

THIẾT KẾ ROBOT TRỢ LÝ GIẢNG DẠY GIAO TIẾP BẰNG GIỌNG NÓI

DESIGN OF VOICE COMMUNICATION-BASED TEACHING ASSISTANT ROBOT

Đoàn Huỳnh Công Sơn, Trương Ngọc Sơn
Trường Đại học Sư phạm Kỹ thuật TP.HCM, Việt Nam

Ngày tòa soạn nhận bài 13/7/2020, ngày phản biện đánh giá 4/8/2020, ngày chấp nhận đăng 5/8/2020

TÓM TẮT

Nghiên cứu này trình bày thiết kế và kiểm nghiệm thử một Robot trợ lý giảng dạy hoạt động như một trợ lý ảo nhưng có khả năng giao tiếp bằng giọng nói, không cần kết nối mạng như các trợ lý ảo hiện tại và sử dụng mã nguồn mở Pocketsphinx để nhận dạng giọng nói. Nghiên cứu cũng sử dụng khối hiệu chỉnh đặt sau Pocketsphinx để nâng cao độ chính xác cho khối nhận dạng giọng nói. Phần lõi xử lý ngôn ngữ tự nhiên được sử dụng là mạng LSTM cho phép đưa ra câu trả lời phù hợp mà mạng đã được huấn luyện. Chương trình được cài đặt trên phần cứng nhúng Raspberry Pi Zero có mức tiêu tốn năng lượng thấp. Kết quả kiểm thử và đánh giá cho thấy, nếu không sử dụng khối hiệu chỉnh robot có khả năng nhận dạng và trả lời câu hỏi với tỷ lệ chính xác là 62.5%. Tỷ lệ nhận dạng được tăng lên đáng kể (87.2%) khi sử dụng khối hiệu chỉnh.

Từ khóa: Robot trợ lý giảng dạy; Xử lý ngôn ngữ tự nhiên; Mạng Long short-term memory; Nhận dạng giọng nói.

ABSTRACT

This study presents a design and testing of a teaching assistant robot working as a virtual assistant that is capable of communicating using voice with humans. Robot can work without connecting to the network, the open source Pocketsphinx is employed for speech recognition. The Pocketsphinx module is followed by a correction module to improve the accuracy. The Long Short-Term Memory is utilized for the natural language processing unit that produces the answers. The model is deployed on the low-cost embedded board, Raspberry Pi Zero. The evaluation was performed with and without using the proposed correction module. The accuracy is 62.5% when using Pocketsphinx without the proposed correction module. With the proposed correction module, the robot improved the identifying and answering questions capacities to 87.2% of accuracy.

Keywords: Assistant robots; Natural language processing; Long Short-Term Memory Network; Machine learning, Speech processing.

1. GIỚI THIỆU

Trong những năm gần đây, ngành khoa học trí tuệ nhân tạo phát triển mạnh mẽ tạo ra nhiều ứng dụng hữu ích trong nhiều mặt của cuộc sống. Một trong các ứng dụng của khoa học trí tuệ nhân tạo là các trợ lý ảo. Các trợ lý ảo hay còn được biết đến với tên gọi

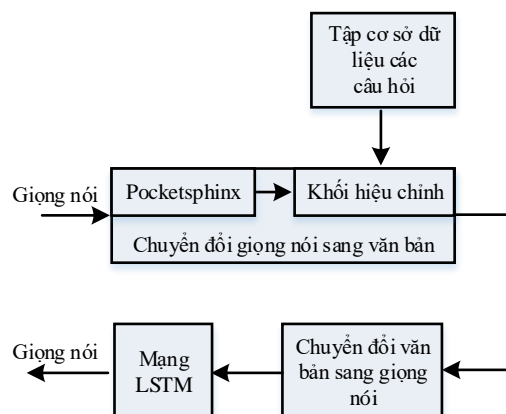
Chatbot là các phần mềm dựa trên trí tuệ nhân tạo được huấn luyện với một lượng kiến thức về một lĩnh vực nào đó và có khả năng đưa ra câu trả lời khi nhận được câu hỏi [1]. Các Chatbot được ứng dụng rộng rãi trong các lĩnh vực như kinh doanh, chăm sóc sức khỏe và giáo dục [2]-[6]. Các Chatbot chỉ dừng lại ở việc nhận và trả lời câu hỏi thông

