

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Quản lý hội thoại trong hệ thống tương tác thông minh SmartDialog với học sâu

Nguyễn Hoàng Kỳ

Hà Nội, 05/2019

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Quản lý hội thoại trong hệ thống tương tác thông minh SmartDialog với học sâu

Sinh viên thực hiện Nguyễn Hoàng Kỳ

Người hướng dẫn TS. Nguyễn Thị Thu Trang

Hà Nội, 05/2019

Lời cam kết

Họ và tên sinh viên: Nguyễn Hoàng Kỳ

Điện thoại liên lạc: +84796 329 663 Email: hoangky17101996@gmail.com

Lớp: CNTT-TT 2.04 - K59 Hệ đào tạo: Đại học chính quy

Tôi – *Nguyễn Hoàng Kỳ* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Thị Thu Trang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 24 tháng 05 năm 2019

Tác giả ĐATN

Lời cảm ơn

Đầu tiên, em xin phép gửi lời cảm ơn chân thành nhất tới TS Nguyễn Thị Thu Trang, cô thực sự là một nhà giáo có tâm và có tầm, đã tận tình chỉ bảo, giúp đỡ em trong suốt khoảng thời gian thực hiện đồ án này.

Em xin chân thành cảm ơn toàn thể Ban lãnh đạo nhà trường, các thầy cô trong trường Đại học Bách Khoa Hà Nội nói chung và những thầy cô trong Viện Công nghệ thông tin và Truyền thông nói riêng, đã hết sức tạo điều kiện cho em trong quá trình hoàn thành đồ án.

Em cũng xin được gửi lời cảm ơn tới toàn thể các anh, các chị và bạn bè cùng học tập và làm việc tại phòng thí nghiệm 914, mọi người đã rất nhiệt tình giúp đỡ, truyền đạt lại cho em nhiều kinh nghiệm trong suốt khoảng thời gian quý báu vừa qua.

Trong quá trình thực hiện đồ án tốt nghiệp, dù em đã cố gắng hoàn thiện những vẫn sẽ không thể nào tránh khỏi được những sai sót. Em mong các thầy cô thông cảm và góp ý để giúp em không tái phạm lại những lỗi sai đó nữa.

... Và cuối cùng, thì ngày ấy cũng đã tới rồi.

“Nắng Bách Khoa thiêu đốt thời trai trẻ

Mây Hà thành che lấp tuổi thanh xuân”

Nhưng nếu được lựa chọn lại, cũng không thể có sự lựa chọn nào khác ngoài Bách Khoa. Năm năm không phải là quãng thời gian quá dài, nhưng sống và học tập trong môi trường Bách Khoa, tiếp xúc với những tinh thần Bách Khoa, đã mang lại thật nhiều giá trị đặc biệt khó mà đo đếm được. Mong sao thứ tinh thần quý giá ấy mà Bách Khoa đang nuôi dưỡng sẽ còn mãi và lan tỏa tới nhiều thế hệ sinh viên mai sau.

Em xin chân thành cảm ơn !

Tóm tắt

Tương tác bằng văn bản hoặc tiếng nói hiện nay đang là kênh tương tác người máy phổ biến trong các ứng dụng thông minh thời đại chuyển đổi số. Có hai thành phần quan trọng trong các hệ thống tương tác hội thoại đó là thành phần hiểu ngôn ngữ tự nhiên và thành phần quản lý hội thoại. Đồ án này tập trung vào trình bày các vấn đề và giải pháp liên quan tới thành phần quản lý hội thoại. Hiện nay, những hệ thống quản lý hội thoại dựa trên luật thủ công đơn giản và dễ xây dựng nhưng cho chất lượng hội thoại thiếu tự nhiên và khó mở rộng. Các cách huấn luyện theo phương pháp end-to-end cho mô hình hội thoại có khả năng học được các biểu diễn ẩn của trạng thái hội thoại mà không cần tới các luật hoặc đặc trưng thủ công, tuy nhiên lại yêu cầu một lượng dữ liệu huấn luyện đáng kể, thường không phù hợp để ứng dụng trong những sản phẩm thương mại.

Hướng tiếp cận đề xuất ứng dụng học sâu không theo phương pháp end-to-end, mà tách biệt hoàn toàn hai thành phần hiểu ngôn ngữ tự nhiên và quản lý hội thoại. Đề dự đoán hành động người dùng, em véc-tơ hóa các trạng thái hội thoại để làm đầu vào cho mô hình Long Short-Term Memory và mô hình Memoization. Cùng với đó giới thiệu phương pháp học tương tác nhằm giúp nhà phát triển huấn luyện hệ thống hội thoại thông qua tương tác trực tiếp một cách đơn giản và trực quan. Mô hình dự đoán hành động đạt độ chính xác lên tới 94,8% trên bộ dữ liệu tư vấn tuyển sinh (của Đại học Bách Khoa Hà Nội). Thêm vào đó, em cũng đưa ra giải pháp tích hợp bộ dữ liệu trên với nền tảng Zalo và triển khai một hệ thống Chatbot tư vấn tuyển sinh mang tới cách tìm kiếm thông tin thú vị cho học sinh, sinh viên sắp nhập học.

Giải pháp quản lý hội thoại trình bày trong đồ án được triển khai trong nền tảng SmartDialog [1], [2], nền tảng đã có bài báo được chấp nhận tại **hội nghị NLPIR 2019** (International Conference on Natural Language Processing and Information Retrieval) và đạt **giải Ba Cuộc thi sản phẩm sáng tạo Viện CNTT 2019**.

Mục lục

Lời cam kết	iii
Lời cảm ơn	iv
Tóm tắt.....	v
Mục lục.....	vi
Danh mục hình vẽ.....	ix
Danh mục bảng.....	xi
Danh mục các từ viết tắt.....	xii
Danh mục thuật ngữ.....	xiii
Chương 1 Bài toán quản lý hội thoại.....	1
1.1 Đặt vấn đề.....	1
1.2 Hệ thống hội thoại	2
1.2.1 Tổng quan.....	2
1.2.2 Phân loại.....	4
1.3 Mô hình quản lý hội thoại và các nghiên cứu liên quan	7
1.3.1 Mô hình luật thủ công (Hand-crafted Rules Dialogue Systems)	7
1.3.2 Mô hình giám sát hoàn toàn (Fully-Supervised Models)	8
1.3.3 Mô hình học tăng cường (Reinforcement Learning-based Models)	9
1.3.4 Mô hình lai (Hybrid Models)	9

1.3.5 Mô hình ràng buộc dữ liệu (Data-Constrained Dialogue Systems)	10
1.4 Mục tiêu và phạm vi đề tài	10
1.5 Định hướng giải pháp	11
1.6 Bố cục đề án	12
Chương 2 Cơ sở lý thuyết	13
2.1 Giới thiệu về học máy	13
2.2 Mạng nơ-ron nhân tạo	14
2.2.1 Kiến trúc	14
2.2.2 Tối ưu hóa tham số	16
2.2.3 Vấn đề Overfitting và phương pháp giải quyết	17
2.3 Mạng nơ-ron nhân tạo Long Short-Term Memory	18
2.3.1 Mạng nơ-ron hồi quy (recurrent neural network - RNN)	18
2.3.2 Mạng Long Short-Term Memory (LSTM)	19
2.4 Phương pháp học trực tuyến (Online Learning)	20
Chương 3 Đề xuất mô hình quản lý hội thoại và học tương tác	22
3.1 Giải pháp tổng quan	22
3.2 Các mô hình dự đoán hành động	25
3.2.1 Trích xuất biểu diễn đặc trưng trạng thái hội thoại	26
3.2.2 Mô hình Long Short-Term Memory	28
3.2.3 Mô hình ghi nhớ (Memoization)	29
3.3 Học tương tác	30
3.4 Tạo kịch bản huấn luyện tự động	32
3.5 Thử nghiệm và đánh giá	35

3.5.1 Dữ liệu.....	35
3.5.2 Kết quả thử nghiệm.....	36
Chương 4 Phát triển phân hệ quản lý hội thoại trong SmartDialog	38
4.1 Kiến trúc tổng quan của SmartDialog.....	38
4.2 Dịch vụ dự đoán hành động	41
4.2.1 Thiết kế.....	41
4.2.2 Xây dựng hệ thống.....	44
4.3 Dịch vụ huấn luyện hành động	45
4.3.1 Thiết kế.....	45
4.3.2 Xây dựng hệ thống.....	47
4.4 Quản lý phiên hội thoại trong máy chủ Gateway	48
4.4.1 Quy trình nghiệp vụ	48
4.4.2 Thiết kế.....	49
4.4.3 Xây dựng hệ thống.....	52
4.5 Triển khai	53
4.5.1 Cách thức triển khai	53
4.5.2 Kết quả triển khai.....	53
4.5.3 Tích hợp Chatbot Tư vấn tuyển sinh với nền tảng Zalo	53
Chương 5 Kết luận và hướng phát triển	56
5.1 Kết luận	56
5.2 Hướng phát triển.....	57
Tài liệu tham khảo.....	58

Danh mục hình vẽ

Hình 1 Các thành phần của hệ thống hội thoại [5].	3
Hình 2 Ví dụ đồ thị hội thoại trong hệ thống tư vấn tuyển sinh.	4
Hình 3 Ví dụ các đoạn hội thoại trong hệ thống dựa trên khung.	5
Hình 4 Tế bào thần kinh sinh học [1].	14
Hình 5 Mạng tế bào thần kinh nhân tạo [26].	15
Hình 6 So sánh mạng nơ-ron đầy đủ và mạng có dropout [32].	17
Hình 7 Các mô-đun lặp chứa nhân <i>tanh</i> trong mạng nơ-ron hồi quy [33].	18
Hình 8 Các mô-đun lặp trong một mạng LSTM với cấu trúc ba cổng [16].	19
Hình 9 Tổng quan giải pháp quản lý hội thoại.	23
Hình 10 Ví dụ về thông tin lưu trữ của một đoạn hội thoại trong hệ thống.	26
Hình 11 Dạng dữ liệu đầu vào và đầu ra của mô hình dự đoán hành động.	27
Hình 12 Mô hình LSTM trong dự đoán hành động.	28
Hình 13 Minh họa giải pháp học tương tác.	31
Hình 14 Kiến trúc tổng quan nền tảng SmartDialog	39
Hình 15 Biểu đồ thiết kế gói của Worker dự đoán hành động.	42
Hình 16 Biểu đồ trình tự dự đoán hành động.	43
Hình 17 Biểu đồ trình tự cập nhật mô hình dự đoán hành động.	43
Hình 18 Biểu đồ lớp chi tiết trong dịch vụ dự đoán hành động	44

Hình 19 Biểu đồ thiết kế gói của dịch vụ huấn luyện hành động.	45
Hình 20 Biểu đồ trình tự huấn luyện mô hình dự đoán hành động.	46
Hình 21 Biểu đồ trình tự học tương tác.	47
Hình 22 Biểu đồ hoạt động người dùng giao tiếp với Chatbot qua Zalo.	48
Hình 23 Biểu đồ thiết kế gói của Gateway.	50
Hình 24 Thông tin lưu trữ trong phiên hội thoại.	51
Hình 25 Biểu đồ trình tự người dùng gửi tin nhắn Zalo.	52
Hình 26 Một số đoạn hội thoại với Chatbot tư vấn tuyển sinh trên nền tảng Zalo. .	54

Danh mục bảng

Bảng 1 Ví dụ kết quả trích xuất thông tin của thành phần NLU.....	23
Bảng 2 Một số ví dụ về hành động của hệ thống	25
Bảng 3 Một số dạng kịch bản được tạo tự động.....	33
Bảng 4 Bộ dữ liệu tư vấn tuyển sinh.....	35
Bảng 5 Độ chính xác của mô hình trên bộ dữ liệu TVTS.....	36
Bảng 6 Thư viện và công cụ sử dụng để xây dựng thành phần dự đoán hành động	44
Bảng 7 Thư viện và công cụ sử dụng để xây dựng dịch vụ huấn luyện hành động.	47
Bảng 8 Thư viện và công cụ sử dụng để xây dựng Gateway	52

Danh mục các từ viết tắt

ĐH BKHN	Đại học Bách Khoa Hà Nội
NLP	Natural Language Processing Xử lý ngôn ngữ tự nhiên
HCI	Human-computer interaction Tương tác người-máy
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
TVTS	Tư vấn tuyển sinh
SFTP	Secure File Transfer Protocol Giao thức truyền tệp an toàn
NLU	Natural Language Understanding Hiểu ngôn ngữ tự nhiên
API	Application Programming Interface Giao diện lập trình ứng dụng

Danh mục thuật ngữ

Intent	Ý định người dùng
Entity	Thực thể
Action	Hành động hệ thống
End-to-end	Đầu-cuối
Sequence-to-sequence	Chuyển đổi dãy sang dãy
Confidence	Độ chính xác trong mỗi lượt dự đoán của mô hình
Agent	Tác tử

Chương 1 Bài toán quản lý hội thoại

Chương này giới thiệu những vấn đề thực tế dẫn tới lý do chọn đề tài, tổng quan về bài toán quản lý hội thoại và thành phần quản lý hội thoại cùng một số cách tiếp cận hiện nay. Sau đó đưa ra phạm vi và mục tiêu của đề án, định hướng giải pháp và bố cục trình bày của đề án.

1.1 Đặt vấn đề

Sự phát triển của trí tuệ nhân tạo song hành cùng cuộc cách mạng 4.0 đang ở trong thời kỳ bùng nổ mạnh mẽ. Trong khi đó, các hệ thống hội thoại thông minh - thuộc lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) [3] và tương tác người máy (Human-computer interaction - HCI) [4] - đang được chú ý và ngày càng trở nên phổ biến hơn bao giờ hết. Một hệ thống hội thoại là một chương trình máy tính có khả năng tương tác với người dùng bằng các cách thức giao tiếp tự nhiên [5], qua đó mang đến những cách thức mới để con người giao tiếp với máy tính một cách thân thiện và hiệu quả hơn. Hiện nay, có nhiều các doanh nghiệp từ nhỏ, vừa tới các doanh nghiệp lớn đang đầu tư vào các hệ thống này, hướng tới xây dựng các hệ thống hội thoại trong nhà thông minh, hệ thống trả lời tự động, hệ thống chăm sóc khách hàng để tiếp cận được tới tập người dùng rộng hơn, giảm chi phí nhân lực và cải thiện trải nghiệm khách hàng. Có thể kể tên tới một số ứng dụng trợ lý ảo nổi tiếng như Siri, Alexa, Cortana, Google Assistant [6],... hoặc các nền tảng Chatbot như Chatfuel, Dialogflow, Microsoft Bot Framework [7], WIT [8],... Sử dụng những nền tảng này, doanh nghiệp hoặc các cá nhân dễ dàng tạo ra những Chatbot có thể tương tác với khách hàng của mình. Tuy vậy, một hệ thống hội thoại nếu phản hồi một cách máy móc, không có khả năng tương tác một cách tự nhiên có thể khiến việc sử dụng chúng phản tác dụng, gây nhàm chán hoặc tạo nhiều phản cảm cho khách hàng. Trên tình hình đó, các nghiên cứu gần đây đều nỗ lực cải thiện chất lượng hội

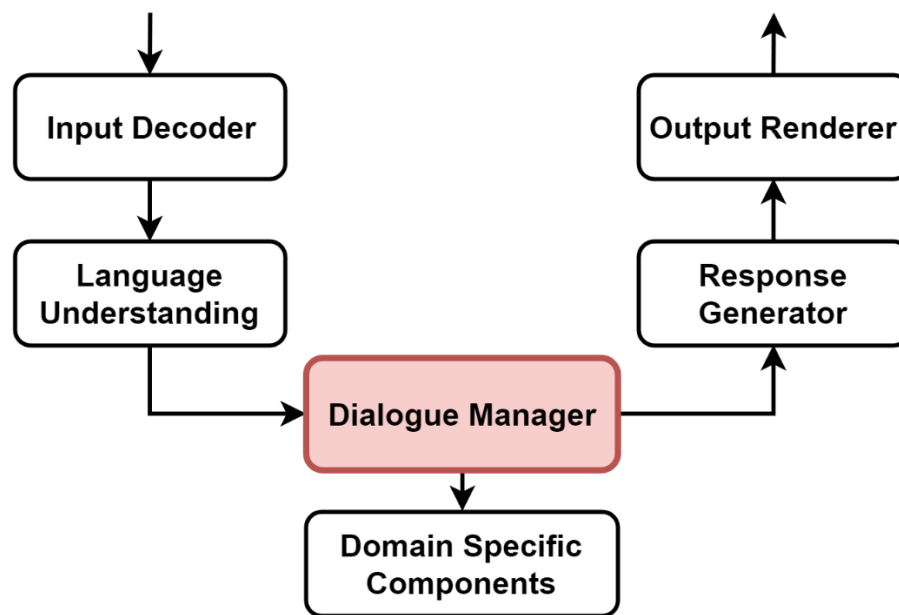
thoại của các hệ thống này, nhằm mong muốn tạo ra những hệ thống ngày càng đạt tới khả năng giao tiếp tương tự con người.

Một hệ thống hội thoại gồm nhiều các thành phần khác nhau (trình bày trong phần 1.2.1), nhưng có hai thành phần quan trọng nhất đó là (i) thành phần hiểu ngôn ngữ tự nhiên (Natural Language Understanding - NLU) và (ii) thành phần quản lý hội thoại (Dialogue Management). Các nghiên cứu hiện nay đang nỗ lực cải thiện chất lượng của hai thành phần trên, đặc biệt là thành phần quản lý hội thoại. Những hệ thống quản lý hội thoại dựa trên luật thủ công đơn giản và dễ xây dựng nhưng cho chất lượng hội thoại thiếu tự nhiên và khó mở rộng. Các cách huấn luyện theo phương pháp end-to-end cho mô hình hội thoại có khả năng học được các biểu diễn ẩn của trạng thái hội thoại mà không cần tới các luật hoặc đặc trưng thủ công, tuy nhiên cách tiếp cận này yêu cầu một lượng dữ liệu huấn luyện đáng kể, thường không phù hợp để ứng dụng trong những sản phẩm thương mại. Bên cạnh đó, có một số mô hình sử dụng phương pháp học tăng cường và mô hình lai được xây dựng nhằm giảm lượng dữ liệu huấn luyện cần thiết, kiến trúc của các hệ thống này tương đối phức tạp. Để hiểu rõ hơn về hệ thống hội thoại cũng như những hạn chế kể trên, em sẽ trình bày tiếp ở mục 1.2 và 1.3.

1.2 Hệ thống hội thoại

1.2.1 Tổng quan

Một hệ thống hội thoại là một chương trình máy tính có khả năng tương tác với người dùng bằng các cách giao tiếp tự nhiên [5]. Hệ thống hội thoại cung cấp một giao diện giữa người dùng và ứng dụng máy tính cho phép người dùng tương tác theo cách tự nhiên với ứng dụng. Các hệ thống hội thoại khác nhau có các kiến trúc khác nhau, tuy nhiên có thể khái quát hóa thành sáu thành phần sau: (i) chuyển hóa thông tin đầu vào (Input Decoder), (ii) hiểu ngôn ngữ tự nhiên (Natural Language Understanding), (iii) quản lý hội thoại (Dialogue Manager), (iv) thành phần của từng lĩnh vực cụ thể (Domain Specific Component), (v) sinh thông tin phản hồi và (vi) chuyển hóa thông tin đầu ra [5].



Hình 1 Các thành phần của hệ thống hội thoại [5].

Hình 1 miêu tả vị trí và thứ tự xử lý của các thành phần trong một hệ thống hội thoại, chi tiết chức năng nhiệm vụ của từng thành phần được mô tả dưới đây.

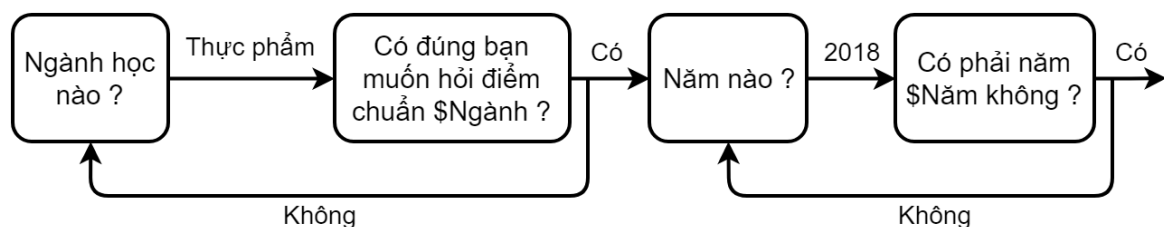
Input Decoder có nhiệm vụ nhận diện và chuyển đổi thông tin đầu vào thành dạng văn bản. Thông thường thành phần này có ở các hệ thống hội thoại với dữ liệu đầu vào không chỉ ở dạng văn bản, mà có thể ở dạng giọng nói, cử chỉ hoặc chữ viết tay. Hệ thống thực hiện nhiệm vụ chuyển đổi văn bản thành giọng nói thường được gọi là Automatic Speech Recognition (ASR), Computer Speech Recognition hoặc đơn giản là Speech to Text (STT). Sau khi nhận thông tin dạng văn bản từ Input Decoder, Language Understanding có nhiệm vụ hiểu ý định mà người dùng muốn bày tỏ. Thành phần này chuyển dãy các từ trong câu thành dạng biểu diễn ngữ nghĩa để làm đầu vào cho thành phần quản lý hội thoại. Thành phần quản lý hội thoại quản lý tất cả các sự kiện diễn ra trong một đoạn hội thoại. Thành phần này lấy dạng biểu diễn ngữ nghĩa từ NLU, sau đó dựa trên ngữ cảnh của cuộc hội thoại để tính toán ra biểu diễn ngữ nghĩa cho phản hồi của hệ thống sao cho thích hợp. Thành phần quản lý hội thoại là một trong những thành phần quan trọng bậc nhất, quyết định độ thông minh của hệ thống và giúp tạo ra những đoạn hội thoại tự nhiên. Thành phần quản lý hội thoại thường truy cập hoặc tương tác với các thành phần nằm ngoài hệ thống hội thoại như cơ sở dữ liệu, các hệ thống bên ngoài. Các câu truy vấn này vì vậy phải được chuyển

đổi từ định dạng bên trong hệ thống hội thoại thành định dạng sử dụng bởi các hệ thống cụ thể ở ngoài. Đây là nhiệm vụ chính của Domain Specific Component. Sau khi thành phần quản lý hội thoại dự đoán ra các biểu diễn ngữ nghĩa phù hợp nhất, Response Generator dựa vào các biểu diễn ngữ nghĩa đó sẽ xây dựng ra các phản hồi dạng văn bản gửi lại cho người dùng, cùng đó quyết định các thông tin cần đưa vào câu phản hồi, cách chọn từ cũng như cấu trúc của câu phản hồi. Hiện nay, các hệ thống hội thoại thường lựa chọn phương pháp đơn giản là chèn thêm các thông tin động (các thông tin thu thập được trong quá trình hội thoại) vào chỗ trống trong các mẫu câu phản hồi được định nghĩa sẵn. Cuối cùng Output Renderer chuyển đổi các câu phản hồi dạng văn bản lấy từ Response Generator thành dạng giọng nói. Có hai cách tiếp cận, cách tiếp cận đầu tiên là sử dụng các câu nói được ghi âm từ trước, cách này cho chất lượng giọng nói rất tự nhiên nhưng lại hạn chế về số lượng nội dung phản hồi và khả năng tùy chỉnh câu phản hồi. Cách tiếp cận thứ hai là sử dụng các kỹ thuật chuyển đổi văn bản thành giọng nói (Text to Speech - TTS).

