ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG CHATBOT HỖ TRỢ TƯ VẤN TUYỂN SINH TẠI TRƯỜNG ĐẠI HỌC BÁCH KHOA, ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn: **TS.Nguyễn Năng Hùng Vân**

Sinh viên thực hiện: **Lê Văn Mạnh**

Số thẻ sinh viên: **102190391**
Lớp: **19TCLC-DT6**

Đà Nẵng, 12/2024

TÓM TẮT

Đề tài: Xây dựng Chatbot hỗ trợ tư vấn tuyển sinh tại Trường Đại học Bách khoa, Đại học Đà Nẵng

Sinh viên thực hiện: Lê Văn Mạnh

Số thẻ SV: 102190391 Lớp: 19TCLC_DT6

Ngày nay chúng ta đang được sống trong kỷ nguyên của tin học nhờ sự vượt bậc, sự bùng nổ mạnh mẽ của công nghệ thông tin. Công nghệ thông tin không chỉ dừng lại ở mục đích phục vụ cho khoa học kỹ thuật mà đi sâu vào đời sống, chính trị, kinh tế, xã hội, trở nên thân thiện, gần gũi, mang lại nhiều lợi ích cho con người. Đặc biệt trong lĩnh vực giáo dục, Công nghệ thông tin giúp tối ưu hóa quy trình quản lý, hỗ trợ giảng dạy, và đặc biệt là cải thiện việc giao tiếp giữa nhà trường và học sinh. Trong bài toán tuyển sinh đại học, nhu cầu cung cấp thông tin kịp thời, chính xác, và dễ tiếp cận cho thí sinh ngày càng trở nên cấp thiết. Việc ứng dụng AI và Chatbot trong tư vấn tuyển sinh không chỉ giúp giảm tải cho nhân viên tuyển sinh mà còn nâng cao trải nghiệm cho thí sinh, giúp họ dễ dàng tìm kiếm thông tin mà không cần chờ đợi hoặc phụ thuộc vào giờ làm việc.

Đồ án này hướng đến việc xây dựng một hệ thống Chatbot thông minh, sử dụng trí tuệ nhân tạo để hỗ trợ tư vấn tuyển sinh. Hệ thống được thiết kế để giải đáp các câu hỏi thường gặp về ngành học, điều kiện tuyển sinh, học phí, và các mốc thời gian quan trọng. Chatbot cũng đóng vai trò cầu nối, giúp thí sinh tiếp cận thông tin một cách nhanh chóng, chính xác và thân thiện.

Sự kết hợp giữa công nghệ thông tin và giáo dục qua dự án này không chỉ mang lại ý nghĩa thực tiễn mà còn thúc đẩy nghiên cứu và ứng dụng các công nghệ tiên tiến trong lĩnh vực giáo dục, góp phần hiện đại hóa ngành giáo dục Việt Nam.

Về cơ bản, khả năng phản hồi của Chatbot bao gồm :

- Thông tin về trường và các ngành học
- Cách thức, quy trình tuyển sinh
- Thông tin về học phí và học bổng
- Cập nhật thông tin xét tuyển.
- Câu hỏi thường gặp (FAQs).

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM
Đôc lâp - Tư do - Hanh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Lê Văn Mạnh Số thẻ sinh viên: 102190391

Lớp:19TCLC-DT6 Khoa: Công nghệ thông tin Ngành: Hệ thống thông tin

1. *Tên đề tài đồ án:*

   Xây dựng Chatbot hỗ trợ tư vấn tuyển sinh tại trường Đại học Bách khoa, Đại học Đà Nẵng

2. *Đề tài thuộc diện:* ☐ *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:*

   Không có

4. *Nội dung các phần thuyết minh và tính toán:*
   - Mở đầu: Đưa ra giới thiệu chung về đề tài, mục đích, mục tiêu, phạm vi và phương pháp nghiên cứu của đồ án.
   - Chương I - Cơ sở lý thuyết và công nghệ: Trình bày kiến thức và lý thuyết liên quan đến đề tài. Đây là phần để người đọc hiểu rõ về các khái niệm và nguyên lý liên quan đến đồ án.
   - Chương II - Phân tích và thiết kế hệ thống: Trình bày về các yêu cầu được đặt ra trong đề tài bao gồm các chức năng của hệ thống, các sơ đồ đặc tả của hệ thống và các thiết kế cho từng thành phần trong hệ thống.
   - Chương III - Triển khai và kết quả: Trình bày quá trình phát triển dự án, triển khai hệ thống và các kết quả về chức năng ứng dụng.
   - Kết luận và hướng phát triển: Tổng kết lại các kết quả, đóng góp của đồ án, đưa ra những đề xuất và kiến nghị để cải thiện hoặc phát triển tiếp hệ thống.
5. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*

   Không có

6. *Họ tên người hướng dẫn:* TS.Nguyễn Năng Hùng Vân
7. *Ngày giao nhiệm vụ đồ án: ……/……./2024*
8. *Ngày hoàn thành đồ án: ……../……./2024.*

Đà Nẵng, ngày tháng năm 202

Trưởng Bộ môn …….. **Người hướng dẫn**

LỜI CẢM ƠN

Trong quãng thời gian học tập tại trường Đại Học Bách Khoa, Đại Học Đà Nẵng, cũng như trong khoản thời gian thực hiện và hoàn thành đồ án này, em đã nhận được sự giúp đỡ, hướng dẫn, giảng dạy nhiệt tình của quý Thầy, Cô trong khoa Công Nghệ Thông Tin. Em xin chân thành cảm ơn các thầy cô đã dìu dắt, dạy dỗ em cả về kiến thức chuyên môn và tinh thần học tập để em có được những kiến thức thực hiện đồ án tốt nghiệp của mình.

Trong quá trình thực hiện đồ án em xin cám ơn thầy Nguyễn Năng Hùng Vân. Thầy là người đã định hướng và giải đáp những thắc mắc trong những buổi gặp mặt đầu tiên và luôn hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện đồ án này.

Em xin chân thành cảm ơn đến các thầy cô giảng viên Khoa Công nghệ Thông tin của trường Đại học Bách khoa Đà Nẵng. Nhờ kiến thức chuyên môn, sự truyền cảm hứng và động lực mà các thầy cô đã truyền đạt cho em, em đã có những nền tảng vững chắc để thực hiện đồ án tốt nghiệp.

Ngoài ra, em muốn gửi lời cảm ơn chân thành tới tất cả những ai đã dành thời gian và công sức để đọc, xem xét và đóng góp ý kiến cho đồ án của tôi. Những góp ý và nhận xét quý báu từ mọi người đã giúp em nắm bắt được những sai sót và khía cạnh cần thiết để nâng cao chất lượng và hoàn thiện đồ án của mình. Em trân trọng mọi đóng góp và sẽ tiếp thu những góp ý đó để phát triển hơn trong tương lai.

Mặc dù đã cố gắng trong quá trình nghiên cứu làm đề tài nhưng do còn nhiều hạn chế về thời gian và trình độ nên đồ án của em không tránh khỏi nhiều thiếu sót, nhiều vấn đề chưa được giải quyết hoàn chỉnh. Vì vậy em rất mong nhận được những ý kiến đóng góp của các thầy cô để có thể hoàn thiện và phát triển đề tài hơn.

Một lần nữa em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Công nghệ đang ngày càng phát triển qua từng năm, dẫn đến việc tra cứu thông tin của con người cũng thay đổi theo, từ những phương pháp tra cứu dữ liệu đơn giản bằng cách tìm kiếm theo các từ trong câu cho đến những phương pháp phức tạp như áp dụng học máy để trích xuất dữ liệu chính xác hơn dựa trên ngữ nghĩa của các từ đó. Sự đột phá trong lĩnh vực tra cứu thông tin được tái định nghĩa lại sau sự ra đời của ChatGPT một mô hình ngôn ngữ được OpenAI phát triển đã làm cách mạng hoá việc tìm kiếm, làm cho việc tìm kiếm thông tin được biểu diễn dưới dạng một chatbot hỏi đáp, nhờ vậy mô hình này đã giúp giải đáp các thông tin theo một cách thức giống con người hơn.

Tuy nhiên ChatGPT cũng tồn tại những hạn chế của một mô hình ngôn ngữ, nhất là khả năng bị lỗi thời thông tin (outdated information), dẫn chứng là thời điểm ra mắt mô hình này chỉ cập nhật kiến thức đến năm 2021. Hiện nay, tất cả những vấn đề này đã được khắc phục nhờ phương pháp tang cường AI tạo sinh, giúp mô hình có thể cập nhật thông tin từ nhiều nguồn theo thời gian thực. Ở đề tài này ta sẽ mô phỏng lại hệ thống đó, bằng việc xây dựng một hệ thống chatbot hỏi đáp có thể lấy dữ liệu từ trang thộng tin tuyển sinh của Trường Đại học Bách Khoa Đà Nẵng. Để xây dựng được hệ thống này ta cần tìm hiểu về cách các mô hình ngôn ngữ lớn hoạt động, các phương pháp trích xuất thông tin liên quan, xử lý dữ liệu, cách triển khai mô hình trên server và làm thế nào để đánh giá hiệu quả của một hệ thống hỏi đáp.

Bằng việc đón đầu xu thế chung về công nghệ AI tạo sinh, đề tài cũng đánh mạnh vào quá trình xử lý dữ liệu, một thứ cốt lõi quyết định độ hiệu quả của mô hình ngôn ngữ trong việc trích xuất thông tin và trả lời câu hỏi, từ đó có thể áp dụng kỹ thuật này trong nhiều lĩnh vực khác nhau có thể là việc tạo một hệ thống tra cứu tài liệu nội bộ, hỗ trợ tư vấn bán hàng hoặc là một trợ thủ đắc lực giúp việc học tập hiệu quả hơn.

CAM ĐOAN

Tôi xin cam đoan:

1. Báo cáo đồ án tốt nghiệp Tên đề tài: Xây dựng Chatbot AI hỗ trợ tư vấn tuyển sinh cho trường Đại học Bách khoa Đà Nẵng là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của giảng viên TS. Nguyễn Năng Hùng Vân.

2. Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.

3. Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Lê Văn Mạnh

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

DANH SÁCH CÁC TỪ VIẾT TẮT

Chữ viết tắt	Diễn giải	Ý nghĩa
AI	Artificial Intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng Nơ-ron nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
BERT	Bidirectional Encoder Representations from Transformers.	Mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer
BPTT	Back-propagation through time	Lan truyền ngược theo thời gian
DL	Deep learning	Học sâu
GPT	Generative Pre-training Tranformer	Bộ biến đổi sinh tiền huấn luyện
LLM	Large Language Model	Mô hình ngôn ngữ lớn
LSTM	Long short-term memory	Bộ nhớ ngắn-dài hạn
IR	Information Retrieval System	Hệ thống truy xuất thông tin
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
ML	Machine learning	Học máy
SL	Self-supervised learning	Học tự giám sát
RAG	Retrieval-Augmented Generation	Tăng cường tạo sinh
RC	Reading Comprehension	Bước đọc hiểu
RNN	Recurrent Neural Network	Mạng thần kinh hồi quy
SEP	Separate token	Token ngăn cách
SOTA	State-of-the-art	Tiên tiến nhất
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
TF	Term Frequency	Tần suất từ
QA	Question Answering System	Hệ thống hỏi đáp

MỞ ĐẦU

Hiện nay, Trường đại học Bách khoa, Đại học Đà Nẵng đang là một ngôi trường được nhiều sĩ tử lựa chọn vào học. Nhưng đối mặt với rất nhiều chuyên ngành với đặc thù khác nhau có thể gây bối rối cho những bạn mới tốt nghiệp THPT. Việc áp dụng CNTT, cụ thể ở đây là chatbox AI, vào quá trình tư vấn tuyển sinh là một yếu tố cần thiết và mang lại nhiều hiệu quả tích cực. Chính vì vậy, tôi đã chọn đề tài xây dựng chatbot tư vấn tuyển sinh nhằm hỗ trợ Trường Bách khoa trong việc giải đáp thắc mắc cho thí sinh một cách nhanh chóng, thuận tiện và hiệu quả, đồng thời giảm tải công việc cho bộ phận tư vấn tuyển sinh.

1. Bối cảnh đề tài

Trong những năm gần đây, sự phát triển của AI đã làm thay đổi sâu sắc đến nhiều khía cạnh của đời sống, bao gồm cả lĩnh vực giáo dục. Đặc biệt, với sự ra đời của các LLM làm tăng khả năng hiểu và xử lý ngôn ngữ tự nhiên của máy tính một cách đáng kể, mở ra những tiềm năng mới trong việc xây dựng các hệ thống tự động hóa, trong đó có Chatbot trong lĩnh vực chăm sóc khách hàng. Những công nghệ này có thể xử lý khối lượng dữ liệu khổng lồ, học hỏi từ những mẫu dữ liệu phức tạp, và tự động hóa các cuộc hội thoại một cách tự nhiên, mang lại trải nghiệm người dùng thân thiện và hiệu quả hơn.

Tại Việt Nam, việc tuyển sinh ở các trường đại học và cao đẳng hàng năm vẫn gặp nhiều khó khăn, đặc biệt là trong việc cung cấp thông tin đầy đủ, chính xác và kịp thời cho thí sinh và phụ huynh. Truyền thống tư vấn tuyển sinh thông qua các cuộc hội thảo, ngày hội tuyển sinh, hay hỗ trợ qua các kênh tư vấn trực tiếp đòi hỏi nhiều thời gian, chi phí nhân sự và khó mở rộng quy mô trong thời gian ngắn. Trong bối cảnh đó, việc áp dụng các công nghệ tự động hóa như chatbot tuyển sinh trở thành một giải pháp tiềm năng để tối ưu hóa quy trình tuyển sinh.

Với mong muốn tiếp cận nhanh chóng và hiệu quả hơn với các thí sinh tiềm năng, em đã đề xuất ý tưởng xây dựng một Chatbot tuyển sinh dựa trên nền tảng LLM. Chatbot này sẽ đóng vai trò là một công cụ hỗ trợ tự động, có khả năng trả lời các câu hỏi từ phía học sinh và phụ huynh, từ thông tin về chương trình đào tạo, điều kiện tuyển sinh, đến chính sách học bổng và hỗ trợ sinh viên. Với khả năng xử lý ngôn ngữ tự nhiên của LLM, Chatbot có thể mang lại trải nghiệm tư vấn tuyển sinh chân thực, giảm tải công

việc cho đội ngũ tư vấn của nhà trường, đồng thời tạo ra một kênh giao tiếp hiện đại và hiệu quả hơn.

Vì vậy, bối cảnh của đề tài này không chỉ là nhu cầu ứng dụng công nghệ vào giáo dục mà còn là một bước tiến trong việc nâng cao trải nghiệm người dùng trong công tác tuyển sinh, giúp nhà trường hiện đại hóa quá trình tuyển sinh và đáp ứng tốt hơn nhu cầu thông tin của thí sinh.

2.Lý do chọn đề tài

Lý do chọn đề tài "Xây dựng Chatbot hỗ trợ tuyển sinh tại Trường Đại học Bách khoa, Đại học Đà Nẵng" xuất phát từ nhu cầu cấp thiết trong việc ứng dụng công nghệ hiện đại để cải thiện quy trình tuyển sinh. Hiện nay, với sự phát triển nhanh chóng của các mô hình ngôn ngữ lớn và trí tuệ nhân tạo, việc sử dụng AI để tối ưu hóa các dịch vụ chăm sóc và hỗ trợ người dùng trở nên khả thi và hiệu quả hơn bao giờ hết. Trong bối cảnh cuộc cách mạng công nghiệp 4.0, việc áp dụng AI không chỉ giúp tiết kiệm nguồn lực mà còn nâng cao trải nghiệm người dùng thông qua sự tương tác thông minh và cá nhân hóa.

Đặc biệt, trong lĩnh vực giáo dục, tuyển sinh là một quy trình quan trọng, ảnh hưởng trực tiếp đến hình ảnh và uy tín của nhà trường. Đối với Trường Đại học Bách khoa Đà Nẵng, mỗi năm có hàng nghìn học sinh và phụ huynh có nhu cầu tìm hiểu thông tin về các chương trình đào tạo, quy chế tuyển sinh, học phí và học bổng. Tuy nhiên, với các phương pháp tư vấn truyền thống, việc đáp ứng kịp thời và chính xác nhu cầu thông tin của tất cả người dùng là một thách thức lớn, đặc biệt là trong thời gian cao điểm của mùa tuyển sinh. Do đó, việc xây dựng một chatbot tuyển sinh không chỉ là giải pháp tiết kiệm chi phí, thời gian mà còn đảm bảo tính liên tục, chính xác trong việc cung cấp thông tin tuyển sinh.

Lý do chọn đề tài còn xuất phát từ tiềm năng mà các mô hình LLM mang lại. Những mô hình này có khả năng xử lý ngôn ngữ tự nhiên một cách linh hoạt và sâu sắc, giúp chatbot không chỉ có thể trả lời các câu hỏi đơn giản mà còn cung cấp thông tin chi tiết, đa dạng và liên tục cập nhật. Việc ứng dụng LLM vào Chatbot tuyển sinh còn là một bước đi tiên phong, mang lại lợi thế cạnh tranh và nâng cao hình ảnh hiện đại, năng động của nhà trường trong mắt các thí sinh và phụ huynh.

Tóm lại, đề tài này vừa đáp ứng nhu cầu thực tiễn trong quy trình tuyển sinh của nhà

trường, vừa là cơ hội để khám phá, nghiên cứu và áp dụng một trong những công nghệ tiên tiến nhất hiện nay – mô hình ngôn ngữ lớn, từ đó góp phần nâng cao trải nghiệm người dùng và tối ưu hóa hoạt động tuyển sinh trong môi trường giáo dục.

3.Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài này là tập trung nghiên cứu các LLM phục vụ cho việc xây dựng chatbot tuyển sinh, cụ thể là việc ứng dụng các mô hình như Ollama, GPTGrok, PhoBERT v.v hoặc các mô hình ngôn ngữ có khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ khác vào lĩnh vực tư vấn giáo dục. Đối tượng nghiên cứu bao gồm các yếu tố kỹ thuật và thuật toán của mô hình ngôn ngữ, phương pháp nạp dữ liệu và tối ưu hóa mô hình, cũng như khả năng tương tác của mô hình với người dùng trong ngữ cảnh cung cấp thông tin tuyển sinh.

Ngoài ra, nghiên cứu cũng xem xét các phương pháp đánh giá chất lượng và hiệu quả hoạt động của Chatbot, bao gồm độ chính xác của câu trả lời, khả năng hiểu ngữ cảnh, mức độ hài lòng của người dùng, và khả năng mở rộng của chatbot để đáp ứng nhu cầu người dùng thực tế.

4.Phạm vi nghiên cứu

Phạm vi ứng dụng: Chatbot được xây dựng nhằm phục vụ công tác tuyển sinh cho Trường Đại học Bách khoa, Đại học Đà Nẵng. Do đó, chatbot sẽ tập trung vào các nội dung xoay quanh thông tin tuyển sinh của trường như chương trình đào tạo, quy chế tuyển sinh, học phí, chính sách học bổng, cũng như các thông tin liên quan đến cơ hội việc làm và các hoạt động hỗ trợ sinh viên. Chatbot cần đảm bảo trả lời chính xác các câu hỏi phổ biến, đồng thời có khả năng mở rộng để cập nhật thông tin mới theo thời gian.

Phạm vi công nghệ: Đề tài sẽ tập trung vào các mô hình ngôn ngữ lớn với khả năng xử lý ngôn ngữ tự nhiên cao, giúp Chatbot có thể hiểu và phản hồi các câu hỏi từ người dùng một cách tự nhiên và chính xác. Các nền tảng và thư viện hỗ trợ Chatbot như PyTorch, TensorFlow, các API dịch vụ ngôn ngữ có sẵn cũng sẽ được nghiên cứu và ứng dụng để tối ưu hóa hiệu suất của chatbot.

Phạm vi người dùng: Chatbot hướng đến nhóm người dùng bao gồm học sinh, sinh viên có nhu cầu tìm hiểu thông tin về Trường Đại học Bách khoa, Đại học Đà Nẵng, phụ huynh học sinh và các cá nhân quan tâm đến việc nhập học tại trường. Đối tượng người dùng có thể truy cập Chatbot qua các kênh truyền thông chính thức của nhà trường, giúp đảm bảo tính thuận tiện và dễ dàng cho người dùng khi tìm kiếm thông tin tuyển sinh.

Với các đối tượng và phạm vi nghiên cứu rõ ràng, đề tài không chỉ tập trung vào việc phát triển Chatbot tuyển sinh mà còn mở ra cơ hội nâng cao trải nghiệm tư vấn giáo dục thông qua ứng dụng AI.

5.Nội dung nghiên cứu

Đề tài này tập trung vào những nội dung nghiên cứu bao: Đầu tiên, đề tài sẽ tập trung tìm hiểu và đánh giá các LLM và các biến thể phổ biến khác. Mục tiêu là phân tích các ưu nhược điểm và lựa chọn mô hình phù hợp nhất để áp dụng vào Chatbot tuyển sinh, đồng thời nghiên cứu kỹ thuật tinh chỉnh nhằm đáp ứng các yêu cầu đặc thù trong bối cảnh của trường. Tiếp theo, đề tài sẽ triển khai hệ thống Chatbot thông qua thiết kế kiến trúc tổng thể, bao gồm mô hình ngôn ngữ và giao diện người dùng. Một pipeline dữ liệu sẽ được xây dựng từ quá trình thu thập, tiền xử lý dữ liệu và nạp dữ liệu vào các LLM, sử dụng các thư viện như PyTorch, TensorFlow, tích hợp qua API dịch vụ ngôn ngữ để tối ưu hóa hiệu suất.

Để đảm bảo chất lượng của Chatbot, nhóm nghiên cứu sẽ thiết lập các tiêu chí đánh giá như độ chính xác, thời gian phản hồi, khả năng hiểu ngữ cảnh và mức độ hài lòng của người dùng. Sau đó, tiến hành thử nghiệm trên các kịch bản tuyển sinh thực tế nhằm đánh giá hiệu quả của Chatbot và từ đó thực hiện các bước tối ưu hóa để đảm bảo Chatbot hoạt động ổn định, chính xác. Cuối cùng, Chatbot sẽ được triển khai trên các kênh truyền thông của nhà trường, như website tuyển sinh và các nền tảng mạng xã hội, nhằm dễ dàng tiếp cận với học sinh và phụ huynh. Sau khi triển khai, đề tài cũng đánh giá khả năng mở rộng các tính năng nâng cao cho Chatbot trong tương lai, ví dụ như tự động cập nhật dữ liệu tuyển sinh, hỗ trợ hướng nghiệp hoặc tích hợp các dịch vụ sinh viên khác. Nội dung nghiên cứu này giúp đảm bảo rằng Chatbot tuyển sinh được xây dựng một cách bài bản, đáp ứng tốt nhu cầu người dùng và góp phần nâng cao hiệu quả quy trình tuyển sinh của trường.

6.Bố cục bài báo cáo

Bài báo cáo ĐATN ngoài phần mở đầu và kết luận thì bố cục được chia làm 4 chương lớn, cụ thể như sau:

Tổng quan và giới thiệu đề tài. Trong mục này sẽ tập trung vào việc nêu lên tính cấp thiết, giới thiệu bối cảnh và mục đích đề tài.

Chương 1. Cơ sở lý thuyết. Trong chương này tập trung vào các cơ sở lý thuyết và các

công cụ được sử dụng trong đề tài này.

Chương 2. Phương pháp đề xuất sử dụng mô hình ngôn ngữ lớn trong Chatbot. Trong chương này tập trung vào phương pháp đề xuất, giải thích từng phương pháp, các bước thực hiện trong mô hình.

Chương 3. Triển khai xây dựng Chatbot hỗ trợ tư vấn tuyển sinh tại Trường Đại học Bách khoa, Đại học Đà Nẵng.

Trong chương này tập trung vào việc nêu rõ các bước thực hiện và cuối cùng là đánh giá kết quả sản phẩm.

Kết luận và phương hướng tiếp theo, phần này sẽ nêu lên những vấn đề đạt được, các vấn đề cần giải quyết và phương hướng được đề xuất trong tương lai.

CHƯƠNG 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

1.1 Tổng quan về Học máy

1.1.1 Giới thiệu về Học máy

Học máy (ML) là lĩnh vực con của AI, một nhánh đang phát triển của các thuật toán tính toán được thiết kế để mô phỏng trí thông minh của con người bằng cách học hỏi từ môi trường xung quanh [1]. Thay vì được lập trình với các quy tắc cứng nhắc, các mô hình học máy được huấn luyện để nhận diện mẫu và đưa ra dự đoán dựa trên dữ liệu đầu vào. Học máy có thể được chia thành ba loại chính: học có giám sát (supervised learning), học không giám sát (unsupervised learning) và học tăng cường (reinforcement learning). Ngày nay, học máy ngày càng trở nên phổ biến trong các ứng dụng thực tiễn, từ nhận diện hình ảnh, phân tích văn bản, đến dự đoán hành vi người dùng [2].

