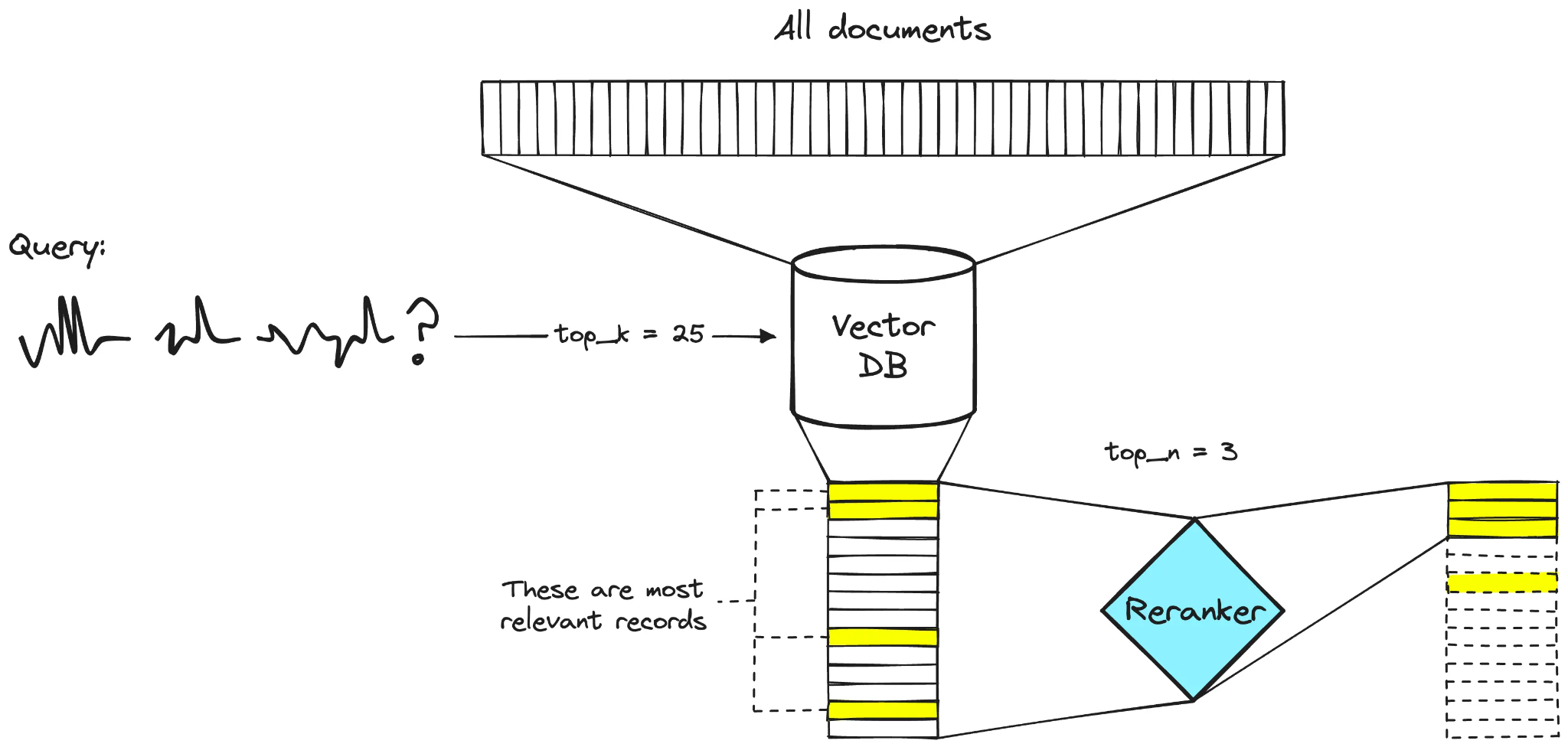
* Không tính đến sự khác biệt tuyệt đối: Độ tương đồng cosin chỉ xem xét góc giữa các vectơ chứ không phải độ lớn của chúng; do đó, điều này có thể bỏ qua sự khác biệt về độ lớn, trong một số bối cảnh nhất định, có thể có liên quan.

Với các ưu, nhược điểm của các phương pháp tính độ tương đồng trên, đồ án sử dụng kết hợp giữa độ tương đồng cosine và độ tương đồng jaccard với các tham số phù hợp để tính toán độ tương đồng giữa hai câu khi mà hai câu đấy đã được chuyển thành vector nhúng.

## 2.5. Xếp hạng lại

**Thế nào là xếp hạng lại?**

Mô hình xếp hạng lại - còn được gọi là bộ mã hóa chéo (cross-encoder) - là một loại mô hình, khi đưa ra một cặp truy vấn và tài liệu, sẽ đưa ra điểm tương đồng. Chúng sử dụng điểm này để sắp xếp lại các tài liệu theo mức độ liên quan đến truy vấn.



Hình 2.16. Hình biểu diễn các giai đoạn khi xếp hạng lại.

Trong hình trên bao gồm hai giai đoạn, giai đoạn đầu là giai đoạn sử dụng mô hình nhúng/truy xuất để truy xuất một tập hợp các tài liệu có liên quan từ một tập dữ liệu lớn hơn. Sau đó, giai đoạn thứ hai là giai đoạn xếp hạng lại được sử dụng các mô hình để xếp hạng lại các tài liệu được truy xuất trong giai đoạn đầu.

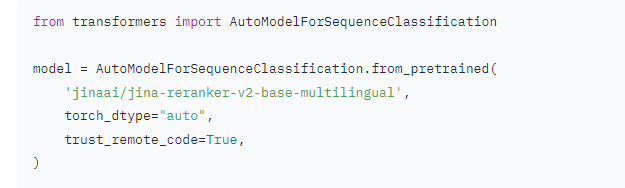
**Tại sao lại là xếp hạng lại?**

Mặc dù xếp hạng lại xử lý chậm, tốn nhiều thời gian hơn nhiều, nhưng vẫn được sử dụng là vì nó mang lại độ chính xác cao hơn so với mô hình nhúng.

Với bộ mã hóa kép (bi-encoder), mô hình phải nén toàn bộ nội dung của một tài liệu vào một vector duy nhất, nên chắc chắn sẽ bị mất đi một phần thông tin. Ngoài ra, bộ mã hóa kép không có ngữ cảnh của truy vấn từ trước, bởi vì ta tạo ra nhúng trước khi biết được truy vấn cụ thể của người dùng.

**Cách sử dụng Jina Reranker**

Sử dụng thư viện transformer



## 2.6. Kết luận chương

Chương này giới thiệu một số kiến thức nền tảng về học sâu, cũng như mạng nơ-ron nhân tạo, mạng nơ ron hồi quy, cung cấp cái nhìn khái quát về nền tảng kỹ thuật sử dụng trong chatbot. Các kỹ thuật nhúng từ, bao gồm TF-IDF, Word2Vec và PhoBERT. Bên cạnh đó, các phương pháp tính độ tương đồng cũng được đề cập để đánh giá mức độ tương đồng giữa câu truy vấn và các câu hỏi tiềm năng. Cuối cùng, xếp hạng lại giúp nâng cao độ chính xác trong việc lựa chọn câu trả lời phù hợp nhất cho người dùng. Các kiến thức này là nền tảng quan trọng để phát triển chatbot phục vụ tuyển sinh trong những chương tiếp theo.