1.2.2 Phân loại

1.2.2.1 Hệ thống dựa trên trạng thái hữu hạn (Finite State based Systems)

Trong các hệ thống dựa trên trạng thái hữu hạn [5], cuộc hội thoại là một dãy các bước hoặc trạng thái (state) được định nghĩa từ trước. Cấu trúc hội thoại có thể được biểu diễn bằng một đồ thị mà các nút biểu diễn các câu hỏi của hệ thống và các cạnh nối giữa các nút biểu diễn tất cả các đường đi có thể có trong đồ thị. Hình 2 mô tả một phần của đồ thị hội thoại bao gồm bốn nút và các cạnh nối tương ứng.



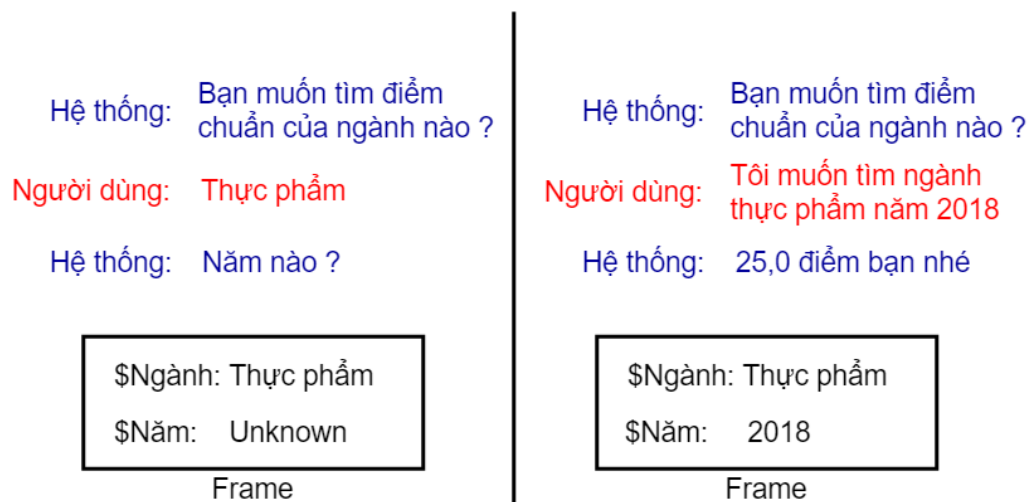
Hình 2 Ví dụ đồ thị hội thoại trong hệ thống tư vấn tuyển sinh.

Theo đồ thị trong Hình 2, hệ thống đưa ra câu hỏi về thông tin tên ngành, sau đó đưa ra câu hỏi kiểm chứng. Nếu người dùng trả lời là “có” thì hệ thống tiếp tục đưa ra câu

hỏi tiếp theo về năm, còn ngược lại nếu người dùng trả lời là “không” thì hệ thống sẽ hỏi lại tên ngành học. Quá trình này sẽ tiếp tục diễn ra cho tới khi các nút thuộc phần đồ thị này được đi qua hết. Các thông tin về ngành học hay năm được lưu vào các biến “\$Ngành” và “\$Năm” và được hệ thống sử dụng để tạo ra câu hỏi cho người dùng.

Các hệ thống dựa trên trạng thái hữu hạn có ưu điểm là xây dựng rất đơn giản, phù hợp với các tác vụ thu thập thông tin từ người dùng để thực hiện mục tiêu nào đó. Tuy nhiên, hệ thống lại làm các cuộc hội thoại thiếu đi tính linh hoạt, người dùng sẽ không thể thay đổi lại các thông tin đã cung cấp phía trước hoặc cung cấp các thông tin không được hệ thống hỏi ở nút hiện tại, làm mất đi tính tự nhiên của cuộc hội thoại.

1.2.2.2 Hệ thống dựa trên khung (Frame Based Systems)



Hình 3 Ví dụ các đoạn hội thoại trong hệ thống dựa trên khung.

Các hệ thống dựa trên khung [5] khá tương đồng với các hệ thống dựa trên trạng thái hữu hạn giới thiệu ở phần 1.2.2.1 do cùng phù hợp với các bài toán form-filling. Tuy nhiên, các hệ thống này không thu thập các thông tin từ người dùng theo một trình tự đã được định nghĩa từ trước, mà hệ thống sẽ quyết định hành động tiếp theo dựa trên nội dung thông tin mà người dùng đã và đang cung cấp. Các thông tin cần thu thập trong cuộc hội thoại được lưu vào một khung (frame), với các giá trị tương ứng là giá trị đã được người dùng cung cấp, và “Unknown” trong trường hợp còn lại. Hình 3

đưa ra ví dụ về hai luồng hội thoại khác nhau do thông tin người dùng cung cấp khác nhau.

Ta có thể thấy sự khác biệt rõ ràng giữa hai đoạn hội thoại, đoạn hội thoại đầu tiên khá giống trong hệ thống dựa trên trạng thái hữu hạn, hệ thống kiểm tra trong Frame thấy thông tin về năm chưa được thu thập, nên đưa ra câu hỏi về năm tương ứng. Trong đoạn hội thoại thứ hai, khi người dùng trả lời, ngoài cung cấp thông tin tên ngành bị hỏi, người dùng còn cung cấp luôn thông tin về năm. Hai thông tin này được lưu vào Frame, nhận thấy không còn thông tin nào cần thu thập nữa, hệ thống thực hiện truy xuất thông tin điểm chuẩn và phản hồi lại cho người dùng.

Loại hệ thống này đã cho phép các đoạn hội thoại trở lên tự nhiên hơn, cho phép người dùng cung cấp một lúc nhiều thông tin, và thay đổi thông tin đã cung cấp phía trước. Tuy nhiên vẫn không thể xử lý các đoạn hội thoại quá phức tạp.

1.2.2.3 Hệ thống dựa trên tác tử (Agent Based Systems)

Hai dạng hệ thống giới thiệu ở 1.2.2.1 và 1.2.2.2 do đơn giản nên khiến cuộc hội thoại không được tự nhiên và không coi trọng yếu tố ngữ cảnh của cuộc hội thoại. Với một cách tiếp cận khác, các hệ thống dựa trên tác nhân (Agent Based Systems) [5] coi quá trình hội thoại là quá trình tương tác giữa hai tác nhân với nhau, tác nhân ở đây có thể là người dùng hoặc hệ thống, mỗi tác nhân đều có khả năng tự lập luận và đưa ra các hành động của riêng mình. Cuộc hội thoại được xây dựng một cách linh hoạt bởi một dãy các hành động của hai tác nhân.

Thông thường, thành phần quản lý hội thoại của hệ thống này bao gồm hai thành phần chính là thành phần theo dõi trạng thái hội thoại (Dialogue State Tracking - DST) và thành phần chính sách hội thoại (Dialogue policy learning) [9] sẽ được phát triển.

Dialogue State Tracking

Thành phần này còn được gọi là state tracker, có nhiệm vụ lưu trữ thông tin trạng thái hội thoại (state) và đánh giá mục tiêu nhắm tới của người dùng ở mỗi lượt của một cuộc hội thoại. Trạng thái hội thoại H_t miêu tả một biểu diễn của phiên hội thoại tính

tới thời điểm t . Các thông tin một trạng thái hội thoại lưu trữ có thể là hành động gần nhất của người dùng, hành động gần nhất của hệ thống, các thông tin người dùng vừa cung cấp, các thông tin được cung cấp từ đầu đoạn hội thoại,...

Dialogue policy learning

Dựa vào các biểu diễn thông tin trạng thái hội thoại từ state tracker, các chính sách hội thoại sẽ tạo ra hành động tiếp theo cho hệ thống sao cho phù hợp với ngữ cảnh hiện tại của đoạn hội thoại nhất.

Ưu điểm của loại hệ thống dựa trên tác nhân là lấy yếu tố ngữ cảnh của hội thoại để dự đoán ra hành động phù hợp nhất cho tác nhân hệ thống, từ đó xử lý được những luồng hội thoại phức tạp và tăng tính tự nhiên cho cuộc hội thoại. Mặt khác, nhược điểm của những hệ thống dạng này là xây dựng khá khó khăn và cần nhiều kỹ thuật phức tạp. Hiện nay khi học máy đang phát triển rất nhanh và mạnh, ngày càng có nhiều hướng tiếp cận mới ứng dụng học máy và đặc biệt là học sâu nhằm xây dựng các hệ thống dạng này một cách hiệu quả, một số cách tiếp cận mới sẽ được giới thiệu ở trong mục 1.3.

1.3 Mô hình quản lý hội thoại và các nghiên cứu liên quan

Với những hệ thống quản lý hội thoại, học sâu (deep learning) có thể tận dụng một số lượng lớn dữ liệu để tự học được các biểu diễn đặc trưng từ những đoạn hội thoại và đưa ra chiến lược phản hồi lại mà không yêu cầu quá nhiều những tinh chỉnh thủ công từ phía người huấn luyện.

1.3.1 Mô hình luật thủ công (Hand-crafted Rules Dialogue Systems)

Các hệ thống hội thoại đã được phát triển từ rất lâu, bắt đầu từ những hệ thống cổ điển sử dụng luật thủ công để quản lý hội thoại. Trong [10] giới thiệu một nền tảng để phát triển hệ thống quản lý hội thoại dựa trên những luật được định nghĩa từ trước (hand-crafted rules). Nền tảng này được xây trên hai khái niệm là trạng thái thông tin (information state) và phương pháp hội thoại di chuyển (dialogue move engine). Các thay đổi trong cuộc hội thoại (dialogue moves) sẽ kích hoạt các cập nhật đối với

information state, các cập nhật này được thành phần chiến lược (update strategy) lựa chọn trong một tập các luật cập nhật cho trước. Với cách tiếp cận tương tự, [11] đã xây dựng một Chatbot có khả năng cung cấp dự báo thông tin thời tiết. Việc xây dựng các luật thủ công cũng tương tự như xây dựng các nhánh mã lệnh dạng if-else để quản lý luồng hội thoại. Ưu điểm của cách tiếp cận này là giúp dễ dàng xây dựng được hệ thống hội thoại một cách nhanh chóng, đây cũng là một trong những cách tiếp cận được ứng dụng nhiều trong các nền tảng hội thoại thương mại hiện nay. Tuy nhiên khi số lượng kịch bản hội thoại gia tăng, các điều kiện luật trở nên phức tạp, sẽ gây ra khó khăn lớn cho quá trình mở rộng và bảo trì hệ thống. Những cách tiếp cận hiện đại hơn nhằm cải tiến hệ thống hội thoại sẽ được giới thiệu qua bốn định hướng nghiên cứu được trình bày ở phần kế tiếp.

1.3.2 Mô hình giám sát hoàn toàn (Fully-Supervised Models)

Ta có thể xây dựng một hệ thống hội thoại theo cách tiếp cận đầu-cuối (end-to-end) hay phương pháp học giám sát sequence-to-sequence [12]. Nhờ đó, ta có thể sử dụng sức mạnh của mạng nơ-ron sâu trên nguyên tắc mã hóa-giải mã (encoder-decoder principle) để học được các biểu diễn của trạng thái hội thoại. Tuy nhiên hướng tiếp cận này yêu cầu một số lượng đáng kể dữ liệu để đưa ra được mô hình tốt.

Các tác giả trong [13] sử dụng mạng nơ-ron hồi quy (RNN) để huấn luyện Chatbot trực tiếp theo phương pháp sequence-to-sequence. Bài báo tiến hành huấn luyện trên bộ dữ liệu các đoạn hội thoại hỏi đáp liên quan tới lĩnh vực công nghệ thông tin, trong đó người dùng gặp những vấn đề liên quan đến máy tính và có chuyên gia giúp đỡ họ tìm kiếm những giải pháp. Do mô hình RNN không có khả năng nhớ các thông tin phụ thuộc dài (đề cập tới ở mục 2.3) nên khả năng phản hồi theo ngữ cảnh của Chatbot này vẫn còn chưa tốt.

Để giải quyết được hạn chế của RNN, trong [14] đã sử dụng nhân LSTM để xây dựng một Chatbot có chức năng tiếp nhận các đơn đặt chỗ nhà hàng. Hệ thống được mô tả là có khả năng suy luận tốt hơn, có thể ghi nhớ và cập nhật các tùy chọn của khách hàng.

1.3.3 Mô hình học tăng cường (Reinforcement Learning-based Models)

Một nhánh nghiên cứu khác cũng đang rất phát triển đó là tập trung vào kỹ thuật Deep Reinforcement Learning (RL) do việc huấn luyện các Fully-Supervised Models cần quá nhiều lượng dữ liệu để đưa ra được kết quả tốt. Các mô hình này có một cấu trúc khá phức tạp, xoay quanh các thành phần như Natural Language Understanding (NLU), Natural Language Generation (NLG) và Dialogue State Tracker (DST).

Dựa trên định hướng này, [15] xây dựng một hệ thống hội thoại cho lĩnh vực đặt chỗ trong nhà hàng. Các hành động của hệ thống chỉ phụ thuộc vào tác nhân dựa trên RL, bằng cách thức hiện lựa chọn hành động trực tiếp từ văn bản thô của người dùng cuối và phản hồi của hệ thống. Hệ thống này không bao gồm thành phần NLU hay DST, do đó nên khá hạn chế. Một Chatbot trả lời câu hỏi (Q&A bot) khác được giới thiệu trong [16], cũng là một end-to-end RL Chatbot, giúp người dùng tìm kiếm các tri thức về phim ảnh mà không cần soạn các câu truy vấn phức tạp.

Mở rộng hơn cho các nghiên cứu trên, [17] đã xây dựng hệ thống Chatbot đặt vé xem phim. Hệ thống này bao gồm một giả lập người dùng (User Simulator) [18] có chức năng giả lập hành động của người dùng trong quá trình huấn luyện, thành phần NLU, NLG và các mô-đun quản lý hội thoại. Hệ thống này cũng được huấn luyện theo phương pháp end-to-end.

1.3.4 Mô hình lai (Hybrid Models)

Hướng nghiên cứu này mong muốn kết hợp được những ưu điểm của hai cách tiếp cận trên. Trong [19], các tác giả đưa ra một phương pháp huấn luyện Chatbot chia thành hai bước. Đầu tiên, ta tiến hành huấn luyện mô hình theo phương pháp giám sát bằng bộ dữ liệu cố định chuẩn bị trước. Tiếp theo, mô hình sẽ được fine-tuned [20] bằng cách sử dụng kỹ thuật RL-based policy gradient [21], nhằm cải thiện chất lượng của mô hình.

Tương tự, [22] giới thiệu giải pháp huấn luyện hệ thống hội thoại có tên Hybrid Code Networks (HCNs) với hai chế độ: chế độ ngoại tuyến (off-line mode) và chế độ trực tuyến (on-line mode). Trong chế độ off-line, mô hình được huấn luyện theo phương

pháp giám sát hoàn toàn, kết hợp mạng LSTM với các mẫu hành động của hệ thống (system action templates) được định nghĩa trước. Sau đó, với chế độ on-line, hệ thống tự động học bằng cách sử dụng RL-based policy gradient. So sánh với các cách tiếp cận end-to-end trước đây, mô hình HCNs giảm một lượng đáng kể dữ liệu huấn luyện cần thiết, trong khi vẫn giữ được lợi ích của việc suy luận được các biểu diễn ẩn của trạng thái hội thoại.

1.3.5 Mô hình ràng buộc dữ liệu (Data-Constrained Dialogue Systems)

Một thuộc tính hữu hiệu mà các Chatbot đều mong muốn là khả năng chuyển sang nhiều domain (lĩnh vực) mới mà không mất đi những tri thức đã học được từ những lần huấn luyện trước. Khả năng này rất cần thiết do sự thiếu hụt dữ liệu huấn luyện để có thể tạo ra một Chatbot chất lượng cao cho các lĩnh vực mới đó.

Các tác giả trong [23] đề xuất một kỹ thuật dựa trên Quy trình Gaussian để học các dialogue policy (chính sách hội thoại) chung chung, tổ chức theo hệ thống phân cấp lớp. Các policy này với lượng dữ liệu khiêm tốn có thể được điều chỉnh thêm tùy theo các use case của hệ thống hội thoại.

Mục đã cho ta cái nhìn tổng quát về các nghiên cứu hiện nay, hầu hết đều tập trung vào ứng dụng các phương pháp học máy vào quản lý hội thoại. Mỗi cách tiếp cận đều có các ưu nhược điểm riêng. Việc lựa chọn phương pháp, mô hình phù hợp với nhu cầu và phạm vi của đề án sẽ được trình bày chi tiết ở Chương 3.

1.4 Mục tiêu và phạm vi đề tài

Từ những tình hình thực tế đó, nền tảng hội thoại thông minh SmartDialog được ra đời. SmartDialog là một nền tảng hội thoại có thể sử dụng bởi các cá nhân, doanh nghiệp để tạo ra hệ thống hội thoại, tương tác với khách hàng một cách dễ dàng. Mục tiêu của đề án này là nghiên cứu, đưa ra được giải pháp cho bài toán quản lý hội thoại, sau đó phát triển và triển khai trong nền tảng hội thoại thông minh SmartDialog. Giải pháp đưa ra cần phải giải quyết được vấn đề của các mô hình thường gặp ở hiện tại. Hệ thống hội thoại phải có khả năng tương tác với người dùng cuối một cách tự nhiên,

tạo cảm giác đang thực sự “chat” với một người khác, chứ không phải đi theo những luồng hội thoại cố định.

Đồng thời, do những người sử dụng SmartDialog là những người dùng không chuyên nên giải pháp đưa ra cần có cách tiếp cận đơn giản, trực quan. Các cách thức và phương pháp huấn luyện cần phải dễ hiểu và không quá mang tính kỹ thuật, yêu cầu nhiều kiến thức nền tảng về hệ thống hội thoại.

Hơn thế nữa, để có thể tiếp cận tới tập người dùng rộng hơn, đề án cũng sẽ đưa ra giải pháp tích hợp nền tảng SmartDialog với nền tảng chat Zalo, cùng với đó đề xuất cách thức quản lý các luồng hội thoại của người dùng cuối giữa hai nền tảng này.

1.5 Định hướng giải pháp

Đề án này đề xuất một mô hình quản lý hội thoại dựa trên nền tảng sử dụng các mô hình học sâu, tuy nhiên không theo hướng tiếp cận end-to-end. Hai thành phần hiểu ngôn ngữ tự nhiên và thành phần quản lý hội thoại được tách rời thành hai phần riêng biệt và được huấn luyện một cách độc lập. Thành phần quản lý hội thoại sẽ được đào tạo trên tập dữ liệu là những kịch bản hội thoại đã được gán nhãn. Cách tiếp cận này sẽ giúp hai thành phần trên có thể được sử dụng một cách độc lập với nhau, giảm số lượng kịch bản cần thiết khi huấn luyện và tạo ra khả năng tái sử dụng mô hình quản lý hội thoại cho nhiều ngôn ngữ đầu vào và đầu ra khác nhau.

Để giúp người dùng tiếp cận dễ dàng hơn với nền tảng, giải pháp được đưa ra là sử dụng phương pháp học tương tác (interactive learning) để giúp những người dùng không chuyên dễ dàng tạo thêm dữ liệu huấn luyện đầu vào và huấn luyện mô hình, dạy cho hệ thống cách nói chuyện qua tương tác trực tiếp với nó.

Tiếp theo, để tích hợp hệ thống hội thoại trên SmartDialog với Zalo, em thực hiện xây dựng máy chủ Gateway nhằm kết nối và quản lý theo phiên (session) những đoạn hội thoại của người dùng cuối từ nền tảng Zalo tới SmartDialog.

Từ những giải pháp đã đưa ra, đề án triển khai thành công mô hình quản lý hội thoại trong nền tảng SmartDialog, từ đó xây dựng được hệ thống hội thoại có khả năng

tương tác tự nhiên với người dùng cuối. Thực hiện triển khai case study Chatbot Tư vấn tuyển sinh HUST với bộ dữ liệu được đồng ý cung cấp bởi bộ phận tuyển sinh của trường Đại học Bách Khoa Hà Nội.

1.6 Bố cục đề án

Những phần còn lại của báo cáo đề án tốt nghiệp này sẽ được tổ chức như sau.

Trong Chương 2, em trình bày tổng quan về học máy và mạng nơ-ron nhân tạo, giải thích khả năng ghi nhớ thông tin theo thời gian dài của mạng LSTM. Sau đó giới thiệu phương pháp học trực tuyến và sự khác biệt của nó so với các phương pháp học thông thường.

Chương 3 giới thiệu giải pháp đề xuất để giải quyết bài toán quản lý hội thoại, giảm những hạn chế từ các cách tiếp cận rule-based hoặc end-to-end. Chương này cũng trình bày về phương pháp học tương tác dựa trên ý tưởng của học trực tuyến, cho khả năng huấn luyện mô hình bằng cách tương tác giữa nhà phát triển và hệ thống. Cuối cùng đưa ra cách thức và tiến hành đánh giá độ tốt của giải pháp.

Tiếp theo, Chương 4 sẽ trình bày các giải pháp thiết kế, xây dựng, triển khai phân hệ quản lý hội thoại trong nền tảng SmartDialog. Bao gồm dịch vụ dự đoán hành động, dịch vụ huấn luyện hành động và máy chủ Gateway quản lý phiên hội thoại, cùng với đó triển khai tích hợp Chatbot Tư vấn tuyển sinh HUST trên nền tảng Zalo.

Và cuối cùng, Chương 5 sẽ kết luận lại nội dung của đề án và đưa ra những hướng hoàn thiện thành phần quản lý hội thoại hiện tại, sau đó đưa ra định hướng nghiên cứu và phát triển tiếp theo trong tương lai.

Sau đây, em sẽ đi vào chi tiết từng phần của đề án.

Chương 2 Cơ sở lý thuyết

Chương 2 này giới thiệu cơ sở lý thuyết về học máy, mạng nơ-ron nhân tạo, các mô hình học sâu mở rộng ra từ mạng nơ-ron nhân tạo: (i) mạng nơ-ron hồi quy (RNN), (ii) mạng Long Short-Term Memory (LSTM). Mạng RNN và LSTM là các mô hình học sâu thường sử dụng và thu được nhiều kết quả tốt khi giải quyết các bài toán trong lĩnh vực xử lý ngôn ngữ tự nhiên. Chương này cũng giới thiệu khái niệm, giải thích cách hoạt động của phương pháp học trực tuyến, đây là phương pháp .

2.1 Giới thiệu về học máy

Học máy (machine learning) [24] là một nhánh con của lĩnh vực trí tuệ nhân tạo, đã trở thành cụm từ khóa phổ biến trong những năm vừa qua. Mục tiêu của học máy, giống như tên gọi của nó, là giúp máy tính có khả năng tự học hỏi được tri thức thông qua dữ liệu đầu vào mà không cần phải được lập trình cụ thể bằng bàn tay của con người.

Dựa trên các phương thức học, các thuật toán học máy được chia thành bốn nhóm chính: học có giám sát (supervised learning), học không giám sát (unsupervised learning), học bán giám sát (semi-supervised learning) và học tăng cường (reinforcement learning) [25].

Học có giám sát là phương pháp xây dựng mô hình tính toán để giải quyết bài toán dựa trên tập dữ liệu bao gồm cả đầu vào và đầu ra mong muốn tương ứng. Tập dữ liệu huấn luyện bao gồm các cặp dữ liệu và nhãn tương ứng. Bài toán phân lớp (classification) và hồi quy (regression) là hai bài toán chủ yếu sử dụng phương pháp này.

Ngược lại, học không giám sát là phương pháp xây dựng mô hình tính toán để giải quyết bài toán dựa trên tập dữ liệu chỉ bao gồm đầu vào mà không bao gồm đầu ra

tương ứng. Phương pháp chủ yếu dựa vào cấu trúc dữ liệu để giải quyết bài toán phân nhóm (clustering) và tìm ra quy luật của dữ liệu (association).