1.1.2 Học máy có giám sát

Học máy có giám sát (SL) là một mô hình học máy, trong đó các đối tượng đầu vào (là một vector đầu vào hoặc các biến dự báo) và một giá trị đầu ra mong muốn (còn được gọi là tín hiệu giám sát được gắn nhãn bởi con người) đào tạo một mô hình [3]. Dữ liệu đào tạo được xử lý, xây dựng một hàm ánh xạ dữ liệu mới thành các giá trị đầu ra mong đợi [4]. Mục tiêu của thuật toán học có giám sát là tìm ra một hàm số

$$f: X \to Y$$

với X là không gian các đặc trưng (features) và Y là không gian các nhãn (labels). Thuật toán học có giám sử dụng dữ liệu đầu vào (x_1, y_1), (x_2, y_2), …, (x_n, y_n), trong đó mỗi cặp (x_i, y_i) chứa một đặc trưng x_i (input) và nhãn tương ứng y_i (output) [5]. Để đo lường mức độ học tập của một hàm với dữ liệu đào tạo, một hàm mất mát

$$L: Y \times Y \to \mathbb{R}^{\geq 0}$$

được định nghĩa. Đối với cặp (x_i, y_i), sự mất mát của việc dự đoán giá trị $\hat{y}$ là $L(y_i, \hat{y})$ [6].

1.1.3 Học máy không giám sát

Học không giám sát là một nhánh của học máy mà trong đó các thuật toán tự khám phá cấu trúc hoặc mẫu từ dữ liệu không có nhãn. Theo Ghahramani (2003) trong bài báo "Unsupervised Learning," học không giám sát chủ yếu tập trung vào việc nhận dạng và

phân nhóm các mẫu trong dữ liệu mà không cần biết trước bất kỳ nhãn hoặc hướng dẫn nào [7].

Mục đích của học không giám sát là tìm cách tối ưu hóa một hàm mục tiêu f sao cho mô hình có thể biểu diễn cấu trúc ẩn trong tập dữ liệu mà không có nhãn. Quá trình này thường liên quan đến việc tối ưu một hàm mất mát L để khám phá các đặc trưng quan trọng hoặc cấu trúc của dữ liệu.

Giả sử ta có tập dữ liệu $X = \{x_1, x_2, \dots, x_n\}$, trong đó mỗi $x_i \in \mathbb{R}^d$ là một điểm dữ liệu trong không gian d_{dim}. Ta xây dựng một hàm mục tiêu

$$f: \mathbb{R}^d \to \mathbb{R}^k \text{ (với } k \leq d)$$

để biến đổi dữ liệu X sao cho $f(X)$ có thể biểu diễn các đặc trưng quan trọng của dữ liệu. Hàm mục tiêu này có thể được chọn sao cho phản ánh được các mối quan hệ, mẫu hoặc cấu trúc ẩn trong dữ liệu. Để tìm các biểu diễn tối ưu, ta tối thiểu hóa một hàm mất mát tổng quát [8]:

$$L\big(X, f(X)\big) = \sum_{i=1}^{n} d\big(x_i, f(x_i)\big)$$

Về cơ bản học máy có giám sát và học máy không giám sát là hai phương pháp học máy phổ biến nhất được sử dụng trong nhiều tác vụ từ đơn giản đến phức tạp và đã chứng minh được hiệu quả của chúng [9]. Ngoài hai phương pháp học máy trên còn nhiều phương pháp tiên tiến khác nhằm giải quyết từng nhiệm vụ cụ thể. Một vài thuật toán có thể kể đến như là học bán giám sát, học tăng cường v.v. Tuy nhiên phương pháp học máy tự giám sát (self-supervised learning) và học đa nhiệm (multi-task learning) là hai phương pháp quan trọng được sử dụng để huấn luyện mô hình ngôn ngữ lớn liên quan tới chatbot [10].

1.1.4 Ứng dụng của học máy

Học máy đóng một vai trò thiết yếu trong sự phát triển và ứng dụng của trí tuệ nhân tạo, mang lại nhiều lợi ích đáng kể trong các lĩnh vực khác nhau. ML giúp tự động hóa và tối ưu hóa các quy trình, giảm thiểu sự can thiệp của con người trong các tác vụ phức tạp [11] và cung cấp khả năng phân tích dữ liệu lớn một cách hiệu quả. Với sự gia tăng khối lượng dữ liệu trong kỷ nguyên số, các phương pháp học máy cho phép doanh nghiệp khai thác dữ liệu để tìm ra thông tin quý giá, từ đó đưa ra quyết định chính xác hơn [12].

1.2 Hệ thống trả lời câu hỏi

1.2.1 Giới thiệu về hệ thống trả lời câu hỏi

Tìm kiếm đáp án cho câu truy vấn là một chức năng quan trọng trong nhiều ứng dụng và công ty trên toàn cầu. Cho dù trong lĩnh vực sản xuất, tài chính, y tế hay hầu như bất kỳ ngành công nghiệp nào khác, các hệ thống đều có kho thông tin và tài liệu nội bộ khổng lồ.

Đáng tiếc là, quy mô dữ liệu của nhiều công ty đồng nghĩa với việc tổ chức và truy cập thông tin có thể trở nên cực kỳ kém hiệu quả. Vấn đề càng trở nên trầm trọng hơn đối với thông tin dựa trên ngôn ngữ. Ngôn ngữ là công cụ để con người giao tiếp các ý tưởng và khái niệm trừu tượng. Tự nhiên, các ý tưởng và khái niệm khó hơn cho máy tính để hiểu và lưu trữ một cách có ý nghĩa.

Hầu hết các hệ thống dựa vào cụm giao diện tìm kiếm dựa trên từ khóa được lưu trữ trên các 'cổng nội bộ' khác nhau để xử lý dữ liệu ngôn ngữ. Nếu được thực hiện tốt, điều này có thể đáp ứng yêu cầu kinh doanh cho một số dữ liệu đó.

Nếu một người biết họ đang tìm kiếm gì và họ biết các từ khóa và thuật ngữ của thông tin họ cần, tìm kiếm dựa trên từ khóa là lý tưởng. Khi từ khóa và thuật ngữ của câu trả lời không được biết, tìm kiếm từ khóa là không đủ. Việc mọi người tìm kiếm câu trả lời không rõ ràng trong các kho tài liệu lớn là một sự lãng phí năng suất [13].

Làm thế nào để chúng ta giảm thiểu vấn đề này? Câu trả lời nằm ở tìm kiếm ngữ nghĩa, đặc biệt là với khía cạnh hỏi-đáp (QA) của tìm kiếm ngữ nghĩa.

1.2.2 Hệ thống truy xuất thông tin

Trước khi đi sâu vào các hệ thống Hỏi Đáp, hãy thảo luận về các hệ thống Truy Xuất Thông Tin (Information Retrieval). Truy xuất thông tin trong khoa học thông tin là nhiệm vụ xác định và truy xuất các nguồn tài nguyên hệ thống thông tin có liên quan đến một nhu cầu thông tin. Nhu cầu thông tin có thể được chỉ định dưới dạng một truy vấn tìm kiếm. Trong trường hợp truy xuất tài liệu, các truy vấn có thể dựa trên toàn văn hoặc lập chỉ mục dựa trên nội dung khác [14]. Truy xuất thông tin là khoa học của việc tìm kiếm thông tin trong một tài liệu, tìm kiếm các tài liệu tự chúng và cũng tìm kiếm siêu dữ liệu (metadata) mô tả dữ liệu, và các cơ sở dữ liệu văn bản, hình ảnh hoặc âm thanh.

Quá trình truy xuất thông tin bắt đầu khi người dùng nhập truy vấn vào hệ thống. Truy vấn là những tuyên bố chính thức về nhu cầu thông tin, ví dụ như chuỗi tìm kiếm trong các công cụ tìm kiếm trên web. Trong truy xuất thông tin, một truy vấn không xác

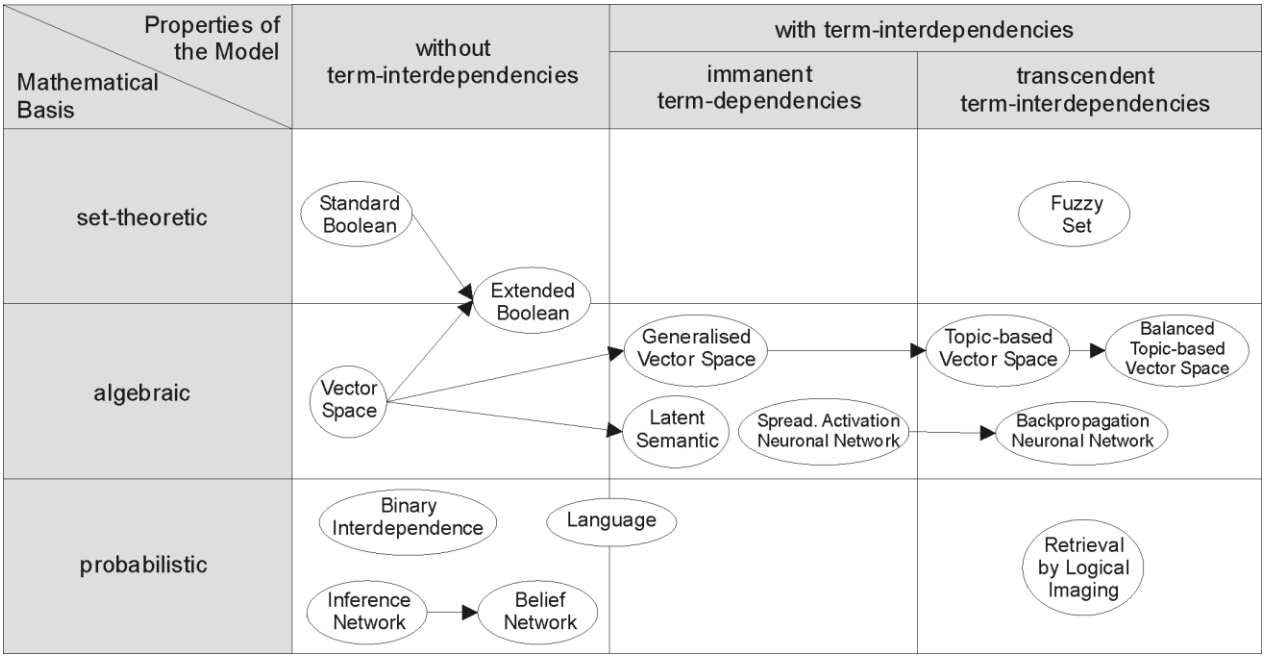
định duy nhất một đối tượng trong bộ sưu tập. Thay vào đó, nhiều đối tượng có thể khớp với truy vấn, có thể với các mức độ liên quan khác nhau [15].

Một đối tượng là một thực thể được đại diện bởi thông tin trong một bộ sưu tập nội dung hoặc cơ sở dữ liệu. Các truy vấn của người dùng được đối chiếu với thông tin trong cơ sở dữ liệu. Tuy nhiên, khác với các truy vấn SQL cổ điển của một cơ sở dữ liệu, trong truy xuất thông tin, các kết quả trả về có thể hoặc không khớp với truy vấn, vì vậy các kết quả thường được xếp hạng. Việc xếp hạng kết quả này là một điểm khác biệt chính của tìm kiếm truy xuất thông tin so với tìm kiếm cơ sở dữ liệu.

Tùy thuộc vào ứng dụng, các đối tượng dữ liệu có thể là, ví dụ, tài liệu văn bản, hình ảnh, âm thanh, bản đồ tư duy hoặc video. Thường thì các tài liệu không được lưu giữ hoặc lưu trữ trực tiếp trong hệ thống IR, mà được đại diện trong hệ thống bởi các đại diện tài liệu hoặc siêu dữ liệu.

Hầu hết các hệ thống IR tính toán một điểm số số hóa về mức độ khớp của mỗi đối tượng trong cơ sở dữ liệu với truy vấn và xếp hạng các đối tượng theo giá trị này. Các đối tượng xếp hạng cao nhất sau đó được hiển thị cho người dùng. Quá trình này có thể được lặp lại nếu người dùng muốn tinh chỉnh truy vấn [16].

Để truy xuất hiệu quả các tài liệu liên quan bằng các phương pháp Truy Xuất Thông Tin, các tài liệu thường được biến đổi thành một dạng đại diện phù hợp. Mỗi phương pháp truy xuất kết hợp một mô hình cụ thể cho mục đích đại diện tài liệu của nó. Hình ảnh bên dưới minh họa mối quan hệ của một số mô hình phổ biến. Trong hình ảnh, các mô hình được phân loại theo hai chiều: cơ sở toán học và các thuộc tính của mô hình [17].



Hình 1.1 Mối quan hệ giữa một số mô hình truy xuất thông tin

Chiều cơ sở toán học:

- Các mô hình lý thuyết (set-theoretic) tập hợp đại diện cho các tài liệu dưới dạng tập hợp các từ hoặc cụm từ. Các điểm tương đồng thường được suy ra từ các phép toán lý thuyết tập hợp trên các tập hợp đó. Các mô hình phổ biến là: Mô hình Boolean tiêu chuẩn (Standard Boolean model), Mô hình Boolean mở rộng (Extended Boolean model), Truy xuất mờ (Fuzzy retrieval).
- Các mô hình đại số đại diện cho tài liệu và truy vấn thường dưới dạng vector, ma trận, hoặc tuple. Sự tương đồng giữa vector truy vấn và vector tài liệu được biểu diễn dưới dạng giá trị vô hướng.
- Các mô hình xác suất coi quá trình truy xuất tài liệu như là suy luận xác suất. Sự tương đồng được tính toán như là xác suất mà một tài liệu có liên quan đến một truy vấn cụ thể. Các định lý xác suất như định lý Bayes thường được sử dụng trong các mô hình này.
- Các mô hình truy xuất dựa trên đặc trưng xem các tài liệu như là các vector giá trị của các hàm đặc trưng (hoặc chỉ là đặc trưng) và tìm cách tốt nhất để kết hợp các đặc trưng này thành một điểm số liên quan đơn lẻ, thường bằng các phương pháp học xếp hạng. Các hàm đặc trưng là các hàm tùy ý của tài liệu và truy vấn, và do đó có thể dễ dàng kết hợp gần như bất kỳ mô hình truy xuất nào khác như chỉ là một đặc trưng khác.

Chiều thuộc tính của mô hình:

- Các mô hình không có sự phụ thuộc giữa các thuật ngữ: coi các thuật ngữ/từ khác nhau là độc lập. Thực tế này thường được biểu diễn trong các mô hình không gian vector bằng giả định trực giao của các vector thuật ngữ hoặc trong các mô hình xác suất bằng giả định độc lập cho các biến thuật ngữ.
- Các mô hình có sự phụ thuộc tiềm ẩn giữa các thuật ngữ: cho phép biểu diễn sự phụ thuộc giữa các thuật ngữ. Tuy nhiên, mức độ phụ thuộc giữa hai thuật ngữ được xác định bởi chính mô hình đó. Thường thì điều này được suy ra trực tiếp hoặc gián tiếp (ví dụ, bằng cách giảm chiều) từ sự đồng xuất hiện của các thuật ngữ đó trong toàn bộ tập hợp tài liệu.
- Các mô hình có sự phụ thuộc siêu việt giữa các thuật ngữ: cho phép biểu diễn sự phụ thuộc giữa các thuật ngữ, nhưng không khẳng định cách thức xác định mức độ phụ thuộc giữa hai thuật ngữ. Họ dựa vào một nguồn bên ngoài để xác định mức độ phụ thuộc giữa hai thuật ngữ (ví dụ, một con người hoặc các thuật toán phức tạp).

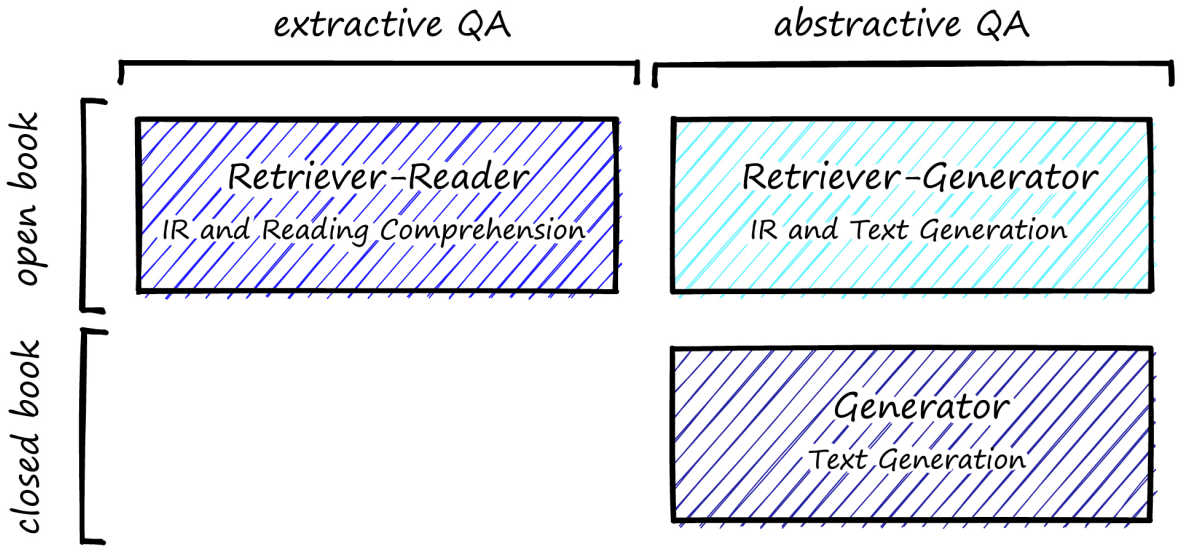
Một nhánh mở rộng tự nhiên của các hệ thống Truy Xuất Thông Tin là hệ thống Hỏi Đáp, không chỉ lấy các tài liệu đã được xếp hạng mà còn trả về một phần được chọn của tài liệu – tương tự như Google hiện nay.

1.2.3 Hệ thống hỏi đáp

Hệ thống Hỏi Đáp là một nhánh tự nhiên của hệ thống Truy xuất thông tin và xử lý ngôn ngữ tự nhiên liên quan đến việc xây dựng các phương pháp tự động trả lời các câu hỏi được đặt ra bởi con người bằng ngôn ngữ tự nhiên. Hệ thống Hỏi Đáp tìm kiếm câu trả lời trong cơ sở tri thức dựa trên các truy vấn của người dùng, với các loại câu trả lời khác nhau tùy thuộc vào ứng dụng của hệ thống, chẳng hạn như truy vấn cơ sở dữ liệu, truy xuất thông tin hoặc các phương pháp dựa trên đồ thị tri thức. Câu trả lời có thể là một từ, một đoạn câu, một câu hoàn chỉnh và có ý nghĩa, hoặc một tập hợp các câu với tính logic. Loại câu trả lời phụ thuộc vào ứng dụng mà hệ thống Hỏi Đáp được phát triển. Hệ thống có thể được phát triển trên các mô hình khác nhau: truy vấn cơ sở dữ liệu, truy xuất thông tin và dựa trên đồ thị tri thức. Phương pháp truy vấn cơ sở dữ liệu thường liên quan đến việc phát triển một cơ sở dữ liệu cặp câu hỏi-câu trả lời cho một lĩnh vực cụ thể và sau đó tìm kiếm câu trả lời dựa trên câu hỏi của người dùng. Truy xuất thông tin thường là tìm kiếm dữ liệu không có cấu trúc, tức là văn bản, đáp ứng yêu cầu của người

dùng từ các tập dữ liệu lớn, thường là trên Web. Phương pháp dựa trên đồ thị tri thức liên quan đến việc phân tích ngữ nghĩa của truy vấn và sau đó truy cập vào cơ sở dữ liệu có cấu trúc. Cơ sở dữ liệu có thể là một cơ sở dữ liệu quan hệ hoàn chỉnh hoặc các cơ sở dữ liệu có cấu trúc đơn giản.

Trước khi đi vào chi tiết, hãy xét một bức tranh tổng quát về Hỏi Đáp (QA). Đầu tiên, chúng ta tập trung vào Hệ thống Hỏi Đáp Miền Mở (Open Domain Question Answering System). Các hệ thống ODQA xử lý các câu hỏi trên nhiều chủ đề rộng lớn và không thể dựa vào một tập hợp các quy tắc cụ thể . Thứ hai là Hệ thống Hỏi Đáp Miền Đóng (Closed-Domain Question Answering System), tập trung vào một lĩnh vực/phạm vi hạn chế và thường có thể dựa vào logic rõ ràng. Các hệ thống Hỏi Đáp Mở và Hỏi Đáp Đóng còn được biết đến là các hệ thống sách mở (Open Book) và sách đóng (Closed Book) [18].



Hình 1.2 Hình một số cách tiếp cận của Hệ thống Hỏi Đáp

Hình thức phổ biến nhất của hệ thống Hỏi Đáp là Exractive QA (trên cùng bên trái). Ở đây, chúng ta kết hợp bước Truy Xuất Thông Tin (IR) và bước Đọc hiểu (Reading comprehension). Bất kỳ hệ thống ODQA nào đều yêu cầu một bước IR để truy xuất thông tin liên quan từ "cuốn sách mở". Giống như trong các kỳ thi mở sách, nơi sinh viên có thể tham khảo sách của họ để lấy thông tin trong kỳ thi, mô hình có thể tham khảo một nguồn thông tin bên ngoài. Nguồn thông tin đó có thể là các tài liệu nội bộ của tổ chức, Wikipedia, Reddit, hoặc bất kỳ nguồn thông tin nào khác không phải là chính mô hình. Bước IR truy xuất các tài liệu liên quan và chuyển chúng sang bước RC (người đọc). RC bao gồm việc trích xuất một câu trả lời ngắn gọn từ một câu hoặc đoạn văn, thường được

gọi là tài liệu hoặc ngữ cảnh.

Hai loại Hỏi Đáp khác dựa vào việc tạo ra câu trả lời thay vì trích xuất chúng. Các mô hình GPT của OpenAI là những ví dụ về mô hình máy biến đổi tạo sinh nổi tiếng.

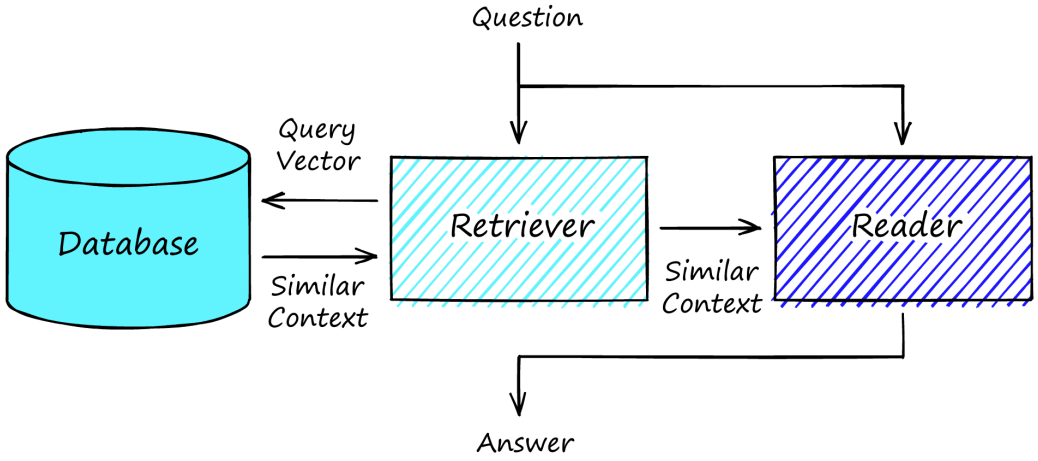
Trong Hỏi Đáp Mở Sách Tóm Lược (open-book abstractive QA), bước IR đầu tiên giống như trong Extractive QA; các ngữ cảnh liên quan được truy xuất từ một nguồn bên ngoài. Những ngữ cảnh này được chuyển tới mô hình sinh văn bản (chẳng hạn như GPT) và được sử dụng để tạo ra (không phải trích xuất) một câu trả lời.

Ngoài ra, chúng ta có thể sử dụng Hỏi Đáp Đóng Sách Tóm Lược. Ở đây chỉ có mô hình sinh văn bản và không có bước IR. Mô hình sinh sẽ tạo ra câu trả lời dựa trên biểu diễn học nội tại của nó về thế giới. Nó không thể tham khảo bất kỳ nguồn thông tin bên ngoài nào, do đó có tên gọi là ‘closed-book’.

QA trích xuất có thể được coi là hình thức hỏi đáp dễ áp dụng nhất. Nó cho phép chúng ta đặt một câu hỏi và sau đó trích xuất một câu trả lời từ một đoạn văn ngắn. Trong phương pháp này, chúng ta đưa ra một ngữ cảnh duy nhất và trích xuất một câu trả lời thông qua việc đọc hiểu (RC). Sau đó, chúng ta có thể kết hợp nó với một nguồn dữ liệu bên ngoài và tìm kiếm qua nhiều ngữ cảnh, không chỉ một. Chúng ta gọi phương pháp này là 'QA trích xuất mở sách', thường được gọi tắt là QA trích xuất. Nó không phải là một mô hình đơn lẻ mà thực ra bao gồm ba thành phần:

- Dữ liệu được lập chỉ mục (kho tài liệu / cơ sở dữ liệu vector)
- Mô hình truy xuất
- Mô hình đọc

ODQA yêu cầu lập chỉ mục dữ liệu mà mô hình truy xuất của chúng ta có thể truy cập sau này. Thông thường, đó sẽ là các đoạn văn bản có kích thước ‘sentence-to-paragraph-sized’ [19].



