

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

NGUYỄN HỮU CƯỜNG

NGHIÊN CỨU HỆ THỐNG TRỢ LÝ THÔNG MINH ẢO

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2017

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

NGUYỄN HỮU CƯỜNG

NGHIÊN CỨU HỆ THỐNG TRỢ LÝ THÔNG MINH ẢO

Chuyên ngành: Hệ thống thông tin

Mã số: 60480104

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Người hướng dẫn khoa học: TS. Nguyễn Văn Vinh

Hà Nội - 2017

LỜI CAM ĐOAN

Tôi là Nguyễn Hữu Cường, học viên khóa K21, ngành Công nghệ thông tin, chuyên ngành Hệ Thống Thông Tin. Tôi xin cam đoan luận văn “Nghiên cứu hệ thống trợ lý thông minh ảo” là do tôi nghiên cứu, tìm hiểu và phát triển dưới sự hướng dẫn của TS. Nguyễn Văn Vinh. Luận văn không phải sự sao chép từ các tài liệu, công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày 05 tháng 04 năm 2017

LỜI CẢM ƠN

Đầu tiên tôi xin gửi lời cảm ơn tới các thầy cô Trường Đại học Công nghệ, Đại học Quốc Gia Hà Nội đã tận tình giảng dạy và truyền đạt kiến thức trong suốt khóa học cao học vừa qua. Tôi cũng xin được gửi lời cảm ơn đến các thầy cô trong Bộ môn Hệ thống thông tin cũng như khoa công nghệ thông tin đã mang lại cho tôi những kiến thức vô cùng quý giá và bổ ích trong quá trình học tập tại trường.

Đặc biệt xin chân thành cảm ơn thầy giáo, TS. Nguyễn Văn Vinh, người đã định hướng, giúp đỡ, trực tiếp hướng dẫn và tận tình chỉ bảo tôi trong suốt quá trình nghiên cứu, xây dựng và hoàn thiện luận văn này.

Tôi cũng xin được cảm ơn tới gia đình, những người thân, các đồng nghiệp và bạn bè đã thường xuyên quan tâm, động viên, chia sẻ kinh nghiệm, cung cấp các tài liệu hữu ích trong thời gian học tập, nghiên cứu cũng như trong suốt quá trình thực hiện luận văn tốt nghiệp.

Hà Nội, ngày 05 tháng 10 năm 2017

MỤC LỤC

MỤC LỤC	iii
DANH MỤC KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT	i
DANH MỤC HÌNH VẼ VÀ ĐỒ THI	ii
DANH MỤC BẢNG BIỂU	iii
MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG TRỢ LÝ ẢO	2
1.1. Giới thiệu về hệ thống trợ lý ảo.....	2
1.2. Cấu trúc hệ thống trợ lý ảo.....	3
1.3. Nhận dạng giọng nói tự động (ASR - Automatic Speech Recognition).....	3
1.4. Xử lý hình ảnh IMM (Image Matching).....	4
1.5. Quản lý câu hỏi trả lời QA (Question-Answering).....	5
1.5.1. Cấu tạo của quản lý câu hỏi.....	5
1.5.2. Xử lý ngôn ngữ tự nhiên trong Quản lý câu hỏi trả lời.....	6
1.5.3. Các mức của NLP.....	6
1.6. Ứng dụng NLP trong chatbot.....	8
1.6.1. Cấu tạo hệ thống chatbot.....	8
1.6.2. Cấu trúc hộp thoại chatbot.....	8
1.6.3. Xử lý dữ liệu trong chatbot.....	9
CHƯƠNG 2: MỘT SỐ KỸ THUẬT SỬ DỤNG TRONG TRỢ LÝ ẢO	11
2.1. Mạng học sâu DNNs.....	11
2.1.1. Khái niệm mạng học sâu.....	11
2.1.2. Xây dựng mô hình.....	11
2.1.3. Vấn đề và giải pháp khắc phục.....	13
2.2. Quản lý hội thoại.....	15
2.2.1. Mô hình quản lý dựa trên khung (Frame based dialog Agents).....	15
2.2.2. VoiceXML.....	16
2.3. Mô hình sinh hội thoại.....	18
2.3.1. Ý nghĩa chính của mô hình sinh hội thoại.....	18
2.3.2. Đặc điểm của mô hình hội thoại.....	18
2.4. Nhận xét.....	19
CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ	20
3.1. Phát triển bài toán.....	20
3.2. Chương trình thực nghiệm.....	20
3.2.1. Cấu trúc Api.ai.....	21
3.2.2. Thành phần API.AI.....	21
3.2.3. Mô hình xây dựng.....	23
3.2.4. Giao diện ứng dụng.....	24
3.3. Dữ liệu thực nghiệm.....	26
3.4. Mô hình hội thoại.....	29
3.5. Dữ liệu phân tích.....	32
3.6. Dữ liệu câu hỏi.....	33
3.7. Nhận xét và kết quả.....	36
KẾT LUẬN	38
TÀI LIỆU THAM KHẢO	39
PHỤ LỤC	40

DANH MỤC KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

Từ viết tắt	Từ chuẩn	Diễn giải
AI	Artificial Intelligence	Trí tuệ nhân tạo
ML	Machine Learning	Máy học, máy móc có khả năng học tập
RNN	Recurrent Neural Network	Mạng nơ ron tái phát
LSTM	Long short-term memory	Mạng cải tiến để giải quyết vấn đề phụ thuộc quá dài
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
SDK	Support Development Kit	Bộ công cụ hỗ trợ phát triển
API	Application Programming Interface	Giao diện lập trình ứng dụng
QA	Question Answering	Các cặp câu hỏi đáp
IPA	Intelligent Personal Assistants	Ứng dụng Hỗ trợ Cá nhân Thông minh
ASR	Automatic Speech Recognition	Xử lý giọng nói
QC	Query Classifier	Phân loại truy vấn
HMM	Hidden Markov	Mô hình Hidden Markov
GMM	Gaussian	mô hình hỗn hợp Gaussian
IMM	Image Matching	Xử lý hình ảnh
SURF	Speeded up robust features	Nhận dạng hình ảnh
QA	Question-Answering	Dịch vụ câu hỏi trả lời
NLU	Natural language understanding	Hiểu ngôn ngữ tự nhiên
CRF	Conditional Random Fields	Mô hình CRF
FSA	Finite State Automata	Mô hình dựa trên máy trạng thái hữu hạn
FSM	Finite State Machine	Máy trạng thái hữu hạn
FST	Finite State Transducer	Máy chuyển đổi trạng thái hữu hạn
A-FSM	Acceptor Finite State Machine	Máy trạng thái hữu hạn chấp nhận
WFST	Weighted Finite State Transducer	Máy chuyển đổi trạng thái hữu hạn có trọng số
DNN	Deep Neural Networks	Mô hình học máy

DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ

Hình 1.1: Sơ đồ hệ thống trợ lý ảo	3
Hình 1.2: Sơ đồ xử lý âm thanh.....	4
Hình 1.3: Sơ đồ xử lý hình ảnh.....	5
Hình 1.4: Quản lý câu hỏi trả lời	6
Hình 1.5: Mô hình cơ bản của hệ thống chatbot	8
Hình 1.6: Cấu trúc hộp thoại chatbot	8
Hình 2.7: Kiến trúc seq2seq.....	10
Hình 2.1: Biểu diễn mô hình mạng	11
Hình 2.2: Mạng nơ ron tái phát RNN	12
Hình 2.4: Kiến trúc tự động trạng thái hữu hạn cho hộp thoại dựa trên khung.....	15
Hình 2.5: Ví dụ về mô hình dialogue dựa trên Frame	15
Hình 2.6: Ví dụ về mô hình VoiceXML	17
Hình 3.1: Sơ đồ phát triển bài toán	20
Hình 3.2: Sơ đồ hoạt động của Api.ai.....	21
Hình 3.3: Giao diện Api.ai intents	22
Hình 3.4: Giao diện Api.ai contexts.....	23
Hình 3.5: Giao diện Api.ai webhook	23
Hình 3.6: Mô hình xây dựng giao diện.....	24
Hình 3.7: Giao diện Chatbot	25
Hình 3.8: Mô hình hội thoại.....	29
Hình 3.9: Dữ liệu đồ uống.....	30
Hình 3.10: Dữ liệu thức ăn.....	30
Hình 3.11: Dữ liệu địa điểm.....	31
Hình 3.12: Dữ liệu thời tiết.....	31
Hình 3.13: Dữ liệu truy vấn mỗi phiên làm việc.....	32
Hình 3.14: Dữ liệu phân tích tương tác	32
Hình 3.15: Dữ liệu intents	32

DANH MỤC BẢNG BIỂU

Bảng 3.1: Kết quả thử nghiệm.....	33
Bảng 3.2: Dữ liệu câu hỏi.....	33

MỞ ĐẦU

Ngày nay, trí tuệ nhân tạo đang ngày càng phát triển mạnh mẽ. Các hãng lớn như Apple, Microsoft, Google đều đưa ra các công nghệ tương tác trực tiếp với người dùng. Trí tuệ nhân tạo đang ngày càng được nâng cấp hoàn thiện giúp người dùng dễ dàng tương tác, dễ sử dụng và giảm quá trình thực hiện. Trí tuệ nhân tạo đang là một lĩnh vực mới mẻ và được sự quan tâm rất lớn từ các hãng công nghệ hàng đầu. Với công nghệ đang ngày càng được áp dụng trong đời sống giúp con người làm việc hiệu quả hơn tiết kiệm thời gian và sức lực, trí tuệ nhân tạo như một hệ thống được xây dựng để phục vụ cho điều đó.

Hệ thống trợ lý ảo là một hệ thống giúp con người giao tiếp với máy thực hiện các yêu cầu ý muốn của người dùng. Hiện nay trên thế giới hệ thống trợ lý ảo đang được nghiên cứu và phát triển mạnh mẽ ở các hãng công nghệ hàng đầu thế giới. Với mong muốn hiểu sâu về trí tuệ nhân tạo, em quyết định chọn đề tài “Nghiên cứu hệ thống trợ lý thông minh ảo” làm đề tài luận văn thạc sĩ. Qua đề tài em muốn nâng cao sự hiểu biết về trí tuệ nhân tạo đồng thời nghiên cứu để có thể áp dụng thực tiễn tại Việt Nam.

Nội dung luận văn được chia ra làm 3 phần như sau:

Chương 1: Giới thiệu tổng quan về hệ thống trợ lý ảo, cấu trúc hệ thống trợ lý ảo, trình bày về xử lý ngôn ngữ tự nhiên NLP và ứng dụng NLP trong chatbot

Chương 2: Nghiên cứu một số kỹ thuật được sử dụng trong chatbot, tìm hiểu quản lý hội thoại, mô hình sinh hội thoại.

Chương 3: Trình bày về quá trình thực nghiệm và đánh giá, các kết quả được thực nghiệm và xây dựng chatbot.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG TRỢ LÝ ẢO

1.1. Giới thiệu về hệ thống trợ lý ảo

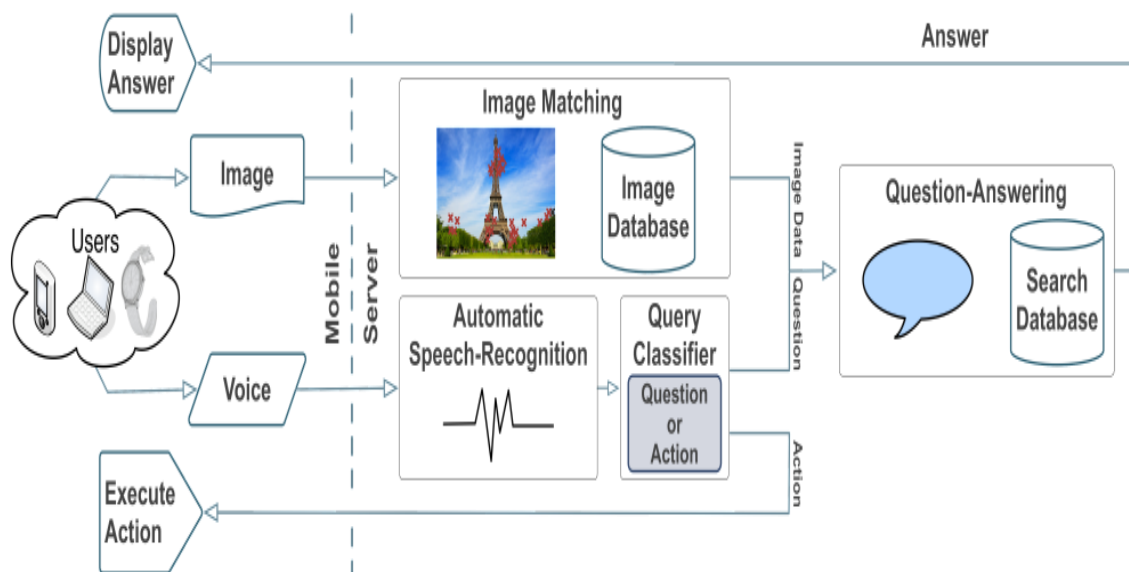
Siri của Apple [1], Google Now của Google [2] và Cortana của Microsoft [3] đại diện cho một lớp dịch vụ web mới nổi gọi là các Ứng dụng Hỗ trợ Cá nhân Thông minh (Intelligent Personal Assistants - IPA). IPA là một ứng dụng sử dụng các yếu tố đầu vào như tiếng nói, hình ảnh và thông tin theo ngữ cảnh của người dùng để hỗ trợ bằng cách trả lời các câu hỏi bằng ngôn ngữ tự nhiên, đưa ra khuyến nghị và hành động. Các IPA đang nổi lên là một trong những dịch vụ Internet phát triển nhanh nhất vì gần đây họ đã triển khai trên các nền tảng nổi tiếng như iOS, Android và Windows Phone, làm cho chúng phổ biến trên các thiết bị di động trên toàn thế giới.

Sử dụng IPAs đang gia tăng nhanh chóng với các sản phẩm mới nhất trong các công nghệ có thể đeo như đồng hồ thông minh và kính thông minh. Sự tăng trưởng thị phần này, cùng với việc thiết kế các sản phẩm đeo có thể phụ thuộc nhiều vào đầu vào hình ảnh và giọng nói, cho thấy nhu cầu sử dụng dịch vụ IPA đang tăng lên nhanh chóng. IPA khác với nhiều khối lượng công việc dịch vụ web hiện có trong các máy tính có quy mô lớn (WSCs). Ngược lại với các truy vấn của các dịch vụ trung tâm duyệt web, luồng truy vấn IPA thông qua các thành phần phần mềm thúc đẩy những tiến bộ gần đây trong nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên và tầm nhìn máy tính để cung cấp cho người dùng một câu hỏi dựa trên câu hỏi và hệ thống trả lời. Do cường độ tính toán của các thành phần này và các mô hình dữ liệu dựa trên mô hình mà họ sử dụng, các nhà cung cấp dịch vụ yêu cầu tính toán trong các nền tảng trung tâm dữ liệu lớn thay cho việc tính toán trên các thiết bị di động. Cách tiếp cận giảm tải này được sử dụng bởi cả Siri của Apple lẫn Google Now của Google khi họ gửi các bản ghi nén lệnh truy vấn bằng giọng nói tới các trung tâm dữ liệu để nhận dạng tiếng nói và khai thác ngữ nghĩa.

Sự tương tác giữa thiết bị thông minh với con người không còn nằm ở việc gõ văn bản mà giờ là cả hình ảnh lẫn giọng nói, để đáp ứng được nhu cầu ngày càng cao của trải nghiệm người dùng, các hãng công nghệ thông tin lớn đang ngày càng đầu tư mạnh vào hệ thống trợ lý thông minh ảo.

1.2. Cấu trúc hệ thống trợ lý ảo

Để hiểu một hệ thống trợ lý ảo hoạt động như thế nào, cần phải hiểu được cấu trúc và cấu tạo của một hệ thống trợ lý ảo. Sơ đồ của một hệ thống trợ lý ảo:



Hình 1.1: Sơ đồ hệ thống trợ lý ảo

Dựa trên hình 1.1 [7], có thể thấy cấu tạo của một hệ thống trợ lý ảo bao gồm 2 khối: khối người dùng (user) có đầu vào là ảnh, giọng nói, văn bản và khối xử lý dữ liệu (server) gồm các khối xử lý hình ảnh, âm thanh, văn bản, khối quản lý hội thoại.

Truy vấn bắt đầu bằng giọng nói, văn bản, hình ảnh của người dùng thiết bị thông minh như điện thoại, thiết bị đeo, kính thông minh. Các file nén hình ảnh, âm thanh hoặc văn bản được gửi tới máy chủ để xử lý.

