

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC THĂNG LONG**



KHOÁ LUẬN TỐT NGHIỆP

PHÁT TRIỂN ỨNG DỤNG TRỢ LÝ ẢO CỐ VẤN HỌC TẬP SỬ DỤNG CÔNG NGHỆ RASA

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN THỊ HUYỀN CHÂU

SINH VIÊN THỰC HIỆN: LÊ QUANG HUY

MÃ SINH VIÊN: A37134

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

HÀ NỘI – 2023

**TRƯỜNG ĐẠI HỌC THĂNG LONG
KHOA CÔNG NGHỆ THÔNG TIN**



KHOÁ LUẬN TỐT NGHIỆP

PHÁT TRIỂN ỨNG DỤNG TRỢ LÝ ẢO CỐ VẤN HỌC TẬP SỬ DỤNG CÔNG NGHỆ RASA

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN THỊ HUYỀN CHÂU

SINH VIÊN THỰC HIỆN: LÊ QUANG HUY

MÃ SINH VIÊN: A37134

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

Hà Nội – 2023

TRƯỜNG ĐẠI HỌC THĂNG LONG
KHOA CÔNG NGHỆ THÔNG TIN



KHOÁ LUẬN TỐT NGHIỆP

PHÁT TRIỂN ỨNG DỤNG TRỢ LÝ ẢO CỐ VẤN HỌC TẬP SỬ DỤNG CÔNG NGHỆ RASA

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN THỊ HUYỀN CHÂU
SINH VIÊN THỰC HIỆN: LÊ QUANG HUY
MÃ SINH VIÊN: A37134
CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

Hà Nội – 2023

LỜI CẢM ƠN

Lời đầu tiên em xin được phép gửi lời cảm ơn chân thành nhất tới toàn bộ các thầy cô giáo trong khoa Công nghệ thông tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long đã truyền đạt những kiến thức quý báu và bổ ích nhất cho em trong những năm học vừa qua.

Đặc biệt, em xin chân thành cảm ơn cô – TS. Nguyễn Thị Huyền Châu và thầy Hồ Hồng Trường, giảng viên khoa Công nghệ thông tin, trường Đại học Thăng Long đã tận tình hướng dẫn, động viên và trực tiếp giúp đỡ em trong suốt quá trình triển khai và thực hiện phần mềm.

Để có được kết quả như ngày hôm nay, em rất biết ơn thầy cô gia đình đã động viên, khích lệ, tạo mọi điều kiện thuận lợi nhất trong suốt quá trình học tập cũng như quá trình thực hiện phần mềm này.

Bên cạnh đó, em xin chân thành cảm ơn các bạn trong khoa Công nghệ thông tin trường Đại học Thăng Long đã ủng hộ, giúp đỡ, chia sẻ kiến thức, kinh nghiệm và các tài liệu có được cho em trong suốt học tại trường.

Vì kiến thức còn hạn chế, trong quá trình triển khai và thực hiện phần mềm này em không tránh khỏi có những sai sót không mong muốn, chính vì vậy em rất mong nhận được những ý kiến đóng góp từ các thầy cô cũng như các bạn trong khoa để hoàn thiện phần mềm này cũng như bổ sung những kiến thức còn thiếu sót.

Cuối cùng em xin kính gửi lời chúc tới toàn thể các thầy cô trong khoa Công nghệ thông tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long luôn luôn mạnh khỏe và tràn đầy nhiệt huyết để truyền đạt kiến thức cho các thế hệ mai sau.

Em xin chân thành cảm ơn!

Hà Nội, ngày 30 tháng 6 năm 2023

Sinh viên thực hiện

LÊ QUANG HUY

LỜI CAM ĐOAN

Tôi xin cam đoan đề tài này được thực hiện dựa trên lý thuyết, thuật toán xử lý dữ liệu, thu thập dữ liệu và các mô hình học máy được trình bày trong khoá luận là do tôi thực hiện dưới sự hướng dẫn của TS. Nguyễn Thị Huyền Châu và thầy Hồ Hồng Trường. Mọi tài liệu, các bài báo và công cụ của các tác giả khác được sử dụng đều có trích dẫn tường minh về nguồn gốc và tác giả trong danh sách tài liệu tham khảo. Tôi xin chịu hoàn toàn trách nhiệm về lời cam đoan này!

Hà Nội, ngày 30 tháng 06 năm 2023

Sinh viên thực hiện

LÊ QUANG HUY

MỤC LỤC

CHƯƠNG MỞ ĐẦU	1
i. Đặt vấn đề	1
ii. Mục tiêu khoá luận	1
iii. Phương pháp nghiên cứu	2
iv. Phạm vi nghiên cứu.....	2
v. Cấu trúc khoá luận.....	2
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	4
1.1. Học máy [1]	4
1.1.1. Mô hình Neural Network [2]	5
1.1.2. Mô hình Recurrent Neural Networks [4].....	6
1.1.3. Siêu tham số	8
1.1.4. Hàm kích hoạt Softmax [9].....	8
1.1.5. Underfitting và Overfitting [11]	9
1.1.6. Phương pháp xử lý các vấn đề trong học máy.....	10
1.1.7. Phương pháp tối ưu Adam optimizer [18].....	12
1.2. Rasa [21]	14
1.2.1. Rasa NLU [23].....	15
1.2.2. Rasa Core [23]	16
1.3. Microsoft SQL Server [31]	21
1.4. Ngrok [33].....	21
1.5. Facebook Developer [35].....	22
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ CHỨC NĂNG	24
2.1. Mô tả bài toán	24
2.2. Sơ đồ kiến trúc tổng thể	27

2.2.1. Sơ đồ	27
2.2.2. Mô tả sơ đồ	27
2.3. Yêu cầu nghiệp vụ	27
2.4. Sơ đồ Usecase các chức năng chính	28
2.4.1. Sơ đồ Usecase	28
2.4.2. Các tác nhân có trong hệ thống.....	29
2.5. Đặc tả nhóm chức năng.....	30
2.5.1. UC1 Yêu cầu thông tin môn học	30
2.5.2. UC2 Yêu cầu thông tin môn học với điều kiện tiên quyết.....	31
2.5.3. UC3 Yêu cầu thông tin lộ trình học	32
2.5.4. UC4 Yêu cầu giải đáp một số vấn đề liên quan ĐKH hoặc CTĐT	33
CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ DỮ LIỆU HUẤN LUYỆN.....	34
3.1. Thiết kế cơ sở dữ liệu	34
3.1.1. Lược đồ quan hệ của cơ sở dữ liệu.....	34
3.1.2. Danh sách các bảng.....	35
3.1.3. Chi tiết các bảng	35
3.2. Tạo dữ liệu huấn luyện.....	36
3.2.1. Dữ liệu cho Intent Yêu cầu thông tin môn học	38
3.2.2. Dữ liệu cho Intent Yêu cầu thông tin môn học với điều kiện tiên quyết	38
3.2.3. Dữ liệu cho Intent Yêu cầu lộ trình học.....	39
3.2.4. Dữ liệu cho Intent: Yêu cầu giải đáp một số thắc mắc CTĐT hoặc ĐKH	39
CHƯƠNG 4. XÂY DỰNG CÁC HÀNH ĐỘNG TRONG RASA.....	41
4.1. Giới thiệu chung	41
4.1.1. Custom Actions	41
4.1.2. Thiết kế chức năng.....	41

4.2. Cấu trúc thư mục.....	42
4.3. Thuật toán trong Custom Actions	44
4.3.1. Mục tiêu	44
4.3.2. Giải pháp	44
4.4. Xây dựng và đánh giá mô hình	47
4.4.1. Xây dựng mô hình	47
4.4.2. Đánh giá mô hình.....	50
CHƯƠNG 5. CÀI ĐẶT VÀ THỰC NGHIỆM	57
5.1. Cài đặt.....	57
5.2. Thực nghiệm	60
5.2.1. Yêu cầu thông tin môn học.....	60
5.2.2. Yêu cầu lộ trình học	61
5.2.3. Hỏi giải đáp một số vấn đề	62
KẾT LUẬN	64
1. Tổng kết.....	64
2. Định hướng	64

DANH SÁCH HÌNH ẢNH

Hình 1.1 Kiến trúc Neural Networks [3]	5
Hình 1.2 Mô hình RNN Many to many [5]	7
Hình 1.3 Đồ thị của hàm Softmax [10]	9
Hình 1.4 Mô tả Early stopping [14].....	10
Hình 1.5 Mô tả Dropout [17].....	12
Hình 1.6 Rasa [22].....	14
Hình 1.7 Pipelines phần 1	17
Hình 1.8 Pipelines phần 2.....	19
Hình 1.9 Policies.....	20
Hình 1.10 Microsoft SQL Server [32].....	21
Hình 1.11 Ngrok [34]	21
Hình 1.12 Facebook developers [36].....	22
Hình 2.1 Khung chương trình đào tạo ngành Công nghệ thông tin khoá 32	26
Hình 2.2 Sơ đồ kiến trúc tổng thể [38]	27
Hình 2.3 Mô hình Usecase	28
Hình 2.4 Mô tả quy trình hoạt động của trợ lý ảo	29
Hình 3.1 Hình ảnh minh họa cho lược đồ của cơ sở dữ liệu	34
Hình 3.2 Cấu trúc thư mục Data [39]	36
Hình 4.1 Sơ đồ tuần tự chức năng giải đáp yêu cầu của User.....	42
Hình 4.2 Cấu trúc thư mục Rasa.....	43
Hình 4.3 Sơ đồ khối thuật toán nhận diện yêu cầu và trả về phản hồi	46
Hình 4.4 Mô hình RNN sử dụng trong khoá luận	48
Hình 4.5 Kết quả huấn luyện của mô hình	51
Hình 4.6 Biểu đồ e_loss	51

Hình 4.7 Biểu đồ i_loss	52
Hình 4.8 Biểu đồ t_loss	52
Hình 4.9 Biểu đồ i_acc	53
Hình 4.10 Biểu đồ e_f1.....	54
Hình 4.11 Confusion matrix của việc xác định entities	55
Hình 4.12 Confusion matrix của việc xác định intents	56
Hình 5.1 Giao diện mã truy cập.....	57
Hình 5.2 Giao diện Webhooks	58
Hình 5.3 Khởi tạo ngrok.....	58
Hình 5.4 Giao diện chỉnh sửa URL gọi lại.....	59
Hình 5.5 Giao diện cài đặt thông tin cơ bản.....	59
Hình 5.6 File credentials.yml	60
Hình 5.7 Liệt kê các môn học có chung tên	60
Hình 5.8 Liệt kê các môn thuộc một học phần.....	61
Hình 5.9 Chi tiết một môn học	61
Hình 5.10 Yêu cầu các môn có thể học	61
Hình 5.11 Lộ trình học của năm học	62
Hình 5.12 Lộ trình học của kỳ và năm học	62
Hình 5.13 Một số giải đáp khác phần 1	62
Hình 5.14 Một số giải đáp khác phần 2.....	63

DANH SÁCH BẢNG BIỂU

Bảng 2.1 Các tác nhân trong hệ thống.....	29
Bảng 2.2 Usecase yêu cầu thông tin môn học	31
Bảng 2.3 Usecase yêu cầu thông tin môn học với điều kiện tiên quyết.....	31
Bảng 2.4 Usecase yêu cầu lộ trình học.....	32
Bảng 2.5 Usecase yêu cầu giải đáp một số vấn đề liên quan ĐKH hoặc CTĐT	33
Bảng 3.1 Mô tả các bảng có trong cơ sở dữ liệu	35
Bảng 3.2 Mô tả các thành phần trong bảng Major	35
Bảng 3.3 Mô tả các thành phần trong bảng Subject.....	36
Bảng 3.4 Khung dữ liệu cho intent yêu cầu thông tin môn học	38
Bảng 3.5 Khung dữ liệu cho intent yêu cầu thông tin môn học với điều kiện tiên quyết .	39
Bảng 3.6 Khung dữ liệu cho intent yêu cầu lộ trình học	39
Bảng 3.7 Khung dữ liệu cho các intent yêu cầu giải đáp về CTH hoặc CTĐT	40

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	AI	Artificial Intelligence – Trí tuệ nhân tạo
2	API	Application Programming Interface – Giao diện lập trình ứng dụng
3	CSDL	Cơ sở dữ liệu
4	CTĐT	Chương trình đào tạo
5	CVHT	Cổ vấn học tập
6	ĐKH	Đăng ký học
7	ĐKTQ	Điều kiện tiên quyết
8	KKT	Khối kiến thức
9	LSTM	Long Short Term Memory – Mạng thần kinh hồi quy theo kiểu bộ nhớ ngắn – dài hạn
10	NN	Neural Network – Mạng thần kinh
11	RNN	Recurrent Neural Network – Mạng thần kinh hồi quy
12	UC	Use-case – Ca sử dụng
13	Web	Webpage/Website

CHƯƠNG MỞ ĐẦU

i. Đặt vấn đề

Ngày nay với đà tăng trưởng mạnh mẽ của khoa học công nghệ, công nghệ thông tin dần trở thành một trong những lĩnh vực đóng góp quan trọng hàng đầu cho sự tiến bộ của kinh tế xã hội. Các ứng dụng tin học đã và đang được áp dụng trong nhiều mặt của đời sống, trở thành công cụ giúp con người lưu trữ lẫn xử lý thông tin nhanh chóng và hiệu quả cao.

Hằng năm, mỗi tân sinh viên của trường Đại học Thăng Long đều được nhận một cuốn hướng dẫn học tập với nội dung chính là chương trình đào tạo các ngành học nhằm giúp sinh viên có cái nhìn tổng quát nhất về quá trình học tập trong tương lai của bản thân. Nhưng cả khi đã có trong tay tài liệu hướng dẫn, sinh viên vẫn sẽ ít nhiều gặp khó khăn trong việc định hướng và lên kế hoạch đăng ký học cho tương lai. Là một sinh viên, em cũng đã từng gặp những vấn đề như vậy. Trong hoàn cảnh đó, nếu một sinh viên có thể liên lạc với cố vấn học tập thì có thể dễ dàng được giải đáp thắc mắc. Tuy nhiên, do số lượng sinh viên quá lớn, các cố vấn học tập khó có thể lập tức trả lời toàn bộ các thắc mắc. Hiện trạng trên kéo theo nhu cầu cần thiết tin học hóa quy trình các cố vấn học tập hướng dẫn về chương trình đào tạo cho sinh viên.

Trí tuệ nhân tạo là một trong những ngành công nghệ thông tin phát triển mạnh mẽ trong thời gian gần đây. Lấy trung tâm là công nghệ học máy, các thuật toán trí tuệ nhân tạo đã và đang giúp giải quyết nhiều bài toán đa dạng như nhận diện hình ảnh, âm thanh, ngôn ngữ, v.v. Trong quá trình tìm hiểu về lĩnh vực này, em quan tâm đến các công nghệ trí tuệ nhân tạo giúp xây dựng ứng dụng trợ lý ảo. Qua trao đổi với giáo viên hướng dẫn, em nhận định đây cũng là cách tiếp cận hợp lý cho bài toán tin học hoá hoạt động cố vấn học tập giải đáp về chương trình đào tạo. Một ứng dụng hỗ trợ thông tin được thiết kế như trợ lý ảo sẽ có khả năng giúp sinh viên tiếp cận dịch vụ này được tự nhiên, thuận tiện và dễ dàng hơn.

ii. Mục tiêu khoá luận

Khoá luận hướng tới phát triển một trợ lý ảo có vai trò như cố vấn học tập.

iii. Phương pháp nghiên cứu

- **Nghiên cứu lý thuyết:** Các thuật toán học máy, các mô hình *Neural Networks*, đặc biệt là mô hình *Recurrent Neural Networks*, và framework *RASA*.
- **Vận dụng lý thuyết:** Thực hiện quy trình phân tích chức năng hệ thống, thiết kế cơ sở dữ liệu và thu thập dữ liệu huấn luyện, xây dựng thuật toán xử lý các hành động của người dùng và thiết kế mô hình học máy, huấn luyện và hiệu chỉnh mô hình.
- **Triển khai:** Kết quả là một ứng dụng trên nền Facebook Messenger.

iv. Phạm vi nghiên cứu

- **Phạm vi:** Mô phỏng hoạt động cố vấn học tập dựa trên chương trình học của khoa Công nghệ thông tin trường Đại học Thăng Long.
- **Thời gian:** 3 tháng.
- **Thông tin môn học:** Dựa theo quyền Trường Đại học Thăng Long Hướng dẫn học tập, Hà Nội, 2019-2020 khung chương trình đào tạo của chuyên ngành Công nghệ thông tin.
- **Dữ liệu huấn luyện:** Các câu hỏi liên quan đến đăng ký học, thông tin môn học và file các câu hỏi thường gặp của cố vấn học tập.

v. Cấu trúc khoá luận

Khoá luận bao gồm các chương với nội dung như sau:

Chương 1: Cơ sở lý thuyết

Chương này giới thiệu các công nghệ, khái niệm làm nền tảng cho việc phát triển ứng dụng trợ lý ảo ở các chương sau.

Chương 2: Phân tích thiết kế chức năng

Nội dung chương tập trung vào việc phân tích và thiết kế các chức năng của trợ lý ảo, từ đó đưa ra các yêu cầu về chức năng và các yêu cầu nghiệp vụ.

Chương 3: Thiết kế cơ sở dữ liệu và dữ liệu huấn luyện

Chương này tập trung vào việc thiết kế cơ sở dữ liệu phù hợp và tạo dữ liệu huấn luyện để phục vụ cho việc xây dựng các mô hình máy học.

Chương 4: Xây dựng các hành động trong RASA

Chương này sẽ tập trung vào việc xây dựng các hành động để *chatbot* có thể tương tác với người dùng một cách thông minh và tự động.

Chương 5: Cài đặt và thực nghiệm

Ở chương cuối cùng này, ta sẽ tìm hiểu cách cài đặt trợ lý ảo và thử nghiệm một số kịch bản để đánh giá kết quả cuối cùng.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

Chương này trình bày và giải thích các khái niệm, nguyên lý, công nghệ cơ bản sẽ làm cơ sở cho các phân tích và cài đặt ở những chương sau.

1.1. Học máy [1]

Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu huấn luyện, để đưa ra dự đoán hoặc quyết định mà không cần được lập trình chi tiết về việc đưa ra dự đoán hoặc quyết định này. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác (spam) hay không và tự động xếp thư vào thư mục tương ứng.

Dưới đây là các họ giải thuật chính trong học máy:

Học có giám sát: là một kỹ thuật học máy hướng tới xây dựng một hàm (function) dự đoán học từ dữ liệu huấn luyện. Dữ liệu huấn luyện bao gồm các cặp gồm đối tượng đầu vào (input - thường dạng vec-tơ), và đầu ra mong muốn (expected output - còn gọi là nhãn của dữ liệu).

Học không có giám sát: là một phương pháp của ngành học máy nhằm tìm ra một mô hình mà phù hợp với các quan sát. Nó khác biệt với học có giám sát ở chỗ là đầu ra đúng tương ứng cho mỗi đầu vào là không biết trước. Trong học không có giám sát, một tập dữ liệu đầu vào được thu thập mà không đi kèm đầu ra mong muốn. Học không có giám sát thường đối xử với các đối tượng đầu vào như là một tập các biến ngẫu nhiên. Sau đó, một mô hình mật độ kết hợp sẽ được xây dựng cho tập dữ liệu này.

Học tăng cường: là một tập giải thuật làm thành một lĩnh vực con của học máy, nó nghiên cứu cách thức một tác tử (agent) trong một môi trường nên chọn thực hiện các hành động nào để cực đại hóa một mục tiêu chi phí nào đó về lâu dài. Các thuật toán học tăng cường cố gắng tìm một chiến lược ánh xạ các trạng thái của thế giới tới các hành động mà agent nên chọn trong các trạng thái đó.

Khoá luận này sẽ sử dụng các thuật toán thuộc về phương pháp học có giám sát.

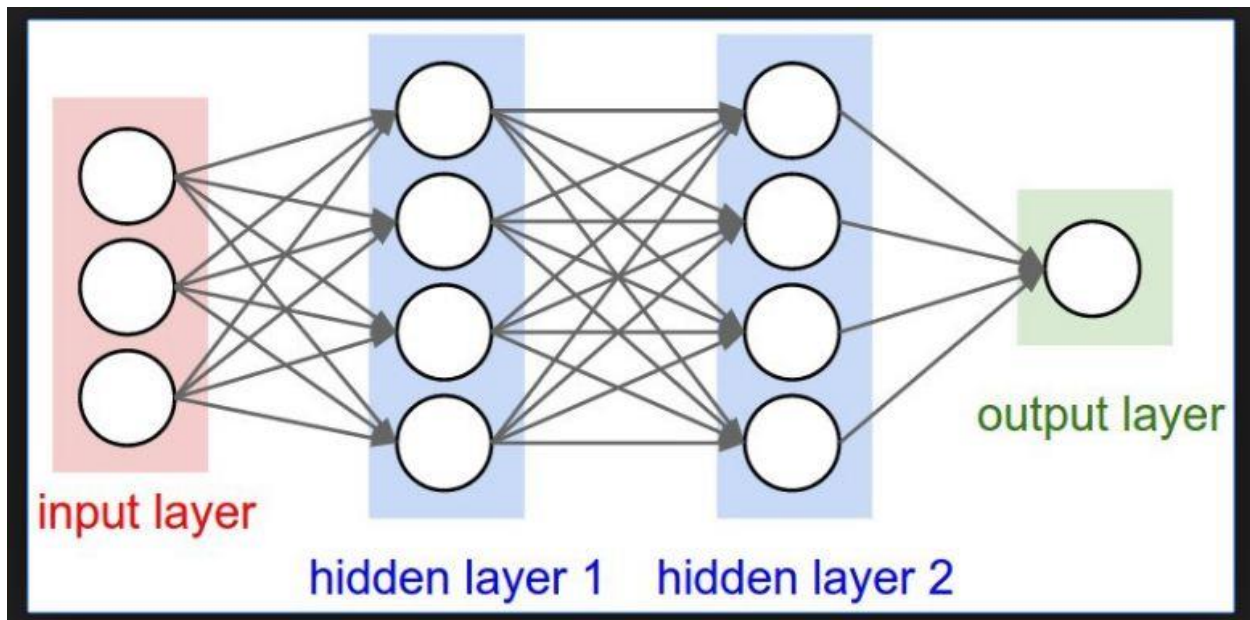
1.1.1. Mô hình Neural Network [2]

Neural Network (NN - mạng thần kinh) là một mô hình tính toán lấy cảm hứng từ cấu trúc và hoạt động của hệ thần kinh trong não bộ. Nó bao gồm một tập hợp các "nút" (**nodes**) hoặc "nơ-ron" (**neurons**) được kết nối với nhau để truyền tải và xử lý thông tin.

Neural Networks là sự kết hợp của những tầng nơ-ron được gọi là perceptron, nên còn gọi là perceptron đa tầng. Mỗi một **Neural Networks** thường bao gồm 3 kiểu tầng là:

- **Tầng input layer (tầng vào):** Tầng này nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng.
- **Tầng output layer (tầng ra):** Là tầng bên phải cùng và nó thể hiện cho những đầu ra của mạng.
- **Tầng hidden layer (tầng ẩn):** Tầng này nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng.