# CHƯƠNG 3: XÂY DỰNG CHATBOT TƯ VẤN TUYỂN SINH

## 3.1. Phát biểu bài toán

Có rất nhiều bài toán ứng dụng cho chatbot nhưng tôi chọn bài toán về vấn đề tuyển sinh để giải quyết những khó khăn trong việc tư vấn của Học Viện, từ đó nhằm mang lại sự thuận tiện và hiệu quả trong việc giải đáp thông tin.

Bài toán được xây dựng tập trung vào các chức năng tư vấn và hỏi đáp trong việc tuyển sinh. Các chức năng chính bao gồm:

* Thông tin về các ngành đào tạo như điểm chuẩn, học phí, chỉ tiêu, mã ngành, xét tuyển theo tổ hợp nào
* Thông tin về các phương thức xét tuyển, hồ sơ nhập học
* Thông tin về các loại học bổng
* Các ngành học mới
* Một số câu hỏi thường gặp về Học Viện.
* Thu thập phản hồi người dùng

## 3.2. Xây dựng dữ liệu cho chatbot

Nguồn dữ liệu được thu thập qua các website chính quy của học viện.

* Thông tin chi tiết về tuyển sinh năm 2024 được lấy từ website <https://tuyensinh.ptit.edu.vn/>, dữ liệu bao gồm các thông tin như phương thức xét tuyển, quy trình xét tuyển, mốc thời gian quan trọng, chỉ tiêu, học bổng…
* Thông tin tổng quan về các chuyên ngành đào tạo của học viện được lấy từ website <https://daotao.ptit.edu.vn/>, dữ liệu bao gồm các thông tin như tổng quan về ngành, tên ngành, mã ngành, học phí, thời gian học, kỳ nhập học, có ở cả 2 cơ sở BVH và BVS không…
* Thông tin về điểm chuẩn các phương thức như thi THPT, ĐGNL QG HN, ĐGNL QG HCM, ĐG tư duy ĐHBKHN các năm 2024, 2023, 2022, 2021 được thu thập từ các website <https://diemthi.tuyensinh247.com/>, <https://portal.ptit.edu.vn/hoc-vien-cong-nghe-buu-chinh-vien-thong-thong-bao-guong-diem-xet-du-dieu-kien-trung-tuyen-vao-dai-hoc-he-chinh-quy-nam-2023-theo-cac-phuong-thuc-xet-tuyen-som/> và <https://thptquocgia.org/diem-chuan-xet-tuyen-ket-hop-hv-cong-nghe-buu-chinh-vien-thong-nam-2021>

Dữ liệu sau khi thu thập từ các website sẽ được lưu dưới dạng thô như file csv hoặc file json. Ví dụ như thông tin về ngành công nghệ thông tin được lưu dưới dạng file json.

{

"titleMajor": "Ngành Công nghệ thông tin",

"maNganh": "7480201",

"thoigianHoc": "4,5 năm",

"kyNhapHoc": "Mùa thu",

"coSo": "Hà Nội và Tp Hồ Chí Minh",

"hocPhi": "Học phí theo tín chỉ năm 2024: 880.000 đ/tín chỉ."

}

Sau đấy, các dữ liệu thô ở trên sẽ được trải qua giai đoạn tiền xử lý dữ liệu và chuyển về dạng câu hỏi và câu trả lời (Q&A form) và lưu lại dưới dạng file csv. Ví dụ

Q: Học phí của Ngành An toàn thông tin là bao nhiêu?

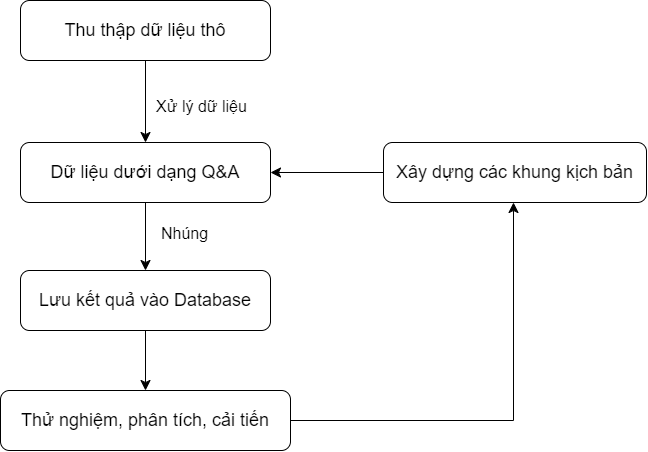
A: Ngành An toàn thông tin có học phí theo tín chỉ năm 2024: 880.000 đ/tín chỉ.  
 Xây dựng các khung kịch bản có thể có. Ví dụ dưới đây là đoạn hội thoại của người dùng khi hỏi chatbot về điểm chuẩn ngành công nghệ thông tin năm nay.

|  |  |
| --- | --- |
| **Người dùng** | **Chatbot** |
| Điểm chuẩn ngành công nghệ thông tin năm nay như nào? | Bạn muốn hỏi điểm chuẩn dựa trên phương thức xét tuyển nào, cơ sở miền bắc hay nam? Ví dụ về phương thức xét tuyển như xét tuyển kết hợp, thi thpt, đánh giá năng lực, đánh giá tư duy. |
| miền bắc, thi đánh giá năng lực đi | Điểm chuẩn ngành công nghệ thông tin theo phương thức thi đánh giá năng lực năm 2024 tại cơ sở miền bắc là 22.55 |
| miền nam, thi trung học phổ thông thì sao | Điểm chuẩn ngành công nghệ thông tin theo phương thức thi thpt năm 2024 tại cơ sở miền nam là 25.17 |
| oke tôi cảm ơn | Tôi rất vui khi có thể giúp được bạn. Nếu bạn còn câu hỏi hay thắc mắc nào thì hãy cứ hỏi nhé. Chúc bạn một ngày tốt lành! 💖 |

Bảng 3.1. Kịch bản về điểm chuẩn ngành công nghệ thông tin.

Sau khi xây dựng các khung kịch bản và dữ liệu dạng câu hỏi, câu trả lời xong, thì dùng PhoBERT để chuyển các câu hỏi thành vector nhúng và lưu tất cả chúng vào cơ sở dữ liệu (MongoDB) để thuận tiện cho việc lưu trữ và truy xuất kết quả.

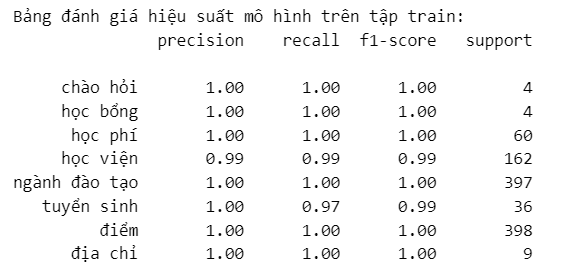
Tóm lại, quá trình xây dựng dữ liệu cho chatbot sẽ được thực hiện các bước chính như sau:



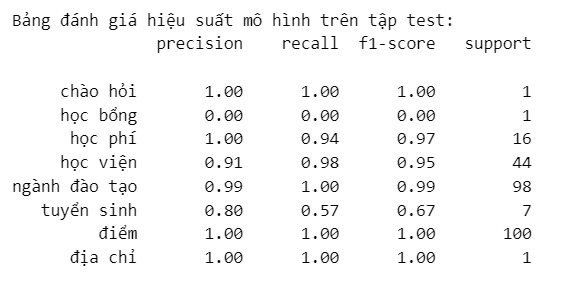
Hình 3.1. Quá trình xây dựng dữ liệu cho chatbot.