Với giọng nói được xử lý bằng giao diện ASR (Automatic Speech Recognition) chuyển câu hỏi nói của người dùng sang văn bản tương đương bằng mô hình thống kê. Sau đó văn bản đi qua trình phân loại truy vấn (Query Classifier - QC) quyết định xem bài phát biểu có phải là hành động hay câu hỏi không. Nếu đó là hành động, lệnh sẽ được gửi lại cho thiết bị di động để thực hiện. Nếu không, thì hệ thống sẽ hiểu là câu hỏi bằng văn bản thuần túy. Sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên NLP (Natural Language Processing), dịch vụ câu hỏi trả lời QA (Question-Answering) sẽ trích xuất thông tin từ đầu vào, tìm kiếm cơ sở dữ liệu của nó và chọn câu trả lời tốt nhất để trả lại cho người dùng.

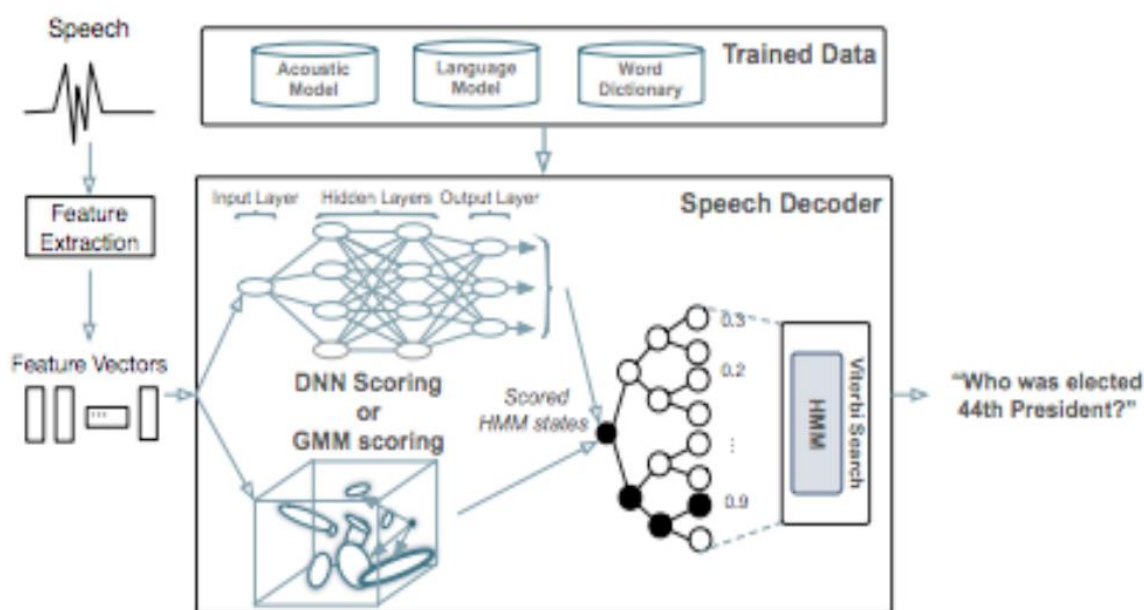
Nếu là một hình ảnh đi kèm với đầu vào giọng nói, hệ thống sẽ sử dụng kỹ thuật thị giác máy tính để cố gắng phù hợp với hình ảnh đầu vào cơ sở dữ liệu hình ảnh của mình và trả về thông tin có liên quan với hình ảnh đó.

1.3. Nhận dạng giọng nói tự động (ASR - Automatic Speech Recognition)

Khối nhận dạng giọng nói tự động là khối đầu vào giúp người dùng tương tác với trợ lý ảo bằng giọng nói. Các đầu vào cho ASR [9] là các vector đặc trưng đại diện cho đoạn nói, được tạo ra bởi quá trình tiền xử lý nhanh và trích xuất đặc tính của bài phát

biểu. Thành phần của ASR dựa vào sự kết hợp của mô hình Hidden Markov (HMM) và một mô hình hỗn hợp Gaussian (GMM) hoặc một mạng nơ-ron sâu (DNN).

HMM xây dựng một cây các trạng thái cho khung lời nói hiện tại sử dụng vector tính năng đầu vào. GMM hoặc DNN đánh giá xác suất của sự chuyển đổi trạng thái trong cây, và thuật toán Viterbi sau đó tìm kiếm con đường có nhiều khả năng nhất dựa trên các điểm này. Đường dẫn có xác suất cao nhất thể hiện kết xuất văn bản cuối cùng. Điểm GMM đánh giá HMM trạng thái chuyển tiếp bằng cách ánh xạ một vector đối tượng đầu vào vào một hệ tọa độ đa chiều và lặp lại điểm số các đặc tính chống lại mô hình âm thanh được đào tạo. DNN điểm số sử dụng xác suất từ một mạng thần kinh. Độ sâu của DNN được xác định bởi số lớp ẩn mà số điểm cho một chuyển tiếp đi qua mạng. Trong luận văn sẽ nghiên cứu chủ yếu về DNN vì nó có độ chính xác cao hơn.

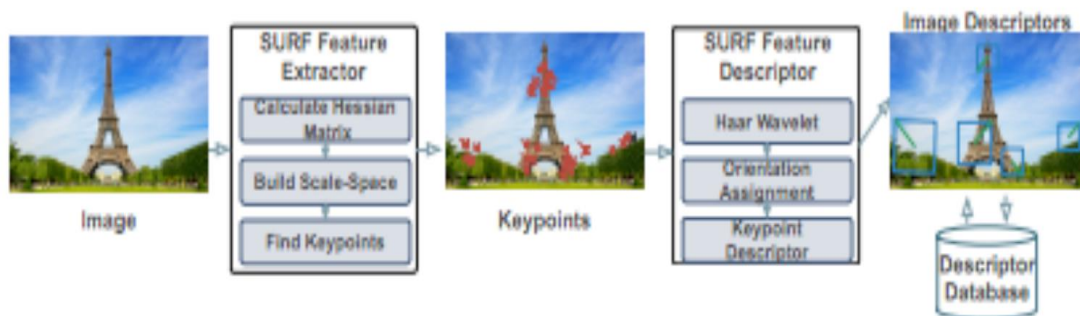


Hình 1.2: Sơ đồ xử lý âm thanh

1.4. Xử lý hình ảnh IMM (Image Matching)

Khối xử lý hình ảnh IMM là khối giúp người dùng tương tác với hệ thống trợ lý ảo thông qua hình ảnh. Khi người dùng muốn hỏi trợ lý ảo về một hình nào đó chẳng hạn bông hoa hồng thì hệ thống trợ lý ảo sẽ xử lý dữ liệu hình ảnh đầu vào và cung cấp đầu ra là thông tin về hình ảnh đó. Từ dữ liệu hình ảnh đưa vào hệ thống sẽ cố gắng tìm kiếm hình ảnh trong cơ sở dữ liệu trước khi chế biến và trả về thông tin về kết hợp hình ảnh. Cơ sở dữ liệu được sử dụng trong hệ thống là cơ sở dữ liệu Mobile Visual Search. Các điểm chính của hình ảnh được rút ra lần đầu tiên từ hình ảnh đầu vào sử dụng thuật toán SURF [7]. Trong tính năng chiết xuất (FE), hình ảnh được lấy mẫu và xáo trộn nhiều lần để tìm các điểm thú vị ở các quy mô khác nhau. Khi ngưỡng đáp ứng lưu trữ các điểm chính của hình ảnh, các điểm then chốt được chuyển tới bộ phận mô tả tính năng (FD), nơi chúng được chỉ định một vector định hướng và các điểm chính được định hướng tương tự được nhóm thành các bộ mô tả tính năng. Quá trình này làm giảm sự biến đổi

giữa các hình ảnh đầu vào, tăng cơ hội tìm kiếm sự kết hợp chính xác. Bộ mô tả từ hình ảnh đầu vào được đối sánh với bộ mô tả nhóm trước đại diện cho các hình ảnh cơ sở dữ liệu bằng cách sử dụng tìm kiếm gần nhất (ANN), hình ảnh cơ sở dữ liệu với số lượng lớn nhất của cơ sở được trả lại.

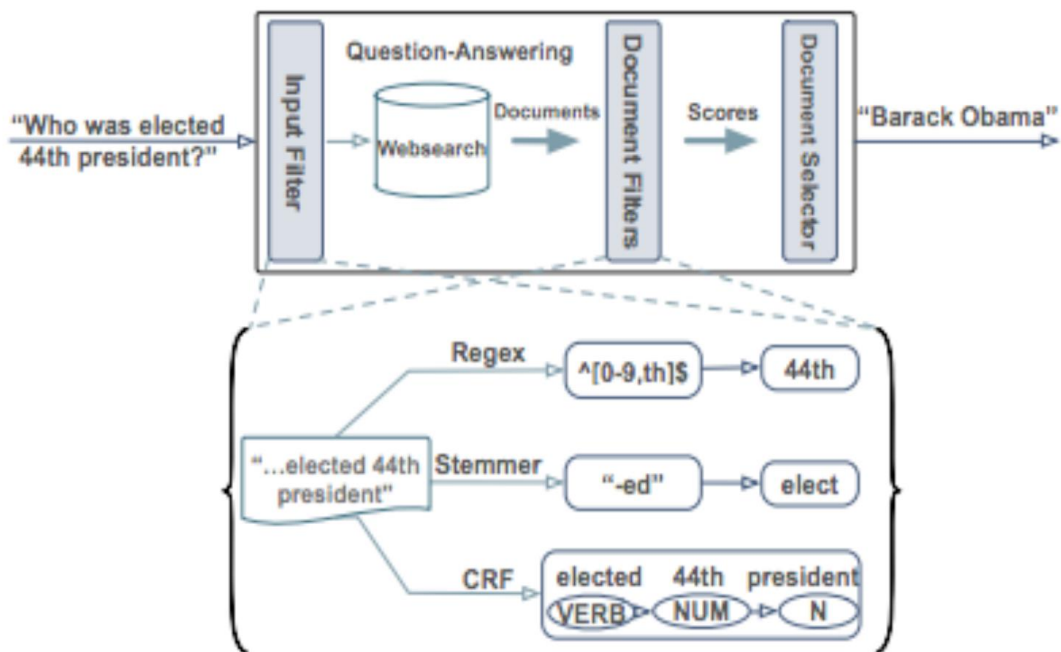


Hình 1.3: Sơ đồ xử lý hình ảnh

1.5. Quản lý câu hỏi trả lời QA (Question-Answering)

1.5.1. Cấu tạo của quản lý câu hỏi

Khối quản lý câu hỏi trả lời là khối xử lý dữ liệu tương tác ở trong hệ thống trợ lý ảo. khi người dùng hỏi hệ thống trợ lý ảo thì nó sẽ lấy dữ liệu từ khối QA để đưa ra câu trả lời phù hợp. Văn bản được xuất ra từ ASR hoặc nhập trực tiếp được chuyển đến OpenEphyra (OE) sử dụng ba quy trình cốt lõi để trích xuất thông tin văn bản, bắt nguồn từ, kết hợp cụm từ thông dụng, và gắn thẻ theo từng thành phần của văn bản [7]. Hình dưới mô tả sơ đồ OE kết hợp thành phần này, tạo ra các truy vấn tìm kiếm web và lọc các kết quả trả về. Thuật toán Porter (stemmer) phơi bày gốc của một từ bằng cách kết hợp và cắt ngắn các từ thông dụng. OE cũng sử dụng một bộ mẫu biểu thức chính quy để đối sánh các từ truy vấn chung (như cái gì, ở đâu ...) và lọc bất kỳ ký tự đặc biệt nào trong đầu vào. Trình phân loại ngẫu nhiên (CRF) lấy một câu, vị trí của mỗi từ trong câu và nhãn các từ hiện tại và trước đó làm đầu vào để đưa ra các dự đoán về phần nói của từng từ của một truy vấn đầu vào. Mỗi truy vấn đầu vào được phân tích cú pháp bằng cách sử dụng các thành phần nói trên để tạo truy vấn cho công cụ tìm kiếm. Tiếp theo, các bộ lọc sử dụng các kỹ thuật tương tự được sử dụng để lấy thông tin từ các tài liệu trả về; tài liệu với điểm tổng thể cao nhất sau khi tổng hợp số điểm được trả về là câu trả lời hay nhất.



Hình 1.4: Quản lý câu hỏi trả lời

1.5.2. Xử lý ngôn ngữ tự nhiên trong Quản lý câu hỏi trả lời

Theo Wikipedia, NLP (Natural Language Processing) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ - công cụ hoàn hảo nhất của tư duy và giao tiếp.

Theo cách hiểu thì có thể hiểu xử lý ngôn ngữ tự nhiên [6] là một phạm vi lý thuyết các kỹ thuật tính toán để phân tích và mô tả các văn bản xảy ra tự nhiên ở một hoặc nhiều mức độ phân tích ngôn ngữ theo yêu cầu của con người mong muốn.

Mục tiêu của NLP là nhằm thể hiện ý nghĩa thực sự và ý định của người dùng khi thao tác dữ liệu. Điển hình ứng dụng NLP:

- Giải thích văn bản đầu vào
- Dịch văn bản sang một ngôn ngữ khác
- Trả lời các câu hỏi về nội dung của một văn bản
- Thu thập các suy luận từ văn bản

1.5.3. Các mức của NLP

- Ngữ âm học: mức này liên quan tới việc giải thích các âm thanh nói trong và giữa các từ. Có ba loại quy tắc được sử dụng trong phân tích âm vị học: quy tắc âm thanh trong từ, quy tắc ngữ âm trong biến thể phát âm khi từ được nói với nhau, quy tắc biến động trong ngữ điệu của một câu. Một hệ thống NLP hỗ trợ đầu vào nói, song âm là phân tích và mã hóa tính hiệu thành tín hiệu số hóa để giải thích

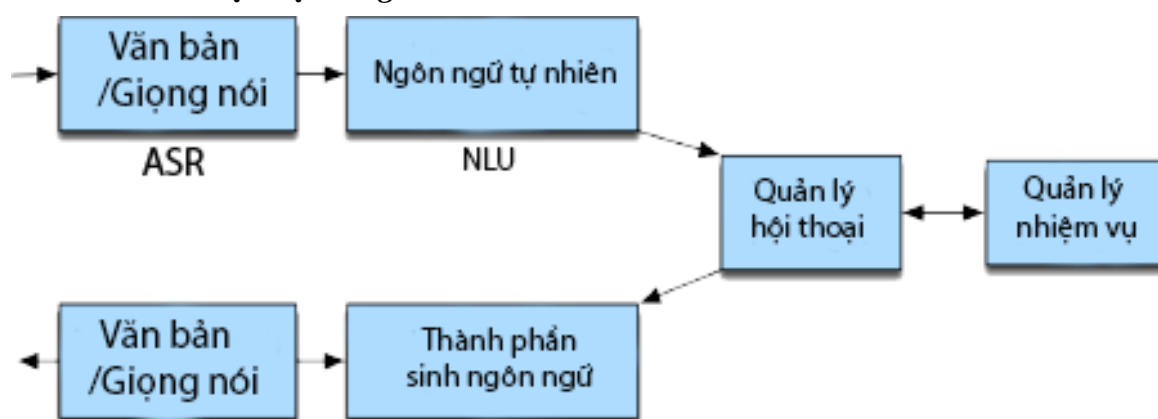
các quy tắc khác nhau hoặc bằng việc so sánh với mô hình ngôn ngữ cụ thể được sử dụng.

