HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHAN QUANG THÀNH

NGHIÊN CỨU XÂY DỰNG CHATBOT TƯ VẤN, HỖ TRỢ NHẬP HỌC TẠI HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI – 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHAN QUANG THÀNH

NGHIÊN CỨU XÂY DỰNG CHATBOT TƯ VẤN, HỖ TRỢ NHẬP HỌC TẠI HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC : TS. TRẦN TIẾN CÔNG

HÀ NỘI – 2024

LỜI CAM ĐOAN

Học viên *Phan Quang Thành,* mã học viên *B22CHIS005* xin cam đoan đề án tốt nghiệp là công trình nghiên cứu của riêng học viên dưới sự hướng dẫn của *TS. Trần Tiến Công*. Tất cả những tham khảo trong đề án tốt nghiệp bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo.

Hà Nội, ngày 28 tháng 02 năm 2024

Tác giả đề án tốt nghiệp ký và ghi rõ họ tên

Phan Quang Thành - B22CHIS005

LỜI CẢM ƠN

Em xin chân thành cảm ơn giảng viên hướng dẫn TS. Trần Tiến Công đã giúp đỡ và định hướng cho em trong suốt quá trình học tập và thực hiện đề án tốt nghiệp.

Dưới sự hướng dẫn của TS. Trần Tiến Công, em đã cố gắng hoàn thành tốt nhất có thể đề án tốt nghiệp này, tuy nhiên trong quá trình thực hiện không thể tránh được những thiếu sót, em rất mong nhận được sự góp ý của các thầy/cô trong hội đồng để em hoàn thiện hơn đề án tốt nghiệp này.

Em xin chân thành cảm ơn!

Hà Nội, ngày 28 tháng 02 năm 2024

Học viên thực hiện

Phan Quang Thành - B22CHIS005

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
IC	Intent Classification	Phân loại ý định
NLU	Natural Language Understanding	Hiểu ngôn ngữ tự nhiên
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
CNN	Convolutional Neural Network	Mạng nơ ron tích chập
RNN	Recurrent neural network	Mạng thần kinh hồi quy
LSTM	Long Short Term Memory networks	Mạng bộ nhớ dài-ngắn
Transformer	Transformer Neural Network	Mạng nơ ron biến áp
TODs	Task-oriented dialogue systems	Hệ thống đối thoại định hướng nhiệm vụ
IPAs	Intelligent Personal Assistants	Trợ lý cá nhân thông minh

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

Trong những năm gần đây, trí tuệ nhân tạo chung và chatbot nói riêng là một trong những công nghệ phát triển mạnh mẽ và có nhiều ứng dụng thực tiễn trong cuộc sống của chúng ta. Từ việc có thể cung cấp dịch vụ, giải đáp thắc mắc, xử lý yêu cầu trong lĩnh vực chăm sóc khách hàng, đến việc có thể mang lại thông tin y tế, hỗ trợ chẩn đoán, điều trị bệnh trong lĩnh vực y tế... Theo khảo sát cua Salesforce, khoảng 23% công ty dịch vụ khách hàng hiên nay đang tích hợp chatbot AI trong hoạt động vận hành của doanh nghiệp. Một khảo sát khác của TIDIO cho thấy 62% doanh nghiệp trên toàn thế giới đang có kế hoạch ứng dụng chatbot trên các nền tảng website cũng như các trang mạng xã hội.

Trong lĩnh vực giáo dục, nhu cầu chatbot hiện nay đối với công tác tuyển sinh, nhập học tại các cơ sở giáo dục đang ngày càng tăng cao. Điều này là do những lợi ích mà chatbot mang lại như có thể tự động trả lời các câu hỏi thường gặp của thí sinh, hướng dẫn thủ tục hồ sơ, giấy tờ cần chuẩn bị, tư vấn lựa chọn chương trình học, tăng cường tương tác với thí sinh bằng cách chatbot có thể giao tiếp với thí sinh 24/7.

Do đó học viên thực hiện đề tài "Nghiên cứu và xây dựng chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông" nhằm tập trung nghiên cứu và xây dựng hệ thống chatbot góp phần hỗ trợ, nâng cao chất lượng, tiết kiệm thời gian và chi phí trong công tác tuyển sinh, nhập học tại Học viện.

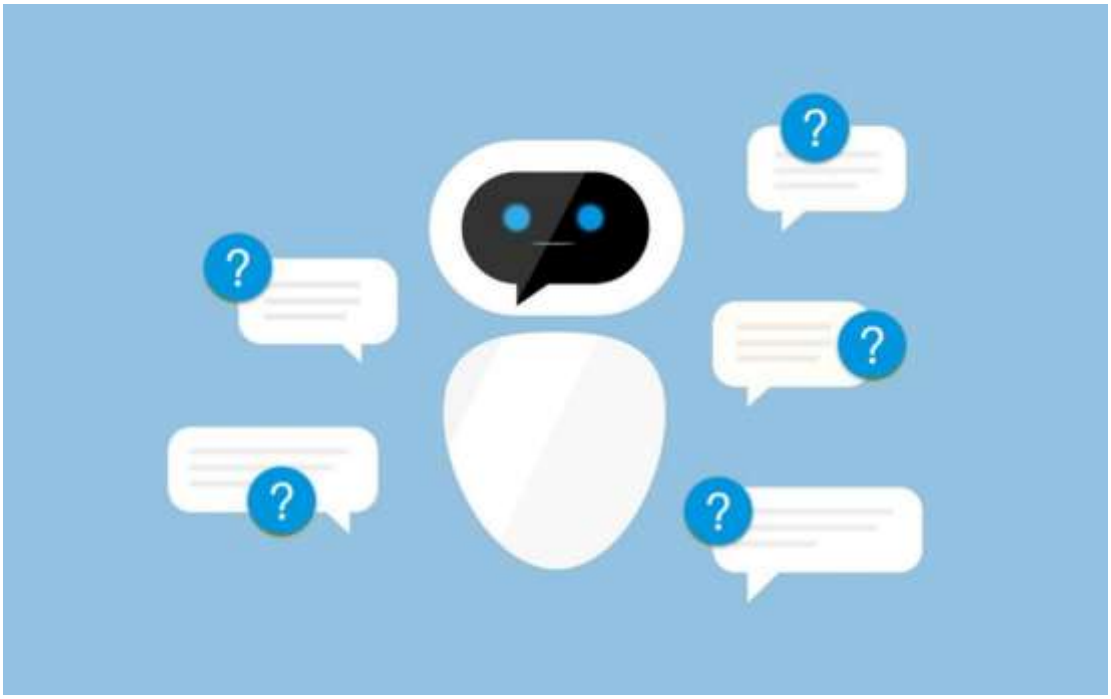
CHƯƠNG 1. TỔNG QUAN VỀ CHATBOT

Chương 1 trình bày tổng quan về hệ thống chatbot, đặc trưng của hệ thống chatbot khi ứng dụng trong lĩnh vực giáo dục. Nội dung chương 1 tập trung tìm hiểu tổng quan về hệ thống chatbot và đặc trưng của công tác tư vấn, hỗ trợ nhập học, từ đó chỉ ra những vấn đề cần giải quyết, khắc phục và đề xuất phương hướng giải quyết bài toán ứng dụng chatbot vào tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu Chính Viễn thông.

1.1 Khái niệm chatbot

Chatbot là một dạng ứng dụng phần mềm nhằm mô phỏng giao tiếp với con người trong việc thực hiện một loạt các tác vụ hoặc giải đáp thắc mắc. Chatbot có thể được lập trình sử dụng trí tuệ nhân tạo (AI) để hiểu và phản hồi một cách linh hoạt hơn, trong khi một số khác có thể hoạt động dựa trên một bộ quy tắc cố định.

Có nhiều dạng Chatbot khác nhau, từ những Chatbot đơn giản chỉ có khả năng phản hồi các câu hỏi đã được lập trình sẵn, đến những Chatbot phức tạp sử dụng trí tuệ nhân tạo (AI), học máy (ML) và xử lý ngôn ngữ tự nhiên (NLP) để tiếp nhận thông tin, phân tích câu hỏi và phản hồi chính xác những gì mà người dùng mong muốn, ngoài ra Chatbot còn có khả năng tự học hỏi từ chính những cuộc trò chuyện với người dùng để đưa ra câu trả lời ngoài phân vùng dữ liệu được lặp đi lặp lại nhiều lần.



Hình 1.1: Minh họa chatbot

1.2 Đặc trưng của công tác tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông

Học viện Công nghệ Bưu chính Viễn thông (PTIT) là một trong những trường đại học công lập hàng đầu tại Việt Nam trong lĩnh vực bưu chính, viễn thông và công nghệ thông tin. Theo số liệu năm 2023, Học viện tuyển sinh đào tạo 19 ngành với tổng chỉ tiêu 4345 sinh viên và dự kiến sẽ tăng lên thành 22 ngành với tổng chỉ tiêu 5200 chỉ tiêu trong năm 2024. Bên cạnh đó Học viện cũng có các chương trình đào tạo chất lượng cao, chương trình liên kết quốc tế với nội dung, chất lượng ưu viết và lợi thế đối với người học. Vì vậy công tác tư vấn, hỗ trợ nhập học tại Học viện có một số điểm đặc trưng sau:

- Tổ chức nhập học cho tân sinh viên thường diễn ra trong thời gian ngắn (1-2 ngày).
- Cung cấp các bộ tài liệu như sách hướng dẫn nhập học, đến các video giới thiệu chuyên ngành và cơ sở vật chất.
- Tổ chức các buổi định hướng, tour tham quan trường, và buổi giới thiệu, giao lưu với các câu lạc bộ, tổ chức sinh viên để giới thiệu văn

hóa và mạng lưới hỗ trợ tại trường.

- Cung cấp thông tin chi tiết về các chương trình học, cơ hội nghề nghiệp sau khi tốt nghiệp và các mối quan hệ với doanh nghiệp, giúp sinh viên hiểu rõ và chọn lựa chương trình học phù hợp với nguyện vọng và khả năng của bản thân.
- Thông tin về học bổng, các chi phí liên quan trong quá trình học tập và hỗ trợ sinh viên trong việc xin học bổng hay chế độ chính sách cho sinh viên.

Với số lượng tân sinh viên nhập học rất lớn trong thời gian ngắn, cùng với việc phải cung cấp nhiều thông tin tới tân sinh viên do đó công tác tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông khó có thể tránh khỏi các sai sót, trong khi vẫn cần huy động số lượng lớn nhân lực tham gia, tốn kém cả về công sức và thời gian.

1.3 Mô tả bài toán

Tổ chức nhập học cho tân sinh viên là một trong những bước quan trọng nhất trong công tác tuyển sinh hàng năm tại Học viện Công nghệ Bưu chính Viễn thông. Tuy nhiên, quá trình này thường gặp nhiều khó khăn và thách thức từ việc kiểm tra các loại giấy tờ, kinh phí nhập học của tân sinh viên tới việc cung cấp thông tin, tư vấn về chương trình học, giới thiệu về Học viện cũng như các câu lạc bộ, tổ chức sinh viên đang hoạt động....

Với số lượng tân sinh viên nhập học đang ngày càng tăng qua các năm, việc xây dựng chatbot tư vấn, hỗ trợ công tác nhập học giúp giảm thiểu sai sót, tiết kiệm công sức và thời gian là việc cấp thiết.

Dưới đây là một số yêu cầu cụ thể của chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông:

- Trả lời các câu hỏi thường gặp của thí sinh: Chatbot có thể được sử dụng để trả lời các câu hỏi thường gặp của thí sinh về các loại giấy tờ, kinh phí cần chuẩn bị để nhập học.
- Hỗ trợ đăng ký chương trình chất lượng cao, liên kết quốc tế, tiếng Anh:

Chatbot có thể được sử dụng để hỗ trợ thí sinh đăng ký thi tiếng Anh, đăng ký học chương trình chất lượng cao, bao gồm nhập thông tin, nộp hồ sơ,...

- Tư vấn các hoạt động khác: Chatbot có thể cung cấp thông tin về các câu lạc bộ, hỗ trợ sinh viên ghi danh tham gia câu lạc bộ mong muốn, cung cấp thông tin về nhà cho thuê, quán ăn, cửa hàng tiện lợi... quanh khu vực Học viện.
- Hỗ trợ chế độ chính sách sinh viên: Chatbot có thể cung cấp thông tin về các loại học bổng, chế độ miễn giảm học phí, ký túc xá cho sinh viên...

Với hướng nghiên cứu và xây dựng chatbot tư vấn, hỗ trợ tuyển sinh, nhập học một số cơ sở giáo dục tại Việt Nam đã triển khai thành công như:

- Trường Đại học Kinh tế Quốc dân: NEU-Chatbot là chatbot tuyển sinh của Trường Đại học Kinh tế Quốc dân, có thể trả lời hơn 50 loại câu hỏi với độ chính xác lên đến hơn 95%.
- Đại học FPT: FPT Chatbot là chatbot tuyển sinh của Đại học FPT, có thể hỗtrợ thí sinh đăng ký tuyển sinh, tư vấn tuyển sinh,...
- Đại học Bách khoa Hà Nội: Bách khoa Chatbot là chatbot tuyển sinh của Đại học Bách khoa Hà Nội, có thể trả lời các câu hỏi về quy chế tuyển sinh, phương thức tuyển sinh,...

Trong tương lai, chatbot sẽ tiếp tục được ứng dụng rộng rãi hơn nữa trong công tác tuyển sinh nói chung và công tác hỗ trợ nhập học nói riêng, giúp các cơ sở giáo dục nâng cao hiệu quả công tác tuyển sinh.

1.4 Kết luận

Chương này trình bày tổng quan về hệ thống chatbot, các đặc trưng công tác tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông. Từ đó, nội dung chương chỉ ra những vấn đề còn tồn tại trong công tác tư vấn, hỗ trợ nhập học và đề xuất phương án giải quyết các vấn đề này. Nội dung chi tiết phương án giải quyết vấn đề được trình bày tại Chương 2 của đề án tốt nghiệp.