## 3.3. Thực nghiệm và kết quả

Tiến hành thử nghiệm tương tác với chatbot để xác định chủ đề của câu hỏi thì thấy độ chính xác của chatbot gần 100% và chỉ tính các câu hỏi xong quanh kịch bản đã đào tạo cho chatbot. Dưới đây là bảng đánh giá chi tiết hiệu suất mô hình dự đoán chủ đề.



Hình 3.2. Bảng đánh giá hiệu suất mô hình trên tập dữ liệu huấn luyện.



Hình 3.3. Bảng đánh giá hiệu suất mô hình trên tập dữ liệu kiểm tra.

## 3.4. Cài đặt ứng dụng

### 3.4.1. Môi trường phát triển

Môi trường cài đặt: hệ điều hành Window, website tương thích với trình duyệt web Cốc Cốc, Google Chrome và Mozilla Firefox.

Môi trường lập trình: hệ điều hành Window 11, RAM 8GB, Intel(R) Core(TM) i5-1135G7 CPU @ 2.40GHz, 2.42 GHz.

Trình soạn thảo mã nguồn: Visual Studio Code, Google Colab.

Kho lưu trữ mã nguồn: Github

### 3.4.2. Các công nghệ sử dụng

**a) React**

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, với mục tiêu xây dựng giao diện người dùng một cách hiệu quả và linh hoạt. Thư viện này chủ yếu tập trung vào việc quản lý và hiển thị các phần giao diện của ứng dụng, giúp bạn dễ dàng tạo ra các thành phần (component) giao diện và cập nhật chúng khi dữ liệu thay đổi. Để tối ưu hóa việc cập nhật giao diện, React sử dụng một cấu trúc dữ liệu được gọi là Virtual DOM. Thay vì cập nhật trực tiếp DOM, điều có thể gây tốn kém về hiệu năng, React sẽ so sánh Virtual DOM với DOM thực tế và chỉ thực hiện những thay đổi cần thiết, từ đó nâng cao hiệu suất hoạt động của ứng dụng.

**b) Tailwind**

Tailwind là một framework CSS tiện lợi giúp xây dựng giao diện web nhanh chóng và dễ dàng bằng cách cung cấp các lớp tiện ích sẵn có. Không như các framework khác, Tailwind không bắt buộc cấu trúc giao diện cố định mà cho phép tùy chỉnh linh hoạt. Các lớp trong Tailwind có thể được kết hợp để tạo nên bất kỳ giao diện nào bạn muốn mà không cần viết CSS từ đầu, nhờ đó tăng tốc độ phát triển giao diện và dễ dàng duy trì mã nguồn.

**c) Python**

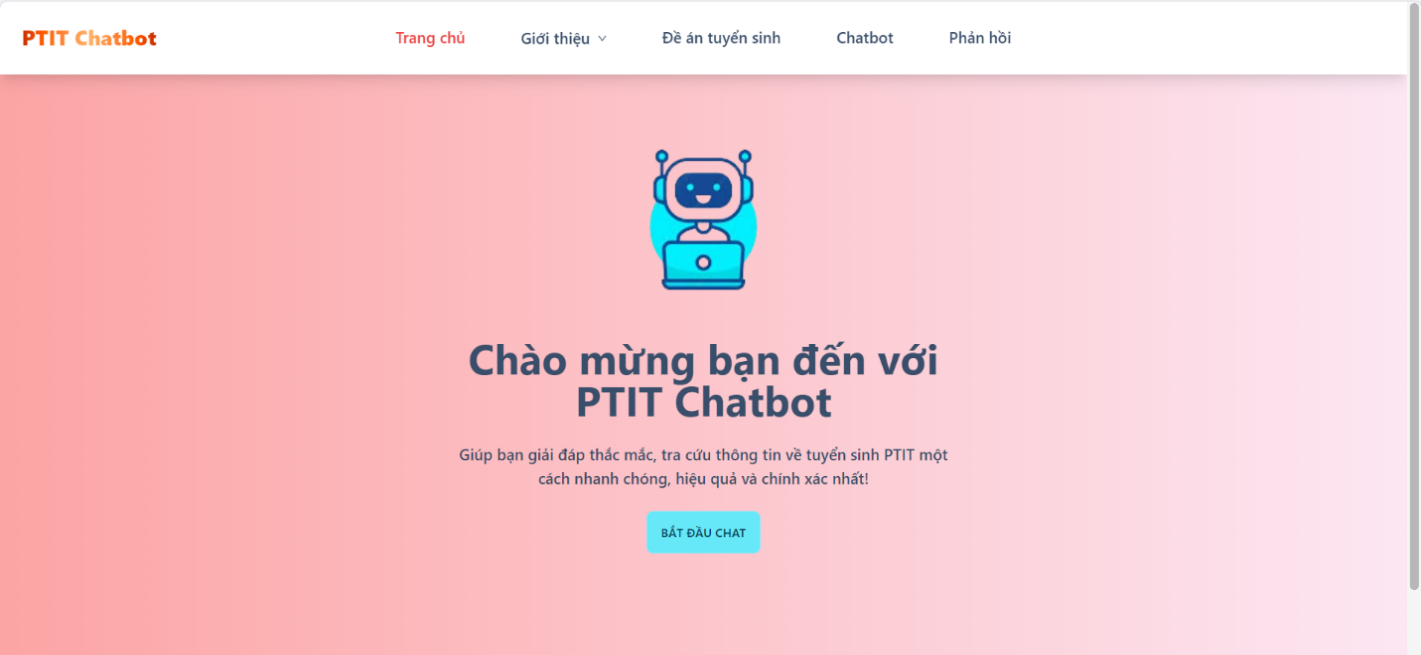
Python là ngôn ngữ lập trình mạnh mẽ và linh hoạt được sử dụng rộng rãi trong lĩnh vực xây dựng chatbot, với các thư viện hỗ trợ nhiều khía cạnh của quy trình. Flask là một framework của Python, giúp xây dựng API nhanh chóng để giao tiếp giữa frontend và backend. Để thu thập dữ liệu, Selenium và Beautiful Soup là những thư viện hữu ích trong việc crawl dữ liệu từ các nguồn web, cung cấp thông tin cần thiết cho chatbot. Ngoài ra, còn một số thư viện liên quan đến xử lý ngôn ngữ tự nhiên của Python, từ đó giúp chatbot hiểu và phản hồi người dùng một cách chính xác.

**d) MongoDB**

MongoDB là hệ quản trị cơ sở dữ liệu NoSQL phổ biến, được thiết kế để lưu trữ dữ liệu phi cấu trúc dưới dạng JSON linh hoạt, dễ mở rộng. Với khả năng xử lý dữ liệu lớn và dễ dàng truy xuất.