Hình 1.3 Pipeline của hệ thống Extractive QA

Các tùy chọn cho loại cơ sở dữ liệu thay đổi dựa trên mô hình truy xuất. Một mô hình truy xuất truyền thống sử dụng truy xuất sparse vector với TF-IDF hoặc BM25. Các mô hình này trả về các ngữ cảnh dựa trên tần suất của các từ khớp giữa ngữ cảnh và câu hỏi. Nhiều từ khớp hơn tương đương với mức độ liên quan cao hơn. Elasticsearch là giải pháp cơ sở dữ liệu phổ biến nhất cho việc này nhờ khả năng tìm kiếm từ khóa mạnh mẽ và có thể mở rộng của họ [20]. Tùy chọn khác là sử dụng truy xuất vector dày đặc với các vector câu được xây dựng bởi các mô hình biến đổi như BERT. Các vector dày đặc có lợi thế cho phép tìm kiếm thông qua ngữ nghĩa.

1.3 Tìm kiếm mức độ tương tự văn bản

1.3.1 Thuật toán TF-IDF

TF-IDF là viết tắt của cụm từ: Term frequence -Inverse document frequency . Đây là một kĩ thuật rất nổi tiếng, được sử dụng trong nhiều bài toàn NLP và khai phá dữ liệu dạng văn bản với mục đích: tính weight (độ quan trọng) của word trong một văn bản cụ thể, văn bản đó nằm trong một tập nhiều văn bản khác nhau [21]. Bản thân chính tên gọi này đã thể hiện được nội dung thuật toán:

- Term frequence: tần suất thuật ngữ xuất hiện trong tài liệu.(3 lần? 30 lần?)
- Inverse document frequency: được tính bằng số lượng tài liệu mà thuật ngữ xuất hiện. Tần suất tài liệu nghịch đảo (1 / df) cho biết mức độ quan trọng của thuật ngữ. Thuật ngữ có phải là một từ hiếm (chỉ xảy ra trong một tài liệu) hay không? Hay thuật ngữ này phổ biến (xảy ra trong gần như tất cả các tài liệu)?

Ý tưởng của thuật toán rất đơn giản: một từ xuất hiện nhiều trong 1 văn bản, nhưng lại ít xuất hiện trong cả dataset thì độ quan trọng của nó càng cao. Ngược lại, một từ xuất hiện ở hầu hết các văn bản thì có độ quan trọng càng thấp [22]. Bằng thuật toán này, người ta có thể xác định ra các stop word (mang ít ngữ nghĩa như: a, an, the ...).

Ta giả sử:

- t: Từ (term) đang xét.
- d: Tài liệu (document) đang xét.
- nd: Tổng số từ trong tài liệu d.

TF: Term frequence - tần xuất xuất hiện của t trong d. Một từ t xuất hiện trong văn bản d càng nhiều thì độ quan trọng của nó càng cao. Ta có:

$$\mathrm{TF}(t,d) = \frac{f_{t,d}}{n_d}$$

Ý tưởng của IDF: một từ xuất hiện trên càng nhiều văn bản, độ quan trọng từ đó càng giảm. VD các stop word như a, an, the, i .... xuất hiện trên hầu hết các văn bản, ngữ cảnh, độ quan trọng của những từ này thường rất thấp. Ta có:

$$\mathrm{IDF}(t,D) = \log\left(\frac{|D|}{1 + |d \in D : t \in d|}\right)$$

Trong đó:

- t: Từ (term) đang xét.
- D: Tập hợp tất cả các tài liệu.
- |D|: Tổng số tài liệu trong tập D.
- |d ∈ D : t ∈ d|: Số tài liệu chứa từ t.
- 1 trong mẫu số được thêm vào để tránh chia cho 0 khi t không xuất hiện trong tài liệu nào.

Bước đầu tiêu của các thuật toán search nói chung, ta cần phải vectorize các document (vector hoá). Tức cần phải chuyển đổi dạng văn bản dạng string thành vector. Phương pháp đơn giản và thông thường nhất là bag of word . Giả sử ta có 1 dataset gồm 5 câu [23]:

“There used to be Stone Age”

“There used to be bronze age”

“There used to be Iron Age”

“There was age of revolution”

“Now it is Digital Age”

Toàn bộ dataset có 15 từ vựng --> mỗi văn bản là 1 vector có độ dài 15. Với 1 văn bản, từ vựng nào xuất hiện thì giá trị tại đó bằng 1, từ nào không xuất hiện thì giá trị bằng 0. Sau khi "bag of word", ta có thu được kết quả:

There used to be bronze age = [1,0,1,1,1,0,1,0,0,0,1,0,0,0,0]

There used to be iron age = [1,0,1,1,1,0,0,1,0,0,1,0,0,0,0]

There was age of revolution = [1,1,0,0,0,0,0,0,1,0,1,1,0,0,0]

Now its digital Age = [0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,1]

Sử dụng tf-idf vectorization, phương pháp này giúp cải thiện kết quả đầu ra rất nhiều. Phương pháp này rất dễ hiểu, với 1 document d, thay vì gán giá trị 1 cho các từ vựng xuất

hiện trong d, ta gán giá trị tf-idf tương ứng của từ đó trong d. VD sau khi tính toán, ta được (giá trì vector bên dưới chỉ là minh họa, không phải từ kết quả tính toán):
"There used to be bronze age" = [0.23 , 0 ,0.22 , 0.13, 1.2 ,0, 1.4 , 0,0,0, 0.99 , 0, 0, 0, 0]

1.3.2 Thuật toán BM25

Trong tìm kiếm thông tin, Okapi BM25 là hàm tính thứ hạng được các công cụ tìm kiếm sử dụng để xếp hạng các văn bản theo độ phù hợp với truy vấn nhất định. Hàm xếp hạng này dựa trên mô hình xác suất, được phát minh ra vào những năm 1970 – 1980. Phương pháp có tên BM25 (BM – best match), nhưng người ta thường gọi "Okapi BM25", vì lần đầu tiên công thức được sử dụng trong hệ thống tìm kiếm Okapi, được sáng lập tại trường đại học London những năm 1980 và 1990 [24].

BM25 là một phương pháp xếp hạng tựa như TF-IDF, được sử dụng rộng rãi trong tìm kiếm. Trong Web search những hàm xếp hạng này thường được sử dụng như một phần của các phương pháp tích hợp để dùng trong machine learning, xếp hạng. Một trong những kỹ thuật tìm kiếm nỗi tiếng hiện nay đang sử dụng thuật toán này là Elasticsearch. Khi tìm kiếm, Elascticsearch trả về cho mình ngoài các kết quả tìm được, còn có đánh giá độ liên quan của kết quả dựa trên giá trị thực dương score. Elasticsearch sẽ sắp xếp các kết quả trả về của các query theo thứ tự score giảm dần [25].

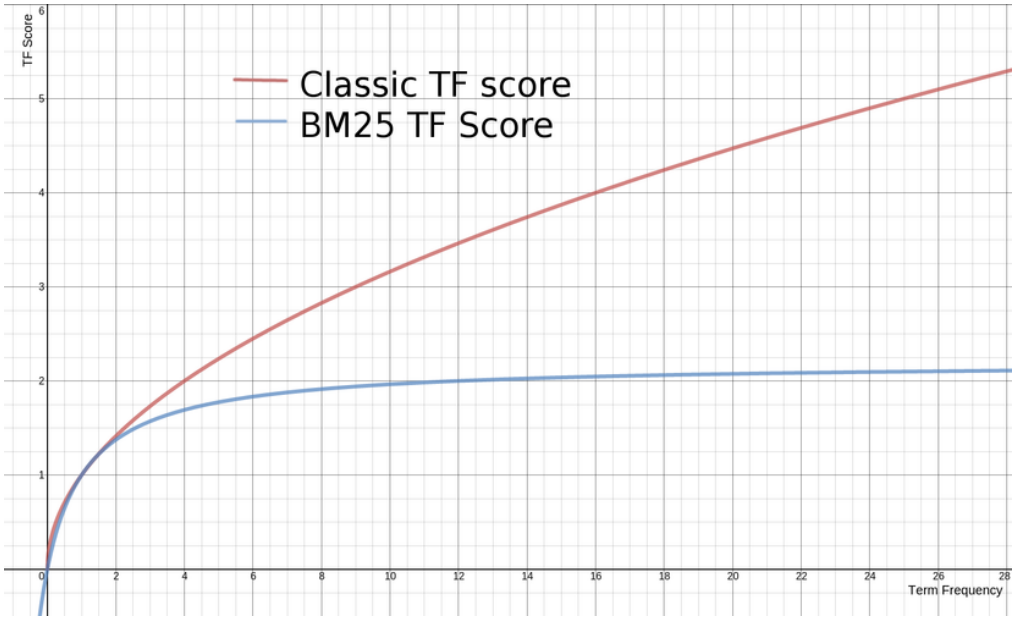
Nếu một thuật ngữ phổ biến trong tài liệu này, nhưng hiếm ở tài liệu khác, thì điểm TFIDF sẽ cao và tài liệu có điểm TF-IDF cao hơn sẽ được coi là phù hợp với cụm từ tìm kiếm. BM25 cải thiện dựa trên TF-IDF bằng cách sử dụng mức độ liên quan với một bài toán xác suất. BM25 sẽ đưa ra điểm liên quan, để xác định xem một truy vấn có mức độ liên quan thế nào đến các tài liệu. Sau đó xếp hạng các điểm liên quan đó để đưa ra kết quả các tài liệu phù hợp với truy vấn. Để xác định mức độ liên quan giữa một truy vấn (tài liệu) với một tài liệu khác, chúng ta có thể sử dụng công thức tính BM25 như sau:

$$\mathrm{BM25}(t,d) = \sum_{t \in q} \mathrm{IDF}(t) \cdot \frac{f_{t,d} \cdot (k_1 + 1)}{f_{t,d} + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)}$$

Trong đó:

- t: Từ (term) đang xét.
- d: Tài liệu (document) đang xét.
- q: Truy vấn (query) chứa tập hợp các từ t.
- $(f_{t,d})$: Số lần từ t xuất hiện trong tài liệu d (tần suất từ).

- $|d|$: Độ dài của tài liệu d (tổng số từ trong tài liệu).
- Avgdl : Độ dài trung bình của các tài liệu trong tập.
- k_1 : Tham số điều chỉnh độ nhạy của tần suất từ, với $k_1 > 0$ và thường nằm trong khoảng [1.2 , 2.0].
- b : Tham số điều chỉnh độ ảnh hưởng của độ dài tài liệu, với b ∈ [0,1]. Thông thường, b=0.75. b và k_1
- IDF(t) : Trọng số nghịch đảo tần suất tài liệu của từ t, được tính theo công thức ở phần trên.



Hình 1.4 Sự ảnh hưởng của TF tới TF Score

Tham số k_1 của BM25 xác định tính bão hòa tần suất từ. Giá trị càng cao, độ bão hòa càng chậm. Nghĩa là nếu một từ xuất hiện nhiều sẽ làm điểm của tài liệu cao, nhưng sẽ nhiều với một mức độ nào đó và mức độ ảnh hưởng tới điểm sẽ giảm dần. $\frac{|d|}{\text{avgdl}}$ ở mẫu số có nghĩa là tài liệu dài hơn các tài liệu trung bình sẽ dẫn đến mẫu số lớn hơn, dẫn đến giảm điểm. Tham khảo các trường hợp thực tế, rút ra được rằng nếu càng nhiều thuật ngữ trong tài liệu mà không khớp với truy vấn đầu vào thì điểm của tài liệu càng thấp [26]. Nói cách khác, nếu một tài liệu dài 300 trang đề cập đến cụm từ truy vấn một lần, thì nó ít có khả năng liên quan đến truy vấn hơn so với một tài liệu ngắn đề cập đến truy vấn một lần.

1.3.3 Tìm kiếm tương tự dựa trên vector nhúng từ

Công việc tìm kiếm tương tự là một khái niệm chính trong nhiều hệ thống truy xuất

thông tin, công cụ đề xuất, tìm kiếm từ đồng nghĩa... Nó được sử dụng ở khắp mọi nơi mà việc tìm kiếm chính xác thông qua cơ sở dữ liệu không phải là một lựa chọn khả thi (ví dụ, ta không thể vừa lưu ý nghĩa và ngữ cảnh của từ). Trong trường hợp đó, hầu hết các công cụ, thuật toán học sâu hoặc các phương pháp học máy thống kê hoạt động với các biểu diễn ban đầu của các mục dưới dạng các vector trong không gian chiều cao, được gọi là các embedding. Điều này cho phép biểu diễn các mục ban đầu (như văn bản, hình ảnh, v.v.) một cách hiệu quả và linh hoạt hơn, cũng như lưu giữ các đặc trưng của nó và đôi khi cả ngữ cảnh. Các vector embedding đã chứng minh được là công cụ hiệu quả trong nhiều lĩnh vực, bao gồm xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo. So sánh các vector embedding và xác định sự tương đồng của chúng là một phần thiết yếu của tìm kiếm ngữ nghĩa, hệ thống đề xuất, phát hiện bất thường và nhiều ứng dụng khác [27].

Việc cơ bản việc tìm kiếm các mục tương tự có nghĩa là tìm kiếm các biểu diễn vector của chúng mà gần nhau trong không gian biểu diễn của chúng (tìm khoảng cách Euclidean hoặc các loại khoảng cách khác giữa chúng). Việc tìm kiếm tương tự sẽ theo các bước:

- Đại diện cho tất cả các mục từ tập dữ liệu dưới dạng các embedding
- Định nghĩa một metric cho một cặp embedding. Biện pháp này có thể là độ tương đồng cosine, khoảng cách Euclidean hoặc tích vô hướng.
- Tìm kiếm embedding của đối tượng truy vấn
- Chọn các embedding gần với đối tượng truy vấn

Việc truy xuất những kết quả này là một tìm kiếm k-láng giềng gần nhất (k-nearest neighbours), có thể được thực hiện theo một số cách khác nhau [28]. Nếu không có nhiều mục trong tập dữ liệu (vài trăm hoặc vài nghìn), có thể tính toán metric khoảng cách giữa tất cả các vector. Tuy nhiên, nếu kết quả cần thiết trong thời gian thực, và tập dữ liệu thực sự lớn, việc tìm kiếm những láng giềng gần nhất cần phải được ước tính. Điều này có thể được thực hiện bằng cách tạo cấu trúc dữ liệu chỉ số cho phép tìm kiếm nhanh các mục. Ta sẽ xem xét sâu hơn cách các metric tương đồng này hoạt động để có nắm bắt về ý nghĩa của việc hai vector embedding là tương tự nhau, trong bối cảnh của một trường hợp sử dụng cụ thể, ở đây là Chatbot hỏi đáp. Như trong xử lý ngôn ngữ tự nhiên, hai vector biểu diễn ý nghĩa từ có thể "gần" nhau nếu chúng được sử dụng trong các ngữ cảnh tương tự hoặc liên quan đến các ý tưởng tương tự.

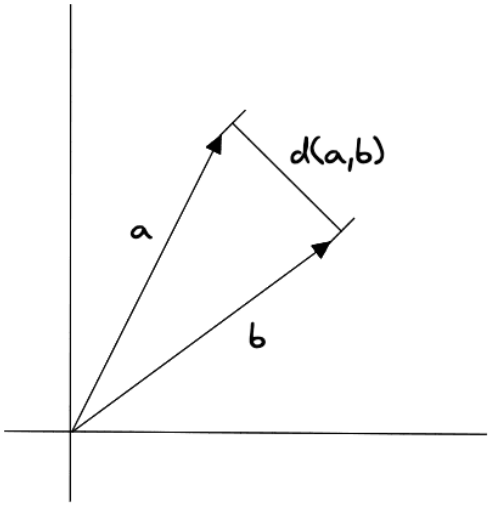
Bảng 1.1 Các metric đo tương đồng

Metric đo độ tương tự	**Thuộc tính vector được xem xét**
Khoảng cách Euclidean	Độ lớn và hướng
Độ tương đồng cosine	Chỉ có hướng
Độ tương đồng tích vô hướng	Độ lớn và hướng

Khoảng cách Euclidean được tính bằng căn bậc hai của tổng các bình phương của các sự chênh lệch giữa các thành phần tương ứng của các vector. Dưới đây là phương trình tính khoảng cách Euclidean giữa hai vector p và q trong không gian n-chiều:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Metric này nhạy cảm với tỷ lệ cũng như vị trí tương đối của các vector trong không gian. Điều này có nghĩa là các vector có giá trị lớn sẽ có khoảng cách Euclidean lớn hơn so với các vector có giá trị nhỏ, ngay cả khi các vector đó tương tự nhau theo cách khác.



Hình 1.5 Ví dụ đơn giản về khoảng cách Euclidean trong không gian 2 chiều

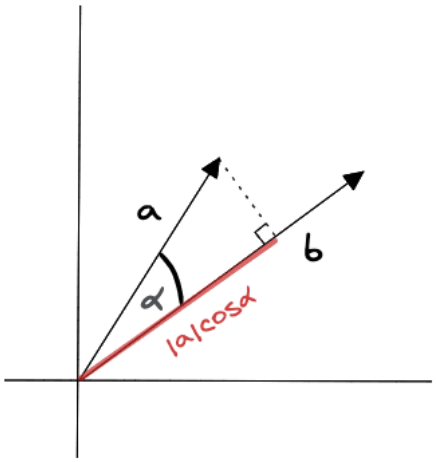
Khoảng cách Euclidean là một metric tương đồng rất đơn giản vì nó phản ánh khoảng cách giữa từng giá trị của các vector được so sánh: nếu khoảng cách Euclidean rất nhỏ, thì các giá trị của từng tọa độ trong các vector rất gần nhau. Điều này khác biệt về tổng thể đối với metric tích vô hướng hoặc cosine. [29] Vì khoảng cách Euclidean nhạy cảm với độ lớn, nó hữu ích khi các embedding chứa thông tin liên quan đến số

lượng hoặc các phép đo của các thứ. Nói chung, khoảng cách Euclidean là một lựa chọn tự nhiên khi mô hình không được huấn luyện với một hàm mất mát cụ thể.

Tích vô hướng của hai vectơ a và b trong không gian n-chiều được tính như sau:

$$a \cdot b = \sum_{i=1}^{n} a_i b_i$$

Tích vô hướng là một giá trị vô hướng, có nghĩa là nó là một số duy nhất thay vì một vector. Tích vô hướng là dương nếu góc giữa các vector nhỏ hơn 90 độ, âm nếu góc giữa các vector lớn hơn 90 độ, và bằng không nếu các vector vuông góc.



Hình 1.6 Ví dụ đơn giản về tích vô hướng trong không gian 2 chiều

Tích vô hướng có thể bị ảnh hưởng bởi độ dài và hướng của các vector. Khi hai vector có cùng độ dài nhưng hướng khác nhau, tích vô hướng sẽ lớn hơn nếu hai vector chỉ về cùng một hướng và nhỏ hơn nếu chúng chỉ về các hướng đối diện. Hãy tưởng tượng hai vector được biểu diễn bằng các mũi tên, vector a và vector b. Nếu các vector a và b chỉ về cùng một hướng, tích vô hướng của a và b sẽ lớn hơn so với khi a và b chỉ về các hướng ngược nhau. Nhiều mô hình ngôn ngữ lớn sử dụng tích vô hướng trong quá trình huấn luyện. Tích vô hướng là một phần quan trọng trong các mô hình học sâu, đặc biệt khi tính toán độ tương đồng giữa các vector trong không gian biểu diễn cao [30]. Ví dụ, mô hình GPT của OpenAI sử dụng tích vô hướng trong quá trình xử lý các embedding để xác định mức độ liên quan giữa các từ trong ngữ cảnh. Điều này giúp mô hình hiểu và dự đoán các từ tiếp theo dựa trên ngữ cảnh hiện tại.

Độ tương đồng cosine là một thước đo của góc giữa hai vector. Nó được tính bằng cách lấy tích vô hướng của các vector và chia nó cho tích của độ lớn của chúng. [31] Metric này không bị ảnh hưởng bởi kích thước của vector mà chỉ bởi góc giữa chúng.

Điều này có nghĩa là các vector có giá trị lớn hoặc nhỏ sẽ có độ tương đồng cosine giống nhau miễn là chúng chỉ về cùng một hướng. Dưới đây là cách tính độ tương đồng cosine cho các vector a và b:

$$\text{Cosine Similarity}(a, b) = \frac{a \cdot b}{|a||b|}$$

Trong đó a và b là các vector được so sánh, "•" biểu thị tích vô hướng, và |a| và |b| biểu thị độ dài của các vector. Độ tương đồng cosine nằm trong khoảng từ -1 đến 1, trong đó 1 có nghĩa là góc 0 độ (các vector gần nhau nhất có thể), 0 có nghĩa là vuông góc, và -1 có nghĩa là các vector chỉ về các hướng ngược lại.

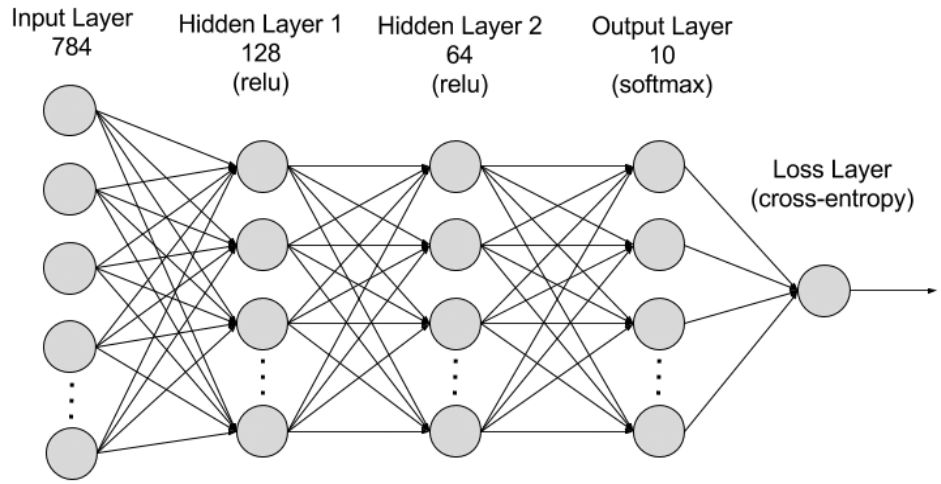
Đầu tiên, phương trình tính tích vô hướng của các vector bằng cách nhân các thành phần của chúng và cộng kết quả lại. Tích vô hướng sau đó được chia cho tích của độ lớn của các vector, được tính bằng cách lấy căn bậc hai của tổng các bình phương của các thành phần của các vector. Nếu mô hình được huấn luyện bằng độ tương đồng cosine, ta có thể sử dụng độ tương đồng cosine hoặc chuẩn hóa và sử dụng tích vô hướng [32]. Độ tương đồng cosine có lẽ không phù hợp khi có dữ liệu mà độ lớn của các vector là quan trọng và cần được xem xét khi xác định sự tương đồng.

Nguyên tắc chung khi chọn metric tìm kiếm tương đồng: sử dụng metric tương đồng giống như metric đã được sử dụng để huấn luyện mô hình embedding.

1.4 Lịch sử hình thành mô hình ngôn ngữ

1.4.1 Mạng Neuron hồi quy truyền thống (RNN)

Mạng nơ-ron hồi quy (RNN) là một mô hình học sâu được đào tạo để xử lý và chuyển đổi đầu vào dữ liệu tuần tự thành đầu ra dữ liệu tuần tự cụ thể. Dữ liệu tuần tự là dữ liệu, chẳng hạn như từ, câu hoặc dữ liệu chuỗi thời gian, trong đó các thành phần tuần tự tương quan với nhau dựa trên ngữ nghĩa phức tạp và quy tắc cú pháp. RNN là một hệ thống phần mềm gồm nhiều thành phần được kết nối với nhau theo cách con người thực hiện chuyển đổi dữ liệu tuần tự, chẳng hạn như dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác. [33] Phần lớn RNN đang được thay thế bằng trí tuệ nhân tạo dựa trên công cụ biến đổi và các mô hình ngôn ngữ lớn, hiệu quả hơn nhiều trong việc xử lý dữ liệu tuần tự.



Hình 1.7 Sơ đồ hoạt động của một RNN

RNN được tạo thành từ các nơ-ron: các nút xử lý dữ liệu kết hợp cùng nhau để thực hiện các tác vụ phức tạp. Các nơ-ron được tổ chức dưới dạng lớp đầu vào, đầu ra và ẩn. Lớp đầu vào nhận thông tin để xử lý và lớp đầu ra cung cấp kết quả [34]. Quá trình xử lý dữ liệu, phân tích và dự đoán diễn ra trong lớp ẩn.