- Hình thái học: mức độ này liên quan tới bản chất cấu thành của các từ bao gồm các đơn vị nhỏ nhất của ý nghĩa. Ví dụ từ preregistration có thể được phân tích thành trước tiền tố, gốc “registra” và hậu tố. Vì ý nghĩa của mỗi hình thái vẫn giữ nguyên qua các từ ngữ, con người có thể phân chia một từ không rõ thành các hình thái cấu thành để hiểu ý nghĩa của nó. Tương tự trong NLP có thể nhận ra ý nghĩa được truyền đạt bởi mỗi hình thái có thể đạt được và đại diện cho ý nghĩa. Ví dụ thêm hậu tố vào một động từ cho biết hành động của động từ diễn ra trong quá khứ.
- Từ vựng học: con người hay hệ thống NLP diễn giải ý nghĩa của từng từ.
- Thuật ngữ: tập trung vào việc phân tích các từ trong một câu để khám phá ra ngữ pháp cấu trúc của câu. Điều này đòi hỏi cả ngữ pháp và trình độ phân tích cú pháp. Kết quả của việc này là đại diện của một câu. Các mối quan hệ phụ thuộc cấu trúc giữa các từ. có nhiều ngữ pháp khác nhau có thể được sử dụng và do đó sẽ ảnh hưởng đến sự lựa chọn của một trình phân tích cú pháp. Ví dụ có câu “con chó đuổi con mèo” và “con mèo đuổi theo con chó” khác nhau về ý nghĩa.
- Ngữ nghĩa: mức độ mà hầu hết mọi người nghĩ rằng ý nghĩa được xác định tuy nhiên chúng ta có thể xem trong xác định ở trên của các cấp, đó là tất cả các cấp có ý nghĩa góp phần vào. Xử lý ngữ nghĩa xác định ý nghĩa của một câu bằng cách tập trung vào tương tác giữa các ý nghĩa cấp từ trong câu. Mức độ này bao gồm việc định hướng ngữ nghĩa của các từ với nhiều giác quan, theo cách tương tự để cách phân định cú pháp của các từ có thể hoạt động như nhiều phần câu bài phát biểu là hoàn thành các cấp cú pháp. Ví dụ trong các nghĩa khác “file” là một danh từ có thể có nghĩa là một thư mục hoặc một công cụ để tạo.
- Đàm luận: mặc dù cú pháp và ngữ nghĩa làm việc với các đơn vị câu, mức độ diễn đạt của NLP làm việc với các đơn vị văn bản dài hơn một câu. Nghĩa là nó không giải thích văn bản như các câu ghép nối, mỗi câu có thể được giải thích đơn lẻ. Thay vào đó bài diễn thuyết tập trung vào tính chất của văn bản tập trung kết nối giữa các thành phần câu.
- Thực dụng: liên quan đến việc sử dụng có mục đích ngôn ngữ trong các tình huống và sử dụng bối cảnh trên các nội dung của văn bản để hiểu mục đích là để giải thích làm thế nào thêm ý nghĩa được đọc vào văn bản. Điều này đòi hỏi nhiều kiến thức bao gồm sự hiểu về ý định, kế hoạch và những mục tiêu.

Hệ thống NLP hiện nay có khuynh hướng thực hiện thành các mô đun để đạt được mức độ yêu cầu. Mức độ thấp thì sử dụng mô đun thấp, mức độ cao thì dùng nhiều mô đun thấp kết hợp.

1.6. Ứng dụng NLP trong chatbot

1.6.1. Cấu tạo hệ thống chatbot



Hình 1.5: Mô hình cơ bản của hệ thống chatbot

Chatbot có thể hiểu là các hệ thống có thể thực hiện các cuộc hội thoại mở với mục đích bắt chước tính năng đàm thoại phi cấu trúc hoặc trò chuyện của con người với máy.

Hình 1.5 mô tả về cấu trúc mô hình của một chatbot [9]. Trong mô hình chatbot kiểu truy xuất thông tin (retrieval-based), thành phần ngôn ngữ tự nhiên NLU sử dụng rất nhiều kỹ thuật xử lý ngôn ngữ tự nhiên NLP để phân tích câu hội thoại, xác định ý định của người dùng, trích xuất những thông tin của người dùng cung cấp. Sau đó những thông tin này được chuyển tới cho thành phần quản lý hội thoại (Dialogue manager). Thành phần quản lý hội thoại sẽ xử lý thông tin và chuyển tới cho để sinh ra câu hội thoại.

1.6.2. Cấu trúc hộp thoại chatbot



Hình 1.6: Cấu trúc hộp thoại chatbot

Giao diện chatbot gồm:

- Nội dung dành cho người dùng thao tác nhập liệu: ghi âm giọng nói, nhập văn bản, thêm file như hình ảnh, âm thanh, địa chỉ.
- Nơi hiển thị: giao diện tương tác là nơi hiển thị nội dung người dùng nói và nơi Chatbot trả lời.

1.6.3. Xử lý dữ liệu trong chatbot

1.6.3.1. Tập văn bản (Corpus-based chatbots)

Corpus-based chatbots [8] dựa trên cơ sở thay vì dựa trên các quy tắc được xây dựng bằng tay, các cuộc hội thoại giữa con người với con người, hoặc đôi khi là phản ứng của con người từ cuộc trò chuyện. Serban et al. (2017) đã tổng hợp một số tập văn có sẵn như vậy chẳng hạn như cuộc trò chuyện trên nền tảng trò chuyện facebook, twitter hoặc trong hội thoại phim, có sẵn với số lượng lớn và đã được cho thấy giống với cuộc trò chuyện tự nhiên. Câu trả lời của Chatbot thậm trí có thể được trích ra từ các câu trong tập tin của văn bản không phải từ hội thoại.

Có 2 loại chatbots dựa trên tập lệnh: các hệ thống dựa trên việc thu thập thông tin, và các hệ thống dựa trên việc học máy có giám sát dựa trên sự truyền tải trình tự.

Chatbots hầu hết đều dựa trên cơ sở có xu hướng làm mô hình rất ít của ngữ cảnh đối thoại. Thay thế điều đó có xu hướng tập trung vào việc tạo ra một biến đáp ứng duy nhất phù hợp với lời nói trước của người sử dụng, vì lý do này chúng thường được gọi là phản ứng hệ thống thế hệ và chỉ tập trung vào các phản hồi đơn lẻ trong khi bỏ qua ngữ cảnh hoặc mục tiêu đàm thoại.

1.6.3.2. Thu thập thông tin (IR Chatbot - Information retrieval chatbot)

Nguyên tắc đằng sau việc thu thập thông tin dựa trên chatbot [8] là đáp ứng các yêu cầu của người sử dụng biến x bằng cách lặp lại một số biến y phù hợp từ một tập hợp của văn bản tự nhiên.

Sự khác biệt giữa các hệ thống như vậy nằm ở cách họ chọn tập văn bản và cách họ quyết định xem những gì được coi là sự biến đổi thích hợp của con người để sao chép. Sự lựa chọn chung của tập văn bản là thu thập các cơ sở dữ liệu về các cuộc đối thoại của con người. Cách tiếp cận khác là sử dụng tập tin trong hội thoại phim. Khi Chatbot được đưa vào thực tế các bước mà con người sử dụng để phản hồi lại chatbot có thể được sử dụng như dữ liệu cuộc hội thoại bổ sung cho việc đào tạo. Với câu của người dùng các hệ thống dựa trên IR có thể sử dụng bất kỳ truy xuất thuật toán để lựa chọn một phản ứng thích hợp từ hệ thống.

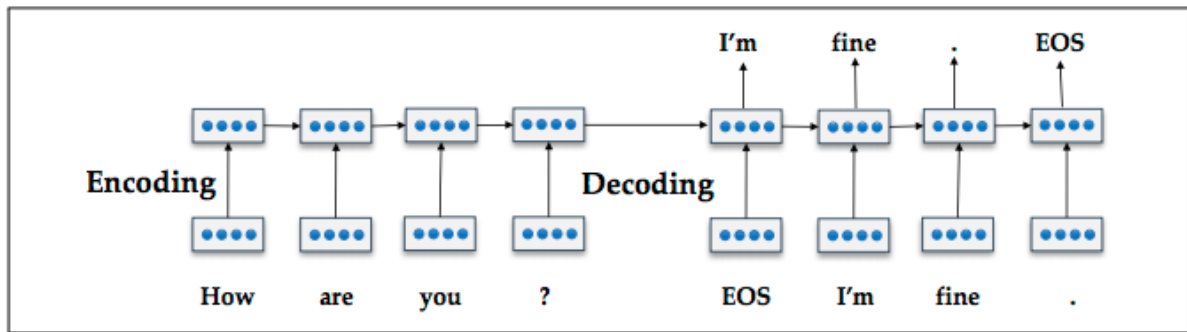
Trong mỗi trường hợp, bất kỳ chức năng tương tự có thể được sử dụng, tính toán phổ biến nhất hoặc qua các từ hoặc trên các phần nhúng.

1.6.3.3. Trình tự liên tiếp trong chatbot (Sequence to sequence chatbots)

Mô hình seq2seq bao gồm hai mạng RNN [10]: một cho bộ mã hoá, và một cho bộ giải mã. Bộ mã hoá nhận một chuỗi (câu) đầu vào và xử lý một phần tử (từ trong câu) tại mỗi bước. Mục tiêu của nó là chuyển đổi một chuỗi các phần tử vào một vector đặc trưng có kích thước cố định mà nó chỉ mã hoá thông tin quan trọng trong chuỗi và bỏ qua

các thông tin không cần thiết. Có thể hình dung luồng dữ liệu trong bộ giải mã hoá dọc theo trục thời gian, giống như dòng chảy thông tin cục bộ từ một phần tử kết thúc của chuỗi sang chuỗi khác.

Mỗi trạng thái ảnh hưởng đến trạng thái ảnh tiếp theo và trạng thái ảnh cuối cùng được xem như tích lũy tóm tắt về chuỗi. Trạng thái này được gọi là bối cảnh hay vector suy diễn, vì nó đại diện cho ý định của chuỗi. Từ bối cảnh đó, các bộ giải mã tạo ra một chuỗi, một phần tử tại một thời điểm. Ở đây, tại mỗi bước các bộ giải mã bị ảnh hưởng bởi bối cảnh và các phần tử được sinh ra trước đó.



Hình 2.7: Kiến trúc seq2seq

Quá trình tìm hiểu về cấu trúc của một mô hình hệ thống trợ lý ảo, có thể thấy để xây dựng một hệ thống trợ lý ảo cần rất nhiều các mô đun khác nhau ghép lại, từ đầu vào cần xử lý dữ liệu, sau đó truyền vào hệ thống xử lý dữ liệu và cuối cùng là trả lại dữ liệu cho người dùng. Hệ thống trợ lý ảo thông minh là nhờ khâu xử lý dữ liệu cực kỳ quan trọng và sẽ được tìm hiểu sâu hơn ở chương tiếp theo.

CHƯƠNG 2: MỘT SỐ KỸ THUẬT SỬ DỤNG TRONG TRỢ LÝ ẢO

2.1. Mạng học sâu DNNs

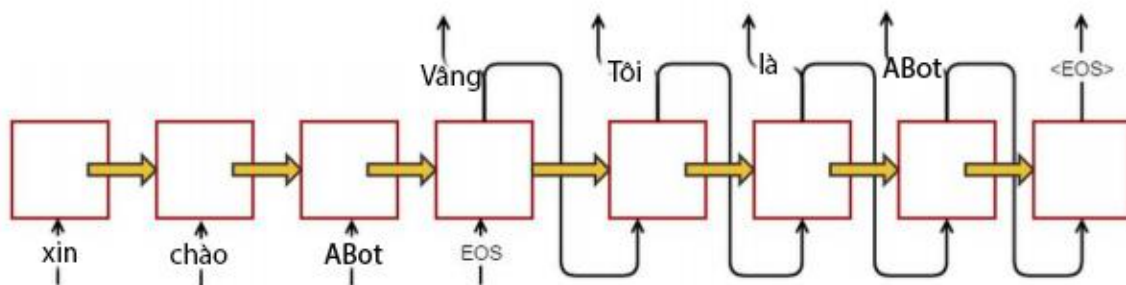
2.1.1. Khái niệm mạng học sâu

Mạng học sâu Deep Neural Networks (DNNs) [10] là mô hình học máy giúp nhận dạng giọng nói và nhận diện đối tượng thị giác rất tốt. DNNs có thể thực hiện tính toán song song tùy ý cho một số bước. Ví dụ như khả năng sắp xếp N số N-bit chỉ sử dụng 2 lớp ẩn của kích thước bậc hai. Vì vậy trong khi mạng nơ ron liên quan đến các mô hình thống kê thông thường, DNN cho phép một tính toán phức tạp. Các DNN có thể được đào tạo với sự đẩy lùi ngược lại sự giám sát bất cứ khi nào bộ tập huấn có dán nhãn đầy đủ thông tin để xác định các thông số của mạng. Do đó nếu tồn tại một tham số thiết lập của một DNN lớn sẽ đạt kết quả tốt. Kiểm soát ngược trở lại được giám sát sẽ tìm ra các thông số và giải quyết vấn đề. Tuy nhiên DNN chỉ có thể áp dụng cho các vấn đề đầu vào và mục tiêu có thể được mã hóa một cách hợp lý với các vector có chiều không cố định. Kiến trúc bộ nhớ ngắn hạn LSTM (Long Short-Term Memory) sẽ giải quyết trình tự chung cho các vấn đề trình tự.

LSTM để đọc chuỗi đầu vào, một trong những timestep tại một thời gian, để có được một chiều lớn định lượng lớn vector đại diện, sau đó sử dụng một LSTM để trích xuất chuỗi đầu ra từ vector đó.

LSTM thứ hai là một mô hình mạng lưới thần kinh mạng tái diễn. Khả năng thành công của LSTM tìm hiểu về dữ liệu phụ thuộc thời gian dài làm cho nó một sự lựa chọn tự nhiên cho ứng dụng này do thời gian trễ đáng kể giữa đầu vào và đầu ra tương ứng của chúng. Đã có một số nỗ lực giải quyết trình tự chung để học hỏi chuỗi về vấn đề với mạng nơ ron.

Xem ví dụ: mô hình chúng tôi đọc một đầu vào “Xin chào ABot” và đầu ra là “Tôi Là ABot”. Các trạng thái ẩn là <EOS> có thể xem là vector ngưỡng suy nghĩ.



Hình 2.1: Biểu diễn mô hình mạng

2.1.2. Xây dựng mô hình

2.1.2.1. Mạng nơ ron tái phát RNN

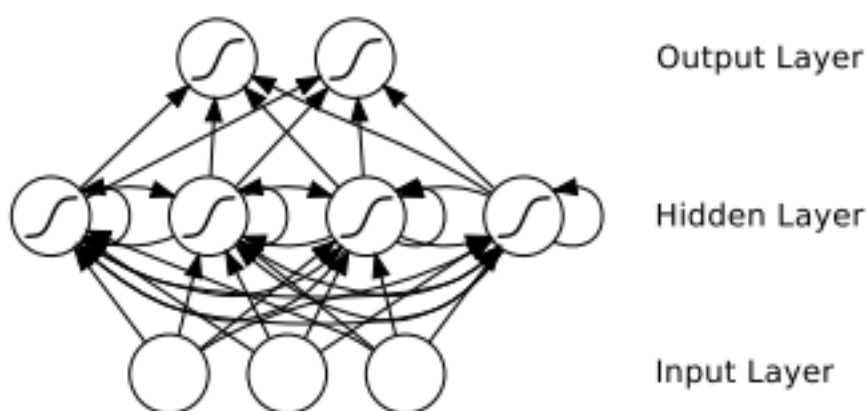
Mạng nơ ron tái phát RNN (Recurrent Neural Network) [10] là một mạng tổng quát các mạng nơ ron cho các chuỗi tuần tự. Là một trong những mô hình học sâu (Deep Learning) được đánh giá nhiều ưu điểm.

RNN lấy ý tưởng thiết kế một mạng lưới sao cho có khả năng xử lý thông tin dạng chuỗi, thực hiện lặp lại các tác vụ cho mỗi thành phần trong chuỗi. Kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán ở những thời điểm trước.

Với mỗi chuỗi đầu vào (x_1, \dots, x_T) , là một mạng RNN chuẩn sẽ tính toán một chuỗi các kết quả đầu ra (y_1, \dots, y_T) , bằng cách duyệt phương trình sau:

$$h_t = \text{sigm}(W^{hx}x_t + W^{hh}h_{t-1}) \quad (2.10)$$

$$y_t = W^{yh}h_t$$



Hình 2.2: Mạng nơ ron tái phát RNN

RNN có thể dễ dàng lập bản đồ trình tự cho các dãy bất cứ khi nào sự liên kết giữa các đầu vào đầu ra được biết đến trước thời hạn. Tuy nhiên không rõ ràng làm thế nào để áp dụng một RNN cho những vấn đề mà đầu vào và các chuỗi đầu ra có độ dài khác nhau với các mối quan hệ phức tạp và không đơn điệu. Một cách đơn giản cho việc học chung chung là lập bản đồ trình tự đầu vào cho một vector có kích thước cố định sử dụng một RNN và sau đó để ánh xạ vector tới dãy đích với một RNN khác. Có thể hoạt động về nguyên tắc kể từ khi được cung cấp với tất cả thông tin liên quan, nhưng rất khó để đào tạo RNN do kết quả lâu dài phụ thuộc vào thời gian. Nên sử dụng LSTM sẽ thành công trong việc khắc phục điều này.

2.1.2.2. Mô hình LSTM

LSTM (Long short-term memory) [10] là một mô hình có cấu trúc tương tự RNN, nhưng có cách tính toán khác đối với các trạng thái ẩn.

Kiến trúc LSTM bao gồm một tập hợp các mạng con kết nối liên tục, được biết đến như khối nhớ. Những khối nhớ này có thể được coi là một phiên bản khác biệt bộ nhớ trong một số máy tính số. Mỗi khối nhớ chứa một hoặc nhiều các tế bào bộ nhớ tự kết nối.