## 3.5. Thiết kế giao diện

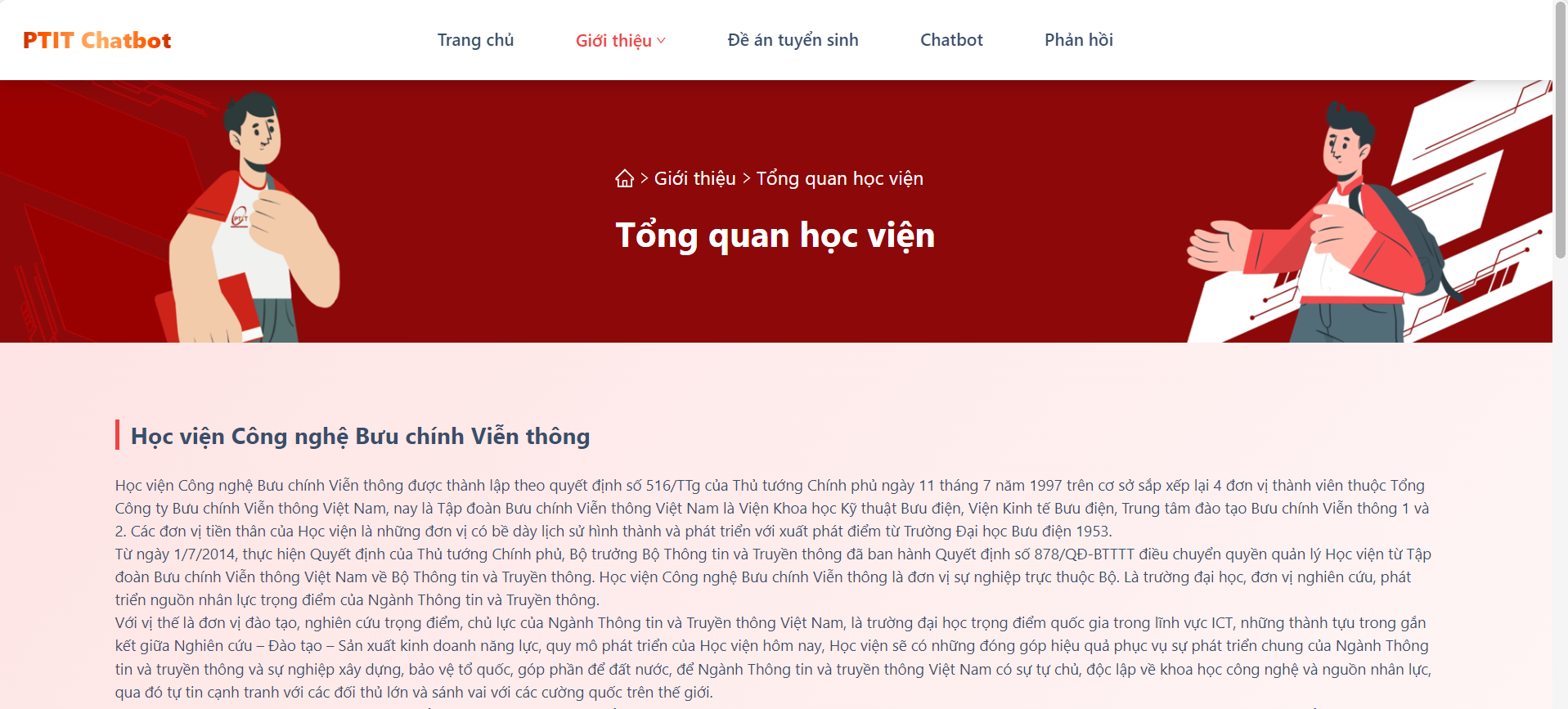
### 3.5.1. Giao diện trang chủ



Hình 3.4. Giao diện trang chủ

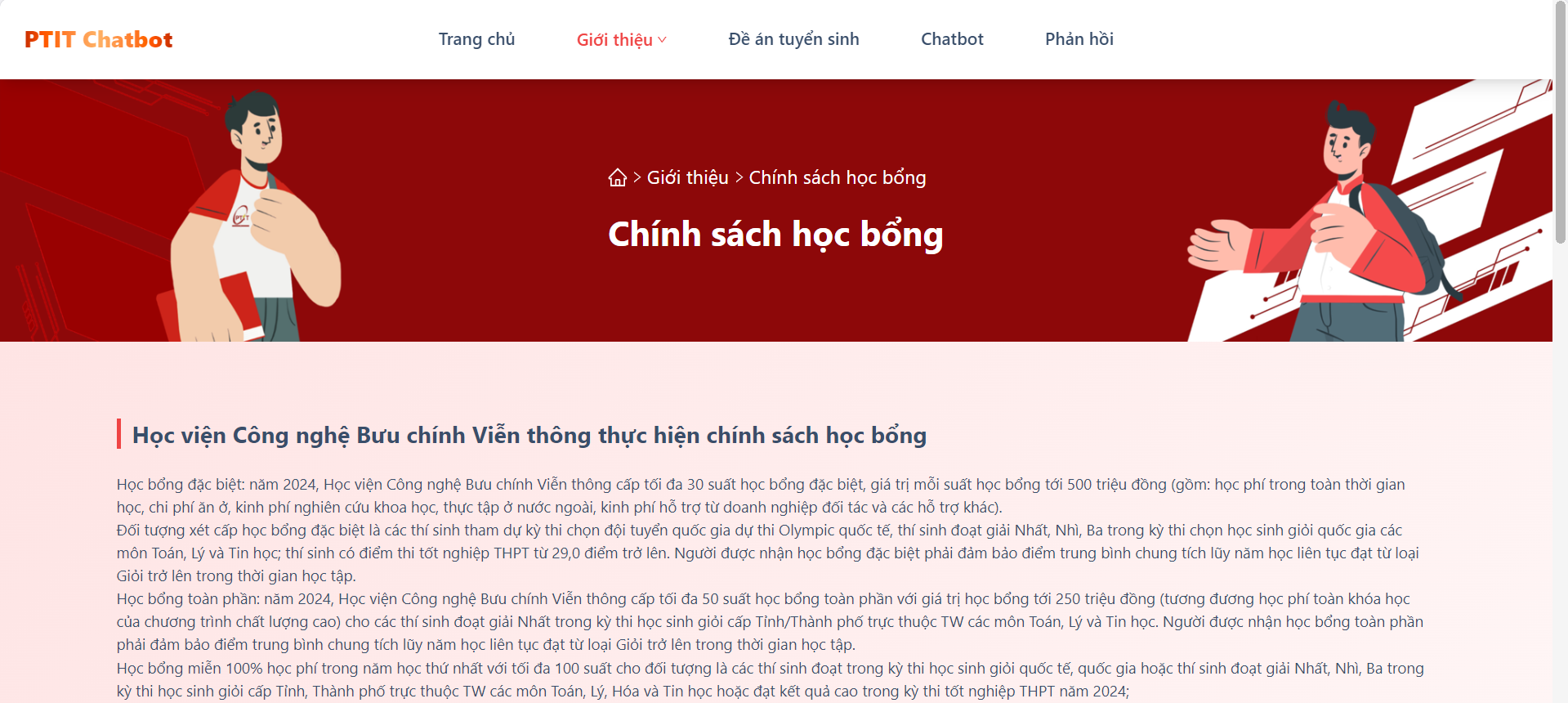
### 3.5.2. Giao diện giới thiệu

* **Tổng quan học viện**