Mỗi nút trong lớp đầu vào đại diện cho một đặc trưng của dữ liệu đầu vào, trong khi các nút trong lớp đầu ra đưa ra kết quả hoặc dự đoán của mô hình. Các lớp ẩn giúp học và trích xuất các đặc trưng phức tạp từ dữ liệu đầu vào.



Hình 1.1 Kiến trúc Neural Networks [3]

Lưu ý: Mỗi một **Neural Networks** chỉ có duy nhất một tầng vào và một tầng ra nhưng lại có thể có nhiều tầng ẩn.

Các kết nối giữa các nút trong mạng thần kinh có trọng số (**weights**), giúp điều chỉnh cách thông tin được truyền qua mạng. Khi một dữ liệu mới được đưa vào mạng, thông tin sẽ truyền qua các lớp và qua các kết nối với trọng số tương ứng. Sau đó, mạng thần kinh tính toán các giá trị đầu ra dựa trên trọng số và các phép tính phi tuyến (**non-linear**), ví dụ như các hàm kích hoạt (**activation function**), để tạo ra kết quả cuối cùng.

Với **Neural Networks**, mỗi nút mạng là một nơron nhưng chúng có thể có hàm kích hoạt khác nhau. Thực tế, người ta thường sử dụng các nút có cùng loại hàm kích hoạt với nhau để việc tính toán thuận lợi hơn. Tại mỗi tầng, số lượng nút mạng vẫn có thể khác nhau tùy vào bài toán hoặc cách giải quyết.

Mạng thần kinh có khả năng học từ dữ liệu thông qua quá trình huấn luyện. Trong quá trình này, mô hình được cung cấp các ví dụ huấn luyện đã biết kết quả mẫu (còn gọi là nhãn của dữ liệu) từ đó điều chỉnh các trọng số của kết nối để tối ưu hóa hiệu suất của mô hình.

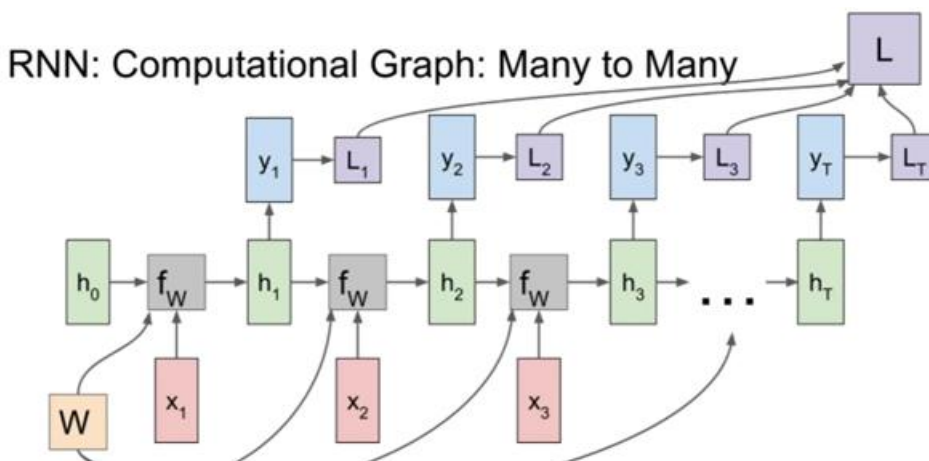
1.1.2. Mô hình Recurrent Neural Networks [4]

Recurrent Neural Networks (RNN) hay còn gọi là mạng hồi quy, là một lớp mô hình con của **NN**, tại đó kết nối giữa các nút sẽ tạo thành một đồ thị có hướng chạy dọc theo một trình tự thời gian. Điều này cho phép mạng có thể thể hiện các hành vi động tạm thời.

Phân loại **RNN**:

- **One to one**: mẫu bài toán cho **NN** và **Convolutional Neural Networks (CNN)**, có 1 đầu vào (**input**) và 1 đầu ra (**output**). Ví dụ như 1 ảnh đầu vào và đầu ra là 1 nhãn cho ảnh.
- **One to many**: mẫu bài toán có 1 **input** nhưng nhiều **output**. Ví dụ: bài toán tạo chú thích (caption) cho ảnh, khi đó **input** là 1 ảnh nhưng output là nhiều chữ mô tả cho ảnh đấy, có thể 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 khung ảnh (**frame**) tách ra từ video, còn **output** là hành động trong video.

- **Many to many**: bài toán có nhiều **input** và nhiều **output**. Ví dụ với bài toán dịch từ tiếng Anh sang tiếng Việt, **input** là 1 câu gồm nhiều chữ: “*I love Vietnam*” và **output** cũng là 1 câu gồm nhiều chữ “*Tôi yêu Việt Nam*”.



Hình 1.2 Mô hình RNN Many to many [5]

Ở các chương sau, khoá luận này sẽ tập trung vào mô hình **RNN** theo kiểu **Many to many**.

Mô hình **RNN** phù hợp với các dạng bài yêu cầu xử lý ngôn ngữ tự nhiên vì nó có những ưu điểm sau đây [6]:

- **Xử lý dữ liệu tuần tự**: Mô hình **RNN** được thiết kế đặc biệt để xử lý dữ liệu tuần tự, như văn bản, câu, hay chuỗi thời gian. Trong ngôn ngữ tự nhiên, các từ và câu thường có mối quan hệ và phụ thuộc vào nhau theo thứ tự. Ví dụ: "Tôi đang ăn bữa trưa" và "Bữa trưa đang được tôi ăn" có cùng nghĩa nhưng khác nhau về cấu trúc. Mô hình **RNN** có khả năng ghi nhớ thông tin từ các đầu vào trước đó và áp dụng chúng cho các đầu ra tiếp theo, từ đó giúp nắm bắt được sự phụ thuộc thời gian trong ngôn ngữ tự nhiên.
- **Độ dài biến đổi**: Mô hình **RNN** có thể xử lý các đầu vào có độ dài khác nhau, với khả năng làm việc với cả các đoạn văn ngắn và dài. Điều này rất hữu ích trong ngôn ngữ tự nhiên vì các câu và đoạn văn thường có độ dài không đồng đều.
- **Giải quyết vấn đề mất mát thông tin**: Trong ngôn ngữ tự nhiên, việc hiểu và làm sáng tỏ ý nghĩa của một từ hay câu thường phụ thuộc vào ngữ cảnh trước đó. Mô

hình **RNN** giúp giữ lại thông tin từ các đầu vào trước đó và sử dụng chúng trong quá trình dự đoán. Điều này giúp giảm thiểu vấn đề mất mát thông tin (**vanishing gradient problem**) – xảy ra khi giá trị độ dốc (**gradient**) ứng với đạo hàm của hàm chi phí tiến quá sát 0 – một vấn đề mà các mô hình truyền thẳng (**feedforward models**) gặp phải.

- **Tính linh hoạt và khả năng mô hình hóa:** Mô hình **RNN** có tính linh hoạt cao và có khả năng mô hình hóa nhiều loại ngôn ngữ tự nhiên khác nhau, từ dịch máy và nhận diện giọng nói đến phân loại và sinh văn bản. Mô hình **RNN** có thể được tinh chỉnh và điều chỉnh để phù hợp với các nhiệm vụ cụ thể.

Tuy nhiên, mô hình **RNN** cũng có một số hạn chế, như vẫn còn vấn đề mất mát thông tin trên các khoảng cách xa trong chuỗi, hay khó khăn trong việc xử lý thông tin không đồng nhất trong chuỗi và khó khăn trong việc học mối quan hệ dài hạn. Do đó, trong thực tế, các phiên bản nâng cấp cho mô hình **RNN** như **LSTM (Long Short Term Memory)** [7] thường được sử dụng để giải quyết những hạn chế này.

1.1.3. Siêu tham số

Dưới đây là các siêu tham số chính thường được sử dụng để hiệu chỉnh một mô hình học máy:

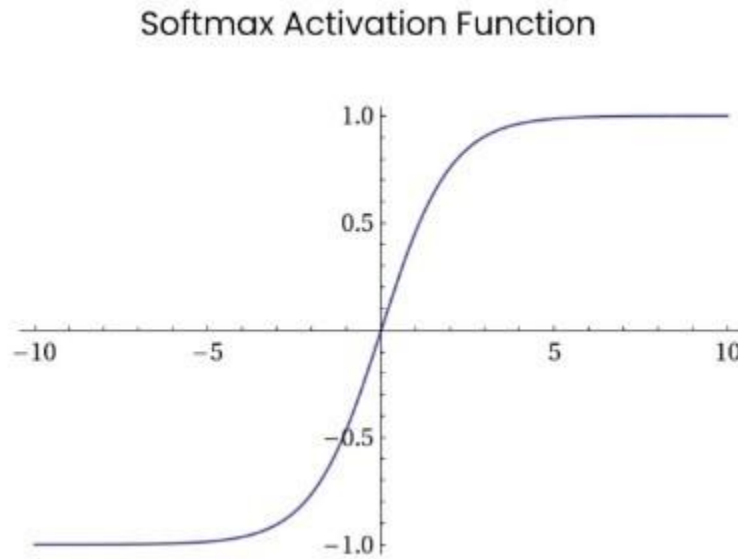
- **Learning rate** [8]: Tốc độ học là một siêu tham số sử dụng trong quá trình huấn luyện mô hình. Giá trị là một số dương, nằm trong khoảng 0 và 1. Tốc độ học giúp kiểm soát sự thay đổi các trọng số của mô hình trong khi tính toán tối ưu hoá theo phương pháp **gradient descent**.
- **Batch size**: là kích thước của mẫu trong một lần huấn luyện
- **Epoch**: là số lần mô hình học toàn bộ tập huấn luyện

1.1.4. Hàm kích hoạt Softmax [9]

Hàm **softmax**, hay còn được gọi là hàm mũ chuẩn hoá, được dùng để tính xác suất xảy ra của một lớp trong tổng số các lớp có thể xuất hiện. Cụ thể, hàm **softmax** biến véc-tơ **k** chiều có các giá trị thực bất kỳ thành véc-tơ **k** chiều có giá trị thực với tổng bằng 1. Giá trị của hàm **softmax** luôn nằm trong khoảng (0,1).

Dưới đây là công thức hàm **softmax**:

$$\sigma_i(z) = \frac{e^{z_i}}{\sum_j^K e^{z_j}}$$



Hình 1.3 Đồ thị của hàm Softmax [10]

1.1.5. Underfitting và Overfitting [11]

Đây là 2 vấn đề cần quan tâm giải quyết trong quá trình huấn luyện mô hình học máy, chúng giúp đảm bảo mô hình chạy tốt với dữ liệu hiện có lẫn dữ liệu trong tương lai (dữ liệu chưa thấy – *unseen data*):

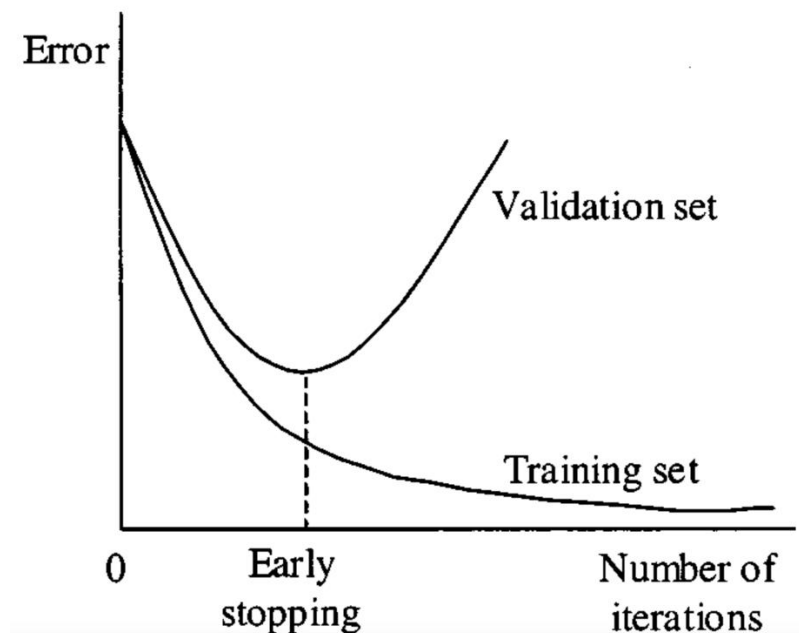
- **Underfitting (thiếu khớp)**: Là việc lựa chọn mô hình chưa được phù hợp với tập dữ liệu huấn luyện và các mẫu mới khi dự đoán. Nguyên do xảy ra **underfitting** thường là bởi mô hình chưa đủ độ phức tạp cần thiết để bao quát được tập dữ liệu.
- **Overfitting (quá khớp)**: Là hiện tượng mô hình quá khớp với tập dữ liệu huấn luyện, việc này sẽ gây ra hậu quả nghiêm trọng nếu tập dữ liệu huấn luyện xuất hiện nhiều. Mô hình **overfitting** chỉ chú trọng vào việc xấp xỉ với tập dữ liệu huấn luyện mà kém khả năng tổng quát hóa, làm cho mô hình này trở nên không thật sự tốt đối với dữ liệu nằm ngoài dữ liệu huấn luyện (dữ liệu kiểm thử và dữ liệu thực tế). **Overfitting** thường xảy ra khi độ phức tạp của mô hình quá lớn hoặc khi quá ít dữ liệu.

1.1.6. Phương pháp xử lý các vấn đề trong học máy

Có rất nhiều phương pháp xử lý các vấn đề kể trên, ví dụ như bằng cách thay đổi các siêu tham số hoặc điều chỉnh kiến trúc mô hình, chấp nhận hy sinh độ chính xác trong quá trình huấn luyện để giảm độ phức tạp của mô hình nhằm tránh được hiện tượng không mong muốn mà vẫn giữ được tính tổng quát của mô hình. Dưới đây là một số phương pháp xử lý cơ bản:

1.1.6.1. Early stopping

Early stopping [12] [13]: Khi ta dùng một phương pháp tối ưu hàm số để giảm thiểu giá trị mất mát thì **jtrain**, **jval** sẽ cùng giảm theo thời gian, nhưng nếu sau một thời gian **jval** tăng lên còn **jtrain** tiếp tục giảm thì đó là lúc bắt đầu dẫn đến **overfitting**. Cách đơn giản nhất để giảm thiểu **overfitting** đó là dừng huấn luyện tại ngay thời điểm bắt đầu **overfitting** và phương pháp này được gọi là **early stopping**. Phương pháp này có mục đích chủ yếu để nhận biết khi nào mô hình có dấu hiệu **overfitting** từ đó ta có thể xây dựng chiến thuật để xử lý. Việc phát hiện **overfitting** sẽ làm giảm thời gian lãng phí và công sức.



Hình 1.4 Mô tả Early stopping [14]

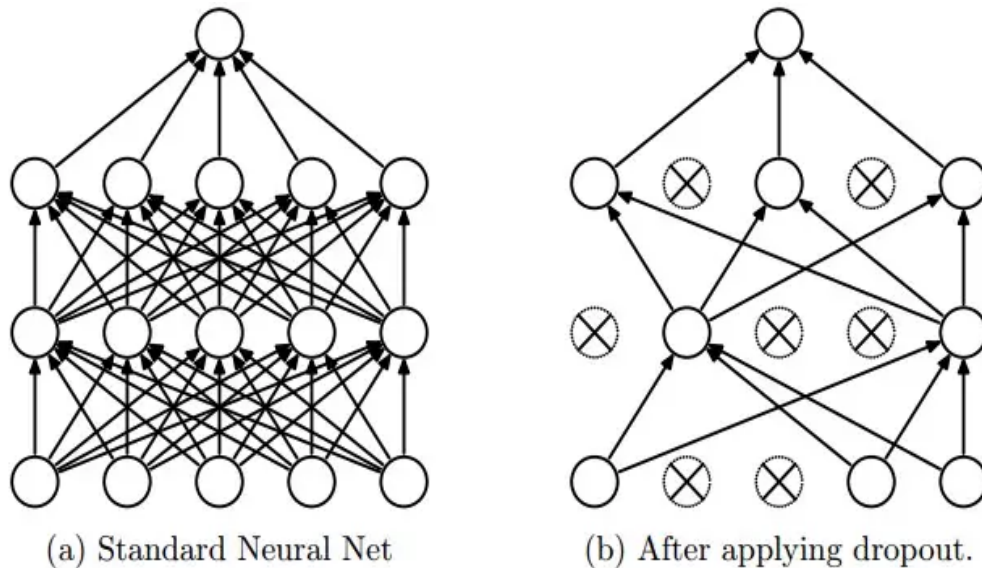
1.1.6.2. Data augmentation

Tăng cường dữ liệu (Data augmentation) [12] [15]: Bằng cách tạo thêm dữ liệu từ dữ liệu có sẵn, tăng cường dữ liệu giúp làm đa dạng dữ liệu giúp máy học đa nhiệm hơn, tổng quát hơn tránh *overfitting*.

- Giả sử có một tập dữ liệu nhưng số lượng quá ít sẽ khiến cho mô hình không học được các đặc trưng do dữ liệu đầu vào không đủ. Như vậy cần phải tìm cách làm tăng dữ liệu lên.
- Nếu đã có lượng lớn dữ liệu, nhưng những dữ liệu đó lại mất cân bằng. Có nghĩa là tồn tại sự chênh lệch số lượng đầu vào lớn giữa các lớp với nhau. Khi đó mô hình sẽ có xu hướng học kĩ các đặc trưng của những lớp mà có số lượng phần tử nhiều hơn, khiến mô hình xảy ra học tử, và những lớp này học kém hơn các lớp còn lại. Điều đó khiến mô hình đạt hiệu quả không tốt. Tìm cách để dữ liệu được cân bằng là cách giúp cho mô hình học được tốt hơn.
- Đối với bài toán xử lý ảnh: dẫn kích thước ảnh, dịch chuyển ảnh, tăng giảm độ sáng, thêm độ nhiễu cho ảnh, ...
- Đối với bài toán xử lý ngôn ngữ tự nhiên: Đổi chỗ các một vài cặp từ trong câu, xóa một từ bất kỳ trong câu, tìm thay thế một một từ trong câu với một từ đồng nghĩa tương ứng, ...

1.1.6.3. Dropout

Dropout [16]: Đây là một kĩ thuật hiệu chỉnh để chống lại vấn đề *overfitting*. Cách *dropout* thực hiện là tạm tắt bỏ một số đơn vị trong các bước huấn luyện. **Dropout** giúp quá trình học nhanh hơn đồng thời có thể giúp tránh *overfitting* tốt hơn.



Hình 1.5 Mô tả Dropout [17]

Cách hoạt động của **dropout**:

- Với một lớp của mạng **neuron**, **dropout** được áp dụng với một xác suất p cho trước (có thể sử dụng nhiều **dropout** khác nhau cho những layer khác nhau, nhưng trên 1 layer sẽ chỉ có 1 **dropout**).
- Tại mỗi bước trong quá trình huấn luyện, khi thực hiện truyền thẳng đến lớp sử dụng **dropout**, thay vì tính toán tất cả đơn vị ở trên lớp, tại mỗi đơn vị **dropout** sẽ thực hiện "giao xóc xúc" với xác suất p xem đơn vị đó được sử dụng (**active**) hay không được sử dụng (**deactive**). Những đơn vị được sử dụng sẽ tính toán bình thường còn với những đơn vị không được sử dụng sẽ đặt giá trị tại đơn vị đó bằng 0.
- Trong quá trình kiểm thử thì tất cả các đơn vị đều được sử dụng và mong muốn đầu ra của các đơn vị giống với đầu ra mong đợi trong quá trình huấn luyện.

1.1.7. Phương pháp tối ưu Adam optimizer [18]

Hàm tối ưu **Adam** (**Adaptive Moment Estimation**) là một trong những thuật toán tối ưu hóa phổ biến được sử dụng trong huấn luyện mạng **neuron sâu**. Nó là một phương pháp kết hợp giữa hai thuật toán tối ưu hóa khác là **RMSprop** (**Root Mean Square Propagation**) [19] và **Momentum**.

Adam kết hợp các lợi ích của **RMSprop** và **Momentum** để điều chỉnh tỉ lệ học tập (**learning rate**) dựa trên độ dốc của gradient cũng như mức độ thay đổi của **gradient** từng tham số. Ý tưởng chính của **Adam** là sử dụng một bước học tập độc lập cho mỗi tham số của mô hình, nhưng tỉ lệ học tập được điều chỉnh cho từng tham số dựa trên sự thay đổi của **gradient** cho tham số đó trong quá khứ.

Cụ thể, **Adam** tính toán hai giá trị chính là **moving average** của **gradient** (**momentum**) và **moving average** của **gradient** bình phương (**RMSprop**). Sau đó, nó sử dụng các giá trị này để tính toán giá trị thích ứng (**adaptive**) cho tỉ lệ học tập của từng tham số. **Adam** cũng có các tham số điều chỉnh như β_1 (**moving average** cho **gradient**), β_2 (**moving average** cho **gradient** bình phương), và θ (giá trị nhỏ được thêm vào để tránh chia cho 0).

Với công thức như sau:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2)$$

Trong đó β_1 và β_2 là hai siêu tham số thường được chọn: $\beta_1 = 0.9$ và $\beta_2 = 0.999$, m là **momentum**, g là **gradient**.

Và công thức cập nhật θ như sau:

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}$$

trong đó α là tốc độ học (**learning rate**), ϵ là giá trị được thêm vào để tránh việc chia cho 0.

Tổng quát, hàm tối ưu **Adam** là một công cụ mạnh mẽ để cải thiện độ chính xác và tốc độ của các mô hình học máy. Cách tiếp cận dựa trên tỷ lệ học tập thích ứng và động lượng (**momentum**) của nó có thể giúp mạng **neuron** học nhanh hơn và hội tụ nhanh hơn đến tập hợp tham số tối ưu nhằm giảm thiểu hàm mất mát (**loss function**). [20]

1.2. Rasa [21]



Hình 1.6 Rasa [22]

Rasa là một nền tảng mã nguồn mở để xây dựng các ứng dụng trò chuyện thông minh. Nó cung cấp một bộ công cụ và thư viện cho việc phát triển các hệ thống trò chuyện **AI**, bao gồm cả **chatbot** và trợ lý ảo.