qua chế độ giao tiếp bằng văn bản (text-based interface). Các Chatbot được kết nối thêm một mô đun nhận dạng giọng nói sẽ tạo ra một Robot có khả năng giao tiếp và trả lời các câu hỏi từ người dùng [7]-[10]. Các Robot được thiết kế có khả năng giao tiếp được sử dụng cho việc hỗ trợ giảng dạy [9], [10]. Phần cốt lõi của các robot trợ lý là kỹ thuật nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên, trong đó nhận dạng giọng nói đóng vai trò cốt yếu đến độ chính xác của robot. Phần lớn các robot trợ lý cũng như các thiết bị điều khiển thiết bị bằng giọng nói hiện nay đều sử dụng công cụ nhận dạng giọng nói của Google (Google Cloud Speech API) bởi nó cho độ chính xác cao và tốc độ đáp ứng nhanh [11]. Tuy nhiên, sử dụng dịch vụ từ Google đòi hỏi hệ thống phải được kết nối internet liên tục và đường truyền phải đảm bảo về mặt tốc độ. Để thiết kế các Robot trợ lý có khả năng giao tiếp bằng giọng nói hoạt động độc lập không cần phải kết nối mạng, việc lựa chọn công cụ nhận dạng giọng nói là cần thiết. Các mô đun nhận dạng giọng nói có thể phát triển dựa trên các mạng học sâu. Tuy nhiên, việc huấn luyện các mạng học sâu đòi hỏi phải có tập dữ liệu huấn luyện lớn và các mạng học sâu không hiệu quả khi được triển khai trên các hệ thống nhúng như Raspberry cho các robot di động. Pocketsphinx là mô đun nhận dạng giọng nói mã nguồn mở được phát triển phù hợp cho các hệ thống có tài nguyên giới hạn [12]. Nhận dạng giọng nói sử dụng mô đun pocketsphinx có khả năng hoạt động độc lập không cần kết nối mạng internet, tuy nhiên, độ chính xác thấp hơn so với các công cụ như Google. Để tăng độ chính xác, trong thiết kế này, tác giả kết hợp mô đun Pocketsphinx để nhận dạng giọng nói và giải thuật tìm kiếm tương quan dựa trên khoảng cách Levenshtein để hiệu chỉnh lỗi. Phương pháp đề xuất cho phép tăng độ chính xác của mô đun nhận dạng giọng nói. Mô đun xử lý ngôn ngữ tự nhiên để lựa chọn câu trả lời sử dụng mạng nơ-ron hồi quy cải tiến (mạng LSTM). Robot được thiết kế để hỗ trợ giảng dạy và được huấn luyện với kiến thức của một môn học. Robot có khả năng trả lời câu

hỏi từ người học về các nội dung của môn học mà Robot được huấn luyện.

2. THIẾT KẾ ROBOT TRỢ LÝ GIẢNG DẠY

Robot trợ lý giảng dạy được phát triển như mô hình các trợ lý ảo (chatbot). Tuy nhiên, khác với các chatbot chỉ có khả năng giao tiếp với người dùng qua chế độ văn bản (text), Robot trợ lý giảng dạy có thể giao tiếp qua ngôn ngữ. Robot được huấn luyện với nội dung một môn học kỹ thuật và có khả năng trả lời các câu hỏi liên quan đến môn học. Để thiết kế robot trợ lý giảng dạy, tác giả chọn một môn học về ngôn ngữ lập trình, môn học cơ sở được dạy nhiều ở các trường kỹ thuật. Sơ đồ khối hệ thống điều khiển robot được trình bày trong hình 1.



Hình 1. Sơ đồ khối hệ thống điều khiển

Đặc điểm quan trọng của thiết kế là Robot hoàn toàn có thể hoạt động độc lập mà không cần kết nối internet (offline). Mô đun nhận dạng giọng nói sử dụng thư viện mã nguồn mở, Pocketsphinx, cho phép chuyển đổi từ giọng nói sang văn bản (speech to text) có khả năng hoạt động offline. Để tăng độ chính xác, tác giả đề xuất thêm khối hiệu chỉnh độ chính xác kết nối sau ngõ ra của Pocketsphinx, như được thể hiện trong hình 1. Hạn chế của mô đun Pocketsphinx là cho độ chính xác kém hơn so với các công cụ hoạt động trực tuyến (online) như Google Speech Recognition API [13]. Khối hiệu chỉnh được lập trình bằng ngôn ngữ Python, sử dụng thuật toán khoảng cách Levenshtein. Thuật toán khoảng cách Levenshtein cho phép tính độ giống như giữa 2 chuỗi [14].