Phương pháp học bán giám sát nằm giữa hai nhóm phương pháp phía trên, tuy tập dữ liệu đầu vào rất lớn nhưng chỉ một phần trong tập dữ liệu đó được gán nhãn, nguyên nhân có thể là do việc gán nhãn tốn nhiều chi phí và thời gian, yêu cầu tri thức từ chuyên gia. Do đó phương pháp này được ứng dụng rất nhiều trong thực tế.

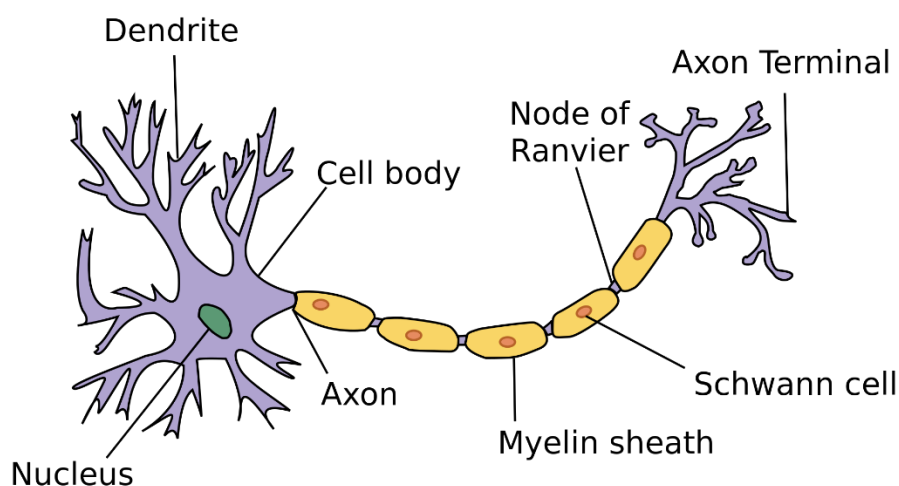
Cuối cùng, phương pháp học tăng cường xây dựng mô hình bằng cách giúp cho chúng có thể tự đưa ra hành động dựa theo ngữ cảnh, sao cho đạt được lợi ích lớn nhất. Các tác động trở lại của môi trường sau hành động đó là cơ sở để mô hình đánh giá và cập nhật.

Trong đồ án này, bài toán quản lý hội thoại được đưa về dạng bài toán phân loại và giải quyết theo hướng tiếp cận sử dụng phương pháp học có giám sát.

2.2 Mạng nơ-ron nhân tạo

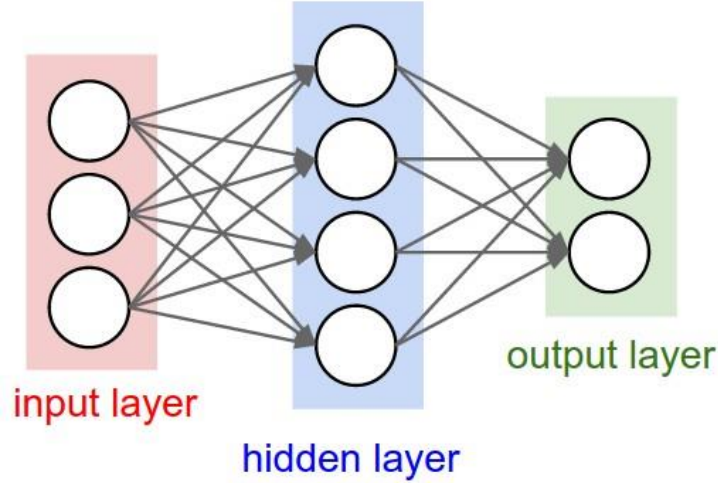
2.2.1 Kiến trúc

Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là một mô hình học máy lấy ý tưởng từ cách thức xử lý thông tin của bộ não sinh học, gồm số lượng lớn các nơ-ron thần kinh được gắn kết với nhau nhằm truyền dẫn các tín hiệu thần kinh.



Hình 4 Tế bào thần kinh sinh học [1].

Giống như các tế bào thần kinh sinh học (Hình 4) có các đuôi gai (dendrite) để nhận tín hiệu, vỏ tế bào (cell body) để xử lý tín hiệu, và sợi trục (axon) để gửi tín hiệu tới các tế bào thần kinh khác, tế bào thần kinh nhân tạo cũng có một số kênh đầu vào, xử lý thông tin và đưa qua hàm kích hoạt, và một đầu ra có thể gửi thông tin đến nhiều các tế bào thần kinh nhân tạo khác.



Hình 5 Mạng tế bào thần kinh nhân tạo [26].

Mạng nơ-ron nhân tạo được mô tả ở Hình 5, được xây dựng từ các tầng nơ-ron liên kết với nhau, mỗi tầng có thể có một hoặc nhiều nơ-ron. Kiến trúc chung của một ANN bao gồm ba loại tầng là tầng vào, tầng ẩn và tầng ra. Một mạng ANN có thể có một hoặc nhiều tầng ẩn, nhưng chỉ có duy nhất một tầng vào và một tầng ra. Số lượng nơ-ron ở mỗi tầng có thể khác nhau tùy theo các bài toán riêng biệt.

$$z_j^k = \sum_i w_{i,j}^k \times a_i^{k-1} + b^k$$

$$a_j^k = f(z_j^k)$$

Công thức 1 Tính giá trị của nơ-ron ở mỗi tầng.

Giả sử n là số lượng nơ-ron ở tầng vào, tập $X = \{x_1, x_2, \dots, x_n\}$ là tập các giá trị đầu vào của tầng vào. Các giá trị này sẽ được lan truyền tới các nơ-ron ở lớp kế tiếp qua các cạnh nối, gọi $w_{i,j}^k$ là trọng số của cạnh nối nơ-ron thứ i thuộc tầng $k - 1$ với nơ-ron thứ j thuộc tầng k , quá trình này được gọi là suy luận tiến (feedforward) [24].

Công thức tính giá trị của mỗi nơ-ron thuộc tầng k được tính dựa trên giá trị các nơ-ron tầng trước đó và trọng số của các cạnh nối tương ứng.

Trong Công thức 1, a_j^k là giá trị nơ-ron thứ j của tầng k , f là hàm kích hoạt và b được gọi là các bias (độ lệch). Riêng với tầng vào, thông thường các $a^{(1)}$ cũng chính bằng các đầu vào x tương ứng của mạng. Mục tiêu của hàm kích hoạt là làm phi tuyến hóa các tín hiệu đầu vào, giúp việc tạo mô hình gồm nhiều lớp ẩn có ý nghĩa và thể hiện ra được các hàm số phức tạp hơn. Một số các hàm kích hoạt thông dụng có thể kể đến là sigmoid, ReLU, tanh, Maxout [27],... Quá trình một mạng nơ-ron được huấn luyện chính là quá trình mạng tự điều chỉnh các tham số w sao cho thỏa mãn tập dữ liệu huấn luyện được gán nhãn.

2.2.2 Tối ưu hóa tham số

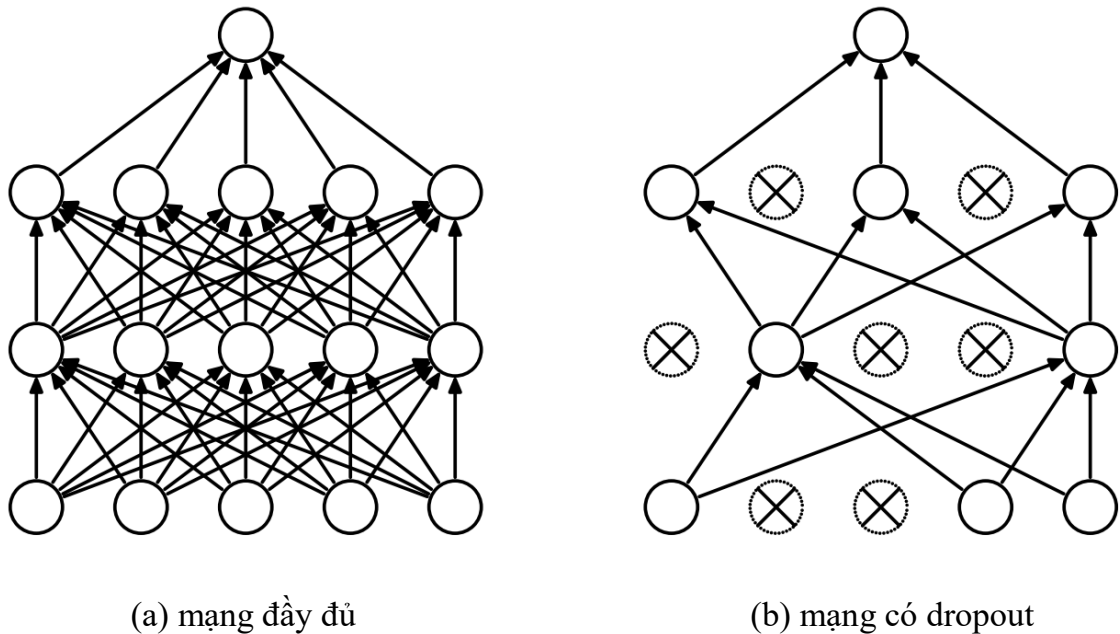
Mạng nơ-ron nhân tạo có khả năng mô phỏng một hàm số phức tạp dựa vào việc tự điều chỉnh các tham số w của mạng. Mục tiêu của việc huấn luyện mạng là tạo ra được một hàm mô phỏng càng tiệm cận với hàm mục tiêu càng tốt. Quá trình huấn luyện gồm hai bước chính lặp đi lặp lại là (i) tính toán sự khác biệt giữa hàm mô phỏng và hàm đích và (ii) tinh chỉnh tham số w để giảm bớt sự khác biệt đó.

Sự khác biệt giữa hàm mô phỏng và hàm đích được tính toán dựa trên hàm mất mát (loss function) [26, p. 23]. Hàm mất mát có rất nhiều dạng khác nhau nhằm phù hợp với từng dạng bài toán và thuật toán khác nhau, tuy nhiên các hàm này đều liên tục và khả vi. Một số hàm mất mát thông dụng như hàm Cross-Entropy [25], hàm CTC, hàm Triplet-loss, hàm khoảng cách Euclid [24],...

Quá trình tinh chỉnh tham số w của mô hình diễn ra nhằm giảm bớt sự khác biệt giữa hàm mô phỏng và hàm đích hay cực tiểu hóa giá trị của hàm mất mát. Dựa trên giá trị của hàm mất mát, các tín hiệu mất mát của từng giá trị nơ-ron tạo ra được tính trên chính lỗi mà nơ-ron đó gây ra cho giá trị đầu ra. Các tín hiệu mất mát lan truyền về cho các nơ-ron ở lớp trước đồng thời cập nhật giá trị tham số w trong quá trình lan truyền ngược về. Đây gọi là giải thuật lan truyền ngược (backpropagation) [26, p. 23].

Hai quá trình suy luận tiến và lan truyền ngược được tiến hành liên tục lặp đi lặp lại để cập nhật tham số w giúp giá trị của hàm mất mát đạt gần tới điểm cực tiểu nhất có thể, giúp mô hình mô phỏng được hàm số gần với hàm mục tiêu nhất. Tên một số các thuật toán tối ưu tham số có thể kể tới là Stochastic Gradient Descent [28], Nesterov accelerated gradient [29], Adagrad [30], RMSprop [28], Adam [31],...

2.2.3 Vấn đề Overfitting và phương pháp giải quyết



Hình 6 So sánh mạng nơ-ron đầy đủ và mạng có dropout [32].

Các mô hình mạng nơ-ron với các cấu trúc nơ-ron nhiều tầng cho khả năng học được các mối quan hệ (đặc trưng) rất phức tạp giữa các dữ liệu đầu vào và dữ liệu đầu ra. Tuy nhiên với lượng dữ liệu huấn luyện hạn chế, các quan hệ phức tạp đó có thể gây ra nhiễu cho mô hình, vì chúng tồn tại trong dữ liệu huấn luyện nhưng lại không tồn tại trong các dữ liệu thực tế, từ đó làm giảm độ chính xác của mô hình. Vấn đề trên trong học máy được gọi là overfitting, khi mô hình học quá khớp với các đặc trưng của dữ liệu trong tập huấn luyện mà không nắm bắt được các đặc trưng tổng quát, dẫn đến độ chính xác thấp khi nhận đầu vào là các dữ liệu mới trong thực tế.

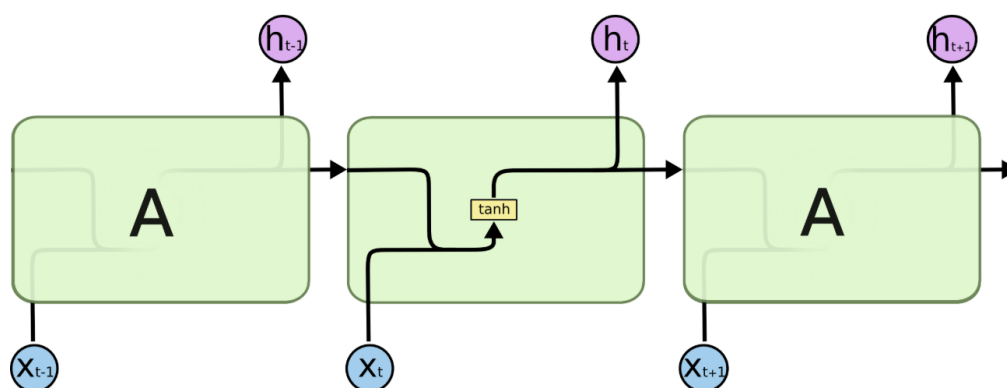
Có một kỹ thuật giải quyết overfitting đơn giản mà lại tỏ ra hiệu quả là kỹ thuật Dropout [32].

Hình 6 mô tả cách Dropout tắt bớt một số các nơ-ron trong mạng một cách ngẫu nhiên, bao gồm cả các kết nối đầu vào và kết nối đầu ra của chúng. Tại mỗi bước huấn luyện, khi tín hiệu lan truyền tới từng tầng, mỗi nơ-ron của tầng đó sẽ có xác suất giữ lại là p và xác suất bị lược bỏ khỏi mạng là $1 - p$. Dropout tạo ra một mạng nơ-ron “thưa” hơn, giảm bớt thông tin học được từ các mối quan hệ phức tạp của dữ liệu, từ đó giảm overfitting.

2.3 Mạng nơ-ron nhân tạo Long Short-Term Memory

2.3.1 Mạng nơ-ron hồi quy (recurrent neural network - RNN)

Trong nhiều nhiều trường hợp, để dự đoán hoặc giải quyết một vấn đề nào đó ta cần phải dựa vào các sự kiện đã xảy ra từ trước đó. Ví dụ khi xem một bộ phim hoặc đọc một quyển truyện, để hiểu được nội dung ta cần theo dõi cả những phần nội dung trước đó chứ không phải chỉ xem một khung hình hoặc đọc một trang sách duy nhất. Trong khi đó, mô hình mạng nơ-ron truyền thống giới thiệu ở phần 2.2 có đầu vào vào đầu ra độc lập với nhau, không có khả năng để giải quyết vấn đề trên.



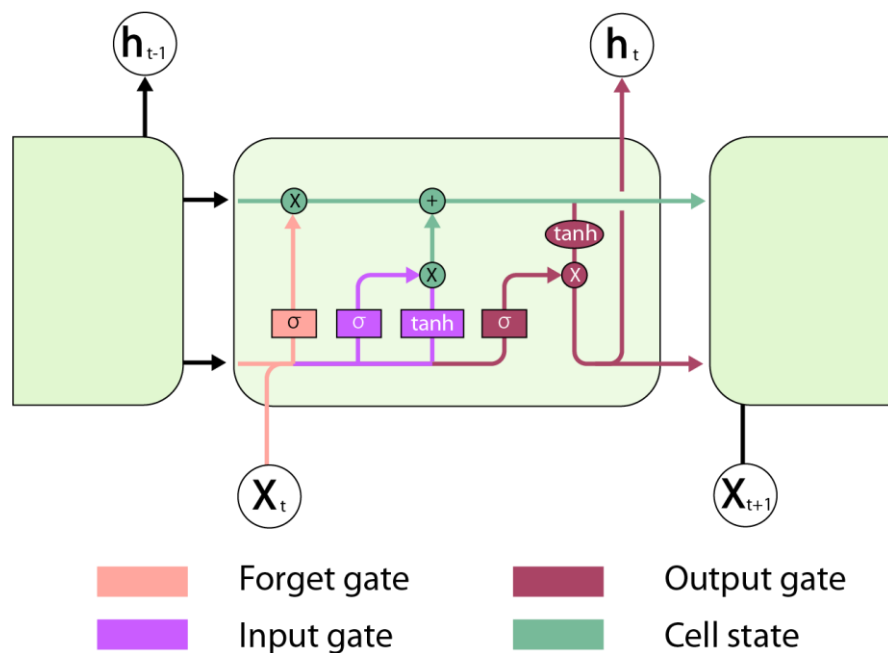
Hình 7 Các mô-đun lặp chứa nhân \tanh trong mạng nơ-ron hồi quy [33].

Do đó RNN ra đời, đây là một mạng nơ-ron gồm nhiều vòng lặp cho phép lưu trữ thông tin theo thời gian.

Trong sơ đồ cấu tạo của RNN được mô tả ở Hình 7, tại mỗi bước thời gian t , nhân A của mạng nơ-ron nhận thông tin đầu vào x_t kết hợp với kết quả đầu ra h_{t-1} của bước thời gian phía trước để tính toán kết quả h_t , rồi sau đó chuyển kết quả này sang bước

thời gian tiếp theo. Theo lý thuyết, việc kết hợp với thông tin đầu ra ở bước thời gian trước trong quá trình tính toán khiến RNN có khả năng ghi nhớ thông tin theo thời gian rất tốt. Tuy nhiên trong thực tế, RNN không có khả năng ghi nhớ các thông tin có ý nghĩa của các bước thời gian xa phía trước, còn gọi là vấn đề phụ thuộc dài hạn (long-term dependencies). Vấn đề này được giải quyết với sự ra đời của mạng Long Short-Term Memory.

2.3.2 Mạng Long Short-Term Memory (LSTM)



Hình 8 Các mô-đun lặp trong một mạng LSTM với cấu trúc ba cổng [16].

Khác với mạng RNN chỉ bao gồm một lớp \tanh ở trong nhân (Hình 7), Hình 8 mô tả mạng Long Short-Term Memory [34] cải tiến từ mạng RNN với bốn lớp trong nhân tương tác với nhau ở mỗi bước lặp thời gian. Điểm mấu chốt của LSTM là có thêm giá trị cell state chạy dọc xuyên suốt ở các bước lặp, chỉ bị tác động bởi một vài phép biến đổi tuyến tính khiến thông tin nó lưu trữ ít bị biến đổi, do đó ghi nhớ được thông tin của toàn bộ chuỗi thời gian. Mạng LSTM có khả năng thêm hoặc bớt thông tin từ cell state theo cơ chế cổng, gồm ba cổng sau: (i) tầng cổng đầu vào (input gate), (ii) tầng cổng quên (forget gate) và (iii) tầng cổng đầu ra (output gate).

Các cổng có thể điều chỉnh lượng thông tin đi qua bằng một tầng sigmoid và một phép nhân có hướng, đầu ra của tầng sigmoid nằm trong đoạn $[0, 1]$ thể hiện lượng thông tin giữ lại nhiều hay ít (1 tương ứng với giữ lại toàn bộ và 0 tương ứng với quên toàn bộ). Hình 8 mô tả quá trình thông tin đầu vào lần lượt được tính toán ở từng cổng. Đầu tiên tầng cổng quên (forget gate) sẽ quyết định với đầu vào hiện tại, trạng thái tổng nên bỏ đi những thông tin không cần thiết nào để tạo ra đầu vào mới. Tiếp theo, tầng cổng đầu vào (input gate) sẽ quyết định lấy thông tin nào từ đầu vào mới đó để lưu vào cell state. Cuối cùng dựa vào đầu vào của bước thời gian hiện tại cùng với cell state vừa cập nhật, tầng cổng đầu ra (output gate) thực hiện tính toán để quyết định phần thông tin nào của cell state sẽ làm đầu ra của bước thời gian hiện tại. Các công thức toán học cụ thể và chi tiết quá trình tính toán được mô tả trong [33].

Việc tách biệt hai loại thông tin theo toàn bộ thời gian và thông tin tại từng bước lặp thời gian giúp LSTM giải quyết được vấn đề về ghi nhớ thông tin dài. Một đoạn hội thoại là một dãy các câu nói có thứ tự giữa hai người. Nên để có thể hiểu đoạn hội thoại đang nói về vấn đề gì và trả lời được người còn lại, ta cần nắm bắt được nội dung của cả đoạn hội thoại phía trước. Do đó LSTM rất phù hợp và được ứng dụng rộng rãi nhằm giải quyết những bài toán trong các hệ thống hội thoại.

2.4 Phương pháp học trực tuyến (Online Learning)

Với phương pháp huấn luyện mô hình thông thường, mô hình sẽ nhận đầu vào toàn bộ tập dữ liệu huấn luyện được chuẩn bị từ trước, sử dụng bộ dữ liệu đó để học và tối ưu hóa tham số. Và chỉ khi quá trình huấn luyện kết thúc, mô hình mới sử dụng các tham số đã tối ưu để dự đoán kết quả cho các mẫu đầu vào mới.

Ta có thể lấy ví dụ về cách học trên như sau, giả sử ta có một số lượng các bánh quy nhiều loại bày trước mặt, ta cần phải học cách dự đoán xem các loại bánh quy tương ứng với những vị nào. Cách huấn luyện thông thường giống như việc ta lần lượt ăn tất cả những chiếc bánh quy đó, sau đó sử dụng những thông tin đã thu thập được sau khi ăn xong để dự đoán vị của một cái bánh quy mới.

Đi theo một hướng tiếp cận khác, đối với phương pháp học trực tuyến [24], không có sự tách biệt hẳn rõ ràng hai giai đoạn huấn luyện và dự đoán như trên. Thay vào đó, mỗi lượt trước khi ăn một cái bánh quy, ta sẽ tiến hành dự đoán thử vị của cái bánh quy đó trước. Tiếp theo, sau khi ăn thử một miếng, ta sẽ biết được vị thực sự của chiếc bánh quy, chiếc bánh lúc này sẽ lại được coi như một thông tin học giúp đỡ ta nâng cao kỹ thuật phân loại vị đối với những chiếc bánh quy ở lượt tiếp theo.

Nói một cách tổng quát, thì quá trình học trực tuyến diễn ra bao gồm một chuỗi các vòng lặp liên tiếp. Tại mỗi vòng lặp, mô hình được truyền vào một mẫu trong tập huấn luyện. Sau đó mô hình được yêu cầu dự đoán nhãn của mẫu đó. Tới cuối vòng lặp mô hình sẽ nhận được nhãn chính xác của mẫu đó. Cuối cùng, mô hình sử dụng thông tin vừa rồi để cải thiện khả năng dự đoán cho những lượt kế tiếp.

Chương 2 đã giới thiệu và giải thích sự thích hợp của mô hình mạng nơ-ron hồi quy đối với các hệ thống hội thoại. Cùng tới đó giới thiệu phương pháp học trực tuyến, trong đó mô hình sẽ lần lượt vừa thực hiện dự đoán vừa học từ dữ liệu mới. Từ kết quả nghiên cứu này, Chương 3 sẽ đưa ra định hướng tiếp cận và xây dựng mô hình phù hợp cho thành phần quản lý hội thoại trong hệ thống SmartDialog.