RNN hoạt động bằng cách lần lượt truyền dữ liệu tuần tự nhận được đến các lớp ẩn. Tuy nhiên, RNN cũng có quy trình làm việc tự lặp lại hay *hồi quy*: lớp ẩn có thể ghi nhớ và sử dụng các đầu vào trước đó cho các dự đoán trong tương lai trong một thành phần bộ nhớ ngắn hạn. [35] Quy trình này sử dụng đầu vào hiện tại và bộ nhớ đã lưu trữ để dự đoán chuỗi tiếp theo. Ví dụ, ta có câu: *Quả táo màu đỏ.* Ta muốn RNN dự đoán màu đỏ khi nhận được chuỗi đầu vào *Quả táo màu.* Khi xử lý từ *Quả táo*, lớp ẩn sẽ lưu trữ một bản sao trong bộ nhớ. Tiếp theo, khi thấy từ *màu*, lớp ẩn gọi lại *Quả táo* từ bộ nhớ của mình và hiểu toàn bộ chuỗi: *Quả táo màu* là ngữ cảnh. Sau đó, lớp ẩn có thể dự đoán *màu đỏ* để cải thiện độ chính xác. Do đó, RNN trở nên hữu ích trong nhận dạng giọng nói, dịch máy và các tác vụ nhận diện ngữ cảnh.

Quá trình đào tạo học máy (ML) các mạng nơ-ron sâu như RNN bằng cách cung cấp dữ liệu đào tạo cho mô hình và tinh chỉnh hiệu năng của mô hình. Trong ML, trọng số của neuron là tín hiệu để xác định mức độ ảnh hưởng của thông tin đã học trong quá trình đào tạo khi dự đoán đầu ra. Mỗi lớp trong RNN đều có trọng số bằng nhau. Ta điều chỉnh trọng số để dự đoán chính xác hơn sử dụng một kỹ thuật gọi là truyền ngược qua thời gian (BPTT) để tính lỗi mô hình và điều chỉnh trọng số của mô hình cho phù hợp. BPTT khôi phục đầu ra về bước thời gian trước và tính lại tỷ lệ lỗi. [36] Qua đó, kỹ thuật này có thể xác định trạng thái ẩn nào trong chuỗi đang gây ra lỗi đáng kể và điều chỉnh

lại trọng số để giảm biên lỗi.

RNN thường có đặc trưng là kiến trúc một-một: một chuỗi đầu vào được liên kết với một đầu ra. Tuy nhiên, nó cũng có thể được điều chỉnh linh hoạt thành các cấu hình khác nhau cho các mục đích cụ thể. Sau đây là một số loại RNN phổ biến:

- Một-nhiều**:** Loại RNN này dẫn một đầu vào đến một số đầu ra. Loại này tạo điều kiện cho các ứng dụng ngôn ngữ như chú thích hình ảnh bằng cách tạo một câu từ một từ khóa duy nhất.
- Nhiều-nhiều**:** Mô hình sử dụng nhiều đầu vào để dự đoán nhiều đầu ra. Ví dụ: có thể tạo một công cụ dịch ngôn ngữ bằng RNN, với khả năng phân tích câu và cấu trúc chính xác các từ trong một ngôn ngữ khác.
- Nhiều-một**:** Một số đầu vào được ánh xạ đến một đầu ra. Loại này rất hữu ích trong các ứng dụng như phân tích cảm xúc, trong đó mô hình dự đoán cảm xúc của người dùng như *tích cực, tiêu cực* và *trung lập* từ lời chứng thực đầu vào.

Kể từ khi RNN được đưa vào sử dụng, nó đã đạt được tiến bộ đáng kể trong các ứng dụng xử lý ngôn ngữ tự nhiên (NLP) bằng RNN cùng các biến thể. Tuy nhiên, bộ mô hình RNN có một số hạn chế:

Độ dốc cực lớn

RNN có thể dự đoán sai đầu ra trong khóa đào tạo ban đầu. Bạn cần lặp lại nhiều lần để điều chỉnh các thông số của mô hình nhằm giảm tỷ lệ lỗi. Bạn có thể mô tả độ nhạy của tỷ lệ lỗi tương ứng với thông số của mô hình dưới dạng độ dốc. Bạn có thể hình dung độ dốc như một đường dốc mà bạn đi xuống từ một ngọn đồi. Độ dốc lớn hơn cho phép mô hình học nhanh hơn và độ dốc nhỏ làm giảm tốc độ học tập.

Độ dốc cực lớn xuất hiện khi độ dốc tăng theo cấp số nhân cho đến khi RNN trở nên không ổn định. Khi độ dốc trở nên lớn vô hạn, RNN hoạt động thất thường, dẫn đến các vấn đề về hiệu năng như quá khớp. Quá khớp là hiện tượng mô hình có thể dự đoán chính xác với dữ liệu đào tạo nhưng không thể dự đoán chính xác với dữ liệu thực tế.

Độ dốc biến mất

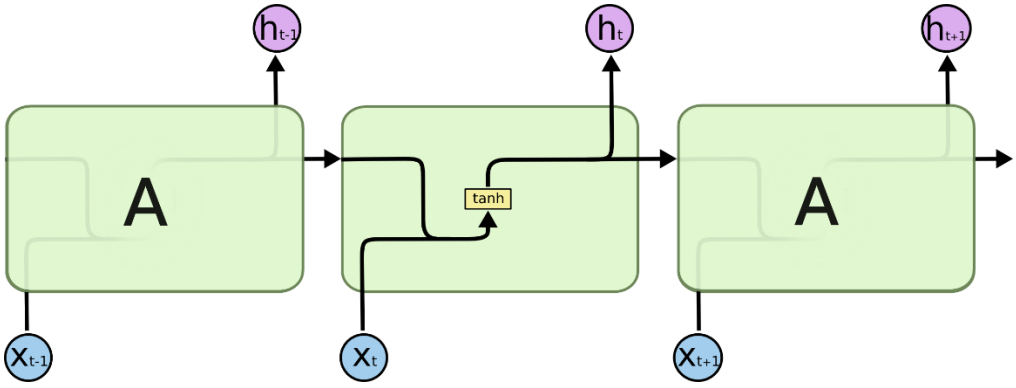
Bài toán độ dốc biến mất là một điều kiện, trong đó độ dốc của mô hình đạt đến 0 trong quá trình đào tạo. Khi độ dốc biến mất, RNN không học hiệu quả từ dữ liệu đào tạo, dẫn đến chưa khớp. Mô hình chưa khớp không thể hoạt động tốt trong các ứng dụng thực tế vì các trọng số chưa được điều chỉnh thích hợp. RNN sẽ có rủi ro gặp phải vấn đề độ dốc cực lớn và biến mất khi xử lý các chuỗi dữ liệu dài.

Bộ chuyển đổi (Tranformers) là các mô hình học sâu sử dụng các cơ chế tự chú ý trong mạng neuron truyền thẳng bộ mã hóa-giải mã. Bộ chuyển đổi có thể xử lý dữ liệu tuần tự theo cách giống như RNN, đồng thời giải quyết các hạn chế của mạng neuron này.

1.4.2 Mạng Neuron hồi quy hiện đại (LSTM và GRU)

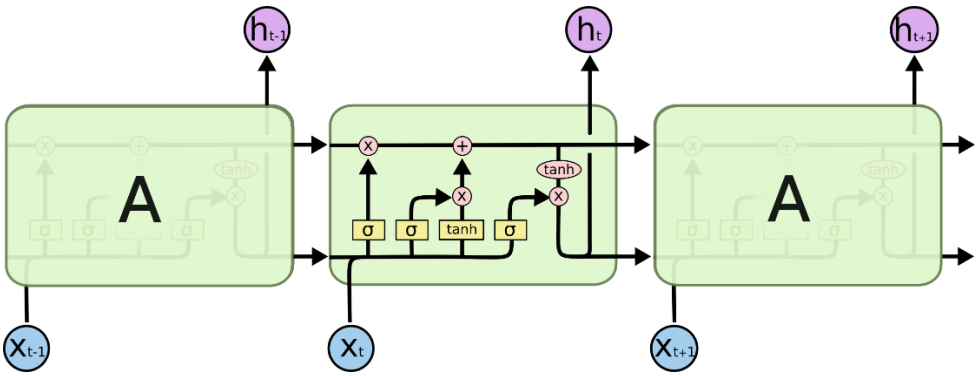
Mạng trí nhớ ngắn hạn định hướng dài hạn còn được viết tắt là LSTM làm một kiến trúc đặc biệt của RNN có khả năng học được sự phục thuộc trong dài hạn (long-term dependencies) được giới thiệu vào năm 1997. Kiến trúc này đã được phổ biến và sử dụng rộng rãi cho tới ngày nay. [37] LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn dữ được tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi.

Một mạng RNN tiêu chuẩn sẽ có kiến trúc rất đơn giản chẳng hạn như đối với kiến trúc gồm một tầng ẩn như hàm tanh bên dưới:



Hình 1.8 Sự lặp lại kiến trúc module trong mạng RNN chứa một tầng ẩn

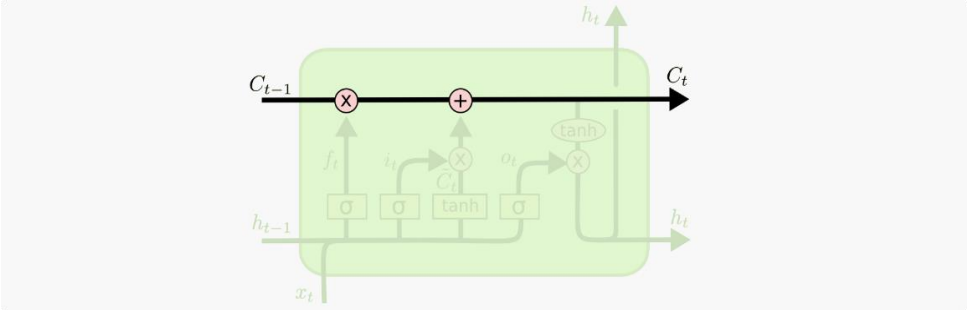
LSTM cũng có một chuỗi dạng như thế nhưng phần kiến trúc lặp lại có cấu trúc khác biệt hơn. Thay vì chỉ có một tầng đơn, chúng có tới 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau theo một cấu trúc đặc biệt:



Hình 1.9 Sự lặp lại kiến trúc module trong mạng LSTM chứa 4 tầng ẩn

Trong 2 sơ đồ tính toán trên, mỗi một phép tính sẽ triển khai trên một vector. Trong đó hình tròn màu hồng biểu diễn một toán tử đối với vector như phép cộng vector, phép nhân vô hướng các vector. Màu vàng thể hiện hàm activation mà mạng neuron sử dụng để học trong tầng ẩn, thông thường là các hàm phi tuyến sigmoid và tanh. Kí hiệu 2 đường thẳng nhập vào thể hiện phép chập kết quả trong khi kí hiệu 2 đường thẳng rẽ nhánh thể hiện cho nội dung vector trước đó được sao chép để đi tới một phần khác của mạng neuron. [38]

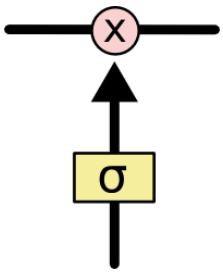
Ý tưởng chính của LSTM là thành phần ô trạng thái (cell state) được thể hiện qua đường chạy ngang qua đỉnh đồ thị như hình vẽ bên dưới:



Hình 1.10 Đường đi của ô trạng thái trong mạng LSTM

Ô trạng thái là một dạng băng chuyền chạy thẳng xuyên suốt toàn bộ chuỗi với chỉ một vài tương tác tuyến tính nhỏ giúp cho thông tin có thể truyền dọc theo đồ thị mạng nơ ron ổn định.

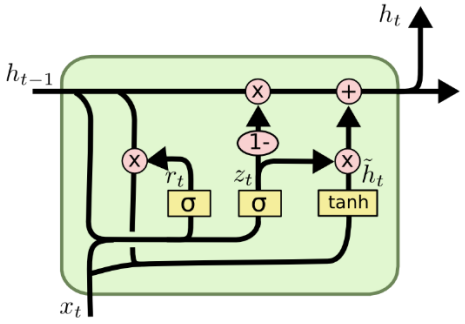
LSTM có khả năng xóa và thêm thông tin vào ô trạng thái và điều chỉnh các luồng thông tin này thông qua các cấu trúc gọi là cổng. Cổng là cơ chế đặc biệt để điều chỉnh luồng thông tin đi qua. Chúng được tổng hợp bởi một tầng ẩn của hàm activation sigmoid và với một toán tử nhân như đồ thị.



Hình 1.11 Một cổng của hàm sigmoid trong LSTM

Hàm sigmoid sẽ cho đầu ra là một giá trị xác xuất nằm trong khoảng từ 0 đến 1, thể hiện rằng có bao nhiêu phần thông tin sẽ đi qua cổng. Giá trị bằng 0 ngụ ý rằng không cho phép thông tin nào đi qua, giá trị bằng 1 sẽ cho toàn bộ thông tin đi qua. Một mạng LSTM sẽ có 3 cổng có kiến trúc dạng này để bảo vệ và kiểm soát các ô trạng thái.

Một dạng biến thể khá mạnh khác của LSTM là cổng truy hồi đơn vị (Gated Recurrent Unit - GRU) được giới thiệu vào năm 2014. Nó kết hợp cổng quên và cổng vào thành một cổng đơn gọi là cập nhật (update gate) [39]. Nó cũng nhập các ô trạng thái và trạng thái ẩn và thực hiện một số thay đổi khác. Kết quả của mô hình đơn giản hơn nhiều so với mô hình LSTM chuẩn, và đã trở nên khá phổ biến.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

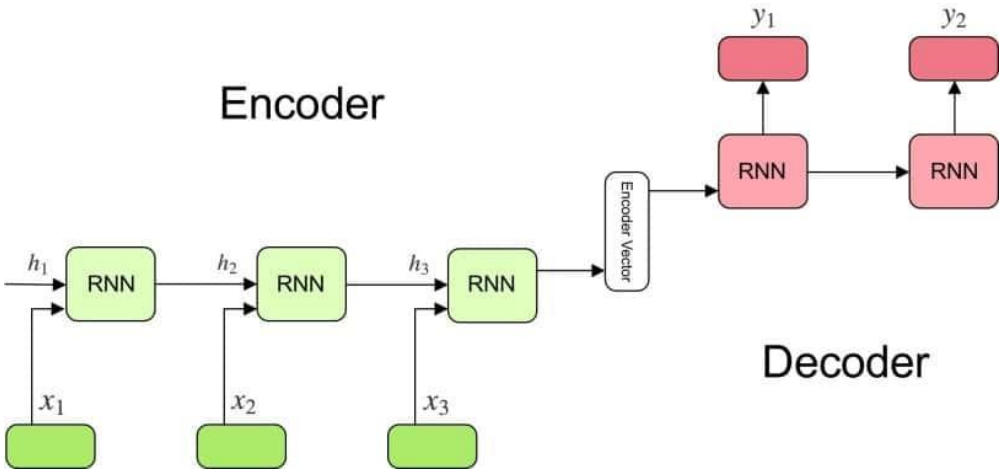
Hình 1.12 Cấu trúc cổng truy hồi đơn vị GRU

Được viết dưới dạng một hệ phương trình, LSTM trông khá là phức tạp và có phần hàn lâm. Mạng neuron này là một bước đột phá lớn mà đã khắc phục được những hạn chế ở RNN đó là khả năng phụ thuộc dài hạn. Một số kĩ thuật học Attention đã được phát triển để được kết hợp với LSTM và đã tạo ra những kết quả khá bất ngờ trong các tác vụ dịch máy cũng như phân loại nội dung, trích lọc thông tin,…. Các mô hình dịch máy hay nhận diện ngữ cảnh của Google đã ứng dụng kiểu kết hợp này trong các bài toán ngôn ngữ tự nhiên của mình và đã cải thiện được đầu ra một cách đáng kể.

1.4.3 Kiến trúc mã hóa – giải mã

Mô hình mã hóa-giải mã (Encoder-Decoder) là một kiến trúc phổ biến trong học sâu, được thiết kế để ánh xạ từ một chuỗi đầu vào sang một chuỗi đầu ra. Kiến trúc này

thường được sử dụng trong các bài toán xử lý ngôn ngữ tự nhiên (NLP) và lớp các mô hình này đã mang đến những kết quả rõ nét trong các nhiệm vụ phức tạp như dịch máy, chú thích video, hỏi đáp.



Hình 1.13 Kiến trúc của một mô hình mã hóa-giải mã

Mô hình điển hình thuộc lớp này bao gồm 3 phần: bộ mã hóa (Encoder), vector mã hóa trung gian (Encoder vector) và bộ giải mã (Decoder).

Bộ mã hóa – Encoder:

- Một ngăn xếp chứa các mạng con là phần tử của RNN (hoặc các ô nhớ của LTSM hay GRU) nơi nhận vào tín hiệu của một phần tử của chuỗi đầu vào và truyền tiếp về phía cuối mạng.
- Trong bài toán hỏi đáp, chuỗi đầu vào là tập hợp tất cả các từ của câu hỏi. Mỗi từ được thể hiện bởi x_i với i là thứ tự của từ đó.
- Các trạng thái ẩn h_i được tính với công thức:

$$h_t = f\left(W^{(hh)}h_{t-1} + W^{(hx)}x_t\right)$$

Công thức trên mô tả kết quả của một mạng neuron hồi quy thông thường [40]. Như chúng ta có thể thấy, các trạng thái ẩn được tính bởi đầu vào tương ứng x_t và trạng thái ẩn trước đó h_{t-1}.

Vector mã hóa trung gian – Encoder vector:

- Đây là trạng thái ẩn nằm ở cuối chuỗi, được tính bởi bộ mã hóa, nó cũng được tính bởi công thức phía trên.
- Véc tơ này có chức năng gói gọn thông tin của toàn bộ các phần tử đầu vào để giúp cho bộ mã hóa dự đoán thông tin chính xác hơn.
- Véc tơ này sau đó hoạt động như trạng thái ẩn đầu tiên của bộ giải mã.

Bộ giải mã – Decoder:

- Một ngăn xếp các mạng con là phần tử của RNN có nhiệm vụ dự đoán đầu ra y_t tại thời điểm *t*.
- Mỗi phần từ này nhận đầu vào là trạng thái ẩn trước đó và tạo kết quả đầu ra cũng như trạng thái ẩn của chính nó.
- Trong bài toán hỏi đáp, chuỗi đầu ra là tập hợp các từ của câu trả lời. Mỗi từ được biểu diễn bởi y_ivới *i* là thứ tự của từ.
- Các trạng thái ẩn h_t được tính bởi công thức:

$$h_t = f\left(W^{(hh)} h_{t-1}\right)$$

  Như đã thấy, các trạng thái ẩn được tính bởi trạng thái ngay trước đó.

- Đầu ra y_t tại thời điểm t được tính bởi công thức:

$$y_t = \text{softmax}(W^S h_t)$$

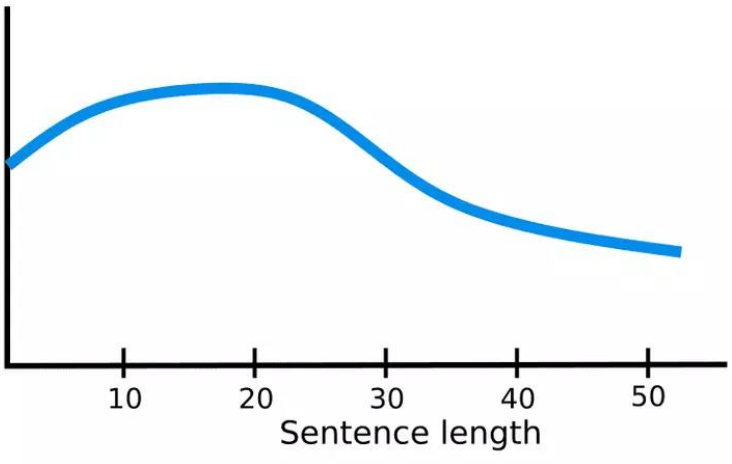
Chúng ta tính đầu ra sử dụng trạng thái ẩn tương ứng tại thời điểm hiện tại và nhân với trọng số tương ứng W^S. Softmax, [41] hay còn gọi là hàm mũ mềm, được sử dụng để tạo ra vector xác suất giúp ta xác định đầu ra cuối cùng (VD: các từ trong bài toán hỏi đáp).

Năng lực đặc biệt của mô hình này là nó có thể ánh xạ chuỗi đầu vào và chuỗi đầu ra có độ dài khác nhau. Vì thế, nó mở ra giải pháp cho một loạt các bài toán trong lĩnh vực hỏi đáp với AI.

1.4.4 Cơ chế chú ý

Sự ra đời của cơ chế chú ý (Attention Mechanism) trong học sâu đã mang lại hiệu quả đáng kể cho nhiều mô hình, nó đã và đang tiếp tục là một thành phần không thể thiếu trong các mô hình tiến tiến nhất.

Mô hình seq2seq (sequence to sequence – từ chuỗi sang chuỗi) thường được sủ dụng với hai thành phần là khối encoder và decoder. Hai khối này đều được tạo thành từ các lớp RNN. Khối Encoder sẽ xử lí thông tin đầu vào và đầu ra là một vector biểu diễn duy nhất, hay còn gọi quá trình này là nén thông tin. [42] Vector biểu diễn này sẽ mang toàn bộ thông tin để khối Decoder có thể tạo ra câu đích. Thực tế, mô hình seq2seq bởi kiến trúc từ RNN hoạt động rất tốt đối với các chuỗi có đồ dài ngắn, tuy nhiên độ dài chuỗi tăng thì chất lượng của mô hình sẽ giảm đáng kể.



Hình 1.14 Minh họa chất lượng mô hình tỉ lệ với độ dài câu

Mô hình seq2seq với RNN thì như vậy, encoder sẽ phải "nén" (compress) toàn bộ chuỗi đầu vào thành một vector duy nhất. [43] Khi mà chuỗi đủ dài và encoder buộc phải đưa toàn bộ thông tin vào 1 vector biểu diễn duy nhất như vậy thì chắc chắn nó có khả năng cao bỏ quên thông tin (bottleneck). Ngoài ra, decoder chỉ nhìn thấy một vector biểu diễn đầu vào duy nhất, mặc dù tại mỗi time-step thì các phần khác nhau của chuỗi vào có thể có ích hơn các phần khác. Nhưng đối với mô hình hiện tại thì decoder sẽ phải trích các thông tin liên quan này từ một vector biểu diễn duy nhất - việc này cũng vô cùng khó. Attention đã được ra đời vào năm 2015 với mục đích giải quyết vấn đề kể trên. Với cơ chế này, tại mỗi time-step khác nhau, mô hình sẽ tập trung vào các phần khác nhau của đầu vào.

Như vậy cơ chế chú ý được ra đời để giải quyết các vấn đề của mô hình mã hóa-giả mã (mà không cần đến các mạng neuron hồi quy), với ý tưởng sử dụng một vector bối cảnh có thể tương tác với toàn bộ vector trạng thái ẩn của encoder thay vì chỉ sử dụng vector trạng thái ẩn cuối cùng để tạo ra vector biểu diễn cho decoder. [44] Ý tưởng chính của cơ chế nhấn mạnh tại mỗi time-step thì phần vào của đầu vào quan trọng. Mô hình với cơ chế attention có thể được huấn luyện để tối ưu hóa cùng lúc (end-to-end) mà không cần phụ thuộc vào mô hình nào khác. Chính cơ chế sẽ giúp sẽ tự học làm điều đó.

Cơ chế attention đã loại bỏ sự phục thuộc về khoảng cách của chuỗi đầu vào và đầu ra. Với attention thì lĩnh vực học máy đã có nhiều cải tiến đáng kể, ngày càng có nhiều các dạng khác nhau của cơ chế attention. Chúng được chia thành 3 loại sau:

- Self-attention
- Global/Soft attention

- Local/Hard attention

Trong số đó, Self-attention hay còn biết đến là intra-attention, là một cơ chế attention chỉ dùng cho một câu. Có thể hiểu rằng ta sẽ tự tạo một ma trận với hàng và cột đều là cùng một câu để hiểu được những phần nào của câu sẽ liên quan đến nhau. [45] Cơ chế này đã chứng minh được sự hiệu quả trong các ứng dụng tóm tắt văn bản, đọc máy, hỏi đáp… Cùng với self-attention là sự ra đời của kiến trúc tranformer, cho phép thay thế hoàn toàn kiến trúc mạng nơ-ron hồi tiếp RNN bằng các mô hình (fully conneted) và vẫn mang lại kết quả rất tốt. Self-attention, mặc dù không quá phức tạp, là cột mốc quan trọng cho việc áp dụng cơ chế attention cho các bài toán về NLP.

1.4.5 Kiến trúc Tranformer

Với sự ra đời của cơ chế attention thì vào năm 2017 paper Attention is all you need đã giới thiệu một kiến trúc mới dành cho các bài toán NLP mà không có sự xuất hiện của các mạng neuron hồi tiếp (RNN, LSTM…) hay là mạng nơ-rơn tích chập (CNN) và đó là kiến trúc Transformer. [46]

Bảng dưới đây sẽ thể hiện sự khác biệt của transformer với các mô hình đi trước, khi mà transformer hoàn toàn hoạt động dựa trên cơ chế attention.