****

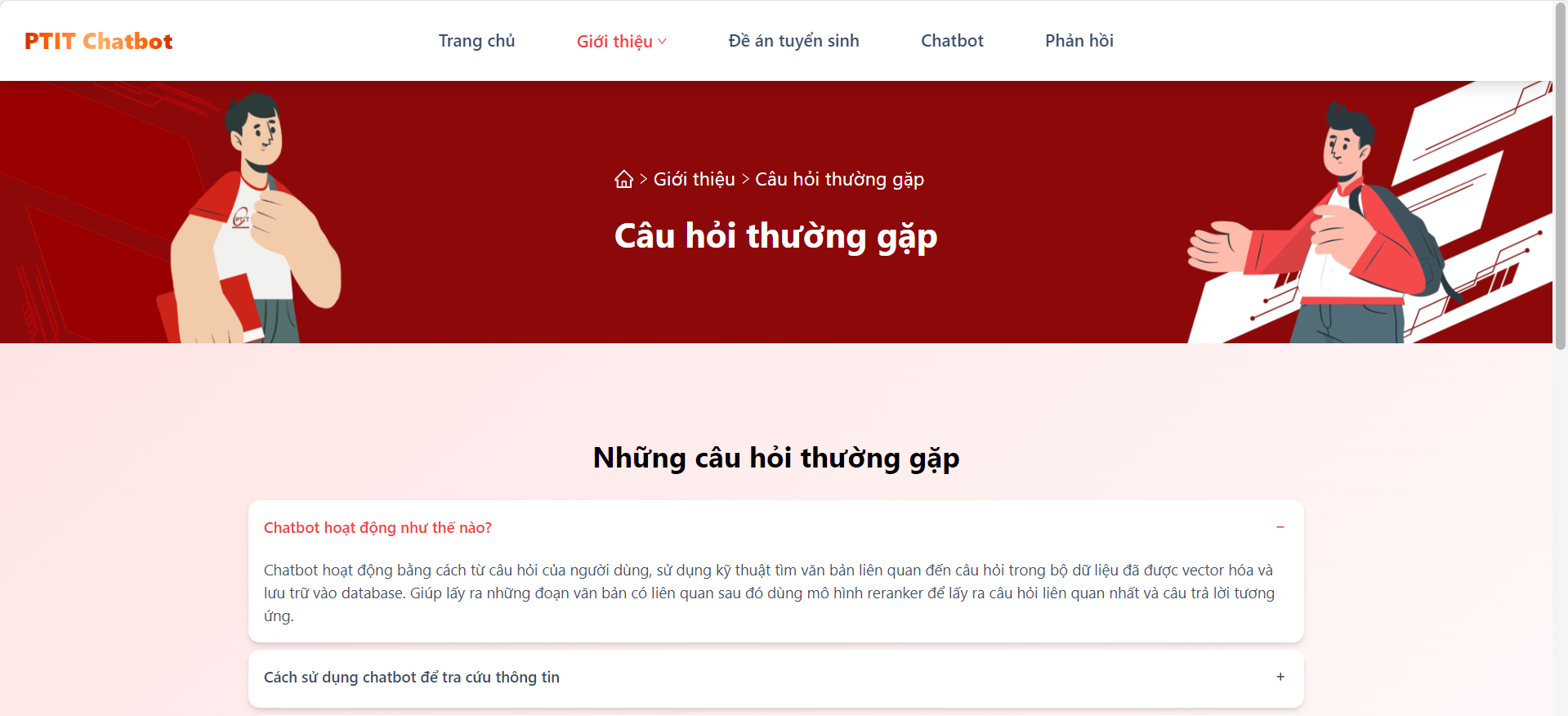
Hình 3.5. Tổng quan học viện.

* **Chính sách học bổng**

****

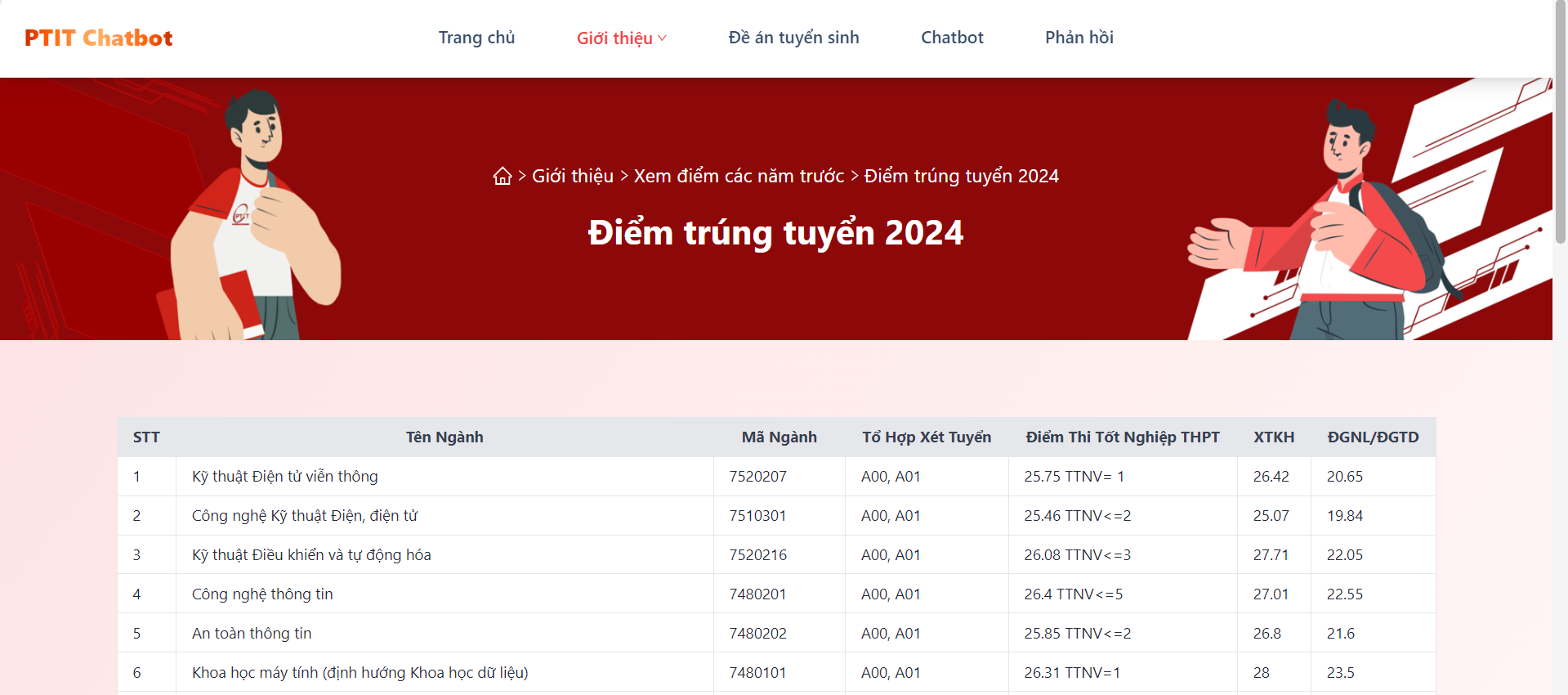
Hình 3.6. Chính sách học bổng.