CHƯƠNG 2. NGHIÊN CỨU CÁC CÔNG NGHỆ, KỸ THUẬT XÂY DỰNG CHATBOT PHỔ BIẾN HIỆN NAY

Chương 2 trình bày nghiên cứu các công nghệ, kỹ thuật xây dựng chatbot phổ biến hiện nay trên thế giới. Từ đó đưa ra công nghệ, kỹ thuật được sử dụng để xây dựng chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông.

2.1 Kiến trúc, thành phần cơ bản của chatbot và những loại chatbot phổ biến hiện nay

2.1.1 Kiến trúc

Có ba kiến trúc chatbot phổ biến hiện nay:

- Kiến trúc dựa trên quy tắc: Kiến trúc dựa trên quy tắc sử dụng một bộ quy tắc để xác định cách chatbot sẽ phản hồi các câu hỏi và yêu cầu của người dùng. Bộ quytắc này được xây dựng thủ công bởi các nhà phát triển chatbot.
- Kiến trúc dựa trên tri thức: Kiến trúc dựa trên tri thức sử dụng một cơ sở tri thức để xác định cách chatbot sẽ phản hồi các câu hỏi và yêu cầu của người dùng. Cơ sở tri thức này chứa thông tin về thế giới thực và cách chatbot có thể sử dụng thông tin này để trả lời các câu hỏi của người dùng.
- Kiến trúc học máy: Kiến trúc học máy sử dụng các mô hình học máy để đào tạo chatbot. Các mô hình này được đào tạo trên một tập dữ liệu lớn bao gồm các cuộc trò chuyện giữa người dùng và chatbot.

2.1.2 Thành phần cơ bản

2.1.2.1 Xử lý ngôn ngữ tự nhiên

NLP là công nghệ giúp chatbot hiểu và tạo ra ngôn ngữ của con người. Các kỹ thuật NLP thường được sử dụng trong chatbot bao gồm:

- Phân tích cú pháp: Phân tích cú pháp là quá trình xác định cấu trúc của câu. Chatbot sử dụng phân tích cú pháp để hiểu ý nghĩa của câu hỏi và yêu cầu của người dùng.
- Tìm kiếm: Tìm kiếm là quá trình tìm kiếm thông tin liên quan đến câu hỏi

hoặc yêu cầu của người dùng. Chatbot sử dụng tìm kiếm để tìm các câu trả lời phù hợp cho người dùng.

- Sinh văn bản: Sinh văn bản là quá trình tạo ra văn bản mới. Chatbot sử dụng sinh văn bản để tạo ra các câu trả lời cho người dùng.
- Dịch ngôn ngữ: Chatbot sử dụng dịch ngôn ngữ để giao tiếp với người dùng nói ngôn ngữ khác.
- Chuẩn hóa: Kiểm tra các lỗi chính tả làm thay đổi ngữ nghĩa trong yêu cầu của người dùng.
- Nhận dạng thực thể: Chatbot tìm kiếm các loại thông tin cần thiết khác nhau như vị trí, con người, đồ vật...

2.1.2.2 Dữ liệu xây dựng chatbot

Chatbot được đào tạo trên một tập dữ liệu lớn bao gồm các cuộc trò chuyện giữa người dùng và chatbot. Dữ liệu này giúp chatbot học cách hiểu và phản hồi các câu hỏi và yêu cầu của người dùng.

Dữ liệu là yếu tố quan trọng nhất trong việc xây dựng chatbot. Dữ liệu tốt sẽ giúp chatbot hiểu được ngôn ngữ tự nhiên và tạo ra các phản hồi phù hợp với ngữ cảnh của cuộc trò chuyện.

Có hai loại dữ liệu chính được sử dụng để xây dựng chatbot:

- Dữ liệu văn bản: Dữ liệu văn bản bao gồm các văn bản từ sách, bài báo, trang web, mạng xã hội,... Dữ liệu văn bản giúp chatbot học cách sử dụng ngôn ngữ tự nhiên một cách chính xác.
- Dữ liệu hội thoại: Dữ liệu hội thoại bao gồm các cuộc trò chuyện giữa con người với nhau. Dữ liệu hội thoại giúp chatbot hiểu được cách con người giao tiếp và tạo ra các phản hồi phù hợp với ngữ cảnh của cuộc trò chuyện.

2.1.2.3 Giao diện người dùng

Công cụ giúp người dùng giao tiếp, cấu hình chatbot. Giao diện người dùng có thể là một ứng dụng web, ứng dụng di động hoặc chatbot tích hợp vào trang web hoặc ứng dụng hiện có.

Ngoài các thành phần cơ bản này, chatbot có thể bao gồm các thành phần khác như:

- Công nghệ học máy: Công nghệ học máy giúp chatbot cải thiện hiệu suất theo thời gian.
- Công nghệ xử lý ngôn ngữ tự nhiên nâng cao: Công nghệ xử lý ngôn ngữ tự nhiên nâng cao giúp chatbot hiểu và tạo ra ngôn ngữ của con người một cách tự nhiên hơn.
- Công nghệ nhận dạng giọng nói: Công nghệ nhận dạng giọng nói giúp chatbot tương tác với người dùng bằng giọng nói.
- Công nghệ trí tuệ nhân tạo tổng quát: Công nghệ trí tuệ nhân tạo tổng quát giúp chatbot trở nên thông minh và linh hoạt hơn.

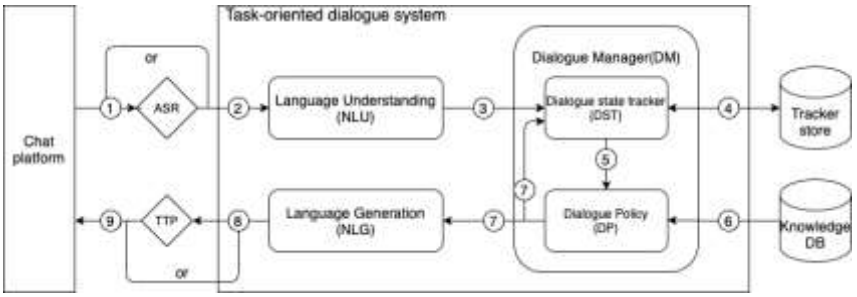
2.1.3 Phân loại chatbot phổ biến hiện nay

2.1.3.1 Hệ thống đối thoại đính hướng nhiệm vụ (TODs)

TODs là loại Chatbot được thiết kế để phục vụ một lĩnh vực nhất định như: mua sắm, cung cấp thông tin về thị trường chứng khoán...Một số thuật ngữ quan trọng sử dụng trong TODs:

- Ý đính: Ý định này được chuyển tải bởi người dùng tới TODs . Ví dụ, "Thời tiết hôm nay thế nào", thì ý định của người dùng ở đây là mong muốn nhận thông tin về thời tiết ngày hôm nay.
- Thực thể: các thông tin có thể trích xuất ra được từ câu nói của người dùng (ví dụ: họ tên, tuổi, địa chỉ...).
- Hành động: là hành động mà Chatbot có thể làm, thể hiện khả năng của Chatbot (ví dụ: tìm đường, tìm công thức nấu ăn...)
- Theo dõi lưu trữ: là CSDL để lưu trữ lịch sử trò chuyện của Chatbot với người dùng. Nó cũng có thể được lưu trong RAM hoặc gộp chung với Cơ sở dữ liệu tri thức.
- Cơ sở dữ liệu tri thức: là cơ sở dữ liệu lưu trữ dữ liệu tạo nên tri thức của Chatbot. Ví dụ như: thông tin về các công viên, đường phố...

Dưới đây là kiến trúc của một hệ thống TODS:



Hình 1.2: Kiến trúc hệ thống TODS

2.1.3.2 Trợ lý cá nhân thông minh (IPAs)

IPAs là loại Chatbot đóng vai trò là trợ lý cá nhân thông minh, là một loại phần mềm được thiết kế để giúp người dùng thực hiện các công việc hàng ngày thông qua việc xử lý ngôn ngữ tự nhiên. Các trợ lý thông minh này có khả năng học hỏi từ các tương tác của người dùng, từ đó cung cấp các dịch vụ cá nhân hóa như lập lịch trình, đặt nhắc nhở, tìm kiếm thông tin, điều khiển các thiết bị thông minh, và thậm chí thực hiện mua sắm trực tuyến. Ví dụ trợ lý cá nhân Siri của Apple có thể thực hiện các tác vụ như: thực hiện cuộc gọi, gửi tin nhắn, tìm quán ăn, quản lý lịch hẹn.

2.1.3.3 Hệ thống đối thoại chit-chat

Hệ thống đối thoại chit-chat là những hệ thống trò chuyện thông minh được thiết kế để mô phỏng cách con người chit-chat (trò chuyện phiếm) với nhau. Mục tiêu của hệ thống này không chỉ là cung cấp thông tin hay thực hiện một công việc cụ thể, mà còn là tạo ra một cuộc trò chuyện tự nhiên, thú vị và có khả năng duy trì quan tâm của người dùng.

Hệ thống đối thoại chit-chat thường sử dụng các kỹ thuật trong lĩnh vực trí tuệ nhân tạo, như máy học, học sâu, và xử lý ngôn ngữ tự nhiên để hiểu và phản hồi các đầu vào từ người dùng một cách thích ứng và linh hoạt. Chúng có thể lắng nghe, học hỏi từ ngữ cảnh và cảm xúc của người dùng để đưa ra các phản hồi phong phú và đa dạng, từ đó làm cho cuộc trò chuyện càng trở nên giống như giữa hai con người.

2.2 Các kỹ thuật sử dụng trong xây dựng chatbot

2.2.1 Mạng hồi quy RNN

Mạng nơ-ron hồi quy (RNN - Recurrent Neural Network) là một loại mạng nơ-ron nhân tạo nổi tiếng trong xử lý ngôn ngữ tự nhiên và nhận diện giọng nói. Khác với các mạng nơ-ron truyền thống, RNN có khả năng xử lý dữ liệu dạng chuỗi, nghĩa là nó có thể xử lý thông tin có thứ tự và liên kết với nhau, như văn bản, âm thanh hoặc chuỗi thời gian.

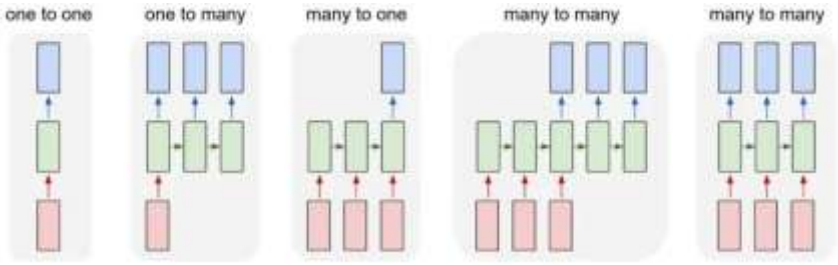
Deep learning có 2 mô hình lớn là Mạng nơ-ron tích chập (CNN) cho bài toán có input là ảnh và Mạng nơ-ron hồi quy (RNN) cho bài toán dữ liệu dạng chuỗi (sequence). Ý tưởng chính của RNN là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ cho bài toán cần phân loại phương tiện giao thông trong video, input là video 60s, output là phân loại phương tiện giao thông như xe đạp, xe máy, ô tô,... Khi xử lý video ta hay gặp khái niệm FPS (frame per second) tức là bao nhiêu frame (ảnh) mỗi giây. Ví dụ 1 FPS với video 60s tức là lấy ra từ video 60 ảnh, mỗi giây một ảnh để xử lý. Do các ảnh có thứ tự nên ta không thể sử dụng mạng CNN mà cần một mô hình mới có thể giải quyết được bài toán với input là sequence. Mạng nơ ron hồi quy (RNN) ra đời để giải quyết vấn đề đó.

2.2.1.1 Dữ liệu dạng sequence

Dữ liệu có thứ tự như các ảnh tách từ video ở trên được gọi là sequence, time-series data. Ví dụ khác là trong bài toán dịch tự động với input là 1 câu, ví dụ "tôi yêu Việt Nam" thì vị trí các từ và sự xắp xếp cực kì quan trọng đến nghĩa của câu và dữ liệu input các từ [’tôi’, ’yêu’, ’việt’, ’nam’] được gọi là sequence data. Trong bài toán xử lý ngôn ngữ (NLP) thì không thể xử lý cả câu được và người ta tách ra từng từ (chữ) làm input, giống như trong video người ta tách ra các ảnh (frame) làm input.

2.2.1.2 Phân loại bài toán RNN

Các mô hình RNN hầu như được sử dụng trong lĩnh vực xử lí ngôn ngữ tự nhiên và ghi nhận tiếng nói. Các ứng dụng khác được tổng kết trong hình dưới đây:



Hình 2.1: Các dạng bài toán RNN [2]

- One to one: mẫu bài toán cho Neural Network (NN) và Convolutional Neural Network (CNN), 1 input và 1 output, ví dụ với CNN input là ảnh và output làảnh được segment.
- One to many: bài toán có 1 input nhưng nhiều output, ví dụ: bài toán caption cho ảnh, input là 1 ảnh nhưng output là nhiều chữ mô tả cho ảnh đấy, dưới dạng một câu.
- Many to one: bài toán có nhiều input nhưng chỉ có 1 output, ví dụ bài toán phân loại hành động trong video, input là nhiều ảnh (frame) tách ra từ video,ouptut là hành động trong video
- Many to many: bài toán có nhiều input và nhiều output, ví dụ bài toán dịch từ tiếng Anh sang tiếng Việt, input là 1 câu gồm nhiều từ: “I drink water” và output cũng là 1 câu gồm nhiều từ “Tôi uống nước”.