Mục tiêu của LSTM là ước tính xác suất có điều kiện $p(y_1, \dots, y_T | x_1, \dots, x_T)$ trong đó (x_1, \dots, x_T) là một chuỗi đầu vào và (y_1, \dots, y_T) là chuỗi đầu ra tương ứng của nó có chiều

dài T' có thể khác nhau từ T . Mạng LSTM tính xác suất có điều kiện này bằng cách có được chiều dài cố định v của chuỗi đầu vào (x_1, \dots, x_T) được tính bởi các trạng thái ẩn cuối cùng của mạng LSTM, và sau đó tính toán xác suất của $(y_1, \dots, y_{T'})$ với một công thức LSTM-LM tiêu chuẩn mà ban đầu trạng thái ẩn được thiết lập để đại diện v của (x_1, \dots, x_T) :

$$p(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \dots, y_{t-1}) \quad (2.11)$$

Mỗi phân phối xác suất $p(y_1, \dots, y_{T'} | x_1, \dots, x_T)$ được biểu diễn bởi một hàm softmax trên tất cả từ trong từ vựng. Mỗi câu kết thúc với một ký hiệu đặc biệt end-of-sentence "<EOS>", cho phép mô hình để xác định một phân phối các chuỗi của tất cả các độ dài có thể, trong đó LSTM tính xác suất đại diện của "Xin", "chào", "ABot", "<EOS>" và sau đó sử dụng đại diện này để tính xác suất của "Vâng", "Tôi", "là", "ABot", "<EOS>".

Mô hình thực tế khác với mô tả ở trên theo 3 cách quan trọng:

- Đầu tiên sử dụng 2 LSTM khác nhau: một cho trình tự đầu vào và một cho trình tự đầu ra. Do đó làm tăng các tham số mô hình số với chi phí tính toán không đáng kể và làm cho nó tự nhiên. Đào tạo LSTM trên nhiều cặp ngôn ngữ cùng một lúc.
- Thứ 2: deep LSTM vượt trội hơn shallow LSTM vì vậy tôi chọn một LSTM với 4 lớp.
- Thứ 3: nó rất có giá trị để đảo ngược thứ tự của các từ của câu đầu vào. Ví dụ thay vào đó ánh xạ các câu a, b, c đến câu x, y, z LSTM được yêu cầu vẽ bản đồ c, b, a đến x, y, z. Trong đó x, y, z là bản dịch của a, b, c. Bằng cách này a nằm gần x, b khá gần y và như vậy một thực tế giúp dễ dàng thiết lập đầu vào và đầu ra.

2.1.3. Vấn đề và giải pháp khắc phục

Để xử lý được chuỗi dài và kích thước từ vựng thì có một số cách như sau:

PADDING – Tạo độ dài cố định

Trước khi huấn luyện, chúng ta cần chuyển đổi độ dài của các phần tử trong chuỗi thành các chuỗi có độ dài cố định, bằng việc thêm vào các phần tử đệm PADDING. Các phần tử đệm đặc biệt mà chúng ta sẽ sử dụng:

- EOS: Kết thúc câu (End of sentence)
- PAD: Phần đệm bù (Filler)
- GO: Bắt đầu giải mã (Start decoding)
- UNK: Unknown; từ không biết, không có trong từ điển từ vựng

Xem xét một cặp ví dụ HỎI – ĐÁP sau đây:

Q: Bạn tên là gì?

A: Vâng tôi là ABoT

Giả sử chúng ta muốn xử lý các đoạn hội thoại có độ dài 10, kết quả cặp Q/A trên sẽ được chuyển đổi thành như sau:

Q: [PAD, PAD, PAD, PAD, PAD, PAD, “?”, “gì”, “là”, “tên”, “Bạn”]

A: [GO, “Vâng”, “tôi”, “là”, “ABoT”, EOS, PAD, PAD, PAD, PAD]

BUCKETING – Tránh lu mờ thông tin

Bucketing giải quyết vấn đề bằng việc đặt các câu vào các xô buckets có kích thước khác nhau. Ví ta có một danh sách các xô buckets: [(5, 10), (10, 15), (20, 25), (40, 50)]. Nếu độ dài của mẫu hỏi là 4 như ví dụ trên sẽ được đặt vào xô (5, 10). Mẫu hỏi sẽ được đệm với độ dài 5 và đáp án được đệm với độ dài 10. Trong lúc chạy mô hình (huấn luyện hoặc dự đoán), chúng ta sẽ sử dụng một mô hình khác cho mỗi bucket, tương ứng với các độ dài của mẫu hỏi và câu trả lời. Tất cả những mô hình này chia sẻ các tham số giống nhau và do đó hoạt động chính xác theo cùng một cách.

Nếu chúng ta sử dụng xô (5, 10), thì các câu sẽ được mã hóa thành:

Q: [PAD, “?”, “gì”, “là”, “tên”, “Bạn”]

A: [GO, “Vâng”, “tôi”, “là”, “ABoT”, EOS, PAD, PAD, PAD, PAD]

WORD EMBEDDING – mật độ dày đặc

Được ứng dụng cho mô hình hóa ngôn ngữ và các kỹ thuật học trong xử lý ngôn ngữ tự nhiên, nơi các từ hoặc cụm từ từ vựng được ánh xạ vector số thực. Khái niệm nó bao hàm một sự nhúng toán học từ không gian với một chiều cho mỗi từ tới một không gian vector liên tục với kích thước thấp hơn nhiều.

Word Embedding được ứng dụng trong lớp đầu tiên của mạng: Trong đó lớp embedding sẽ ánh xạ một từ (chỉ số index của từ trong từ điển từ vựng) từ từ điển sang một vector dày đặc với kích thước đã cho. Trong mô hình seq2seq, trọng số của lớp embedding được huấn luyện giống nhau với các tham số khác của mô hình.

Trong nghiên cứu của Mikolov và cộng sự, 2013 [29]. Tác giả đề xuất hai kiến trúc để xây dựng word vector đó là Continuous Bag-of-Words model (CBOW), và Continuous Skip-gram model. Trong đó, kiến trúc CBOW: Dự đoán từ hiện tại dựa trên ngữ cảnh của các từ trước đó. Skip-gram: Dự đoán các từ xung quanh khi cho bởi từ hiện tại. Ví dụ:

Giả sử có câu: Tôi muốn mua bia. Tương ứng với mỗi từ trong câu này, chúng ta sẽ khởi tạo một vector random với số chiều được quy định trước (ví dụ số chiều bằng 50). Người ta sử dụng một mạng nơ ron và dụng mạng nơ ron này để điều chỉnh dần dần các vector của các từ sao cho chúng thỏa mãn một số điều kiện nào đó. Để trả lời câu hỏi thì trước tiên chúng ta cần quan tâm tới đặc điểm ngôn ngữ, đó là những từ có mối liên hệ với nhau thường sẽ xuất hiện trong những ngữ cảnh khác nhau. Giả sử thay từ “mua”

bằng từ “cắt” rõ ràng câu đó không có ý nghĩa và không hợp lệ. Từ đó mạng nơ ron sẽ điều chỉnh tham số cho hợp lý để đưa ra đầu ra hợp lý.

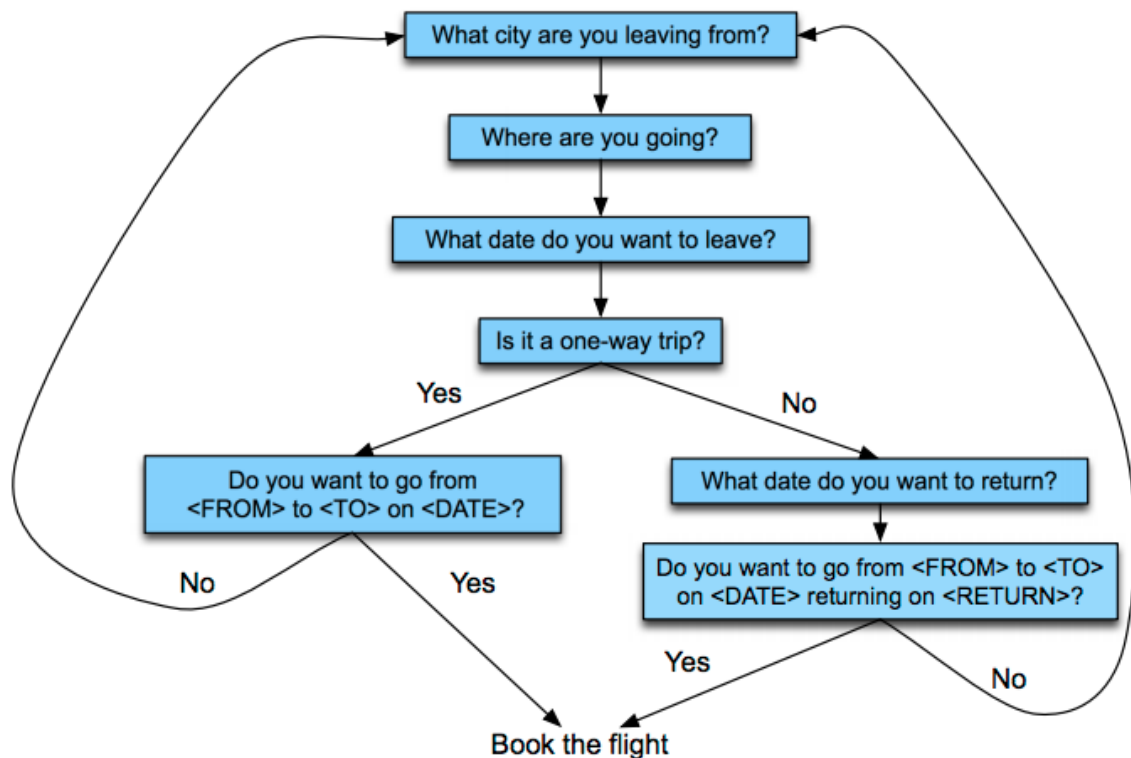
2.2. Quản lý hội thoại

2.2.1. Mô hình quản lý dựa trên khung (Frame based dialog Agents)

2.2.1.1. Kiểm soát cấu trúc hội thoại dựa trên khung

Kiến trúc điều khiển của hệ thống hội thoại dựa trên khung [8] được thiết kế xung quanh khung. Mục đích là để điền vào các khe trong khung với các chất độn mà người sử dụng dự định, và sau đó thực hiện hành động có liên quan cho người dùng (trả lời một câu hỏi hoặc đặt một chuyến bay).

Hầu hết các hệ thống thoại dựa trên khung dựa trên các máy tự động hữu hạn được thiết kế bằng tay cho nhiệm vụ của hội thoại.



Hình 2.3: Kiến trúc tự động trạng thái hữu hạn cho hội thoại dựa trên khung

Hình trên triển khai một hệ thống du lịch mà công việc của họ là yêu cầu người dùng sử dụng thông tin cho 4 tiêu trí: departure city, a destination city, a time, and whether the trip cho một chiều hoặc khứ hồi. Trước hết hãy liên kết với mỗi tiêu trí một câu hỏi để yêu cầu:

Slot	Question
ORIGIN CITY	“From what city are you leaving?”
DESTINATION CITY	“Where are you going?”
DEPARTURE TIME	“When would you like to leave?”
ARRIVAL TIME	“When do you want to arrive?”

Hình 2.4: Ví dụ về mô hình dialogue dựa trên Frame

Hệ thống này kiểm soát hoàn toàn cuộc nói chuyện với người sử dụng. Nó yêu cầu người sử dụng một loại các câu hỏi, bỏ qua (hoặc hiểu sai) bất cứ điều gì không phải là một câu trả lời trực tiếp cho câu hỏi và sau đó tiếp tục câu hỏi tiếp theo.

Giải quyết nhược điểm của FSA khi người dùng đưa cùng lúc nhiều thông tin. Tại mỗi trạng thái FSA chỉ nhận 1 câu trả lời cho 1 câu hỏi.

- Dựa trên các frame định sẵn để định hướng cuộc hội thoại
- Mỗi Frame sẽ bao gồm các thông tin cần điền và các câu hỏi tương ứng
- Dialogue Manager sẽ hỏi cho đến khi các slot được điền hết.
- Trong 1 hội thoại có thể có nhiều hơn 1 frame.

Ví dụ: khách hàng có thể vừa hỏi về vấn đề mạng chậm vừa hỏi muốn đổi gói internet

- Dialogue manager cần biết khi nào thì cần chuyển sang frame khác.

2.2.1.2. Sự hiểu biết ngôn ngữ tự nhiên để điền vào các slot

Mục tiêu của thành phần hiểu biết ngôn ngữ tự nhiên là trích ra ba điều từ lời nói của người sử dụng.

- Nhiệm vụ đầu tiên là phân loại tên miền, ví dụ như nói về các hãng hàng không, lập trình một đồng hồ báo thức, hay giải quyết lịch.
- Thứ 2 là xác định mục đích người dùng hoặc mục tiêu người dùng như cố gắng hoàn thành. Ví dụ như tìm phim, hiển thị chuyến bay
- Thứ 3 trích xuất các slot điền cụ thể mà người dùng dự định hệ thống hiểu từ sự phát biểu của họ về ý định của họ.

Ví dụ: Cho tôi xem các chuyến bay buổi sáng từ Hà nội đến Hồ Chí Minh vào thứ ba
Một hệ thống có thể xây dựng một biểu diễn như sau:

Domain: air-travel

Intent: show flight

Origin City: Hà nội

Origin Date: thứ ba

Origin Time: buổi sang

Dest City: Hồ chí minh

Ví dụ 2: đánh thức tôi vào ngày mai lúc 6h

Domain: alarm clock

Intent: đánh thức

Time: 6h

Date: ngày mai

2.2.2. VoiceXML

Có rất nhiều hệ thống thương mại cho phép các nhà phát triển thực hiện hệ thống hội thoại dựa trên khung, chẳng hạn như Amazon Alexa hay Google Assistant [8]. Các hệ thống công nghiệp như vậy cung cấp các thư viện để xác định các quy tắc để phát hiện ý định người dùng cụ thể và điền vào các slot và để diễn tả kiến trúc để kiểm soát những khung và hành động nào mà hệ thống nên thực hiện. Thay vì tập trung vào một trong

những hệ thống thương mại này, có thể lựa chọn một sim VoiceXML có khả năng tương tự với mỗi người. Ngôn ngữ Voice Extractible Markup Language dựa trên ngôn ngữ thiết kế hộp thoại được sử dụng để tạo các hộp thoại khung. VoidXML đơn giản hơn một hệ thống khung thương mại đầy đủ (nó xác định và do đó chỉ cho phép các mô hình ngữ pháp dựa trên ngữ pháp phi xác suất và các trình phân tích ngữ nghĩa chuẩn), vẫn còn là một cách hữu ích để nắm bắt được khung hình dựa trên khung thiết kế hệ thống. tài liệu VoiceXML chứa một tập hợp các hộp thoại, mỗi một trình đơn hoặc một biểu mẫu. Mẫu là một khung, có slot được gọi là các trường.

```

<noinput>
I'm sorry, I didn't hear you. <reprompt/>
</noinput>

<nomatch>
I'm sorry, I didn't understand that. <reprompt/>
</nomatch>

<form>
  <block>    Welcome to the air travel consultant.  </block>
  <field name="origin">
    <prompt>    Which city do you want to leave from?  </prompt>
    <grammar type="application/x-nuance-gsl">
      [(san francisco) barcelona (new york)]
    </grammar>
    <filled>
      <prompt>    OK, from <value expr="origin"/>  </prompt>
    </filled>
  </field>
  <field name="destination">
    <prompt>    And which city do you want to go to?  </prompt>
    <grammar type="application/x-nuance-gsl">
      [(san francisco) barcelona (new york)]
    </grammar>
    <filled>
      <prompt>    OK, to <value expr="destination"/>  </prompt>
    </filled>
  </field>
  <field name="departdate" type="date">
    <prompt>    And what date do you want to leave?  </prompt>
    <filled>
      <prompt>    OK, on <value expr="departdate"/>  </prompt>
    </filled>
  </field>
  <block>
    <prompt> OK, I have you are departing from <value expr="origin"/>
      to <value expr="destination"/> on <value expr="departdate"/>
    </prompt>
    send the info to book a flight...
  </block>
</form>

```

Hình 2.5: Ví dụ về mô hình VoiceXML

Đoạn mở đầu của ví dụ cho thấy xử lý lỗi. Nếu người dùng không trả lời sau khi nhắc tức là sự im lặng vượt quá ngưỡng thời gian chờ, trình thông dịch VoiceXML sẽ chạy lời nhắc <noinput>. Nếu người dung nói điều gì đó mà không khớp với ngữ pháp cho trường đó, trình thông dịch sẽ phát <nomatch> nhắc. VoiceXML cung cấp một lệnh <reprompt/> lặp lại nhắc nhở cho bất cứ lĩnh vực nào gây ra lỗi.