Chương 3 Đề xuất mô hình quản lý hội thoại và học tương tác

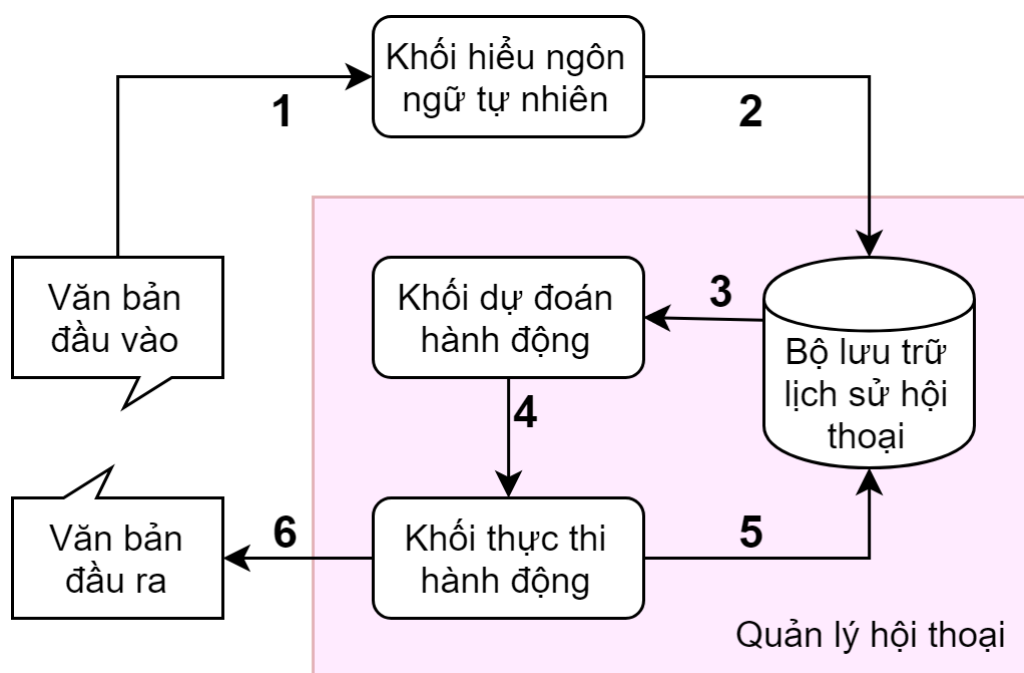
Chương 3 này sẽ trình bày tổng quan về giải pháp, sau đó giới thiệu phương pháp vec-tơ hóa trạng thái hội thoại ở phần 3.2.1, từ đó tạo dữ liệu đầu vào cho hai mô hình đề xuất nhằm giải quyết bài toán quản lý hội thoại bao gồm: (i) mô hình Long Short-Term Memory trong phần 3.2.2 và (ii) mô hình ghi nhớ (Memoization) trong phần 3.2.3. Phương pháp học tương tác sẽ được giới thiệu trong phần 3.3. Cuối cùng là thử nghiệm và đánh giá kết quả.

3.1 Giải pháp tổng quan

Hướng tiếp cận đề xuất nhằm ứng dụng học sâu trong thành phần quản lý hội thoại có tham khảo những nghiên cứu từ công trình [22], tuy nhiên có sửa đổi và áp dụng những giải pháp được trình bày trong [35].

Như đã đề cập tới ở những mục trước, giải pháp đề xuất sẽ không đi theo hướng tiếp cận end-to-end mà tách rời hoàn toàn hai thành phần hiểu ngôn ngữ tự nhiên (Natural Language Understanding - NLU) và thành phần quản lý hội thoại (Dialogue Management). Cách tiếp cận này giúp hai thành phần trên có thể được huấn luyện và sử dụng một cách độc lập với nhau, giảm số lượng kịch bản cần thiết khi huấn luyện và tạo ra khả năng tái sử dụng mô hình quản lý hội thoại cho nhiều ngôn ngữ. Bên cạnh đó mang tới cho những người dùng không chuyên một giải pháp quản lý và xây dựng một hệ thống hội thoại sử dụng học sâu. Các phương pháp huấn luyện mô hình theo hướng tiếp cận end-to-end giống như một hộp đen hoàn toàn (black box), ta không biết được quá trình xử lý bên trong từ lúc nhận thông tin đầu vào tới lúc nhận thông tin đầu ra diễn ra như thế nào. Việc tách rời quá trình huấn luyện NLU và thành

phần quản lý hội thoại mang tới khả năng “chia nhỏ” hộp đen đó ra, mở rộng khả năng tinh chỉnh mô hình. Giải pháp tổng quan được mô tả ở Hình 9.



Hình 9 Tổng quan giải pháp quản lý hội thoại.

Tại bước (1), tin nhắn đầu vào dạng văn bản của người dùng cuối được đưa vào NLU, thành phần này trích xuất ra các thông tin bao gồm ý định người dùng (intent) và các thông tin quan trọng trong câu hay còn được gọi là các thực thể (entity). Một ví dụ về kết quả đầu ra của NLU được thể hiện trong Bảng 1.

Bảng 1 Ví dụ kết quả trích xuất thông tin của thành phần NLU

Đầu vào	năm_nay	CNTT	lấy	bao_nhiêu	chỉ_tiêu
Thực thể	thời-gian	ngành-đào-tạo	khác	khác	khác
Ý định	hỏi-chỉ-tiêu-tuyển-sinh				

Ở Bảng 1, câu đầu vào của người dùng là “năm nay CNTT lấy bao nhiêu chỉ tiêu”. Khi đi vào khối NLU, câu văn được tách thành các từ, sau đó gán nhãn thực thể tương ứng cho từng từ. Từ “năm nay” được gán nhãn “thời-gian” để thể hiện ý nghĩa chỉ mốc thời gian, từ “CNTT” được gán nhãn *ngành-đào-tạo* để chỉ rằng cụm từ này là tên của một ngành đào tạo thuộc trường đại học. Các từ khác không có ý nghĩa gì đặc

biệt nên được gán nhãn *khác*. Tiếp theo đó, ý định người dùng được NLU dự đoán là *hỏi-chỉ-tiêu-tuyển-sinh*, đây là tên của một trong những ý định thuộc một tập ý định được định nghĩa từ trước, thể hiện các mong muốn trong câu nói của người dùng.

Các thông tin về thực thể và ý định người dùng trích xuất được ở bước phía trước sẽ được ghi lại vào bộ lưu trữ lịch sử hội thoại ở bước (2), bộ này tiếp nhận thông tin thay đổi từ cuộc hội thoại và lưu trữ lại vào trong bộ nhớ. Tiếp theo ở (3), khối dự đoán hành động nhận thông tin các trạng thái hội thoại từ bộ lưu trữ lịch sử, tạo ra các biểu diễn véc-tơ đặc trưng trạng thái hội thoại và đưa ra dự đoán hành động tiếp theo của hệ thống sao cho phù hợp nhất và thực thi hành động đó tại bước (4). Hành động đã dự đoán ra sẽ được ghi chép lại vào bộ lưu trữ lịch sử hội thoại tại bước (5) và cuối cùng, trả về phản hồi lại cho người dùng ở bước (6).

Hành động của hệ thống (action)

Ta sẽ đi vào chi tiết hơn một chút về hành động hệ thống ở trong Hình 9 để hiểu rõ hơn bản chất của giải pháp đề xuất. Bài toán quản lý hội thoại tiếp cận với giải pháp này có thể được quy về bài toán phân loại (classification problem). Khi thực hiện dự đoán, mô hình quản lý hội thoại dựa vào các đặc trưng trích xuất từ trạng thái hội thoại, để dự đoán ra một hành động hệ thống nằm trong danh sách hành động hệ thống được định nghĩa từ trước, hành động nào có độ chính xác (confidence) cao nhất sẽ được lựa chọn. Những hành động này có sẵn hoặc do người dùng tự định nghĩa, nó có thể là những tin nhắn đơn giản để phản hồi lại cho người dùng cuối, hoặc là việc thực thi các hàm để lấy dữ liệu từ bên thứ ba. Khi một hành động được thực hiện, thông tin của nó được chuyển lại về và cập nhật vào bộ lưu trữ lịch sử hội thoại.

Ở Bảng 2, hai hành động là *chào-hỏi* và *liệt-kê-chương-trình-đt* là hai hành động trả về phản hồi dạng văn bản. Giả sử khi hành động *chào-hỏi* được thực hiện, hệ thống sẽ chọn ngẫu nhiên một trong ba câu phản hồi được định nghĩa trước là “*Chào bạn, mình có thể giúp gì được cho bạn*”, “*Xin chào, mình có thể giúp gì được cho bạn*” hoặc “*Hello, bạn cần mình giúp gì nhĩ*” để phản hồi lại cho người dùng cuối. Mặt khác, khi hành động ở dưới cùng *chỉ-tiêu-ngành-đào-tạo* được thực thi, hệ thống sẽ

gọi một hàm có tên là *timKiemChiTieuNganhDaoTao()* có chức năng gọi API tới hệ thống bên ngoài để truy vấn thông tin về chỉ tiêu tuyển sinh.

Bảng 2 Một số ví dụ về hành động của hệ thống

Hành động hệ thống	Nội dung
chào-hỏi	Chào bạn, mình có thể giúp gì được cho bạn
	Xin chào, mình có thể giúp gì được cho bạn
	Hello, bạn cần mình giúp gì nhỉ
liệt-kê-chương-trình-đt	ĐHBK HN có các chương trình đào tạo sau: Chương trình đào tạo đại trà, chương trình ELITECH, chương trình đào tạo quốc tế và các chương trình đào tạo sau đại học. Bạn muốn tìm hiểu về chương trình đào tạo nào ?
	Có chương trình chính quy, chương trình ELITECH, đào tạo quốc tế và các chương trình đào tạo sau đại học bạn nhé. Bạn muốn tìm thông tin về chương trình nào ạ ?
chỉ-tiêu-ngành-đào-tạo	<i>hàm timKiemChiTieuNganhDaoTao()</i>

Có thể hiểu rằng một cuộc hội thoại giữa người dùng cuối và hệ thống hội thoại được tổng quát hóa thành một dãy các ý định người dùng và hành động hệ thống sắp xếp liên kế nhau.

3.2 Các mô hình dự đoán hành động

Như đã đề cập, khác với những cách tiếp cận end-to-end, ta cần phải tự định nghĩa ra những đặc trưng để biểu diễn cho một trạng thái hội thoại tại một thời điểm của phiên hội thoại. Sau đó sử dụng những biểu diễn của trạng thái hội thoại phía trước để làm đầu vào giúp các mô hình dự đoán hành động đưa ra dự đoán phù hợp.

Trong mỗi một quá trình dự đoán hành động, có thể gồm một hoặc nhiều mô hình dự đoán hành động tham gia. Các mô hình lần lượt đưa ra dự đoán về hành động hệ thống phù hợp nhất, mô hình nào đưa ra kết quả hành động có độ chính xác

(confidence) cao nhất sẽ được lựa chọn làm kết quả cuối cùng. Đồ án đề xuất hai mô hình dự đoán hành động là (i) mô hình LSTM và (ii) mô hình ghi nhớ (Memorization).

3.2.1 Trích xuất biểu diễn đặc trưng trạng thái hội thoại

Ta biểu diễn một trạng thái hội thoại dưới dạng một véc-tơ nhị phân. Các đặc trưng được lựa chọn bao gồm một số thông tin sau: (i) thông tin về hành động được hệ thống thực thi gần đây nhất, (ii) thông tin về các ý định và thực thể trích xuất được từ tin nhắn gần đây nhất của người dùng cuối, (iii) thông tin về các thực thể đã được cung cấp từ khi bắt đầu phiên hội thoại cho tới thời điểm hiện tại và (iv) thông tin về kết quả dữ liệu lấy được từ những lần gọi API.

Để hiểu rõ hơn về các thông tin này, ta cùng xem xét một ví dụ ở Hình 10.

Người dùng: Xin chào, mình tên là Kỳ	Intent: giới-thiệu Entity: tên(Kỳ)
Hệ thống: Chào Kỳ, mình có thể giúp gì cho bạn ?	Action: chào-hỏi
Người dùng: Năm nay CNTT lấy bao nhiêu chỉ tiêu ?	Intent: hỏi-chỉ-tiêu-tuyển-sinh Entity: thời-gian(năm nay) ngành-đào-tạo(CNTT)
→ Hệ thống: Khoảng 1000 bạn nhé !	Action: chỉ-tiêu-ngành-đào-tạo

Hình 10 Ví dụ về thông tin lưu trữ của một đoạn hội thoại trong hệ thống.

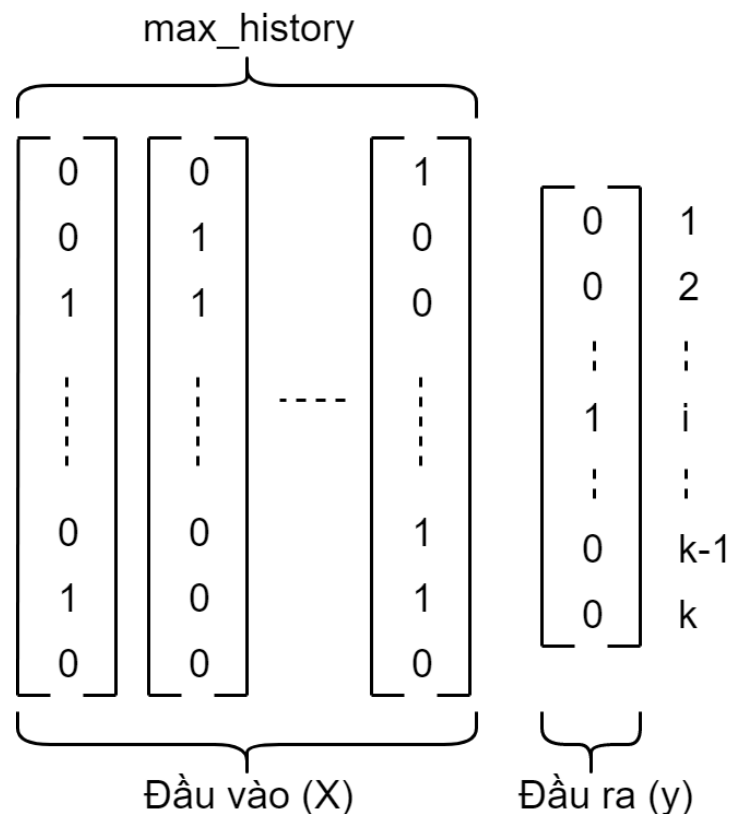
Ở Hình 10, phần bên trái là đoạn hội thoại trong thực tế, còn phần bên phải là thông tin được xử lý và lưu trữ trong hệ thống. Mỗi lượt người dùng gửi tin nhắn mới hay hệ thống phản hồi lại đều tạo ra một trạng thái hội thoại mới, giả sử ta xét tới câu phản hồi “*Khoảng 1000 bạn nhé*” (dòng có mũi trên màu xanh dương). Sau khi hệ thống phản hồi lại cho người dùng cuối, một trạng thái hội thoại mới sẽ được tạo ra gồm những đặc trưng như sau:

- Hành động thực thi gần đây nhất: *chỉ-tiêu-ngành-đào-tạo*
- Thông tin trong tin nhắn gần nhất của người dùng cuối: câu “*Năm nay CNTT lấy bao nhiêu chỉ tiêu*” có ý định là *hỏi-chỉ-tiêu-tuyển-sinh*, đồng thời cung

cấp hai thực thể là *thời-gian* với giá trị là “*năm nay*” và *ngành-đào-tạo* với giá trị là “*CNTT*”.

- Các thực thể đã cung cấp từ lúc bắt đầu cuộc hội thoại gồm ba loại thực thể, bao gồm *tên*, *thời-gian* và *ngành-đào-tạo*.
- Thông tin thu được từ kết quả gọi API về chỉ tiêu tư vấn tuyển sinh của ngành CNTT là 1000 chỉ tiêu.

Các thông tin trên sẽ được lần lượt chuyển thành từng dạng véc-tơ nhị phân. Ví dụ trong bộ dữ liệu có tổng cộng n ý định được định nghĩa trước đánh số từ 1 tới n , thì véc-tơ thể hiện đặc trưng ý định thứ i được chọn sẽ là một véc-tơ có kích thước $n \times 1$, trong đó phần tử thứ i có giá trị là 1, còn các phần tử còn lại đại diện cho các ý định khác được gán là giá trị là 0. Sau cùng, các véc-tơ này được kết nối lại thành một véc-tơ nhị phân duy nhất đại diện cho một trạng thái hội thoại.



Hình 11 Dạng dữ liệu đầu vào và đầu ra của mô hình dự đoán hành động.

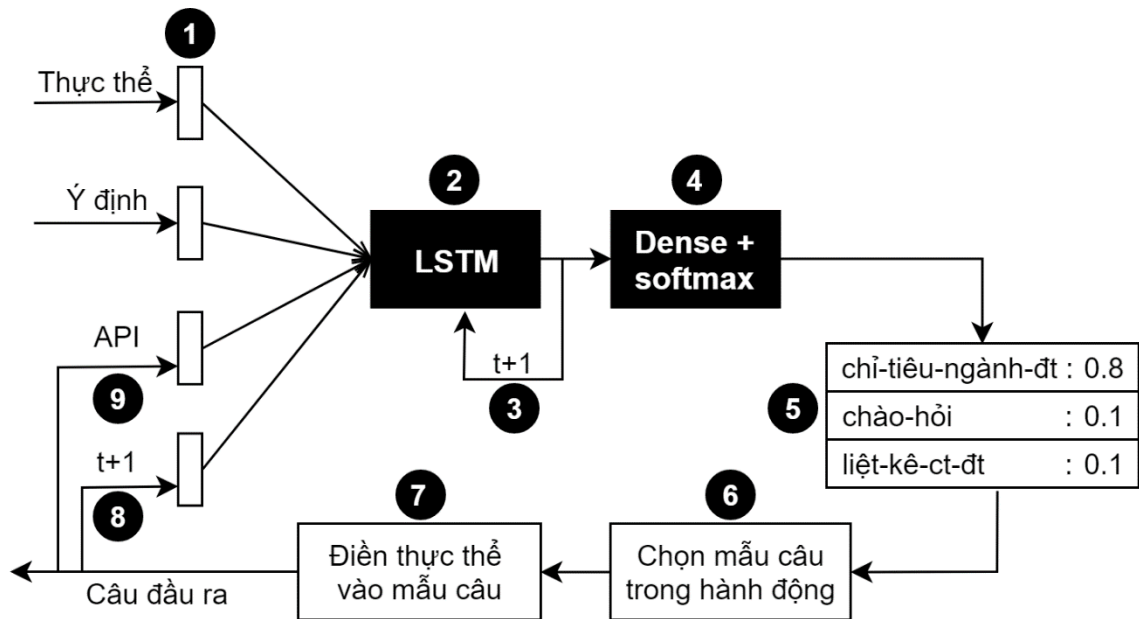
Trong quá trình dự đoán và hoặc huấn luyện, mô hình sẽ trở nên chính xác hơn nếu quan tâm tới thông tin của cả những trạng thái hội thoại trước đó chứ không chỉ để ý

tới trạng thái hội thoại hiện tại. Giả sử số lượng hành động là k , dạng dữ liệu đầu vào và đầu ra của các mô hình dự đoán hành động được minh họa ở trong Hình 11.

Dữ liệu đầu vào của mô hình được gọi là tập đặc trưng X , bao gồm véc-tơ trạng thái hội thoại hiện tại và một số lượng những véc-tơ trạng thái hội thoại phía trước đó, số lượng này được gọi là *max_history*, có giá trị từ 3 tới 6 tùy theo dữ liệu huấn luyện. Dữ liệu đầu ra là nhãn y , là một véc-tơ nhị phân có độ dài bằng số lượng hành động hệ thống, với phần tử thứ i có giá trị bằng 1 thể hiện hành động thứ i là hành động sẽ được thực thi.

3.2.2 Mô hình Long Short-Term Memory

Như đã giới thiệu ở mục 2.3.2, mạng LSTM có khả năng lưu giữ thông tin theo thời gian dài tương đối tốt, do đó rất phù hợp với dạng dữ liệu là một dãy các trạng thái hội thoại như đã trình bày ở mục 3.2.1. Hình 12 miêu tả chi tiết quá trình hoạt động của mạng LSTM đã đề xuất.



Hình 12 Mô hình LSTM trong dự đoán hành động.

Trong Hình 12, những con số là thứ tự các bước, những khối có màu đen là những thành phần có thể huấn luyện được. Giả sử ta đang ở bước lặp thời gian t , quá trình bắt đầu từ bước (1) khi véc-tơ biểu diễn trạng thái hội thoại ở thời điểm t được tạo

ra. Tiếp theo, véc-tơ này được truyền vào để tính toán trong mạng LSTM ở bước (2). Ở (3), LSTM thực hiện tính toán ra một số véc-tơ trạng thái ẩn, véc-tơ này được giữ lại cho việc tính toán ở bước lặp thời gian $t - 1$ tiếp theo, đồng thời truyền qua mạng dense (mạng nơ-ron đầy đủ) với hàm kích hoạt softmax ở bước (4). Véc-tơ đầu ra có số chiều bằng với số lượng hành động trong hệ thống. Tới bước (5), hành động có độ chính xác cao nhất sẽ được lựa chọn. Ở bước (6), nếu hành động động này là dạng văn bản trả về thì một mẫu câu sẽ được chọn ngẫu nhiên trong số các mẫu câu của hành động đó, còn nếu là hành động gọi API thì hàm của hành động sẽ được thực thi. Tại bước (7), các thông tin từ entity thu được từ đầu cuộc hội thoại được điền vào mẫu câu chọn ra ở bước trước để tạo ra một câu phản hồi hoàn chỉnh, ví dụ mẫu câu “điểm chuẩn ngành {ngành-đào-tạo} là 25.5” sẽ được hoàn chỉnh thành “điểm chuẩn ngành CNTT là 25.5”. Ở bước (8), hành động vừa được chọn sẽ trở thành đặc trưng thêm cho véc-tơ trạng thái hội thoại ở bước lặp $t - 1$ tiếp theo. Ngoài ra nếu hành động có gọi API, thì trong bước (9), các dữ liệu lấy từ API cũng sẽ trở thành đặc trưng thêm của véc-tơ nói trên.

3.2.3 Mô hình ghi nhớ (Memoization)

Mô hình này được đề xuất nhằm làm giảm các lỗi không đáng có trong quá trình dự đoán. Khi người dùng định nghĩa các kịch bản để huấn luyện cho mô hình, thì họ sẽ mong muốn các trường hợp có trong tập huấn luyện sẽ được dự đoán chính xác y hệt như kịch bản đã được tạo ra. Mô hình dự đoán Memoization sẽ giải quyết vấn đề này, nó lưu dữ liệu dưới dạng *khóa-giá trị* (*key-value*), với *khóa* là thông tin *max_history* trạng thái hội thoại và *giá trị* là hành động của hệ thống được chọn tương ứng với các trạng thái hội thoại đó.

Trong một phiên hội thoại, bất cứ khi nào các thông tin của *max_history* trạng thái hội thoại đang tồn tại y hệt trong kịch bản huấn luyện, thì hành động tương ứng sẽ được đưa ngay ra với độ chính xác (confidence) là 100%. Điều này khiến các mô hình dự đoán khác không thể đưa ra đáp án hành động khác nào có độ chính xác cao hơn, đảm bảo độ chính xác precision của hệ thống luôn bằng 100%.