2.2.1.3 Ứng dụng bài toán RNN

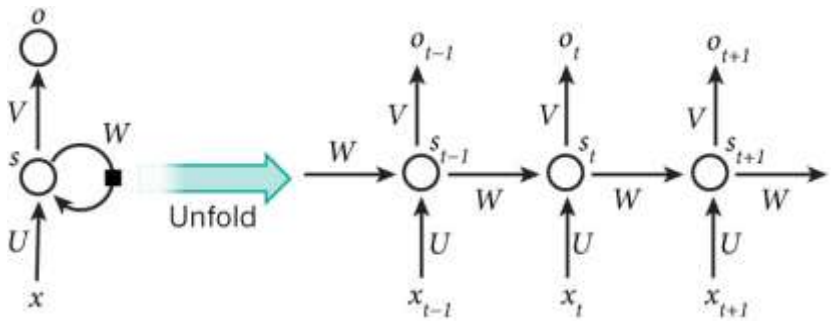
Về cơ bản nếu ta thấy sequence data hay time-series data và muốn áp dụng deep learning thì nghĩ ngay đến RNN. Dưới đây là một số ứng dụng của RNN:

- Speech to text: Chuyển giọng nói sang text.
- Sentiment classification: Phân loại bình luận của người dùng, tích cực hay tiêu cực.
- Video recognition: Nhận diện hành động trong video.

- Heart attack: Dự đoán đột quỵ tim.

2.2.1.4 Mô hình RNN

Về cơ bản một mạng RNN có dạng như sau:



Hình 2.2: Mô hình RNN [3]

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự. Ví dụ ta có một câu gồm năm chữ "Hôm nay trời đẹp quá", thì mạng nơ-ron được triển khai sẽ gồm năm tầng nơron tương ứng với mỗi chữ một tầng. Lúc đó việc tính toán bên trong RNN được thực hiện như sau:

- x_t: là đầu vào tại bước t. Ví dụ, x_1 là một vec-tơ one-hot tương ứng với từ thứ
- hai của câu.
- s_t: là trạng thái ẩn tại t. Nó chính là bộ nhớ của mạng. s_t được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó: $s_t = f(Ux_t + Ws_t - 1)$. Hàm f thường là một hàm phi tuyến tính như tang hyperbolic (tanh) hay ReLu. Để làm phép toán cho phần tử ẩn đầu tiên ta cần khởi tạo thêm s_{-1} , thường giá trị khởi tạo được gắn bằng 0.
- o_t: là đầu ra tại bước t. Ví dụ, muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì ot chính là một vectơ xác xuất các từ trong danh sách từ vựng: $o_t = \text{softmax}(Vs_t)$

Bên cạnh RNN truyền thống, RNN còn có 2 biến thể khác là RNN hai chiều (Bidirectional, BRNN) và RNN sâu (Deep, DRNN).

Mặc dù RNN có khá nhiều ưu điểm như khả năng xử lí đầu vào với bất kì độ

dài nào, kích cỡ mô hình không tăng theo kích cỡ đầu vào, quá trình tính toán sử dụngcác thông tin cũ, trọng số được chia sẻ trong suốt thời gian.

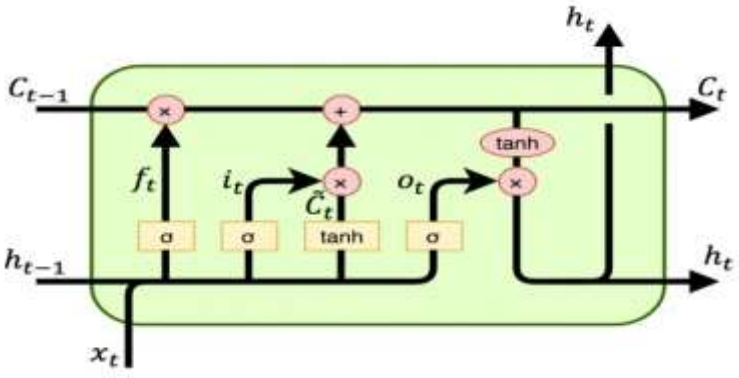
Tuy nhiên, RNN tính toán chậm, khó để truy cập các thông tin từ một khoảng thời gian dài trước đây khi đạo hàm quá nhiều dẫn đến hiện tượng g vanishing gradient ở các state đầu nên ta cần mô hình tốt hơn để giảm hiện tượng vaninshing gradient. Do đó, trong phần tiếp theo em sẽ trình bày về Long short term memory (LSTM) có thể đối phó với vấn đề vanishing gradient khi gặp phải bằng mạng RNNs truyền thống.

2.2.2 Mạng LSTM

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu lần đầu năm 1997. Trải qua nhiều lần cải tiến, chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã cóthể ghi nhớ được mà không cần bất kì can thiệp nào.

Dưới đây là mô hình mạng LSTM:



Hình 2.3: Mô hình LSTM [4]

Ở state thứ t của mô hình LSTM:

- Output: c_t, h_t, ta gọi c là cell state, h là hidden state.

- Input: c_{t-1}, h_{t-1}, x_t. Trong đó xt là input ở state thứ t của model. c_{t-1}, h_{t-1} là output của layer trước. h đóng vai trò khá giống như s ở RNN, trong khi c làđiểm mới của LSTM.

Kí hiệu σ, tanh lần lượt là sigma, tanh activation function. Phép nhân ở đây là element-wise multiplication, phép cộng là cộng ma trận.

f_t, i_t, o_t tương ứng với forget gate, input gate và output gate. Ta có:

- Forget gate: $f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)$
- Input gate: $i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i)$
- Output gate: $o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o)$

Tại c_t, forget gate quyết định xem cần lấy bao nhiêu từ cell state trước và input gate sẽ quyết định lấy bao nhiêu từ input của state và hidden layer của layer trước.

Tại h_t, output gate quyết định xem cần lấy bao nhiêu từ cell state để trở thành output của hidden state. Ngoài ra ht cũng được dùng để tính ra output yt cho state t.

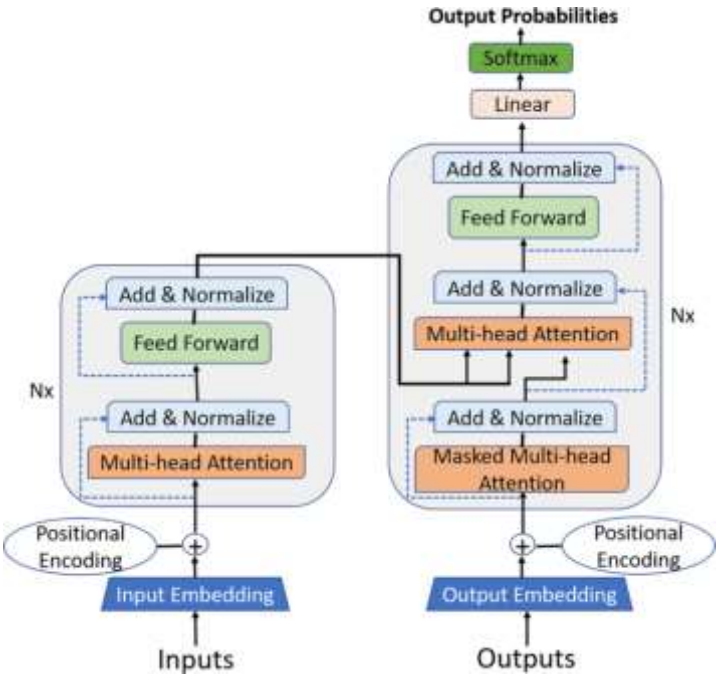
Như vậy, so với RNN thì LSTM là một bước lớn trong việc sử dụng RNN vì LSTM có cả có cả short term memory và long term memory giúp thông tin nào cần quan trọng và dùng ở sau sẽ được gửi vào và dùng khi cần và có thể mang thông tin từ đi xa.

2.2.3 Transformer

Một kiến trúc mới về Transformer hoàn toàn khác so với các kiến trúc RNN trước đây, mặc dù cả hai đều thuộc lớp model seq2seq nhằm chuyển 1 câu văn input ở ngôn ngữ A sang 1 câu văn output ở ngôn ngữ B. Quá trình biến đổi (transforming) được dựa trên 2phần encoder và decoder.

RNN là lớp model tốt trong dịch máy vì ghi nhận được sự phụ thuộc thời gian của các từ trong câu. Tuy nhiên các nghiên cứu mới đã chỉ ra rằng với chỉ với cơ chế attention mà không cần đến RNN đã có thể cải thiện được kết quả của các tác vụ dịch máy và nhiều tác vụ khác. Một trong những cải thiện đó là model BERT.

Kiến trúc của Transformer được thể hiện qua sơ đồ bên dưới:



Hình 2.4: Kiến trúc Transformer

Kiến trúc này gồm 2 phần encoder bên trái và decoder bên phải.

Encoder: là tổng hợp xếp chồng lên nhau của 6 layers xác định. Mỗi layer bao gồm 2 layer con (sub-layer) trong nó. Sub-layer đầu tiên là multi-head self- attention. Layer thứ 2 đơn thuần chỉ là các fully-connected feed-forward layer. Sử dụng một kết nối residual ở mỗi sub-layer ngay sau layer normalization. Kiến trúc này có ý tưởng tương tự như mạng resnet trong CNN. Đầu ra của mỗi sub-layer là $LayerNorm(x + Sublayer(x))$ có số chiều là 512 theo như bài viết.

Decoder: Decoder cũng là tổng hợp xếp chồng của 6 layers. Kiến trúc tương tự như các sub-layer ở Encoder ngoại trừ thêm 1 sub-layer thể hiện phân phối attention ở vị trí đầu tiên. Layer này không gì khác so với multi-head self-attention layer ngoại trừ được điều chỉnh để không đưa các từ trong tương lai vào attention. Tại bước thứ i của decoder chỉ biết được các từ ở vị trí nhỏ hơn nên việc điều chỉnh đảm bảo attention chỉ áp dụng cho những từ nhỏ hơn vị trí thứ i. Cơ chế residual cũng được áp dụng tương tự như trong Encoder.

Lưu ý luôn có một bước cộng thêm Positional Encoding vào các input của encoder và decoder nhằm đưa thêm yếu tố thời gian vào mô hình làm tăng độ chuẩn xác. Đây chỉ đơn thuần là phép cộng vector mã hóa vị trí của từ trong câu với vector biểu diễn từ. Có thể mã hóa dưới dạng [0, 1] vector vị trí hoặc sử dụng hàm *sin,cos* như trong bài báo.

2.2.3.1 Cơ chế Self Attention

Đầu vào của khối encoder đầu tiên là các vector embeddings của các từ trong câu. Đầu vào của các khối encoder còn lại là đầu ra của khối endcoder phía dưới. Các embeddings được tạo thành từ việc kết hợp vector word embedding + positional embedding.

Phép tính đầu tiên trong sef-attention là nhân mỗi vector embedding đầu vào với 3 ma trận trọng số Wq, Wk, Wv để tạo ra 3 vector q, k v. Các ma trận trọng số này sẽ được cập nhật trong quá trình đào tạo. Vector q và k được dùng để tính trọng số khuếch đại thông tin cho các từ trong câu. Vector v là vector biểu diễn của các từ trong câu.

Quá trình tính toán hàm attention trên toàn bộ tập các câu truy vấn một cách đồng thời được đóng gói thông qua ma trận q. keys và values cũng được đóng góicùng nhau thông qua matrix k và v. Phương trình Attention như sau:

$$Attention(q, k, v) = softmax(\frac{qk^T}{\sqrt{d_k}})V$$

Việc chia cho là số dimension của vector key nhằm mục đích tránh tràn luồng nếuso mũ là quá lớn.

2.2.3.2 Cơ chế Multi-head Attention

Như vậy sau quá trình Scale dot production ta sẽ thu được 1 ma trận attention. Các tham số mà model cần tinh chỉnh chính là các ma trận W_q, W_k, W_v. Mỗi quá trình như vậy được gọi là 1 head của attention. Khi lặp lại quá trình này nhiều lần (trong bài báo là 3 heads) ta sẽ thu được quá trình Multi-head Attention.

Để trả về output có cùng kích thước với ma trận input ta chỉ cần nhân với ma trận W_0 có chiều rộng bằng với chiều rộng của ma trận input. Sau đó đi qua một

bước gọi là Add Normalize nữa trước khi đưa vào layer Feed Forward.

Ý nghĩa của cơ chế multi-head này là để tăng thêm phần chắc chắn trong việc quyết định thông tin nào cần khuếch đại, thông tin nào cần bỏ qua.

2.2.4 Điểm cải tiến của Transformer so với LSTM

Kiến trúc transformer cho phép thực hiện các phép tính song song -> giảm đáng kể thời gian train/inference, tận dụng được sức mạnh tính toán của multi-GPU.

Yêu cầu ít thời gian đào tạo hơn so với các kiến recurrent neural architectures trước đây, chẳng hạn như bộ nhớ ngắn hạn dài (LSTM), và biến thể sau này của nó đã được áp dụng phổ biến để đào tạo các mô hình ngôn ngữ lớn trên các bộ dữ liệu (ngôn ngữ) lớn.

Kiến trúc này hiện không chỉ được sử dụng trong xử lý ngôn ngữ tự nhiên và thị giác máy tính, mà còn trong xử lý âm thanh và đa phương thức. Nó cũng đã dẫn đến sự phát triển của các hệ thống được đào tạo trước, chẳng hạn như generative pre-trained transformers (GPT) và BERT.

2.3 Lựa chọn kỹ thuật, công nghệ và nền tảng

2.3.1 Rasa

Rasa là một framework chatbot mã nguồn mở dựa trên kiến trúc machine learning.Rasa cung cấp một bộ công cụ và thư viện giúp nhà phát triển xây dựng chatbot thông minh và linh hoạt.

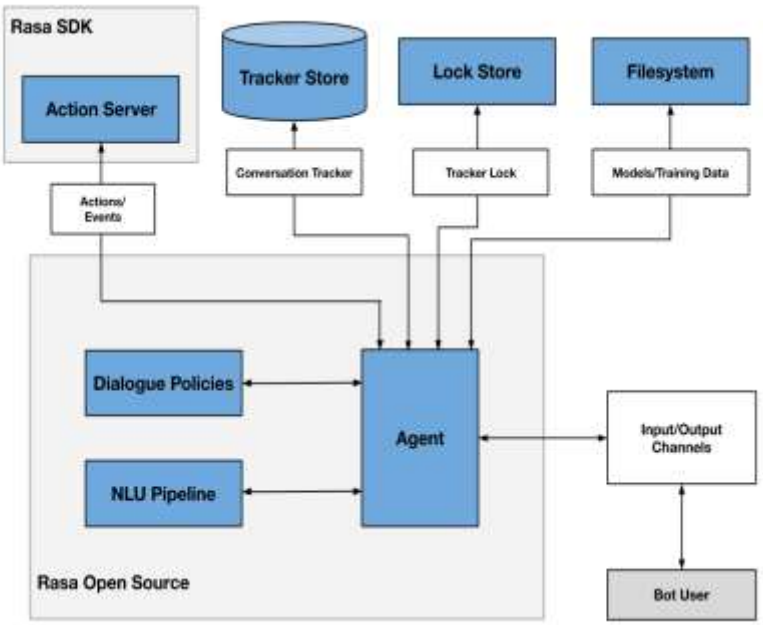
2.3.1.1 Thành phần cơ bản

Rasa có hai thành phần chính là Rasa NLU và Rasa Core.