* **Các câu hỏi thường gặp**

****

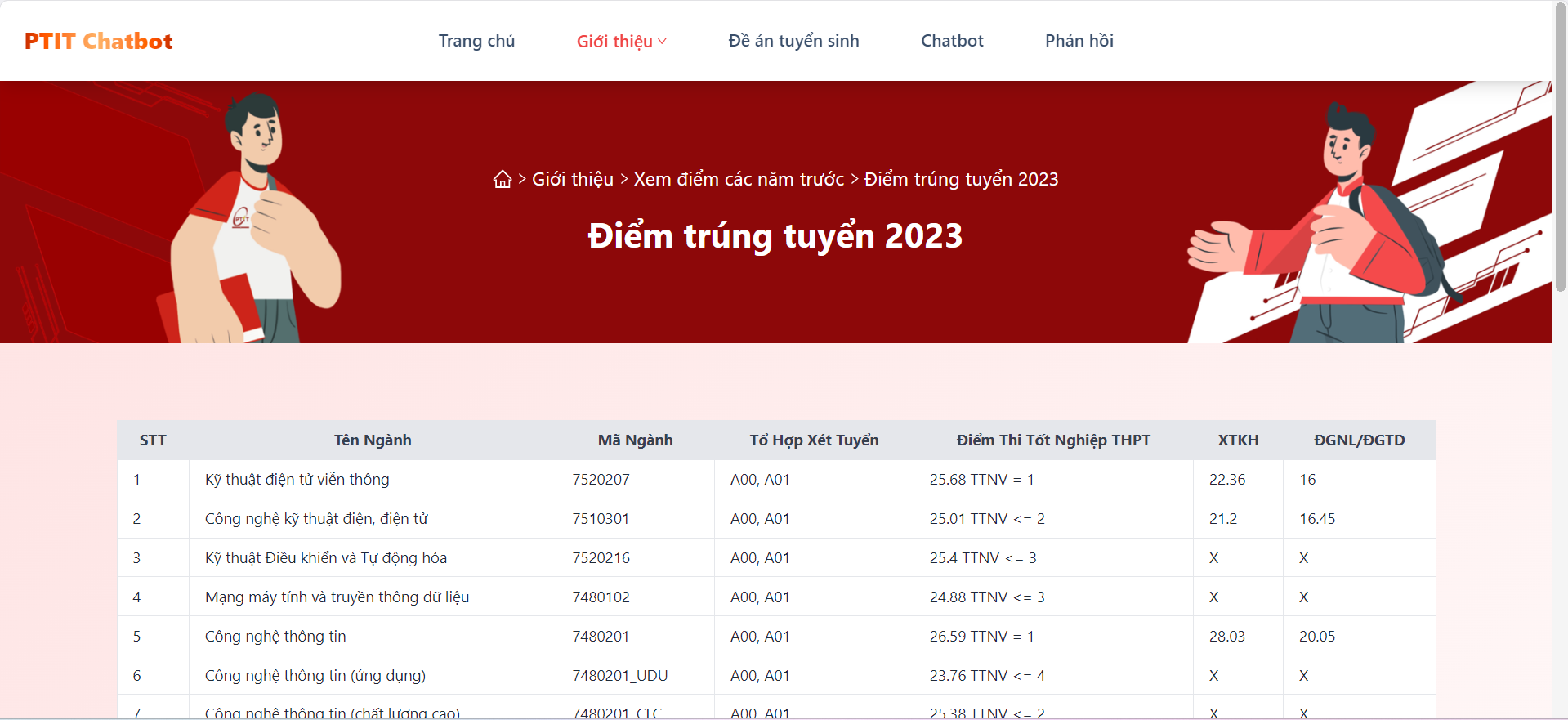
Hình 3.7. Các câu hỏi thường gặp.

* **Điểm các năm trước**
  + **Năm 2024**

****

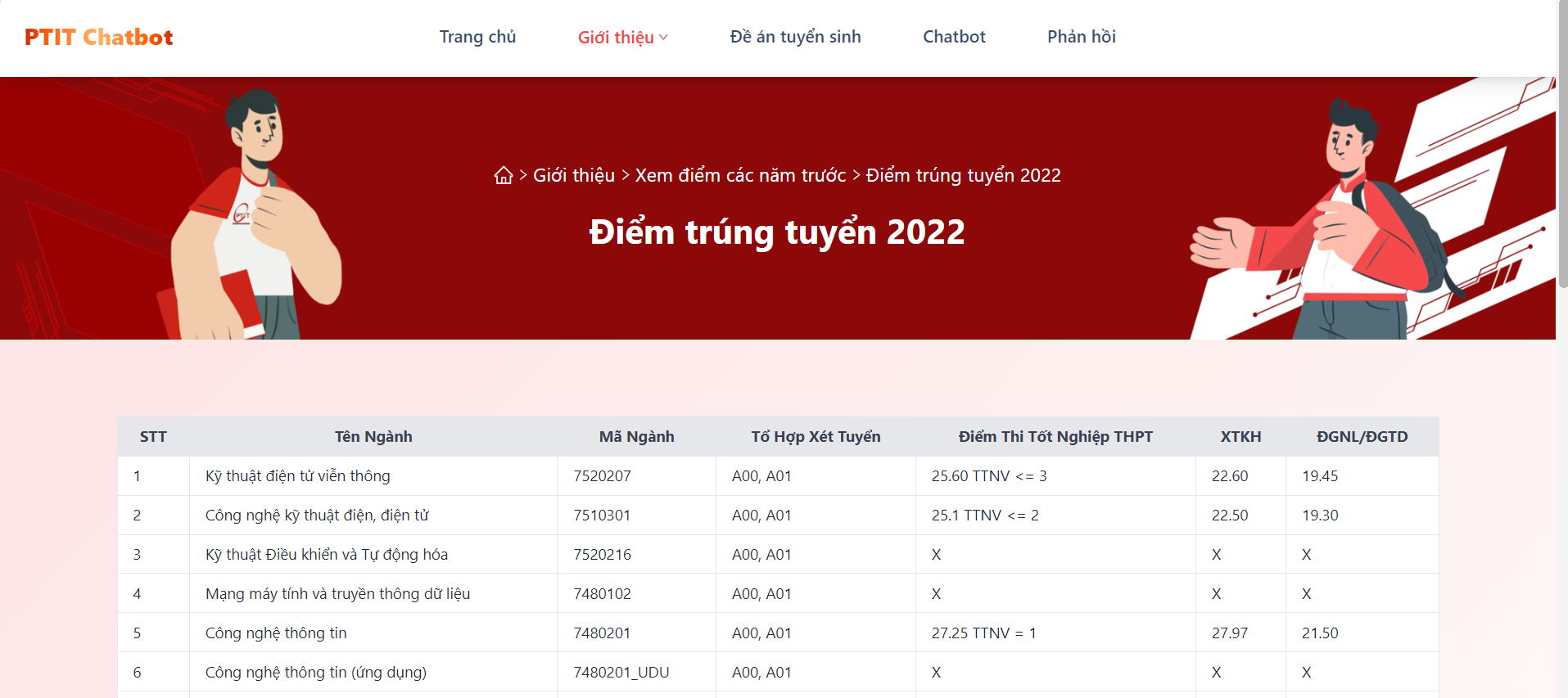
Hình 3.8. Điểm các ngành năm 2024.