3.3 Học tương tác

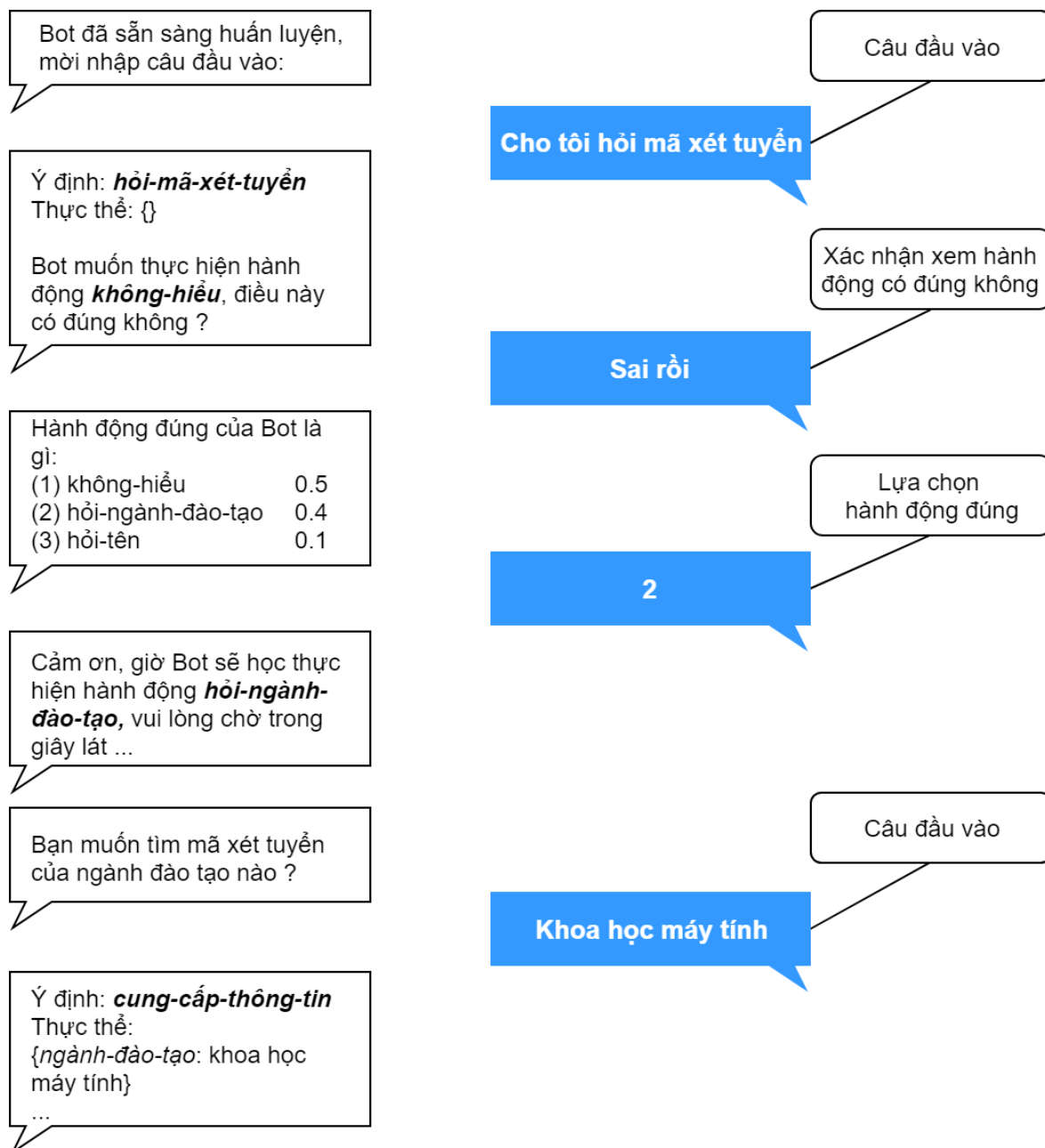
Khi tạo ra một hệ thống hội thoại, người sử dụng nền tảng hội thoại luôn muốn tạo dữ liệu và huấn luyện cho hệ thống một cách đơn giản và trực quan nhất. Do đó, một chức năng mà các nền tảng hội thoại luôn hướng tới là làm sao để hỗ trợ người sử dụng thực hiện việc huấn luyện các kịch bản hội thoại mới thông qua tương tác trực tiếp với hệ thống.

Từ đó, đề án đề xuất một phương án giúp giải quyết vấn đề này, đó là phương pháp học tương tác (Interactive Learning). Giải pháp này dựa trên cơ sở của cách thức huấn luyện có tên là học trực tuyến (Online Learning) đã giới thiệu ở mục 2.4.

Trong quá trình học tương tác, người dùng (người tạo hệ thống tương tác) sẽ lần lượt chuyển đổi qua lại giữa hai vai trò là người dùng cuối và người dạy học. Người dùng cuối thực hiện nhắn tin với Chatbot, người dạy học quan sát kết quả dự đoán hành động của Chatbot và sửa sai nếu cần thiết. Với giải pháp này ta sẽ chỉ áp dụng để huấn luyện mô hình LSTM và tạm thời không sử dụng tới mô hình ghi nhớ, việc này nhằm đưa ra cái nhìn khách quan hơn về độ chính xác của mô hình LSTM. Hình 13 minh họa một đoạn hội thoại tương tác nhằm huấn luyện Chatbot Tư vấn tuyển sinh.

Đầu tiên, hệ thống khởi tạo một mô hình dự đoán hành động dựa trên bộ tham số ngẫu nhiên trong trường hợp mô hình chưa được huấn luyện lần nào, hoặc dựa trên bộ tham số của lần huấn luyện trước, chờ đợi nhận câu đầu vào. Người dùng - đóng vai trò của một người dùng cuối - nhập câu “Cho tôi hỏi mã xét tuyển”. Hệ thống sau đó dự đoán ra ý định người dùng là *hỏi-mã-xét-tuyển*, và không trích xuất được thực thể nào. Từ những thông tin đó, bộ dự đoán thực hiện dự đoán ra hành động tương ứng là *không-hiểu*, rồi yêu cầu người dùng xác nhận xem hành động đó có đúng hay không. Trong trường hợp người dùng xác nhận hành động đúng, hệ thống sẽ thực hiện hành động đó và tiếp tục luồng hội thoại. Tuy nhiên, hành động *không-hiểu* nếu thực hiện ở đây sẽ không hợp lý. Do đó, người dùng - đóng vai trò của người dạy học - sẽ báo với hệ thống rằng hành động dự đoán ra vừa rồi là “Sai rồi”. Tiếp theo, hệ thống cung cấp thông tin về danh sách các hành động và độ chính xác tương ứng dự đoán ra bởi mô hình dự đoán hành động. Hành động thứ (1) là *không-hiểu* với độ

chính xác cao nhất (50%) chính là hành động đã được hệ thống yêu cầu xác nhận ở bước trước, tuy nhiên hành động (2) *hỏi-ngành-đào-tạo* mới là hành động đúng ở trong trường hợp này, do để tìm thông tin mã xét tuyển thì hệ thống cần có thông tin về tên ngành đào tạo. Người dùng nhập vào số thứ tự (2) của hành động mong muốn.



Hình 13 Minh họa giải pháp học tương tác.

Tới đây, hệ thống đã nhận thêm được một mẫu dữ liệu huấn luyện gồm đầu vào là các trạng thái hội thoại phía trước và đầu ra là hành động vừa được người dùng nhập.

Ngay lập tức, mô hình dự đoán hành động LSTM sẽ được học thêm mẫu dữ liệu này và tham số của mô hình được cập nhật lại. Sau khi quá trình huấn luyện diễn ra, hệ thống thực hiện hành động vừa học được và gửi về phản hồi cho người dùng. Và vòng lặp học tương tác sẽ tiếp tục được diễn ra cho tới khi nào người dùng quyết định kết thúc quá trình huấn luyện. Khi quá trình học tương tác kết thúc, lịch sử cuộc hội thoại sẽ được lưu lại dưới dạng các kịch bản huấn luyện, các kịch bản này có thể được sử dụng cho các lần huấn luyện mô hình ở lần sau theo hình thức truyền thống.

3.4 Tạo kịch bản huấn luyện tự động

Các mô hình học máy, đặc biệt là những mô hình học sâu luôn cần một lượng dữ liệu nhất định để đưa ra được kết quả tốt, do đó việc hỗ trợ người dùng tạo ra kịch bản huấn luyện thường gặp nhất là rất cần thiết. Một trong những luồng hội thoại hay gặp nhất trong các cuộc trò chuyện là việc người dùng cuối cùng cấp các thông tin cá nhân hoặc thông tin muốn quan tâm cho Chatbot. Mục tiêu đặt ra là đưa ra một cách thức tạo kịch bản tự động để giúp Chatbot học được các nghiệp vụ trên và biết cách thu thập thông tin từ người dùng.

Trong số các ý định người dùng, có một số ý định mang ý nghĩa đưa ra ý kiến hoặc ý nghĩa chung chung như *chào-hỏi*, *tạm-biệt*, *cảm-on*, *đồng-ý*, *từ-chối*,... Mặt khác, có một số các ý định có các mục đích nhất định, và để thực hiện mục đích đó người dùng cần cung cấp một số thông tin cần thiết. Ví dụ với ý định *hỏi-điểm-chuẩn-ngành*, mục đích của người dùng là truy vấn thông tin về điểm chuẩn của một ngành học nào đó. Để lấy thông tin, người dùng cần cung cấp tên ngành đào tạo cần hỏi dưới dạng các thực thể ở trong câu.

Ví dụ trong câu “điểm chuẩn ngành khoa học máy tính là bao nhiêu”, thì “khoa học máy tính” chính là thông tin entity *ngành-đào-tạo* mà Chatbot cần để truy vấn thông tin điểm chuẩn. Các entity dạng này trong câu sẽ được đánh dấu là các *required-param*, mang ý nghĩa là entity cần thiết để thực hiện mục đích của ý định.

Để chuẩn bị thực hiện tạo kịch bản tự động thu thập thông tin, người dùng cần đánh dấu các entity nào trong ý định cần phải là *required-param*, đồng thời chỉ ra hành

động nào sẽ được thực hiện nếu đã cung cấp đầy đủ entity. Mỗi khi người dùng nhấn chọn đánh dấu một entity nào đó là *required-param*, một hành động sẽ được hệ thống tạo ra tự động với mục đích đưa ra câu hỏi lại về thông tin entity đó, ví dụ: *action-hỏi-ngành-đào-tạo*. Sau đó người dùng tạo ra thêm một ý định mang ý nghĩa cung cấp thông tin tất cả các entity cần thiết, em tạm đặt tên ý định này là *cung-cấp-thông-tin*, trong ý định này sẽ có các câu cung cấp các loại *required-param* entity, ví dụ: “khoa học máy tính nhé”, “ngành khoa học máy tính” , “chương trình elitech”, “chương trình đại trà”,... Toàn bộ các bước trên sẽ được thực hiện trên giao diện của SmartDialog một cách dễ dàng.

Khi quá trình tạo kịch bản tự động bắt đầu, hệ thống sẽ duyệt vào tạo kịch bản cho lần lượt từng ý định, các ý định không chứa *required-param* nào sẽ được bỏ qua. Ngược lại, với các ý định chứa *required-param*, hệ thống sẽ tạo ra một số luồng kịch bản hay gặp như (i) người dùng cung cấp đầy đủ thông tin ngay từ câu đầu tiên, (ii) người dùng cung cấp thiếu 1 thông tin cần thiết, (iii) người dùng cung cấp thiếu từ 2 thông tin cần thiết trở lên và (iv) người dùng đang cung cấp thiếu thông tin nhưng lại nói sang một vấn đề khác.

Bảng 3 dưới đây là minh họa một số các kịch bản được hệ thống tạo ra tự động cho ý định *hỏi-điểm-chuẩn-ngành* được đề cập tới ở phần trên. Cột bên trái bảng là kịch bản được tạo tự động và cột bên phải là đoạn hội thoại được mô tả tương ứng trong thực tế.

Bảng 3 Một số dạng kịch bản được tạo tự động

Kịch bản huấn luyện	Hội thoại thực tế tương ứng
(i) Người dùng cung cấp đầy đủ thông tin ngay từ câu đầu tiên	
* hỏi-điểm-chuẩn-ngành {“ngành-đào-tạo”: “CNTT”, “năm”: “năm nay”} - tìm-điểm-chuẩn-ngành	* Năm nay điểm chuẩn CNTT là bao nhiêu - Là 20.5 bạn nhé
(ii) Người dùng cung cấp thiếu 1 thông tin cần thiết	

Kịch bản huấn luyện	Hội thoại thực tế tương ứng
<ul style="list-style-type: none"> * hỏi-điểm-chuẩn-ngành {“ngành-đào-tạo”: “CNTT”} - action-hỏi-năm * cung-cấp-thông-tin{“năm”: “năm nay”} - tìm-điểm-chuẩn-ngành 	<ul style="list-style-type: none"> * Điểm chuẩn CNTT là bao nhiêu - Bạn muốn hỏi năm nào * Năm nay ấy - Là 20.5 bạn nhé
<ul style="list-style-type: none"> * hỏi-điểm-chuẩn-ngành{“năm”: “năm nay”} - action-hỏi-ngành-đào-tạo * cung-cấp-thông-tin{“ngành-đào-tạo”: “CNTT”} - tìm-điểm-chuẩn-ngành 	<ul style="list-style-type: none"> * Điểm chuẩn năm nay là bao nhiêu - Bạn muốn hỏi về ngành nào * Ngành CNTT - Là 20.5 bạn nhé
(iii) Người dùng cung cấp thiếu từ 2 thông tin cần thiết trở lên	
<ul style="list-style-type: none"> * hỏi-điểm-chuẩn-ngành - action-hỏi-năm * cung-cấp-thông-tin{“năm”: “năm nay”} - action-hỏi-ngành-đào-tạo * cung-cấp-thông-tin{“ngành-đào-tạo”: “CNTT”} - tìm-điểm-chuẩn-ngành 	<ul style="list-style-type: none"> * Điểm chuẩn là bao nhiêu - Bạn muốn hỏi năm nào * Năm nay ấy - Bạn muốn hỏi ngành nào * Ngành CNTT - Là 20.5 bạn nhé
<ul style="list-style-type: none"> * hỏi-điểm-chuẩn-ngành - action-hỏi-năm * cung-cấp-thông-tin {“ngành-đào-tạo”: “CNTT”, “năm”: “năm nay”} - tìm-điểm-chuẩn-ngành 	<ul style="list-style-type: none"> * Điểm chuẩn là bao nhiêu - Bạn muốn hỏi năm nào * Ngành CNTT năm nay ấy - Là 20.5 bạn nhé
(iv) Người dùng đang cung cấp thiếu thông tin nhưng lại hỏi sang một vấn đề khác	
<ul style="list-style-type: none"> * hỏi-điểm-chuẩn-ngành - action-hỏi-năm * mã-xét-tuyển{“ngành-đào-tạo”: “CNTT”} - tìm-mã-xét-tuyển 	<ul style="list-style-type: none"> * Điểm chuẩn là bao nhiêu - Bạn muốn hỏi năm nào * À mã xét tuyển CNTT là gì - IT3 nhé

Trong Bảng 3, phần (i) miêu tả kịch bản trong đó người dùng cung cấp đủ thông tin cần ở ngày tin nhắn đầu tiên và hệ thống phản hồi thông tin về điểm chuẩn ngay lập tức. Trong phần (ii) và (iii), người dùng cung cấp thiếu thông tin về *năm* hoặc/và *ngành-đào-tạo*, hệ thống đưa ra những hành động hỏi thông tin thiếu tương ứng là *action-hỏi-năm* và *action-hỏi-ngành-đào-tạo*. Còn ở phần (iv), người dùng đang hỏi

về vấn đề điểm chuẩn thì lại hỏi sang vấn đề mã xét tuyển, nhưng hệ thống vẫn sẽ phản hồi một cách chính xác.

3.5 Thử nghiệm và đánh giá

3.5.1 Dữ liệu

Ta tiến hành thử nghiệm khả năng học và dự đoán của mô hình trên bộ dữ liệu tư vấn tuyển sinh (TVTS) do văn phòng tư vấn tuyển sinh Trường Đại học Bách Khoa Hà Nội cung cấp. Bộ dữ liệu được tạo ra trên nền tảng SmartDialog, dựa vào những lịch sử hội thoại giữa sinh viên, tư vấn viên và dữ liệu về tư vấn tuyển sinh trên trang web của ĐH BKHN. Chi tiết thông tin về bộ dữ liệu được đưa ra ở Bảng 4.

Bảng 4 Bộ dữ liệu tư vấn tuyển sinh

Số lượng ý định	22
Số lượng thực thể	4
Số lượng mẫu câu	4652
Số lượng hành động	23
Số lượng kịch bản	145

Bộ dữ liệu TVTS gồm 22 ý định người dùng với tổng số 4652 mẫu câu được tạo ra ở trong các ý định đó. Dữ liệu dùng để huấn luyện cho mô hình quản lý hội thoại gồm 145 kịch bản, mỗi một kịch bản là một đoạn hội thoại bao gồm các ý định người dùng, thực thể và hành động hệ thống được sắp xếp theo thứ tự. Một ví dụ về kịch bản trong tập huấn luyện như sau:

```
* chào-hỏi
  - chào-hỏi
* hỏi-điểm-chuẩn-ngành{"ngành-đào-tạo": "CNTT"}
  - tìm-điểm-chuẩn-ngành
* cảm-on
  - tạm-biệt
```


Ví dụ trên mô tả một kịch bản trong tệp dữ liệu kịch bản TVTS. Những dòng bắt đầu bằng kí tự “*” thể hiện ý định và các thực thể của câu nói người dùng, những dòng bắt đầu bằng kí tự “-” thể hiện hành động phản hồi từ hệ thống.

Ta tiến hành chuyển đổi dạng kịch bản trong tệp thành dạng véc-tơ đầu vào và đầu ra phù hợp (đã trình bày trong mục 3.2.1). Tiếp theo, ta chia bộ dữ liệu thành hai tập dữ liệu huấn luyện (train) và kiểm thử (test) theo tỉ lệ 80 – 20.

3.5.2 Kết quả thử nghiệm

Bảng 5 Độ chính xác của mô hình trên bộ dữ liệu TVTS

Độ chính xác trong quá trình huấn luyện				
Lần lặp	Acc (%)	Precision (%)	Recall (%)	F1 (%)
1	96,9	96,1	97,0	96,5
2	96,2	96,0	96,1	96,0
3	98,0	98,1	98,0	98,0
4	94,2	93,3	94,1	93,7
5	96,7	98,0	97,2	97,6
6	98,0	98,3	98,2	98,2
7	93,5	91,1	93,1	92,1
8	93,3	92,2	93,0	92,6
9	94,8	94,1	95,1	94,6
10	96,0	94,0	96,0	95,0
Trung bình	94,2	95,1	95,8	95,4
Độ chính xác trên bộ Test				
Trung bình	93,7	94,5	95,2	94,8

Ta tiến hành huấn luyện mô hình với kỹ thuật **k-fold cross validation** [36] nhằm tránh các vấn đề về overfitting. Thực hiện chia bộ dữ liệu huấn luyện ra thành 10 tập con, lần lượt thực hiện 10 lần, mỗi lần huấn luyện trên 9 tập và kiểm thử trên 1 tập còn lại. Sau khi huấn luyện xong, ta tiến hành đo độ chính xác trên tập kiểm thử. Các kết quả thu được với mô hình dự đoán hành động LSTM được trình bày ở Bảng 5. Ta thấy mô hình không có hiện tượng bị overfitting trong quá trình huấn luyện, thực hiện kiểm tra ở các lần lặp khác nhau đều cho độ chính xác accuracy lớn hơn 93% và f1-score trên 92%. Độ chính xác trên tập dữ liệu kiểm thử chỉ thấp hơn so với huấn luyện một chút, accuracy đạt 93,7% so với 94,2% và f1-score đạt 94,8% so với 95,4%.

Chương này đã trình bày chi tiết giải pháp đề xuất cho bài toán quản lý hội thoại. Tạo đầu vào dưới dạng các véc-tơ nhị phân biểu diễn trạng thái hội thoại cho mô hình dự đoán hành động LSTM và Memoization. Giải pháp học tương tác giúp người dùng không chuyên huấn luyện Chatbot một cách tự nhiên và dễ dàng hơn, đồng thời cung cấp một giao thức tạo dữ liệu đầu vào tiện lợi. Cuối cùng thử nghiệm trên bộ dữ liệu TVTS đưa ra độ chính xác cao lên tới 94,8%.

Khả năng dự đoán hành động của mô hình tương đối tốt, tuy nhiên ta cần thực hiện giao tiếp trực tiếp với hệ thống thì mới có thể đưa ra đánh giá trực quan nhất. Phần đánh giá Chatbot TVTS sẽ được trình bày ở trong Chương 4 tiếp theo, liên quan tới phần nội dung kết nối, triển khai Chatbot TVTS trên Zalo qua nền tảng hội thoại thông minh SmartDialog.

Chương 4 Phát triển phân hệ quản lý hội thoại trong SmartDialog

Chương này sẽ giới thiệu tổng quan về kiến trúc nền tảng hội thoại thông minh SmartDialog, trình bày những nội dung liên quan tới xây dựng, triển khai phân hệ quản lý hội thoại bao gồm (i) dịch vụ dự đoán hành động, (ii) dịch vụ huấn luyện và (iii) giải pháp quản lý phiên hội thoại thông qua máy chủ Gateway. Sau đó trình bày phân tích hợp Chatbot Tư vấn tuyển sinh trên nền tảng Zalo.

4.1 Kiến trúc tổng quan của SmartDialog

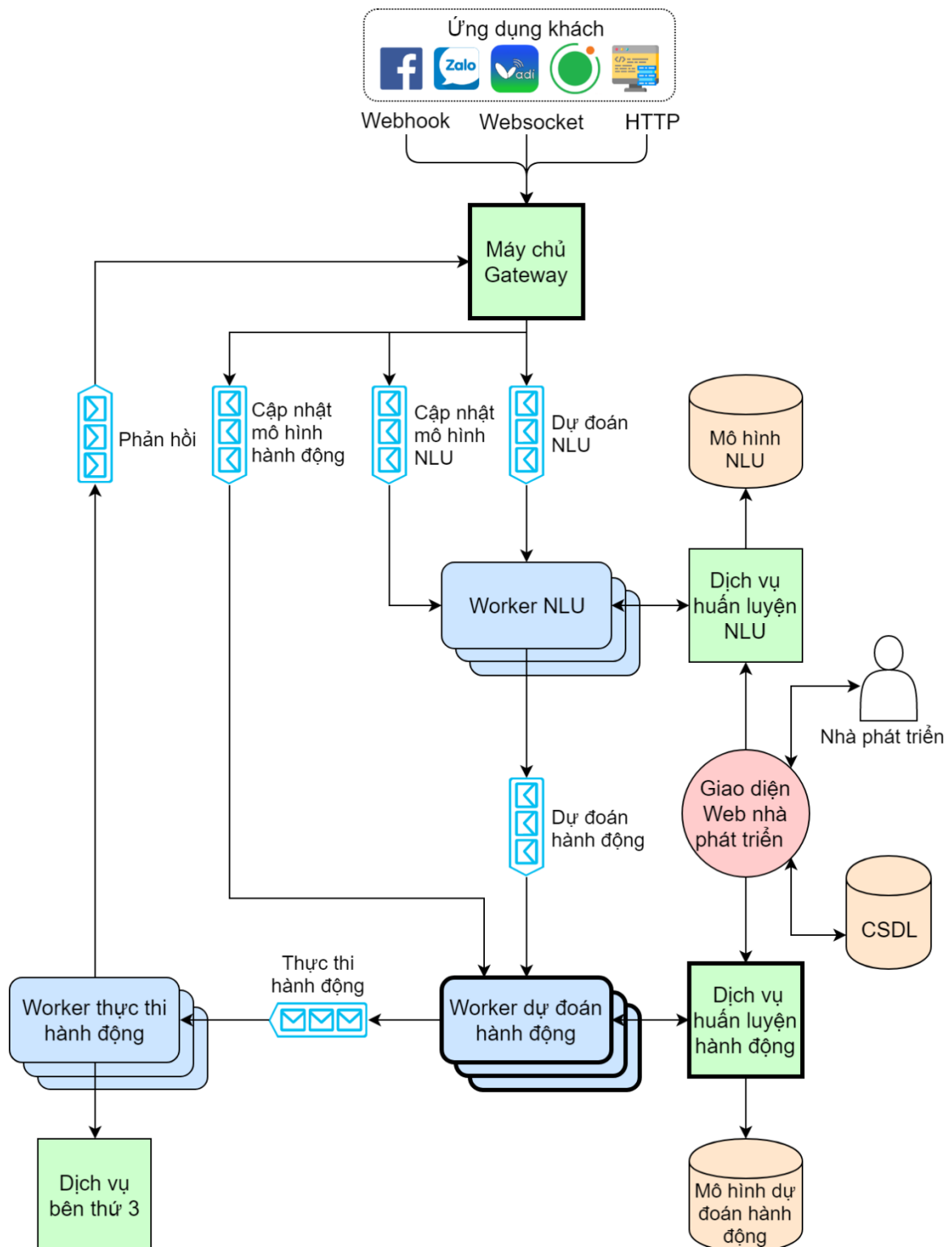
SmartDialog được thiết kế theo mô hình kiến trúc Microservices, trong đó bao gồm một số dịch vụ chính như dịch vụ huấn luyện NLU, dịch vụ huấn luyện hành động, dịch vụ dự đoán NLU, dịch vụ dự đoán hành động, dịch vụ thực thi hành động.