Cấu trúc các tệp của rasa bao gồm:

- **actions/__init__.py**: Một file rỗng giúp python định vị hành động của ta.
- **actions/actions.py**: Tệp này được sử dụng để tạo các hành động tùy chỉnh. Trong trường hợp ta muốn gọi một máy chủ bên ngoài hoặc tìm nạp dữ liệu **API** bên ngoài, ta có thể xác định hành động của mình tại đây.
- **config.yml**: File này định nghĩa cấu hình của **Natural Language Understanding (NLU)** và **core model**. Nếu ta đang sử dụng bất kỳ mô hình nào bên ngoài mô hình **NLU**, ta phải xác định đường dẫn tại đây.
- **Pipeline**: một chuỗi các thành phần xử lý dữ liệu và tích hợp trong quá trình xử lý thông điệp người dùng. Mỗi thành phần trong pipeline đóng vai trò trong việc biến đổi và chuẩn bị dữ liệu cho các bước tiếp theo trong quá trình xử lý trò chuyện

- **Policies**: một chuỗi các thành phần xử lý dữ liệu và tích hợp trong quá trình xử lý thông điệp người dùng. Mỗi thành phần trong pipeline đóng vai trò trong việc biến đổi và chuẩn bị dữ liệu cho các bước tiếp theo trong quá trình xử lý trò chuyện
- **credentials.yml**: Tập này được sử dụng để lưu trữ thông tin đăng nhập để kết nối với các dịch vụ bên ngoài như Facebook Messenger, Slack, v.v.
- **data/nlu.md**: Trong tập này, chúng ta xác định ý định của mình (người dùng có thể yêu cầu bot làm gì?). Những ý định này sau đó được sử dụng trong đào tạo mô hình **NLU**.
- **data/stories.md**: Stories là cuộc trò chuyện mẫu giữa người dùng và bot dưới dạng ý định, phản hồi và hành động. Các câu chuyện Rasa là một dạng dữ liệu đào tạo được sử dụng để đào tạo các mô hình quản lý đối thoại của Rasa.
- **domain.yml**: Tập này liệt kê các ý định khác nhau (những điều ta mong đợi từ người dùng) với các phản hồi và hành động của bot mà nó có thể thực hiện.
 - + **intents**: khai báo các chủ ý của người dùng
 - + **actions**: khai báo các hành động thường quy ước là utter_nameaction
 - + **responses**: định nghĩa các action sẽ thực hiện nội dung được quy định
- **endpoints.yml**: Điều này xác định chi tiết để kết nối các kênh như Slack, FB messenger, v.v. để lưu trữ dữ liệu trò chuyện trong cơ sở dữ liệu trực tuyến như Redis, v.v.
- **models/<name>.tar.gz**: mô hình ban đầu, tất cả các mô hình được đào tạo được lưu trữ trong thư mục mô hình. Để đào tạo lại mô hình, chúng ta sử dụng lệnh đào tạo rasa.

1.2.1. Rasa NLU [23]

Rasa NLU (Natural Language Understanding) là thành phần xử lý ngôn ngữ tự nhiên của Rasa, được sử dụng để hiểu và phân tích các thông điệp người dùng. Nó giúp trích xuất yêu cầu (**intent**) và các thực thể (**entities**) từ văn bản đầu vào bằng tiếng Việt, cho phép hệ thống hiểu được ý định của người dùng và thông tin liên quan.

Các chức năng của **Rasa NLU** bao gồm:

- **Intent Recognition (Nhận dạng ý định)** [24]: **Rasa NLU** cho phép ta xác định ý định (*intent*) của người dùng từ văn bản đầu vào. Ý định thể hiện mục đích hoặc ý muốn của người dùng trong cuộc trò chuyện. Ví dụ, trong câu “*Tôi muốn đặt một cuốn sách*”, ý định có thể là “*Đặt sách*”.
- **Entity Extraction (Trích xuất thực thể)** [25]: **Rasa NLU** giúp trích xuất các thực thể (*entities*) từ văn bản người dùng. Thực thể là các thông tin cụ thể trong văn bản, như tên, địa chỉ, ngày tháng, số lượng, v.v. Ví dụ, trong câu “*Tôi muốn đặt một cuốn sách về lịch sử*”, thực thể có thể là “*sách*” và “*lịch sử*”.
- **Preprocessing (Tiền xử lý)**: **Rasa NLU** cung cấp các công cụ và kỹ thuật để tiền xử lý dữ liệu ngôn ngữ tự nhiên trước khi thực hiện nhận dạng ý định và trích xuất thực thể. Điều này bao gồm các quy trình như *tokenization* (phân tách thành từ), *stemming* (rút gọn từ), *stop word removal* (loại bỏ từ dừng), v.v.
- **Machine Learning Models (Mô hình học máy)**: **Rasa NLU** sử dụng các mô hình học máy để nhận dạng ý định và trích xuất thực thể từ dữ liệu huấn luyện. Các mô hình phổ biến được sử dụng trong **Rasa NLU** là mô hình máy học có giám sát như *Support Vector Machines (SVM)* [26], *Recurrent Neural Networks (RNN)*, và *Transformers* [27].
- **Training Data (Dữ liệu huấn luyện)**: Để xây dựng mô hình **Rasa NLU**, ta cần cung cấp dữ liệu huấn luyện. Đây là tập các câu huấn luyện có nhãn, trong đó mỗi câu được gắn kết với ý định và các thực thể tương ứng. Dữ liệu huấn luyện này được sử dụng để huấn luyện các mô hình nhận dạng ý định và trích xuất thực thể.

1.2.2. **Rasa Core** [23]

Rasa Core là thành phần quản lý luồng trò chuyện và quyết định hành động trong hệ thống trò chuyện. nó nhận đầu vào có cấu trúc từ **NLU** và dự đoán hành động tiếp theo tốt nhất bằng cách sử dụng mô hình xác suất như mạng *neural LSTM* thay vì câu lệnh if/else. **Rasa Core** sử dụng các chính sách (*policies*) để xác định hành động phù hợp dựa trên trạng thái hiện tại của cuộc trò chuyện và ý định của người dùng. Nó cho phép xây dựng các luồng trò chuyện phức tạp và điều chỉnh hành vi của hệ thống trò chuyện dựa trên phản hồi từ người dùng.

Các chức năng của **Rasa Core** bao gồm [28]:

- **Dialogue Management (Quản lý cuộc trò chuyện):** *Rasa Core* quản lý và điều khiển luồng cuộc trò chuyện giữa hệ thống trò chuyện và người dùng. Nó xác định trạng thái hiện tại của cuộc trò chuyện, lưu trữ thông tin và quyết định hành động tiếp theo dựa trên trạng thái và các thông điệp đầu vào.
- **Action Selection (Lựa chọn hành động):** *Rasa Core* xác định hành động cần thực hiện dựa trên trạng thái hiện tại của cuộc trò chuyện và các tùy chọn khác. Nó sử dụng các chính sách (*policies*) để quyết định hành động phù hợp dựa trên các quy tắc và chiến lược xác định trước.
- **Policies (Chính sách):** *Policies* xác định cách *Rasa Core* quyết định hành động tiếp theo dựa trên trạng thái hiện tại và các tùy chọn khác. Có nhiều chính sách khác nhau có thể được sử dụng, bao gồm *Memoization* (ghi nhớ trạng thái và hành động trước đó), *Fallback* (xử lý các tình huống không rõ ràng hoặc không thể nhận dạng), *Rule-based* (quy tắc dựa trên các tình huống cụ thể), và *Machine Learning-based* (dựa trên mô hình học máy).

1.2.2.1. Pipelines [29]

Trong *Rasa*, *Pipeline* là một danh sách các thành phần được sử dụng để xử lý và trích xuất thông tin từ đầu vào ngôn ngữ tự nhiên và dự đoán *intent*, *entities*, và *actions*. Các thành phần trong pipeline hoạt động theo thứ tự từ trên xuống dưới, với đầu ra của thành phần trước được đưa vào làm đầu vào cho thành phần tiếp theo.

```

9  pipeline:
10     - name: whitespaceTokenizer
11     - name: RegexFeaturizer
12     - name: LexicalSyntacticFeaturizer
13     - name: EntitySynonymMapper
14     - name: ResponseSelector
15     | epochs: 100
16     - name: CountVectorsFeaturizer
17     | analyzer: char_wb
18     | min_ngram: 1
19     | max_ngram: 4
20     | epochs: 100

```

Hình 1.7 Pipelines phần 1

WhitespaceTokenizer: Đây là một *tokenizer* đơn giản, dùng để tách các từ trong câu dựa trên khoảng trắng. Ví dụ: “*Hello world*” sẽ được tách thành [“*Hello*”, “*world*”].

RegexFeaturizer: Đây là một phần tử được sử dụng để trích xuất đặc trưng từ văn bản sử dụng các biểu thức chính quy. Nó có thể sử dụng các biểu thức chính quy để tìm kiếm và đánh dấu các mẫu, ví dụ như đánh dấu các từ có chứa số, các từ in hoa, các từ có ký tự đặc biệt, v.v.

LexicalSyntacticFeaturizer: Đây là một phần tử để trích xuất đặc trưng từ thông tin từ vựng và cú pháp của câu. Nó có thể sử dụng các kỹ thuật như **POS tagging** (gán nhãn phần từ), **dependency parsing** (phân tích cú pháp phụ thuộc), **sentiment analysis** (phân tích cảm xúc) để tạo ra các đặc trưng dựa trên thông tin ngôn ngữ.

EntitySynonymMapper: Đây là một phần tử được sử dụng để ánh xạ các đồng nghĩa của các thực thể. Nó có thể dùng để gán nhãn đồng nghĩa cho các từ hoặc cụm từ có cùng ý nghĩa, giúp cải thiện quá trình hiểu và xử lý ngôn ngữ tự nhiên.

ResponseSelector: Đây là một phần tử được sử dụng để chọn phản hồi phù hợp từ một tập hợp các câu trả lời. Nó có thể dựa trên các đặc trưng từ văn bản đầu vào để dự đoán câu trả lời tốt nhất cho câu hỏi hoặc yêu cầu của người dùng. Trong trường hợp này, mô hình được huấn luyện trong 100 **epoch** để tối ưu hóa quá trình chọn phản hồi.

CountVectorsFeaturizer là một phần tử trong pipeline xử lý ngôn ngữ tự nhiên (NLP) được sử dụng để trích xuất đặc trưng từ văn bản đầu vào. Dưới đây là cấu hình chi tiết cho **CountVectorsFeaturizer**:

- **name:** Tên của phần tử **CountVectorsFeaturizer**.
- **analyzer:** Phương pháp phân tích (**analyzer**) được sử dụng để tách từ và xây dựng các **n-gram**. Trong trường hợp này, **char_wb** được sử dụng, nghĩa là từ sẽ được chia thành các ký tự và chỉ xây dựng **n-gram** nếu nó xuất hiện trong từ hoặc ở đầu hoặc cuối từ. Ví dụ: từ “example” sẽ được chia thành các ký tự [‘e’, ‘x’, ‘a’, ‘m’, ‘p’, ‘l’, ‘e’] và tạo ra các **n-gram** [‘e’, ‘x’, ‘a’, ‘m’, ‘p’, ‘l’, ‘e’, ‘ex’, ‘xa’, ‘am’, ‘mp’, ‘pl’, ‘le’, ‘exa’, ‘xam’, ‘amp’, ‘mpl’, ‘ple’, ‘exam’, ‘xamp’, ‘ampl’, ‘mple’].
- **min_ngram:** Số lượng tối thiểu của **n-gram** để xây dựng. Trong trường hợp này, giá trị là 1, cho nghĩa là chỉ xây dựng các ký tự đơn lẻ.

- **max_ngram**: Số lượng tối đa của **n-gram** để xây dựng. Trong trường hợp này, giá trị là 4, cho nghĩa là xây dựng các **n-gram** từ ký tự đơn lẻ đến 4 ký tự.
- **epochs**: Số lượng vòng lặp huấn luyện của **CountVectorsFeaturizer**. Trong trường hợp này, giá trị là 100.

```

59 - name: FallbackClassifier
60   threshold: 0.3
61   ambiguity_threshold: 0.1
62

```

Hình 1.8 Pipelines phần 2

FallbackClassifier là một thành phần được sử dụng để xử lý các trường hợp không chắc chắn hoặc mơ hồ trong việc phân loại **intent**. Thông qua việc đặt ngưỡng (**threshold**) và ngưỡng mơ hồ (**ambiguity threshold**), **FallbackClassifier** quyết định liệu một câu được xem là mơ hồ hay không và xử lý nó một cách phù hợp.

- **threshold**: 0.3: Đây là ngưỡng được sử dụng để xác định liệu một câu được phân loại intent hay không. Nếu xác suất phân loại intent cao hơn ngưỡng này, câu sẽ được coi là được phân loại thành công. Nếu xác suất phân loại **intent** thấp hơn ngưỡng này, **FallbackClassifier** sẽ được kích hoạt.
- **ambiguity_threshold**: 0.1: Đây là ngưỡng mơ hồ được sử dụng để xác định liệu một câu có mức độ mơ hồ cao hay không. Nếu xác suất phân loại intent tương đối gần nhau (chênh lệch nhỏ hơn ngưỡng mơ hồ), câu sẽ được xem là mơ hồ và **FallbackClassifier** sẽ thực hiện các hành động liên quan đến việc xử lý mơ hồ.

Khi một câu không được phân loại intent thành công hoặc được xem là mơ hồ, **FallbackClassifier** sẽ thực hiện các hành động như trả về **intent fallback** (**intent** dự phòng) hoặc thực hiện các hành động xử lý mơ hồ khác, ví dụ như yêu cầu người dùng cung cấp thêm thông tin hoặc giúp đỡ.

1.2.2.2. Policies [30]

Trong **Rasa**, **Policies** là các thành phần quyết định trong việc xử lý hoạt động của trợ lý ảo. Chúng xác định cách trợ lý ảo phản hồi và tương tác với người dùng dựa trên dữ liệu huấn luyện và các quy tắc được định nghĩa trước.

```

65 policies:
66     - name: MemoizationPolicy
67     - name: TEDPolicy
68       max_history: 5
69       epochs: 130
70     - name: RulePolicy

```

Hình 1.9 Policies

MemoizationPolicy: Đây là một chính sách đơn giản dựa trên việc nhớ các hành động đã được thực hiện trước đó. Nó lưu trữ các đường dẫn hành động trước đó trong bộ nhớ và thực hiện các hành động tương ứng nếu tìm thấy một tình huống tương tự trong lịch sử tương tác. Chính sách này hữu ích trong việc xử lý các trường hợp đã được huấn luyện một cách rõ ràng và cần phản hồi nhanh chóng.

TEDPolicy: Đây là một chính sách sử dụng mô hình học sâu **Transformer Embedding Dialogue (TED)** để dự đoán intent và hành động dựa trên lịch sử tương tác. **max_history** là số lượng câu trước đó được sử dụng để xây dựng bộ nhớ cho mô hình **TED**. **epochs** chỉ định số lần huấn luyện mô hình. **constrain_similarities** là một cờ để hạn chế độ tương đồng của các vector nhúng (**embeddings**) được sử dụng trong mô hình.

RulePolicy: Đây là một chính sách dựa trên quy tắc được định nghĩa trước để xác định các hành động dựa trên mẫu và điều kiện. Quy tắc được định nghĩa theo cú pháp của Rasa và được sử dụng để phân loại intent và thực hiện hành động tương ứng.

Các thành phần **Policies** này được sắp xếp theo thứ tự ưu tiên từ trên xuống dưới trong pipeline xử lý của **Rasa**. Khi một câu được nhận diện, **Rasa** sẽ thực hiện các chính sách theo thứ tự để xác định hành động phù hợp để trả lời hoặc tương tác với người dùng.

1.3. Microsoft SQL Server [31]



Hình 1.10 Microsoft SQL Server [32]

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (**Relational Database Management System - RDBMS**) do Microsoft phát triển và cung cấp. Nó là một trong những hệ quản trị cơ sở dữ liệu phổ biến và mạnh mẽ được sử dụng trong nhiều ứng dụng và môi trường kinh doanh.

1.4. Ngrok [33]



Hình 1.11 Ngrok [34]

Ngrok là một công cụ proxy công cộng và phần mềm **tunnel** mã nguồn mở. Nó cho phép ta tạo ra một kết nối an toàn giữa máy cục bộ của ta và mạng Internet thông qua một kết nối ngược qua tường lửa (**reverse tunneling**). Điều này cho phép ta truy cập vào các

dịch vụ đang chạy trên máy cục bộ của ta từ bất kỳ đâu trên **Internet** mà không cần cầu hình phức tạp hoặc thiết lập các quy tắc tường lửa.

1.5. Facebook Developer [35]



Hình 1.12 Facebook developers [36]

Facebook Developer là một nền tảng và tập hợp các công cụ, tài liệu và tài nguyên được cung cấp bởi Facebook để hỗ trợ các nhà phát triển xây dựng ứng dụng và tích hợp với các sản phẩm của Facebook. Nền tảng này cung cấp các **API (Application Programming Interface)** và **SDK (Software Development Kit)** cho phép nhà phát triển tạo ra các ứng dụng, tích hợp tính năng của Facebook vào ứng dụng của mình và sử dụng dịch vụ của Facebook như đăng nhập bằng tài khoản Facebook, chia sẻ nội dung, gửi tin nhắn, phân tích dữ liệu và quảng cáo.

Facebook Developer cung cấp các công cụ và tài nguyên để giúp nhà phát triển xây dựng ứng dụng chất lượng, tương tác với cộng đồng phát triển, và tìm hiểu về các hướng dẫn và tiêu chuẩn của Facebook. Các nhà phát triển có thể tạo ứng dụng di động, ứng dụng web, **chatbot**, trò chơi và nhiều ứng dụng khác sử dụng các công nghệ của Facebook.

Facebook Developer cũng cung cấp một hệ thống quản lý ứng dụng, cho phép nhà phát triển đăng ký, quản lý và quản lý các ứng dụng của mình trên nền tảng Facebook.

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ CHỨC NĂNG

Chương này là một phần quan trọng trong khoá luận. Nội dung chương tập trung vào việc phân tích và thiết kế các chức năng của trợ lý ảo, từ đó đưa ra các yêu cầu về chức năng và các yêu cầu nghiệp vụ. Mục đích của chương phân tích thiết kế chức năng là định nghĩa các chức năng cần thiết để sản phẩm hoạt động đúng theo yêu cầu của khoá luận. Chương này cũng giúp tạo ra một bản thiết kế chi tiết về các chức năng cần có, bao gồm cả các trường hợp sử dụng cụ thể và cách thức hoạt động của các chức năng đó.

2.1. Mô tả bài toán

Hằng năm, mỗi sinh viên mới nhập học tại trường đại học Thăng Long đều được phát một cuốn hướng dẫn học tập, trong đó chứa nội dung quan trọng là chương trình đào tạo của từng ngành. Chương trình đào tạo (CTĐT) mỗi ngành bao gồm danh sách các môn học được phân chia vào các học phần khác nhau như: Giáo dục đại cương, Giáo dục chuyên nghiệp: Học phần thuộc cơ sở khối ngành, Học phần bắt buộc, Học phần lựa chọn và Thực tập, Khoá luận tốt nghiệp, Chuyên đề tốt nghiệp.

Việc có nhiều học phần trong mỗi chương trình đào tạo của từng chuyên ngành học dẫn đến việc sinh viên mới nhập học dễ bị rối và khó có thể hình dung và lên được kế hoạch học tập trong tương lai. Mặt khác, việc chưa hiểu rõ một số quy tắc đăng ký học cũng dẫn đến sinh viên thường phải liên tục liên lạc với cố vấn học tập trong giai đoạn đăng ký. Việc này khiến cho cố vấn học tập bị quá tải khi trả lời yêu cầu, cũng như sinh viên khó có thể nhận được kịp thời các giải đáp.

Khoá luận hướng tới xây dựng một trợ lý ảo có chức năng thay thế cho một cố vấn học tập, dựa vào các học phần đã có và chương trình học của mỗi khối ngành. Trợ lý ảo cần có khả năng trả lời các câu hỏi liên quan đến vấn đề lộ trình học, tra cứu thông tin môn học, giải đáp thắc mắc sinh viên liên quan đến việc ĐKH.

Sinh viên hay người sử dụng hệ thống có thể chỉ thị cho trợ lý ảo giải đáp các thắc mắc như cung cấp lộ trình học của năm học hay năm học và kỳ học cho trước, thông tin môn học cần tra cứu, danh sách điều kiện tiên quyết và đề xuất danh sách môn học tiếp theo. Trợ lý ảo sẽ nhận thắc mắc từ phía sinh viên hay người sử dụng, phân loại chúng vào các yêu cầu (*Intent*) để từ đó đưa ra các phản hồi (*Actions*) tương ứng.

Dưới đây là mô tả một chương trình học cụ thể của ngành Công nghệ thông tin khoa 32 tại trường Đại học Thăng Long [37]:

KHUNG CHƯƠNG TRÌNH ĐÀO TẠO NGÀNH CÔNG NGHỆ THÔNG TIN

1. Giáo dục đại cương

Mã học phần	Tên học phần	Số tín chỉ	Điều kiện tiên quyết	Số giờ	Hệ số
ML111	Triết học Mác - Lênin	2		30	1.0
ML112	Kinh tế chính trị và CNXH khoa học	3	ML111	45	1.0
ML202	Tư tưởng Hồ Chí Minh	2	ML112	30	1.0
ML203	Đường lối cách mạng của Đảng CS Việt Nam	3	ML202	45	1.0
CS101	Công dân số	2		30	1.3
MA101	Logic, suy luận toán học và kỹ thuật đếm	3		27LT+36BT	1.3
CS100	Tin đại cương	2		18LT+24TH	1.5
NA151	Khoa học môi trường	2		30	1.0
EC101	Kinh tế học đại cương	3		45	1.2
VL101	Tiếng Việt thực hành	2		30	1.0
SH131	Pháp luật đại cương	2		30	1.0
GE101	Tiếng Anh sơ cấp 1	2		54	1.5
GE102	Tiếng Anh sơ cấp 2	2	GE101	54	1.5
GE103	Tiếng Anh sơ cấp 3	2	GE102, GE101 \geq 5	54	1.5
GE201	Tiếng Anh sơ trung cấp 1	2	GE103, GE102 \geq 5	54	1.5
GE202	Tiếng Anh sơ trung cấp 2	2	GE201, GE103 \geq 5	54	1.5
GE205	Tiếng Anh sơ trung cấp 3	2	GE202, GE201 \geq 5	54	1.5
GE301	Tiếng Anh trung cấp 1	2	GE205, GE202 \geq 5	54	1.5
GE303	Tiếng Anh trung cấp 2	2	GE301, GE205 \geq 5	54	1.5
GE305	Tiếng Anh trung cấp 3	2	GE303, GE301 \geq 5	54	1.5
GF101	Tiếng Pháp 1	2		54	1.5
GF102	Tiếng Pháp 2	2	GF101	54	1.5
GJ101	Tiếng Nhật 1	2		54	1.5
GJ102	Tiếng Nhật 2	2	GJ101	54	1.5
GZ101	Tiếng Trung 1	2		54	1.5
GZ102	Tiếng Trung 2	2	GZ101	54	1.5
GI101	Tiếng Ý 1	2		54	0.75
GI102	Tiếng Ý 2	2	GI101	54	0.75
GK101	Tiếng Hàn 1	2		54	1.5
GK102	Tiếng Hàn 2	2	GK101	54	1.5
PG100	Giáo dục thể chất	4		100	1.4
PG121	Giáo dục quốc phòng	4		36LT+60TH	1.2

* Sinh viên chọn tiếng Pháp 1+2 hoặc tiếng Nhật 1+2, hoặc tiếng Trung 1+2, hoặc tiếng Ý 1+2, hoặc tiếng Hàn 1+2.