Thẻ <fileed> cho một trường được thực hiện bởi trình thông dịch ngay khi trường đã được điền bởi người dung. Ở đây tính năng này được sử dụng để xác nhận đầu vào của người dung.

2.3. Mô hình sinh hội thoại

2.3.1. Ý nghĩa chính của mô hình sinh hội thoại

- Tận dụng các hội thoại để có thể tự động sinh hội thoại. Chatbot học theo cách trả lời của con người trong các đoạn hội thoại trong quá khứ

Ví dụ về đoạn hội thoại trong dữ liệu IT Helpdesk [Vinyals and Le 2015] [9]

Describe your problem: i am having issues accessing vpn Machine: could you please let me know what are the operating systems you are using?

Human: i am using linux

Machine: ok it was in the ssl vpn on which you stated it was broken

Machine: please check if you are logged in to corp network Machine: what is the error that you are running please

Human: i am seeing an error related to vpn

Machine: what is the error message that you are getting when connecting to vpn using network connect?

Human: connection refused or something like that

Machine: may i know the version of network connect you connect?

- Chatbot theo mô hình sinh bắt chước các từ ngữ trong câu hội thoại của nhân viên hỗ trợ

Machine: try this

Machine: goto <URL> page and enter your name and password

Human: done

Machine: try logging in with <NAME> and check if its working fine now

Human: yes, now it works!

Machine: great (The <URL> indeed contains information about vpn access.)

2.3.2. Đặc điểm của mô hình hội thoại

- Không cần dữ liệu gán nhãn hoặc cần lượng dữ liệu gán nhãn ít hơn rất nhiều so với mô hình chatbot truyền thống
- Một mô hình duy nhất học trực tiếp từ dữ liệu hội thoại Cần dữ liệu hội thoại lớn
- Mô hình không thực sự phản ánh bản chất của hội thoại giữa người với người
- Chất lượng hội thoại sinh ra phụ thuộc lớn vào dữ liệu hội thoại. Chatbot Tay AI của Microsoft phân biệt chủng tộc do bị người dùng "huấn luyện"
- Các bài toán NLP cơ bản trong cách tiếp cận truyền thống trong phát triển chatbot
 - o Xác định intent
 - o Trích xuất thông tin
 - o Quản lý hội thoại
- Cách tiếp cận truyền thống
 - o Là mô hình chatbot phổ biến trong các sản phẩm chatbot thực tế
 - o Theo mô hình truy xuất thông tin

- Cần nhiều dữ liệu huấn luyện và luật chuẩn bị bằng tay
- Thích hợp cho miền ứng dụng đóng
- Neural chatbot: cách tiếp cận mới trong phát triển chatbot. Học từ dữ liệu hội thoại theo mô hình “sequence to sequence” trong mô hình học sâu (Deep learning).

2.4. Nhận xét

Một hệ thống trợ lý ảo thông minh đều nhờ các khối xử lý dữ liệu bên trong. Với mạng học sâu DNN phần nào đã giúp được hệ thống trả lời thông minh hơn. Tuy có những hạn chế nhất định nhưng cũng đã cải thiện dần dần và có thể áp dụng được vào thực tế. Ở chương 3, em sẽ xây dựng một hệ thống trợ lý ảo bao gồm các mô hình:

- Đầu vào cho người dùng: sử dụng khối ASR, khối IMM, khối nhập văn bản
- Khối xử lý dữ liệu: Api.ai (hệ thống sử dụng mạng học sâu DNN và xây dựng các frame hộp thoại)
- Đầu ra cho người dùng sẽ là văn bản trả lời cho người dùng.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

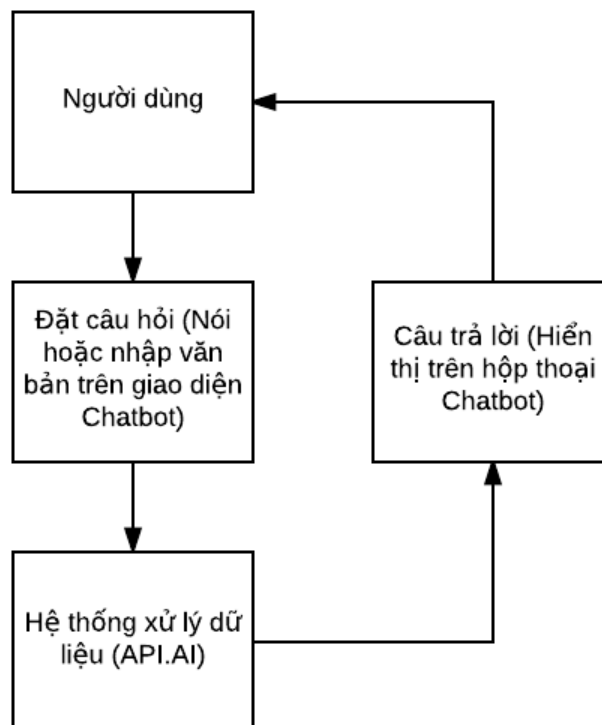
3.1. Phát triển bài toán

Như trình bày ở chương 1 và chương 2 về tìm hiểu hệ thống trợ lý ảo, chương 3 em sẽ trình bày về việc xây dựng một hệ thống chatbot được áp dụng kiến thức tìm hiểu từ chương 1 và chương 2. Bài toán được xây dựng như sau:

Khi người mua hàng hỏi một nội dung: “Tôi muốn mua bia?” thì nếu hỏi người bán hàng sẽ rất dễ trả lời, người bán hàng sẽ trả lời là “Đã xong, bạn muốn mua gì nữa không?”. Nếu không thì có thể thực hiện thanh toán hoặc mua tiếp thì sẽ tiếp tục hỏi lại câu hỏi và kết thúc là thanh toán. Đó là điều rất dễ hình dung về một cuộc hội thoại giữa người với người. Nhưng giả sử nếu không có người bán hàng ở đó, người dùng cũng sẽ không thể mua hay đặt hàng được. Đó là điều hạn chế với rất nhiều cửa hàng và cũng không thể luôn có người trực 24h hàng ngày để thực hiện các câu hỏi và câu trả lời như vậy được. Hoặc có rất nhiều người mua cùng một lúc thì sao?. Người bán hàng cũng không thể trả lời một lúc tất cả được. Chính vì vậy xây dựng một hệ thống trợ lý ảo thông minh sẽ giải quyết được những hạn chế đó.

Chatbot về đặt đồ ăn là một ý tưởng được xây dựng cho các cửa hàng bán đồ ăn đồ uống. Với mục đích hỗ trợ người bán hàng có thể tương tác với nhiều vị khách mà không phải thuê quá nhiều người bán hàng. Người mua chỉ cần lên giao diện chatbot đặt đồ ăn và sau khi đặt đồ ăn xong, thanh toán xong thì chỉ cần đợi đồ ăn vận chuyển tới mà không cần phải tới xếp hàng tại cửa hàng hoặc tới tận cửa hàng để mua.

Tổng quan lại bài toán có thể dựa trên hình 3.1:



Hình 3.1: Sơ đồ phát triển bài toán

3.2. Chương trình thực nghiệm

Chương trình thực nghiệm trong luận văn em chia làm 2 mô hình:

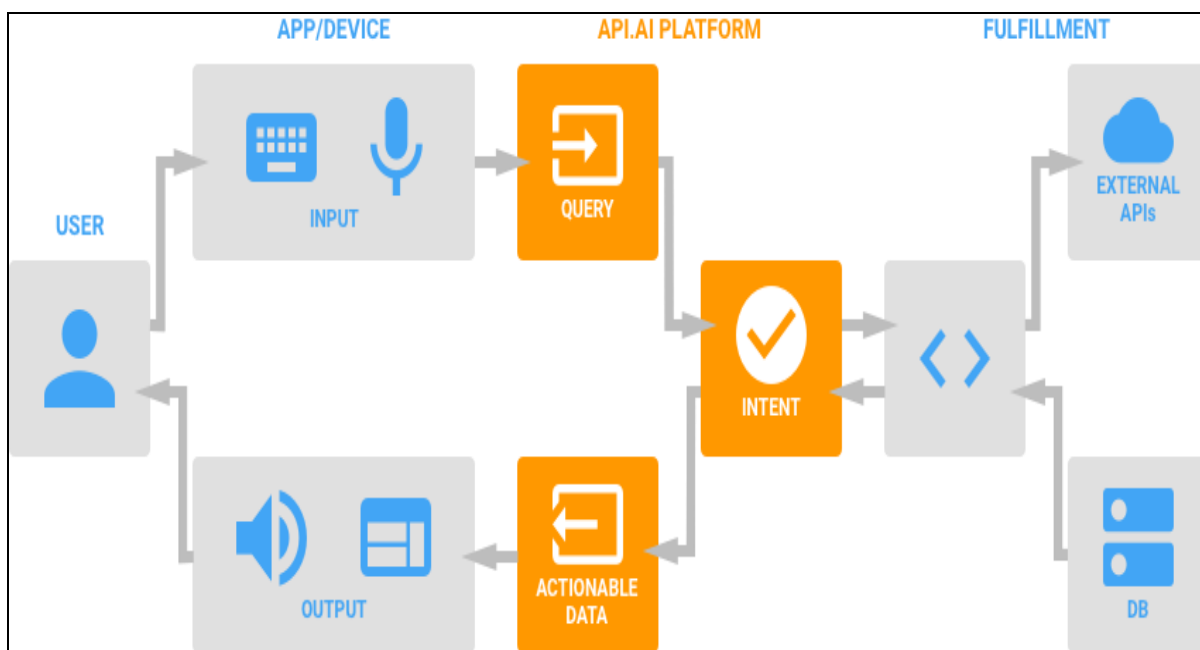
Mô hình người dùng: là giao diện để người dùng có thể tương tác và được xây dựng trên nền tảng iOS của Apple. Đầu vào của mô hình gồm: hình ảnh, âm thanh, văn bản. Đầu ra của mô hình là văn bản.

Mô hình hệ thống trợ lý ảo (hay còn gọi là mô hình chatbot) là nơi sẽ xử lý những tương tác của người dùng và đưa ra câu trả lời phù hợp theo ý người dung. Ở đây mô hình được tích hợp trên API.AI của Google. Api.ai là một hệ thống mở giúp các lập trình viên có thể tự xây dựng một hệ thống trợ lý ảo theo mục đích phát triển. Hệ thống trợ lý ảo em xây dựng trên API.AI là một hệ thống giúp người dùng đặt đồ ăn hỏi những câu hỏi liên quan tới thời tiết. Để hiểu rõ hơn về hệ thống thì trước hết cần hiểu cấu tạo của API.AI từ đó huấn luyện dữ liệu cho chatbot để nó có thể trả lời tự động những câu hỏi của người dùng.

3.2.1. Cấu trúc Api.ai

Là hệ thống phát triển công nghệ tương tác máy tính với con người dựa trên cuộc trò chuyện tự nhiên sử dụng mạng học sâu DNN. Phát triển cho nhiều nền tảng thực hiện nhiệm vụ trả lời câu hỏi của người dùng bằng ngôn ngữ tự nhiên kết hợp ngữ cảnh hội thoại như lịch sử, vị trí, sở thích người dùng.

Hình 3.1 là cấu tạo của mô hình chatbot bao gồm mô hình người dùng gồm đầu vào là văn bản hoặc âm thanh, đầu ra là văn bản hoặc âm thanh và mô hình api.ai gồm các intent, khối xử lý dữ liệu đầu vào đầu ra. Khi người dùng gửi văn bản thì sẽ được gửi lên api.ai. Tại đây văn bản sẽ được qua khối query rồi truyền vào khối intent để xử lý dữ liệu. Trong khối intent gồm rất nhiều frame để đáp ứng các câu hỏi của người dùng. Sau khi lấy được câu trả lời từ Database api.ai sẽ trả về cho người dùng câu trả lời phù hợp.

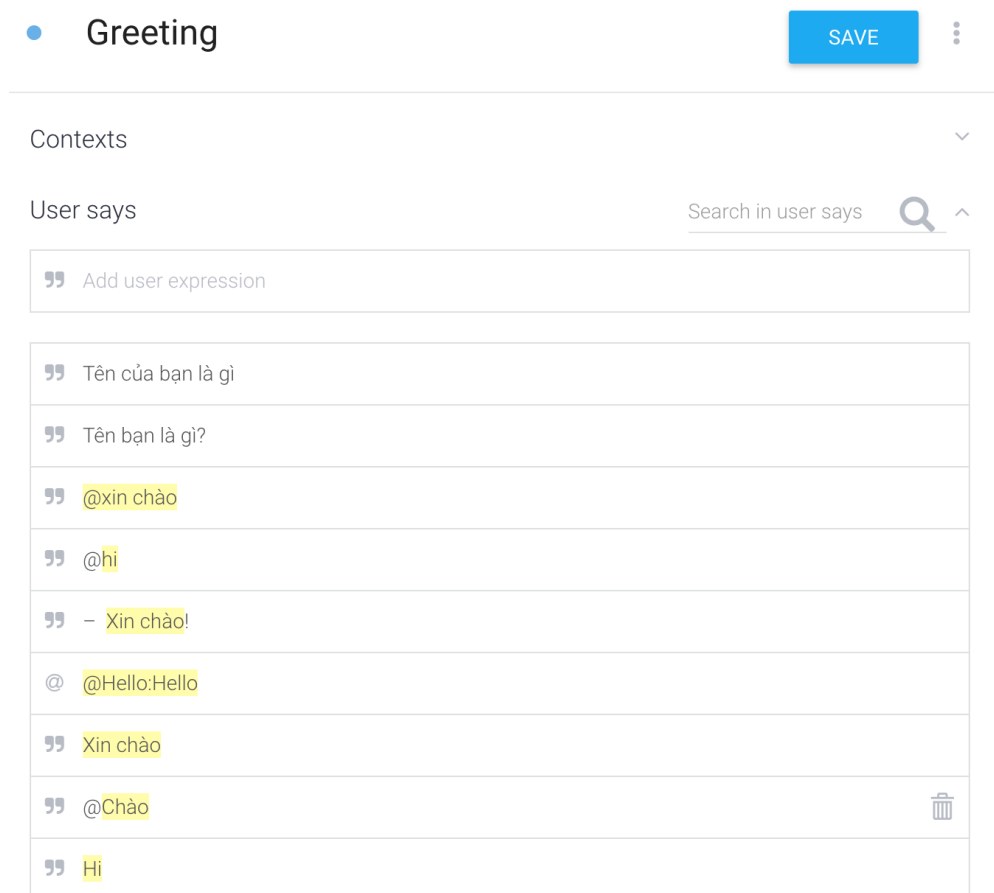


Hình 3.2: Sơ đồ hoạt động của Api.ai

3.2.2. Thành phần API.AI

Để hiểu rõ hơn về API.AI cần hiểu về từng mô đun của hệ thống. Bao gồm:

API.AI Intents: một intents (ý định) đại diện cho việc lập bản đồ giữa những gì người dùng nói và hành động của máy tính. Trong giao diện Intents gồm: user says, action, response, contexts.



Hình 3.3: Giao diện Api.ai intents

API.AI contexts:

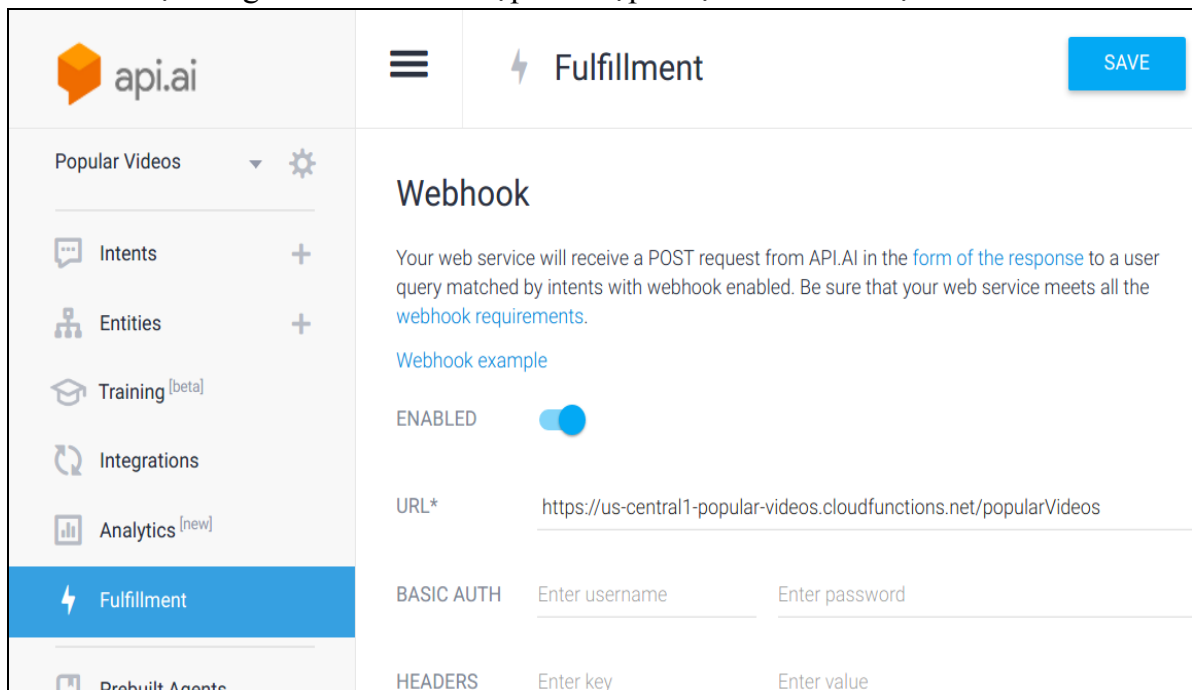
Là khối chứa những câu trả lời để api.ai trả lời cho người dùng những câu phù hợp với câu hỏi. Ví dụ người dùng nói “xin chào” thì chatbot sẽ lấy ngẫu nhiên một câu trả lời phù hợp và trả lời người dùng như “Tôi là Abot. Chào bạn”.