* + **Năm 2023**

****

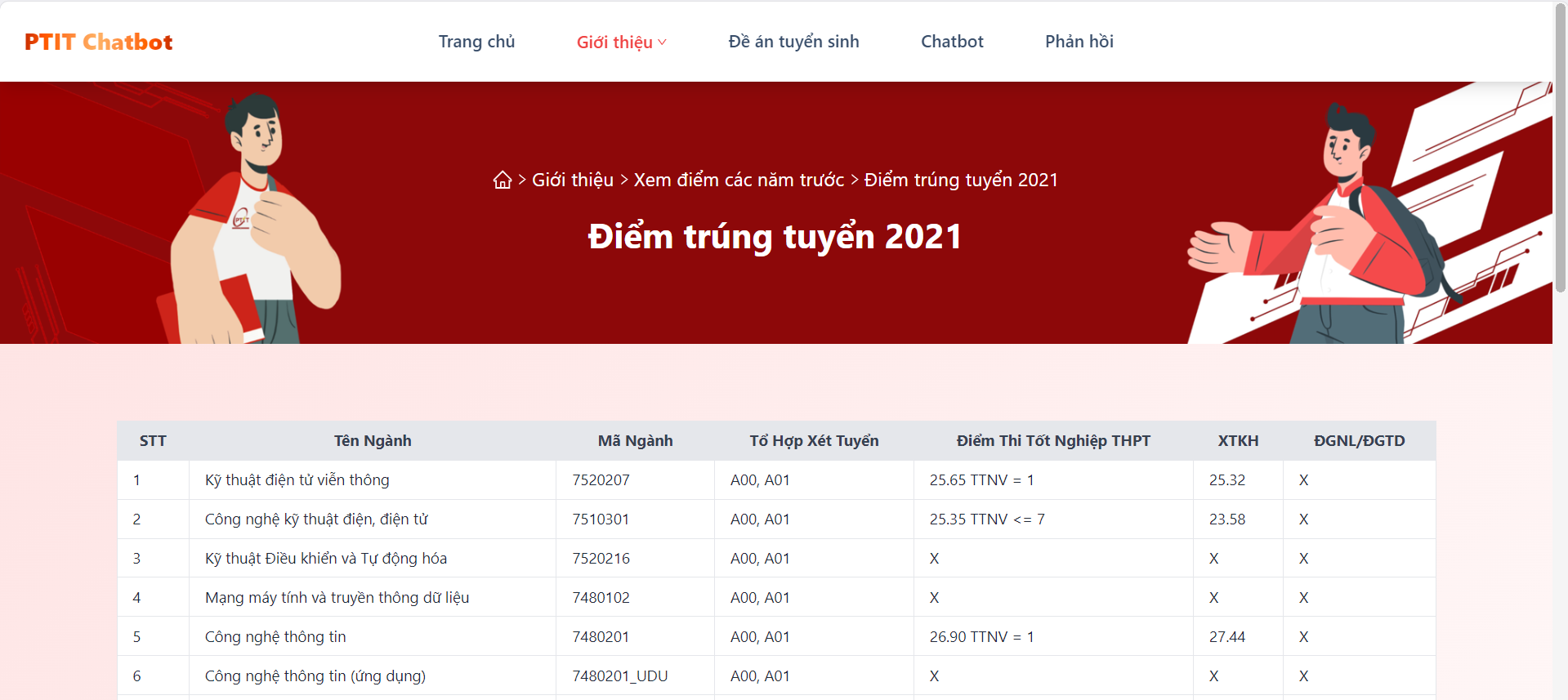
Hình 3.9. Điểm các ngành năm 2023.

* + **Năm 2022**

****

Hình 3.10. Điểm các ngành năm 2022.

* + **Năm 2021**

****

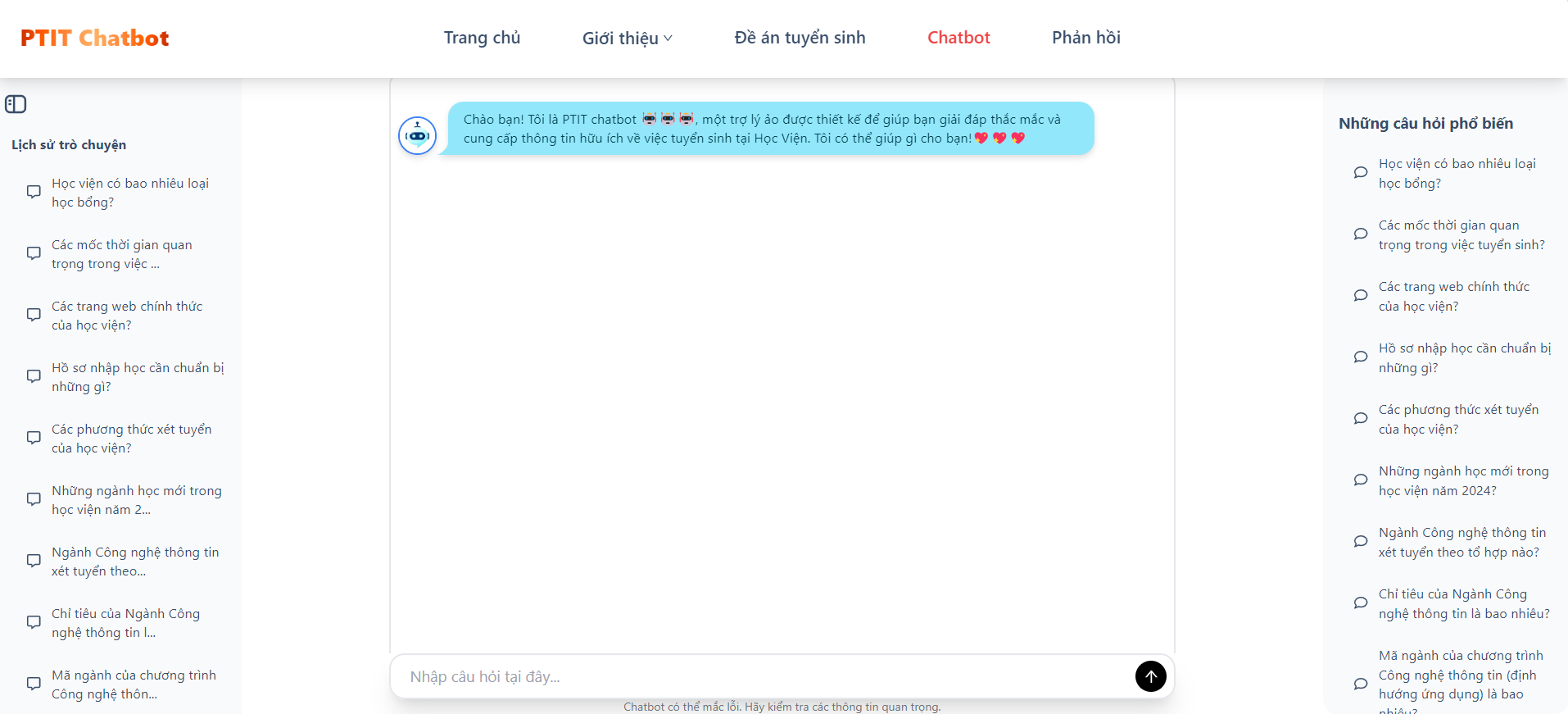
Hình 3.11. Điểm các ngành năm 2021.