2. Giáo dục chuyên nghiệp

2.1. Các học phần thuộc cơ sở khối ngành

Mã học phần	Tên học phần	Số tín chỉ	Điều kiện tiên quyết	Số giờ	Hệ số
MA104	Số và cấu trúc đại số	3	MA101	27LT+36BT	1.5
MA110	Giải tích 1	3	MA101	27LT+36BT	1.5
MA111	Giải tích 2	3	MA110	27LT+36BT	1.5
MA120	Đại số tuyến tính	3	MA101	27LT+36BT	1.5
MA231	Xác suất thống kê ứng dụng	4	MA120, CS101	45LT+30TH	1.6
MI201	Toán rời rạc	3	CS122	45	1.6
CF212	Cấu trúc dữ liệu	3	CS122	27LT+36TH	1.6
CS110	Kỹ thuật số	2	MA101	18LT+24BT	1.5
CS121	Ngôn ngữ lập trình	3	CS100	27LT+36TH	1.6
CS122	Lập trình hướng đối tượng	3	CS121	27LT+36TH	1.6
CS212	Kiến trúc máy tính	3	CS110, CS122	45	1.6
CS315	Nguyên lý hệ điều hành	3	CS212	45	1.6
IS222	Cơ sở dữ liệu	3	CS121, MA104	45	1.6
NW212	Mạng máy tính	2	CS212	18LT+24BT	1.5

2.2. Các học phần bắt buộc của ngành

Mã học phần	Tên học phần	Số tín chỉ	Điều kiện tiên quyết	Số giờ	Hệ số
IS314	Hệ thống thông tin	3	IS222 hoặc IS223	45	1.6
IS322	Hệ quản trị cơ sở dữ liệu	3	IS222 hoặc IS223	45	1.6
IS330	Dữ liệu lớn	2	IS222 hoặc IS223	18LT+24TH	1.6
IS332	Phân tích thiết kế hướng đối tượng	3	CS122, IS222 hoặc IS223	45	1.6
IS334	Quản lý dự án hệ thống thông tin	3	IS314	45	1.6
MI322	Trí tuệ nhân tạo và công nghệ tri thức	3	MI201, CF212	45	1.6
SE302	Công nghệ phần mềm	2	IS332	18LT+24BT	1.5
CS314	Lập trình ứng dụng di động	2	CS122	18LT+24TH	1.6
IT320	Lập trình Python	3	MA120, CF212	45	1.6
IT332	Internet of Things	2	SE302	18LT+24TH	1.6
IT333	Công nghệ Web	3	NW212	45	1.6
IT380	Dự án Công nghệ thông tin	2	IS314	18LT+24TH	1.6

2.3. Các học phần lựa chọn của ngành: Chọn ≥ 6 tín chỉ trong các học phần sau

Mã học phần	Tên học phần	Số tín chỉ	Điều kiện tiên quyết	Số giờ	Hệ số
CF211	Phân tích và thiết kế thuật toán	2	CS121	18LT+24BT	1.6
CS223	Lập trình Java	3	CS122	45	1.6
CS224	Lập trình .Net	3	IS222 hoặc IS223	45	1.6
CS320	Học máy	3	MA231 hoặc MA230	27LT+36TH	1.6
CS325	Lập trình PHP	3	IS222 hoặc IS223	45	1.6
IS324	Phân tích dữ liệu	3	IS322	45	1.6
IS424	Lập trình cơ sở dữ liệu	3	IS322	45	1.6
SE312	Kiểm thử và đảm bảo chất lượng phần mềm	3	SE302	45	1.6
MI312	Đồ họa	2	CS122, MA120	18LT+24TH	1.6
MI414	Giao diện người máy	2	MI312	18LT+24BT	1.6

2.4. Thực tập, Khóa luận tốt nghiệp (KLTN) và Chuyên đề tốt nghiệp (CDTN)

Mã học phần	Tên học phần	Số tín chỉ	Điều kiện tiên quyết	Số giờ	Hệ số
IP404	Thực tập ngành CNTT	2	100 tín chỉ	60TH	1.3
IT499	KLTN ngành CNTT	6	120 tín chỉ	30LT+120TH	3.0
Sinh viên không làm KLTN chọn 1 trong 2 CDTN sau					
IS484	CDTN: Cơ sở dữ liệu	6	110 tín chỉ, IS322	30LT+120TH	2.5
SE487	CDTN: Phát triển phần mềm	6	110 tín chỉ, SE302	30LT+120TH	2.5

3. Lựa chọn tự do: Chọn thêm ≥ 6 tín chỉ trong các học phần được giảng ở Trường

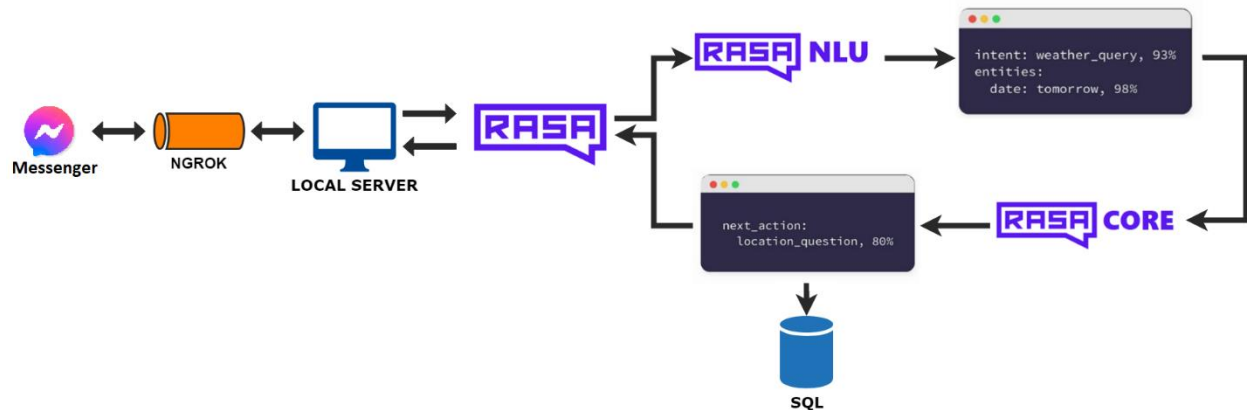
* Giải thích về điều kiện tiên quyết: Đã tích lũy đủ số tín chỉ và đạt điều kiện dự thi kết thúc học phần có mã nêu trong điều kiện tiên quyết mới được đăng ký học học phần tương ứng.

Hình 2.1 Khung chương trình đào tạo ngành Công nghệ thông tin khoá 32

2.2. Sơ đồ kiến trúc tổng thể

2.2.1. Sơ đồ

Kiến trúc tổng thể của hệ thống được minh họa như hình dưới đây:



Hình 2.2 Sơ đồ kiến trúc tổng thể [38]

2.2.2. Mô tả sơ đồ

Sinh viên (**người dùng**) truy cập Messenger của Facebook thực hiện thao tác gửi tin nhắn, thông qua *Ngrok* gửi một request đến server tại đây *server* nhận *request* và chuyển tin đến *Rasa server* thông qua *Rasa NLU* nhận diện được yêu cầu (*intent*) của người dùng hay từ bây giờ sẽ gọi là ý định khi xác định được ý định (*intent*) thông qua *Rasa Core* sẽ thực hiện thao tác lựa chọn hành động (*Actions Selection*) phản hồi về phía sinh viên thông qua Messenger.

2.3. Yêu cầu nghiệp vụ

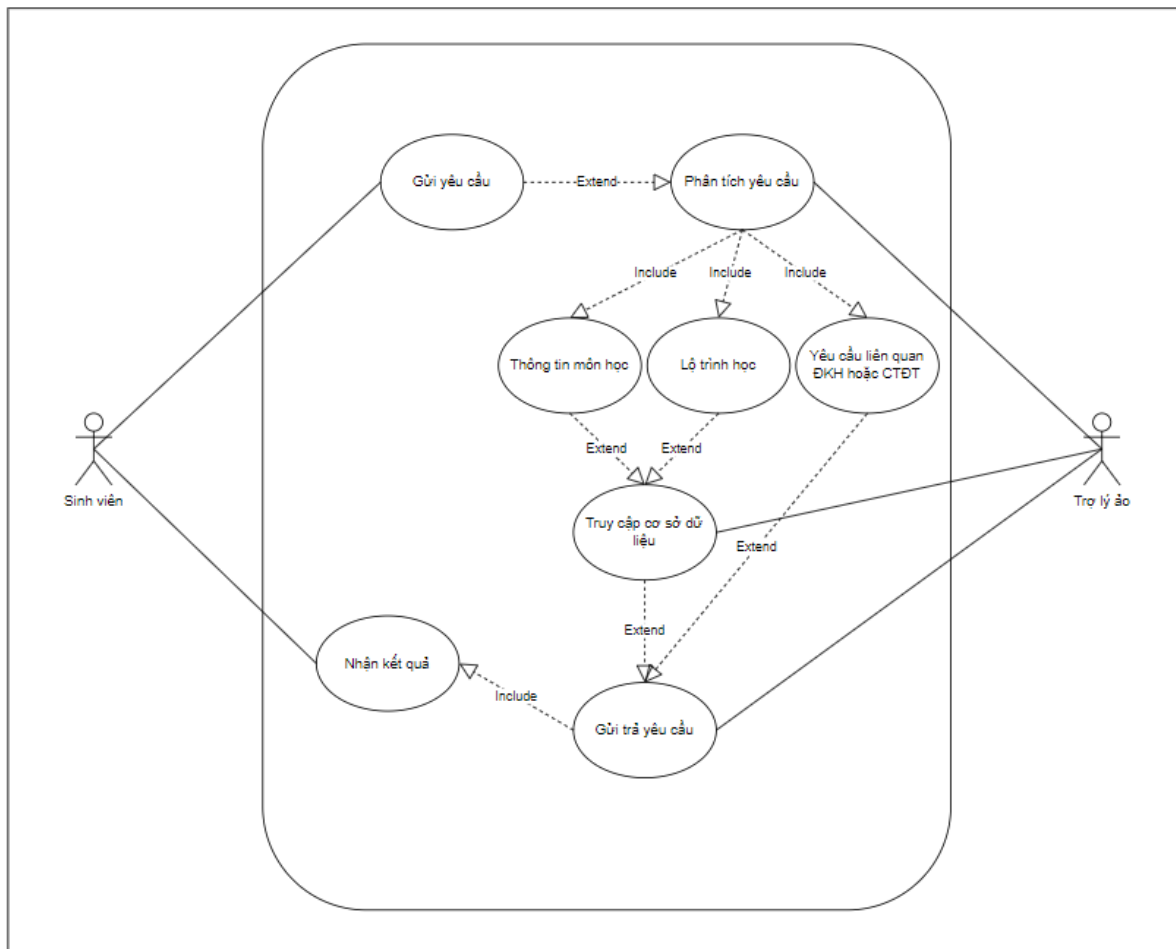
Trợ lý ảo mà ta xây dựng sẽ cần đáp ứng được những yêu cầu nghiệp vụ sau:

- Phân loại được yêu cầu (*intent*) của người dùng.
- Dựa vào yêu cầu (*intent*) đã phân loại được phản hồi (*actions*) lại phía người dùng kết quả ứng với yêu cầu (*intent*) của người dùng.
- Nếu yêu cầu không hợp lại hoặc không xác định được yêu cầu trả về “Vui lòng liên hệ trực tiếp với cố vấn học tập”.

2.4. Sơ đồ Usecase các chức năng chính

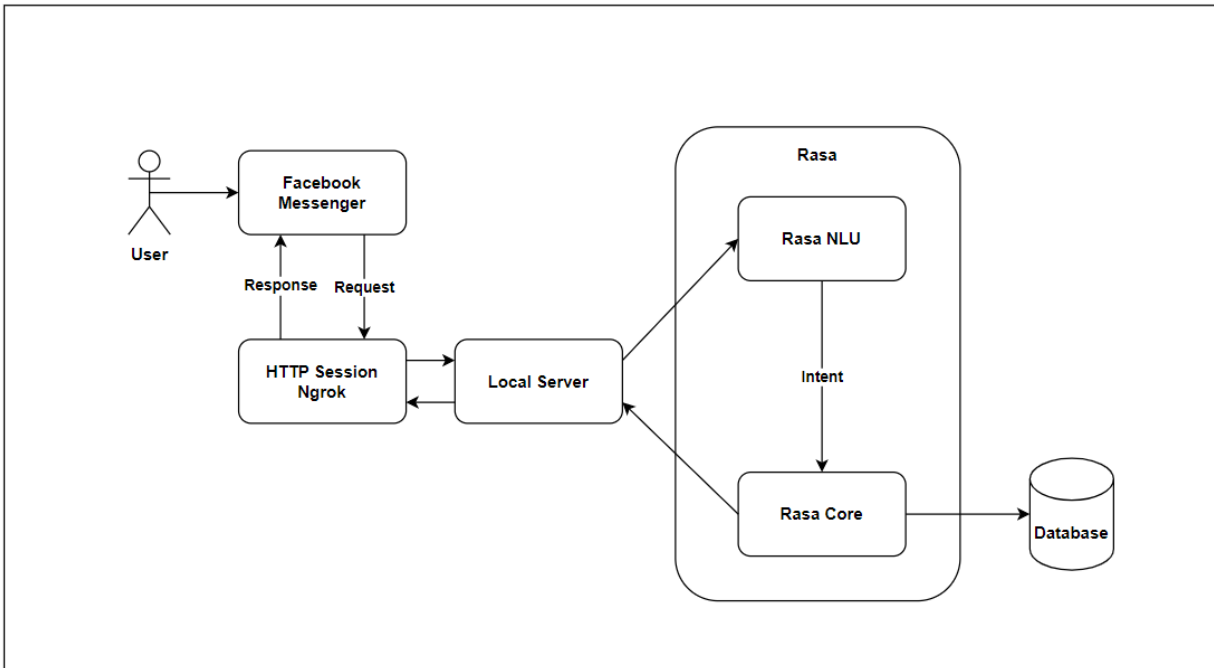
2.4.1. Sơ đồ Usecase

Hình ảnh minh họa cho Usecase:



Hình 2.3 Mô hình Usecase

Minh hoạ quy trình hoạt động của trợ lý ảo:



Hình 2.4 Mô tả quy trình hoạt động của trợ lý ảo

2.4.2. Các tác nhân có trong hệ thống

Tác nhân	Vai trò
Sinh viên (<i>User</i>)	<p>Tác nhân có vai trò sử dụng các chức năng chính của hệ thống:</p> <ul style="list-style-type: none"> - Truy cập vào Facebook Messenger - Gửi yêu cầu (Intent) cho trợ lý ảo
Trợ lý ảo (<i>ChatBot</i>)	<p>Tác nhân có vai trò phản hồi người dùng:</p> <ul style="list-style-type: none"> - Truy cập CSDL và trích xuất dữ liệu - Phản hồi (Actions) người dùng tương ứng với yêu cầu (Intent)

Bảng 2.1 Các tác nhân trong hệ thống

2.5. Đặc tả nhóm chức năng

2.5.1. UC1 Yêu cầu thông tin môn học

UC1		Yêu cầu thông tin môn học	Độ phức tạp: Cao
Mô tả		User tiến hành đặt yêu cầu về thông tin môn học với trợ lý ảo CVHT	
Tác nhân		User	
Tiền điều kiện		Nhập thông tin môn học cần tra cứu	
Hậu điều kiện	Thành công	Trợ lý ảo phản hồi chi tiết thông tin môn học được yêu cầu	
	Lỗi	Trợ lý ảo yêu cầu User kiểm tra lại thông tin môn học	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
<div>Luồng chính</div> <ul style="list-style-type: none">- User gửi một yêu cầu thông tin môn học kèm theo tên môn học đến trợ lý ảo CVHT- Trợ lý ảo xác nhận yêu cầu từ User- Phân loại yêu cầu của User là thông tin môn học- Trợ lý ảo sau khi phân loại được yêu cầu thì tiến hành trích xuất dữ liệu tên môn học từ yêu cầu của User- Trợ lý ảo kiểm tra tên môn học vừa nhận được- Nếu hợp lệ truy cập Database trích xuất thông tin môn học và phản hồi thành công, nếu không thì tiến hành luồng phát sinh bên dưới			
Luồng sự kiện phát sinh			
Nhập sai tên môn học / Không nhập tên môn học			
<ul style="list-style-type: none">- Trợ lý ảo CVHT yêu cầu User kiểm tra lại thông tin môn học- User có thể yêu cầu lại thông tin môn học với tên môn học đã nhập đúng			

Bảng 2.2 Usecase yêu cầu thông tin môn học

2.5.2. UC2 Yêu cầu thông tin môn học với điều kiện tiên quyết

UC2		Yêu cầu thông tin môn học có thể đăng ký với điều kiện cung cấp	Độ phức tạp: Cao
Mô tả		User tiến hành gửi yêu cầu với những điều kiện cung cấp thì sẽ có thể ĐKH những môn học gì	
Tác nhân		User	
Tiền điều kiện		Nhập thông tin điều kiện	
Hậu điều kiện	Thành công	Trợ lý ảo phản hồi lại những môn học có thể ĐKH	
	Lỗi	Yêu cầu kiểm tra lại thông tin	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
Luồng chính: <ul style="list-style-type: none">- User gửi yêu cầu đến CVHT- CVHT sẽ phân loại yêu cầu và trích xuất dữ liệu từ yêu cầu của User- CVHT kiểm tra thông tin đã trích xuất được- Nếu hợp lệ thực hiện truy cập Database lấy thông tin các môn học có điều kiện tương ứng với điều kiện do User cung cấp phản hồi thông tin các môn học có thể ĐKH, nếu không hợp lệ thực hiện luồng phát sinh			
Luồng sự kiện phát sinh			
Nhập sai thông tin điều kiện <ul style="list-style-type: none">- User nhập thông tin sai dẫn đến CVHT không thể truy xuất dữ liệu từ Database- CVHT yêu cầu User kiểm tra lại thông tin- User có thể gửi lại yêu cầu với thông tin chính xác			

Bảng 2.3 Usecase yêu cầu thông tin môn học với điều kiện tiên quyết

2.5.3. UC3 Yêu cầu thông tin lộ trình học

UC3		Yêu cầu thông tin lộ trình học gợi ý cho sinh viên	Độ phức tạp: Cao
Mô tả		User gửi yêu cầu lộ trình học của năm hoặc của kỳ cụ thể của năm cụ thể tới CVHT	
Tác nhân		User	
Tiền điều kiện		Thông tin năm học hoặc kỳ học năm học cần tra cứu	
Hậu điều kiện	Thành công	Phản hồi danh sách môn học	
	Lỗi	Yêu cầu User kiểm tra lại thông tin cung cấp	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
Luồng chính <ul style="list-style-type: none">- User gửi yêu cầu cho CVHT- CVHT phân loại xác định loại yêu cầu của User và tiến hành trích xuất dữ liệu năm học hoặc kỳ học năm học từ yêu cầu- CVHT kiểm tra thông tin vừa trích xuất- Nếu hợp lệ tiến hành truy cập Database dựa vào năm học và kỳ học lấy ra danh sách các môn học sau đó phản hồi lại phía User- Nếu không tiến hành luồng phát sinh			
Luồng sự kiện phát sinh			
Nhập sai năm học / Nhập kỳ học nhưng không có năm học <ul style="list-style-type: none">- User cung cấp kỳ học nhưng không có năm học dẫn đến CVHT không thể biết kỳ học của năm học nào nên không thể phản hồi- CVHT phản hồi yêu cầu User kiểm tra lại thông tin			

Bảng 2.4 Usecase yêu cầu lộ trình học

2.5.4. UC4 Yêu cầu giải đáp một số vấn đề liên quan ĐKH hoặc CTĐT

UC4		Yêu cầu giải đáp một số vấn đề liên quan ĐKH	Độ phức tạp: Trung bình
Mô tả		User yêu cầu CVHT gửi đáp một số vấn đề liên quan CTĐT hoặc ĐKH	
Tác nhân		User	
Tiền điều kiện		Không có	
Hậu điều kiện	Thành công	CVHT phản hồi giải đáp thắc mắc của User	
	Lỗi	Không có	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
Luồng chính <ul style="list-style-type: none">- User yêu cầu CVHT giải đáp thắc mắc- CVHT phân loại yêu cầu giải đáp và dựa vào yêu cầu để phản hồi giải đáp User			

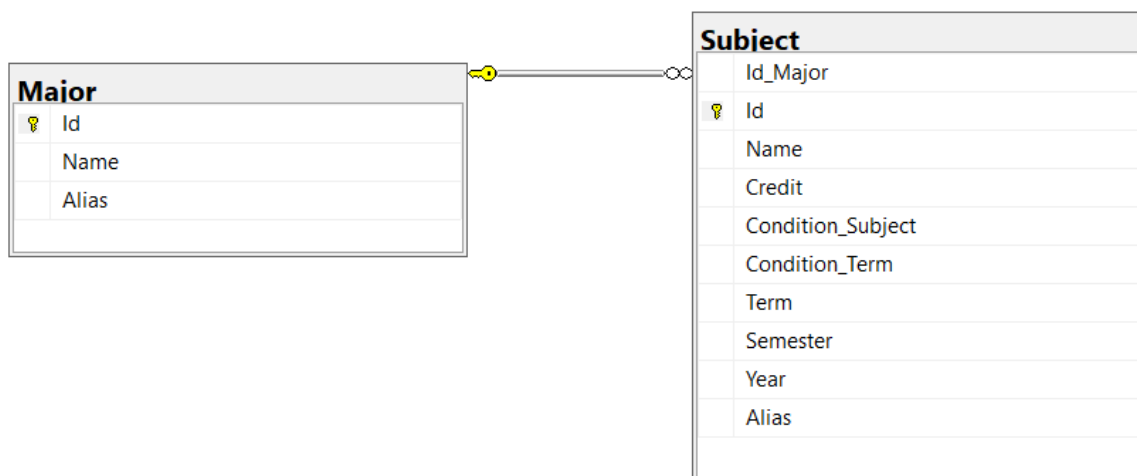
Bảng 2.5 Usecase yêu cầu giải đáp một số vấn đề liên quan ĐKH hoặc CTĐT

CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ DỮ LIỆU HUẤN LUYỆN

Chương này tập trung vào việc thiết kế cơ sở dữ liệu phù hợp và tạo dữ liệu huấn luyện để phục vụ cho việc xây dựng các mô hình máy học. Mục đích của chương 3 là đưa ra các bước cần thiết để thiết kế cơ sở dữ liệu và tạo dữ liệu huấn luyện, từ đó đảm bảo độ chính xác và hiệu quả cho các mô hình học máy.

3.1. Thiết kế cơ sở dữ liệu

3.1.1. Lược đồ quan hệ của cơ sở dữ liệu



Hình 3.1 Hình ảnh minh họa cho lược đồ của cơ sở dữ liệu

Ta xây dựng một cơ sở dữ liệu bao gồm 2 bảng: bảng Major chứa các thông tin về các chuyên ngành ví dụ: Công nghệ thông tin, Khoa học máy tính,... và bảng Subject sẽ lưu trữ các thông tin liên quan đến môn học ví dụ: Học máy, Lập trình hướng đối tượng,...