Hình 3.4: Giao diện Api.ai contexts

API.AI webhook:

Mục đích của khối webhook là giúp cho các lập trình viên tích hợp nhiều frame với dữ liệu đã được huấn luyện sẵn mà không cần xây dựng lại từ đầu. Điều này sẽ giúp nhiều cho hệ thống nếu muốn tích hợp các hộp thoại có sẵn dữ liệu.

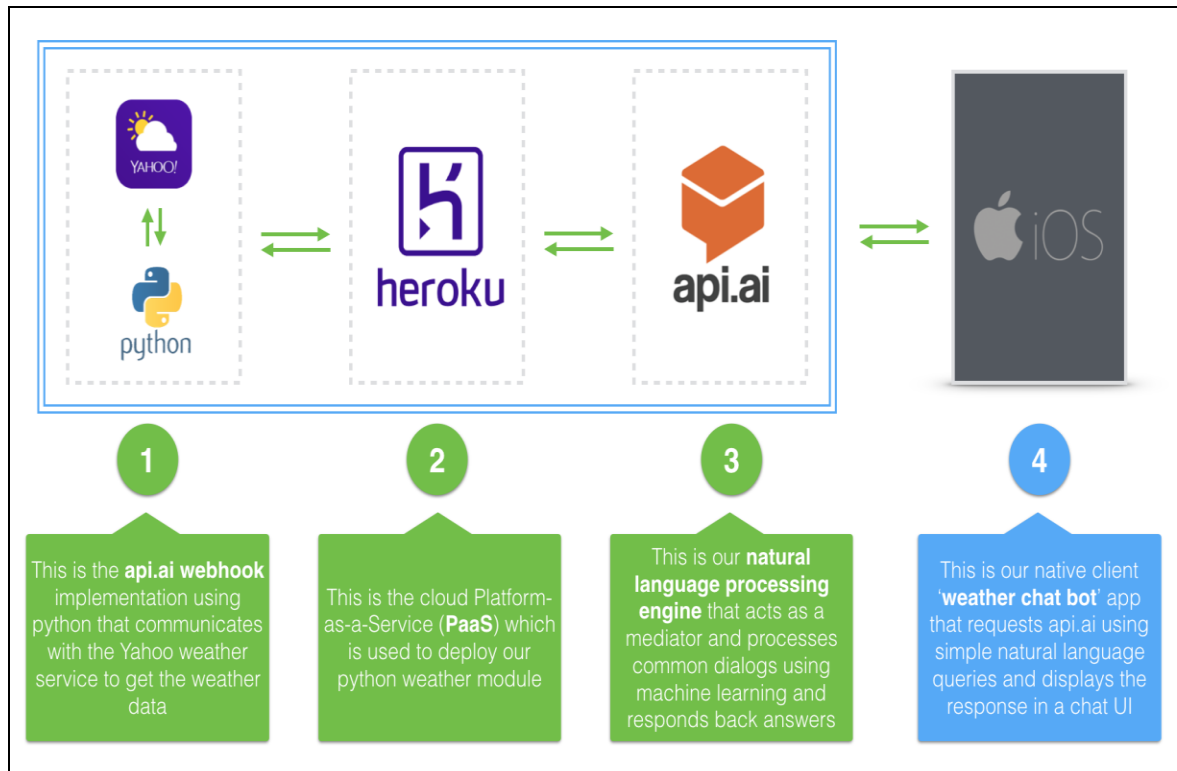


Hình 3.5: Giao diện Api.ai webhook

3.2.3. Mô hình xây dựng

Hệ thống của webhook được xây dựng theo hình 3.5. Khi lập trình viên muốn xây dựng một webhook với nội dung thời tiết thì có thể lấy dữ liệu từ yahoo weather sau đó

sử dụng mã nguồn python để xây dựng một webhook. Khi hoàn thành thì sẽ tải lên service heroku đây là nơi hỗ trợ lưu trữ nhiều webhook. Từ heroku sẽ được huấn luyện cho api.ai để nó hiểu. Từ đó nó sẽ tương tác với người dùng thông qua rất nhiều webhook.



Hình 3.6: Mô hình xây dựng giao diện

3.2.4. *Giao diện ứng dụng*

Mô hình người dùng được xây dựng trên nền tảng iOS của Apple. Giao diện hỗ trợ tương tác trên các thiết bị di động. Để cài đặt phần mềm người dùng có thể sử dụng file ipa do phần mềm Xcode của apple xuất ra. Giao diện gồm 2 khối:

- Khối hiển thị hộp thoại giữa chatbot với người dùng
- Khối người dùng muốn nhập văn bản hoặc giọng nói hoặc hình ảnh.

Hướng dẫn cài đặt:

Cách 1: có thể chạy trực tiếp từ Xcode vào thiết bị

Cách 2: trong đĩa của luận văn có file Abot.ipa để cài trực tiếp. Người dùng cần sử dụng phần mềm hỗ trợ là ifunbox để cài. Có thể tải ifunbox và xem cách sử dụng tại trang web chính của ifunbox (<http://www.i-funbox.com/>).

Cách 3: gửi mail hỗ trợ trực tiếp nguyenhuucuong2405@gmail.com để được hướng dẫn cài đặt.

Hướng dẫn sử dụng:

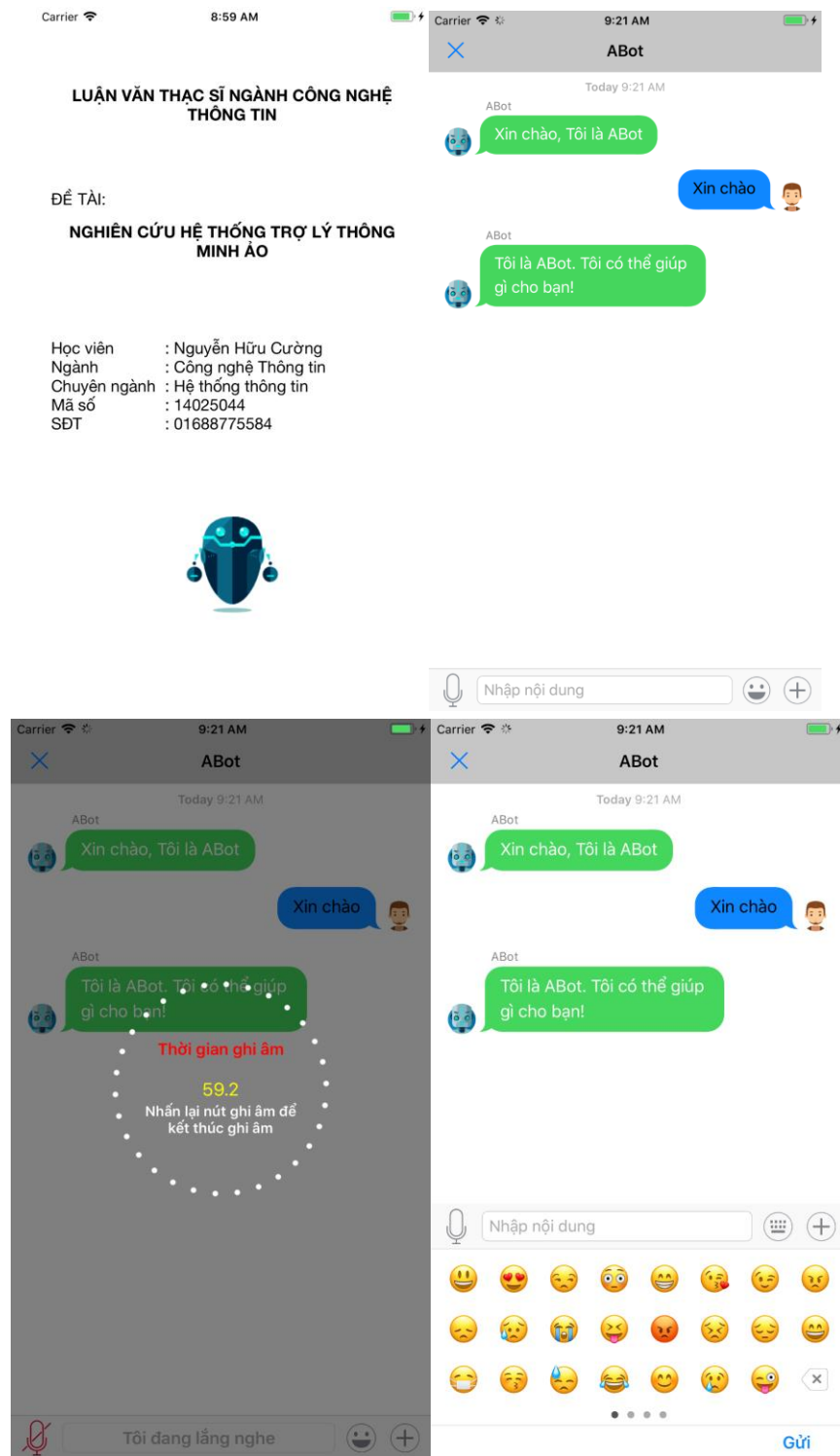
Bước 1: người dùng nhấn vào icon Robot ở giao diện đầu tiên.

Bước 2: Hộp thoại chatbot hiện lên. Có 3 giao thức giúp người dùng nói chuyện với chatbot.

Nếu người dùng gửi văn bản bằng cách nhập keyboard thì chỉ cần nhập nội dung vào ô nhập văn bản.

Nếu muốn chuyển sang trạng thái nói thì có thể nhấn vào nút ghi âm. Khi nói xong người dùng cần nhấn lại nút ghi âm lần nữa để kết thúc quá trình ghi âm. Tại thời điểm này hệ thống sẽ sử dụng công nghệ xử lý âm thanh ASR [9] để có thể chuyển đổi âm thanh thành văn bản, từ đó gửi lên API.AI để chatbot hiểu.

Ngoài ra có thêm tính năng gửi hình ảnh, địa điểm và bật máy ảnh chụp ảnh rồi gửi lên cho api.ai. Tuy nhiên việc xử lý hình ảnh và địa chỉ hiện vẫn đang phát triển nên tính năng này còn hạn chế.

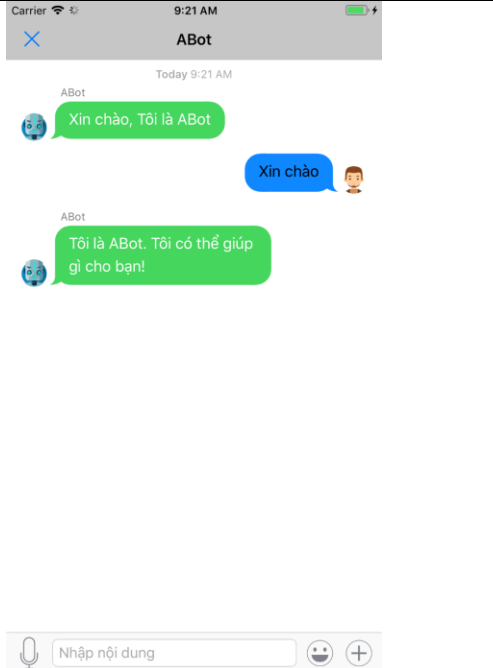


Hình 3.7: Giao diện Chatbot


3.3. Dữ liệu thực nghiệm

Sau khi đã xây dựng xong cả 2 khối người dùng và api.ai, em sẽ tiến hành huấn luyện chatbot. Dưới đây là các huấn luyện ban đầu để chatbot hiểu nội dung người dùng. Sau nhiều lần huấn luyện thì ABot cũng đã bắt đầu hiểu được nội dung của câu hỏi người dùng nói hoặc gửi văn bản.


Dữ liệu huấn luyện lần 1

Nhận xét	Hộp thoại
<ul style="list-style-type: none"> + Dữ liệu sử dụng tiếng việt + ABot không hỗ trợ ngôn ngữ khác. 	 <p>The screenshot shows a chat interface with a header 'Carrier 9:21 AM ABot'. The bot's messages are in green bubbles: 'Xin chào, Tôi là ABot' and 'Tôi là ABot. Tôi có thể giúp gì cho bạn!'. The user's reply is in a blue bubble: 'Xin chào'.</p>

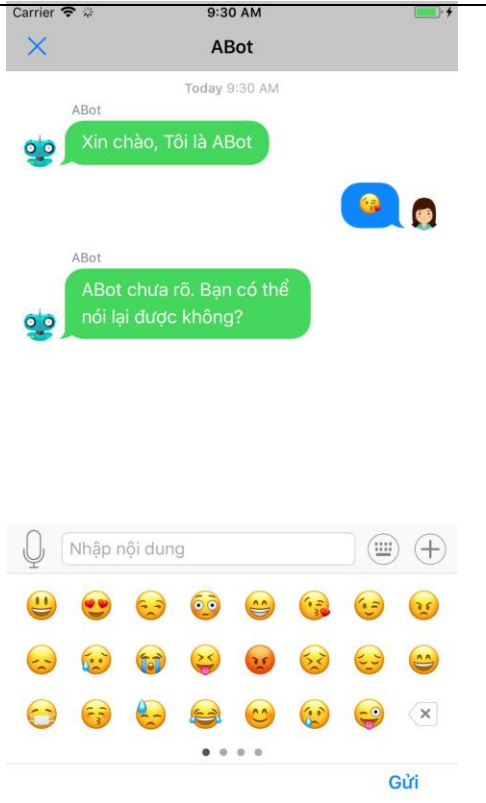
Dữ liệu huấn luyện lần 2

Nhận xét	Hộp thoại
<ul style="list-style-type: none"> + Nếu hỏi theo đúng mẫu câu thì ABot hiểu dữ liệu và trả lời rất chính xác câu hỏi và câu trả lời được dạy. + Dữ liệu huấn luyện được huấn luyện sẵn cho ABot. + Kết quả tạm đạt yêu cầu 	 <p>The screenshot shows a chat interface with a header 'Carrier 9:26 AM ABot'. The bot's messages are in green bubbles: 'Xin chào, Tôi là ABot', 'Tôi là ABot. Tôi có thể giúp gì cho bạn!', and 'ABot chưa rõ. Bạn có thể nói lại được không?'. The user's replies are in blue bubbles: 'Xin chào' and 'Bạn tên gì?'.</p>

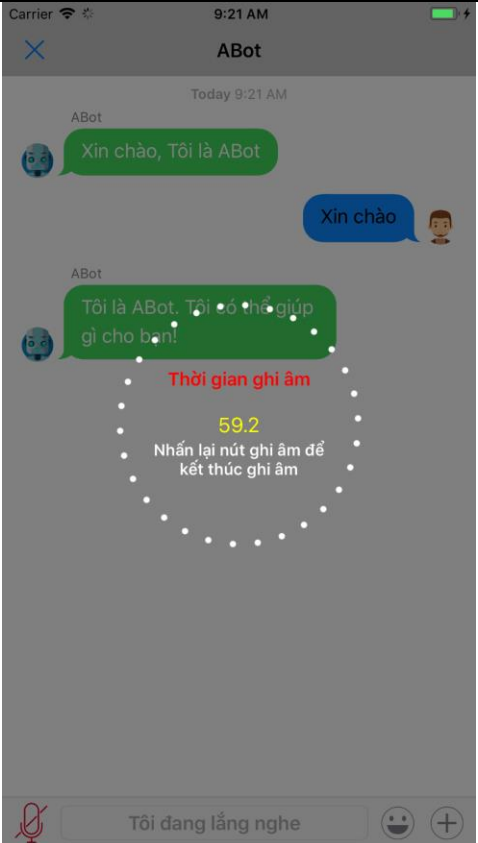
Dữ liệu huấn luyện lần 3

Nhận xét	Hộp thoại
<p>+ Abot được huấn luyện với dữ liệu liên quan tới chủ đề thức ăn.</p> <p>+ Việc hỏi và trả lời câu hỏi đơn giản Abot trả lời gần chính xác câu hỏi người dùng.</p> <p>+ Kết quả tạm đạt yêu cầu</p>	 <p>The screenshot shows a chat interface with a grey header bar containing 'Carrier', signal strength, '9:28 AM', and battery level. The chat title is 'ABot'. The messages are as follows:</p> <ul style="list-style-type: none"> ABot (green bubble): Xin chào, Tôi là ABot User (blue bubble): Tôi muốn mua bia ABot (green bubble): Ok! Bạn có muốn thêm gì không? User (blue bubble): 2 suất bún và 2 cốc bia ABot (green bubble): Bạn muốn gọi thêm gì? User (blue bubble): 3 suất thịt bò xào ABot (green bubble): Ok. Bạn vừa đặt suất thịt. Bạn có muốn thêm gì nữa không? <p>The input field at the bottom contains the text 'Nhập nội dung'.</p>