- Rasa NLU là một mô hình ngôn ngữ lớn được đào tạo để phân tích ngôn ngữ tự nhiên. Rasa NLU có thể được sử dụng để hiểu ý định của người dùng và trích xuất các thực thể từ các cuộc trò chuyện.
- Rasa Core là một bộ điều khiển hội thoại được sử dụng để quản lý luồng hội thoại giữa chatbot và người dùng. Rasa Core có thể được sử dụng để thực hiện các hành động, chẳng hạn như tìm kiếm thông tin hoặc gửi email.

2.3.1.2 Các thành phần hệ thống chatbot RASA

Bên cạnh hai thành phần cốt lõi trên, hệ thống chatbot RASA cần thêm các thành phần để xử lý bộ nhớ, kết nối các thành phần lại với nhau để hoạt động hiệu quả. Dưới đây là sơ đồ kết nối các thành phần của rasa: NLU Pipeline: Các module RASA NLU Dialog Policies: Các policy của RASA Core Action Server: Các action đáp lại người dùng được viết bằng python (action mặc định là action dạng text) Tracker Store: Module lưu trữ các slot, entity, câu hội thoại (Đây chính là module bộ nhớ lưu trữ của chatbot) Lock Store: Module đảm bảo các câu hội thoại gửi đến chatbot được xử lý tuần tự, không bị race condition Filesystem: Module lưu trữ và quản lý các tệp của chatbot Agent: Module xử lý chung và kết nối các thành phần khác Trong các thành phần trên, chúng ta chủ yếu làm việc với NLU Pipeline, Dialog Policies, Action Server và sử dụng Tracker để xử lý dữ liệu trong Action Server.



Hình 2.5: Sơ đồ kết nối các thành phần của rasa

2.3.1.3 Tính năng của rasa và mục đích sử dụng

Rasa là một framework mạnh mẽ và linh hoạt có thể được sử dụng để xây dựng chatbot cho nhiều mục đích khác nhau. Rasa cung cấp một bộ công cụ và thư viện đầy đủ tính năng giúp nhà phát triển xây dựng chatbot nhanh chóng và

dễ dàng. Dưới đây là một số tính năng chính của Rasa:

- Kiến trúc machine learning: Rasa sử dụng kiến trúc machine learning để hiểu và phản hồi các câu hỏi và yêu cầu của người dùng. Điều này cho phép chatbot học hỏi và cải thiện theo thời gian.
- Dễ sử dụng: Rasa cung cấp một bộ công cụ và thư viện đầy đủ tính năng giúpnhà phát triển xây dựng chatbot nhanh chóng và dễ dàng.
- Mở rộng: Rasa là một framework mở và có thể được tùy chỉnh để đáp ứng nhu cầu cụ thể.

Rasa có thể được sử dụng để xây dựng chatbot cho các mục đích khác nhau, bao gồm:

- Hỗ trợ khách hàng: Chatbot có thể được sử dụng để trả lời các câu hỏi của khách hàng và giải quyết các vấn đề.
- Giáo dục: Chatbot có thể được sử dụng để cung cấp đào tạo và hướng dẫn.
- Giải trí: Chatbot có thể được sử dụng để chơi trò chơi và trò chuyện với ngườidùng.

Rasa là một lựa chọn tuyệt vời cho nhà phát triển chatbot muốn xây dựng chatbot thông minh và linh hoạt. Tuy nhiên rasa chưa hỗ trợ nhiều về mặt giao diện thao tác đối với nhà phát triển để quản lý luồng hội thoại giữa chatbot và người dùng nên em quyết định sử dụng thêm Botpress studio của Botpress kết với với Rasa NLU để xây dựng chatbot.

2.3.2 Botpress

Botpress là một framework chatbot mã nguồn mở, được xây dựng dựa trên Node.js. Botpress cung cấp một bộ công cụ và thư viện giúp nhà phát triển xây dựng chatbot một cách nhanh chóng và dễ dàng.

Botpress có một số tính năng nổi bật như:

- Kiến trúc modular: Botpress được thiết kế với một kiến trúc modular, chophép nhà phát triển dễ dàng mở rộng và tùy chỉnh chatbot.
- Giao diện người dùng trực quan: Botpress cung cấp một giao diện người dùng

trực quan giúp nhà phát triển xây dựng chatbot mà không cần phải viết mã.

- Hỗ trợ đa kênh: Botpress hỗ trợ triển khai chatbot trên nhiều kênh khác nhau,bao gồm web, di động, ứng dụng hội thoại và các kênh nhắn tin khác.
- Cộng đồng lớn: Botpress có một cộng đồng lớn và hoạt động, giúp nhà pháttriển dễ dàng tìm kiếm sự trợ giúp khi cần thiết.

Botpress có thể được sử dụng để xây dựng chatbot cho nhiều mục đích khácnhau, bao gồm:

- Hỗ trợ khách hàng: Chatbot có thể được sử dụng để trả lời các câu hỏi củakhách hàng và giải quyết các vấn đề.
- Giáo dục: Chatbot có thể được sử dụng để cung cấp đào tạo và hướng dẫn.
- Giải trí: Chatbot có thể được sử dụng để chơi trò chơi và trò chuyện với ngườidùng.

Botpress là một framework mạnh mẽ và linh hoạt có thể được sử dụng để xây dựng chatbot cho nhiều mục đích khác nhau. Botpress cung cấp một bộ công cụ và thư viện đầy đủ tính năng giúp nhà phát triển xây dựng chatbot nhanh chóng và dễ dàng. Dưới đây là một số ưu điểm và nhược điểm của Botpress:

- Ưu điểm:
  - Dễ sử dụng
  - Giao diện người dùng trực quan
  - Hỗ trợ đa kênh
  - Cộng đồng lớn và hoạt động
- Nhược điểm:
  - Không mạnh mẽ như Rasa
  - Yêu cầu kiến thức về Node.js

Nhìn chung, Botpress là một lựa chọn tuyệt vời cho nhà phát triển chatbot muốn xây dựng chatbot một cách nhanh chóng và dễ dàng. Botpress phù hợp cho các doanh nghiệp và tổ chức có quy mô nhỏ và vừa.

2.3.3 Django

Django là một framework web mã nguồn mở, được viết bằng Python. Django cung cấp một bộ công cụ và thư viện đầy đủ tính năng giúp nhà phát triển xây dựngwebsite và ứng dụng web một cách nhanh chóng và dễ dàng.

Django sử dụng Python làm ngôn ngữ lập trình chính. Điều này cho phép Django tận dụng lợi thế của nhiều tính năng mạnh mẽ của Python, chẳng hạn như tính năng gõ linh hoạt, cú pháp đơn giản và thư viện phong phú.

Python là một ngôn ngữ lập trình rất phổ biến và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, bao gồm phát triển trang web, học máy, khoa học máy tính và tự động hóa. Điều này giúp Django có một cộng đồng lớn và hoạt động, giúp nhà phát triển dễ dàng tìm kiếm sự trợ giúp khi cần thiết.

Dưới đây là một số lợi ích của việc sử dụng Python trong Django:

- Tốc độ phát triển nhanh: Python là một ngôn ngữ lập trình rất nhanh, cho phép nhà phát triển xây dựng website và ứng dụng web một cách nhanh chóng và dễ dàng.
- Tính linh hoạt: Python là một ngôn ngữ lập trình rất linh hoạt, giúp nhà phát triển dễ dàng mở rộng và tùy chỉnh website và ứng dụng web của mình.
- Bảo mật: Django cung cấp nhiều tính năng bảo mật tích hợp sẵn, giúp bảo vệ website và ứng dụng web khỏi các tấn công.
- Cộng đồng lớn: Python và Django có một cộng đồng lớn và hoạt động, giúp nhà phát triển dễ dàng tìm kiếm sự trợ giúp khi cần thiết.

2.3.4 ReactJS

React.js là một thư viện để xây dựng giao diện. React sử dụng một kiến trúc gọi là "components", cho phép nhà phát triển xây dựng các giao diện người dùng phức tạp từ các thành phần nhỏ hơn, đơn giản hơn. Hiệu suất cao: React sử dụng một kỹ thuật gọi là "virtual DOM" để cập nhật giao diện người dùng một cách hiệu quả. Điều này giúp React có thể render giao diện người dùng nhanh chóng và mượt mà, ngay cả đối với các ứng dụng phức tạp.

React có một số ưu điểm, bao gồm:

- Dễ học và sử dụng: React có một cú pháp đơn giản và dễ học. React cũng cung cấp nhiều tài liệu và công cụ học tập, giúp nhà phát triển dễ dàng bắt đầu sử dụng React.
- Linh hoạt: React có thể được sử dụng để xây dựng các giao diện người dùng cho nhiều loại ứng dụng khác nhau, bao gồm ứng dụng web và ứng dụng di động.
- Cộng đồng lớn: React có một cộng đồng lớn và hoạt động, giúp nhà phát triển dễ dàng tìm kiếm sự trợ giúp khi cần thiết.

2.4 Kết luận

Chương này trình bày về nghiên cứu các công nghệ, kỹ thuật xây dựng chatbot phổ biến hiện nay trên thế giới. Từ đó đưa ra công nghệ, kỹ thuật được sử dụng để xây dựng chatbot. Trong Chương 3 của luận văn sẽ trình bày về việc áp dụng công nghệ, kỹ thuật đã lựa chọn ở Chương 2 để xây dựng chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông.

CHƯƠNG 3. XÂY DỰNG CHATBOT TƯ VẤN, HỖ TRỢ NHẬP HỌC TẠI HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Chương 3 trình bày về cách thức áp dụng công nghệ, kỹ thuật đã lựa chọn ở Chương 2 vào xây dựng chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông.

3.1 Thu thập dữ liệu

Bộ dữ liệu được thu thập từ fanpage Tuyển sinh PTIT trong gần 6 năm từ tháng 3 năm 2017 đến tháng 3 năm 2023 dựa trên các tin nhắn và bình luận giữa page với học sinh, sinh viên. . . Toàn bộ bộ dữ liệu có dung lượng khoảng 41MB với gần 357200 dòng dữ liệu.

3.1.1 Xử lý dữ liệu và làm sạch

Để dễ thao tác với dữ liệu, em chuyển dữ liệu từ dạng txt sang csv. Sau đó quytrình xử lý dữ liệu được thực hiện như sau:

- Chuẩn hóa lại dấu của từ
- Regex
- Xóa dấu câu, xóa số trong câu, xóa ký tự đặc biệt, xóa emoji, xóa url, htmltag, …
- Xóa các lỗi chính tả, lỗi ngữ pháp, lỗi đánh máy,... khỏi dữ liệu văn bản. Cáclỗi này có thể làm cho dữ liệu văn bản trở nên khó hiểu và khó sử dụng.
- Loại bỏ dữ liệu không cần thiết
- Xóa stopword, danh sách stopword được thu thập và lọc từ dữ liệu được gần 1950 từ và cụm từ
- Tách văn bản thành các câu, đoạn nhỏ
- Sửa từ viết tắt, dưới đây là bảng các từ viết tắt trong bộ dữ liệu được sử dụng

Bảng 3.1: Bảng từ viết tắt

Từ viết tắt	Từ gốc
e	Em
a	Anh
cj	Chị
Add, ad, adm	admin
Ko, k, kh, khg, kg, hong, hok, khum	Không
Đc, dc	Được
r	Rồi
Nv, nvong	Nguyện vọng
ttnv	Thứ thự nguyện vọng
Mk	Mình
b	Bạn
t	Tôi
Ak, ah	À
Baoh, bh	Bao giờ, bây giờ
Trg, tr, trườg	Trường
Hv	Học viện
Ntn	Như thế nào
clc	Chất lượng cao
cccd	Căn cước công dân
Bn, bnh, baonh	Bao nhiêu
ktx	Ký túc xá
bhyt	Bảo hiểm y tế
Dky, dki, đăng kí, dki	Đăng ký
j	Gì
hsg	Học sinh giỏi
m	Mình
dgnl	Đánh giá năng lực
dgtd	Đánh giá tư duy
dhqg	Đại học quốc gia
thpt	Trung học phổ thông
Thptqg	Trung học phổ thông quốc gia
tk	Tài khoản
Kv, khvuc, kvuc	Khu vực
cmnd	Chứng minh nhân dân
Lm	Làm
hcm	Hồ chí minh
Tp hcm	Thành phố hồ chí minh

D, đ	Điểm
vs	Với
vd	Ví dụ
Oke, oki, okeee	Ok
nnao	Như nào
2k1, 2k2, 2k3, 2k4, 2k5	2001, 2002,2003, 2004, 2005

Bảng 3.2: Bảng từ viết tắt 2

Từ viết tắt	Từ gốc
Chòa	Chào
v	Vậy
cntt	Công nghệ thông tin
attt	An toàn thông tin
tmdt	Thương mại điện tử
Dtvt, dt vt	Điện tử viễn thông
Marketting, makettting	Marketing
hc	Học
vc	Việc
Cn dpt, cndpt	Công nghệ đa phương tiện
dpt	Đa phương tiện
Q1, q9	Quận 1, quận 9
tnthpt	Trắc nghiệm trung học phổ thông
Sdt, sđt, dt, đt	Số điện thoại, điện thoại
Tk, ck	tài khoản, chuyển khoản
hssv	Hồ sơ sinh viên
Gddt, gd dt	Giáo dục đào tạo
Uh, uk, um	ừ
bâyh	Bây giờ
xtkh	Xét tuyển kết hợp
hs	Học sinh
cccd	Căn cước công dân
sv	Sinh viên
dkxt	Đăng ký xét tuyển
lpxt	Lệ phí xét tuyển
z	Vậy
P, pk	Phải, phải không
dk	Điều kiện, đăng ký, đúng không, được không
kh	Không, khoa học

hk	Không, học kì
c	Chị, chưa
bh	Bây giờ, bao giờ
nt	Nhắn tin, nông thôn
cmt	Comment, chứng minh thư
v	Vâng, và, vậy
M	Mình, mỗi, mà
tn	Tin nhắn, tốt nghiệp

3.1.2 Chuẩn hóa dữ liệu

Chuẩn hóa dữ liệu văn bản là quá trình loại bỏ các lỗi và bất thường khỏi dữ liệu văn bản. Quá trình này giúp dữ liệu văn bản trở nên chính xác, dễ hiểu và dễ sử dụng hơn.