Vì một môn học có thể có trong nhiều chuyên ngành ví dụ như các môn học đại cương, và một số môn học phân cơ sở của ngành Công nghệ thông tin và Khoa học máy

tính cũng có phần tương đồng, ngoài ra nhiều chuyên ngành còn cũng có một môn học. Do đó, kết nối giữa 2 bảng này là kết nối nhiều-nhiều hay còn gọi là kết nối **M : N**.

3.1.2. Danh sách các bảng

STT	Tên bảng	Mô tả
1	Major	Danh sách lưu thông tin các ngành học
2	Subject	Danh sách lưu thông tin các môn học

Bảng 3.1 Mô tả các bảng có trong cơ sở dữ liệu

3.1.3. Chi tiết các bảng

3.1.3.1. Major

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	Id	NVARCHAR(255)	Mã định danh của chuyên ngành
2	Name	NVARCHAR(255)	Tên của chuyên ngành
3	Alias	NVARCHAR(255)	Bí danh của chuyên ngành

Bảng 3.2 Mô tả các thành phần trong bảng Major

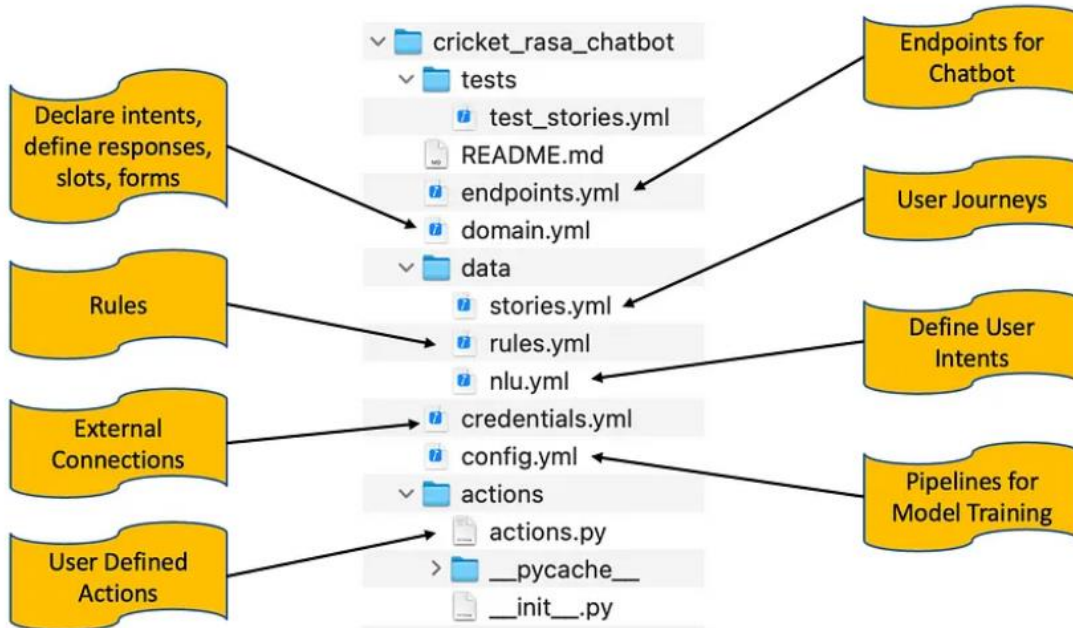
3.1.3.2. Subject

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	Id_Major	NVARCHAR (255)	Mã định danh của chuyên ngành
2	Id	NVARCHAR (255)	Mã môn học
3	Name	NVARCHAR(255)	Tên môn học
4	Credit	INT	Số tín chỉ của môn học
5	Condition_Subject	NVARCHAR(255)	Môn học điều kiện tiên quyết
6	Condition_Term	INT	Số tín chỉ tiên quyết
7	Term	INT	Mã số loại học phần
8	Semester	INT	Kỳ học gợi ý nên học

9	Year	INT	Năm học gợi ý nên học
10	Alias	NVARCHAR(255)	Bí danh môn học

Bảng 3.3 Mô tả các thành phần trong bảng Subject

3.2. Tạo dữ liệu huấn luyện



Hình 3.2 Cấu trúc thư mục Data [39]

Giới thiệu **Folder Data** trong **Rasa**:

- **nlu.yml**: File này bao gồm các thành phần sau:
 - + **intent**: ý định hay còn gọi là loại yêu cầu của **User** trong **Machine Learning** có thể hiểu là tương tự như **Class** hay **Nhãn**.
 - + **examples**: nơi ta liệt kê các loại câu hỏi của intent tức nhiều hình thức câu từ khác nhau nhưng đều mang chung một ý nghĩa.
- **stories.yml**:
 - + **story**: định nghĩa câu chuyện, đoạn hội thoại mà ta muốn trợ lý ảo đi theo hay có thể hiểu là kịch bản được viết ra để điều khiển câu chuyện.

- + **steps**: các bước của kịch bản được xây dựng từ trước.
- + **intent**: ý định của người dùng.
- + **actions**: trợ lý ảo trả lời.
- **rules.yml**: các luật được chúng ta đặt ra bắt buộc trợ lý ảo phải tuân theo khi gặp một intent cụ thể của **User**.

Tiếp theo chúng ta tìm hiểu đến một thành phần quan trọng khác của **Rasa** trong file **domain.yml**:

- **Actions**: Liệt kê các **custom actions** - một thành phần quan trọng khác mà ta sẽ nói đến ở phần sau.
- **Intent**: Liệt kê các intent của User.
- **Response**: Trợ lý ảo thực hiện phản hồi.
- **Slots**: là các vị trí lưu trữ thông tin trong **chatbot**. Mỗi slot tương ứng với một kiểu dữ liệu cụ thể như ngày tháng, chuỗi ký tự hoặc số. Khi **chatbot** tương tác với người dùng, các giá trị entities được trích xuất sẽ được gán vào các slots tương ứng. Việc lưu trữ các giá trị **entities** trong slots giúp **chatbot** duy trì thông tin của người dùng trong suốt quá trình hội thoại, cho phép nó theo dõi và sử dụng các thông tin này để thực hiện các hành động phù hợp.
- **Entities**: là các thành phần quan trọng trong một câu truy vấn hoặc thông điệp, đại diện cho các giá trị cụ thể của dữ liệu mà chatbot cần trích xuất từ đó. **Entities** có thể là các thông tin như tên, địa chỉ, thời gian, số lượng, và nhiều hơn nữa. Khi ta xác định và trích xuất entities từ câu truy vấn, chatbot có thể sử dụng thông tin này để hiểu ý định của người dùng và cung cấp các phản hồi phù hợp.

Ở chúng ta sẽ xây dựng dữ liệu huấn luyện file **data/nlu.yml**.

Khoá luận này chúng ta có tất cả **928** dữ liệu huấn luyện tương ứng với **928** câu hỏi khác nhau có thể gặp và được chia tương ứng với các loại sau:

- Với các yêu cầu dạng chi tiết và thông tin một môn học có: **97** mẫu câu hỏi.
- Yêu cầu dạng lộ trình học tập có: **95** mẫu câu hỏi.
- Với yêu cầu dạng điều kiện học tập có: **100** mẫu câu hỏi.

- Với chào hỏi và tạm biệt, cảm ơn tương ứng với **2** và **8** mẫu câu.
- Với những yêu cầu liên quan đến phần chương trình học và đăng ký học gồm nhiều dạng khác nhau và có tổng là **626** mẫu câu hỏi tất cả tương ứng với **30** dạng yêu cầu thường gặp.

Sau đây sẽ là cụ thể một vài dữ liệu mẫu cho một *intent* cụ thể.

3.2.1. Dữ liệu cho Intent Yêu cầu thông tin môn học

Trong tập dữ liệu này chúng ta sẽ chú trọng vào tên môn học vì vậy ở đây tên môn học sẽ được lưu vào một *slot subject* để trợ lý ảo có thể biết được môn học là gì và từ đó truy xuất thông tin chính xác môn học được yêu cầu.

Ví dụ:

STT	Data	Slot
1	Thông tin [CĐTN: Phát triển phần mềm](subject)	subject – Text lưu chữ tên môn học
2	Thêm thông tin về chuyên đề [CĐTN: Cơ sở dữ liệu](subject) là gì?	subject
3	Tôi cần cụ thể chuyên đề [CĐTN: Cơ sở dữ liệu](subject)	subject
4	Mình cần thông tin chuyên đề [CĐTN: Phát triển phần mềm](subject)	subject

Bảng 3.4 Khung dữ liệu cho intent yêu cầu thông tin môn học

3.2.2. Dữ liệu cho Intent Yêu cầu thông tin môn học với điều kiện tiên quyết

Ở tập dữ liệu này chúng ta cần quan tâm đến danh sách các môn học vậy nên chúng ta cần lưu vào một danh sách ở đây sẽ là *slot* kiểu *list* và đặt tên là *list*. Ngoài ra ta vẫn cần quan tâm đến tín chỉ cũng là một thông tin quan trọng và sử dụng *slot* tên *credit* để lưu trữ.

Ví dụ:

STT	Data	Slot
1	Hoàn thành các môn học [Đồ họa](list), [An toàn thông tin](list), [An toàn mạng](list)	list – Một danh sách chứa các môn
2	[101 tín chỉ](credit) và qua các môn [Kiến trúc	credit – Text lưu trữ số tín

	máy tính](list) và [Nguyên lý hệ điều hành](list)	chỉ list
3	Tớ học môn [Học máy](list)	list
4	Đã qua môn [Học máy](list) và [90 tín](credit)	list credit
5	Tôi có thể học được những gì khi đã học các môn [Tiếng Trung 1](list), [Tiếng Ý 1](list), [Tiếng Pháp 2](list)?	list

Bảng 3.5 Khung dữ liệu cho intent yêu cầu thông tin môn học với điều kiện tiên quyết

3.2.3. Dữ liệu cho Intent Yêu cầu lộ trình học

Với dạng yêu cầu lộ trình học thông tin quan trọng nhất sẽ là năm học và ký học tương ứng với các **slot year** và **semester**.

Vì vậy dạng dữ liệu huấn luyện sẽ có dạng như bảng bên dưới:

Ví dụ:

STT	Data	Slot
1	Với năm [2](year) kỳ [2](semester), tôi cần chọn những môn học nào?	year – Text semester - Text
2	Cho tôi xin gợi ý về môn học phù hợp với năm [bốn](year) và kỳ [ba](semester) của tôi được không?	year – Text semester - Text
3	Năm [3](year) học gì?	year – Text
4	Tôi học [2 năm rưỡi](ysem) nên học gì?	ysem – Text lưu trữ năm và học kỳ
5	Tôi học được [nửa năm](ysem) nên học gì?	ysem

Bảng 3.6 Khung dữ liệu cho intent yêu cầu lộ trình học

3.2.4. Dữ liệu cho Intent: Yêu cầu giải đáp một số thắc mắc CTĐT hoặc ĐKH

Ngoài ra với dạng yêu cầu giải đáp thắc mắc chúng ta không phải luôn luôn gặp những dạng câu cần lưu trữ thông tin mà phải là toàn bộ câu hỏi sẽ là thông tin cần lưu ý.

Ví dụ cụ thể như bên dưới bảng sau.

Ví dụ:

STT	Data	Slot
1	Tại sao học phần [Lập trình hướng đối tượng](name_subject) không mở cửa đăng ký học cho sinh viên?	name_subject – Text lưu trữ tên môn học
2	Vì sao sinh viên không thể xem học phần A trong lịch học cá nhân dù đã đăng ký thành công?	
3	Vì sao sinh viên nhóm 1, 2 không được phép đăng ký học cùng nhóm 3 dù học phần này đã mở trong thời khóa biểu của nhóm 1, 2?	
4	Để tránh việc trễ hạn và sắp xếp tài chính, sinh viên nhóm 1, 2 khi học hoặc thi lại cùng nhóm 3 nên thực hiện đóng học phí và lệ phí thi lại các học phần theo quy định của trường.	
5	Sinh viên có điểm tổng kết học phần Tiếng Anh sơ cấp 2 (GE102) < 4, nhưng sau đó đã học, thi và được cấp chứng chỉ tiếng Anh quốc tế thì có được quy đổi điểm học phần tiếng Anh sơ cấp 1, 2 không?	

Bảng 3.7 Khung dữ liệu cho các intent yêu cầu giải đáp về CTH hoặc CTĐT

CHƯƠNG 4. XÂY DỰNG CÁC HÀNH ĐỘNG TRONG RASA

Chương 4 trình bày về khía cạnh xây dựng thuật toán và áp dụng công nghệ trí tuệ nhân tạo phát triển *chatbot* dựa trên *NLP (Natural Language Processing)*. Chương này sẽ tập trung vào việc xây dựng các hành động để *chatbot* có thể tương tác với người dùng một cách thông minh và tự động. Chương này sẽ đưa ra các bước cần thiết để xây dựng các hành động trong *Rasa*, từ đó đảm bảo *chatbot* có thể hiểu và xử lý các yêu cầu của người dùng một cách chính xác. Sau chót, chương 4 cũng sẽ mô tả cách thức xây dựng mô hình học máy để trích xuất ra các yêu cầu người dùng.

4.1. Giới thiệu chung

4.1.1. Custom Actions

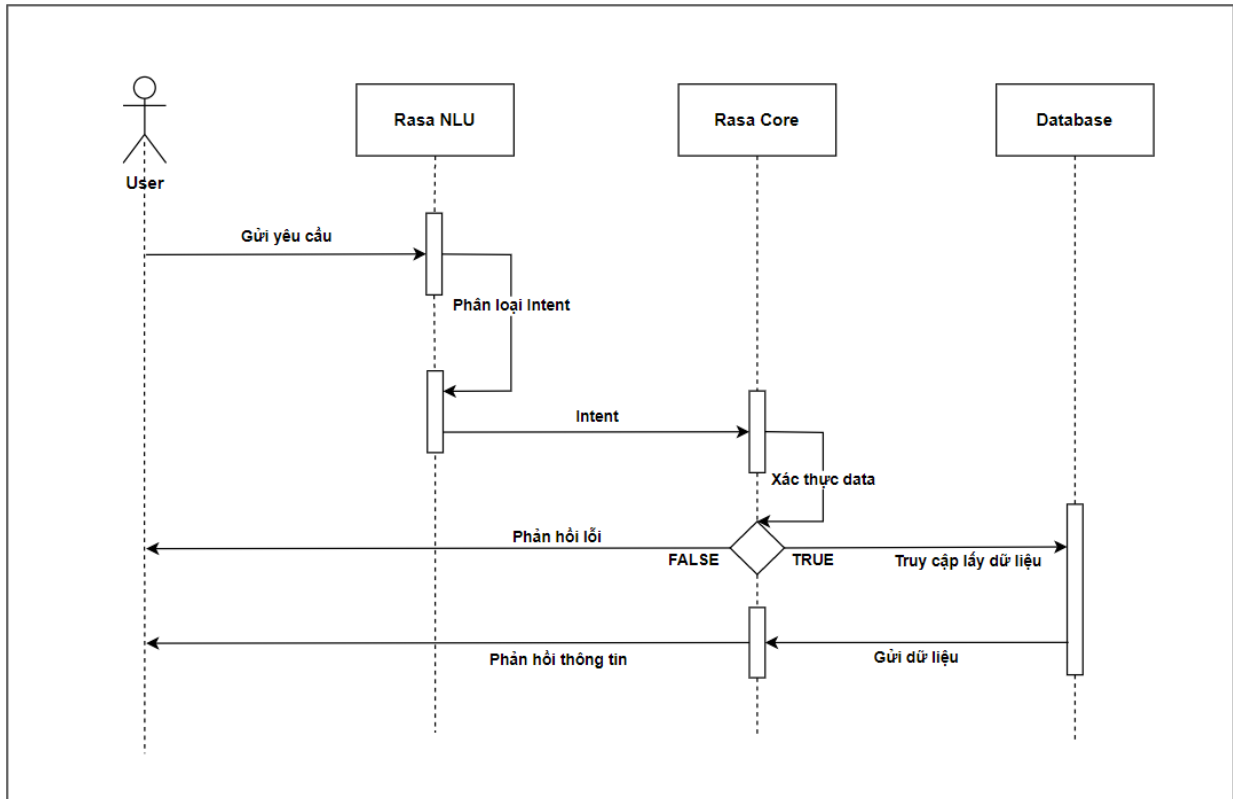
Hành động tùy biến (Custom Actions) trong *Rasa* là các thao tác mà lập trình viên có thể định nghĩa để trợ lý ảo sẽ thực hiện trong quá trình hội thoại. Khi trợ lý ảo nhận được thông điệp từ người dùng, nó sẽ sử dụng các “kinh nghiệm” thu được dữ liệu huấn luyện để từ đó đánh giá ý định trong thông điệp và trích xuất ra các *Entities*. Từ đây, trợ lý ảo có thể thực hiện một hoặc nhiều *Custom Actions* để cung cấp phản hồi phù hợp hoặc thực hiện các tác vụ khác nhau.

Custom Actions cho phép ta tương tác với các *API* bên ngoài, cơ sở dữ liệu hoặc bất kỳ hệ thống nào khác để lấy thông tin, cập nhật dữ liệu hoặc thực hiện hành động phức tạp. Ta có thể xử lý logic kinh doanh, truy vấn cơ sở dữ liệu, gọi *API* bên ngoài, gửi email và thực hiện nhiều tác vụ khác thông qua *Custom Actions*.

Custom Actions sẽ được định nghĩa trong file *actions/actions.py*.

4.1.2. Thiết kế chức năng

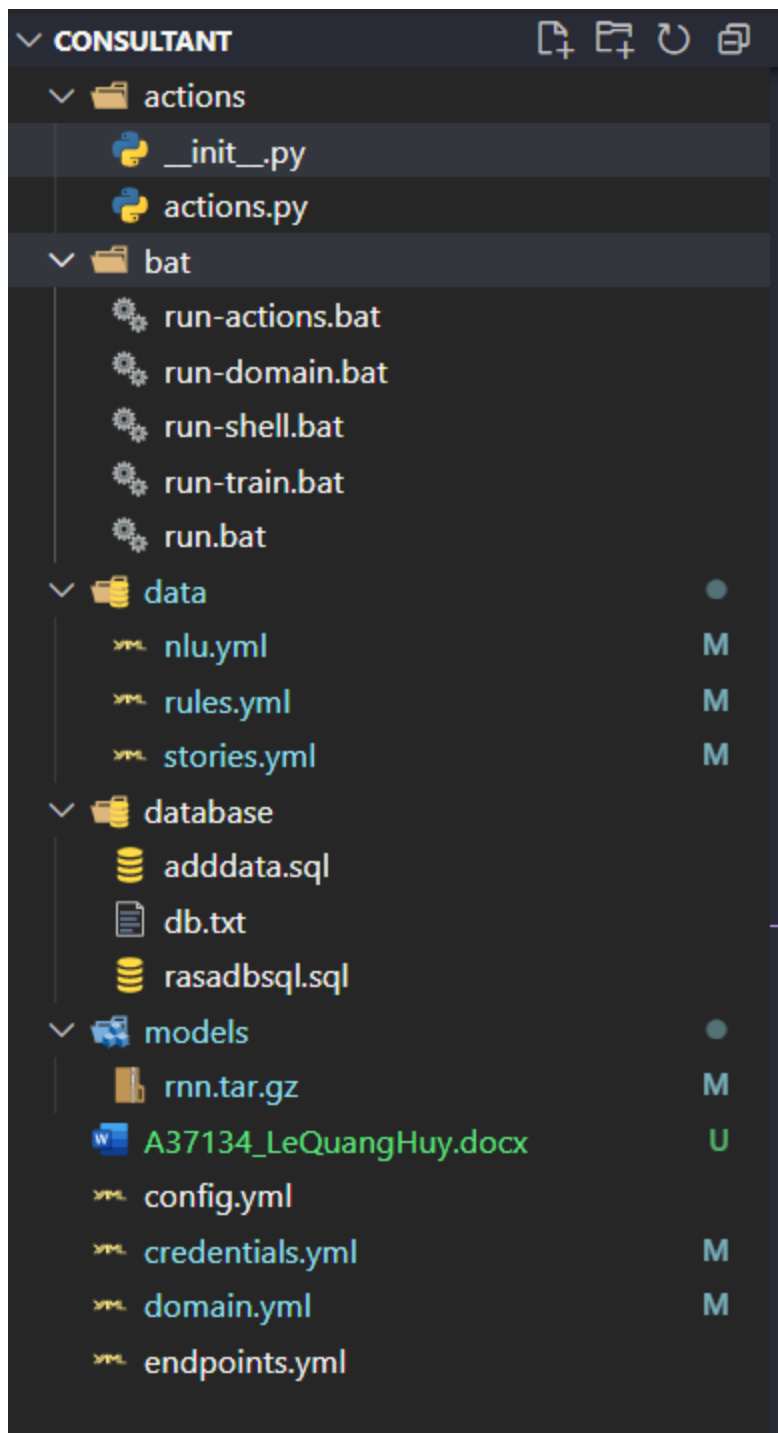
Sơ đồ tuần tự chung cho các chức năng của hệ thống là như sau:



Hình 4.1 Sơ đồ tuần tự chức năng giải đáp yêu cầu của User

4.2. Cấu trúc thư mục

Cấu trúc thư mục được tổ chức như sau:



Hình 4.2 Cấu trúc thư mục Rasa

4.3. Thuật toán trong Custom Actions

4.3.1. Mục tiêu

Chúng ta cần phải linh hoạt trong câu trả lời ví dụ như khi nhận câu “Xin chào” thì trợ lý sẽ trả lời “Chào bạn tôi có thể giúp gì được cho bạn?”. Đây là trường hợp cơ bản vì khi xác định được “Xin chào” là **Intent** greet câu phản hồi của trợ lý ảo có thể cố định ví dụ như trên. Nhưng trong trường hợp đặc biệt hơn như “Cho tôi thông tin môn Học máy” trợ lý ảo sẽ trả lời “Thông tin môn Học máy là...” nhưng với câu “Hãy cho tôi chi tiết môn Giải tích 1” câu trả lời sẽ nhận được là “Thông tin môn Giải tích 1 là...” như ta có thể thấy tên môn học ở đây là biến thay đổi tùy vào câu hỏi và thông tin môn học được lấy từ **Database** vậy câu phản hồi của trợ lý ảo trong trường hợp này là không thể để cố định.

Để giải quyết vấn đề này chúng ta đã biết đến khái niệm Custom Actions trong Rasa đã được giới thiệu ở trên.