Dữ liệu huấn luyện lần 4

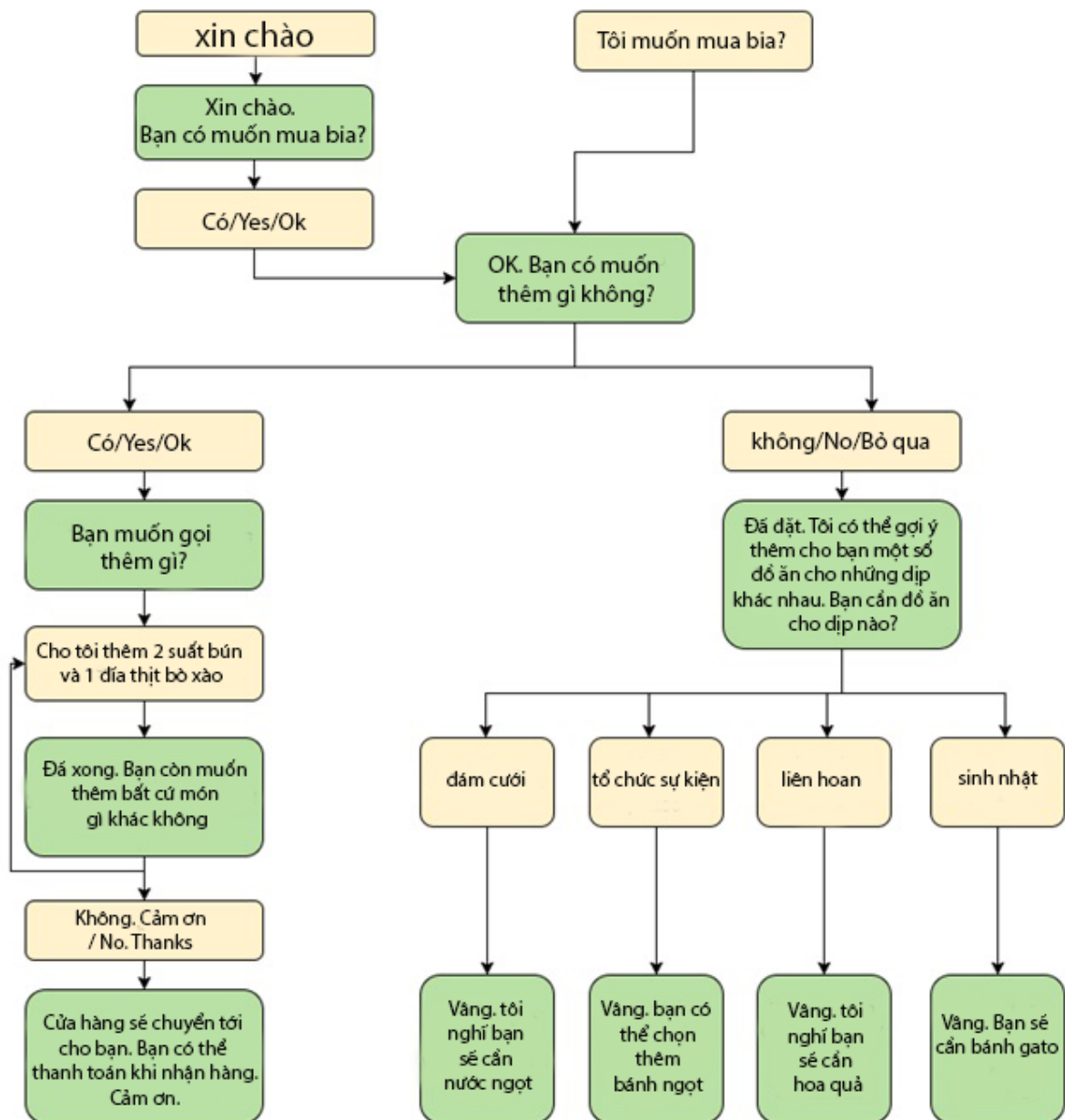
Nhận xét	Hộp thoại
<p>+ Dữ liệu sử dụng ảnh, sticker, vị trí</p> <p>+ Abot đang trong giai đoạn huấn luyện nên nó chưa hiểu về những dữ liệu này.</p> <p>+ Kết quả trả lời rất chung và không đúng mục đích hoặc không hiểu nội dung.</p>	 <p>The screenshot shows a chat interface with a grey header bar containing 'Carrier', signal strength, '9:30 AM', and battery level. The chat title is 'ABot'. The messages are as follows:</p> <ul style="list-style-type: none"> ABot (green bubble): Xin chào, Tôi là ABot User (blue bubble): [Unreadable] ABot (green bubble): ABot chưa rõ. Bạn có thể nói lại được không? <p>The input field at the bottom contains the text 'Nhập nội dung'. Below the input field is an emoji picker with various emojis and a 'Gửi' button.</p>

Dữ liệu huấn luyện lần 5

Nhận xét	Hộp thoại
<ul style="list-style-type: none">+ Dữ liệu sử dụng âm thanh+ Giữ liệu giọng nói được chuyển thành văn bản và gửi cho ABot.+ Kết quả nếu nói đúng nội dung được huấn luyện nó sẽ trả lời chính xác. Còn nếu nói chưa đúng nội dung huấn luyện sẽ trả lời chưa biết.	

3.4. Mô hình hội thoại

Bài toán luận văn xây dựng là xây dựng chatbot thực hiện việc đặt đồ ăn khi người dùng muốn đặt đồ ăn. Hình 3.7 là một frame giúp trong hộp thoại để người dung có thể thực hiện đặt đồ ăn. Ví dụ khi người dung nói “tôi muốn mua bia?” chatbot sẽ trả lời “OK” và hỏi người dùng có muốn đặt thêm gì không. Nếu người dùng trả lời “Có” thì Chatbot sẽ tiếp tục hỏi đặt món gì. Đến khi người dùng ngừng đặt đồ ăn thì Chatbot sẽ tiến hành thanh toán.



Hình 3.8: Mô hình hội thoại

Tiếp theo là dữ liệu về các chủ đề như đồ ăn, đồ uống, địa chỉ, thời tiết. Để xây dựng một con chatbot đủ thông minh đáp ứng được tất cả các câu hỏi thì cần rất nhiều dữ liệu và thời gian huấn luyện.

Drinks

SAVE

Define synonyms  Allow automated expansion



7 up	7 up, chai 7 up, cốc 7 up
coca	coca, Côca, Coca, chai coca, chai Côca, chai Coca
bia	bia, panh bia, cốc bia, bom bia, chai bia
pepsi	pepsi, chai pepsi, cốc pepsi
trà	trà, trà sen, trà đào, trà nhài, trà bạc hà, trà đào, trà lipton, trà mạn
mocktail	mocktail, nằng vàng hawaii, virgin colada, honey moon
sữa	sữa, sữa chua, sữa chua nếp cẩm, sữa chua hộp, sữa chua đánh đá, sữa tươi
nước ngọt	nước ngọt, cafe, sprite, Fanta, Twister, Redbull, String, Nước suối, Soda, Tonic
nước	nước cam, cốc nước cam, chai cam, nước chanh tươi, nước ổi, nước chanh leo, nước dừa, nước xoài, nước dưa hấu, nước táo, nước cà rốt, nước bưởi
sinh tố	sinh tố, cốc sinh tố, sinh tố bơ, sinh sốt mãng cầu, xoài, sinh tố đu đủ, sinh tố dưa hấu, sinh tố đu đủ, chanh leo tuyết
hoa quả dầm	hoa quả dầm, hoa quả thập cẩm, hoa quả dầm sữa chua

Hình 3.9: Dữ liệu đồ uống

Food

SAVE

Define synonyms  Allow automated expansion

Search entries 	
1 OF 6 	
Mì Spaghetti Hải Sản	mì spaghetti hải sản, mì mì ý hải sản, mì mì ý hải sản, mì spaghetti hải sản, mì mì ý hải sản
Tôm Luộc	tôm luộc, shrimp luộc
Bún	bún
Bún Chả	bún chả
Cơm Trộn	cơm trộn
Gỏi	gỏi
Cháo Gà Phô Mai	cháo gà phô mai, cháo gà pho mai, cháo gà phomai, cháo gà phô mát, cháo gà pho mát, cháo gà cheese, cháo gà phômai
Sò Điệp Nướng Mỡ Hành	sò điệp nướng mỡ hành, sò quạt nướng mỡ hành
Cháo Gà	cháo gà
Cánh Gà Nướng Phomai	cánh gà nướng phomai, cánh gà nướng pho mai, cánh gà nướng phô mát, cánh gà nướng pho mát, cánh gà nướng cheese, cánh gà nướng phô mai, cánh gà nướng phômai

Hình 3.10: Dữ liệu thức ăn

Location

SAVE

quận Gò Vấp	quận Gò Vấp, gò vấp
bình thành	bình thành, quận bình thành
quận 5	quận 5, ở quận 5, gần quận 5, xung quanh quận 5, trong quận 5, chỗ quận 5
tân phú	tân phú
quận 12	quận 12, ở quận 12
quận 10	quận 10
quận 7	quận 7, ở quận 7, gần quận 7, xung quanh quận 7, trong quận 7, chỗ quận 7
quận bình tân	quận bình tân
quận 9	quận 9
Hà nội	Hà nội, hanoi
Danang	Danang, Đà nẵng
Hochiminh	Hochiminh, Hồ chí minh
Click here to edit entry	

Hình 3.11: Dữ liệu địa điểm

WeatherNow

SAVE

Define synonyms ⓘ Allow automated expansion

Thời tiết bây giờ	Thời tiết bây giờ
Bây giờ thời tiết	Bây giờ thời tiết
Thời tiết hiện tại	Thời tiết hiện tại
Thời tiết hôm nay	Thời tiết hôm nay
Thời tiết như thế nào	Thời tiết như thế nào
Click here to edit entry	

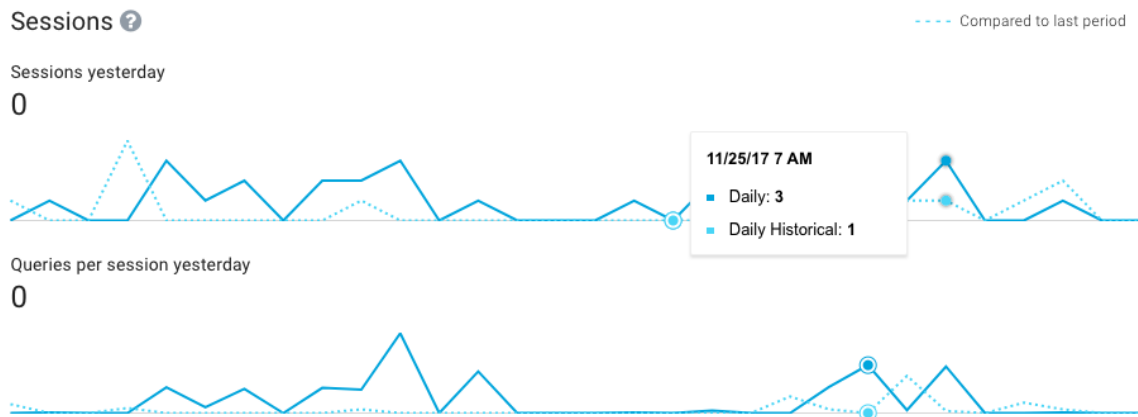
Hình 3.12: Dữ liệu thời tiết

3.5. Dữ liệu phân tích

Dựa trên dữ liệu được lấy trong 30 ngày gần nhất. Biểu đồ Sessions cho thấy tương tác giữa người dùng với Abot.



Hình 3.13: Dữ liệu truy vấn mỗi phiên làm việc



Hình 3.14: Dữ liệu phân tích tương tác

Dựa trên dữ liệu được lấy trong 30 ngày gần nhất. Biểu đồ Intents cho thấy dữ liệu Abot sử dụng các frame như Greeting, Foodyescompose... để lấy giá trị từ đó có câu trả lời phù hợp đáp ứng người dùng.

Intent	Count	Percentage	Time Range
foodyescompose	5	62	0%
Greeting	7	52	1.92%
foodnoaddcomposechoose	5	46	6.52%
foodcompose	5	43	2.33%
wether-yahoo	6	39	5.13%
food-webhook	3	33	6.06%

Hình 3.15: Dữ liệu intents

Tiến hành thử nghiệm chat ngẫu nhiên với ABot mỗi lần 50 câu lấy kết quả. Bảng kết quả đánh giá:

Bảng 3.1: Kết quả thử nghiệm

Thử nghiệm	Số câu đúng	Số câu sai	Độ chính xác
1	28/50	22/50	56%
2	31/50	19/50	62%
3	24/50	26/50	48%
4	39/50	11/50	78%
5	41/50	9/50	82%

Đánh giá kết quả thử nghiệm

Dựa trên bảng kết quả có thể thấy độ chính xác sẽ tăng dần. Có thể giải thích vì:

Lần 1: chat với bot thì có những câu bot chưa được huấn luyện nên sẽ dẫn tới nó không hiểu và trả lời sai. Vì thế độ chính xác sẽ thấp

Lần 2: những câu trả lời mới sẽ được huấn luyện lại cho bot hiểu, nên lần sau gặp câu đó nó sẽ trả lời đúng ý của người dùng. Vì thế độ chính xác sẽ tăng thêm.

Lần 3: chat ở những nội dung khác nhau, do huấn luyện chưa có nội dung đó nên trả lời sai vì thế độ chính xác cũng thấp.

Lần 4, 5: khi được huấn luyện tiếp, độ chính xác sẽ tăng và người dùng nói đúng nội dung bot được huấn luyện.

3.6. Dữ liệu câu hỏi

Trong quá trình thử nghiệm thì dữ liệu câu hỏi mỗi lần thử nghiệm là 50 câu. Và bảng dưới là lần thứ 4 thử nghiệm ABot với độ chính xác 78% trong đó có số câu trả lời đúng là 39/50, số câu trả lời sai là 11/50.