Các bước chuẩn hóa dữ liệu văn bản có thể được thực hiện thủ công hoặc sử dụng các công cụ tự động. Các công cụ tự động có thể giúp tiết kiệm thời gian và công sức trong quá trình chuẩn hóa dữ liệu văn bản.

Chuẩn hóa dữ liệu văn bản là một bước quan trọng trong quá trình xử lý dữ liệu văn bản. Quá trình này giúp dữ liệu văn bản trở nên chính xác, dễ hiểu và dễ sử dụng hơn, từ đó giúp nâng cao hiệu quả của việc xử lý dữ liệu văn bản.

Dưới đây là các bước chuẩn hóa dữ liệu văn bản được áp dụng trong đề tài:

- Loại bỏ dấu phụ: Tiếng Việt có một hệ thống dấu phụ phức tạp, hay còn gọi là dấu trọng âm, dùng để phân biệt các âm khác nhau. Tuy nhiên, dấu phụ cũng có thể gây ra sự thiếu nhất quán trong cách thể hiện văn bản nên chúngthường bị loại bỏ trong quá trình chuẩn hóa văn bản.
- Xử lý từ viết tắt: Tiếng Việt có một số từ viết tắt là sự kết hợp của các từ đã được rút gọn thành một từ duy nhất. Những sự rút gọn này có thể được mở rộng thành dạng đầy đủ trong quá trình chuẩn hóa văn bản.
- Xử lý số: Số tiếng Việt có thể được viết dưới nhiều dạng khác nhau, bao gồm chữ số Ả Rập, chữ số tiếng Việt và các dạng hỗn hợp. Chuẩn hóa văn bản thường chuyển đổi tất cả các số sang định dạng nhất quán, chẳng hạn như chữsố Ả Rập.

- Xử lý dấu câu: Dấu câu tiếng Việt có thể được sử dụng theo nhiều cách khác nhau và có thể có sự nhất quán trong cách sử dụng. Chuẩn hóa văn bản có thể liên quan đến việc tiêu chuẩn hóa việc sử dụng dấu câu để cải thiện tính nhấtquán.
- Xử lý khoảng trắng: Văn bản tiếng Việt có thể có lượng khoảng trắng khác nhau, bao gồm dấu cách, tab và dòng mới. Chuẩn hóa văn bản có thể liên quan đến việc tiêu chuẩn hóa việc sử dụng khoảng trắng để cải thiện tính nhất quán.
- Xử lý các thực thể được đặt tên: Văn bản tiếng Việt có thể chứa các thực thể được đặt tên, chẳng hạn như người, địa điểm, tổ chức. Chuẩn hóa văn bản có thể liên quan đến việc xác định và gắn thẻ các thực thể được đặt tên để cải thiện độ chính xác của các tác vụ NLP.

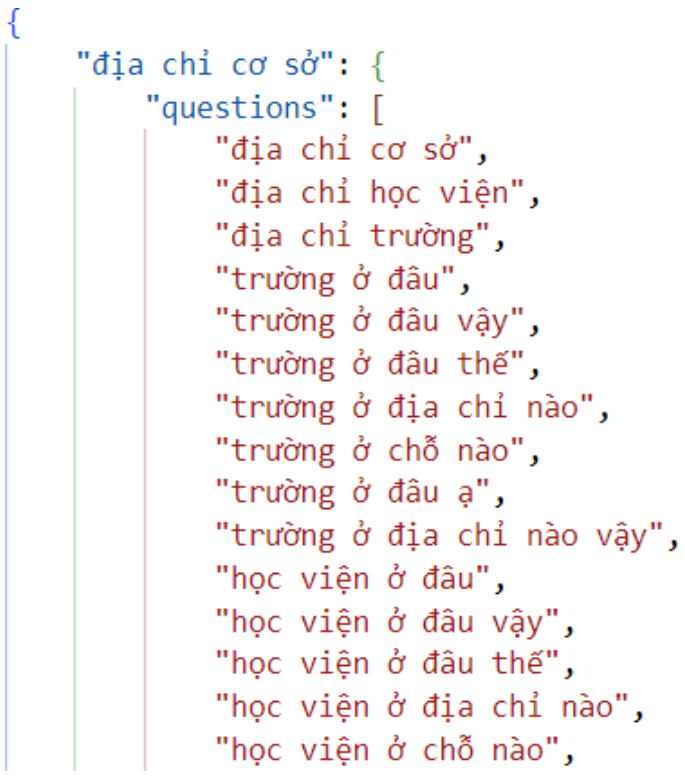
Bên cạnh đó, có nhiều công cụ và tài nguyên khác nhau dành cho việc chuẩn hóa văn bản bằng tiếng Việt như:

- Hunspell: Hunspell là trình kiểm tra chính tả nguồn mở cũng có thể được sử dụng để chuẩn hóa văn bản. Nó cung cấp một từ điển tiếng Việt và các quy tắc để xử lý dấu phụ, rút gọn và các khía cạnh khác của việc chuẩn hóa văn bản.
- Stanford CoreNLP: Stanford CoreNLP là bộ công cụ xử lý ngôn ngữ tự nhiên bao gồm bộ chuẩn hóa văn bản tiếng Việt. Nó cung cấp một bộ công cụ toàn diện để chuẩn hóa văn bản, bao gồm xử lý dấu phụ, rút gọn, số, dấu câu, khoảng trắng và các thực thể được đặt tên.
- VKSpeech: VKSpeech là bộ công cụ nhận dạng giọng nói và chuyển văn bản thành giọng nói tiếng Việt, bao gồm cả trình chuẩn hóa văn bản. Nó cung cấpmột tùy chọn nhẹ và hiệu quả để chuẩn hóa văn bản.
- Thư viện VIT (Công nghệ thông tin Việt Nam): Thư viện VIT là tập hợp các thư viện phần mềm mã nguồn mở phục vụ xử lý tiếng Việt. Nó cung cấp một trình chuẩn hóa văn bản tập trung vào việc xử lý các dấu phụ và sự rút gọn.
- NLTK (Bộ công cụ ngôn ngữ tự nhiên): NLTK là thư viện Python để xử lý

ngôn ngữ tự nhiên. Nó cung cấp một trình chuẩn hóa văn bản tiếng Việt tập trung vào việc xử lý dấu phụ và khoảng trắng.

3.1.3 Xây dựng bộ câu hỏi, câu trả lời

Danh sách bộ câu hỏi câu trả lời gồm 30 chủ đề xoay quanh chủ đề chính là nhập học gồm: địa chỉ cơ sở, mã trường, thông tin liên hệ, điểm trúng tuyển, số lượng chỉ tiêu tuyển sinh, các ngành đào tạo, mã ngành, hồ sơ xét tuyển, học phí, ký túc xá, tổ hợp xét tuyển, học bổng, cơ hội việc làm, đánh giá năng lực, xét tuyển kết hợp, phương thức xét tuyển, xét tuyển học bạ, hướng dẫn nhập học, hồ sơ nhập học chung, thời gian nhập học, giấy báo nhập học, minh chứng ưu tiên, chất lượng cao, nghĩa vụ quân sự, bảo hiểm y tế, sổ đoàn, giấy báo trúng tuyển, học phí nhập học.



```
{
    "địa chỉ cơ sở": {
        "questions": [
            "địa chỉ cơ sở",
            "địa chỉ học viện",
            "địa chỉ trường",
            "trường ở đâu",
            "trường ở đâu vậy",
            "trường ở đâu thế",
            "trường ở địa chỉ nào",
            "trường ở chỗ nào",
            "trường ở đâu ạ",
            "trường ở địa chỉ nào vậy",
            "học viện ở đâu",
            "học viện ở đâu vậy",
            "học viện ở đâu thế",
            "học viện ở địa chỉ nào",
            "học viện ở chỗ nào",
```

Hình 3.1: Câu hỏi mẫu

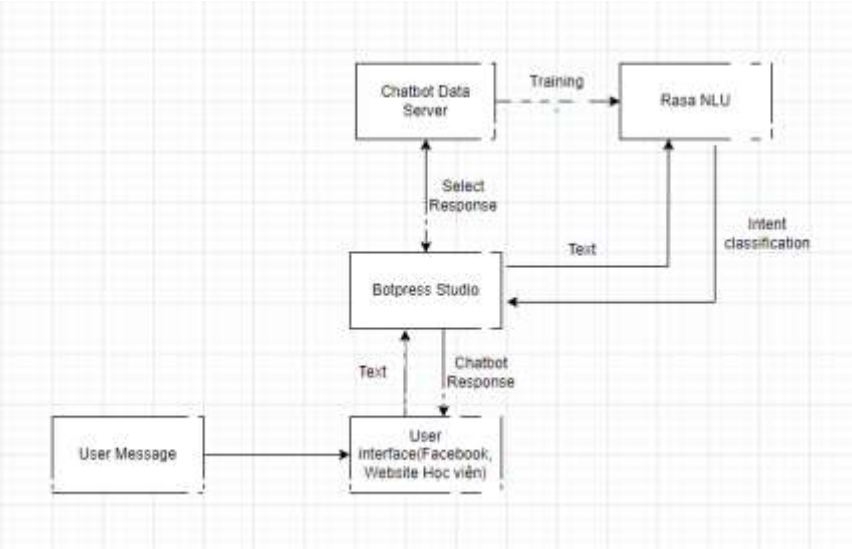
3.2 Kiến trúc tổng quan hệ thống

Kiến trúc tổng quan của hệ thống chatbot bao gồm các thành phần chính sau:

- Giao diện người dùng Botpress Studio: Là thành phần giao tiếp trực tiếp

với người dùng, cung cấp cho người dùng cách thức tương tác với chatbot. Giao diện người dùng có thể được xây dựng dưới dạng ứng dụng web, ứng dụng di động, hoặc chatbot tích hợp trên các nền tảng mạng xã hội.

- NLU Rasa: Là thành phần chịu trách nhiệm xử lý ngôn ngữ tự nhiên (NLP),hiểu và phân loại ý định của người dùng.
- Cơ sở dữ liệu Chatbot Data Server: Lưu trữ dữ liệu cần thiết cho hệ thống chatbot, dữ liệu về bộ câu hỏi câu trả lời, dữ liệu về người dùng, dữ liệu vềcác truy vấn của người dùng, và dữ liệu về các phản hồi của chatbot.



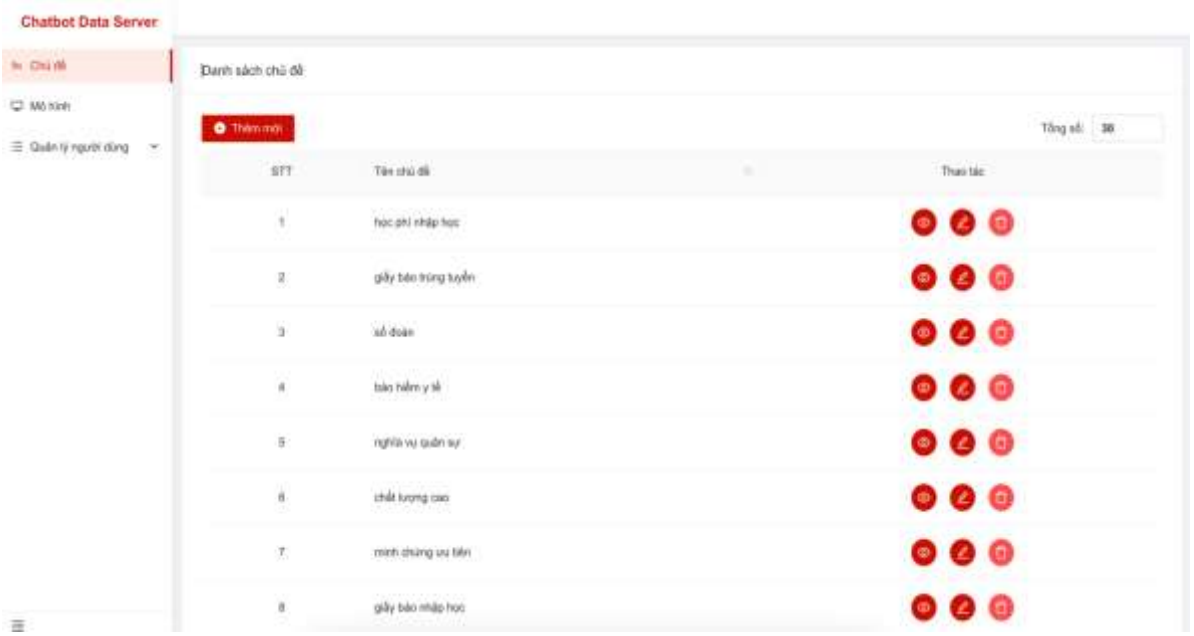
Hình 3.2: Kiến trúc tổng quan hệ thống

3.3 Xây dựng module quản lý dữ liệu bằng Django và ReactJS

Module quản lý dữ liệu chatbot được xây dựng nhằm mục đích quản lí danh sách chủ đề, danh sách câu hỏi thuộc chủ đề, quản lí câu trả lời chung cảu các chủ đề, quản lí mô hình và nhiều tính năng khác như quản lí người dùng, phân quyền ngườidùng hệ thống.