4.3.2. Giải pháp

- Sử dụng Custom Actions trong Rasa
- Để giải quyết vấn đề trên và sử dụng Custom Actions chúng ta trước tiên cần:

4.3.2.1. Phân tích dữ liệu đầu vào và đầu ra

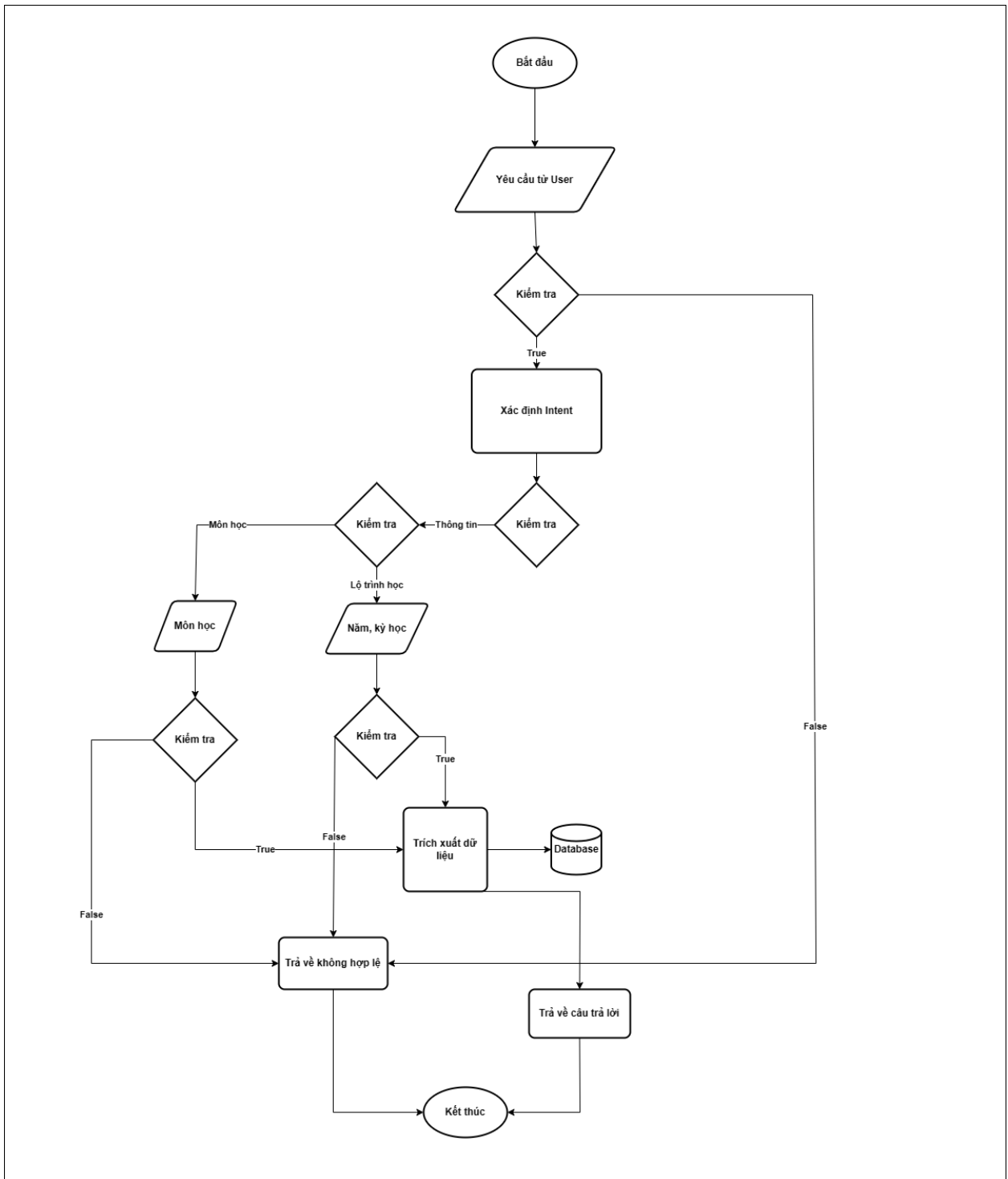
- **Đầu vào - Input:**
 - + **Yêu cầu của User (Intent):** Là câu yêu cầu đưa vào của User để dành cho CVHT, phân loại ra Intent của User
 - + **Dữ liệu từ yêu cầu của User:** Từ yêu cầu của User CVHT ta có được **Intent** sau đó sẽ lọc ra các Keyword và lấy ra dữ liệu từ yêu cầu của User
- **Đầu ra - Output:** Thông tin giải đáp cho yêu cầu của User.
 - + Thông tin có thể là môn học, giải đáp thắc mắc hoặc thông tin lộ trình học.

4.3.2.2. Thiết kế thuật toán

Mô tả thuật toán:

- **Bước 1:** Tiếp nhận yêu cầu từ **User**.

- **Bước 2:** Dự đoán *intent* của yêu cầu, nếu yêu cầu hợp lệ thì chuyển **bước 3** nếu không hợp lệ thì chuyển **bước 12**.
- **Bước 3:** Từ xác định được *intent* nếu là loại yêu cầu lộ trình học thì chuyển **bước 4**, dạng hỏi thông tin môn học chuyển **bước 7**, những dạng câu hỏi cần giải đáp thắc mắc chuyển **bước 9**.
- **Bước 4:** Từ yêu cầu loại ra các thông tin như năm học, kỳ học hoặc môn học.
- **Bước 5:** Kiểm tra năm học hợp lệ hoặc có năm học và kỳ học hợp lệ thì chuyển **bước 6** trong trường hợp sai chuyển **bước 12**.
- **Bước 6:** Truy cập cơ sở dữ liệu truy xuất dữ liệu tương ứng với lộ trình học yêu cầu chuyển **bước 11**.
- **Bước 7:** Kiểm tra thông tin môn học vừa được đề cập, nếu hợp lệ chuyển **bước 8** trường hợp còn loại chuyển **bước 12**.
- **Bước 8:** Truy cập cơ sở dữ liệu truy xuất dữ liệu tương ứng với thông tin môn học được yêu cầu chuyển **bước 11**.
- **Bước 9:** Với dạng yêu cầu là giải đáp một số thông tin khác cần xác định *intent*, chuyển **bước 10**.
- **Bước 10:** Dựa vào Intent đã xác định trả lời thắc mắc của *User*, chuyển **bước 11**.
- **Bước 11:** Từ các câu trả lời đã có được in ra màn hình cho *User*.
- **Bước 12:** In ra yêu cầu không hợp cho *User*.



Hình 4.3 Sơ đồ khối thuật toán nhận diện yêu cầu và trả về phản hồi

4.4. Xây dựng và đánh giá mô hình

4.4.1. Xây dựng mô hình

Như đã giới thiệu ở **Chương 1**, ta biết được rằng mô hình **RNN** có nhiều ưu điểm trong xử lý ngôn ngữ tự nhiên như: Xử lý dữ liệu tuần tự, dữ liệu có độ dài biến đổi, giải quyết vấn đề mất mát thông tin, tính linh hoạt và khả năng mô hình hóa. Chính từ các lý do này, khoá luận lựa chọn sử dụng mô hình **RNN** để xây dựng mô hình huấn luyện cho trợ lý ảo, là một ứng dụng liên quan tới xử lý ngôn ngữ tự nhiên với dữ liệu đầu vào là các chuỗi hội thoại.

Dưới đây là mô tả kiến trúc của mô hình **RNN** được xây dựng trong khoá luận:

```

model:
  type: tensorflow_embedding
  embedding_dim: 20
  hidden_layers_sizes: [[512, 256, 128]]
  rnn:
    unit_type: lstm
    num_units: 32
    direction: bidirectional
    dropout_rate: 0.2
    return_sequences: false
  dense_layer_sizes: [256, 128, 64]
  optimizer:
    type: adam
    learning_rate: 0.01
    decay: 0
    beta_1: 0.9
    beta_2: 0.999
    epsilon: 1e-8
  activation: softmax
  callbacks:
    - name: ModelCheckpoint
      filepath: ./model-rnn.h5
      monitor: val_loss
      save_best_only: true
    - name: EarlyStopping
      patience: 5
      monitor: val_loss
      restore_best_weights: true
      verbose: 0

```

Hình 4.4 Mô hình RNN sử dụng trong khoá luận

Mô hình trong **Rasa** có tên là **DIETClassifier** và được cấu hình với các tham số như sau:

- **epochs**: Số lượng vòng lặp huấn luyện của mô hình là 200.

- **batch_size**: Kích thước của mỗi batch được sử dụng trong quá trình huấn luyện là 64.
- **shuffle**: Dữ liệu huấn luyện sẽ được xáo trộn sau mỗi **epoch**.
- **validation_split**: Tỷ lệ phần trăm dữ liệu sẽ được sử dụng cho việc đánh giá mô hình trong quá trình huấn luyện là 20%.
- **random_seed**: Một giá trị được sử dụng để khởi tạo ngẫu nhiên và đảm bảo sự nhất quán của quá trình huấn luyện.
- **model_confidence**: Hàm **softmax** được sử dụng để tính toán độ tin cậy của các dự đoán của mô hình.

Mô hình được xây dựng bằng cách sử dụng kiến trúc "**tensorflow_embedding**" với các tham số sau:

- **embedding_dim**: Kích thước của không gian nhúng từ (**word embeddings**) là 20.
- **hidden_layers_sizes**: Kích thước của các tầng ẩn (**hidden layers**) là [512, 256, 128].
- **rnn**: mạng RNN sử dụng với các tham số sau:
 - + **unit_type**: Loại đơn vị **RNN** được sử dụng là **LSTM (Long Short-Term Memory)**.
 - + **num_units**: Số lượng đơn vị trong mỗi tầng **LSTM** là 32.
 - + **direction**: Mô hình sử dụng mạng **LSTM** song hướng (**bidirectional**) để học biểu diễn từ cả văn bản đầu vào và đầu ra.
 - + **dropout_rate**: Tỷ lệ **dropout** là 0.2, giúp tránh **overfitting** trong quá trình huấn luyện.
 - + **return_sequences**: Mô hình chỉ trả về kết quả ở output cuối cùng của **LSTM**.
 - + **dense_layer_sizes**: Kích thước của các tầng **fully connected (dense layers)** sau khi mạng **RNN** đã được thực hiện là [256, 128, 64].
 - + **optimizer**: Một thuật toán tối ưu hóa **Adam** được sử dụng với các tham số sau:

learning_rate: Tốc độ học của thuật toán **Adam** là 0.01.

decay: Hệ số giảm tốc độ học sau mỗi **epoch** là 0.

beta_1 và **beta_2**: Các hệ số điều chỉnh trong thuật toán **Adam** là 0.9 và 0.999 tương ứng.

+ **epsilon**: Một hằng số nhỏ được sử dụng để tránh chia cho 0.

activation: Hàm kích hoạt **softmax** được sử dụng cho tầng đầu ra của mô hình.

Mô hình được cấu hình với các **callbacks** sau:

- **ModelCheckpoint**: Lưu trữ trọng số của mô hình vào tệp tin **"./model-rnn.h5"** sau mỗi **epoch** nếu chỉ số **val_loss** (hàm mất mát trên tập **validation**) tốt hơn ở **epoch** trước đó.
- **EarlyStopping**: Dừng quá trình huấn luyện nếu không có sự cải thiện trong **val_loss** sau 5 **epoch** liên tiếp. Trọng số tốt nhất được khôi phục và sử dụng để đưa ra dự đoán cuối cùng.

4.4.2. Đánh giá mô hình

Sau khi training chúng ta sẽ đánh giá mô hình dựa theo các tiêu chí đánh giá như:

- **e_loss**: Là **entity loss**, đo lường mức độ sai số trong việc dự đoán các thực thể (**entities**) trong câu của người dùng. Đây là một hàm mất mát được sử dụng để đo lường sự sai khác giữa các thực thể được dự đoán và các thực thể chính xác trong dữ liệu huấn luyện. Mục tiêu là giảm thiểu **e_loss** để cải thiện độ chính xác của việc dự đoán thực thể.
- **i_loss**: Là **intent loss**, đo lường mức độ sai số trong việc dự đoán **intent** (ý định) của người dùng. Đây cũng là một hàm mất mát được sử dụng để đo lường sự sai khác giữa các **intent** được dự đoán và các **intent** chính xác trong dữ liệu huấn luyện. Mục tiêu là giảm thiểu **i_loss** để cải thiện độ chính xác của việc dự đoán **intent**.
- **t_loss**: Là **total loss**, tổng hợp các mất mát bao gồm **e_loss** và **i_loss**. Trong quá trình huấn luyện, mô hình cố gắng giảm thiểu tổng hợp các mất mát này để cải thiện độ chính xác tổng thể của việc dự đoán cả **intent** và **entity**.

- ***i_acc***: Là ***intent accuracy***, đo lường tỷ lệ dự đoán đúng ***intent*** so với tổng số câu trong tập dữ liệu huấn luyện. Nó đo lường hiệu suất của mô hình trong việc xác định đúng ***intent*** của người dùng.
- ***e_f1***: Là ***entity F1 score***, đo lường độ chính xác của việc dự đoán các thực thể (***entities***) của người dùng dựa trên ***F1 score***. ***F1 score*** là một phép đo tổng hòa giữa độ chính xác (***precision***) và độ bao phủ (***recall***) của việc dự đoán thực thể.

Kết quả quá trình huấn luyện thu được như sau:

```

)) to a dense tensor of unknown shape. This may consume a large amount of memory.
"shape. This may consume a large amount of memory." % value)
Epochs: 100%|██████████| 200/200 [45:23<00:00, 13.62s/it, t_loss=1.49, i_loss=0.0471, e_loss=0.051, i_acc=1, e_f1=0.992]
2023-07-29 20:10:23 DEBUG rasa.engine.storage.local_model_storage - Resource 'train_DIETClassifier6' was requested for writing.
2023-07-29 20:19:23 DEBUG rasa.engine.storage.local_model_storage - Resource 'train_DIETClassifier6' was persisted.

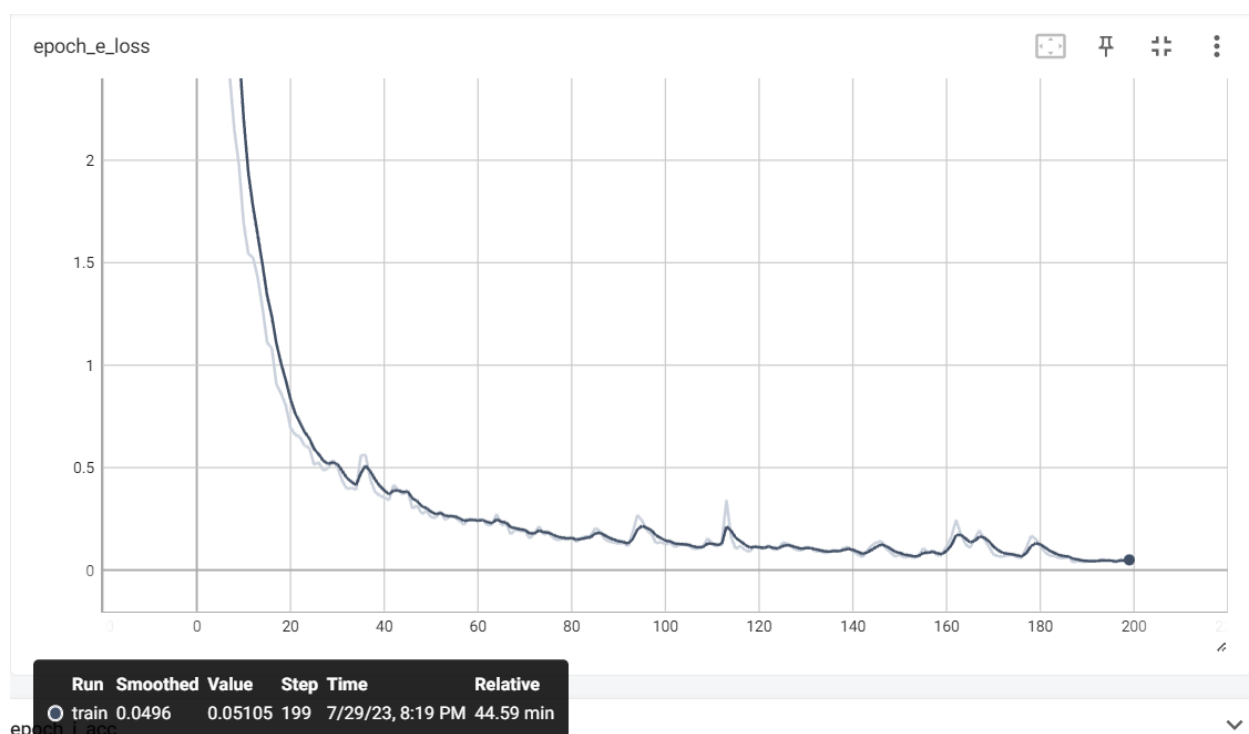
```

Hình 4.5 Kết quả huấn luyện của mô hình

Các tiêu chí đánh giá lần lượt:

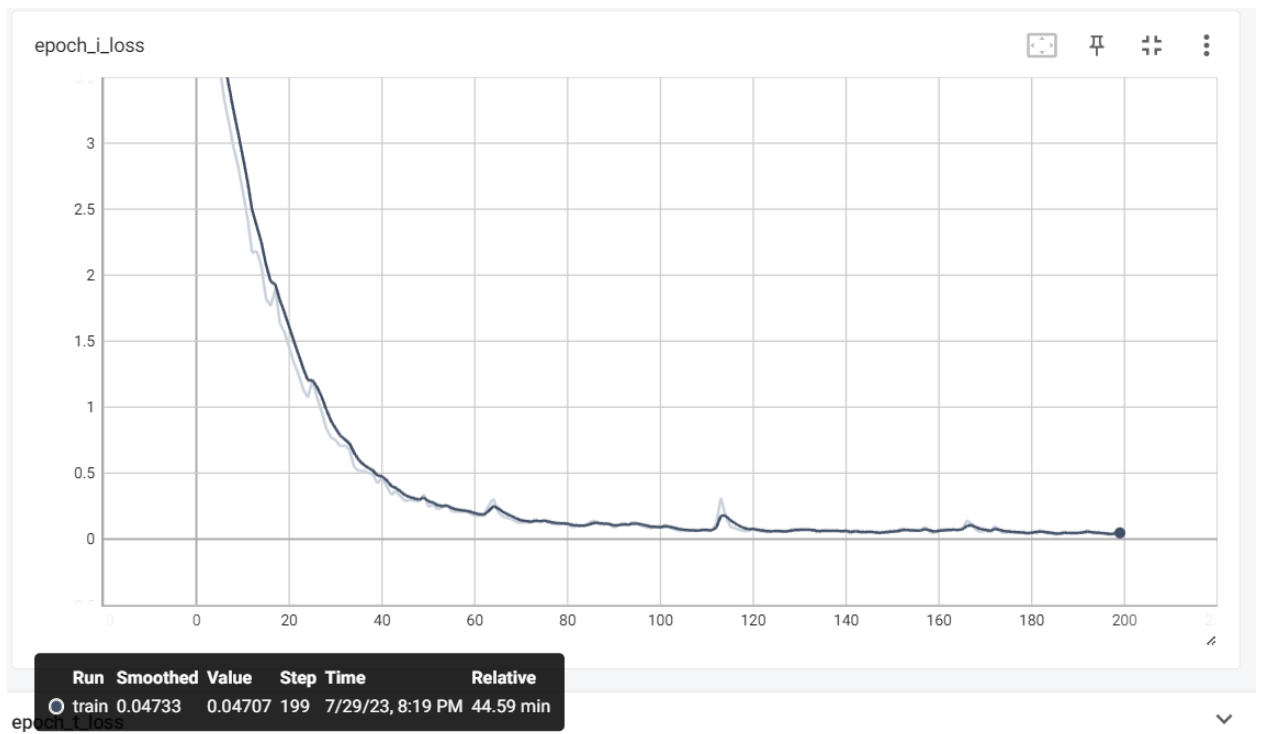
Tại epoch thứ 200:

- ***e_loss***: 0.051



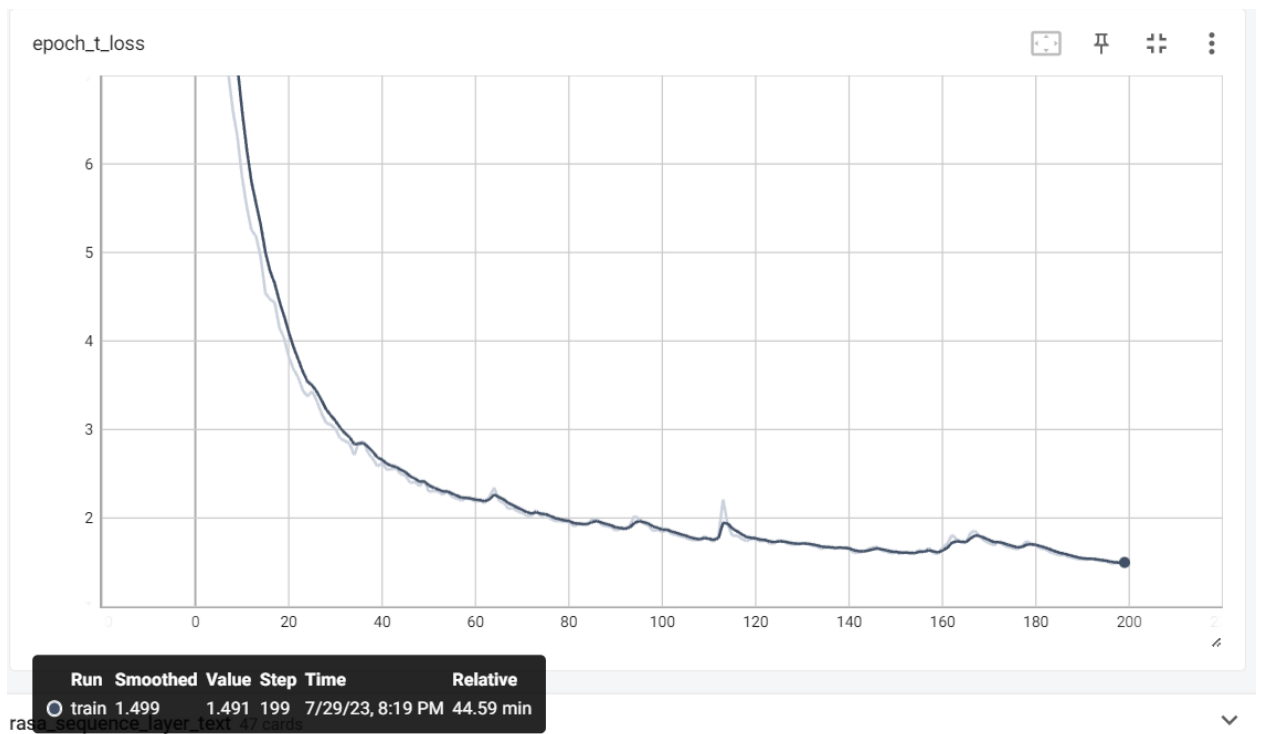
Hình 4.6 Biểu đồ ***e_loss***

- ***i_loss***: 0.0471



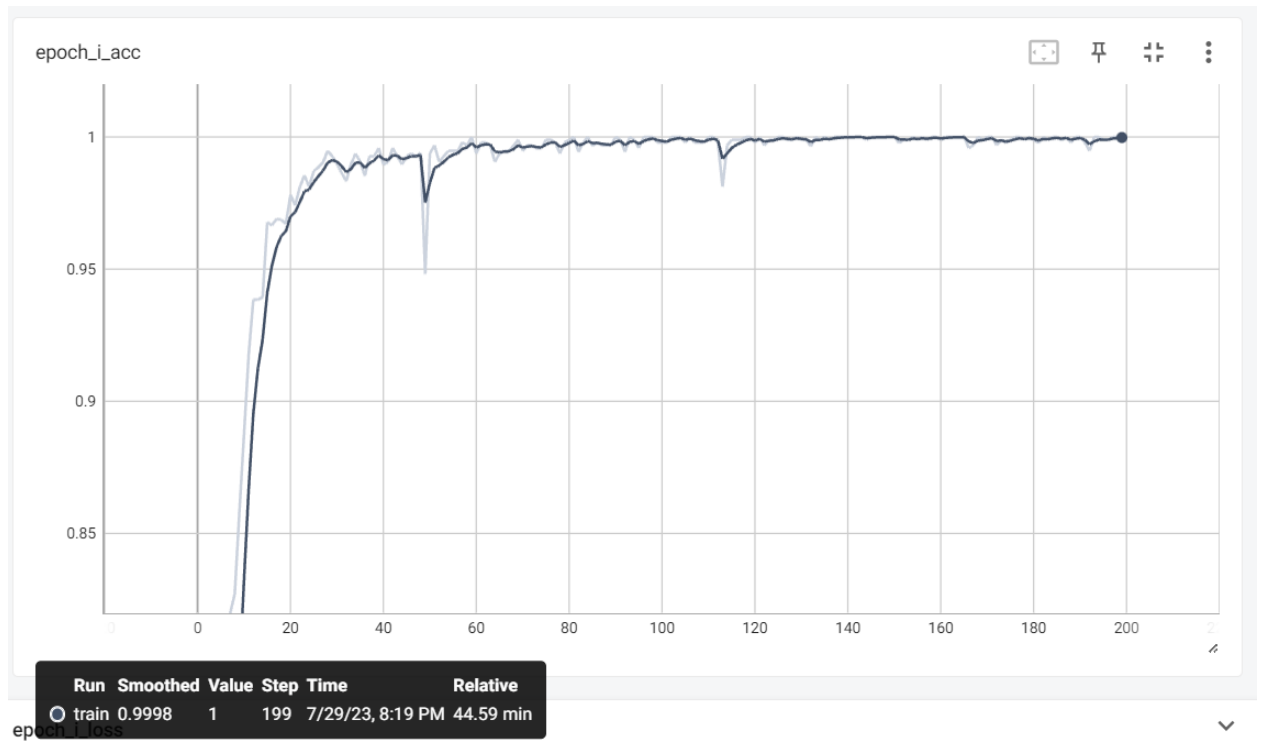
Hình 4.7 Biểu đồ i_loss

- t_loss : 1.49



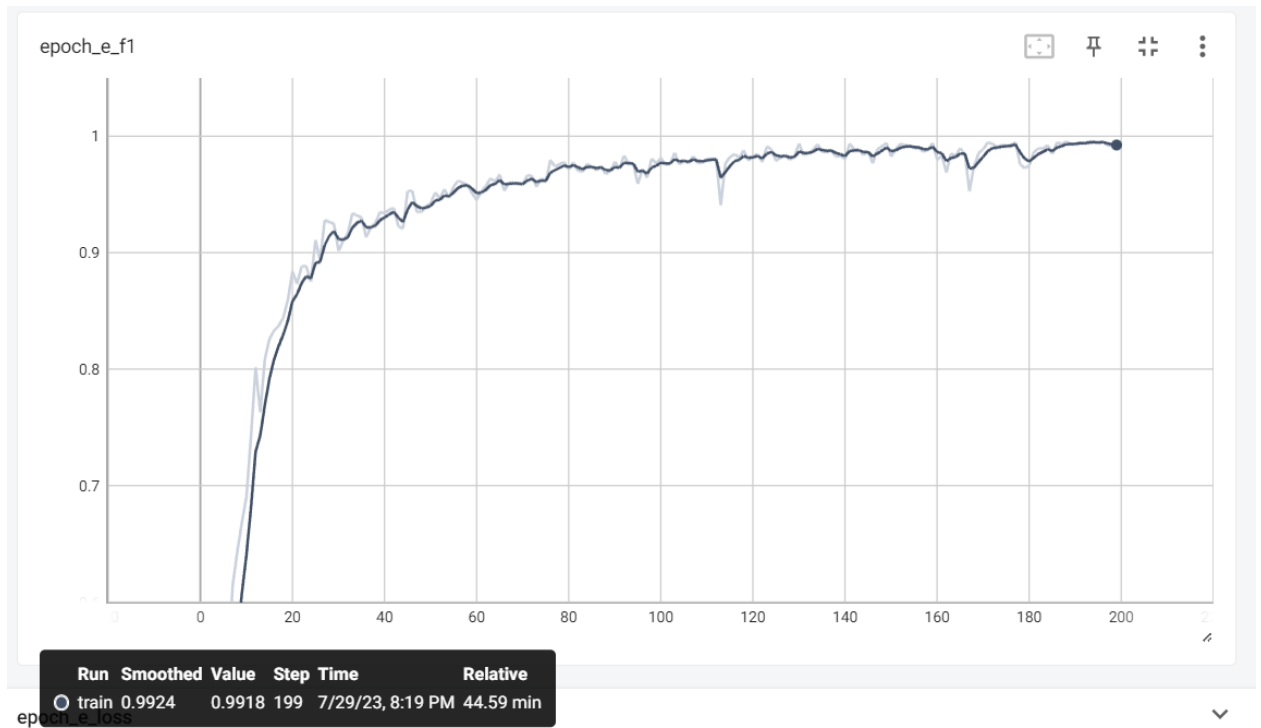
Hình 4.8 Biểu đồ t_loss

- i_acc : 1



Hình 4.9 Biểu đồ i_acc

- e_f1 : 0.992



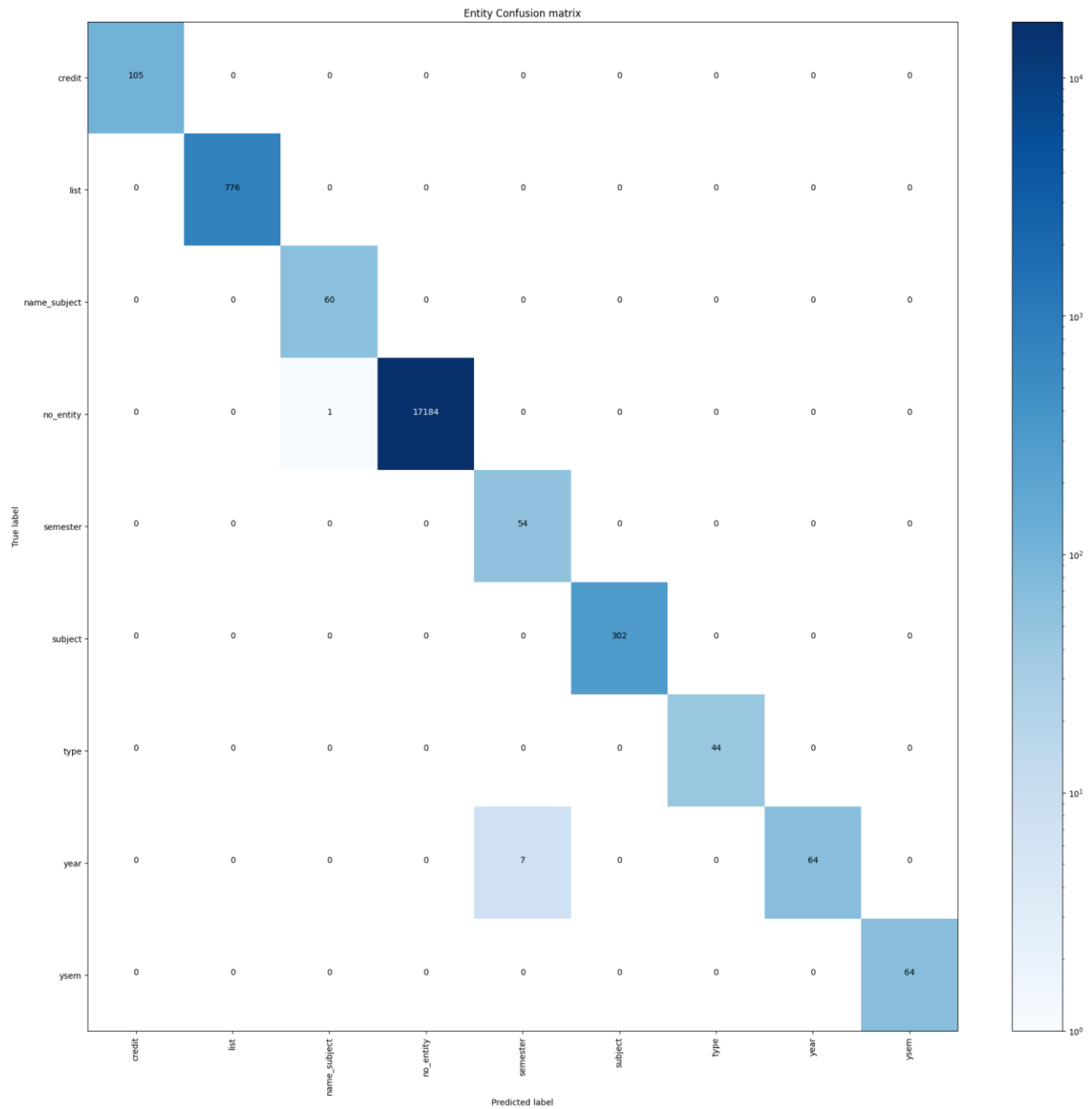
Hình 4.10 Biểu đồ e_{f1}

Từ kết quả trên chúng ta có thể có được một số kết luận như sau:

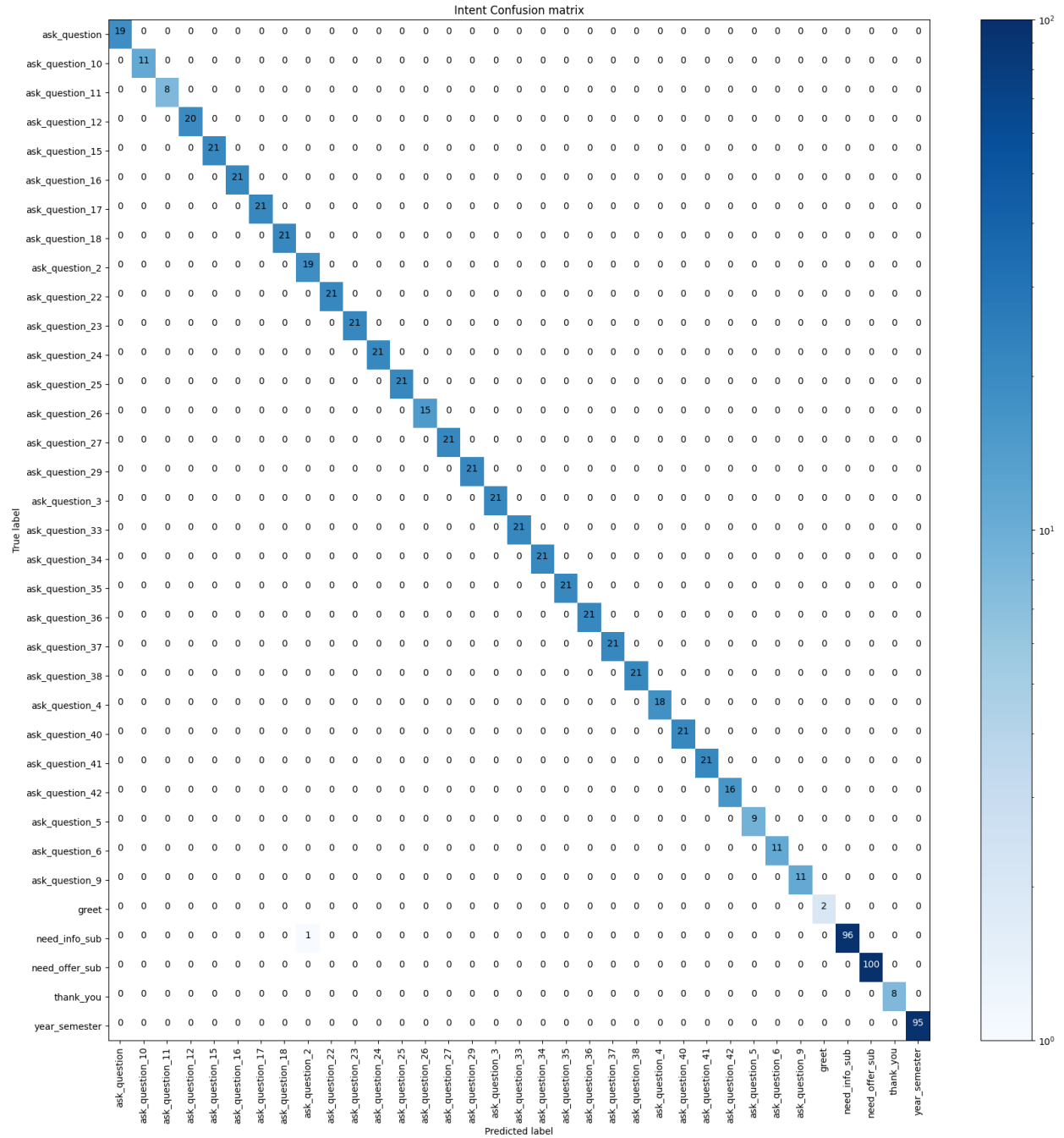
- Với e_{f1} và e_{loss} chúng ta có thể xác định mô hình có thể mô hình có thể xác định được các thực thể trong yêu cầu của người dùng nhưng sai số vẫn còn cao.
- Với i_{loss} cho thấy sai số trong dự đoán ý định người dùng là thấp.
- Dù t_{loss} bé nhưng vẫn cho ta thấy mô hình vẫn có sai số trong việc xác định thực thể và ý định mà như chúng ta có thể thấy phần lớn là từ việc sai số trong dự đoán thực thể.
- i_{acc} cho chúng ta thấy mô hình hoàn toàn dự đoán đúng ý định của người dùng trên tập huấn luyện.

Từ các kết luận trên, chúng ta có thể đưa ra tổng quát như sau: về tổng quan đây là một mô hình hoạt động tốt trong việc xác định ý định của người dùng nhưng vẫn còn hạn chế về mặt xác định chính xác thực thể trong ý định của người dùng. Có thể thu thập thêm dữ liệu hoặc tăng cường dữ liệu sẽ giúp cải thiện tình trạng này.

Ngoài ra chúng ta còn có thể đánh giá qua tiêu chí *confusion matrix* (ma trận nhầm lẫn) của việc xác định thực thể và ý định như sau:



Hình 4.11 Confusion matrix của việc xác định entities



Hình 4.12 Confusion matrix của việc xác định intents

CHƯƠNG 5. CÀI ĐẶT VÀ THỰC NGHIỆM

Ở chương cuối cùng này, ta sẽ tìm hiểu cách cài đặt trợ lý ảo và thử nghiệm một số kịch bản để đánh giá kết quả cuối cùng.

5.1. Cài đặt

Bước 1: Training model bằng rasa train –fixed-model-name tenmodel.

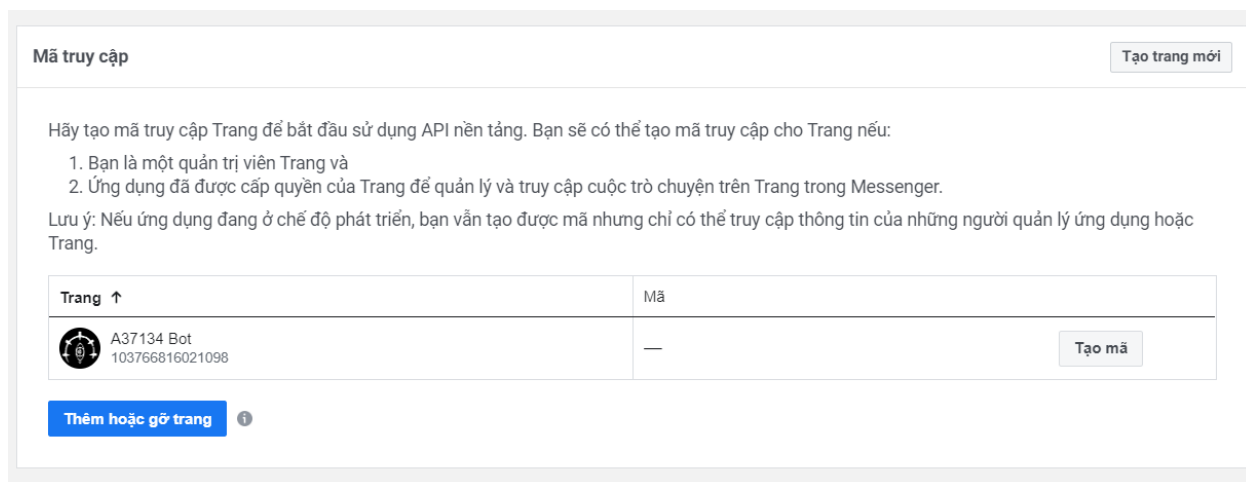
Bước 2: Chạy lệnh rasa run actions và rasa run.

Bước 3: Chạy ngrok http 5005.

Bước 4: Truy cập Facebook Developer khởi tạo ứng dụng khởi tạo ứng dụng.

Bước 5: Chọn messenger chọn cài đặt.

Bước 6: Tại mục mã truy cập tiến hành tạo một Page mới cần thêm 2 trường messages và messaging_postbacks để bot có thể trả lời tin nhắn.




Mã truy cập Tạo trang mới

Hãy tạo mã truy cập Trang để bắt đầu sử dụng API nền tảng. Bạn sẽ có thể tạo mã truy cập cho Trang nếu:

1. Bạn là một quản trị viên Trang và
2. Ứng dụng đã được cấp quyền của Trang để quản lý và truy cập cuộc trò chuyện trên Trang trong Messenger.

Lưu ý: Nếu ứng dụng đang ở chế độ phát triển, bạn vẫn tạo được mã nhưng chỉ có thể truy cập thông tin của những người quản lý ứng dụng hoặc Trang.

Trang ↑	Mã
 A37134 Bot 103766816021098	— Tạo mã

Thêm hoặc gỡ trang ⓘ

Hình 5.1 Giao diện mã truy cập

Bước 7: Khởi tạo Webhooks.

Webhooks

Để nhận tin nhắn và các sự kiện khác do người dùng Messenger gửi, ứng dụng sẽ bật tích hợp webhooks.

URL gọi lại

https://d904-2405-4802-1f9f-4200-e18f-b164-e8d-e612.ap.ngrok.io/web...

Mã xác minh

.....

Yêu cầu xác thực và thông báo Webhook cho đối tượng này sẽ được gửi đến URL này.

Mã Meta sẽ trả về cho bạn như một phần của quá trình xác minh URL gọi lại.

Chỉnh sửa URL gọi lại

Hiện thị lỗi gần đây

Trang ↑

Webhooks

A37134 Bot

103766816021098

2 trường

messages, messaging_postbacks

Chỉnh sửa

Thêm hoặc gỡ trang

Tùy chọn kiểm soát phiên bản của trường webhook

Hình 5.2 Giao diện Webhooks

Bước 8: Lấy mã (token) của ứng dụng trong bước 7 và dán vào page-access-token trong file credentials.yml.

Bước 9: Chọn chỉnh sửa URL gọi lại ở giao diện Webhooks.

Bước 10: Khi chạy ngrok ở bước 3 ta sẽ được:

```

C:\WINDOWS\system32\cmd.exe
ngrok
Send your ngrok traffic logs to Datadog: https://ngrok.com/blog-post/datadog-logs

Session Status      online
Session Expires     1 hour, 59 minutes
Update              update available (version 3.3.1, Ctrl-U to update)
Terms of Service     https://ngrok.com/tos
Version             3.2.2
Region              Asia Pacific (ap)
Latency              127ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://7dba-2405-4802-1c58-fb30-f9e7-f29c-d533-2041.ap.ngrok.io -> http://localhost:5005

Connections          ttl    opn    rt1    rt5    p50    p90
0                   0      0.00   0.00   0.00   0.00

```

Hình 5.3 Khởi tạo ngrok

Dán đường link ở Forwarding vào giao diện chỉnh sửa URL mục URL gọi lại và thêm /webhooks/facebook/webhook vào cuối như sau:

58

Chỉnh sửa URL gọi lại

URL gọi lại

`https://27ca-2405-4802-1c58-fb30-f9e7-f29c-d533-2041.ap.ngrok.io/webhooks/facebook/webhook`

Mã xác minh

`a37134`

[Tìm hiểu thêm](#) Hủy Xác minh và lưu

Hình 5.4 Giao diện chỉnh sửa URL gọi lại

Mã xác minh sẽ được lấy từ verify trong file credentials.yml sẽ do chúng ta tự định nghĩa. Chọn xác minh và lưu.

Bước 11: Tạo trang cài đặt chọn thông tin cơ bản lấy khoá bí mật của ứng dụng sau đó dán vào secret trong file credentials.yml.

Cài đặt

Thông tin cơ bản

Năng cao

Vai trò trong ứng dụng

Cảnh báo

Xét duyệt ứng dụng

Sản phẩm [Thêm sản phẩm](#)

Webhooks

Messenger

Nhật ký hoạt động

Nhật ký hoạt động

ID ứng dụng: 550055470338746

Khóa bí mật của ứng dụng: [Masks] Hiện thị

Tên hiển thị: A37134 Bot

Vùng tên: [Empty]

Miền ứng dụng: [Empty]

Email liên hệ: qhuy20012001@gmail.com

URL chính sách quyền riêng tư: Hộp thoại Chính sách quyền riêng tư để đăng nhập và chi tiết ứng...