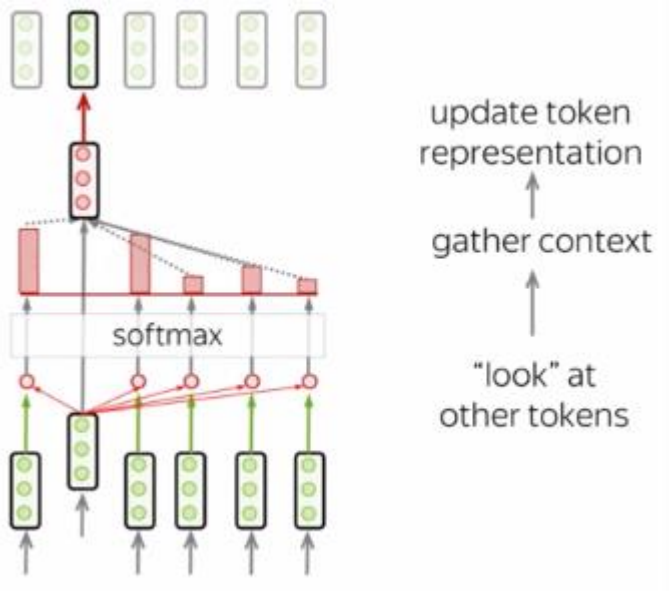
Bảng 1.2 So sánh cơ chế hoạt động của một số mô hình ngôn ngữ

	Seq2seq	**Seq2seq + attention**	**tranformer**
Encoder	RNN	RNN	attention
Decoder	RNN	RNN	attention
Tương tác mã hóa-giải mã	vector	attention	attention

Làm một phép so sánh giữa encoder của RNN và transformer: Đối với quá trính encode một câu thì RNNs sẽ cần một khoảng thời gian để đọc lần lượt từng từ trong câu, đối với 1 câu dài quá trình này có thể diễn ra khá lâu, như vậy độ phức tạp của encoder để xử lí một câu có độ dài N là O(N). [47] Ngược lại, trong quá trình encode của transformer, các source tokens sẽ "quan sát" nhau bằng cơ chế tự chú ý và cố gắng hiểu nhau trong ngữ cảnh của câu. Và độ phức tạp của quá trình này chỉ là O(1).

Có thể nói cơ chế tự chú ý chính là thành phần nổi bật và quan trọng nhất của transformer. Ở một bên, cơ chế attention sẽ tính toán dựa trên trạng thái của decoder ở time-step hiện tại và tất cả các trạng thái ẩn của encoder. Một bên khác, self-attention có

thể hiểu là chú ý trong cùng một câu, từng thành phần trong câu sẽ tương tác với nhau. Từng token sẽ "quan sát" các tokens còn lại trong, thu thập ngữ cảnh của câu và cập nhập vector biểu diễn. [48]



Hình 1.15 Minh họa khả năng tự tương tác của cơ chế tự chú ý

Để xây dựng cơ chế self attention cần chú ý đến hoạt động của 3 vector biểu diễn cho mỗi từ lần lượt là:

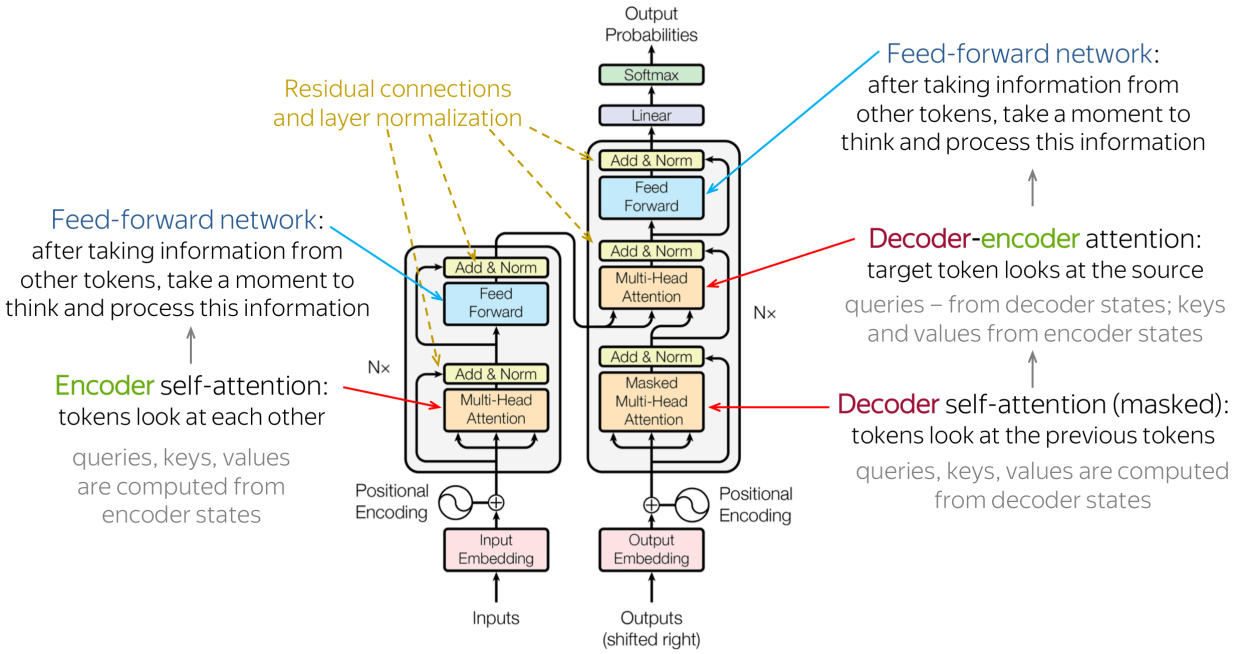
- **Query**: hỏi thông tin
- **Key**: trả lời rằng nó có một số thông tin
- **Value**: trả về thông tin đó

Query được sử dụng khi một token "quan sát" những tokens còn lại, nó sẽ tìm kiếm thông tin xung quanh để hiểu được ngữ cảnh và mối quan hệ của nó với các tokens còn lại. Key sẽ phản hồi yêu cầu của Query và được sử dụng để tính trọng số attention. Cuối cùng, Value được sử dụng trọng số attention vừa rồi để tính ra vector đại diện (attention vector).

Kiến trúc của Transformer

Sau đây ta đi qua các thành phần chính cấu tạo nên transformer. Đây là cấu trúc của transformer được giới thiệu trong bài báo "Attention is all you need" năm 2015. Ở bên trái là encoder, thông thường có Nx = 6 layers chồng lên nhau. Mỗi layer sẽ có multi-head attention như đã tìm hiểu và khối feed-forward. [49] Ngoài ra còn các các kết nối residual giống như trong mạng Resnet. Ở bên phải là decoder, tương tự cũng có Nx = 6 layers chồng lên nhau. Kiến trúc thì khá giống encoder những chỉ có thên khối masked

multi-head attention ở vị trí đầu tiên. Các thành phần trong kiến trúc transformer:



Hình 1.16 Kiến trúc mô hình tranformer

- Positional encoding: Bởi vì transformer không có các mạng hồi tiếp hay mạng tích chập nên nó sẽ không biết được thứ tự của các token đầu vào. Vì vậy, cần phải có cách nào đó để cho mô hình biết được thông tin này. Đó chính là nhiệm vụ của positional encoding. Như vậy, sau bước nhúng từ (embedding layers) để thu được các tokens thì sẽ cộng nó với các vector thể hiện vị trí của từ trong câu.
- Lớp Normalization: Trong hình ảnh cấu trúc, có lớp "Add & Norm" thì từ Norm thể hiện cho lớp Normalization. Lớp này đơn giản là sẽ chuẩn hóa lại đầu ra của multi-head attention, mang lại hiệu quả cho việc nâng cao khả năng hội tụ.
- Kết nối Residual: Kết nối residual bản chất rất đơn giản: thêm đầu vào của một khối vào đầu ra của nó. Với kết nối này giúp mạng có thể chồng được nhiều layers. Như trên hình, kết nối residual sẽ được sử dụng sau các khối FFN và khối attention. Như trên hình từ "Add" trong "Add & Norm" sẽ thể hiện cho kết nối residual.
- Khối Feed-Forward: Đây là khối cơ bản, sau khi thực hiện tính toán ở khối attention ở mỗi lớp thì khối tiếp theo là FFN. Có thể hiểu là cơ chế attention giúp thu thập thông tin từ những tokens đầu vào thì FFN là khối xử lí những thông tin đó.

Mô hình Transformer đã mang lại những cải tiến vượt bậc trong lĩnh vực NLP nhờ

cơ chế self-attention và khả năng xử lý song song dữ liệu. Một số ứng dụng quan trọng của Transformer trong NLP phải kể đến như:

- Dịch máy
- Trả lời câu hỏi (QA)
- Phân tích cảm xúc
- Sinh văn bản

Một trong những ứng dụng lớn nhất của Transformer là xây dựng các mô hình ngôn ngữ lớn, Transformer đã trở thành nền tảng của các mô hình LLM này, với sự thành công vượt trội nhờ khả năng xử lý ngữ cảnh và dữ liệu lớn một cách hiệu quả. Các mô hình LLM như BERT, GPT, T5, JinaAI, Alibaba, Nomic và các phiên bản tiên tiến hơn đều dựa trên kiến trúc Transformer.

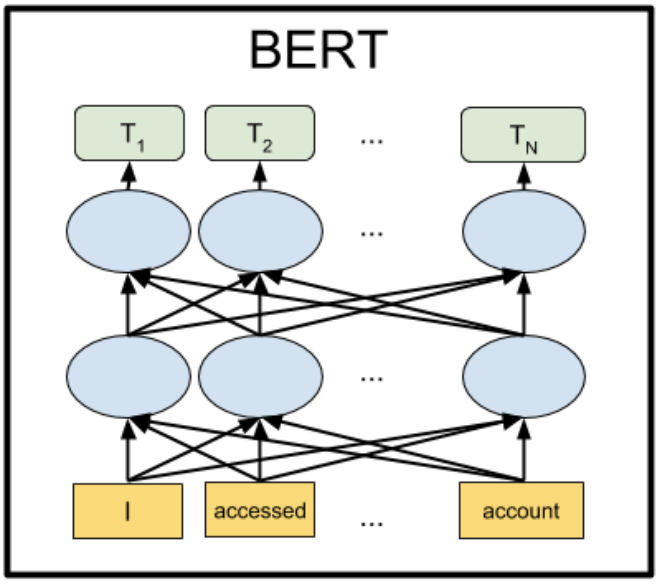
1.4.6 Mô hình BERT

BERT là viết tắt của Bidirectional Encoder Representations from Transformers được hiểu là một mô hình học sẵn hay còn gọi là pre-train model, học ra các vector đại diện theo ngữ cảnh 2 chiều của từ, được sử dụng để transfer sang các bài toán khác trong lĩnh vực xử lý ngôn ngữ tự nhiên. BERT được đánh giá cao bởi việc cải thiện những công việc tìm ra đại diện của từ trong không gian số (không gian mà máy tính có thể hiểu được) thông qua ngữ cảnh của nó. [50]

Trước khi BERT ra đời thì các tác vụ như: phân loại cảm xúc văn bản (tốt hay xấu, tích cực hay tiêu cực), sinh văn bản, dịch máy,…. đều sử dụng kiến trúc RNN. Kiến trúc này có nhiều nhược điểm cố hữu như train chậm, mất quan hệ giữa các từ xa nhau… Không giống như các mô hình trước đó, BERT được thiết kế để đào tạo ra các vector đại diện cho ngôn ngữ văn bản thông qua ngữ cảnh 2 chiều(trái và phải) của chúng. Kết quả là, vector đại diện được sinh ra từ mô hình BERT được tính chỉnh với các lớp đầu ra bổ sung đã tạo ra nhiều kiến trúc cải tiến đáng kể cho các nhiệm vụ xử lý ngôn ngữ tự nhiên như Question Answering, Language Inference,...mà không cần thay đổi quá nhiều từ các kiến trúc cũ.

Khả năng vượt trội của mô hình BERT đến từ việc nó có nhúng thêm ngữ cảnh vào trong các vector embedding. Ngữ cảnh là một thứ vô cùng quan trọng trong ngôn ngữ. Các Language model càng bao gồm được ngữ cảnh thì càng có thể đạt được chất lượng tốt.

Sử dụng kiến trúc transformer thế nhưng mô hình BERT lại bỏ đi phần Decoder mà chỉ tập trung vào Encoder. Mô hình thêm văn bản vào và có đầu ra là các encoder output :



Hình 1.17 Mô hình BERT chỉ bao gồm Encoder

Do sự đặc biệt trên nên BERT được train đồng thời 2 task gọi là Masked LM (để dự đoán từ thiếu trong câu) và Next Sentence Prediction (NSP – dự đoán câu tiếp theo câu hiện tại). [51] Hai task này được train đồng thời và tổng loss sẽ là kết hợp loss của 2 task. Model sẽ cố gắng giảm tối thiểu tổng loss này. Cụ thể 2 task này như sau:

- Masked LM: Với task này, việc training sẽ thực hiện che đi tầm 15% số từ trong câu và đưa vào model. Model sẽ được train để dự đoán ra phần câu bị che đó dựa vào các từ còn lại. Loss sẽ được tính tại vị trí masked và bỏ qua các vị trí khác để đánh giá xem model dự đoán phần bị mask đúng hay sai.
- Next Sentence Prediction (NSP): Trong task này, model sẽ được feed cho từng cặp câu một. Nhiệm vụ của model là xác định câu thứ 2 trong cặp câu có phải đúng là đứng sau câu đầu tiên hay không. Trong quá trình train, chọn 50% mẫu là Positive (output là 1) và 50% còn lại là Negative được ghép linh tinh (output là 0).

Mô hình BERT có thể được coi là 1 bước nhảy vượt bậc của Google trong lĩnh vực xử lý ngôn ngữ tự nhiên và cũng đã được tùy chỉnh cải tiến và rất phổ biến trong các bài toán trong xử lí ngôn ngữ tự nhiên cho tiếng Việt.

1.4.7 Mô hình GPT

Generative Pre-trained Transformer, thường được gọi là GPT, là một dòng mô hình mạng neuron sử dụng kiến trúc của bộ chuyển đổi và là một tiến bộ quan trọng trong lĩnh

vực trí tuệ nhân tạo AI, hỗ trợ cho các ứng dụng AI tạo sinh như ChatGPT. Các mô hình GPT rất phù hợp để tạo các văn bản giống như của con người khi đưa ra một lời nhắc như một câu hỏi. Nó cũng có thể được sử dụng để trả lời các câu hỏi được nhắc, tóm tắt văn bản, dịch thuật,…

Khi được tương tác, các mô hình GPT sử dụng các pattern và mối quan hệ đã học được từ dữ liệu văn bản để dự đoán những từ nào sẽ xuất hiện tiếp theo trong câu, dựa trên ngữ cảnh được cung cấp. Nó tạo ra văn bản theo từng từ, điều chỉnh xác suất của những gì sẽ xuất ra với mỗi từ mới, để tạo ra một câu hoàn chỉnh – tuân theo các quy tắc ngữ pháp và có ý nghĩa. Các mô hình GPT dự đoán ngôn ngữ dựa trên mạng neuron được xây dựng trên kiến trúc Transformer [52]. Các mô hình phân tích truy vấn ngôn ngữ tự nhiên, được gọi là lời nhắc và dự đoán phản ứng tốt nhất có thể dựa trên hiểu biết của họ về ngôn ngữ. Để làm được điều đó, các mô hình GPT dựa vào kiến thức mà họ có được sau khi được đào tạo với hàng trăm tỷ tham số trên các tập dữ liệu ngôn ngữ khổng lồ. Các mô hình có thể tính đến ngữ cảnh nhập liệu và tự động tham gia vào các phần khác nhau của nhập liệu, khiến cho chúng có khả năng tạo ra các phản hồi dài, không chỉ là từ tiếp theo trong một trình tự.

Được xây dung trên kiến trúc transformer, các mô hình GPT là các mạng neuron biến đổi, sử dụng các cơ chế tự tập trung để tập trung vào các phần khác nhau của văn bản nhập liệu trong mỗi bước xử lý. Mô hình biến đổi nắm bắt nhiều bối cảnh hơn và cải thiện hiệu suất trên các tác vụ xử lý ngôn ngữ tự nhiên. Mô hình này có hai mô-đun chính giống với hều hết transformer đó là Encoder và Decoder.

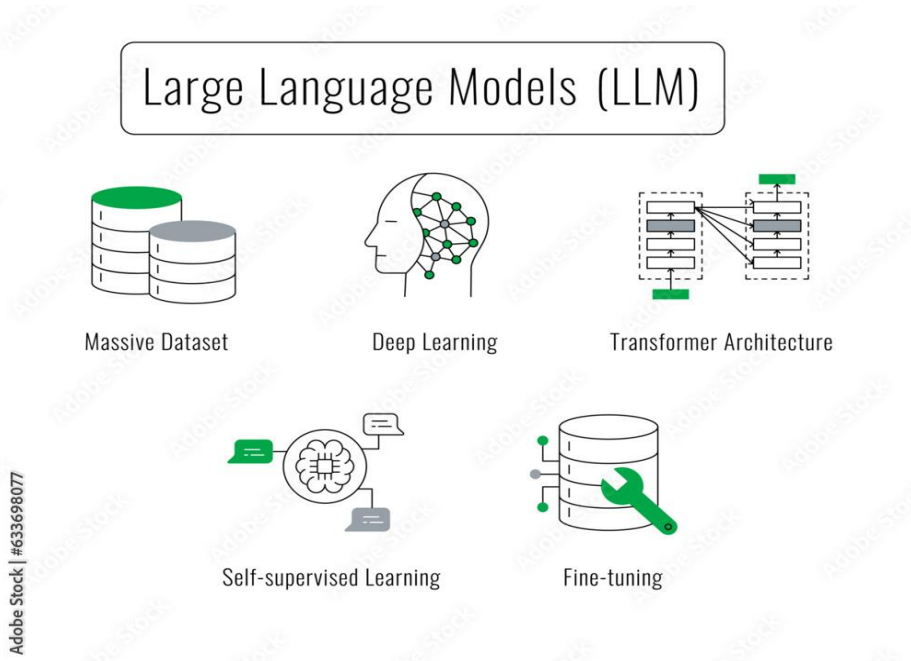
1.5 Mô hình ngôn ngữ lớn

1.5.1 Giới thiệu về ngôn ngữ lớn

Mô hình ngôn ngữ lớn (LLM) là một loại chương trình trí tuệ nhân tạo (AI) có thể nhận dạng và tạo văn bản, trong số các tác vụ khác. LLM được đào tạo trên các tập dữ liệu khổng lồ do đó có tên là "lớn". LLM được xây dựng trên máy học: cụ thể là một loại mạng nơ-ron được gọi là mô hình Transformer [53].

Nói một cách đơn giản hơn, LLM là một chương trình máy tính đã được cung cấp đủ ví dụ để có thể nhận dạng và diễn giải ngôn ngữ con người hoặc các loại dữ liệu phức tạp khác. Nhiều LLM được đào tạo trên dữ liệu đã được thu thập từ Internet hàng nghìn hoặc hàng triệu gigabyte văn bản. LLM sử dụng một loại máy học gọi là học sâu để hiểu cách

các ký tự, từ và câu hoạt động cùng nhau. Học sâu liên quan đến phân tích xác suất dữ liệu phi cấu trúc, cuối cùng cho phép mô hình học sâu nhận ra sự khác biệt giữa các phần nội dung mà không cần sự can thiệp của con người [54], [55].



Hình 1.18 Minh họa xử lý đa tác vụ của một LLM.

Sau đó, các LLM được đào tạo thêm thông qua quá trình điều chỉnh, mô hình được điều chỉnh chính xác hoặc điều chỉnh theo yêu cầu cho nhiệm vụ cụ thể mà lập trình viên muốn nó thực hiện (được gọi là là fine-tuning model), chẳng hạn như diễn giải câu hỏi và tạo phản hồi hoặc dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác [54].

1.5.2 Một số mô hình ngôn ngữ lớn nổi bật

Sự phát triển mạnh mẽ của các mô hình LLM trong thời gian gần đây trong cuộc đua AI đã tạo nên sự đa dạng mô hình LLM, khi ngày càng có nhiều mô hình được ra mắt với khả năng không ngừng cải tiến về độ chính xác, tốc độ và khả năng xử lý ngữ cảnh. Các LLM hiện đại không chỉ tập trung vào việc sinh văn bản mà còn mở rộng ứng dụng vào các lĩnh vực như tìm kiếm ngữ nghĩa (semantic search), lập luận logic, và hỗ trợ đa ngôn ngữ, đáp ứng nhu cầu ngày càng tăng trong nghiên cứu và công nghiệp. Các mô hình LLM mạnh mẽ hầu hết được phát triển cho mục đích công nghiệp, tuy nhiên vẫn có những mô hình LLM được phát hành miễn phí dành cho các nhà phát triển thử nghiệm và nghiên cứu. Dưới đây là một vài mô hình LLM mở được cân nhắc sử dụng trong báo cáo này:

Llama3

Mô hình ngôn ngữ lớn *đa ngôn ngữ* Llama 3.3 của Meta là một mô hình tạo văn bản được tinh chỉnh theo hướng dẫn, với 70 tỷ tham số (đầu vào văn bản/đầu ra văn bản). Mô hình Llama 3.3 được tinh chỉnh cho các tác vụ văn bản [56], được tối ưu hóa cho các trường hợp sử dụng đối thoại đa ngôn ngữ và vượt trội hơn nhiều mô hình chat mã nguồn mở và mô hình đóng hiện có và đang phổ biến.

Mistral7B

Được Mistral AI phát triển, mô hình này đặt trọng tâm vào khả năng xử lý dữ liệu nhanh. Dựa vào dữ liệu benchmark trên internet, có thể thấy được đây là một mô hình cực kỳ vượt trội. Mistral 7B tuy có hiệu suất kém hơn Llama3 trong các tác vụ đa ngôn ngữ, nhưng vẫn duy trì mức độ hiệu quả tiêu chuẩn trong các tác vụ ngôn ngữ tiếng Anh. Hiệu suất cân bằng của Mistral 7B được cho là nhờ vào hai yếu tố khác nhau: Mistral 7B tận dụng cơ chế chú ý theo nhóm truy vấn (grouped-query attention - GQA) và chú ý cửa sổ trượt (sliding window attention - SWA) [57] trong kiến trúc của mình để mang lại những lợi ích độc đáo. Các phản hồi đối với truy vấn về cách sử dụng Mistral 7B sẽ thể hiện giá trị của GQA và SWA. GQA mang lại thời gian suy luận nhanh hơn so với cơ chế chú ý toàn phần tiêu chuẩn. Trong khi đó, SWA giúp Mistral 7B xử lý các chuỗi văn bản dài hơn với chi phí thấp hơn.

Gemma2

Một mô hình mã nguồn mở còn khá mới của Google, được phát triển dựa trên cùng nền tảng nghiên cứu và công nghệ đã tạo ra các mô hình Gemini. Đây là các mô hình ngôn ngữ lớn chỉ sử dụng bộ giải mã (decoder-only) với khả năng xử lý văn bản-đến-văn bản (text-to-text), hiện chỉ hỗ trợ tiếng Anh và được cung cấp trọng số mở cho cả các biến thể đã được tiền huấn luyện và tinh chỉnh theo hướng dẫn. [58]

Điểm chung của cả 3 mô hình ngôn ngữ lớn trên là chúng đều là mô hình ngôn ngữ lớn mã nguồn mở, phù hợp với nhiều nhiệm vụ tạo văn bản, bao gồm trả lời câu hỏi, tóm tắt và lập luận. Với kích thước khá nhỏ, chúng có thể triển khai trong các môi trường có tài nguyên hạn chế như laptop, máy tính để bàn, hoặc hạ tầng đám mây riêng, dễ dàng tang tính tiếp cận cho người dung thông thường.

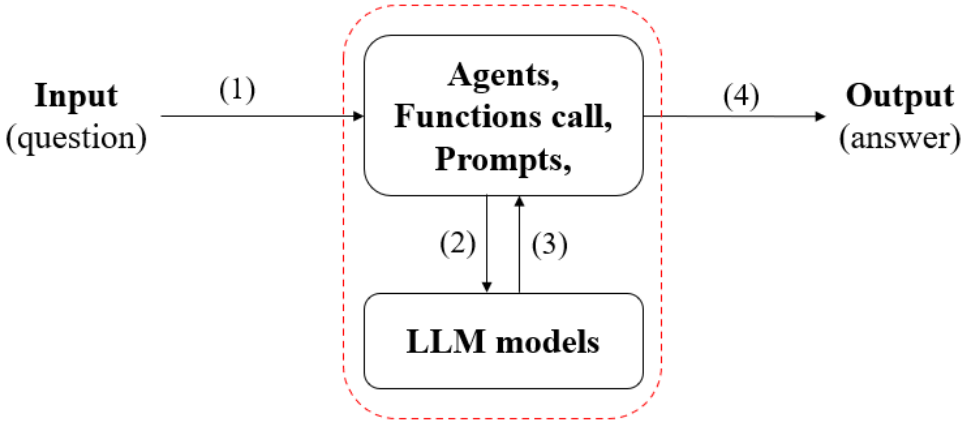
Bảng 1.3 Bảng so sánh các mô hình ngôn ngữ lớn mã nguồn mở nổi bật.