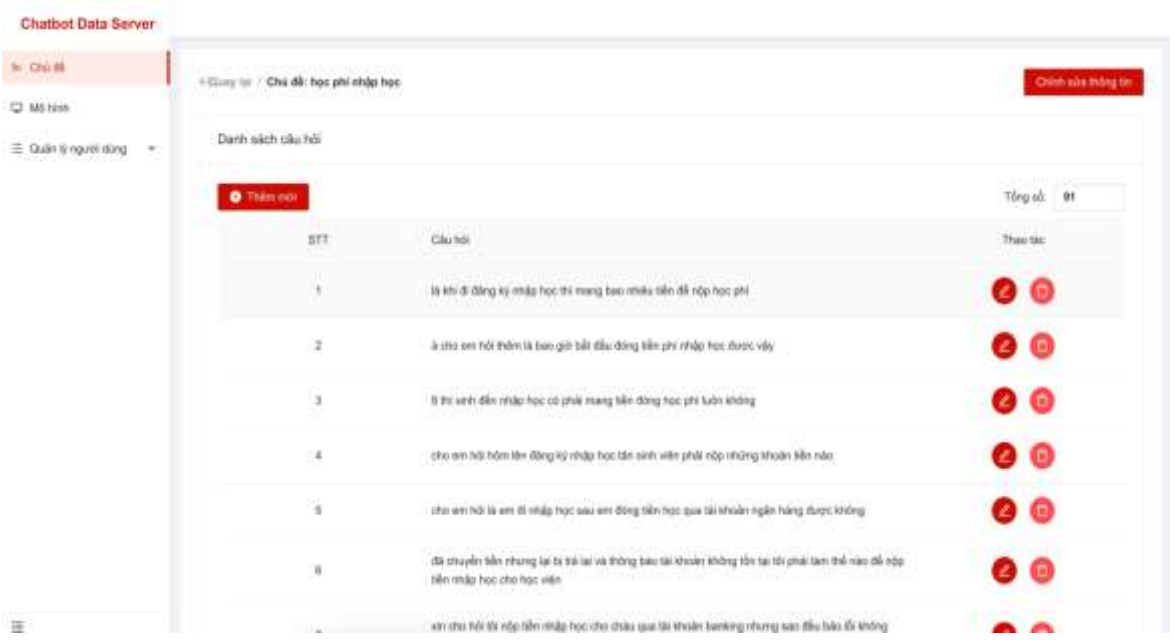
Dưới đây là các hình ảnh giao diện hệ thống:

Sau khi đăng nhập thành công người dùng được chuyển hướng đến giao diện quản lí chủ đề như sau:



Hình 3.3: Giao diện quản lí danh sách chủ đề

Tại giao diện quản lí chủ đề chọn xem chi tiết chủ đề sẽ chuyển hướng đến giao diện chi tiết chủ đề gồm danh sách các câu hỏi tương đồng của chủ đề đó và thứ tựcâu trả lời chung cho các câu hỏi:



Hình 3.4: Giao diện chi tiết chủ đề - Câu hỏi

3.4 Xây dựng NLU dựa trên Rasa

Mục đích chính của Rasa NLU là phân tích thông tin do người dùng cung cấp cho chatbot, thông tin này bao gồm các ý định và thực thể cần được trích xuất. Trên nền tảng mã nguồn mở Rasa, tin nhắn gửi đến được xử lý bởi một chuỗi các hàm

chức năng, các hàm này được thực thi lần lượt bên trong quy trình xử lý được định danh trong config.yml . Với quy trình nhúng có giám sát, ta có thể huấn luyện với bất kỳ ngôn ngữ trên thế giới vì công việc này sẽ bắt đầu huấn luyện mọi thứ từ đầu. Quy trình chi tiết Rasa pipeline như sau:

Hình 3.5: Rasa NLU Pipeline

Rasa cũng cho phép người dùng có thể tùy ý thay đổi các thành phần và xây dựng các quy trình mới. Bên cạnh quy trình đã được đề cập ở trên, em đã tùy chỉnh một quy trình hợp lý cho chatbot bằng cách sử dụng mô hình ngôn ngữ hiện đại BERT multilingual base model để xử lý các vectơ từ vựng đã được tiền huấn luyện vì mô hình có hỗ trợ tiếng Việt. Ví dụ về quy trình BERT multilingual base modelmà em đã sử dụng cho Chatbot có thể được thấy ở hình sau:

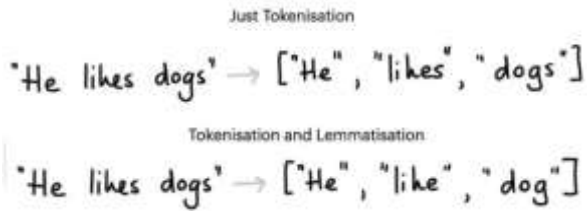
```yaml
- name: components.vi_tokenizer.VietnameseTokenizer
- name: WhitespaceTokenizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: LanguageModelFeaturizer
  model_name: bert
  model_weights: bert-base-multilingual-cased

- name: DIETClassifier
  epochs: 100
  constrain_similarities: true
  batch_strategy: sequence
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
- name: FallbackClassifier
  threshold: 0.7
  ambiguity_threshold: 0.1
```

Hình 3.6: Rasa pipeline đã chỉnh sửa

Bước đầu tiên là chia đoạn văn thành các đoạn văn bản nhỏ hơn, được gọi làTokenizer. Điều này phải xảy ra trước khi văn bản Featurizer cho máy học, đó là lý

do tại sao Tokenizer được liệt kê đầu tiên ở đầu quy trình.

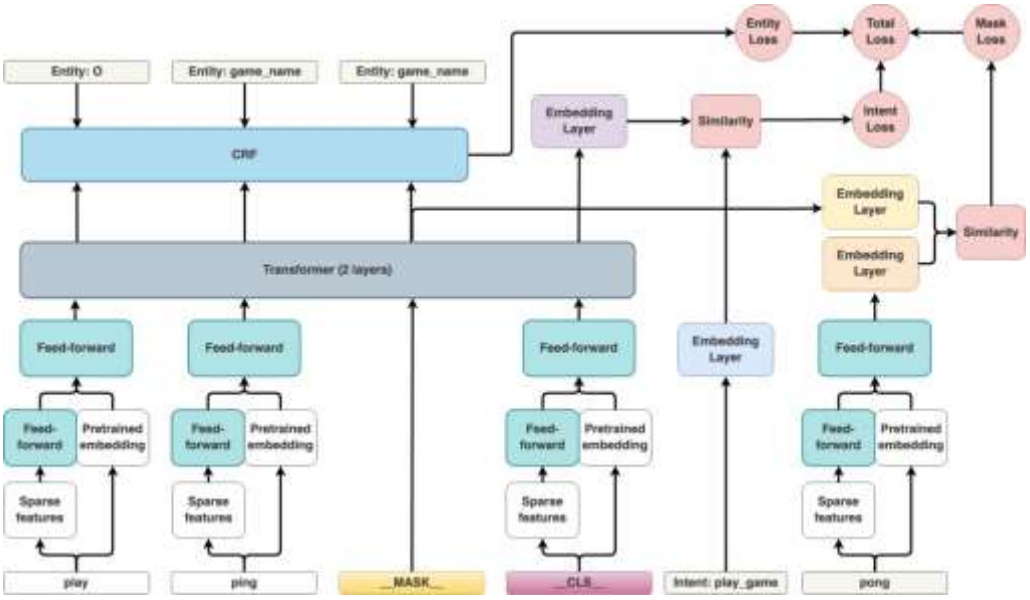


Hình 3.7: Chi tiết Tokenizer

Việc lựa chọn Tokenizer có thể ảnh hưởng đến loại Featureizer, và thứ tự của

các thành phần trong quy trình NLU cũng cần được xem xét, vì không thể sắp xếp một Featureizer trước Tokenizer vì đầu ra của Tokenizer sẽ đóng vai trò là đầu vào của Featureizer. Trong hầu hết các trường hợp, WhitespaceTokenizer hoạt động tốt cho bất kỳ ngôn ngữ nào. Tuy nhiên, em đã tự viết 1 file tokenizer của mình kết hợp với thư viện under the sea là VietnamTokenizer để sử dụng cho đề tài.

Dựa trên các mô hình kiến trúc Transformer ở chương 2, đề án sẽ sử dụng mô hình phân loại BERT để nhúng, bên cạnh đó em cũng sử dụng Dual Intent và Entity Transformer (DIET) để xử lý cả phân loại ý định và nhận dạng thực thể cùng lúc. Kiến trúc DIET được mô tả bằng hình dưới đây:

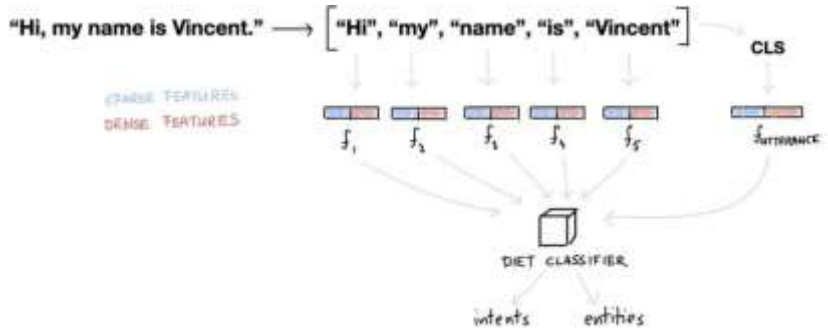


Hình 3.8: Kiến trúc DIET

Sau khi tạo xong các đặc điểm cho tất cả các tokens và cho toàn bộ câu, có thểchuyển nó sang mô hình phân loại ý định.Ở đây em sử dụng mô hình DIET của

Rasa để có thể xử lý cả việc phân loại ý định cũng như trích xuất thực thể. Nó cũngcó thể học hỏi từ cả tokens cũng như câu.

Dưới đây là cách DIET được sử dụng trong Rasa pipeline:



Hình 3.9: Quy trình DIET hoạt động

Bất cứ khi nào người dùng nói chuyện với chatbot, Rasa sẽ theo dõi nội bộ trạng thái của lời nói thông qua đối tượng 'Tin nhắn'. Đối tượng này được xử lý theo từng bước trong quy trình. Sơ đồ bên dưới cung cấp thông tin tổng quan về những gì xảyra khi Tin nhắn được xử lý.

Hình 3.10: Chi tiết quy trình NLU Rasa Pipeline

Thông báo đầu tiên bắt đầu dưới dạng một thùng chứa chỉ với lời nói đơn giản của người dùng. Sau khi tin nhắn đi qua Tokenizer, nó sẽ được chia thành các token. Khi tin nhắn đi qua CountVectorsFeaturizer, các feature sparse_features đã được thêm vào. Có sự khác biệt giữa các đặc điểm của chuỗi và toàn bộ câu. DIETClassifier sẽ tìm kiếm ‘sparse_features‘ và ‘dense_features‘ trong token để đưa ra dự đoán. Sau khi xử lý xong, nó sẽ đính kèm dự đoán ý định vào đối tượngtin nhắn.

3.5 Xây dựng giao diện hỗ trợ quản lý đoạn hội thoại bằng Botpress

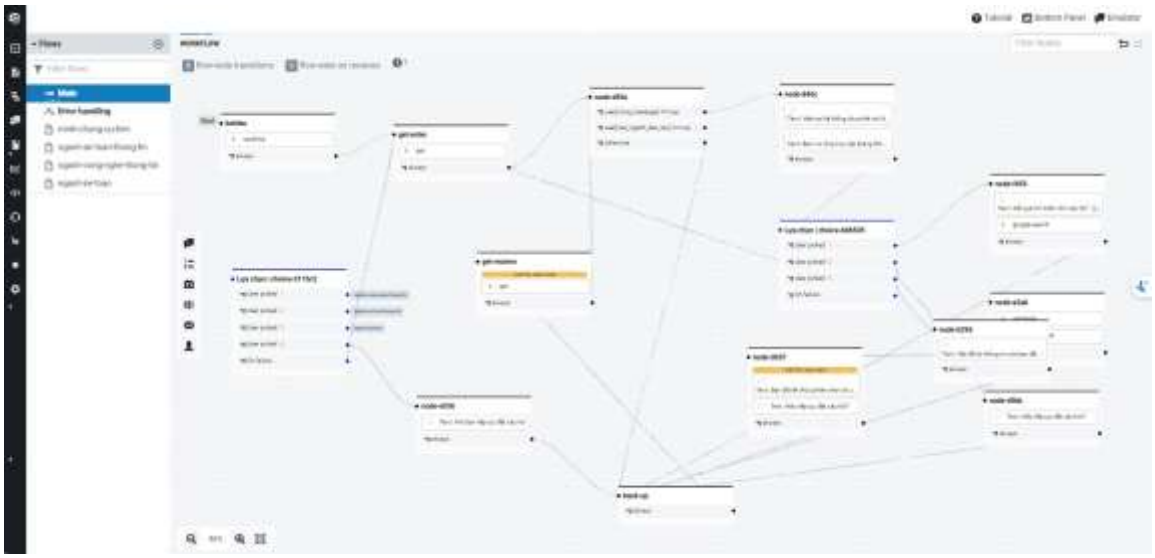
Trong đề tài, em sử dụng mã nguồn mở botpress phiên bản v12 của botpress trên github và điều chỉnh 1 số thành phần mã nguồn trong đó như thêm tiếng việt,chỉnh sửa giao diện để phù hợp với đề tài.

3.5.1 Thành phần giao diện quản lí luồng của chatbot

Từ giao diện đăng nhập của Botpress, người dùng đăng nhập bằng tài khoản được cấp.

Hình 3.11: Giao diện đăng nhập Botpress Studio

Sau khi đăng nhập thành công, người dùng được chuyển hướng đến trang chủ của giao diện. Tại đây, người dùng chọn click tên bot muốn truy cập và được chuyển hướng đến giao diện quản lí luồng hội thoại.