Mặt khác, nền tảng có thể được chia dựa theo đối tượng sử dụng thành hai phần chính là (i) phần dành cho nhà phát triển để thực hiện xây dựng và huấn luyện agent, và (ii) phần triển khai các mô hình đã huấn luyện để tương tác với người dùng cuối.

Hình 14 mô tả kiến trúc tổng quan của toàn bộ nền tảng tương tác hội thoại thông minh SmartDialog. Những thành phần có viền đậm màu đen là một trong những thành phần thuộc phân hệ quản lý hội thoại được trình bày ở các mục tiếp theo trong chương này.

Phần dành cho nhà phát triển bao gồm giao diện Web, các dịch vụ huấn luyện NLU và dịch vụ huấn luyện hành động. Những dịch vụ này giúp nhà phát triển có thể dễ dàng tạo ra và xây dựng một bộ dữ liệu cho agent của mình, sau đó tiến hành huấn luyện trên bộ dữ liệu đó. Dữ liệu của agent được lưu lại trong CSDL MongoDB [37]. Phần giao diện Web giao tiếp với các dịch vụ huấn luyện thông qua các REST API.

Các dịch vụ huấn luyện này sau khi huấn luyện xong agent sẽ lưu lại mô hình thu được trên phần cứng của máy chủ.



Hình 14 Kiến trúc tổng quan nền tảng SmartDialog

Phần triển khai các mô hình đã huấn luyện để tương tác với người dùng cuối được thực hiện theo mô hình kiến trúc phân tán sử dụng phần mềm RabbitMQ [38]. Bao gồm một máy chủ Gateway tiếp nhận thông tin yêu cầu đầu vào, các dịch vụ dự đoán NLU, dịch vụ dự đoán hành động và dịch vụ thực thi hành động. Các dịch vụ này được đóng gói thành một hoặc nhiều Worker, có thể hoạt động một cách độc lập với nhau. Các Worker NLU lưu các mô hình NLU đã huấn luyện của mỗi agent. Tương tự, các Worker dự đoán hành động lưu trữ các mô hình dự đoán hành động của mỗi agent.

Máy chủ Gateway và các Worker giao tiếp với nhau thông qua việc gửi tin nhắn (messaging) tới các hàng đợi (queue) nằm ở vị trí giữa các Worker. Các hàng đợi khác nhau nhận những loại tin nhắn yêu cầu khác nhau, ví dụ như tin nhắn yêu cầu dự đoán NLU, tin nhắn yêu cầu cập nhật mô hình,... Một số luồng giao tiếp chính giữa các thành phần này như sau:

- Luồng xử lý tin nhắn từ những ứng dụng khách:

Máy chủ Gateway có thể tiếp nhận tin nhắn từ một số ứng dụng khách. Các ứng dụng này có thể là các nền tảng chat như Zalo, Facebook, hay bất kỳ một Webchat nào, hoặc có thể là các ứng dụng hoặc thiết bị tương tác bằng giọng nói như Vadi, loa giao tiếp trong nhà thông minh. Các tin nhắn này được gửi tới hàng đợi “Dự đoán NLU”. Một trong các Worker NLU đang trong trạng thái “rỗi” (đang không xử lý tin nhắn nào khác) sẽ nhận tin nhắn, thực hiện dự đoán thực thể và ý định trong câu đầu vào từ người dùng, sau đó thêm thông tin vừa dự đoán được vào tin nhắn, gửi sang hàng đợi “Dự đoán hành động”. Worker dự đoán hành động nhận tin nhắn, dựa vào ngữ cảnh thực hiện dự đoán hành động tương ứng, tiếp theo thêm thông tin hành động vào tin nhắn và gửi sang hàng đợi “Thực thi hành động”. Worker thực thi hành động lắng nghe được tin nhắn này, nhận và tiến hành xử lý. Worker này có thể thực hiện gọi tới các dịch vụ của bên thứ 3, ví dụ như gọi nhận thông tin thời tiết, lấy đường dẫn nghe nhạc, hoặc gọi tới dịch vụ tổng hợp tiếng nói (Text To Speech - TTS) để chuyển nội dung phản hồi dạng văn bản thành dạng âm thanh. Sau đó gửi thông tin phản hồi sang hàng đợi “Phản hồi”. Máy chủ Gateway khi lắng nghe

hàng đợi “Phản hồi” sẽ nhận được nội dung phản hồi và gửi lại cho người dùng cuối.

- Luồng xử lý yêu cầu cập nhật mô hình NLU:

Do các Worker NLU lưu trữ các mô hình đã huấn luyện và hoạt động một cách độc lập với nhau, do đó mỗi khi nhà phát triển huấn luyện lại mô hình, các Worker này cũng cần cập nhật lại mô hình tương ứng.

Sau nhà phát triển thực hiện quá trình huấn luyện NLU xong, một yêu cầu cập nhật mô hình sẽ được dịch vụ huấn luyện NLU gửi tới máy chủ Gateway. Gateway nhận yêu cầu, tạo tin nhắn gửi tới hàng đợi “Cập nhật mô hình NLU”. Tin nhắn này sẽ được gửi lại tới toàn bộ các Worker NLU đang lắng nghe hàng đợi này. Mỗi Worker NLU sẽ thực hiện lấy dữ liệu mô hình mới từ dịch vụ huấn luyện NLU thông qua giao thức sFTP [39].

- Luồng xử lý yêu cầu cập nhật mô hình dự đoán hành động:

Tương tự như luồng xử lý yêu cầu cập nhật mô hình NLU, tuy nhiên Gateway sẽ gửi tin nhắn yêu cầu vào hàng đợi “Cập nhật mô hình hành động”.

Em đã trình bày toàn bộ kiến trúc tổng quan của nền tảng SmartDialog, sau đây em xin phép trình bày phần nội dung thiết kế và xây dựng Worker dự đoán hành động và xử lý vấn đề về quản lý phiên hội thoại trong máy chủ Gateway.

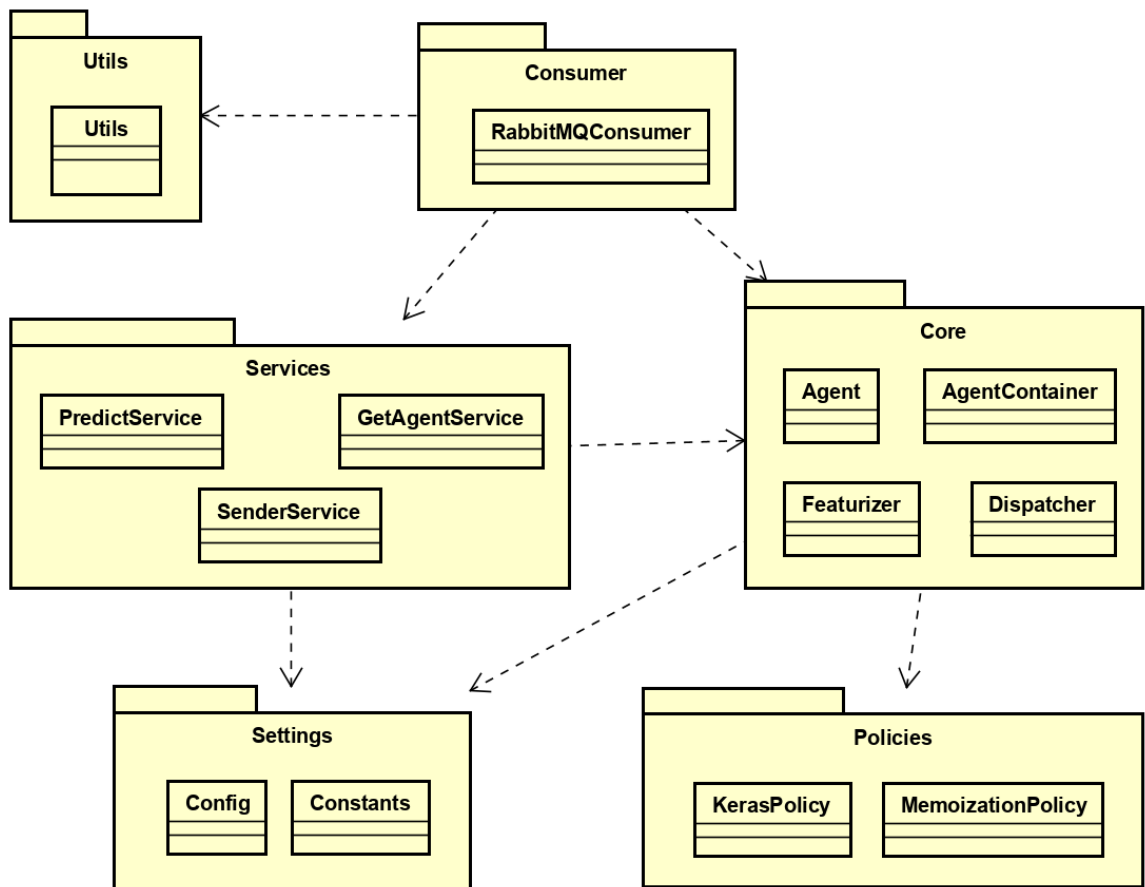
4.2 Dịch vụ dự đoán hành động

4.2.1 Thiết kế

Hình 15 mô tả thiết kế các gói và các lớp của một trong các Worker thuộc dịch vụ dự đoán hành động. Các Worker nhận tin nhắn bằng cách lắng nghe hàng đợi tin nhắn từ lớp RabbitMQConsumer. Gói Service chứa một số dịch vụ quan trọng như dịch vụ PredictService dự đoán hành động, dịch vụ GetAgentService cập nhật mô hình của agent và dịch vụ SenderService gửi tin nhắn tới các hàng đợi.

Một số lớp quan trọng trong gói core như AgentContainer chứa các Agent, Featurizer chuyển trạng thái hội thoại thành véc-tơ và Dispatcher thực thi các hành

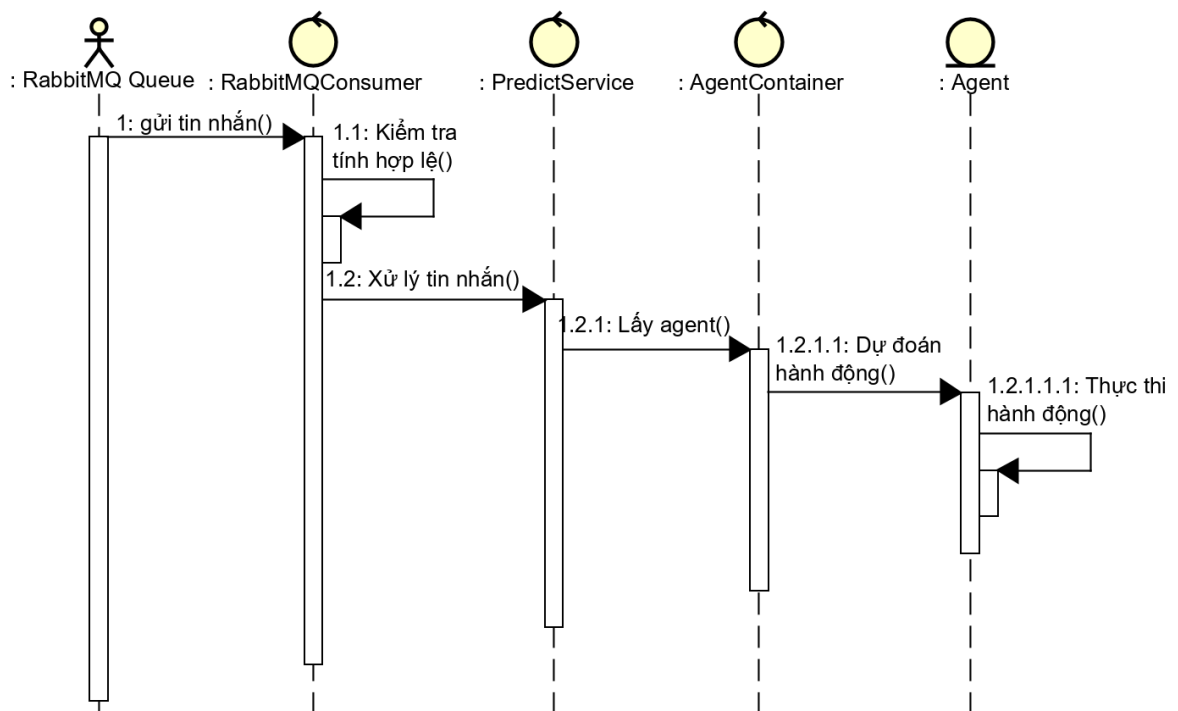
động dự đoán được. Agent sử dụng mô hình mạng LSTM trong KerasPolicy và mô hình MemoizationPolicy để dự đoán hành động.



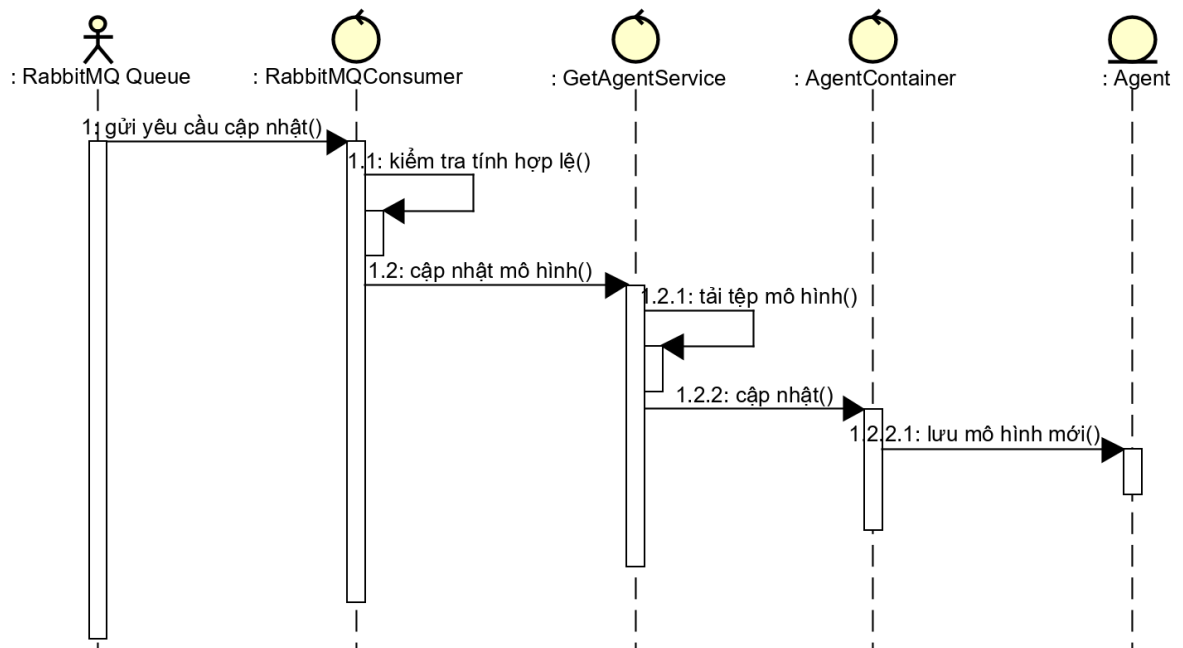
Hình 15 Biểu đồ thiết kế gói của Worker dự đoán hành động.

Hình 16 mô tả trình tự dự đoán hành động, tin nhắn được gửi từ hàng đợi tới lớp RabbitMQConsumer. Sau khi được lớp này kiểm tra tính hợp lệ, PredictService yêu cầu AgentContainer lấy mô hình lưu trong Agent ra để dự đoán hành động. Hành động sau khi dự đoán ra sẽ được thực thi.

Hình 17 mô tả trình tự cập nhật mô hình sau khi quá trình huấn luyện hành động hoàn thành. Hàng đợi RabbitMQ gửi tin nhắn yêu cầu cập nhật mô hình tới RabbitMQConsumer. Sau khi kiểm tra tính hợp lệ, lớp này gọi tới GetAgentService thực hiện tải tệp mô hình xuống (thông qua giao thức SFTP đã giới thiệu ở mục trước). Mô hình tải về sẽ được cập nhật vào Agent trong AgentContainer.



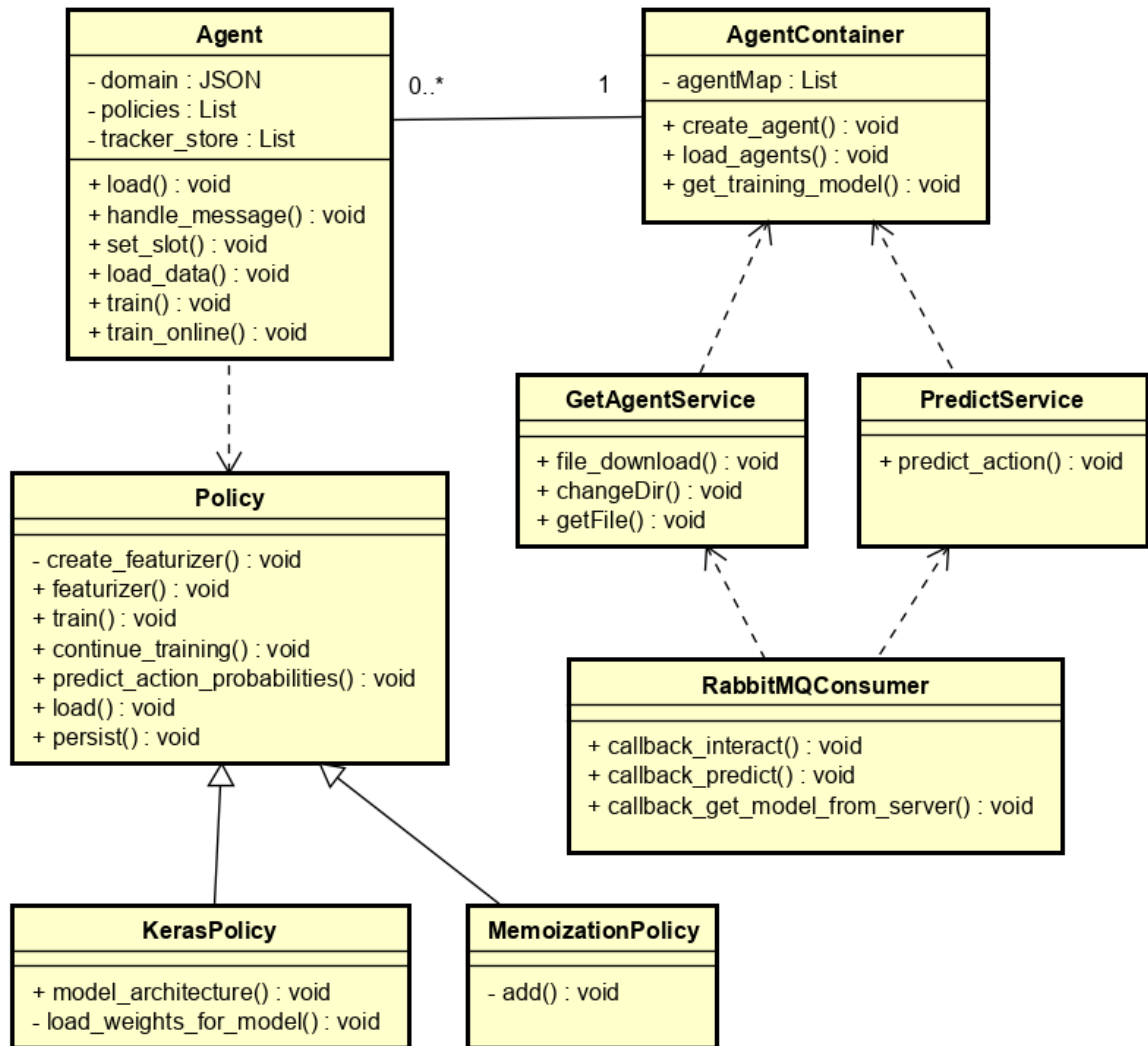
Hình 16 Biểu đồ trình tự dự đoán hành động.



Hình 17 Biểu đồ trình tự cập nhật mô hình dự đoán hành động.

Hình 18 Đưa ra thiết kế chi tiết của một số lớp chính trong dịch vụ dự đoán hành động. Lớp AgentContainer chứa một danh sách các Agent bên trong nó. Các Agent

sử dụng các lớp Policy để tải mô hình và thực hiện dự đoán hành động. Lớp Policy này được kế thừa bởi KerasPolicy và MemoizationPolicy.