Mô hình	Ưu điểm	Hạn chế	Kết luận về ứng dụng
Llama3	Đa ngôn ngữ, hiệu năng cao, được nhiều framework hỗ trợ	Khá đòi hỏi về cấu hình	Chatbot giáo dục, dịch vụ khách hàng
Mistral7B	Tốc độ xử lý nhanh, tối ưu tài nguyên	Hiệu năng tốt nhất ở ngôn ngữ tiếng Anh, có temperature cao	Hỗ trợ lập luận, trợ lý ảo
Gemma2	Phân tích logic tốt, tìm kiếm ngữ nghĩa chính xác	Có vẻ như chưa tối ưu cho hội thoại thông thường	Tìm kiếm thông tin, phân tích dữ liệu

Một mô hình với hiệu năng tốt khi xử lý ngữ cảnh có độ dài dưới 5.000 token, tương tự như làm việc với một vài trang sách sẽ là một lợi thế. Mô hình này sẽ đặc biệt phù hợp với các trường hợp yêu cầu câu trả lời nhanh chóng và chính xác từ một tập dữ liệu giới hạn. Với ít token như thế, model có thể được sử dụng để nhanh chóng truy xuất thông tin liên quan từ cơ sở tri thức, từ đó cung cấp phản hồi hiệu quả và chính xác hơn.

1.5.3 Ứng dụng mô hình ngôn ngữ lớn để xây dựng chatbot

Các mô hình LLM ngày nay đã được chứng minh là có hiệu quả vượt bậc trong các tác vụ NLP so với trước đây, và do đó việc ứng dụng LLM vào nhiều lĩnh vực khác nhau đang ngày càng trở nên phổ biến [59]. Chính vì thế việc áp dụng một mô hình LLM phù hợp để xây dựng chatbot tuyển sinh là điều cần thiết và mang tính thực tiễn cao, giúp tối ưu hóa quy trình tuyển sinh và chăm sóc đối tượng khách hàng tiềm năng. Việc lựa chọn mô hình LLM phù hợp và huấn luyện trên lượng dữ liệu tuyển sinh chuyên biệt của Trường Đại học Bách khoa Đà Nẵng sẽ tăng hiệu quả trả lời và cung cấp thông tin chính xác hơn. Ngoài ra tâm lý của giới trẻ ngày nay đề cao sự riêng tư, do đó việc tương tác với một chatbot sẽ không ảnh hưởng đến sự riêng tư của người dùng mà còn tạo tâm lý thoải moái nếu người dùng có nhiều câu hỏi và muốn tìm hiểu nhiều vấn đề. Hình 20 dưới đây là sơ đồ tổng quan cho việc sử dụng LLM để xây dựng chatbot.



Hình 1.19 Tổng quan hệ thống Chatbot sử dụng LLM.

1.6 Các công cụ hỗ trợ

1.6.1 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao, đa năng, được Guido van Rossum phát triển vào năm 1991. Ngôn ngữ này nổi bật nhờ cú pháp rõ ràng, tập trung vào tính hiệu quả và khả năng ứng dụng thực tế trong nhiều lĩnh vực. [60]

Hình 1.20 Hình minh họa ngôn ngữ lập trình Python.

Python được sử dụng rộng rãi trong:

- Phát triển web: Cung cấp các framework như Django, Flask, hỗ trợ xây dựng ứng dụng từ cơ bản đến phức tạp.
- Khoa học dữ liệu và trí tuệ nhân tạo: Với các thư viện như NumPy, Pandas, TensorFlow, Python là công cụ quan trọng để phân tích dữ liệu và xây dựng hệ thống AI.
- Tự động hóa: Thích hợp để viết script xử lý các tác vụ lặp lại, từ quản lý tệp đến kiểm thử phần mềm.
- Hệ thống nhúng và IoT: Thường được dùng trên Raspberry Pi để phát triển các

giải pháp IoT hoặc các hệ thống nhúng đơn giản.

Điểm mạnh của Python là khả năng tương thích đa nền tảng, hệ sinh thái thư viện phong phú và cộng đồng phát triển mạnh mẽ, giúp đáp ứng tốt các nhu cầu từ lập trình cơ bản đến ứng dụng chuyên sâu. Với sự phong phú đa dạng các thư viện hỗ trợ trong lĩnh vực AI và NLP, đây là lý do nghiên cứu này lựa chọn ngôn ngữ Python để thực hiện xây dựng chatbot.

1.6.2 Các framework sử dụng

Bộ công cụ thiết lập giao diện chatbot Streamlit

Streamlit là một framework mã nguồn mở được thiết kế để hỗ trợ các nhà phát triển trong việc xây dựng, thử nghiệm và triển khai giao diện người dùng (UI) tương tác cho các ứng dụng sử dụng LLM. Streamlit giúp đơn giản hóa quy trình phát triển, cung cấp các công cụ trực quan để tương tác với mô hình và nhanh chóng tạo các nguyên mẫu ứng dụng AI có giao diện thân thiện. Chainlit đặc biệt phù hợp cho việc phát triển chatbot, trợ lý ảo, và các ứng dụng hội thoại phức tạp, cho phép nhà phát triển dễ dàng theo dõi, kiểm tra logic, và tối ưu hóa phản hồi của mô hình AI . Việc sử dụng Chainlit trong dự án này không chỉ giúp tiết kiệm thời gian và giảm độ phức tạp trong việc xây dựng ứng dụng chatbot mà còn cung cấp một môi trường mạnh mẽ để kiểm tra và tối ưu hóa các mô hình ngôn ngữ lớn. Nhờ các tính năng trực quan và tích hợp tốt, Chainlit là một công cụ lý tưởng để phát triển các ứng dụng NLP hiện đại.

Bộ công cụ lưu trữ dữ liệu vector store FAISS

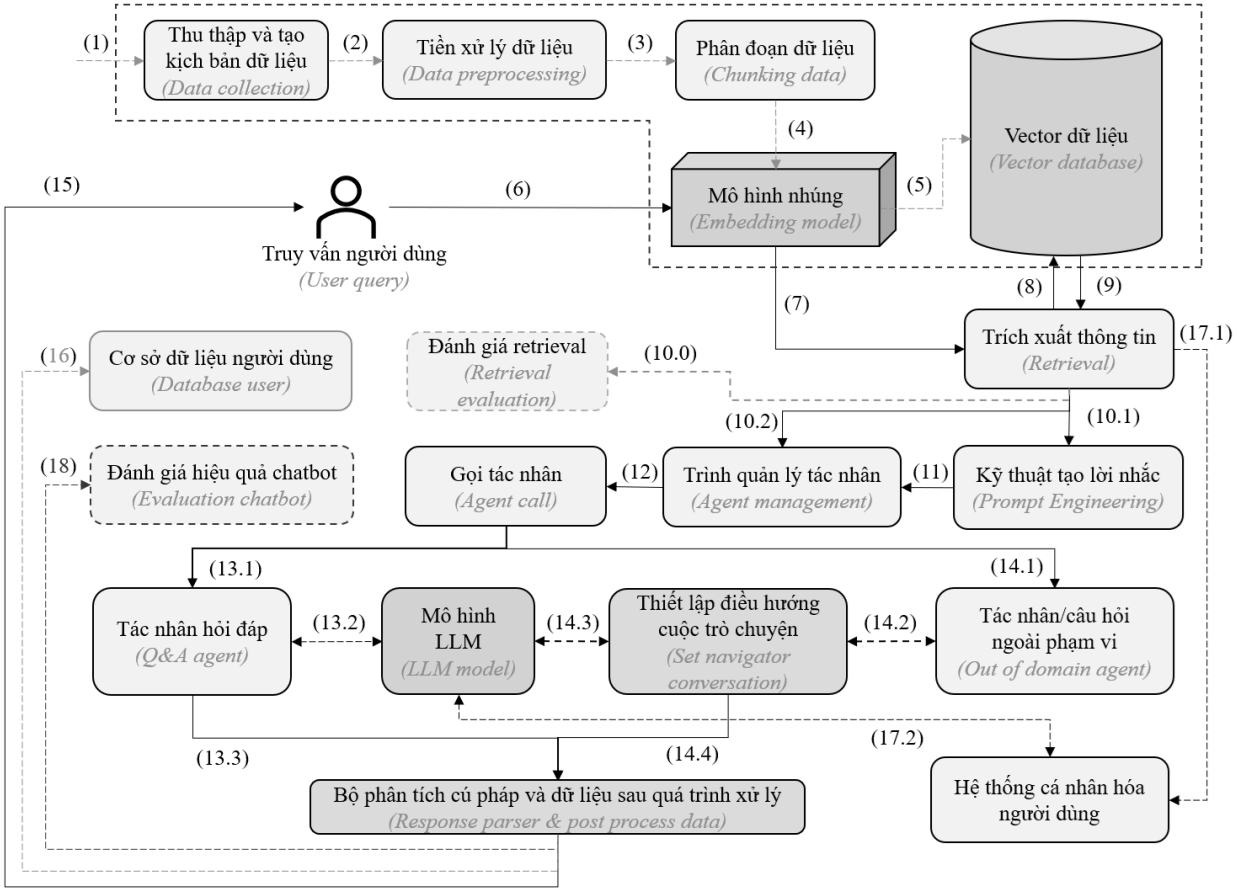
Faiss là một thư viện được phát triển bởi Facebook AI Research để tìm kiếm tương đồng hiệu quả và phân cụm các vector đặc gian. Nó chứa các thuật toán để tìm kiếm trong các tập hợp các vector có kích thước bất kỳ, thậm chí là các tập hợp có thể không vừa vào RAM. Nó cũng bao gồm mã hỗ trợ cho việc đánh giá và điều chỉnh tham số. Đối với việc xây dựng một hệ thống chatbot cần một cơ sở dữ liệu để lưu trữ thông tin nhúng văn bản đồng thời tìm kiếm chúng nhằm trả lời câu hỏi, lựa chọn FAISS trong dự án này là hợp lý khi nó cũng được ứng dụng rộng rãi trong các hệ thống chatbot khác.

Chương 1 đã tập trung vào việc giới thiệu về lý thuyết và các ngôn ngữ và các công nghệ được sử dụng trong dự án. Chúng ta đã tìm hiểu về khái niệm và tính năng của mỗi công nghệ, cùng với cách chúng liên kết và tương tác với nhau trong việc xây dựng hệ thống. Chương tiếp theo sẽ mô tả về rõ hơn về phương pháp đề xuất và thực hiện.

CHƯƠNG 2: PHƯƠNG PHÁP ĐỀ XUẤT VÀ THỰC HIỆN

2.1 Phương pháp đề xuất

Phương pháp đề xuất xây dựng chatbot tuyển sinh sử dụng mô hình ngôn ngữ lớn (LLM) được thiết kế nhằm tối ưu hóa quy trình tư vấn tuyển sinh, đảm bảo cung cấp thông tin nhanh chóng, chính xác và thân thiện với người dùng. Quy trình bao gồm từ việc thu thập và xử lý dữ liệu đến vận hành Chatbot, sử dụng các tác nhân thông minh và đánh giá hiệu quả. Hệ thống được chia thành 18 thành phần chính, mỗi thành phần đóng vai trò cụ thể trong việc xử lý truy vấn và cải thiện trải nghiệm người dùng. Hình 2.1 là sơ đồ tổng quan mô hình phương pháp đề xuất xây dựng chatbot sử dụng LLM.



Hình 2.1. Phương pháp đề xuất xây dựng chatbot hỏi đáp tuyển sinh sử dụng LLM.

Bước đầu tiên trong phương pháp đề xuất là thu thập và tạo kịch bản dữ liệu. Dữ liệu tuyển sinh được thu thập trên website tuyển sinh chính thức của Trường Đại học Bách khoa Đà Nẵng, đồng thời tôi cũng tạo ra các kịch bản câu hỏi có thể xảy ra dựa trên tình huống thực tế (1).

Bước thứ hai là thực tiện tiền xử lý dữ liệu, trong bước này thực hiện làm sạch dữ liệu, chuẩn hóa dữ liệu và gắn nhãn cho dữ liệu tùy theo từng danh mục hoặc pham vi

chủ đề câu hỏi (2).

Bước thứ ba là thực hiện phân đoạn dữ liệu, các mẫu có độ dài vượt quá khả năng xử lý của các mô hình LLM sẽ được cắt ngắn và sắp xếp lại để có thể dễ dàng xử lý và phân tích. Việc chia nhỏ giúp cải thiện tìm kiếm thông tin và giảm tải cho các mô hình xử lý dữ liệu lớn.

Bước thứ tư sau khi hoàn thành tất cả các quy trình chuẩn hóa dữ liệu sẽ cần nhúng dữ liệu thành các vector bằng mô hình Embedding để mô có thể tính toán sau này (4). Dữ liệu được chuyển đổi thành vector (dạng số học) thông qua mô hình nhúng, giúp biểu diễn ý nghĩa ngữ nghĩa của dữ liệu. Mô hình nhúng đảm bảo rằng các đoạn dữ liệu tương tự sẽ gần nhau trong không gian vector. Sau đó lưu trữ các vector embedded này vào một vector dabase. Điều này giúp tối ưu hóa quá trình truy xuất thông tin trong các bước sau.

Trong quá trình xử lý đầu vào của người dùng (5), đầu vào cũng phải được embedding để hệ thống có thể xử lý. Lúc này câu truy vấn đã được embedding này sẽ được dùng để truy xuất thông tin từ cơ sở dữ liệu vector database (6). Hệ thống RAG tìm kiếm thông tin phù hợp từ vector database bằng cách so sánh vector truy vấn của người dùng với các vector trong cơ sở dữ liệu (7). Từ đó các kết quả tương tự nhất sẽ được lấy ra (8).

Tiếp theo là thực hiện tạo ra các lời nhắc (prompt) tối ưu để định hướng đầu vào cho mô hình ngôn ngữ lớn (LLM). Các lời nhắc này được thiết kế sao cho LLM hiểu rõ yêu cầu của ngữ cảnh. Cùng với đó là việc thiết kế và tạo một hệ thống quản lý các tác nhân. Cần phải quản lý các tác nhân khác nhau trong hệ thống, đảm bảo rằng mỗi yêu cầu của người dùng được chuyển đến tác nhân phù hợp. Điều này bao gồm việc định tuyến yêu cầu tới mô hình LLM hoặc các tác nhân chuyên biệt. Trên thực tế, sau thi hệ thông RAG thực hiện truy xuất thông tin từ câu hỏi người dùng, các thông tin sẽ được truyền qua trình xử lý lời nhắc (9) và trình quản lý tác nhân, thông tin được trình xử lý lời nhắc sau khi xử lý cũng sẽ được chuyển tới trình quản lý tác nhân (10) để tổng hợp thông tin và thực hiện xử lý.

Khi đã có đầy đủ thông tin, trình quản lý tác nhân sẽ thực hiện gọi tác nhân (agent call) để phân luồng truy vấn đến tác nhân phù hợp(12). Tác nhân đặt lịch hẹn tư vấn (Appointment scheduling agent): Đảm nhiệm việc hỗ trợ người dùng đặt lịch hẹn và hành động đặt lịnh hẹn sau đó gửi tin nhắn tới người dùng (13), trong suốt quá trình này, hệ thống luôn tương tác với LLM để hoàn thiện đầu ra để tương tác với người dùng để tạo ra

câu trả lời phù hợp, dựa trên dữ liệu đầu vào và prompt đã được tối ưu hóa, Nếu người dùng yêu cầu đặt lịch, tác nhân đặt lịch sẽ kiểm tra thông tin và thực hiện quá trình đặt lịch, thông tin về lịch hẹn sẽ được lưu vào cơ sở dữ liệu người dùng.. Tác nhân hỏi đáp (Q&A agent) có nhiệm vụ giải đáp các câu hỏi thông thường trong phạm vi dữ liệu bằng cách tương tác với mô hình LLM (14). Tác nhân ngoài phạm vi (Out of domain agent) chịu trách nhiệm xử lý các câu hỏi hoặc các yêu cầu vượt ngoài phạm vi kiến thức hoặc dữ liệu đã có, tại đây cũng cần tương tác với LLM để đưa ra câu trả lời phù hợp (15). Nếu truy vấn không thuộc phạm vi dữ liệu, tác nhân ngoài phạm vi sẽ điều hướng cuộc trò chuyện. Mục đích là hướng người dùng đến một giải pháp hoặc gợi ý phù hợp (ví dụ: liên hệ nhân viên tư vấn trực tiếp).

Cuối cùng các đầu ra của các tác nhân này sẽ được truyền tới một bộ phân tích cú pháp để thực hiện phản hồi người dùng cuối cùng(16). Sau khi mô hình LLM hoặc các tác nhân trả lời, kết quả sẽ được phân tích và tối ưu hóa để đảm bảo dễ hiểu và chính xác cho người dùng. Tại đây quá trình chatbot tạo để lại lịch sử hỏi đáp sẽ được sử dụng để tạo dữ liệu lịch sử chatbot người dùng bằng MemGPT(17), đồng thời các lịch sử các cuộc trò chuyện cũng sẽ được sử dụng với mục đích thử nghiệm và đánh giá hiệu quả của toàn hệ thống chatbot (18).

Trên đây là chi tiết toàn bộ quy trình kì vọng sẽ được triển khai trong hệ thống Chatbot tuyển sinh. Trong quá trình trên đã ứng dụng các phương pháp hiện đại trong bối cảnh hiện tại. Đây là một quy trình chặt chẽ và có tính hệ thống cao. Bằng cách kết hợp công nghệ hiện đại và quy trình rõ ràng, phương pháp này đảm bảo Chatbot hoạt động hiệu quả, đáp ứng tốt nhu cầu người dùng và dễ dàng mở rộng trong tương lai và tích hợp với các hệ thống khác.

2.2 Thu thập và tạo dữ liệu

Dữ liệu tuyển sinh qua các năm thường thay đổi tùy theo cơ chế của từng trường. Việc thu thập dữ liệu chính xác là rất quan trọng, giúp mô hình phân tích đưa ra câu trả lời chính xác hơn thay vì chỉ mang tính chung chung. Trong nghiên cứu này, tôi tập trung thu thập dữ liệu tuyển sinh của Trường Đại học Bách khoa Đà Nẵng qua các năm, bao gồm thông tin chi tiết từng ngành học. Đồng thời thực hiện tạo dữ liệu kịch bản cho những câu hỏi mà sinh viên có thể sẽ quan tâm.

Cụ thể, đối với việc thu thập dữ liệu, tôi nhắm đến dữ liệu bao gồm chỉ tiêu tuyển

sinh, điểm chuẩn, tổ hợp xét tuyển, thông tin khối ngành xét tuyển và các phương thức xét tuyển như học bạ, điểm thi tốt nghiệp THPT, hoặc đánh giá năng lực. Ngoài ra những thông tin mà phụ huynh và học sinh thường quan tâm là chính sách học bổng và học phí, tôi thu thập thông tin chính sách học bổng các năm, các thông tin và thông báo cũng được xem xét để xây dựng dữ liệu tham khảo đối chiếu. Ngoài ra, chương trình đào tạo ngành học và cơ hội việc làm cũng được quan tâm lớn. Tôi thu thập các module chương trình đào tạo, nêu rõ các môn học cho mỗi ngành qua từng module, đồng thời thu thập những thông tin tổng quan về ngành để sinh viên có thể nắm rõ về ngành cụ thể mà sinh viên có sự quan tâm. Việc thu thập thông tin được lấy từ trang website chính thức của Trường Đại học Bách Khoa Đà Nẵng để đảm bảo tính chính xác nhất. Những thông tin này không chỉ giúp chatbot trả lời đúng trọng tâm câu hỏi mong muốn mà còn đáp ứng vấn đề thời gian.

Nhìn chung trong cả hai trường hợp, quy tắc thu thập dữ theo một khuôn mẫu cố định bao gồm chủ đề quan tâm – nội dung bao quát hoặc câu hỏi - trả lời. Đối với các mẫu có số lượng từ lớn hoặc quá dài, tôi thực hiện chia nhỏ dữ liệu theo từng đoạn theo phương pháp chunking. Sau khi thu thập và tạo dữ liệu, dữ liệu của tôi có 4 tập dữ liệu cho tất cả các ngành học và các lĩnh vực dữ liệu liên quan đến tuyển sinh. Bảng 4 dưới đây là minh họa một vài ví dụ về mẫu dữ liệu được thu thập:

Bảng 2.1 Một số dữ liệu đạt được qua thu thập

Câu hỏi	Nội dung phản hồi
Có bao nhiêu phương thức đăng ký xét tuyển?	Năm 2024 Trường Đại học Bách khoa, ĐHĐN xét tuyển sớm các phương thức Tuyển sinh riêng, Học bạ, Đánh giá năng lực. Đăng ký xét tuyển tại trang web tuyển sinh của Trường:https://dut.udn.vn/tuyensinh2024
Nguyên tắc xét tuyển theo Phương thức TSR như thế nào?	Xét tuyển theo thứ tự ưu tiên Điểm xét tuyển (ĐXT) từ cao xuống thấp và xét đến khi đủ chỉ tiêu. Đối với mỗi thí sinh, nếu đăng ký xét tuyển vào nhiều ngành thì việc xét tuyển được thực hiện theo thứ tự ưu tiên của các nguyện vọng; thí sinh chỉ trúng tuyển vào 01 nguyện vọng ưu tiên cao nhất có thể trong danh sách các nguyện vọng đã đăng ký Nguyện vọng 1 là nguyện vọng cao nhất. Trường hợp có nhiều thí sinh đồng hạng vượt quá chỉ tiêu sẽ xét đến tiêu chí phụ (xem Phụ lục I).Cách tính ĐXT các nhóm xét tuyển theo phương thức Tuyển sinh riêng (xem Phụ lục V). Thí sinh cung cấp thiếu minh chứng hoặc minh chứng không hợp lệ khi thực hiện đăng ký trực tuyến sẽ không được xét tuyển. Thí sinh không cung cấp minh chứng để hưởng chính sách ưu tiên theo đối tượng hoặc minh chứng không hợp lệ sẽ không được cộng điểm ưu tiên đối tượng.

Học ngành CNTT là học về cái gì?	Học công nghệ thông tin là học về cách sử dụng máy tính và hệ thống phần mềm để xử lý, lưu trữ, và truyền tải thông tin. Bạn sẽ được học lập trình, quản trị hệ thống, an ninh mạng, trí tuệ nhân tạo, phát triển ứng dụng, và các công nghệ mới trong lĩnh vực.
Học bạ THPT với giấy chứng nhận TN mình công chứng ở đâu ạ?	Em có thể phôtô học bạ và giấy chứng nhận tốt nghiệp đến trường THPT em đã học (nơi cấp) hoặc đến Phường/Xã (Bộ phận Tư pháp), các văn phòng công chứng để chứng thực em nhé!
Cách quy đổi điểm tuyển sinh riêng như thế nào?	Đối với Tuyển sinh riêng, cách Quy đổi điểm theo 7 nhóm sẽ như sau: Điểm quy đổi nhóm 1: Giải Khuyến khích HSG cấp QG: 300 điểm; Giải Khuyến khích KHKT cấp QG: 300 điểm. Điểm quy đổi nhóm 2: Giải nhất: 290 điểm; Giải nhì: 280 điểm; Giải ba: 270 điểm; Giải khuyến khích: 260 điểm Điểm quy đổi nhóm 3: Giải nhất: 250 điểm; Giải nhì: 240 điểm;Giải ba: 230 điểm; Giải Khuyến khích: 220 điểm. Điểm quy đổi nhóm 4: 220 Điểm quy đổi nhóm 5: 210 Điểm quy đổi nhóm 6: Điểm SAT trên 1400, hoặc Điểm ACT trên 30: 205 điểm; Điểm SAT từ 1100 đến 1400, hoặc Điểm ACT từ 24 đến 30: 200 điểm Điểm quy đổi nhóm 7: IELTS trên 6.5, TOELF IBT trên 93, TOIEC trên 845: 200 điểm; IELTS từ 6.0 đến 6.5, TOELF IBT từ 80 đến 93, TOIEC từ 800 đến 845: 195 điểm; IELTS từ 5.5 đến dưới 6.0, TOELF IBT từ 46 dưới 80, TOIEC từ 600 đến 800: 190 điểm

2.3 Xử lý và phân đoạn dữ liệu

Để xây dựng hệ thống chatbot tuyển sinh, việc chuẩn bị dữ liệu và phân đoạn dữ liệu là một bước quan trọng nhằm đảm bảo chất lượng và hiệu quả trong các giai đoạn xử lý tiếp theo.