Hình 3.12: *Giao diện quản lí luồng hội thoại trên Botpress Studio*

Tại giao diện luồng hội thoại của Botpress gồm 2 thành phần chính là Node và Lựa chọn

Node là đơn vị logic đàm thoại chính của chatbot của Botpress . Một

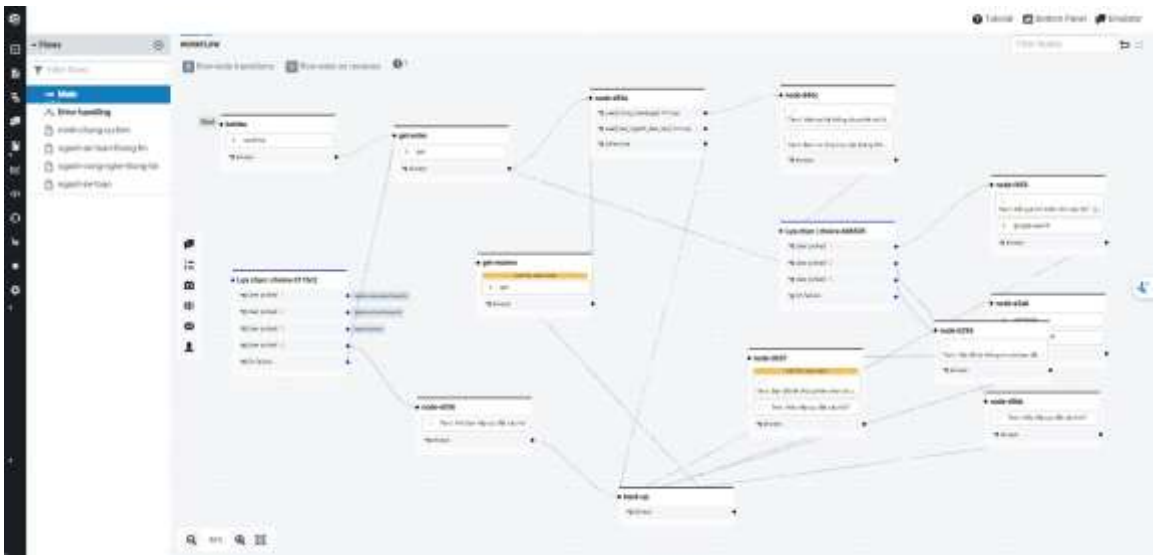
node thường chuyển sang một node hoặc luồng khác. Mỗi node sẽ có 3 thuộc tính là:

- On enter là nơi tạo các hành động, xử lí của bot
- On receive là nơi nhận input từ người dùng và sẽ xử lí chúng
- Transitions là nơi đặt biến, điều kiện nối giữa các nodeLựa chọn là một node để người dùng chọn khi hệ thống không thể đưa ra câu trả lời chính xác hoặc muốn đưa ra lựa chọn cho người dùng chọn.

Hình 3.13: Node lựa chọn của giao diện quản lí luồng kịch bản

3.5.2 Xây dựng luồng kịch bản quản lí hội thoại giữa người dùng và bot

Dưới đây là giao diện quản lý hội thoại của botpress:



Hình 3.14: Giao diện quản lí luồng hội thoại trên Botpress Studio

- Mỗi người dùng sẽ sử dụng luồng kịch bản trên với session nhất định, sau 1 thời gian người dùng không nhắn với bot thì sẽ được reset lại cuộc trò chuyện.
- Bot sẽ bắt đầu bằng 1 node bắt đầu với câu trả lời mặc định là chào người dùng.
- Tiếp đến sẽ đến node nhận thông tin, tại đây em sử dụng custom action get để nhận thông tin từ người dùng và xử lý. Trong file get.js này nhóm tiến hành load câu hỏi và câu trả lời từ danh sách intent thông qua 2 hàm getIntentUtterances lấy câu trả lời và getAnswer lấy câu hỏi. Sau đó sử dụng hàm replyUser để xử lý câu hỏi của người dùng, trong hàm này sẽ lấy input từ người dùng sau khi qua NLU sẽ trả lại danh sách intent và độ tin cậy của chúng. Nếu các intent có độ tin cậy dưới 0.7 sẽ bị loại, ngược lại nếu có 1 intent lớn hơn 0.7 thì đưa ra câu trả lời, từ 2 intent trở lên sẽ cho người dùngchọn câu trả lời.
- Nếu qua node get mà hệ thống không xác định được câu trả lời thì sẽ chuyển qua node thông báo với người dùng là hệ thống không có câu trả lời cho câu hỏi hiện tại và chuyển tiếp qua node lựa chọn đối với các câu trả lời không trảlời được sẽ có 3 lựa chọn sau cho người dùng:
  - Để lại thông tin vào form google
  - Tìm kiếm google, hệ thống sẽ đưa ra link, title 1 số page kết quả tìmkiếm về câu hỏi cho người dùng
  - Tiếp tục hỏi câu khác
- Sau khi đi qua hết các node thì sẽ quay lại node get để tiếp tục nhận câu hỏi từ người dùng.

3.6 Kết luận

Chương 3 đã trình bày về cách thức áp dụng công nghệ, kỹ thuật đã lựa chọn ở Chương 2 vào xây dựng chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông. Ở Chương cuối sẽ trình bày về quy trình thử nghiệm chatbot và triển khai trên một số nền tảng phổ biến như website của Học viện.

CHƯƠNG 4. TRIỂN KHAI THỬ NGHIỆM CHATBOT TƯ

VẤN, HỖ TRỢ NHẬP HỌC TRÊN FACEBOOK MESSENGER, WEBSITE CỦA HỌC VIỆN

4.1 Môi trường thử nghiệm và các thước đo đánh giá

4.1.1 Môi trường thử nghiệm

Các thử nghiệm được thực hiện trên máy tính cá nhân với thông số cấu hình củamô trường được mô tả cụ thể trong bảng sau:

Bảng 4.1: Bảng cấu hình môi trường thử nghiệm

Thông tin	Môi trường máy huấn luyện
Vi xử lý	CPU Intel Core I5-12400F
Dung lượng RAM	32GB
Dung lượng bộ nhớ	256GB
GPU	NVIDIA QUADRO RTX 4000 8GB GDDR6
Hệ điều hành	Ubuntu 22.04.3 LTS
Mô hình huấn luyện	bert-base-multilingual-cased
Python	3.10
Rasa	3.1

4.1.2 Thử nghiệm và đánh giá độ chính xác của Rasa NLU

Trong đề án này sẽ sử dụng các phương pháp đo gồm: accuracy score, Precision và Recall, F1 score.

- Accuracy: Cách tính đơn giản và thường được sử dụng nhiều nhất. Tính tỉ lệ giữa số mẫu được dự đoán đúng và tổng số mẫu. Thường được sử dụng cho các bài toán phân loại mà ổn định và không bị lệch hoặc không có sự mất cânbằng của các lớp.
- Precision – là tỷ lệ bao nhiêu cái đúng được lấy ra, Cân nhắc trên tập dữ liệu kiểm soát xem có bao nhiêu dữ liệu được mô hình phán đoán đúng.
- Recall – là tỷ lệ bao nhiêu cái được lấy ra là đúng, Chỉ số này còn được gọi

là độ bao phủ, tức là xét xem mô hình tìm được có khả năng tổng quát hóa như nào. Từ hai yếu tố độ chuẩn xác và độ bao phủ người ta đặt ra một chỉ số khác gọi là F1-Score

- F1-Score: Là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0). Do đó nó đại diện hơn trong việc đánh giáđộ chính xác trên đồng thời precision và recall.

Bộ dữ liệu được phân chia 80% đào tạo và 20% thử nghiệm từ file nlu.yml. Sau khi phân chia dữ liệu, em tiến hành đánh giá NLU pipeline với 2 mô hình là bert

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Hình 4.1: Các chỉ số đánh giá mô hình

và bert-base-multilingual-cased thu được kết quả như sau:

Bảng 4.2: Bảng kết quả xác định ý định với mô hình bert

Accuracy	Precision	Recall	F1-Score
0.92	0.93	0.92	0.92

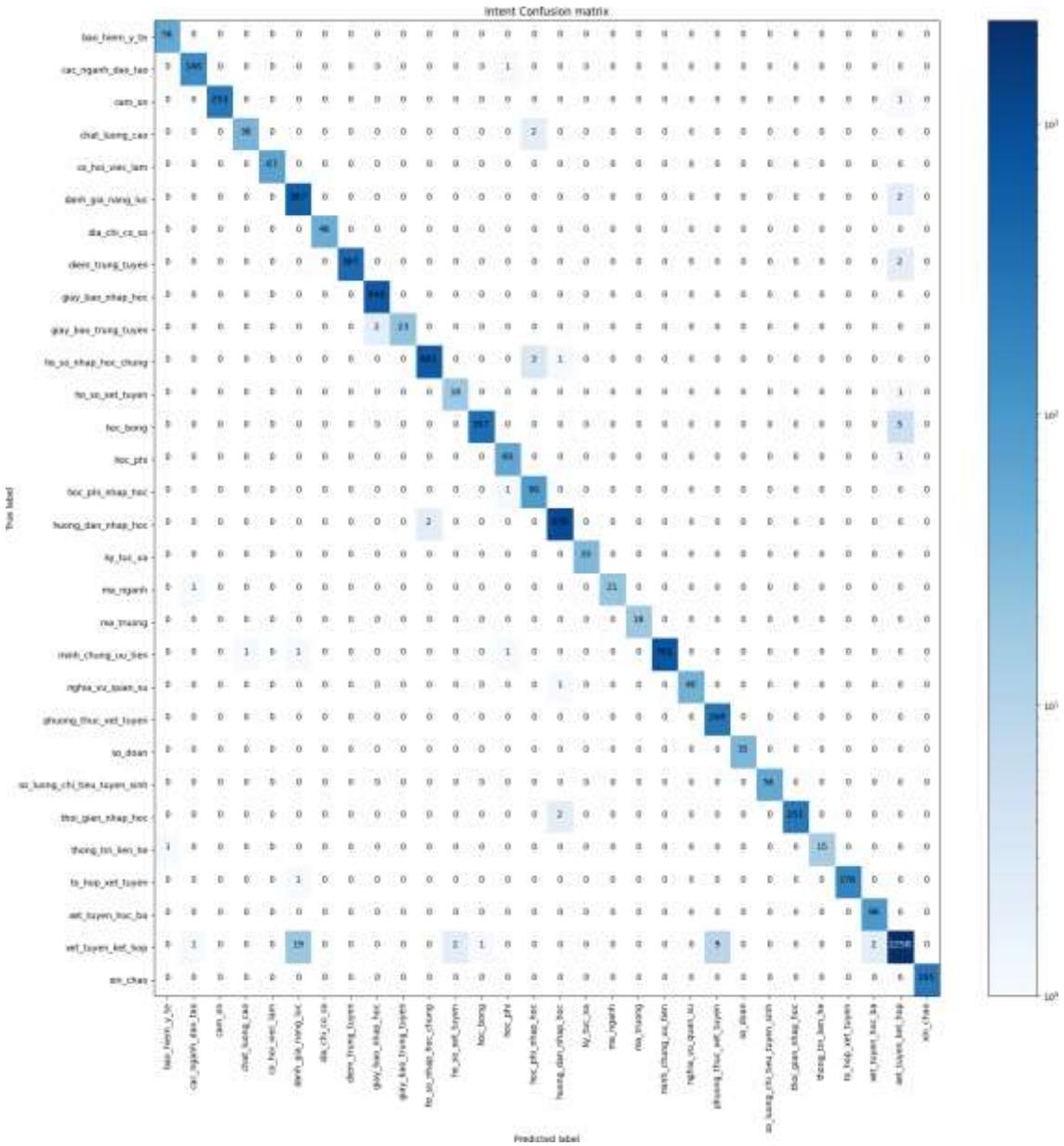
Bảng 4.3: Bảng kết quả xác định ý định với mô hình bert-base-multilingual-cased

Accuracy	Precision	Recall	F1-Score
0.94	0.95	0.94	0.94

Từ kết quả trên ta thấy mô hình bert-base-multilingual-cased thực hiện tốt hơn

mô hình bert vì mô hình này có hỗ trợ tiếng việt.

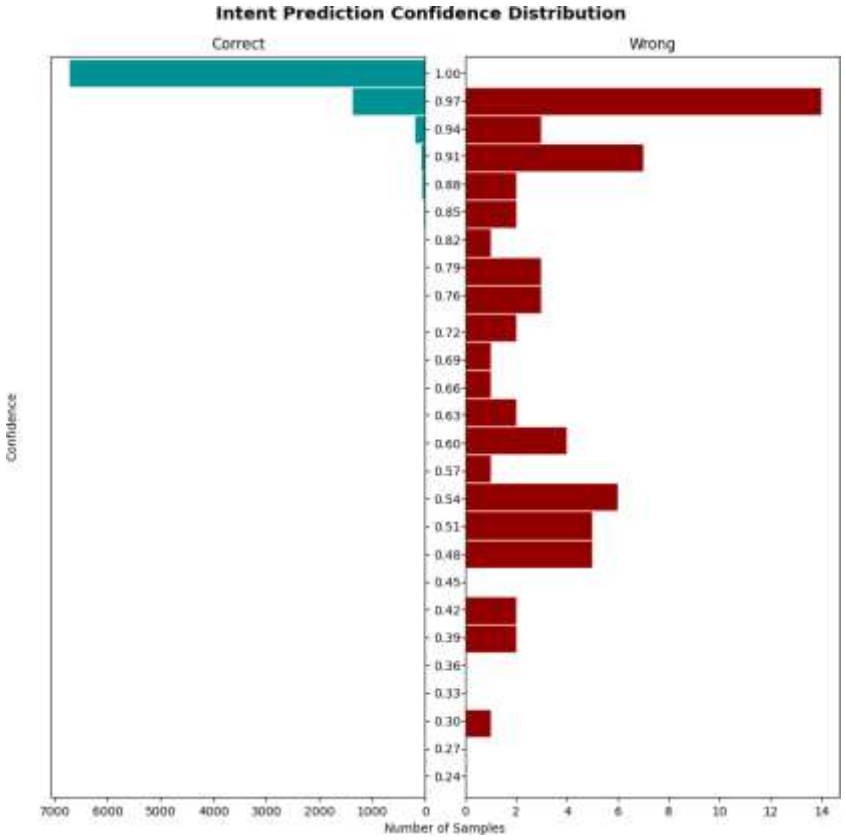
Sau khi lựa chọn được mô hình đào tạo bert-base-multilingual-cased chop Rasa Pipeline NLU tối ưu nhất thì kết quả thực hiện thử nghiệm với tập test ta được kếtquả sau các lần thử như sau:



Hình 4.2: Ma trận ước lượng nhầm lẫn xây dựng dữ liệu intent

Ma trận nhầm lẫn trên cho thấy ý định nào bị nhầm lẫn với ý định khác. Bất

kỳ mẫu nào đã được dự đoán không chính xác sẽ được ghi lại và lưu lại để gỡ lỗi dễ dàng hơn.