Hình 18 Biểu đồ lớp chi tiết trong dịch vụ dự đoán hành động

4.2.2 Xây dựng hệ thống

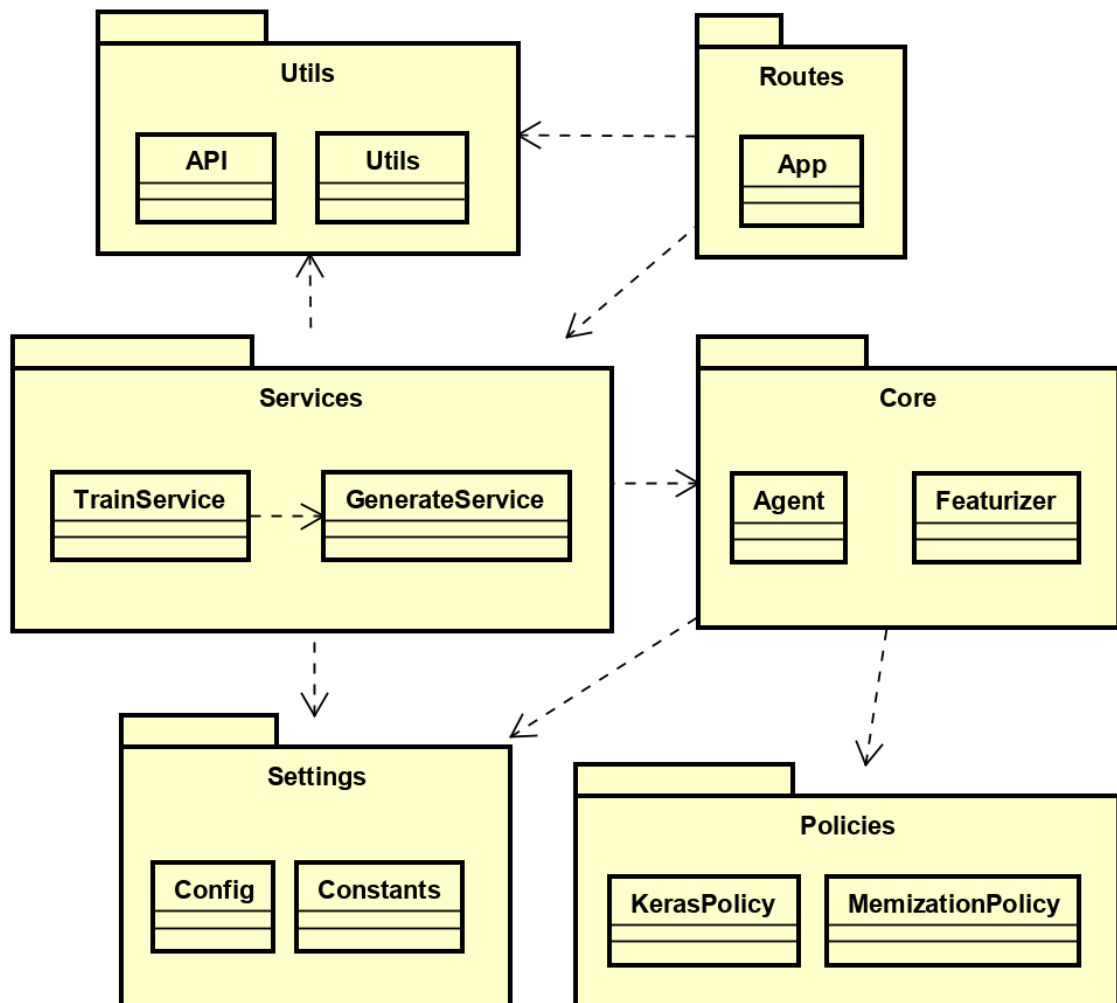
Thư viện và công cụ sử dụng

Bảng 6 Thư viện và công cụ sử dụng để xây dựng thành phần dự đoán hành động

Mục đích	Công cụ	Địa chỉ URL
IDE	PyCharm	https://www.jetbrains.com/pycharm/
CSDL	MongoDB	https://www.mongodb.com/

4.3 Dịch vụ huấn luyện hành động

4.3.1 Thiết kế

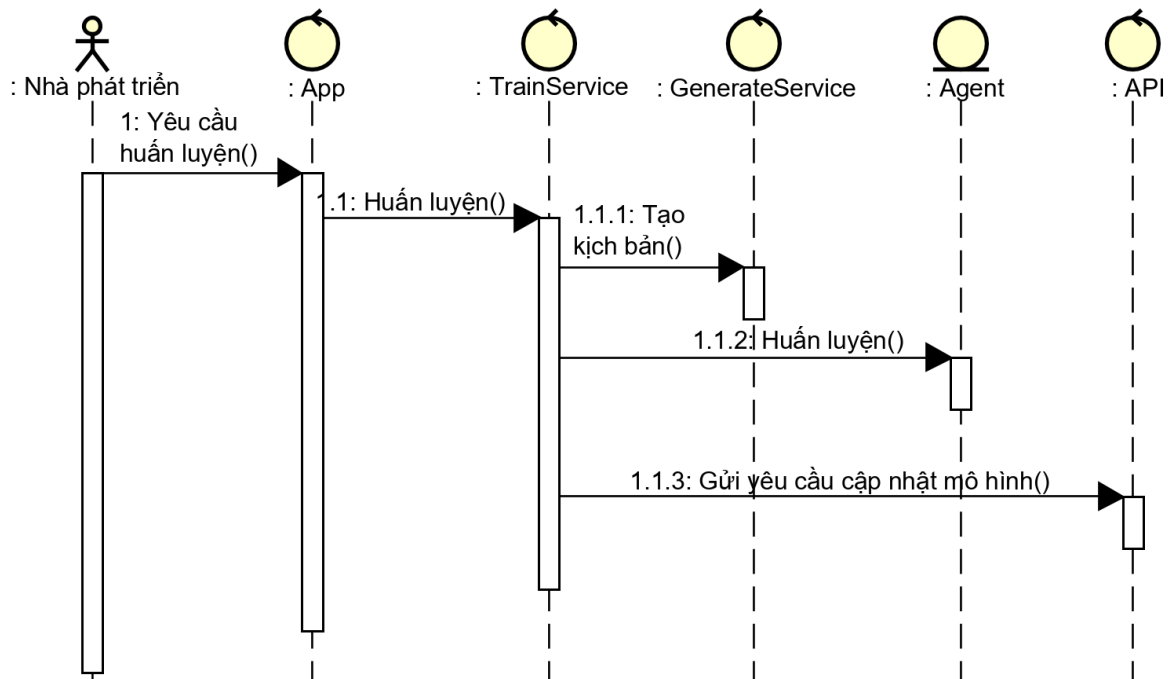


Hình 19 Biểu đồ thiết kế gói của dịch vụ huấn luyện hành động.

Hình 19 mô tả thiết kế các gói và các lớp của dịch vụ huấn luyện hành động. Dịch vụ nhận yêu cầu huấn luyện mô hình thông qua các API gửi tới App. Gói Service chứa một số dịch vụ quan trọng như dịch vụ TrainService huấn luyện các kịch bản hành động. Dịch vụ GenerateService tạo ra tập dữ liệu kịch bản đầu vào cho mô hình, gồm

các kịch bản tạo tự động, kịch bản do người dùng tạo ra khi thực hiện huấn luyện theo phương pháp học tương tác, và một số kịch bản có sẵn.

Một số lớp quan trọng trong gói core như Agent, Featurizer chuyển trạng thái hội thoại thành véc-tơ. Agent sử dụng mô hình mạng LSTM trong KerasPolicy và mô hình MemoizationPolicy để huấn luyện dữ liệu đầu vào.



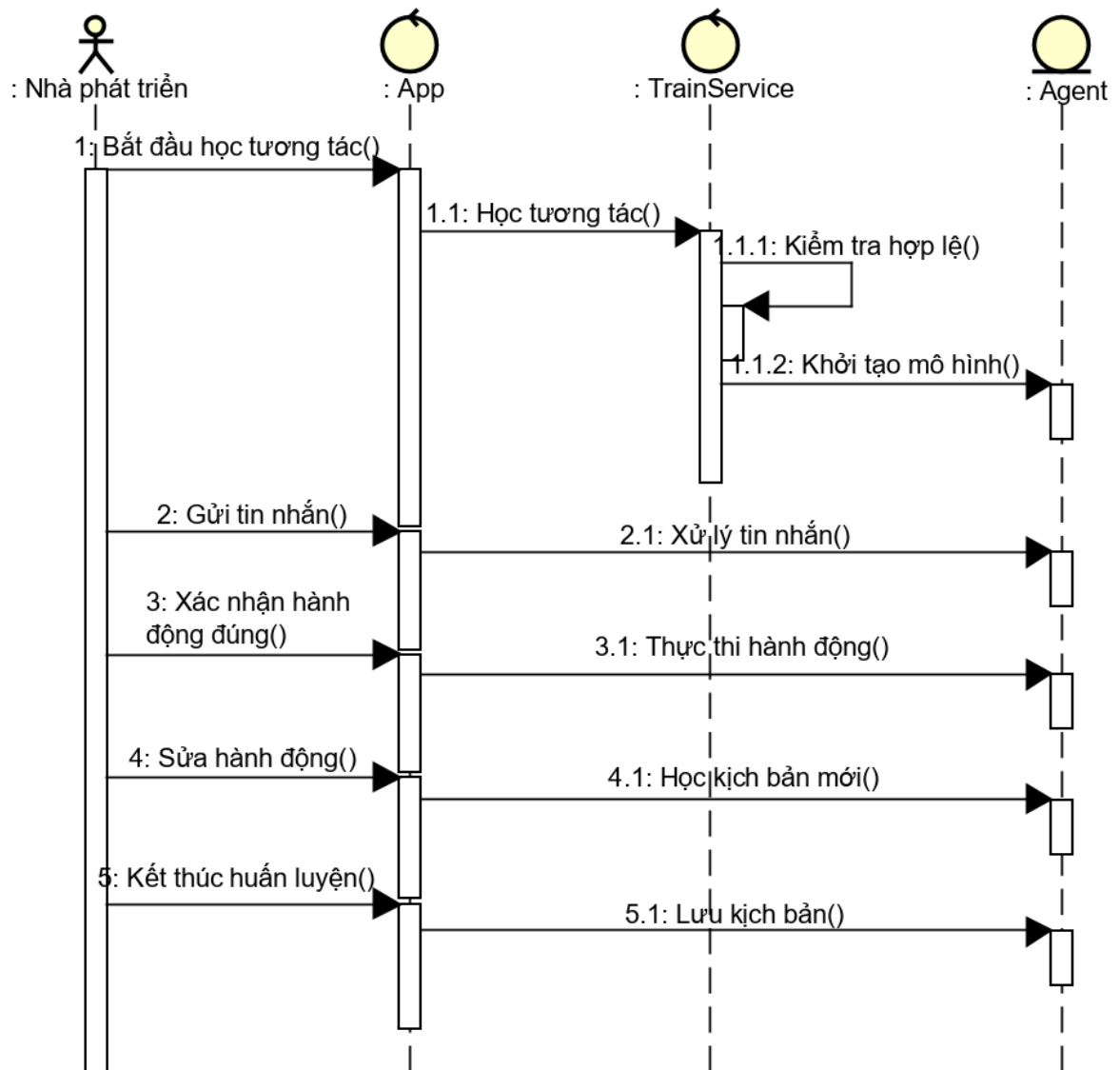
Hình 20 Biểu đồ trình tự huấn luyện mô hình dự đoán hành động.

Hình 20 mô tả quá trình huấn luyện mô hình dự đoán hành động. Nhận yêu cầu huấn luyện từ Nhà phát triển, App gọi tới TrainService yêu cầu GenerateService tạo kịch bản huấn luyện.

Sau đó lớp Agent sẽ thực hiện huấn luyện và lưu lại mô hình huấn luyện xong lên phần cứng. Cuối cùng, TrainService gọi API yêu cầu cập nhật mô hình mới tới các Worker dự đoán hành động.

Hình 21 mô tả quá trình học tương tác trong dịch vụ huấn luyện. Nhà phát triển sau khi gửi yêu cầu khởi động quá trình học tương tác, có thể gửi tin nhắn, các xác nhận hành động đúng hoặc sửa các hành động sai. Các yêu cầu này sẽ được TrainService và Agent xử lý tương ứng.

Sau khi huấn luyện xong Nhà phát triển có thể gửi yêu cầu kết thúc huấn luyện và các kịch bản mới được tạo ra sẽ lưu vào CSDL.



Hình 21 Biểu đồ trình tự học tương tác.

4.3.2 Xây dựng hệ thống

Thư viện và công cụ sử dụng

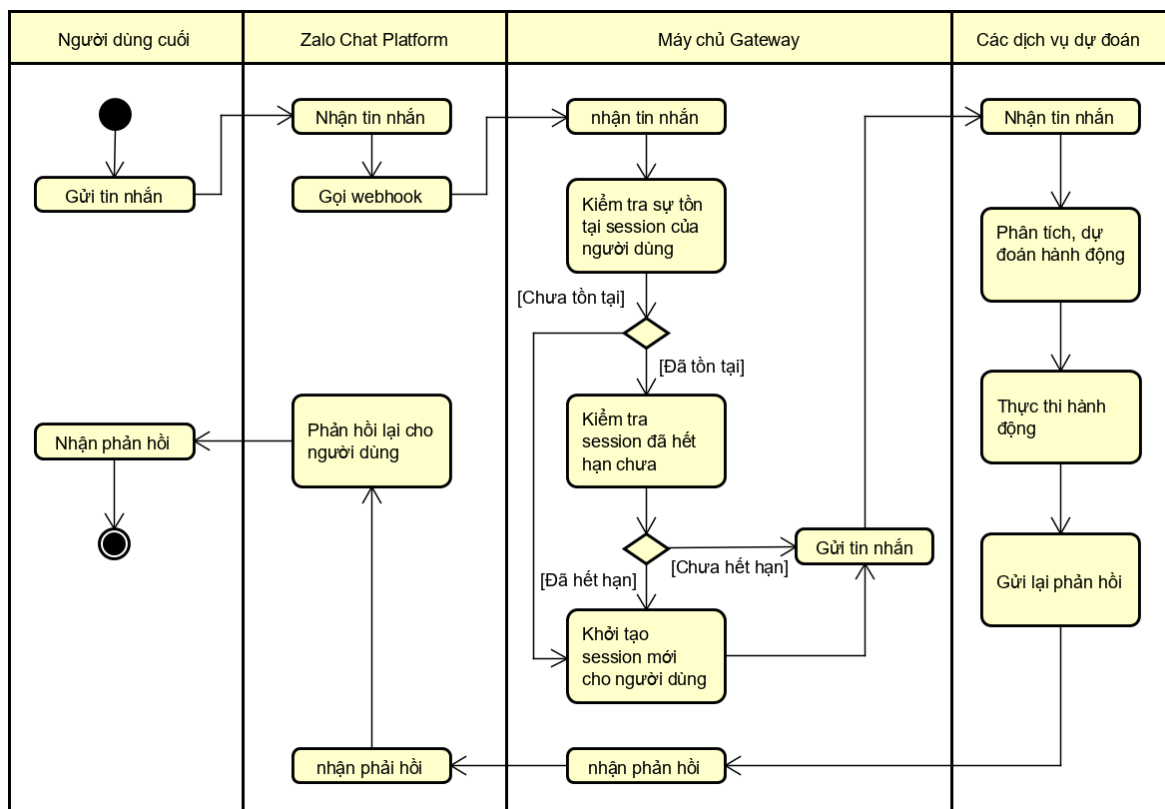
Bảng 7 Thư viện và công cụ sử dụng để xây dựng dịch vụ huấn luyện hành động

Mục đích	Công cụ	Địa chỉ URL
IDE	PyCharm	https://www.jetbrains.com/pycharm/

4.4 Quản lý phiên hội thoại trong máy chủ Gateway

Để có thể triển khai agent lên nền tảng chat Zalo thì nhiệm vụ của máy chủ Gateway là vô cùng quan trọng. Gateway không chỉ nhận thông tin đầu vào từ các nền tảng kể trên để chuyển tới cho các thành phần dự đoán NLU và dự đoán hành động, mà còn thực hiện việc quản lý các phiên hội thoại. Phần nội dung này sẽ được trình bày rõ hơn ở những mục tiếp theo đây.

4.4.1 Quy trình nghiệp vụ



Hình 22 Biểu đồ hoạt động người dùng giao tiếp với Chatbot qua Zalo.

Để hiểu rõ về vị trí cũng như nhiệm vụ của máy chủ Gateway trong việc quản lý các phiên hội thoại, ta sẽ xem xét biểu đồ hoạt động ở Hình 22.

Biểu đồ miêu tả quá trình một tin nhắn từ người dùng cuối đi qua các thành phần của hệ thống và nhận về phản hồi như thế nào. Đầu tiên, khi người dùng cuối gửi tin nhắn tới, Zalo sẽ nhận tin nhắn, lấy thêm một số các thông tin về người dùng đó và gửi tất cả những thông tin này qua một đường dẫn webhook. Đường dẫn này là một API tới máy chủ Gateway, là địa chỉ để gửi tin nhắn tới. Sau khi nhận được thông tin, Gateway tiến hành công đoạn kiểm tra và quản lý các phiên hội thoại.

Phiên hội thoại

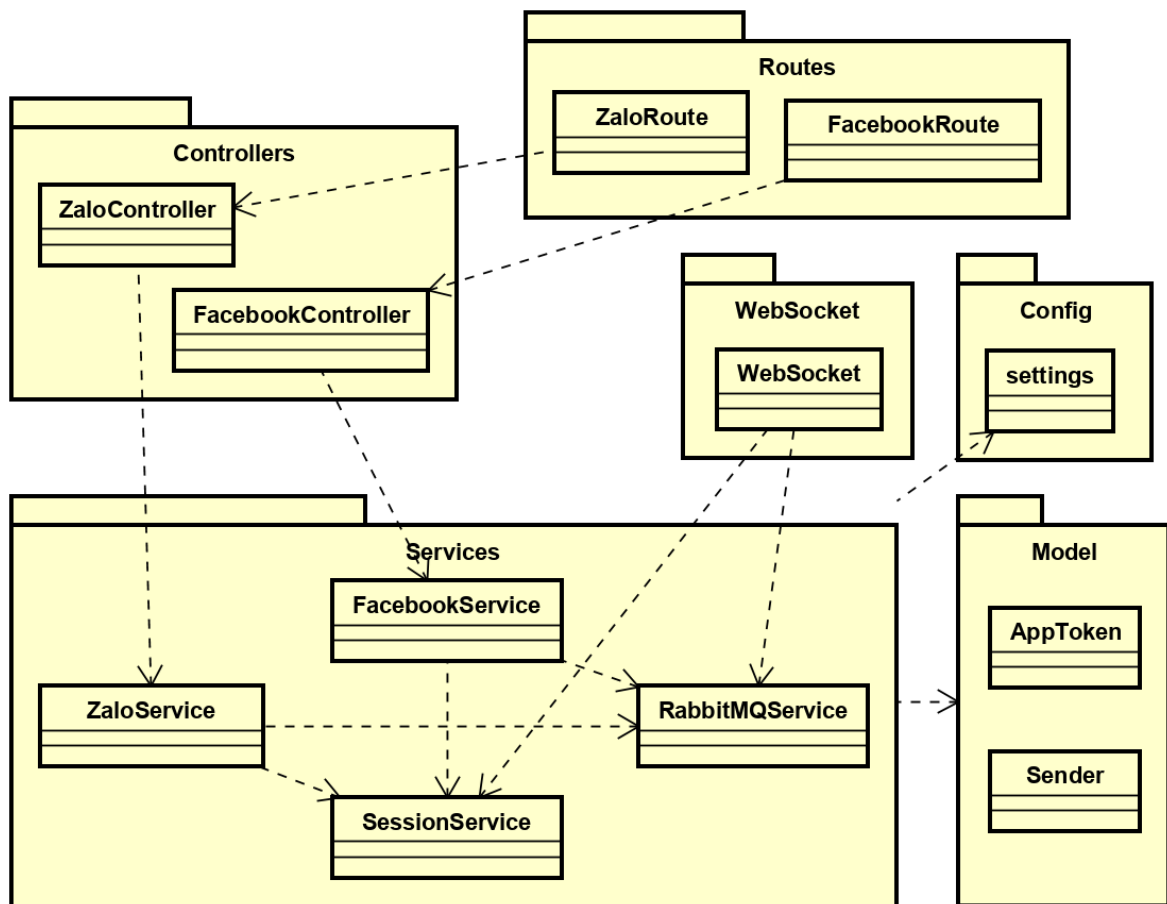
Thông thường, những cuộc hội thoại sẽ không kéo dài mãi mà đều có thời gian bắt đầu và thời gian kết thúc. Thậm chí ở trong một cuộc hội thoại dài sẽ có thể có nhiều đoạn hội thoại nhỏ liên tiếp cách nhau một khoảng thời gian ngắn nào đó, những đoạn hội thoại như vậy được gọi là các phiên hội thoại. Việc quản lý các phiên hội thoại là cần thiết, vì việc lưu trữ thông tin về cuộc hội thoại của người dùng kể cả khi họ không còn trực tuyến (online) hoặc không nhắn tin với Chatbot nữa là không cần thiết. Chatbot ở phiên hội thoại hiện tại không cần phải ghi nhớ những thông tin người dùng đã cung cấp từ những phiên hội thoại thậm chí là ở rất xa trong quá khứ.

Sau khi nhận được tin nhắn, Gateway sẽ kiểm tra xem người dùng có đang ở trong một phiên hội thoại nào với Chatbot không, nếu có thì tiếp tục kiểm tra xem phiên hội thoại đó đã hết hạn (outdated) hay chưa. Một phiên hội thoại hết hạn là phiên hội thoại đã quá một khoảng thời gian mà người dùng không gửi tin nhắn tới nữa phiên đó nữa. Nếu đã hết hạn thì Gateway sẽ đưa tin nhắn vào một phiên hội thoại mới. Sau đó, tin nhắn được gửi tới các dịch vụ dự đoán, thực thi hành động dự đoán được và lấy phản hồi trả về. Cuối cùng phản hồi dạng văn bản sẽ được trả về cho người dùng cuối, hiển thị trên giao diện chat của Zalo.

4.4.2 Thiết kế

Hình 23 Mô tả biểu đồ thiết kế các gói và một số các thành phần chính của Gateway. Các route và websocket nhận thông tin đầu vào và chuyển tới các lớp service. ZaloService sử dụng SessionService để quản lý phiên hội thoại và gửi tin nhắn qua

RabbitMQService. Một số các thông tin lấy từ CSDL về người gửi tin nhắn và mã access_token được lưu tương ứng trong lớp Sender và AppToken.

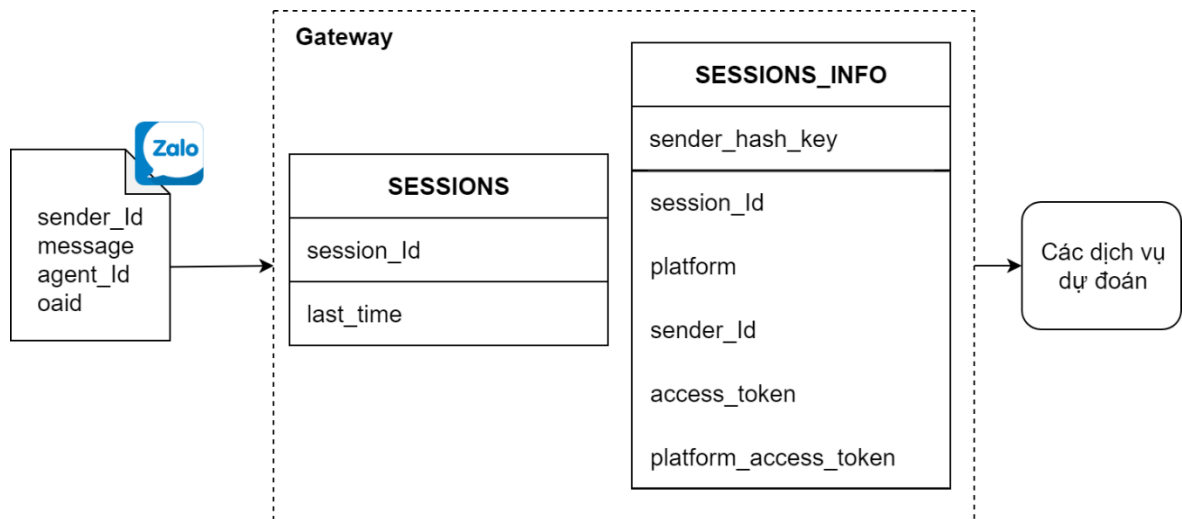


Hình 23 Biểu đồ thiết kế gói của Gateway.

Máy chủ Gateway lưu trữ thông tin về phiên hội thoại thông qua hai danh sách lưu trữ chính có tên là `SESSIONS` và `SESSIONS_INFO`.

Một đối tượng trong danh sách `SESSIONS` biểu diễn cho một phiên hội thoại với khóa của phiên hội thoại là `session_id` và trường `last_time` lưu trữ thời điểm cuối cùng mà tin nhắn được gửi tới trong phiên hội thoại đó.