URL Điều khoản dịch vụ: Hộp thoại Điều khoản dịch vụ để đăng nhập và chi tiết ứng dụng

Biểu tượng ứng dụng (1024 x 1024): [Image placeholder]

Hạng mục: Chọn hạng mục

Hãy tìm thêm thông tin về hạng mục ứng dụng [tại đây](#)

Bỏ Lưu thay đổi

<https://developers.facebook.com/apps/550055470338746/settings/basic/>

Hình 5.5 Giao diện cài đặt thông tin cơ bản

Bước 12: Sau khi làm xong ta sẽ có file credentials.yml như sau:

```

credentials.yml
...
1  # This file contains the credentials for the voice & chat platforms
2  # which your bot is using.
3  # https://rasa.com/docs/rasa/messaging-and-voice-channels
4
5  rest:
6  # # you don't need to provide anything here - this channel doesn't
7  # # require any credentials
8
9
10 facebook:
11   verify: "a37134"
12   secret: "xxxxxxxxxxxxx"
13   page-access-token: "xxxxxxxxxxxxx"
14

```

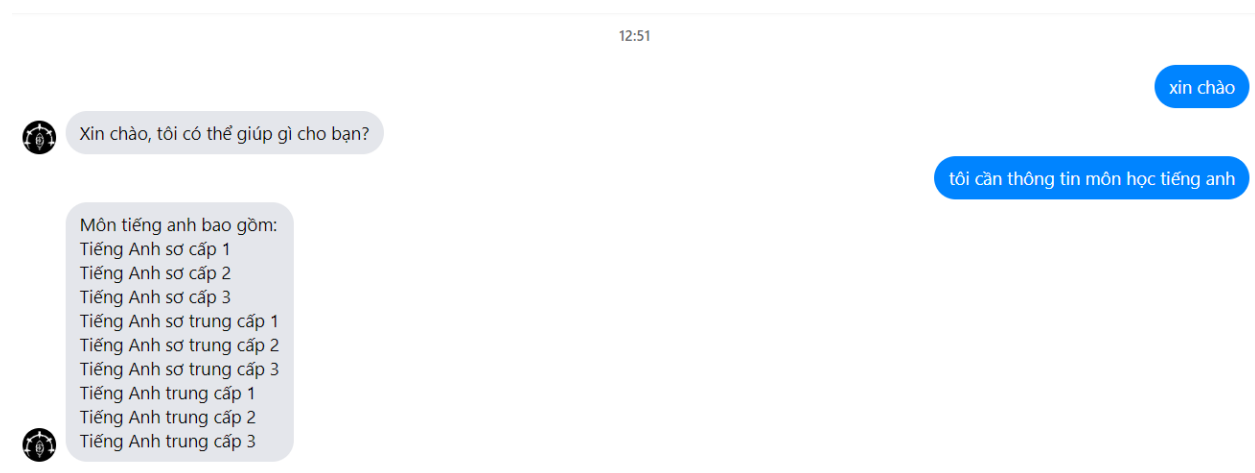
Hình 5.6 File credentials.yml

Bước 13: Truy cập vào Page bằng tài khoản Facebook cá nhân sau đó nhấn tin để thử nghiệm.

5.2. Thực nghiệm

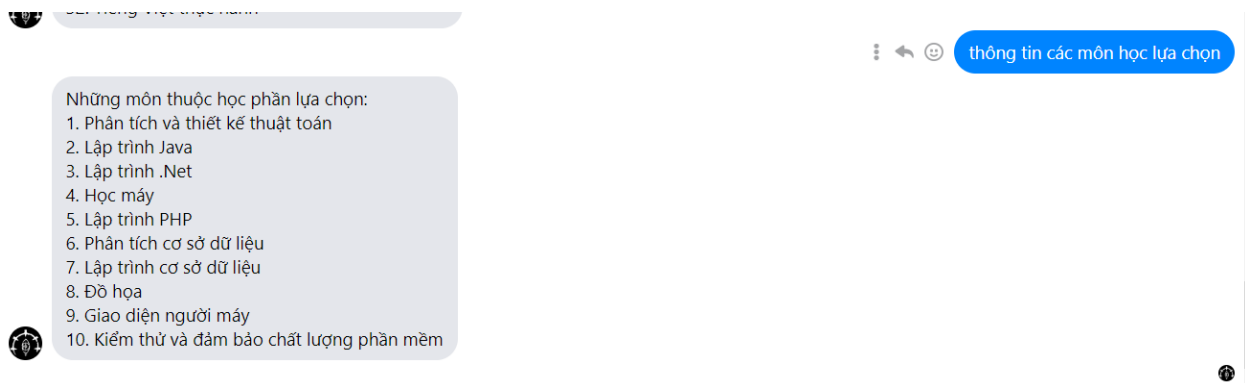
5.2.1. Yêu cầu thông tin môn học

Chúng ta sẽ yêu cầu trợ lý ảo cung cấp thông tin môn học chung chung như “môn học tiếng anh”. Trợ lý ảo sẽ không rõ bạn hỏi môn tiếng Anh nào nên sẽ liệt kê ra các môn học tiếng Anh như hình bên dưới.



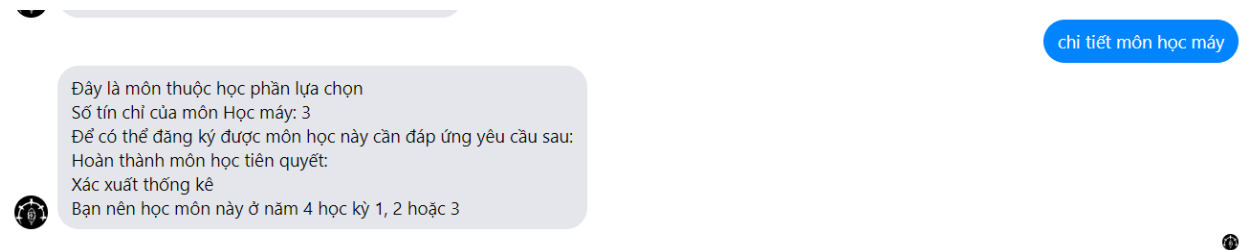
Hình 5.7 Liệt kê các môn học có chung tên

Tương tự khi yêu cầu thông tin các môn học lựa chọn, ta sẽ nhận được danh sách các môn học thuộc học phần lựa chọn theo chương trình.



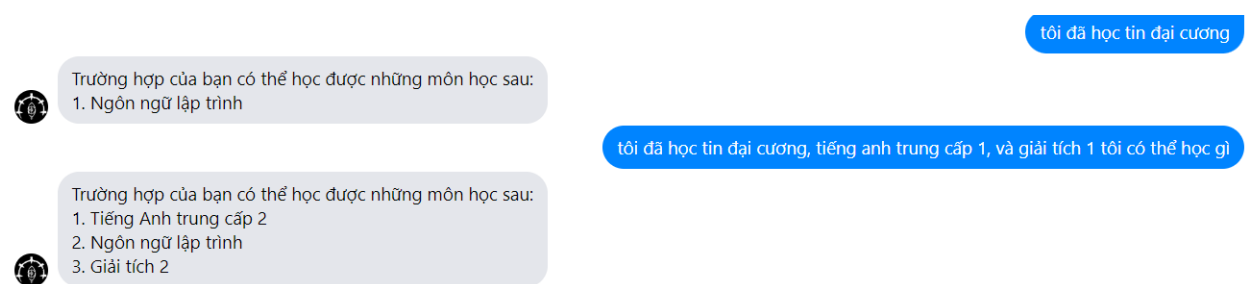
Hình 5.8 Liệt kê các môn thuộc một học phần

Tiếp theo, khi yêu cầu chi tiết thông tin một môn học cụ thể, ta sẽ nhận được các thông tin như sau: môn học thuộc học phần nào, số tín chỉ, điều kiện tiên quyết và gợi ý nên học môn học này bao giờ.



Hình 5.9 Chi tiết một môn học

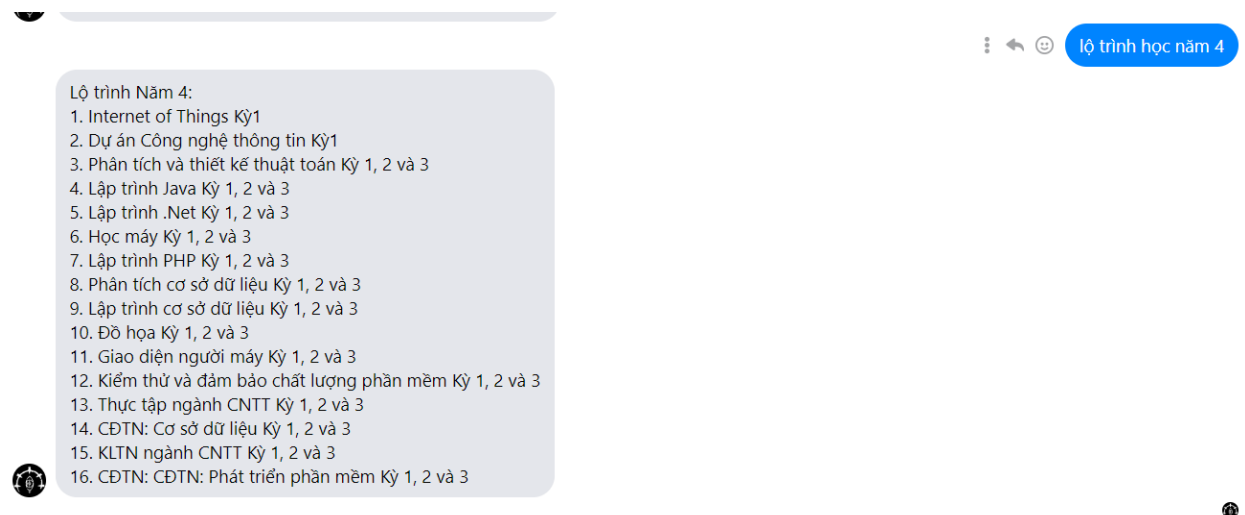
Sau đây là một trường hợp cung cấp thông tin và nhận về những môn học có thể học.



Hình 5.10 Yêu cầu các môn có thể học

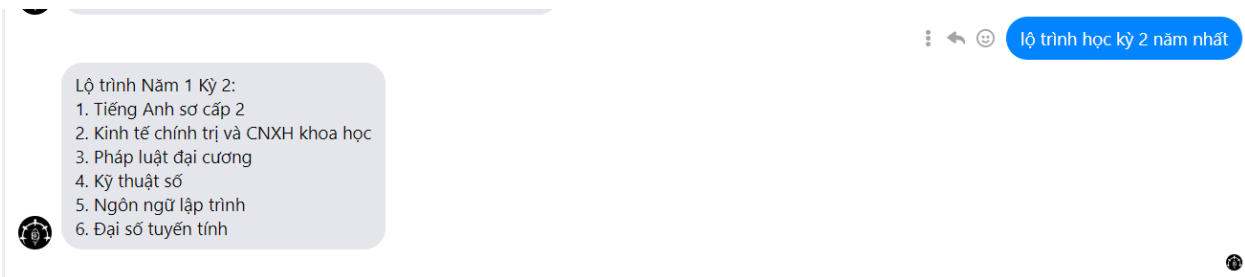
5.2.2. Yêu cầu lộ trình học

Ta sẽ thử với kịch bản người dùng hỏi lộ trình học năm 4 năm học sẽ viết bằng số, trợ lý ảo hiểu và đưa ra lộ trình học cụ thể.



Hình 5.11 Lộ trình học của năm học

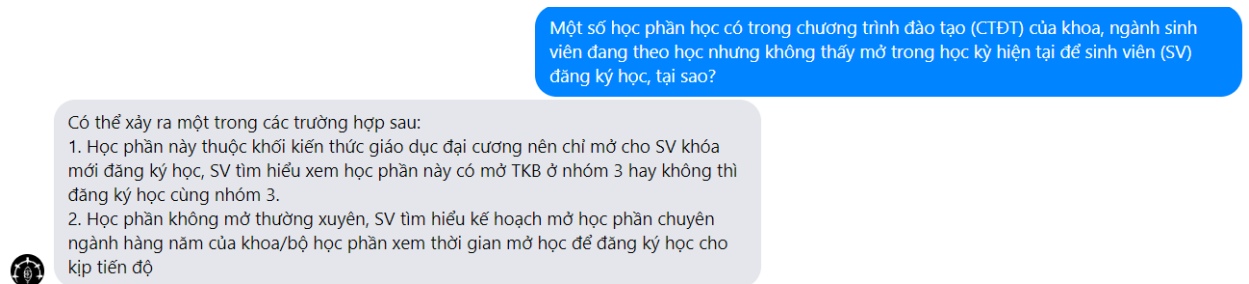
Một kịch bản khác là khi ta hỏi chi tiết hơn bao gồm cả kỳ học và năm học nhưng khi này năm học sẽ viết bằng chữ, ta vẫn nhận về được kết quả theo mong muốn.



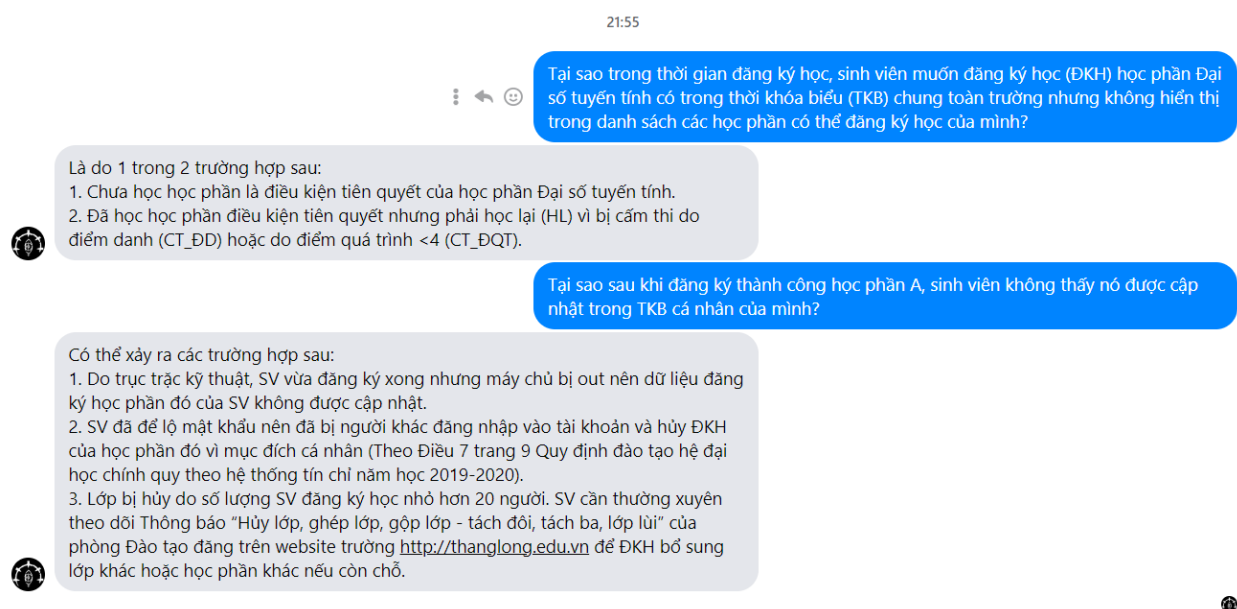
Hình 5.12 Lộ trình học của kỳ và năm học

5.2.3. Hỏi giải đáp một số vấn đề

Ngoài ra trợ lý ảo còn có thể trả lời thêm một số câu hỏi khác không liên quan đến chương trình học, hay thông tin môn học như:



Hình 5.13 Một số giải đáp khác phần 1



Hình 5.14 Một số giải đáp khác phần 2

KẾT LUẬN

1. Tổng kết

Qua quá trình thực hiện khóa luận này, em đã thu nhận thêm được các kiến thức về thu thập, xử lý dữ liệu. Ngoài ra, em cũng có cơ hội nắm vững các kiến thức về xây dựng mô hình *học máy*, mạng *neuron hồi quy*, cũng như tìm hiểu được thêm nhiều mô hình sử dụng cho xử lý ngôn ngữ tự nhiên và biết được thêm nhiều phương pháp xử lý, tăng cường dữ liệu. Khóa luận cũng đồng thời giúp em có được nhiều kinh nghiệm thực tiễn về phương pháp xây dựng ứng dụng trí tuệ nhân tạo trên một framework mới là *Rasa*.

Do giới hạn thời gian thực hiện, số lượng dữ liệu khóa luận thu thập được vẫn chưa phải dồi dào. Tuy nhiên, thông qua việc tìm tòi các phương pháp học máy, khóa luận đã rút ra một số kết quả hữu ích có thể là định hướng kinh nghiệm cho các nghiên cứu trên chủ đề này về sau. Cụ thể là, dựa trên các kết quả này, khóa luận rút ra nên quan tâm trích rút thêm các đặc trưng khác bên cạnh việc xác định được mục đích yêu cầu của người dùng, từ đó có thể làm phong phú hơn các đầu ra câu trả lời.

2. Định hướng

Từ kết quả đạt được, trong tương lai, em muốn thử nghiệm thực hiện thêm một số phương pháp tăng cường dữ liệu khác và tìm thêm các nguồn dữ liệu bổ sung cho tập dữ liệu hiện có. Bên cạnh đó, em cũng muốn bổ sung thêm nhiều kịch bản hơn để giúp trợ lý ảo sẽ càng linh hoạt hơn. Một hướng phát triển nữa là hướng tới cải thiện câu trả lời của trợ lý ảo để tuy ý nghĩa trả lời có thể giống nhau, trợ lý ảo có thể cho ra các hình thức trả lời đa dạng không lặp lại – điều này sẽ giúp tăng tính tự nhiên và “người” hơn cho ứng dụng. Sau chót, trong tương lai, em dự định tìm kiếm và tích hợp các phương pháp huấn luyện cũng như học tăng cường cải tiến hơn để giúp mô hình học máy có khả năng tự cập nhật từ đó có thể tự nâng cao chất lượng.

TÀI LIỆU THAM KHẢO

- [1] B. Qifang, G. E. Katherine, K. Joshua and L. Justin, "American Journal of Epidemiology;," *What is Machine Learning? A Primer for the Epidemiologist*, vol. 188, no. 12, p. 2222–2239, December 2019.
- [2] M. S. Warren and P. Walter, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys*, vol. 5, no. 4, pp. 115-133, 1943.
- [3] i2tutorials, "Hình ảnh kiến trúc Neural Networks," [Online]. Available: <https://www.i2tutorials.com/hidden-layers-in-neural-networks/>.
- [4] J. J. Hopfield, "Recurrent Neural Networks," *Artificial Neural Networks*, vol. 4, no. 5, pp. 3-10, 1988.
- [5] Y. Gu, "Hình ảnh mô hình RNN Many to many," [Online]. Available: https://www.researchgate.net/figure/A-Many-to-Many-Recurrent-Neural-Network-RNN-Figure-1-shows-an-RNN-which-takes-an-input_fig1_338867344.
- [6] a. Tondak, "Ưu điểm của RNN," [Online]. Available: <https://k21academy.com/datascience-blog/machine-learning/recurrent-neural-networks/>.
- [7] S. H. a. J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, p. 1735–1780, 1997.
- [8] S. Andrew, H. Georg, R. Marc'Aurelio and Y. Ke, "An empirical study of learning rates in deep neural networks for speech recognition," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 26-31 May 2013.
- [9] R. E. David, E. H. Geoffrey and j. W. Ronald, "Learning representations by back-propagating errors," 1986.
- [10] K. Pawar, "Hình ảnh đồ thị của hàm Softmax," [Online]. Available: <https://insideaiml.com/blog/SoftMaxActivation-Function-1034>.

- [11] "Underfitting và Overfitting," [Online]. Available: <https://trituenhantao.io/kien-thuc/van-de-overfitting-underfitting-trong-machine-learning/#:~:text=Overfitting%3A%20khi%20mô%20hình%20có,mô%20tả%20tổ%20thể%20mới..>
- [12] N. Morgan and H. Bourlard, "Generalization and Parameter Estimation in Feedforward Nets: Some Experiments," *Advances in Neural Information Processing Systems 2 (NIPS 1989)*, 1989.
- [13] P. Lutz, "Early Stopping — But When?," *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, vol. 7700, pp. 53-67, 2012.
- [14] R. Gençay, "Hình ảnh mô tả Early stopping," [Online]. Available: <https://paperswithcode.com/method/early-stopping>.
- [15] F. . Y. Steven, G. Varun, W. Jason, C. Sarath, V. Soroush, M. Teruko and H. Eduard, *A Survey of Data Augmentation Approaches for NLP*, 1 Dec 2021.
- [16] S. Nitish, H. Geoffrey, K. Alex, S. Ilya and S. Ruslan, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 101, p. 1929–1958, January 2014.
- [17] S. Nitish, H. E. Geoffrey, K. A. S. Ilya and S. R. , "Hình ảnh mô tả Dropout," [Online]. Available: <https://www.semanticscholar.org/paper/Dropout%3A-a-simple-way-to-prevent-neural-networks-Srivastava-Hinton/34f25a8704614163c4095b3ee2fc969b60de4698>.
- [18] K. P. Diederik and B. Jimmy, "Adam: A method for stochastic optimization," *ArXiv Prepr. ArXiv14126980*, 2014.
- [19] G. Hinton, "Overview of mini-batch gradient descent," *Neural Networks for Machine Learning*, p. Lecture 6a, 2012.
- [20] S. Mahendra, "What is the Adam Optimizer and How is It Used in Machine

Learning," June 13, 2023.

- [21] RASA, "Rasa," [Online]. Available: <https://rasa.com/docs/rasa/>.
- [22] Rasa, "Hình ảnh Rasa," [Online]. Available: <https://github.com/RasaHQ/rasa>.
- [23] pawangfg, "Rasa NLU - Rasa Core," [Online]. Available: <https://www.geeksforgeeks.org/chatbots-using-python-and-rasa/>.
- [24] T. Wochinger, "RasaNLU - Intent Recognition," [Online]. Available: <https://rasa.com/blog/rasa-nlu-in-depth-part-1-intent-classification/>.
- [25] T. Wochinger, "RasaNLU - Entity Recognition," [Online]. Available: <https://rasa.com/blog/rasa-nlu-in-depth-part-2-entity-recognition/>.
- [26] V. N. Vladimir and C. Y. Alexey, *The Nature of Statistical Learning Theory*, 1995.
- [27] V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, G. N. Aidan, K. Łukasz and P. Illia, *Attention Is All You Need*, 2017.
- [28] s. guhan, "RasaCore - Chức năng," [Online]. Available: <https://indiaai.gov.in/article/rasa-chatbot-framework-nlu-core>.
- [29] Rasa, "Pipelines," [Online]. Available: <https://rasa.com/docs/rasa/tuning-your-model/>.
- [30] Rasa, "Policies," [Online]. Available: <https://rasa.com/docs/rasa/policies/>.
- [31] A. Hughes, "Microsoft SQL Server," [Online]. Available: <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server>.
- [32] Toric, "Hình ảnh Microsoft SQL Server," [Online]. Available: <https://www.toric.com/integrations/microsoft-sql>.
- [33] Ngrok, "Ngrok," [Online]. Available: <https://ngrok.com/>.

- [34] S. R. Lee, "Hình ảnh Ngrok," [Online]. Available:
<https://www.endtoend.ai/tutorial/ngrok-ssh-forwarding/>.
- [35] Facebook, "Facebook Developers," [Online]. Available:
<https://developers.facebook.com/>.
- [36] T. Hoang, "Hình ảnh Facebook developers," [Online]. Available:
<https://cellphones.com.vn/sforum/facebook-developer-la-gi-cach-dang-ky-tai-khoan-facebook-developer-cuc-de>.
- [37] Trường Đại học Thăng Long Hướng dẫn học tập, Hà Nội, 2019-2020.
- [38] Zahichemaly, "Hình ảnh sơ đồ kiến trúc tổng thể," [Online]. Available:
<https://github.com/zahichemaly/chatbot-for-binge-eating-disorders>.
- [39] L. Ajay, "Hình ảnh cấu trúc thư mục Data," [Online]. Available:
<https://towardsdatascience.com/ai-chatbots-made-easy-courtesy-rasa-8ecdc2853e66>.