### 3.5.3. Giao diện đề án tuyển sinh



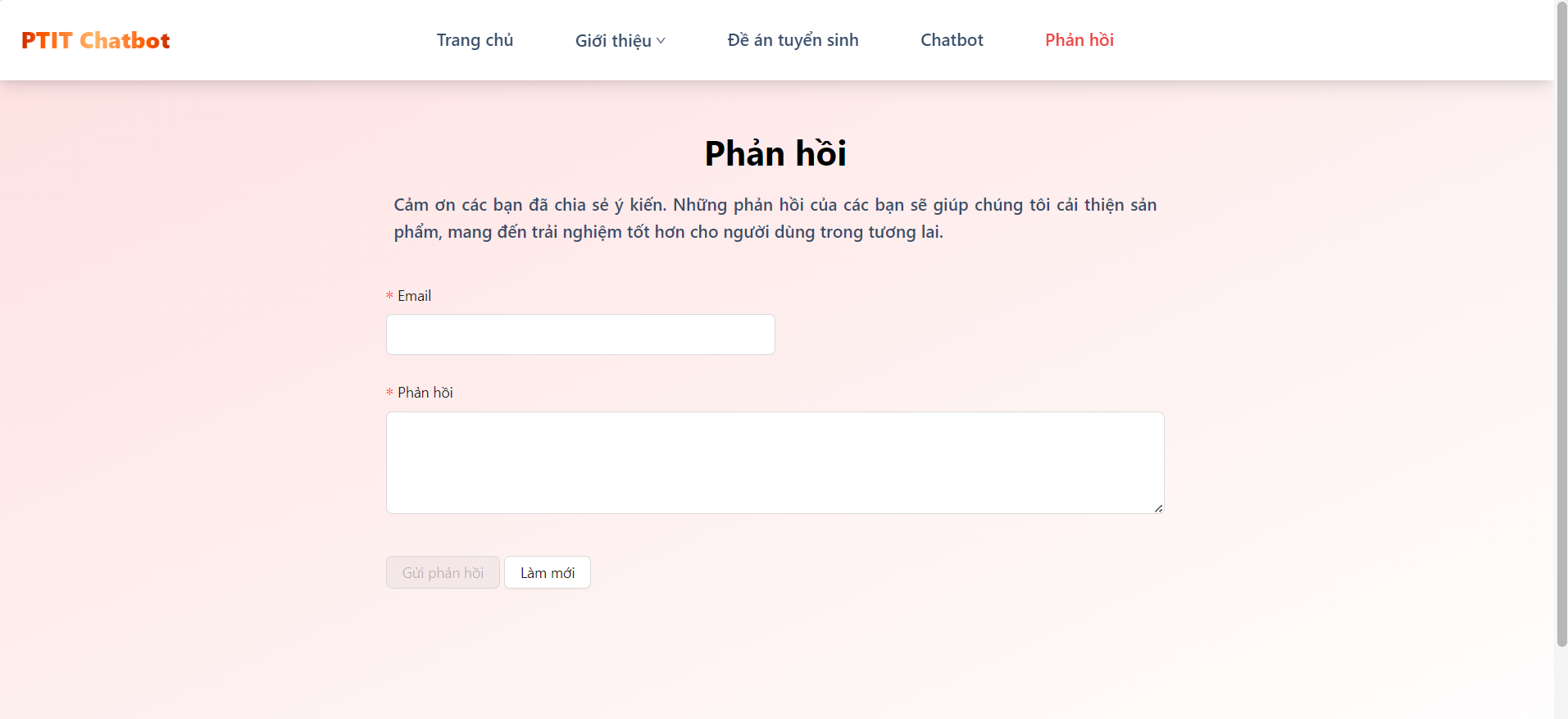
Hình 3.12. Giao diện đề án tuyển sinh.

### 3.5.4. Giao diện chatbot



Hình 3.13 . Giao diện chatbot.

### 3.5.5. Giao diện phản hồi



Hình 3.14. Giao diện phản hồi.

## 3.6. Kết luận chương.

Chương này đã trình bày quá trình xây dựng chatbot tư vấn tuyển sinh, từ việc phát biểu bài toán đến xây dựng dữ liệu, thực nghiệm và cài đặt ứng dụng. Các công nghệ và môi trường phát triển được lựa chọn phù hợp để đảm bảo tính linh hoạt và hiệu quả của hệ thống. Chương này cũng đề cập chi tiết về thiết kế giao diện nhằm mang lại trải nghiệm tối ưu cho người dùng.

# KẾT LUẬN

Qua thời gian tìm hiểu, học hỏi và xây dựng đề tài **“Xây dựng ứng dụng chatbot phục vụ tuyển sinh tại Học viện Công nghệ Bưu chính Viễn thông”,** tôi đã đạt được một số kết quả nhất định như sau:

## 1. Kết quả đạt được

- Tìm hiểu các cách tiếp cận và phương pháp để giải quyết bài toán.

- Ứng dụng xử lý ngôn ngữ tự nhiên cho bài toán.

- Nâng cao kỹ năng phân tích, thiết kế hệ thống.

- Trau dồi thêm kỹ năng lập trình về frontend cũng như backend.

## 2. Hạn chế của đề tài

- Giới hạn về ngữ cảnh cũng như chủ đề, chatbot chỉ trả lời chủ yếu dựa trên các câu hỏi có sẵn trong kho dữ liệu, nên gặp khó khăn trong việc hiểu và xử lý các câu hỏi phức tạp, đặc biệt là khi ngữ cảnh của cuộc hội thoại thay đổi.

- Các thông tin tuyển sinh có thể thay đổi thường xuyên, nên chatbot cần được cập nhật liên tục, đòi hỏi công sức và thời gian.

## 3. Hướng phát triển

Mặc dù các đồ án đã đạt được các mục tiêu đề ra, nhưng do thời gian nghiên cứu cũng như kiến thức, tài nguyên còn hạn chế nên vẫn còn tồn tại một số vấn đề chưa giải quyết được.

Từ những kết quả đã đạt được và những hạn chế đã nêu ở phần trên, tôi xin phép đề xuất hướng phát triển xây dựng RAG chatbot. RAG là phương pháp kết hợp giữa hai kỹ thuật retrieval-based và generative chatbot, giúp chatbot không chỉ tìm kiếm câu trả lời từ các tập tài liệu đã có sẵn mà còn tạo ra những câu trả lời mới một cách linh hoạt, phù hợp với ngữ cảnh câu hỏi. Việc phát triển chatbot lên RAG sẽ cải thiện đáng kể khả năng xử lý ngôn ngữ tự nhiên và đáp ứng các câu hỏi phức tạp của người dùng. Ngoài ra, nên xây dựng cơ chế cập nhật dữ liệu tự động từ các nguồn chính thống của Học viện, giúp chatbot luôn cung cấp thông tin tuyển sinh mới nhất, đảm bảo độ chính xác và tăng cường trải nghiệm cho người dùng.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Tanay Sharma, Tarunpratap Singh Rathore, Narendrapal Singh Rathore, RETRIEVAL BASED CHATBOT SYSTEM paper, 2021 . |
| [2] | N. K. Manaswi, Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras, 2018. |
| [3] | Dat Quoc Nguyen, Anh Tuan Nguyen, PhoBERT: Pre-trained language models for Vietnamese paper, 2020. |