Một đối tượng trong danh sách `SESSIONS_INFO` lưu trữ các thông tin liên quan tới phiên hội thoại mà một người dùng đang tham gia với khóa là trường `sender_hash_key`.



Hình 24 Thông tin lưu trữ trong phiên hội thoại.

Trường `sender_hash_key` được tạo ra theo công thức:

$$sender_hash_key = md5(sender_Id + platform_name + access_token)$$

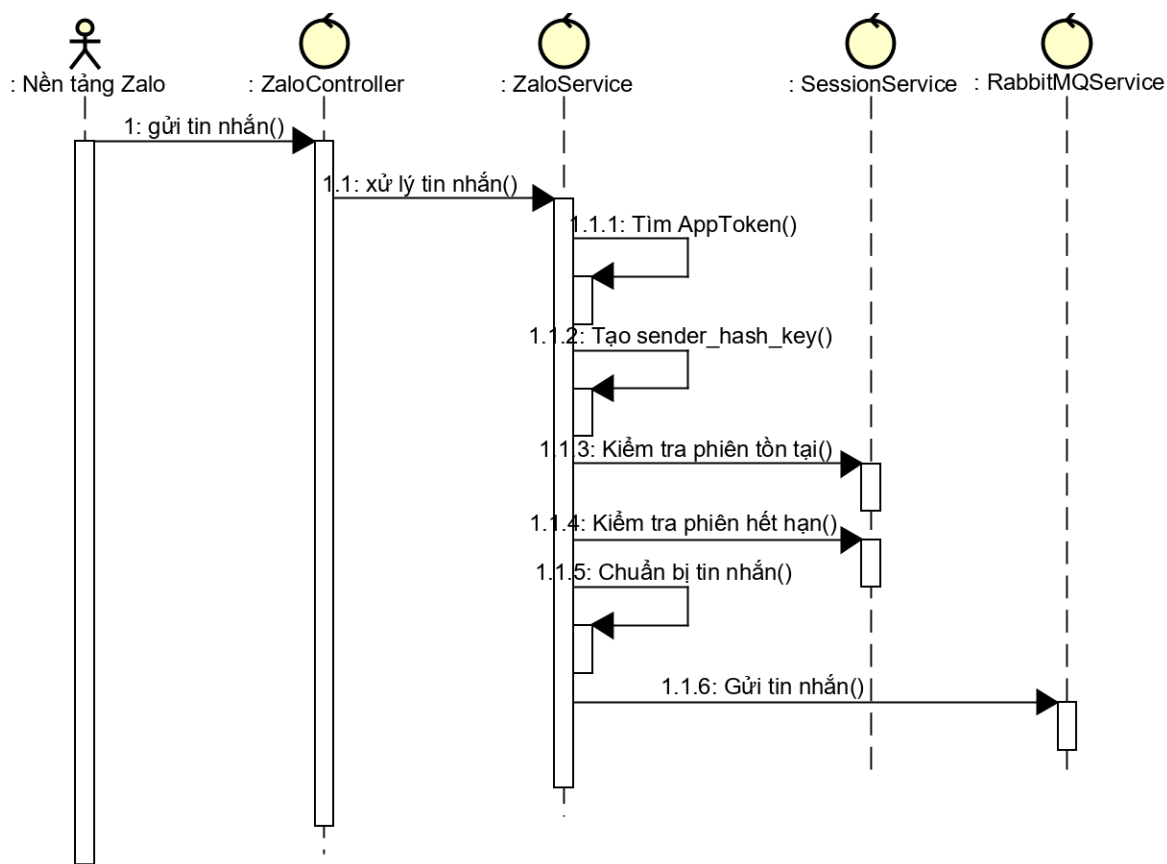
Trong đó, ta sử dụng hàm mã hóa md5 để mã hóa xâu ký tự thu được từ phép ghép các xâu nhận được từ tin nhắn tới: `sender_id` là mã của người dùng ở hệ thống gửi tin nhắn tới, `platform_name` có thể là tên của các nền tảng gửi tin nhắn tới như “zalo”, “facebook”,... và `access_token` là mã mà SmartDialog cung cấp cho nhà phát triển.

Ngoài ra `SESSIONS_INFO` còn lưu các thông tin như: mã số của phiên hội thoại (`session_id`), nền tảng gửi tin nhắn tới (`platform`), mã người dùng (`sender_id`), mã bí mật cung cấp từ SmartDialog (`access_token`), mã bí mật cung cấp từ nền tảng gửi tin nhắn tới (`platform_access_token`).

Chi tiết trình tự xử lý tin nhắn từ Zalo được thể hiện trong biểu đồ trình tự ở Hình 25.

ZaloController nhận tin nhắn từ người dùng gửi chuyển tin nhắn tới ZaloService, dựa vào `agent_id` tìm ra `access_token` và thực hiện tạo `sender_hash_key` theo công thức đã trình bày ở trước. Nếu SessionService kiểm tra phiên hội thoại này chưa tồn tại, nó sẽ tạo ra một phiên hội thoại mới và tạo ra hai đối tượng `SESSIONS` và `SESSIONS_INFO` mới. Còn nếu SessionService kiểm tra thấy tin nhắn lần cuối gửi cách tin nhắn hiện tại quá 5 phút, thì sẽ trả về kết quả phiên hết hạn. Khi đó, một

phiên mới sẽ được tạo ra, và các thông tin trong SESSIONS_INFO của phiên hết hạn sẽ được lưu lại ở trong phiên mới vừa tạo.



Hình 25 Biểu đồ trình tự người dùng gửi tin nhắn Zalo.

4.4.3 Xây dựng hệ thống

Thư viện và công cụ sử dụng

Bảng 8 Thư viện và công cụ sử dụng để xây dựng Gateway

Mục đích	Công cụ	Địa chỉ URL
IDE	Visual Studio Code	https://code.visualstudio.com/
CSDL	MongoDB	https://www.mongodb.com/
CSDL trong bộ nhớ	Redis	https://redis.io/
Backend Framework	Express.js	https://expressjs.com/

Redis là một dạng CSDL theo kiểu key-value (khóa-giá trị), cung cấp khả năng lưu trữ và truy cập dữ liệu với tốc độ nhanh.

4.5 Triển khai

4.5.1 Cách thức triển khai

Như đã đề cập ở phần 4.1, các thành phần trong phân hệ quản lý hội thoại được triển khai nằm trong mô hình Microservice của nền tảng SmartDialog. Các Worker của dịch vụ dự đoán hành động và dịch vụ huấn luyện hành động được đóng gói và triển khai trên các môi trường Anaconda [40]. Còn máy chủ Gateway được triển khai bằng trình quản lý PM2 [41, p. 2], đây là trình quản lý tiến trình dành cho các ứng dụng sử dụng công nghệ Nodejs.

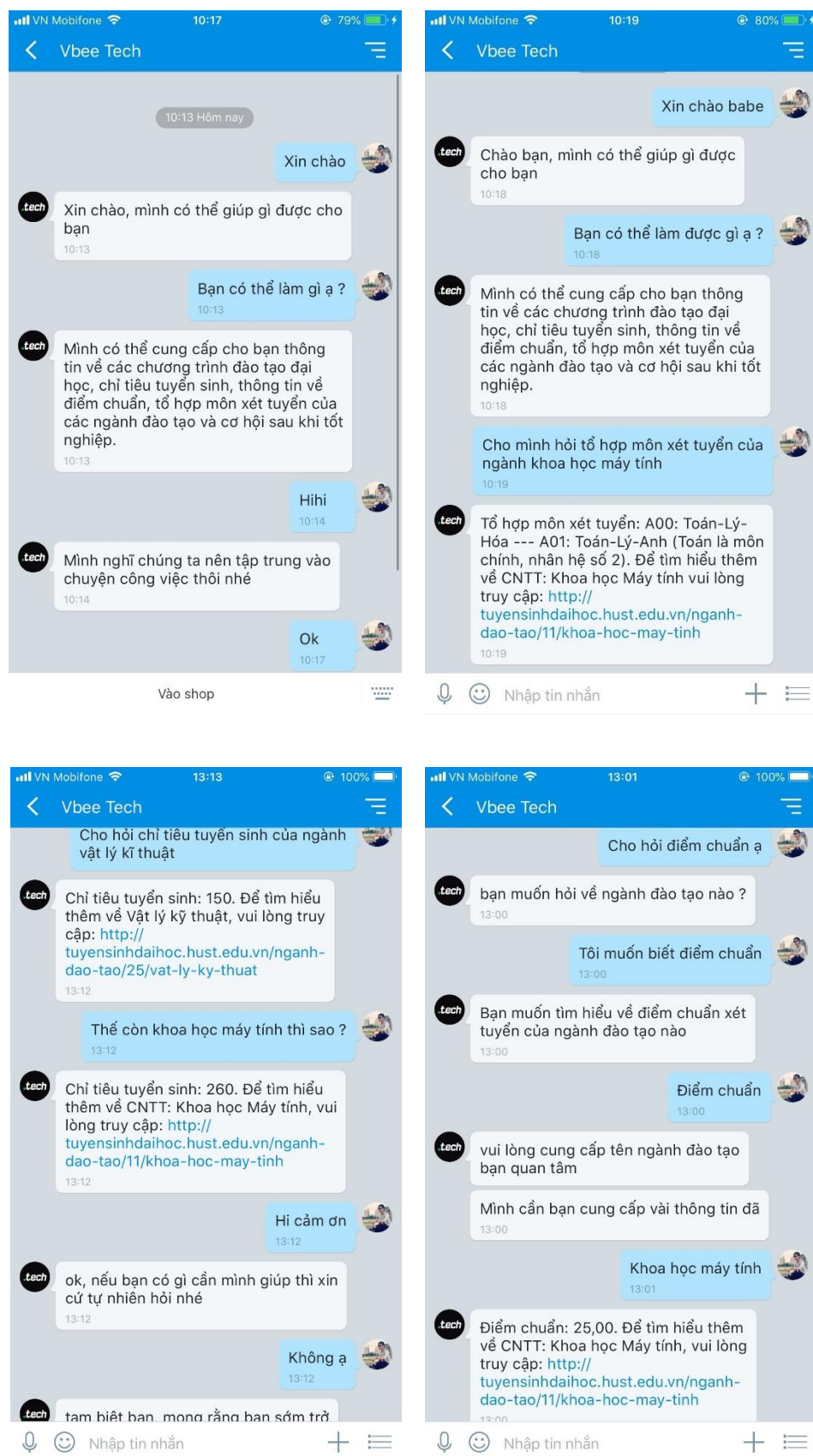
4.5.2 Kết quả triển khai

Toàn bộ các thành phần thuộc phân hệ quản lý hội thoại đã được triển khai thành công và chạy thật trên các máy chủ, cùng địa chỉ ip và cổng thực sự. Hơn thế nữa, nền tảng SmartDialog đã nghiên cứu và đang triển khai một số hệ thống hội thoại như Chatbot VCB, Chatbot ACB, Chatbot Mofin, loa thông minh Milo (của nhà thông minh Lumi), Chatbot Tư vấn tuyển sinh HUST,.... Phần tiếp theo sẽ trình bày về tích hợp Chatbot Tư vấn tuyển sinh trên nền tảng Zalo.

4.5.3 Tích hợp Chatbot Tư vấn tuyển sinh với nền tảng Zalo

Zalo là ứng dụng nhắn tin và gọi điện miễn phí hoạt động trên nền tảng di động và máy tính [42]. Hơn thế nữa, Zalo cung cấp nhiều dịch vụ hỗ trợ các tổ chức, doanh nghiệp hoặc nhà bán lẻ tương tác tự động với khách hàng của mình bằng Chatbot.

Sau khi đã huấn luyện xong bộ dữ liệu tư vấn tuyển sinh (giới thiệu ở mục 3.5.1), mục tiêu tiếp theo là tạo ra một Chatbot của ĐH BKHN có thể giao tiếp với học sinh, sinh viên qua kênh chat Zalo, hướng tới phần nào giảm gánh nặng nhân lực cho nhà trường, tạo ra một kênh thông tin mới mẻ, thú vị và hữu ích cho học sinh, sinh viên có nhu cầu nhập học.



Hình 26 Một số đoạn hội thoại với Chatbot tư vấn tuyển sinh trên nền tảng Zalo.

Ta thực hiện kết nối agent tư vấn tuyển sinh trên nền tảng SmartDialog với Zalo. Sau khi đã phát triển và triển khai thành công máy chủ Gateway thì việc tích hợp giờ chỉ còn là những thao tác đơn giản trên giao diện của hai nền tảng này.

Các bước kết nối gồm (i) tạo ứng dụng trên Zalo, (ii) tạo liên kết từ ứng dụng tới Official Account, (iii) xin xét quyền sử dụng API cho Official Account, (iv) lấy Access Token của ứng dụng và điền trên giao diện SmartDialog và (v) Đăng ký Webhook của Chatbot cho ứng dụng.

Hình 26 đưa ra một số ví dụ về đoạn hội thoại của người dùng cuối và Chatbot sau khi triển khai. Những đoạn hội thoại ngắn cho thấy Chatbot có khả năng ghi nhớ những thông tin người dùng đã cung cấp và phản hồi người dùng theo ngữ cảnh hội thoại. Những đoạn hội thoại như vậy sẽ giúp người dùng cuối cảm thấy thú vị, thân thiện và có thể giao tiếp một cách tự nhiên hơn.

Chương 5 Kết luận và hướng phát triển

5.1 Kết luận

Tổng kết lại, em đã trình bày về mô hình quản lý hội thoại triển khai trên nền tảng tương tác hội thoại thông minh SmartDialog. Nêu ra phương pháp ứng dụng mô hình học sâu LSTM vào bài toán dự đoán hành động trong quản lý hội thoại. Đưa ra mô hình học tương tác dựa trên tư tưởng của học trực tuyến, trong đó dữ liệu được người dùng đưa vào lần lượt để huấn luyện và sửa sai dần cho mô hình. Em cũng trình bày về cơ chế quản lý phiên hội thoại trên máy chủ Gateway kết nối Zalo và các dịch vụ dự đoán. Và cuối cùng triển khai một case study Chatbot với bộ dữ liệu tư vấn tuyển sinh tích hợp trên nền tảng Zalo. Hệ thống đưa ra dự đoán trên bộ dữ liệu này với độ chính xác cao lên tới 94,8%.

Giải pháp đề xuất cho phép xây dựng những hệ thống hội thoại tự nhiên, tạo cảm giác thân thiện như khi giao tiếp với người thật. Giải quyết các vấn đề về yêu cầu dữ liệu lớn ở các nền tảng hoặc giải pháp khác, cung cấp khả năng tái sử dụng phần mô hình dự đoán hành động cho nhiều ngôn ngữ đầu vào và đầu ra khác nhau. Giải pháp học tương tác giúp người dùng huấn luyện mô hình một cách đơn giản và trực quan, đồng thời tạo ra một kênh cung cấp dữ liệu kịch bản đầu vào. Giải pháp giúp nhà phát triển không phải chuyên gia có thể dễ dàng tiếp cận với quy trình và thực hiện việc phát triển một hệ thống hội thoại thông minh.

Tuy vậy, khi số lượng kịch bản trong tập huấn luyện tăng lên, đồng nghĩa với thời gian huấn luyện dài hơn so với những nền tảng truyền thống khác. Tốc độ huấn luyện mô hình hội thoại mới chậm có thể khiến các nhà phát triển không hài lòng. Ngoài ra, hiện tại các hành động của hệ thống chỉ hỗ trợ phản hồi dạng văn bản hoặc tiếng nói, cần thêm các dạng dữ liệu khác phong phú hơn.

Giáp pháp quản lý hội thoại trình bày trong đồ án được triển khai trong nền tảng SmartDialog [1], [2], nền tảng đã có bài báo được chấp nhận tại **hội nghị NLPIR 2019 (International Conference on Natural Language Processing and Information Retrieval)** và đạt **giải Ba Cuộc thi sản phẩm sáng tạo viện CNTT 2019**.

5.2 Hướng phát triển

Trong thời gian sau khi đồ án này hoàn thành, em sẽ tiếp tục hoàn thiện mô hình và các chức năng của phần quản lý hội thoại ở trong nền tảng SmartDialog. Bổ sung thêm các dạng phản hồi mới cho hành động hệ thống như hình ảnh, icon, dạng ô lựa chọn,... Ngoài ra, với các bộ dữ liệu lớn, nhiều kịch bản, thì thời gian huấn luyện mô hình và thời gian chờ mô hình sẵn sàng khi học tương tác khá lâu. Em sẽ sửa đổi và tối ưu hơn nữa các tham số của mô hình LSTM, và thêm vào các điều kiện dừng huấn luyện sớm để cải thiện tốc độ huấn luyện mô hình.

Một hướng phát triển xa hơn đó là thử nghiệm và sử dụng cơ chế Attention (cơ chế chú ý) cho mô hình dự đoán hành động. Trong một đoạn hội thoại, không phải câu nói nào từ đối phương cũng có ý nghĩa đối với mục đích của cuộc hội thoại đó. Ví dụ một số câu mang ý nghĩa tán gẫu như “hihi”, “tớ yêu chatbot”, “bạn thông minh quá nhỉ”,... Mô hình sẽ có thể chỉ cần quan tâm tới một số trạng thái hội thoại nhất định mà không phải là toàn bộ *max_history* trạng thái như đã trình bày ở mục 3.2.1. Do đó, sử dụng cơ chế Attention có thể giải quyết được vấn đề này.

Ngoài ra, hiện tại chức năng học tương tác chỉ cho phép nhà phát triển sửa các hành động bị dự đoán sai. Chức năng này hoàn toàn có thể được phát triển tiếp, kết hợp với thành phần NLU để hỗ trợ việc sửa các ý định và thực thể bị mô hình dự đoán sai từ câu đầu vào của người dùng.

Tài liệu tham khảo

- [1] Trang Nguyen Thi Thu, Ky Nguyen Hoang, Son Hoang, Huan Nguyen Danh, and Hung Nguyen Thanh, “Natural language understanding in SmartDialog - A platform for Vietnamese intelligent interactions,” presented at the International Conference on Natural Language Processing and Information Retrieval, Tokushima, Japan, 2019.
- [2] “Smartdialog.” [Online]. Available: <https://smartdialog.vbee.vn>. [Accessed: 15-May-2019].
- [3] “What is Natural Language Processing?” [Online]. Available: https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html. [Accessed: 10-May-2019].
- [4] “What is Human-Computer Interaction (HCI)?,” *The Interaction Design Foundation*. [Online]. Available: <https://www.interaction-design.org/literature/topics/human-computer-interaction>. [Accessed: 10-May-2019].
- [5] S. Arora, K. Batra, and S. Singh, “Dialogue System: A Brief Review,” *ArXiv13064134 Cs*, Jun. 2013.
- [6] G. López, L. Quesada, and L. A. Guerrero, “Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces,” in *Advances in Human Factors and Systems Interaction*, 2018, pp. 241–250.
- [7] S. Janarthanam, *Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills*, 1 edition. Packt Publishing, 2017.
- [8] M. Nakano *et al.*, “WIT: A Toolkit for Building Robust and Real-Time Spoken Dialogue Systems,” in *SIGDIAL Workshop*, 2000.
- [9] V. Ilievski, “Building Advanced Dialogue Managers for Goal-Oriented Dialogue Systems,” *ArXiv180600780 Cs*, Jun. 2018.
- [10] S. Larsson and D. R. Traum, “Information state and dialogue management in the TRINDI dialogue move engine toolkit,” *Nat. Lang. Eng.*, vol. 6, pp. 323–340, 2000.

- [11] V. Zue *et al.*, “JUPITER: a telephone-based conversational interface for weather information,” *IEEE Trans. Speech Audio Process.*, vol. 8, no. 1, pp. 85–96, Jan. 2000.
- [12] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.
- [13] O. Vinyals and Q. Le, “A Neural Conversational Model,” *ArXiv150605869 Cs*, Jun. 2015.
- [14] A. Bordes, Y.-L. Boureau, and J. Weston, “Learning End-to-End Goal-Oriented Dialog,” *ArXiv160507683 Cs*, May 2016.
- [15] H. Cuayáhuitl, “SimpleDS: A Simple Deep Reinforcement Learning Dialogue System,” *ArXiv160104574 Cs*, Jan. 2016.
- [16] B. Dhingra *et al.*, “Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access,” *ArXiv160900777 Cs*, Sep. 2016.
- [17] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, “End-to-End Task-Completion Neural Dialogue Systems,” *ArXiv170301008 Cs*, Mar. 2017.
- [18] X. Li, Z. C. Lipton, B. Dhingra, L. Li, J. Gao, and Y.-N. Chen, “A User Simulator for Task-Completion Dialogues,” *ArXiv161205688 Cs*, Dec. 2016.
- [19] P.-H. Su *et al.*, “Continuously Learning Neural Dialogue Management,” *ArXiv160602689 Cs*, Jun. 2016.
- [20] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A comparative study of fine-tuning deep learning models for plant disease identification,” *Comput. Electron. Agric.*, Mar. 2018.
- [21] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Mach. Learn.*, vol. 8, no. 3, pp. 229–256, May 1992.
- [22] J. D. Williams, K. Asadi, and G. Zweig, “Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning,” *ArXiv170203274 Cs*, Feb. 2017.
- [23] M. Gašić, D. Kim, P. Tsiakoulis, and S. Young, “Distributed dialogue policies for multi-domain statistical dialogue management,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5371–5375.
- [24] Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press; 1 edition, 2014.

- [25] Tiep Vu, “Machine Learning cơ bản,” *Tiep Vu’s blog*. [Online]. Available: <https://machinelearningcoban.com/>. [Accessed: 12-May-2019].
- [26] “CS231n Convolutional Neural Networks for Visual Recognition.” [Online]. Available: <https://cs231n.github.io/>. [Accessed: 21-May-2019].
- [27] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” *ArXiv171005941 Cs*, Oct. 2017.
- [28] Aurélien Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. 2019.
- [29] Y. E. NESTEROV, “A method for solving the convex programming problem with convergence rate $O(1/k^2)$,” *Dokl Akad Nauk SSSR*, vol. 269, pp. 543–547, 1983.
- [30] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *J. Mach. Learn. Res.*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [31] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Dec. 2014.
- [32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” p. 30.
- [33] “Understanding LSTM Networks -- colah’s blog.” [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed: 11-May-2019].
- [34] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [35] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, “Rasa: Open Source Language Understanding and Dialogue Management,” *ArXiv171205181 Cs*, Dec. 2017.
- [36] J. Brownlee, “A Gentle Introduction to k-fold Cross-Validation,” *Machine Learning Mastery*, 22-May-2018. .
- [37] “The most popular database for modern apps | MongoDB.” [Online]. Available: <https://www.mongodb.com/>. [Accessed: 19-May-2019].
- [38] “Messaging that just works — RabbitMQ.” [Online]. Available: <https://www.rabbitmq.com/>. [Accessed: 09-May-2019].
- [39] “sFTP là gì và cách sử dụng sFTP để di chuyển dữ liệu trên máy chủ Linux.” [Online]. Available: <https://thachpham.com/linux-webserver/sftp-la-gi-va-cach-su-dung-sftp-de-upload-du-lieu-tren-may-chu-linux.html>. [Accessed: 19-May-2019].

- [40] “Anaconda,” *Anaconda*. [Online]. Available: <https://www.anaconda.com/>. [Accessed: 23-May-2019].
- [41] “Overview | PM2 Documentation,” *PM2*. [Online]. Available: [/doc/en/runtime/overview/](https://pm2.keymetrics.io/docs/en/runtime/overview/). [Accessed: 23-May-2019].
- [42] “Zalo.” [Online]. Available: <https://zalo.me>. [Accessed: 15-May-2019].