Sau khi hoàn thành thu thập dữ liệu tuyển sinh từ các nguồn tin cậy từ [61], [62], cần kiểm tra các mẫu dữ liệu để đảm bảo tính chính xác và đúng đắn. Tiếp theo tôi thực hiện loại bỏ các thông tin có thể gây nhiễu hoặc không cần thiết, làm sạch các lỗi chính tả hoặc các ký tự đặc biệt. Để đảm bảo Chatbot có khả năng hiểu tốt hơn về ngữ cảnh, tôi

tiến hành xử lý các trường hợp có chữ viết tắt. Nắm bắt xu hướng học sinh thường có hay có thói quen viết tắt các từ khóa thông dụng, chính vì thế chúng tôi thực hiện tái tạo thêm mẫu dữ liệu với các câu trả lời có chữ viết tắt để tăng thêm hiệu quả trả lời của chatbot. Cụ thể tôi tái tạo các câu trả lời với các từ khóa thông dụng như "Công nghệ thông tin" thành các từ khóa "CNTT" hoặc "cntt", một ví dụ khác như từ khóa "Trung học Phổ thông" sẽ được tái tạo thêm các mẫu như là "THPT" hoặc "thpt" và các từ khóa các ngành các nữa. Sau đó đối với mỗi mẫu dữ liệu, chúng tôi thực hiện kiểm tra lại văn phong câu trả lời và hiệu chỉnh để câu trả lời trôi chảy tự nhiên hơn. Cuối cùng là thực hiện gắn nhãn dữ liệu có cùng chủ đề với nhau, các câu hỏi có tính liên quan này sẽ được đề xuất trong suốt quá trình chatbot, điều này làm tăng kích thích người dùng tìm hiểu sâu hơn về một chủ đề người dùng quan tâm.

Sau khi làm sạch dữ liệu, tôi thực hiện kiểm tra các mẫu có độ dài vượt quá kích thước xử lý của các mô hình embedding và LLM, ta thực hiện phân đoạn lại các mẫu trả lời và hiệu chỉnh lại văn phong của dữ liệu được trôi chảy và tự nhiên hơn.

2.4 Lựa chọn mô hình Embedding

So sánh nhiều mô hình giúp mang lại góc nhìn toàn diện và tối ưu hóa việc lựa chọn mô hình phù hợp. Đầu tiên, việc đưa ra nhiều mô hình và đánh giá giúp hiểu rõ các ưu điểm, hạn chế, cũng như hiệu suất thực tế của từng mô hình trên các tiêu chí cụ thể như tốc độ, độ chính xác, và tài nguyên yêu cầu. Điều này đặc biệt quan trọng khi ứng dụng các mô hình trong bối cảnh cụ thể, nơi hiệu suất thực tế có thể thay đổi tùy thuộc vào dữ liệu hoặc nhu cầu. Hơn thế nữa, so sánh nhiều mô hình mang lại lợi ích về mặt kinh tế và chiến lược giúp chọn được giải pháp tối ưu về chi phí và hiệu năng, đồng thời giảm thiểu rủi ro khi triển khai [63, [64].

- **Mô hình Sentence Transformers - all-MiniLM-L6-v2**: Được phát triển bởi nhóm nghiên cứu Sentence Transformers, dẫn đầu bởi Nils Reimers, trong cộng đồng Hugging Face. Được công bố vào năm 2020-2021 như một phần của bộ mô hình sử dụng MiniLM của Microsoft. Mô hình này được tối ưu hóa để mã hóa văn bản thành vector 384 chiều, hỗ trợ các tác vụ như tìm kiếm ngữ nghĩa (semantic search), cụm văn bản (clustering) và đo độ tương tự (sentence similarity). Nó nổi bật vì sự cân bằng giữa hiệu suất và tốc độ xử lý. Dữ liệu huấn luyện trên hơn 1 tỷ cặp câu sử dụng kỹ thuật học tương phản (contrastive learning), kết hợp dữ liệu từ

nhiều nguồn như Reddit, WikiAnswers, và Stack Exchange. Mô hình này miễn phí và mã nguồn mở, đây là một mô hình nhẹ và rất phổ biến trong các ứng dụng tìm kiếm ngữ nghĩa và embedding văn bản

- **Mô hình nomic-ai/nomic-embed-text-v1.5**: Được phát triển bởi Nomic AI, một công ty tập trung vào xây dựng công cụ cho hệ thống NLP, xuất hiện trong năm 2022 như một giải pháp tối ưu hóa embedding cho các ứng dụng tìm kiếm văn bản. Mục tiêu của Nomic là tập trung vào việc xử lý dữ liệu văn bản lớn với hiệu quả cao, chủ yếu hướng tới các ứng dụng trong tìm kiếm và phân tích ngữ nghĩa. Ưu điểm của Nomic nằm ở chỗ khả năng embedding văn bản nhanh với hiệu suất phù hợp cho các tác vụ xử lý ngôn ngữ quy mô lớn (large-scale text processing) [65]. Mô hình này là miễn phí, cung cấp cho cộng đồng sử dụng thông qua các nền tảng mã nguồn mở như Hugging Face. Đây là một lựa chọn phù hợp cho các tác vụ embedding văn bản [66].
- **Mô hình Alibaba-NLP/gte-multilingual-base**: Ra đời bởi Alibaba NLP Team. Mô hình này được công bố vào khoảng năm 2022. Mục tiêu của nó là hướng tới việc xử lý ngôn ngữ đa ngữ (multilingual), mô hình này hỗ trợ embedding cho hơn 100 ngôn ngữ. Đây là một giải pháp mạnh mẽ cho các hệ thống toàn cầu, yêu cầu hỗ trợ nhiều ngôn ngữ trong các tác vụ như tìm kiếm đa ngữ hoặc dịch thuật. Ưu điểm tích hợp dễ dàng vào các pipeline đa ngữ, với hiệu suất cao trên nhiều ngữ cảnh. Mô hình này miễn phí và mã nguồn mở, được cung cấp bởi nhóm Alibaba NLP. Mô hình này có thể tải về từ Hugging Face.
- **Mô hình BAAI/bge-m3**: Ra đời bởi Beijing Academy of Artificial Intelligence (BAAI), Trung Quốc. Được giới thiệu trong năm 2021. Mục tiêu là được thiết kế đặc biệt cho các tác vụ tìm kiếm văn bản và hiểu ngữ nghĩa sâu hơn trong văn bản lớn. Mô hình này hướng tới cải thiện độ chính xác của các tác vụ embedding thông qua việc sử dụng kỹ thuật học sâu (deep learning) với kiến trúc hiện đại. Dữ liệu huấn luyện được tận dụng kho dữ liệu phong phú từ các nguồn Trung Quốc và quốc tế, giúp hỗ trợ tốt cả ứng dụng đơn ngữ và đa ngữ. Mô hình này miễn phí do Beijing Academy of Artificial Intelligence phát hành trên nền tảng Hugging Face. Nó được sử dụng rộng rãi trong các tác vụ tìm kiếm và xử lý ngữ nghĩa.
- **jinaai/jina-embeddings-v3:** Được phát hành vào năm 2024 bởi Jina AI, jina-embeddings-v3 là mô hình nhúng văn bản đa ngôn ngữ đa tác vụ được thiết kế cho

nhiều ứng dụng NLP. Dựa trên kiến trúc Jina-XLM-RoBERTa, mô hình này hỗ trợ Nhúng vị trí quay (Rotary Position embedding) để xử lý chuỗi đầu vào dài lên đến 8192 tokens, điều này giúp mở rộng độ dài ngữ cảnh, điều này làm tăng hiệu suất ghi nhớ và hiểu ngữ cảnh, tuy nhiên điều này cũng tăng thêm nhu cầu tài nguyên tính toán. Mô hình này hỗ trợ hơn 30 ngôn ngữ trên thế giới trong đó có tiếng Việt, và mở cho mục đích nghiên cứu.

Bảng 2.2. **S***o sánh hiệu suất RAG trên một số mô hình LLM Embedding*

Bangr so sánh hiệu suất tăng cường truy xuất tạo sinh (RAG) trên một số mô hình LLM.

Top 1	Top 3	Top 5	Top 10	Số chiều	Độ dài bối cảnh	Số lượng tham số	Không gian GPU	Trung bình độ dài câu	Trung bình thời gian nhúng câu	Trung bình độ dài câu hỏi	Trung bình thời gian nhúng câu hỏi
0,233	0,436	0,519	0,672	384	256	22.713.216	0,109	2.058,935	0,015	185,543	0,005
0,352	0,579	0,651	0,722	768	8.192	136.731.648	0,656	2.58,935	0,070	185,543	0,013
0,379	0,615	0,707	0,860	768	8.192	305.368.320	1,466	1.744,195	0,056	166,484	0,010
0,555	0,827	0,875	0,937	1.024	8.192	567.754.762	2,725	1.745,013	0,170	166,484	0,021

	Tên mô hình
	Sentence-transformer/all-MiniLM-L6-v2
	Nomic-ai/nomic-embeded-text-v1.5
	Alibaba-NLP/gte-multilingual-base
	BAAI/bge-m3

Trong bảng 2.2 so sánh hiệu suất tăng cường truy xuất tạo sinh (RAG) cho một số mô hình LLM nổi bật miễn phí. Trong đó, chỉ số Top k (Top-K Accuracy) biểu thị tỷ lệ phần trăm các câu trả lời đúng xuất thành công xuất hiện trong danh sách kết quả đứng đầu (tương ứng với 1, 3, 5 hoặc 10 kết quả đầu tiên). Các giá trị này đánh giá khả năng truy xuất chính xác của mô hình. Mô hình có giá trị cao hơn có nghĩa là tốt hơn trong việc xếp hạng các kết quả phù hợp với câu hỏi truy vấn.

Số chiều (Dimensions) là kích thước của vector embedding được tạo bởi mô hình. Ví dụ: 384, 768, hoặc 1.024 chiều. Vector có số chiều cao hơn thường nắm bắt được nhiều đặc trưng ngữ nghĩa hơn, nhưng cũng đòi hỏi nhiều bộ nhớ và tính toán hơn khi thực hiện so sánh vector.

Độ dài bối cảnh (Context Length) là số token tối đa mà mô hình có thể xử lý trong một lần (context window). Mô hình có độ dài bối cảnh cao hơn có thể xử lý các câu dài hoặc đoạn văn lớn hơn, phù hợp cho các ứng dụng như tìm kiếm hoặc phân tích văn bản dài.

Số lượng tham số (Parameters) là tổng số tham số trong mô hình, thường biểu thị kích thước của mô hình. Mô hình lớn hơn (số tham số cao) thường mạnh mẽ hơn trong việc nắm bắt các mối quan hệ ngữ nghĩa phức tạp, nhưng tốn nhiều tài nguyên tính toán hơn.

Không gian GPU là lượng bố nhớ GPU chiếm dụng khi triển khai truy vấn.

Trung bình độ dài câu biểu thị độ dài trung bình của các câu hoặc đoạn văn được nhúng (embedding). Thường được tính bằng số token hoặc ký tự. Độ dài lớn hơn có thể yêu cầu mô hình thực hiện nhiều bước tính toán hơn, ảnh hưởng đến tốc độ và hiệu suất.

Trung bình thời gian nhúng câu (Seconds) là thời gian trung bình để mã hóa một đoạn văn bản (hoặc câu) thành vector embedding. Giá trị này quan trọng trong các ứng

dụng xử lý dữ liệu lớn, vì nó ảnh hưởng đến hiệu suất tổng thể khi xử lý hàng loạt dữ liệu.

Trung bình độ dài câu hỏi biểu thị độ dài trung bình của các câu hỏi (số lượng token hoặc ký tự) được sử dụng để kiểm tra mô hình. Độ dài này ảnh hưởng đến tốc độ xử lý vì câu hỏi dài hơn yêu cầu nhiều tính toán hơn.

Trung bình thời gian nhúng câu hỏi (Seconds) là thời gian trung bình (tính bằng giây) để mã hóa một câu hỏi thành vector embedding. Mô hình có giá trị thấp hơn sẽ nhanh hơn trong việc xử lý các truy vấn, phù hợp cho các ứng dụng thời gian thực hoặc yêu cầu độ trễ thấp.

Trong phạm vi đồ án này, tôi sẽ chọn mô hình **all-MiniLM-L6-v2** bởi mô hình đáp ứng tốt các tiêu chí quan trọng trong việc nhúng văn bản, bao gồm khả năng xử lý chính xác, hiệu quả và linh hoạt. Trước tiên, mô hình có chỉ số Top-k Accuracy cao, cho phép truy xuất các câu trả lời đúng một cách hiệu quả trong danh sách các kết quả hàng đầu, đảm bảo chất lượng và độ tin cậy trong các tác vụ tìm kiếm và phân loại. Mô hình cũng có yêu cầu không gian GPU thấp, dễ triển khai trên máy tính cá nhân với cấu hình trung bình, đáp ứng tốt nhu cầu nhúng văn bản với hiệu năng và đo chính xác cân bằng nhất.

2.5 Mô hình LLM sử dụng

Vào tháng 4 năm 2024, Meta đã tiếp tục khẳng định vị thế trong lĩnh vực AI tạo sinh bằng cách ra mắt mô hình ngôn ngữ lớn Llama 3, cung cấp mã nguồn mở và miễn phí cho cả mục đích nghiên cứu lẫn thương mại (với giới hạn giấy phép chỉ áp dụng cho các công ty có hơn 700 triệu người dùng hoạt động hàng tháng) [67]. Trong khi đó, OpenAI vẫn giữ nguyên cách tiếp cận độc quyền với các mô hình GPT của mình, như GPT-4, với các chi tiết hoạt động nội bộ và dữ liệu đào tạo được giữ kín. Sự khác biệt này nhấn mạnh hai triết lý phát triển AI: Meta thúc đẩy sự minh bạch và hợp tác, trong khi OpenAI ưu tiên kiểm soát và quản lý chặt chẽ.

Llama 3

Mô hình này đã được tiền huấn luyện bằng các nguồn dữ liệu trực tuyến công khai, chẳng hạn như Common Crawl, Wikipedia và Project Gutenberg. Tại thời điểm viết, dữ liệu tiền huấn luyện lên đến tháng 12 năm 2023, mặc dù một số dữ liệu tinh chỉnh mới hơn, lên đến tháng 2 năm 2024. Mô hình Llama 3 có ba biến thể kích thước (dựa trên số lượng tỷ tham số): 8B, 70B và thậm chí là 405B. Hãy nhớ rằng, trong học máy, các tham số là

các biến có trong mô hình trong quá trình huấn luyện, đóng vai trò như "kho kiến thức của mô hình." Các biến thể kích thước nhỏ hơn sẽ chạy nhanh hơn nhưng tạo ra đầu ra có chất lượng thấp hơn [68].

GPT-4

Được OpenAI tiền huấn luyện bằng dữ liệu công khai, bao gồm dữ liệu từ internet và dữ liệu được OpenAI cấp phép. Theo họ, bộ dữ liệu này bao gồm: "các lời giải đúng và sai cho các bài toán, lý luận yếu và mạnh, các tuyên bố tự mâu thuẫn và nhất quán, và đại diện cho nhiều hệ tư tưởng và ý tưởng khác nhau." Tại thời điểm này, dữ liệu huấn luyện cập nhật đến tháng 9 năm 2021 [69]. Mô hình GPT-4 cung cấp các biến thể dựa trên độ dài ngữ cảnh tối đa: 8K và 32K token. Kích thước của GPT-4 ước tính là 1,76 nghìn tỷ tham số.

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	**86.1**	85.0
Natural Questions (1-shot)	–	–	29.3	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	**65.7**	51.2

Hình 2.2 Các tiêu chí đánh giá hiệu suất cho một số mô hình ngôn ngữ lớn.

Từ kết quả benchmark của nhiều mô hình đánh giá hiệu quả, ta thấy được điểm của OpenAI GPT-4 khá vượt trội. Tuy nhiên, với bản chất là một mô hình mã nguồn đóng với kích thước lớn và phải trả phí, GPT-4 không phù hợp với phạm vi của đề tài này. Trong khi đó, mô hình mã nguồn mở LLama 3 có hiệu suất vượt trội hơn trên tất cả các tham số so với phiên bản tiền nhiệm Llama 2 [70], bao gồm khả năng hiểu ngữ nghĩa, bối cảnh, xử lý các tác vụ phức tạp và đặc biệt nó là mô hình mã nguồn mở, tỏ ra phù hợp với ứng dụng của đề tài.

2.6 Phương pháp đánh giá hiệu quả chatbot

Một phương pháp đánh giá mô hình LLM là cần thiết để đảm bảo chatbot tuyển sinh hoạt động hiệu quả, đáp ứng đúng nhu cầu người dùng, cải thiện trải nghiệm giao tiếp, tính chính xác ngữ nghĩa, và sự phù hợp ngữ cảnh. Đánh giá còn cung cấp dữ liệu để cải tiến, tối ưu hóa hiệu suất, và đảm bảo chatbot đáng tin cậy trong các ứng dụng thực tế.

Trong dự án này chúng tôi đánh giá hiệu quả của cho một mô hình LLM bằng phương pháp Deep Eval, một framework đánh giá LLM mã nguồn mở [71].

CHƯƠNG 3: TRIỂN KHAI XÂY DỰNG HỆ THỐNG VÀ KẾT QUẢ

3.1 Triển khai phân chia văn bản thành đoạn nhỏ

Text chunking là bước quan trọng trong hệ thống Retrieval-Augmented Generation, đặc biệt khi làm việc trên máy chủ cá nhân với phần cứng có hạn. Quá trình này giúp chia nhỏ dữ liệu văn bản thành các đoạn (chunks) có kích thước hợp lý, phù hợp với khả năng xử lý của mô hình ngôn ngữ lớn như Llama 3.

Mục tiêu của Text chunking là để tang hiệu quả truy xuất và tối ưu truy vấn người dung. Các mô hình LLM đều có giới hạn về độ dài ngữ cảnh, việc chia nhỏ tài liệu đảm bảo rằng dữ liệu không vượt quá giới hạn này. Chunks nhỏ gọn và độc lập giúp tăng chính xác khi thực hiện tìm kiếm tương tự. Các đoạn phải đủ ngắn để tang chính xác nhưng cũng phải đủ dài để tránh trích xuất thông tin không toàn vẹn.

Quy trình Text chunking:

- Các tài liệu đưa vào được chuyển thành định dạng pdf và được đọc bở hàm thư viện LlamaParse.
- Chia tài liệu thành các đoạn nhỏ, mỗi đoạn có độ dài 500 kí tự.
- Để bảo toàn tính logic và đầy đủ ngữ nghĩa, đặt độ xếp chồng giữa mỗi đoạn chunk là 50 kí tự.
- Đính kèm metadata, các thông tin bổ sung cho từng chunk, như nguồn, vị trí trong tài liệu, hoặc loại nội dung.

Quy trình này đảm bảo rằng dữ liệu được tổ chức hợp lý, dễ dàng truy xuất và tối ưu hóa khả năng phản hồi từ mô hình LLM. Sau khi chia nhỏ, các chunks được mã hóa (embedding) và lưu trữ trong vector database (FAISS). Mỗi chunk đi kèm với metadata, đảm bảo khả năng tìm kiếm nhanh và chính xác.

3.2 Tạo dựng vector store

FAISS (Facebook AI Similarity Search) là thư viện mã nguồn mở được thiết kế để tìm kiếm nhanh các vector và thực hiện so khớp ngữ nghĩa trên cơ sở dữ liệu lớn. FAISS cho phép xử lý các truy vấn vector một cách hiệu quả nhờ vào các thuật toán tối ưu hóa và hỗ trợ GPU để tăng tốc tính toán.

Tạo Embedding cho dữ liệu :

- Sử dụng mô hình Sentence Tranformer **all-MiniLM-L6-v2** để chuyển đổi các tập tài liệu PDF tuyển sinh thành vector số.

- Lưu vector này kèm theo thông tin metadata (ID câu trả lời, nguồn tài liệu).

Khởi tạo chỉ mục FAISS:

- Sử dụng Flat Index do dữ liệu tương đối nhỏ, cần tìm kiếm nhanh và chính xác.
- Đưa các vector sau embedding vào chỉ mục.
- Lưu trữ vector store: Dữ liệu vector được tuần tự hóa (serialize), sau đó lưu vào 2 file với định dạng .faiss và .pkl.
- Khi cần truy vấn vào vector store, dữ liệu sẽ được giải tuần tự hóa (deserialize) và đưa vào hàm tính toán. Điều này cho phép tái sử dụng chỉ mục mà không cần xây dựng lại.

3.3 Triển khai nhúng truy vấn người dùng

Nhúng truy vấn (Embedding queries) là quá trình chuyển đổi câu hỏi hoặc truy vấn của người dùng từ dạng văn bản thành biểu diễn vector số, giúp mô hình có thể hiểu và so sánh ý nghĩa ngữ nghĩa của câu hỏi với dữ liệu đã được lưu trữ. Phương pháp này rất quan trọng trong RAG, song song với việc nhúng dữ liệu.

Quá trình nhúng truy vấn diễn ra tương tự như bước xây dựng vector :

- Câu hỏi của người dùng được đưa truy vấn qua mô hình nhúng để sinh ra vector embedding.
- Ví dụ: Câu hỏi "Điểm chuẩn ngành Công nghệ Thông tin là bao nhiêu?" được mã hóa thành một vector có 384 chiều.
- Vector embedding của truy vấn sau khi chuẩn hóa sẽ được sử dụng để so sánh với các vector trong cơ sở dữ liệu (thông qua FAISS) để tìm kiếm kết quả phù hợp nhất.

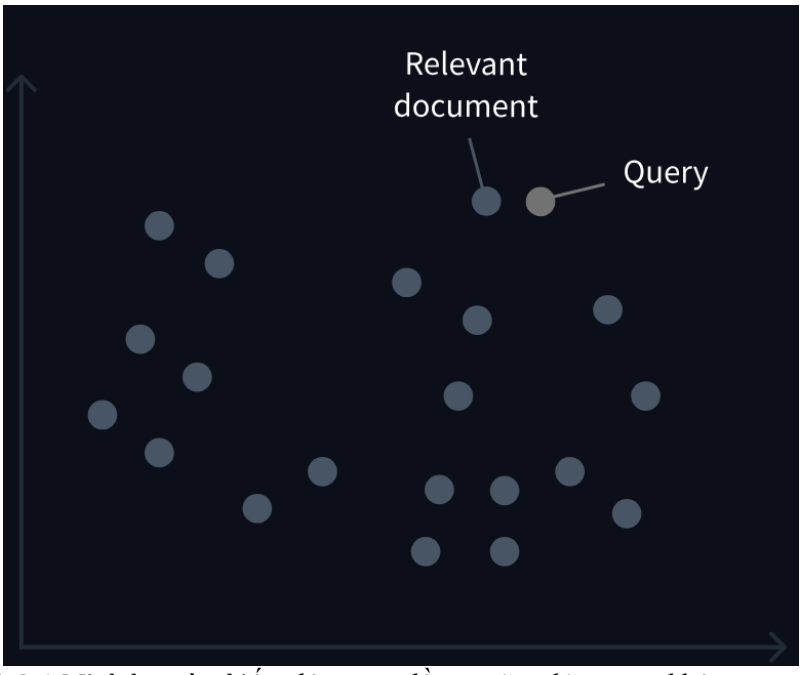
3.4 Triển khai thành phần Retrieval

Retrieval là một thành phần quan trọng trong mô hình RAG, đóng vai trò như cầu nối giữa nguồn dữ liệu bên ngoài và mô hình sinh ngôn ngữ. Cụ thể, nó thực hiện các nhiệm vụ sau:

- Retrieval giúp trích xuất độ tương đồng giữa truy vấn và cơ sở dữ liệu database. Mục tiêu là cung cấp thông tin ngữ cảnh cụ thể và liên quan nhất để hỗ trợ quá trình sinh câu trả lời.

- Mô hình sinh thường gặp khó khăn khi xử lý thông tin chi tiết như điểm trúng tuyển. Retrieval đảm bảo rằng mô hình chỉ cần làm việc với một phần dữ liệu nhỏ, chính xác và có liên quan, giúp cải thiện hiệu suất và độ chính xác.

Tóm lại, Retrieval không chỉ giúp RAG trở nên hiệu quả hơn trong việc sinh ngôn ngữ tự nhiên mà còn đảm bảo thông tin đầu ra có độ chính xác cao và đúng ngữ cảnh tuyển sinh đại học.



Hình 3.1 Minh họa tìm kiếm độ tương đồng ngữ nghĩa trong không gian vector

Datasets tuyển sinh và câu hỏi truy vấn từ người dùng đều đã được biểu diễn trên cấu trúc dữ liệu FAISS Index. Cũng giống như các tài liệu về tuyển sinh, ta có vector 384 chiều đại diện cho truy vấn, mà chúng ta có thể so sánh với toàn bộ kho dữ liệu để tìm ra các cách biểu diễn tương tự nhất. Vector truy vấn càng gần với thông tin nào trong không gain vector, chúng càng tương đồng về ngữ nghĩa và càng có khả năng chứa thông tin mà người dùng mong muốn.

Áp dụng kỹ thuật Reranking, chatbot còn tích hợp một thuật toán tương tự vào Retrieval đó là BM25. BM25 (Best Matching 25) là một thuật toán xếp hạng dựa trên xác suất, thường được sử dụng để truy xuất tài liệu liên quan trong hệ thống tìm kiếm. Thuật toán này tập trung trọng tâm vào sự xuất hiện của các từ khóa trong tài liệu và tính điểm liên quan giữa truy vấn và các tài liệu trong Datasets. Kết hợp cùng với khả năng tìm kiếm tương đồng ngữ cảnh của FAISS, chatbot sẽ chính xác hơn trong việc hiểu ngữ cảnh và trong quá trình sinh câu trả lời. Tìm kiếm tương tự với FAISS chiếm trọng số

0.7, trong khi thuật toán BM25 sẽ chiếm 30% của hàm Retrieval. Số vector tương đồng mà hàm Retrieval trả về 4, nghĩa là mô hình sinh sẽ tính toán dựa trên 4 vector gần nhất với vector truy vấn trong tập dữ liệu tuyển sinh.