Bảng 3.2: Dữ liệu câu hỏi

STT	Người dùng	ABot	Đánh giá
1	Xin chào	Chào bạn, tôi sẽ giúp gì được cho bạn	
2	Tôi muốn mua bia?	Ok! Bạn có muốn thêm gì không?	
3	Có	Bạn muốn gọi thêm gì	
4	Tôi muốn mua 2 suất bún	Ok. Bạn vừa đặt suất Bún. Bạn có muốn thêm gì nữa không?	
5	Không cảm ơn	Cửa hàng sẽ chuyển tới cho bạn. Bạn có thể thanh toán khi nhận hàng. Cảm ơn.	
6	Tối nay ăn gì?	ABot chưa rõ. Bạn có thể nói lại được không?	Dữ liệu chưa được huấn luyện

7	Bún ở đâu ngon?	Bún có thể tới phở 10 Lý Quốc Sư	
8	Thời tiết hôm nay thế nào	Thời tiết chắc chắn sẽ tuyệt vời	
9	Thời tiết ở Hà Nội	Cũng không lạnh lắm	
10	Phở ở đâu ngon nhất	Cũng không lạnh lắm	Trả lời sai
11	Ăn phở ở đâu	Phở có thể tới phở 10 Lý Quốc Sư	
12	Tôi muốn đi chơi	Ok. Bạn muốn thêm gì không?	Câu hỏi chưa được huấn luyện nên trả lời sai.
13	Nhiệt độ Hà nội	ABot chưa rõ. Bạn có thể nói lại được không?	Dữ liệu chưa được huấn luyện
14	Tôi muốn đồ uống	Bạn muốn gọi thêm gì?	
15	bia	ABot chưa rõ. Bạn có thể nói lại được không?	Nội dung không rõ ràng nên Abot không hiểu
16	Tôi muốn 2 cốc bia	OK! Bạn có muốn thêm gì không?	Nội dung rõ ràng nên ABot trả lời đúng.
17	Đi ăn ở đâu	Cũng không lạnh lắm	Dữ liệu chưa được huấn luyện
18	Cho 2 suất thịt bò xào	OK. Bạn vừa đặt suất thịt. Bạn có muốn thêm gì nữa không?	
19	Tôi muốn mua đồ cho sinh nhật	OK. Bạn có muốn thêm gì không?	
20	Bánh ga tô	Bạn muốn gọi thêm gì?	
21	Tôi muốn ăn bún	Cửa hàng sẽ chuyển tới cho bạn. Bạn có thể thanh toán khi nhận hàng. Cảm ơn.	
22	Thời tiết bây giờ	Cũng không lạnh lắm	
23	Tôi muốn ăn bí tết	OK! Bạn có muốn thêm gì không?	
24	Không	Đã đặt. Tôi có thể gợi ý thêm cho bạn một số đồ ăn cho những dịp khác nhau. Bạn cần đồ ăn cho dịp nào?	
25	Đám cưới	Vâng. Tôi nghĩ bạn sẽ cần	

		nước ngọt	
26	Không	ABot chưa rõ. Bạn có thể nói lại được không?	Do không đúng kịch bản, ABot không hiểu dữ liệu.
27	Cho tôi 2 cốc bia	OK. Bạn có muốn thêm gì không.	
28	Sinh nhật	Vâng bạn có thể sẽ cần bánh gato	
29	Bánh ga tô mua ở đâu	Cũng không lạnh lắm	Trả lời sai do chưa huấn luyện tình huống này.
30	Mua bánh ga tô ở đâu	OK. Bạn muốn thêm gì không?	Trả lời không chính xác
31	Tối nay ăn bún ở đâu ngon?	Bún có thể tới phố 10 Lý quốc sư	
32	Đặt đồ ăn	Bạn muốn gọi thêm gì	
33	Thịt xào	Abot chưa rõ. Bạn có thể nói lại được không?	Nội dung không rõ ràng nên ABot không hiểu
34	3 suất bún 2 suất thịt bò	Đã xong. Bạn đã đặt thêm 3 suất Bún 2 Thịt bò. Bạn có muốn thêm gì không?	
35	Không	Cửa hàng sẽ chuyển tới cho bạn. Bạn có thể thanh toán khi nhận hàng. Cảm ơn.	
36	Cho tôi cà phê đen	OK. Bạn muốn thêm gì không?	
37	Suất thịt bò xào	OK. Bạn vừa đặt suất thịt. Bạn có muốn thêm gì nữa không?	
38	Đặt bánh ga tô	OK. Bạn muốn thêm gì không?	
39	1 cốc kem	OK. Bạn vừa đặt cốc kem. Bạn có muốn thêm gì nữa không?	
40	Nước trái cây	Bạn muốn gọi thêm gì?	
41	Bia với pepsi cái nào ngon hơn?	Bia sẽ ngon hơn pepsi. ABot nghĩ vậy	
42	Thịt bò với bún cái nào	Bún sẽ ngon hơn thịt bò.	

	ngon hơn?	Bạn có muốn thử không?	
43	Bia với pepsi	Bia sẽ ngon hơn pepsi. Bạn muốn đặt gì?	
44	Bia với nước lọc	Bia sẽ ngon hơn nước. Bạn muốn đặt gì?	
45	2 thịt bò xào	Abot chưa rõ. Bạn có thể nói lại được không?	
46	Bia đi	Bạn muốn gọi thêm gì?	
47	Bia với thịt bò xào	OK! Bạn có muốn thêm gì không?	
48	2 bia	OK. 2 bia đã đặt. Bạn muốn gì nữa không?	
49	Hai bát phở đi	Cửa hàng sẽ chuyên tới cho bạn. Bạn có thể thanh toán khi nhận hàng. Cảm ơn.	
50	Cảm ơn bot	Cảm ơn	

3.7. Nhận xét và kết quả

Sau khi thực hiện thử nghiệm Abot trên mobile kết quả thu được khi đặt câu hỏi là tỷ lệ đáp án chấp nhận được quá thấp, tuy nhiên đã giải quyết được một số vấn đề:

- Vấn đề phát sinh những câu hỏi nằm ngoài vùng tập dữ liệu huấn luyện, phần câu hỏi đặt ra hoàn toàn mang tính khách quan vì vậy Abot đã tự đưa ra một số câu trả lời khách quan và có độ phù hợp với câu hỏi của người dùng.
- Xây dựng mô hình có thể hiểu được ngôn ngữ tiếng việt mặc dù APLAI của Google không hỗ trợ tiếng việt và trả lời bằng câu trả lời tiếng việt.
- Với câu hỏi dài và phức tạp, tùy theo tham số độ dài huấn luyện và tùy theo tham số truyền vào thì Abot có khả năng đưa ra câu trả lời phù hợp.
- Độ tự động trả lời của Abot là có nhưng do thời gian huấn luyện chưa lâu nên việc trả lời mọi khía cạnh là không thể nên chỉ đưa ra câu trả lời khách quan.

Với những vấn đề hiện tại và kết quả đạt được thì Abot vẫn chưa đạt tới mức có thể áp dụng rộng rãi và cần thêm thời gian để phát triển thêm về ý tưởng và tập dữ liệu.

Ý tưởng phát triển cho Abot:

- Nâng cấp giao diện để có thể sử dụng Abot trong các cửa hàng, nhà hàng thực phẩm có sử dụng mạng để quảng cáo thức ăn
- Bổ xung thêm dữ liệu huấn luyện về nhiều mảng như địa chỉ, thời tiết, thời gian, thực đơn, thanh toán...
- Xây dựng framework riêng cho dự án không phụ thuộc vào google.
- Nâng cấp huấn luyện để Abot có thể là một trợ lý ảo thực sự như Siri.

Kết quả thu được sau quá trình nghiên cứu và hoàn thành luận văn

- Tìm hiểu về cấu trúc của hệ thống trợ lý ảo
- Hiểu các mô đun trong cấu trúc về cấu tạo, vận hành và thuật toán ứng dụng để có thể dựng riêng một framwork.
- Hiểu được cách huấn luyện chatbot để ứng dụng trong nhiều lĩnh vực trong công việc và đời sống.

KẾT LUẬN

Hệ thống trợ lý ảo đang là lĩnh vực mà các công ty công nghệ tập trung đầu tư nghiên cứu mạnh mẽ và ngày càng được phát triển. Ứng dụng hệ thống trợ lý ảo đang dần được đưa vào đời sống, được tích hợp từ những hệ thống cơ bản như chatbot cho tới tích hợp vào hệ thống phức tạp như y tế, dạy học... Qua luận văn, em đã đạt được những kết quả nhất định.

Đầu tiên là tìm hiểu về mô hình hệ thống trợ lý ảo, về cấu trúc hệ thống. Từ mô hình hệ thống trợ lý ảo giúp em hiểu được cấu tạo, thuật toán ứng dụng trong từng mô hình. Từ đó giúp em có thể đi sâu nghiên cứu từng mô đun trong hệ thống và ứng dụng thực tiễn là xây dựng một hệ thống trợ lý ảo phù hợp với mục đích của mình.

Thứ hai là từ việc huấn luyện chatbot đã giúp em có những kiến thức, độ hiểu biết sâu hơn về mạng học sâu, cách xây dựng các mô hình hộp thoại, xây dựng dữ liệu huấn luyện. Từ đó đúc kết được kiến thức để có thể ứng dụng thực tế trong công việc và mục đích huấn luyện một hệ thống trợ lý ảo.

Thứ ba là từ việc xây dựng mô hình chatbot, em có thể xây dựng nhiều mô hình chatbot với nhiều mục đích khác nhau ví dụ như chatbot cho hệ thống thực phẩm, cho y tế hay trong dạy học... Tùy theo yêu cầu và mục đích của thực tiễn có thể xây dựng được hệ thống trợ lý ảo đáp ứng yêu cầu người dùng.

Qua những kết quả đạt được ban đầu, em nhận thấy còn rất nhiều việc phải làm, cần phải tối ưu. Nhưng cách tiếp cận này ban đầu đã cho những kết quả rất tích cực và đúng đắn, có thể giải quyết được những vấn đề xây dựng và tính toán trong hệ thống đối thoại.

Định hướng nghiên cứu tiếp theo, em sẽ tiếp tục làm mượt dữ liệu, để tạo ra các mô hình mới có khả năng trả lời sát với ngữ cảnh, đạt chất lượng cao hơn, giảm khả năng lãng tránh và đưa tính cá nhân vào trong đoạn hội thoại. Từ đó áp dụng cho nhiều hệ thống trợ lý ảo với những yêu cầu và mục đích xây dựng khác nhau.

TÀI LIỆU THAM KHẢO

- [1] Apple's Siri. <https://www.apple.com/ios/siri/>.
- [2] Google's Google Now. <http://www.google.com/landing/now/>.
- [3] Microsoft's Cortana. <http://www.windowsphone.com/enus/features>
- [4] <https://web.stanford.edu/~jurafsky/slp3/>
- [5] Google's. <https://api.ai/docs/getting-started/basics>
- [6] Elizabeth D.Liddy. “*Natural Language Processing*”,papers 1-15, Syracuse University 2001.
- [7] Johann Hauswald, Michael A. Laurenzano, Yunqi Zhang, Cheng Li, Austin Rovinski, Arjun Khurana, Ronald G. Dreslinski, Trevor Mudge, Vinicius Petrucci1, Lingjia Tang, Jason Mars. “*Sirius: An Open End-to-End Voice and Vision Personal Assistant and Its Implications for Future Warehouse Scale Computers.*”,papers 1-16.
- [8] Daniel Jurafsky & James H. Martin.papers “*Speech and Language Processing*”. chapter 29, papers 1-25, August 7, 2017.
- [9] Oriol Vinyals, Quoc V. Le “*A Neural Conversational Model*”, papers 1-8, Jul 22, 2015.
- [10] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, Montreal, Canada, “*Sequence to Sequence Learning with Neural Networks*”,paper 1-9, December 08 - 13, 2014.

PHỤ LỤC

Mã nguồn ứng dụng

Kết nối API.AI

```
#pragma mark - APIAI
-(void)sendMessageText:(NSString *)text
{
    if ([Util boolForKey:IS_AI] ) {
        [MBProgressHUD showHUDAddedTo:self.view animated:YES];
        ApiAI *apiai = [ApiAI sharedApiAI];
        AITextRequest *request = [apiai textRequest];
        request.query = @[text?:@""];
        __weak typeof(self) selfWeak = self;
        [request setMappedCompletionBlockSuccess:^(AIRequest *request, AIResponse
        *response) {
            __strong typeof(selfWeak) selfStrong = selfWeak;
            [MBProgressHUD hideHUDForView:selfStrong.view animated:YES];
            NSString *str = response.result.fulfillment.speech;
            NSLog(@"% @",str);          [self addMessageForAPIAIwithText:str];
                [self.message TableView scrollToRowAtIndexPath:[NSIndexPath
indexPathForRow:self.demoData.messages.count-1                      inSection:0]
atScrollPosition:UITableViewScrollPositionMiddle animated:NO];
        } failure:^(AIRequest *request, NSError *error) {
            __strong typeof(selfWeak) selfStrong = selfWeak;
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Alert"
                message:[error localizedDescription]
                delegate:nil
                cancelButtonTitle:@"OK"
                otherButtonTitles:nil];
            [alertView show];
            [MBProgressHUD hideHUDForView:selfStrong.view animated:YES];
                [self.message TableView scrollToRowAtIndexPath:[NSIndexPath
indexPathForRow:self.demoData.messages.count-1                      inSection:0]
atScrollPosition:UITableViewScrollPositionMiddle animated:NO];
        }];
        [apiai enqueue:request];
    }else{
    }
}
```


Thêm dữ liệu Api.ai vào giao diện

```
-(void)addMessageForAPIAIwithText:(NSString *)text{
    if (text.length>0) {
        [self textToSpeed:text];
        ZHCMessage *message = [[ZHCMessage alloc] initWithSenderId:kIDAbot
                               senderDisplayName:kChatABot
                               date:[NSDate date]
                               text:text];

        [self.demoData.messages addObject:message];
        [self finishSendingMessageAnimated:NO];
    }else{
        text = @"ABot chưa rõ. Bạn có thể nói lại được không?";
        [self textToSpeed:text];
        ZHCMessage *message = [[ZHCMessage alloc] initWithSenderId:kIDAbot
                               senderDisplayName:kChatABot
                               date:[NSDate date]
                               text:text];

        [self.demoData.messages addObject:message];
        [self finishSendingMessageAnimated:NO];
    }
}
```

Thêm dữ liệu người dùng vào giao diện

```
-(void)addMessageForPerson:(NSString *)text
    senderId:(NSString *)senderId
    senderDisplayName:(NSString *)senderDisplayName
    date:(NSDate *)date{
    ZHCMessage *message = [[ZHCMessage alloc] initWithSenderId:senderId
        senderDisplayName:senderDisplayName
        date:date
        text:text];

    [self.demoData.messages addObject:message];
    [self finishSendingMessageAnimated:YES];
    [self sendMessageText:text];
}

-(void)addMVoiceForPerson:(NSString*)text{
    if (text.length>0) {
```

```

ZHCMessage *message = [[ZHCMessage alloc] initWithSenderId:kIDUser
                        senderDisplayName:kChatUser
                        date:[NSDate date]
                        text:text];

[self.demoData.messages addObject:message];
[self finishSendingMessageAnimated:YES];
[self sendMessageText:text];
}
}

```

Chuyển giọng nói sang văn bản

Hàm khởi tạo chuyển âm thanh sang văn bản

```

/#!
* @brief Starts listening and recognizing user input through the
* phone's microphone
*/

- (void)startListening {
    // Initialize the AVAudioEngine
    audioEngine = [[AVAudioEngine alloc] init];
    // Make sure there's not a recognition task already running
    if (recognitionTask) {
        [recognitionTask cancel];
        recognitionTask = nil;
    }
    // Starts an AVAudio Session
    NSError *error;
    AVAudioSession *audioSession = [AVAudioSession sharedInstance];
    [audioSession setCategory:AVAudioSessionCategoryRecord error:&error];
    [audioSession setActive:YES
    withOptions:AVAudioSessionSetActiveOptionNotifyOthersOnDeactivation
    error:&error];
    // Starts a recognition process, in the block it logs the input or stops the audio
    // process if there's an error.
    recognitionRequest = [[SFSpeechAudioBufferRecognitionRequest alloc] init];
    AVAudioInputNode *inputNode = audioEngine.inputNode;
    recognitionRequest.shouldReportPartialResults = YES;
    recognitionTask = [speechRecognizer
    recognitionTaskWithRequest:recognitionRequest

```

```

resultHandler:^(SFSpeechRecognitionResult * _Nullable result, NSError * _Nullable
error) {
    BOOL isFinal = NO;
    if (result) {
        // Whatever you say in the microphone after pressing the button should be being
logged
        // in the console.
        NSLog(@"RESULT:%@",result.bestTranscription.formattedString);
        [MBProgressHUD hideHUDForView:self.view animated:YES];
        lastString = result.bestTranscription.formattedString;
        [ZHCMessagesAudioProgressHUD zhc_changeSubTitle:lastString];
        isFinal = !result.isFinal;
    }
    if (error) {
        NSLog(@"error:%@",result.bestTranscription.formattedString);
        [audioEngine stop];
        [recognitionRequest endAudio];
        recognitionRequest = nil;
        recognitionTask = nil;
    }
}];
// Sets the recording format
AVAudioFormat *recordingFormat = [inputNode outputFormatForBus:0];
[inputNode installTapOnBus:0 bufferSize:1024 format:recordingFormat
block:^(AVAudioPCMBuffer * _Nonnull buffer, AVAudioTime * _Nonnull when) {
    [recognitionRequest appendAudioPCMBuffer:buffer];
}];

// Starts the audio engine, i.e. it starts listening.
[audioEngine prepare];
[audioEngine startAndReturnError:&error];
}

```

Hàm bắt tín hiệu người dùng nói

```

-(void)microPhoneTapped:(id)sender {
    if (audioEngine.isRunning) {
        [audioEngine stop];
        [recognitionRequest endAudio];
    } else {
        [self startListening];
    }
}

```

```
    }  
  }  
  #pragma mark - SFSpeechRecognizerDelegate Delegate Methods  
  - (void)speechRecognizer:(SFSpeechRecognizer *)speechRecognizer  
  availabilityDidChange:(BOOL)available {  
    NSLog(@"Availability:%d",available);  
  }  
}
```