Hình 4.3: Biểu đồ độ tin cậy cho các dự đoán

Biểu đồ giúp ta hình dung độ tin cậy cho tất cả các dự đoán, với các dự đoán đúng và sai được hiển thị bằng các thanh màu xanh lam và màu đỏ tương ứng. Việc cải thiện chất lượng dữ liệu đào tạo sẽ di chuyển các thanh biểu đồ màu xanh lam lên trên biểu đồ và các thanh biểu đồ màu đỏ xuống dưới biểu đồ. Nó cũng sẽ giúpgiảm số lượng thanh biểu đồ màu đỏ.

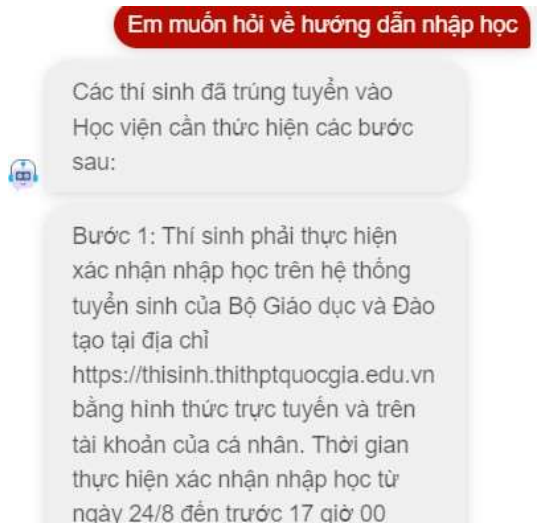
4.1.3 Thử nghiệm trò chuyện với chatbot

Chatbot được em xây dựng với gần 30 chủ đề về tuyển sinh, nhập học như điểm tuyển sinh, chỉ tiêu tuyển sinh, hướng dẫn nhập học, hướng dẫn chuẩn bị hồ sơ nhập học, thông tin về các vấn đề liên quan nhập học như giấy báo nhập học, sổ đoàn, giấy chuyển nghĩa vụ quân sự, bảo hiểm y tế. Ngoài ra chatbot cũng được

trang bị các thông tin cơ bản về học viện như địa chỉ, mã trường hay thông tin liên hệ.

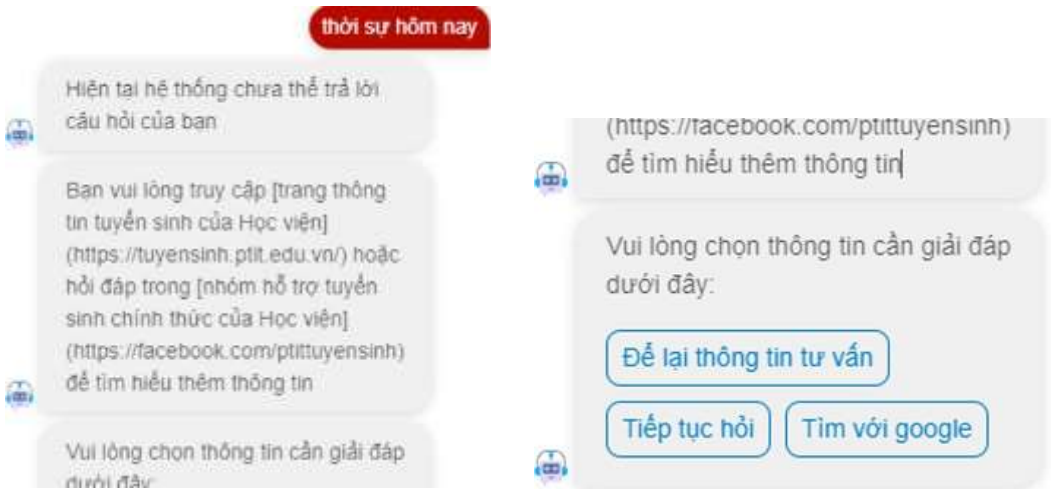
Từ các chủ đề nêu trên, em đã thử nghiệm một số kịch bản hội thoại với chatbot cụ thể như sau:

- Hỏi đáp về hướng dẫn nhập học, thông tin hồ sơ nhập học và các thông tin nhập học khác như bảo hiểm y tế, giấy chuyển nghĩa vụ quân sự, giấy báo nhập học, sổ đoàn, kí túc xá…



Hình 4.4: Hỏi đáp với chatbot về hướng dẫn nhập học

Hình 4.5: Hỏi đáp với chatbot về hồ sơ nhập học



Hình 4.6: Chatbot đưa ra lựa chọn khi gặp câu hỏi ngoài phạm vi

Hình 4.7: Chatbot đưa ra lựa chọn khi người dùng để lại thông tin tư vấn

Với các kịch bản thử nghiệm nêu trên, chatbot đã trả lời được gần 30 chủ đề với tỉ lệ đúng khá cao. Đối với các câu hỏi chatbot chưa trả lời được, em sẽ đưa ra hai hướng là tìm kiếm trên google và để lại thông tin để tiếp tục xử lí, thêm dữ liệu và tiến hành nâng cấp cho chatbot ở các phiên bản sau.

4.2 Cài đặt và triển khai hệ thống

4.2.1 Cài đặt hệ thống

Hệ thống chatbot gồm 2 dự án nhỏ là quản lý data chatbot và Botpress Studio. Toàn bộ mã nguồn đều được lưu trữ tại gitlab cá nhân của em và được để ở chế độ riêng tư. Dưới đây là hướng dẫn cài đặt chúng:

- Cài đặt hệ thống quản lý data: sau khi tải project về và giải nén, chạy lệnh ***docker compose up -d*** và đợi project chạy lên. Để kiểm tra hệ thống đã chạyđược chưa thì cần xem log của docker.

- Cài đặt Botpress Studio: sau khi tải project về và giải nén, chạy lệnh ***yarn install*** sau đó chạy ***yarn start***.

  Lần đầu truy cập giao diện cần đăng kí tài khoản, nên đăng kí là email thật để botpress có thể gửi thông tin về email trong 1 số trường hợp.

  Lưu ý cấu hình hệ thống của bot hiện tại file bot.config.js. Ở file này cần chủy đến đoạn code sau:

```
  "messaging": {
    "channels": {
      "messenger": {
        "enabled": true,
        "accessToken": "EAALP29cLQLgBAPNOZAM5WwE3akb5Bum71E7C2Z
Bx1jwlpElZAVuxbpy9diECgmN7vhDCKjApPO2jskma11ZB3u9r01ljspxC4gHPN
eEEnz8bJ4R0R97EccobgoN48yDougj1d648jPxwha9DwHwQ6IAPcVZAnjcRrnNC
DfJuVzQBtbVoKZCEmnz31dlwCtd246kJBXZACdDLwZDZD",
        "appSecret": "ebf68967859ccc9127144f809f90dd63",
        "verifyToken": "3498621986922582188213299256694 7"
      }
    }
  }
}
```

Hình 4.8: Cấu hình của bot để kết nối với facebook messenger

Ở đoạn code trên là các thông tin accessToken của page cần kết nối trên facebook, appSecret khóa kết nối đến app messenger của page và verifyToken là thông tin khóa xác thực với tên miền https của bot để kết nối với messengercủa page.

4.2.2 Các yêu cầu đối với cấu hình máy cài đặt và lưu ý

Đối với máy cài đặt cần một số yêu cầu sau để phù hợp với đề án:

- Cần cài node phiên bản 12.xx.xx trên máy cài.
- Cài cài docker và docker compose.
- Máy cài đặt cần có cấu hình RAM tối thiểu 16GB, Bộ nhớ trống 128GB. Cần

  chạy máy liên tục và không tắt đi.
- Cần backup dữ liệu bằng cách export data của bot hiện tại để tránh trường

hợp khi hệ thống bị hỏng do lỗi hay mất điện thì phải build lại và thao tác từđầu lại trên sudio.

- Dưới đây là hướng dẫn xuất thông tin bot tại giao diện của admin chọn xem thêm và xuất thông tin của bot để tránh mất dữ liệu bot:

Hình 4.9: Trích xuất thông tin bot tại giao diện chính

4.2.3 Triển khai hệ thống chatbot

Sau khi hoàn thành xây dựng và cài đặt hệ thống. Em đã thử nghiệm triển khai trên giao diện website của Học Viện

Hình 4.10: Triển khai chatbot trên website học viện

4.3 Kết luận

Chương này đã trình bày về quy trình thử nghiệm chatbot, cách thức triển khai hệ thống trong thực tế và kết quả thực nghiệm. Từ đó tiến hành đánh giá kết quả của

hệ thống và đề xuất phương pháp nghiên cứu, phát triển sau này.

KẾT LUẬN

Đề án tập trung nghiên cứu bài toán **Nghiên cứu và xây dựng chatbot tư vấn, hỗ trợ nhập học tại Học viện Công nghệ Bưu chính Viễn thông**. Trong đó tập trung chính vào các kĩ thuật xây dựng NLU, xử lí ngôn ngữ tự nhiên, áp dụng các mô hình học máy, học sâu vào quy trình xây dựng phân loại ý định người dùng. Từ NLU đã xây dựng kết hợp với giao diện hỗ trợ xây dựng các luồng kịch bản trò chuyện giữa người dùng và bot. Thông qua thực nghiệm, phân tích và đánh giá các kịch bản, sau đó đánh giá hiệu năng bằng việc triển khai trên các nền tảng như facebook hay website của Học viện. Cuối cùng đưa ra hệ thống chatbot tư vấn, hỗ trợ nhập học. Cụ thể đề án đã được môt số kết quả sau:

- Tìm hiểu tổng quan về bài toán và quy trình xây dựng chatbot hỗ trợ hỏi đáp các thông tin tuyển sinh và nhập học.
- Nắm được kiến trúc và nhiệm vụ các thành phần, thuật toán và các kỹ thuật sử dụng trong chatbot như Mạng RNN, mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), Kiến trúc Transformer, Mô hình BERT...
- Xây dựng bộ dữ liệu về nhập học, tuyển sinh 2023 chính xác dựa trên các nguồnthu và thông tin từ Học viện cho bài toán chatbot.
- Nghiên cứu và ứng dụng Framework Rasa, tối ưu hóa các thuật toán khi áp dụng framework này. Tạo mô hình huấn luyện riêng để xác định ý định người dùng. Đồng thời sử dụng mô hình BERT Base Multilingual để cải tiến chất lượng của bài toán xác định ý định. Kết hợp NLU rasa với giao diện người dùng Botpress Studio để quản lý luồng hội thoại giữa người dùng và chatbot dễ dàng hơn.
- Hoàn thành mục tiêu xây dựng chatbot giúp giảm thiểu thời gian và công sức trong công tác hỗ trợ, tư vấn nhập học tại Học viện. Tích hợp trên website Học viện để tương tácvới người dùng.

Tuy nhiên đề án tốt nghiệp vẫn còn một số điểm hạn chế như nội dung của dữ liệu hỏi đáp phải được cập nhật thủ công hàng năm trên hệ thống theo đề án tuyển sinh, quy định của Học viện trong công tác tuyển sinh, nhập học. Hiện tại chatbot chỉ mới phân loại theo ý định người dùng và đưa ra câu trả lời chung cho ý định đó. Để cải thiện hệ thống, đề án đưa ra hướng phát triển trong tương lai như cải thiện thêm về nhận diện theo ngữ cảnh đề tăng độ chính xác của câu trả lời.

TÀI LIỆU THAM KHẢO

[1] "Minh họa chatbot," Siam Computing. [Online]. Available: https://siamcomputing.com/digital-transformation/chatbot/

[2] Nttuan8, "Recurrent neural network," Github, 05 2019. [Online]. Available: https://nttuan8.com/bai-13-recurrent-neural-network/

[3] ——, "Recurrent neural network," Github, 05 2019. [Online]. Available: https://nttuan8.com/bai-13-recurrent-neural-network/

[4] S. Rathor, "Simple rnn vs gru vs lstm :- difference lies in more flexible control," Medium, 06 2018. [Online]. Available: https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57

[5] P. Nam, "Tổng quan về chatbot," Viblo, 11 2021. [Online]. Available: https://viblo.asia/p/tong-quan-ve-chatbot-yMnKMByaZ7P

[6] H. T. H. T. N. Trung Thanh Nguyen, Anh Duc Le, "Neu-chatbot: Chatbot for admission of national economics university," NEU, 6 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X21000308

[7] N. P. J. U. L. J. A. N. G. K. I. P. Ashish Vaswani, Noam Shazeer, "Attention is all you need," Google, University of Toronto, 08 2023. [Online]. Available:https://arxiv.org/pdf/1706.03762.pdf

[8] Alessandro Lenci, "Understanding Natural Language Understanding Systems. A Critical Analysis", 2023.

[9] Kamal Nigam, "Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning", 1999.

[10] Tom Bocklisch, Joey Faulkner, Alan Nichol "Rasa: Open Source Language Understanding and Dialogue Management" 2017.

[11] Yoshua Bengio, "A Neural Probabilistic Language Model, J. of Machine Learning Research", 2003.

[12] Rico Sennrich, "Neural Machine Translation of Rare Words with Subword Units", 2016.

[13] Pranav Rajpurkar, "SQuAD: 100,000+ Questions for Machine Comprehension of Text", 2015.