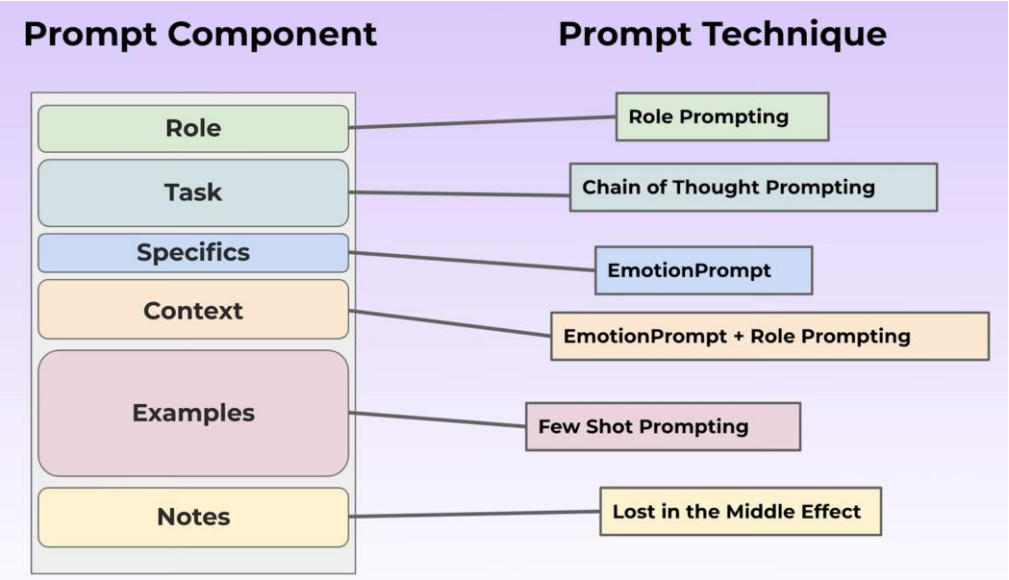
3.5 Triển khai mô hình ngôn ngữ lớn

Như đã đề cập ở chương trước, chương trình sẽ sử dụng Llama 3 làm mô hình sinh để tạo câu trả lời cho chatbot. Llama 3 được phát triển bởi Meta, được tối ưu hóa để sử dụng trong các tác vụ đối thoại và đạt hiệu suất vượt trội so với nhiều mô hình nguồn mở khác, đặc biệt là trên nền ngôn ngữ Tiếng Việt. Để sử dụng mô hình Llama 3 trong hệ thống chatbot của chúng ta, Ollama sẽ được sử dụng như một công cụ để chạy mô hình này, giúp tối ưu hóa chatbot nhờ khả năng tận dụng phần cứng máy chủ.

Ollama là một công cụ nguồn mở cho phép các nhà phát triển dễ dàng chạy các mô hình ngôn ngữ lớn trên máy tính cá nhân hoặc máy chủ. Với Ollama, việc triển khai các mô hình như Llama 3 trở nên rất thuận tiện. Ollama cung cấp các API giúp kết nối và tương tác với mô hình Llama 3 thông qua các câu lệnh đơn giản, giảm bớt yêu cầu về phần cứng và làm cho quá trình tích hợp vào chatbot trở nên nhanh chóng và tiết kiệm tài nguyên.

Để kết nối mô hình Llama 3 với hệ thống chatbot, ta sẽ sử dụng Ollama để tạo các API endpoint mà chatbot có thể gửi các truy vấn tới. Mỗi truy vấn sẽ được xử lý bởi mô hình Llama 3, và mô hình sẽ trả về các phản hồi tự động. Để tối ưu hóa việc gọi và xử lý truy vấn từ Llama 3, chương trình áp dụng kỹ thuật như caching, giúp lưu trữ tạm thời các câu trả lời cho các truy vấn lặp lại để tăng tốc độ phản hồi. Việc Ollama đã tận dụng được dGPU để chạy mô hình sinh đã giảm tải rất nhiều công việc cho CPU xử lý các tác vụ khác.

3.6 Triển khai prompt trả lời câu hỏi



Hình 3.2 Các thành phần và kỹ thuật Prompting thông dụng

Prompts đóng vai trò quan trọng trong việc định hướng câu trả lời của mô hình ngôn ngữ lớn như Llama 3. Một prompt tốt giúp chatbot trả lời chính xác, tự nhiên và phù hợp với bối cảnh. Cách xây dựng và tối ưu prompts có thể dễ dàng nâng độ chính xác của câu trả lời của chatbot tuyển sinh. Prompt được xây dựng dưới dạng câu lệnh hoặc câu hỏi, cung cấp đủ ngữ cảnh để LLM hiểu và phản hồi chính xác. Các thành phần chính của một prompt gồm:

- Ngữ cảnh: Thông tin nền để mô hình hiểu vấn đề.
- Câu hỏi: Yêu cầu cụ thể của người dùng.
- Hướng dẫn: Quy định cách trả lời (ngắn gọn/chi tiết…).

Dưới đây là prompt chính của chatbot**:**

'Bạn là tư vấn viên tuyển sinh của trường Đại học Bách khoa, Đại học Đà Nẵng. Công việc của bạn là trả lời câu hỏi của các thí sinh muốn theo học tại trường về các thắc mắc về vấn đề tuyển sinh. Hãy luôn trả lời bằng ngôn ngữ tiếng việt. Hãy cố gắng trả lời thông tin chính xác, đừng cố sáng tạo. Nếu bạn không chắc hay không có câu trả lời cho vấn đề được hỏi, hãy phản hồi là: Hiện tại tôi không có thông tin cho câu hỏi của bạn. Hãy gửi câu hỏi vào hotrotuyensinh@dut.udn.vn để được tư vấn thêm.'

Ngoài prompt chính, như trên, chatbot còn có prompt riêng tùy thuộc vào việc người dung muốn hỏi về lĩnh vực gì về tuyển sinh. Datasets được chia làm nhiều Collection khác nhau, bao gồm: ngành học, đề án tuyển sinh và Q&A. Trên mỗi Data Collection, có gắn một prompt giúp Retrieval giúp xác định chọn Collection nào để thực hiện tìm kiếm thông tin.

- Prompt ngành học: *"Nơi này chứa thông tin về các trường đại học và thông tin các ngành học. Hãy giải thích rõ về chương trình đào tạo và cơ hội nghề nghiệp của ngành."*

- Prompt đề án tuyển sinh: *"Nơi này chứa thông tin về chỉ tiêu tuyển sinh, điểm chuẩn, điều kiện, thủ tục tuyển sinh"*
- Prompt về Q&A: *"Ở đây chứa thông tin về các câu hỏi thông thường nằm ngoài chủ đề về trường, ngành học và đề án tuyển sinh. Nếu không có câu trả lời thì hãy phản hồi: Hãy gửi câu hỏi vào tuyensinh2024.dut.vn để được giải đáp thắc mắc."*

Việc xây dựng prompts hiệu quả là yếu tố then chốt để chatbot hiểu đúng ý định của người dùng và cung cấp câu trả lời phù hợp. Quy trình thiết kế prompts cần dựa trên nhu cầu thực tế, kiểm tra và tối ưu liên tục để đảm bảo chất lượng cho chatbot tư vấn tuyển sinh. Những prompts ở trên vẫn chưa hoàn thiện mà sẽ luôn được đánh giá và tinh chỉnh theo quá trình đánh gái hiệu quả chatbot

3.7 Triển khai chatbot với giao diện

Giao diện người dùng là phần không thể thiếu trong hệ thống chatbot hỗ trợ tuyển sinh. Sử dụng Streamlit, một thư viện Python mã nguồn mở, được thiết kế để giúp các nhà phát triển nhanh chóng tạo ra ứng dụng web tương tác cho chatbot. Với Streamlit, ta có thể chuyển đổi mã Python thành giao diện web dễ sử dụng mà không cần sử dụng HTML, CSS hay JavaScript. Giao diện cho phép:

- Người dùng gửi câu hỏi liên quan đến tuyển sinh.
- Hiển thị câu trả lời từ mô hình AI.
- Tùy chỉnh nguồn dữ liệu và mô hình AI để phù hợp.

Trong quá trình khởi tạo:

- Đọc các thông số cấu hình từ file .env (chứa thông tin về các biến API).
- Cài đặt giao diện trang web với tiêu đề, biểu tượng, và bố cục.

Khởi tạo thanh công cụ sidebar bên trái cho phép:

- Chọn mô hình nhúng (Embeddings) và mô hình AI.
- Cấu hình nguồn dữ liệu từ file hoặc URL.
- Nhập tên collection trong cơ sở dữ liệu vector Milvus để truy vấn.

Giao diện chính bao gồm:

- Một hộp nhập liệu để người dùng gửi câu hỏi.
- Hiển thị lịch sử câu các câu hôi thoại với chatbot.
- Kết nối với mô hình AI để xử lý câu hỏi.

Hệ thống tích hợp với các mô hình AI như GPT-4 (có giới hạn truy vấn và cần kết nối mạng) hay Llama 3 thông qua các retriever và executor được xây dựng sẵn. Việc chọn

mô hình và kết nối dựa trên các tùy chọn mà người dùng đã cấu hình trên giao diện. Khi người dùng nhập câu hỏi:

- Hệ thống hiển thị câu hỏi trên giao diện.
- Gửi câu hỏi đến backend để xử lý.
- Câu hỏi được tiến hành nhúng và được đưa vào tìm kiếm tương tự với vector store.
- Mô hình ngôn ngữ lớn xử lý dữ liệu vector tương tự và tiến hành sinh câu trả lời.
- Trả câu trả lời về trên khung chat.

Ứng dụng giao diện được phát triển bằng Streamlit không chỉ dễ triển khai mà còn trực quan, thân thiện với người dùng. Hệ thống này đáp ứng tốt nhu cầu tư vấn tuyển sinh với khả năng tùy chỉnh cao, phản hồi nhanh cho dù chạy trên máy tính cá nhân với cấu hình hạn chế.

3.8 Đánh giá hiệu quả chatbot

Mục tiêu của phần đánh giá là kiểm tra khả năng trả lời chính xác và hiệu quả của chatbot trong việc trả lời hỗ trợ tư vấn. Tiến hành đánh giá qua phương pháp : đánh giá này dựa trên phản hồi thực tế từ con người một cách chủ quan.

Bộ câu hỏi bao gồm 50 câu thuộc các nhóm sau:

- Câu hỏi thông tin chung: Ví dụ về thông tin trường, thông tin ngành học.
- Câu hỏi chi tiết: Yêu cầu thông tin đào tạo về từng ngành, điều kiện xét tuyển, cơ hội việc làm.
- Câu hỏi phức tạp: câu hỏi về cách đăng ký thủ tục với những điều kiện chi tiết.

Mỗi câu trả lời từ chatbot được đánh giá dựa trên các tiêu chí:

- Độ chính xác: Nội dung trả lời đúng hay không.
- Mức độ phù hợp: Câu trả lời có đáp ứng được nhu cầu của người dùng không.
- Tính dễ hiểu: Trả lời có rõ ràng, mạch lạc không.
- Cho điểm thủ công từng tiêu chí theo thang điểm từ 1 (rất kém) đến 10 (rất tốt).

Bảng 3.1 Kết quả đánh giá hiệu quả Chatbot thủ công

Tiêu chí	**Điểm đánh giá (trên thang 10)**	**Nhận xét**
Độ chính xác	9	Chatbot trả lời chính xác hầu hết các câu hỏi thông tin, ngoại trừ vài trường hợp quá chi tiết.
Mức độ phù hợp	8	Trả lời đúng trọng tâm câu hỏi
Tính mạch lạc	6	Tả lời dễ hiểu và tự nhiên, nhưng ngắt câu chưa đúng.

KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

Kết luận

Trong quá trình thực hiện đề tài "Xây dựng chatbot hỗ trợ tuyển sinh Trường Đại học Bách khoa Đà Nẵng", em đã giải quyết thành công bài toán xây dựng một hệ thống hỗ trợ tuyển sinh thông minh, tận dụng các tiến bộ trong công nghệ xử lý ngôn ngữ tự nhiên và học máy. Với tầm quan trọng ngày càng tăng của chuyển đổi số trong giáo dục, việc triển khai chatbot không chỉ giúp tự động hóa quy trình tư vấn tuyển sinh mà còn nâng cao trải nghiệm của các bạn thí sinh với mong muốn theo học tại Trường thông qua phản hồi nhanh chóng và chính xác.

Quan trọng hơn, đồ án đã hoàn thành mục tiêu cốt lõi: xây dựng một chatbot không chỉ cung cấp thông tin tuyển sinh mà còn giải đáp các câu hỏi liên quan, hỗ trợ xử lý các vấn đề ngoài phạm vi thông tin tuyển sinh, và quản lý dữ liệu người dùng. Những kết quả đạt được đã chứng minh tính hiệu quả và tiềm năng ứng dụng rộng rãi của hệ thống, góp phần nâng cao chất lượng dịch vụ tư vấn tuyển sinh của nhà trường.

Những kết quả đã đạt được

Với sự triển khai toàn diện, đồ án đã đạt được nhiều kết quả đáng chú ý. Trên phương diện lý thuyết, đồ án đã nghiên cứu và ứng dụng các công nghệ tiên tiến như mô hình Transformer, LLM, và kỹ thuật Prompt Engineering, đồng thời xây dựng nền tảng lý thuyết vững chắc về xử lý ngôn ngữ tự nhiên và học máy. Những nền tảng này không chỉ hỗ trợ việc phát triển hệ thống chatbot mà còn tạo cơ sở tham khảo hữu ích cho các nghiên cứu trong tương lai.

Về mặt thực tiễn, chatbot đã được thiết kế và triển khai với nhiều tính năng quan trọng. Hệ thống sử dụng cơ sở dữ liệu vector để lưu trữ và truy xuất thông tin hiệu quả, đồng thời tích hợp các thuật toán tiên tiến nhằm nâng cao độ chính xác của kết quả phản hồi.

Phương hướng phát triển

Mặc dù đã đạt được những kết quả khả quan, hệ thống chatbot vẫn còn nhiều tiềm năng để phát triển, nhằm nâng cao hiệu quả và mở rộng phạm vi ứng dụng. Một định hướng quan trọng là mở rộng khả năng hỗ trợ sang các lĩnh vực hoặc bộ phận khác trong nhà trường, chẳng hạn như tư vấn học thuật, hỗ trợ thủ tục hành chính, hoặc cung cấp thông tin chi tiết về từng ngành học và phòng ban. Điều này sẽ giúp chatbot trở thành một công cụ đa năng, phục vụ hiệu quả cho cả sinh viên lẫn cán bộ nhà trường.

Hệ thống cần được tối ưu hóa hiệu suất để đảm bảo khả năng phục vụ đồng thời nhiều người dùng. Việc áp dụng các thuật toán truy vấn và xếp hạng tiên tiến hơn có thể nâng cao tốc độ phản hồi và độ chính xác của kết quả. Bên cạnh đó, tích hợp chatbot vào các nền tảng phổ biến như mạng xã hội, ứng dụng di động hoặc hệ thống nội bộ sẽ mở rộng khả năng tiếp cận, mang lại trải nghiệm đồng nhất và thuận tiện cho người dùng trên nhiều loại thiết bị.

Bên cạnh đó, để đảm bảo hệ thống vận hành an toàn và bền vững, việc nghiên cứu và áp dụng các biện pháp bảo mật dữ liệu cá nhân là rất cần thiết, đặc biệt khi triển khai trên các môi trường công cộng. Những định hướng này không chỉ giúp tối ưu hóa hệ thống hiện tại mà còn mở ra cơ hội mở rộng nghiên cứu và ứng dụng vào các lĩnh vực khác, đóng góp vào sự phát triển toàn diện của nhà trường.

TÀI LIỆU THAM KHẢO

[1] S. C. Mana, G. Kalaiarasi, Y. R, L. S. Helen, and R. Senthamil Selvi, "Application of Machine Learning in Healthcare: An Analysis," in *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2022, pp. 1611–1615. doi: 10.1109/ICESC54411.2022.9885296.

[2] S. Das, A. dey, A. Pal, and N. Roy, "Applications of Artificial Intelligence in Machine Learning: Review and Prospect," *Int J Comput Appl*, vol. 115, pp. 31–41, Nov. 2015, doi: 10.5120/20182-2402.

[3] M. Abedi, I. Alshybani, M. R. B. Shahadat, and M. Murillo, "Beyond Traditional Teaching: The Potential of Large Language Models and Chatbots in Graduate Engineering Education," Nov. 2023. doi: 10.48550/arXiv.2309.13059.

[4] B. M. Gurusamy *et al.*, "An analysis of large language models: their impact and potential applications," *Knowl Inf Syst*, vol. 66, pp. 1–24, Nov. 2024, doi: 10.1007/s10115-024-02120-8.

[5] S. Brightwood, "Implementing AI Chatbots for Real-Time Supply Chain Monitoring and Risk Management," Nov. 2024.

[6] Y. Zhang, S. Kumar, D. Paikar, H. Ali, and K. Vutukuri, "KatzBot: Revolutionizing Academic Chatbot for Enhanced Communication," Nov. 2024. doi: 10.13140/RG.2.2.12847.96162.

[7] M. J. El Naqa Issam and Murphy, "What Is Machine Learning?," in *Machine Learning in Radiation Oncology: Theory and Applications*, R. and M. M. J. El Naqa Issam and Li, Ed., Cham: Springer International Publishing, 2015, pp. 3–11. doi: 10.1007/978-3-319-18305-3_1.

[8] V. Nasteski, "An overview of the supervised machine learning methods," *HORIZONS.B*, vol. 4, pp. 51–62, Nov. 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05.

[9] M. Mohri, "Foundations of machine learning," 2018, *MIT press*.

[10] Wikipedia, "Supervised learning," p. How supervised learning algorithms work, Nov. 2024.

[11] Z. Ghahramani, "Unsupervised Learning," in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, U. and R. G. Bousquet Olivier and von Luxburg, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 72–112. doi: 10.1007/978-3-540-28650-9_5.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[13] Amazon, "Điểm khác biệt giữa máy học và học sâu là gì?," 2024, *https://aws.amazon.com/vi/compare/the-difference-between-machine-learning-and-deep-learning/*.

[14] S. Shanmuganathan, "Artificial Neural Network Modelling: An Introduction," in *Artificial Neural Network Modelling*, S. Shanmuganathan Subana and Samarasinghe, Ed., Cham: Springer International Publishing, 2016, pp. 1–14. doi: 10.1007/978-3-319-28495-8_1.

[15] B. M. Gurusamy *et al.*, "An analysis of large language models: their impact and potential applications," *Knowl Inf Syst*, vol. 66, pp. 1–24, Nov. 2024, doi: 10.1007/s10115-024-02120-8.

[26] H. Pothina and K. V Nagaraja, "Artificial Neural Network and Math Behind It," in

Smart Trends in Computing and Communications, Y.-D. Zhang, T. Senjyu, C. So-In, and A. Joshi, Eds., Singapore: Springer Nature Singapore, 2023, pp. 205–221.
[17]S. Haykin, *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.
[18]R. Raj, "Components of an Artificial Neural Network," 2024.
[19]A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A Review of Activation Function for Artificial Neural Network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2020, pp. 281–286. doi: 10.1109/SAMI48414.2020.9108717.
[20]Wikipedia, "Activation function," p. https://en.wikipedia.org/wiki/Activation_function, Nov. 2024.
[21]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
[22]T. Babs, "The Mathematics of Neural Networks," 2018.
[23]L. F. Guilhoto, "An overview of artificial neural networks for mathematicians," *Univ. Chicago*, 2018.
[24]J. Schmidhuber, "A 'self-referential'weight matrix," in *ICANN'93: Proceedings of the International Conference on Artificial Neural Networks Amsterdam, The Netherlands 13–16 September 1993 3*, 1993, pp. 446–450.
[25]S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4–5, pp. 185–196, 1993.
[26]J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans Neural Netw*, vol. 6, no. 2, pp. 296–317, 1995.
[27]L. F. Guilhoto, "An overview of artificial neural networks for mathematicians," *Univ. Chicago*, 2018.
[28]J. Schmidhuber, "A 'self-referential'weight matrix," in *ICANN'93: Proceedings of the International Conference on Artificial Neural Networks Amsterdam, The Netherlands 13–16 September 1993 3*, 1993, pp. 446–450.
[29]S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4–5, pp. 185–196, 1993.
[30]J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans Neural Netw*, vol. 6, no. 2, pp. 296–317, 1995.
[31]indrasingh52, "Layers in Artificial Neural Networks (ANN)," 2024.
[32]anshumanm2fja, "What is Forward Propagation in Neural Networks?," 2024.
[33]Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[34]T. Beysolow II, "What Is Natural Language Processing?," in *Applied Natural Language Processing with Python : Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing* , Berkeley, CA: Apress, 2018, pp. 1–12. doi: 10.1007/978-1-4842-3733-5_1.
[35]J. H. Cole Stryker, "What is NLP?," 2024.
[36]A. Vaswani, "Attention is all you need," *Adv Neural Inf Process Syst*, 2017.
[37]4technews.net, "Ứng dụng xử lý ngôn ngữ tự nhiên trong kinh doanh," 2024.

[38]J. Alammar, "The Illustrated Transformer," 2024.
[39]T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," *CoRR*, vol. abs/1904.09675, 2019, [Online]. Available: http://arxiv.org/abs/1904.09675
[40]M. Sharma, "What is Add & Norm, as quick as possible?," 2024.
[41]S. Targ, D. Almeida, and K. Lyman, "Resnet in Resnet: Generalizing Residual Architectures," *CoRR*, vol. abs/1603.08029, 2016, [Online]. Available: http://arxiv.org/abs/1603.08029
42] N. T. H. Trần Hồng Việt, "TRANSFORMERS MODEL AND APPLY IN NATURAL LANGUAGE PROCESSING," *Tài liệu NCKH GOV*, 2017.
[43]Cloudflare, "Mô hình ngôn ngữ lớn (LLM) là gì?," 2024.
[44]R. Dilts, R. Bandler, J. Grinder, and J. DeLozier, "What is NLP," *Режим доступу: http://www. nlpu. com/NLPU_WhatIsNLP. html*, 1999.
[45]M. Gothankar, "What is LLM & How to Build Your Own Large Language Models?," 2024.
[46]J. C. Luna, "What is an LLM? A Guide on Large Language Models and How They Work," 2024.
[47]Nithyashree, "What is Chunking in Natural Language processing?," 2021.
[48]D. Raikar, "How Chunking Strategies Work: Paragraph, Sentence and Smart Techniques," 2024.
[49]T. ML, "Embedding," 2024.
[50]P. Phan, "ChatGPT Series 5: Tìm hiểu về Retrieval Augmented Generation (RAG)," 2023.
[51]S. Ahmed, "What is Retrieval-Augmented Generation(RAG) in LLM and How it works?," 2024.
[52]D. Shah, "Reciprocal Rank Fusion (RRF) explained in 4 mins — How to score results form multiple retrieval methods in RAG," 2024.

[53]P. Phan, "ChatGPT Series 5: Tìm hiểu về Retrieval Augmented Generation (RAG)," 2023.
[54]S. Ahmed, "What is Retrieval-Augmented Generation(RAG) in LLM and How it works?," 2024.
[55]D. Shah, "Reciprocal Rank Fusion (RRF) explained in 4 mins — How to score results form multiple retrieval methods in RAG," 2024.
[56]Z. Rackauckas, "RAG-Fusion: a New Take on Retrieval-Augmented Generation," 2024. doi: https://doi.org/10.5121/ijnlc.2024.13103.
[57]T. Q. Huy, "LLM agent: Tổng quan về LLM agent," 2024.
[58]T. Palaniyappan, "How LLM Agent works?," 2024.
[59]L. Weng, "LLM Powered Autonomous Agents," 2023.
[60]R. Graph, "Prompt Engineering," 2024.
[61]L. Phan, "Tất tần tật những kĩ thuật Prompt Engineering hữu ích nhất cho chatGPT," 2024.
[62]S. Ahmed, "LLM Prompt Engineering for Beginners: What It Is and How to Get Started," 2024.
[63]R. Khawaja, "Demystifying Embeddings 101: The Foundation of Large Language Models," 2023.
[64]novita.ai, "What is LLM Embeddings: All You Need To Know," 2024.
[65]Sbert, "Pretrained Models: Sentence Transformers," 2024.

[66]Deepinfra, "sentence-transformers/all-MiniLM-L6-v2," 2024.
[67]Sbert, "Pretrained Models: Multilingual Models," 2024.
[68]H. Face, "Alibaba-NLP/gte-multilingual-base," 2024.
[69]huggingface, "BAAI/bge-m3," 2024. [Online]. Available: https://huggingface.co/BAAI/bge-m3
[70]J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation," 2024. [Online]. Available: https://arxiv.org/abs/2402.03216
[71]Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, "Nomic Embed: Training a Reproducible Long Context Text Embedder," 2024.
[72]A. Trivedi, "Top 10 Open-Source LLMs for 2025 and Their Uses," 2024.

--HẾT--