Kết quả nhận dạng được từ Pocketsphinx là một câu hỏi dạng chuỗi, câu hỏi này được đưa sang khối hiệu chỉnh. Tại khối hiệu chỉnh, độ giống nhau của câu hỏi và các câu hỏi mẫu trong tập huấn luyện sẽ được tính dựa trên thuật toán khoảng cách Levenshtein để tìm xem câu hỏi nào trong tập câu hỏi mẫu có độ giống nhau với câu hỏi nhận được lớn nhất. Khi xác định được độ giống nhau lớn nhất, câu hỏi có độ giống nhau lớn nhất với câu hỏi nhận được từ Pocketsphinx sẽ được lựa chọn và đưa sang khối tiếp theo. Khối hiệu chỉnh giúp nâng cao độ chính xác của khối nhận dạng tiếng nói đáng kể. Bảng 1 trình bày ví dụ hoạt động của khối hiệu chỉnh khi khối pocketsphinx nhận được câu hỏi và chuyển sang văn bản.

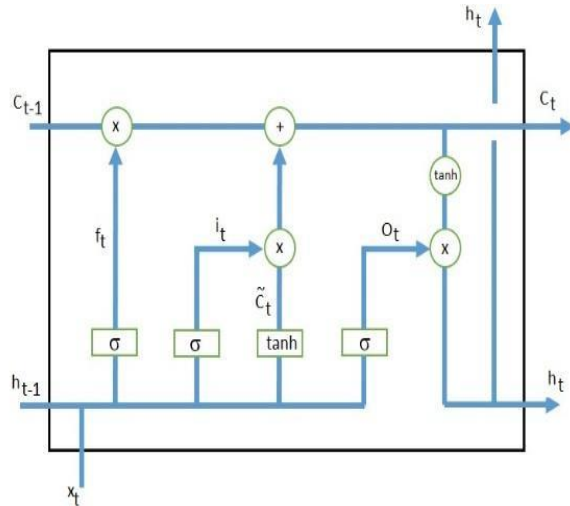
Hoạt động của khối hiệu chỉnh được ví dụ minh họa trong bảng 1. Khi khối Pocketsphinx nhận dạng được giọng nói, kết quả nhận dạng được chuyển sang văn bản (text) và chuyển sang khối hiệu chỉnh. Tại đây khối hiệu chỉnh sẽ tính độ giống nhau giữa câu hỏi và bộ câu hỏi mẫu lưu trong cơ sở dữ liệu sử dụng thuật toán Levenshtein và chọn ra câu hỏi giống nhất từ trong cơ sở dữ liệu.

Bảng 1. Hoạt động của khối hiệu chỉnh với câu nhận được từ Pocketsphinx

Pocketsphinx	What the function be since
Câu hỏi mẫu 1	What is a function in C
Độ giống nhau với câu hỏi mẫu 1	0.593
Câu hỏi mẫu 2	What is a variable in C
Độ giống nhau với câu hỏi mẫu 2	0.407
Lựa chọn ngõ ra	What is a function in C

Khối xử lý ngôn ngữ tự nhiên sử dụng mạng Long Short-Term Memory, một dạng cải tiến của mạng hồi quy (Recurrent Neural Network). LSTM cho phép nhận dạng các đối tượng theo thứ tự thời gian, phù hợp với các ứng dụng mà trong đó ngõ vào có ràng buộc về mặt thứ tự như chuỗi. LSTM được

ứng dụng nhiều trong kỹ thuật xử lý ngôn ngữ tự nhiên [15], [16]. Nhược điểm của mạng hồi quy là các giá trị lỗi có xu hướng nhỏ dần khi lan truyền qua nhiều lớp mạng làm cho quá trình cập nhật trọng số không hiệu quả trong quá trình huấn luyện. Mạng LSTM là một cải tiến để khắc phục nhược điểm này của mạng hồi quy [15]. Một lớp của mạng LSTM được thể hiện ở hình 2.



Hình 2. 1 lớp của mạng LSTM [15]

Mạng LSTM có khả năng giữ lại hoặc loại bỏ các thông tin không cần thiết thông qua các cổng Gate, đó là sự kết hợp giữa một phép nhân và tầng Sigmoid để sàng lọc thông tin. Hàm sigmoid cho ra kết quả nằm trong khoảng $[0,1]$, nếu đầu ra bằng 0 thì sẽ loại bỏ thông tin đó, nếu bằng 1 thì cho tất cả thông tin đi qua. Đầu tiên các thông tin sẽ được đi qua tầng 1 hay còn được gọi là tầng forget gate layer, tại đây thông tin sẽ giữ lại hoặc loại bỏ đi tùy thuộc vào kết quả của tầng sigmoid.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

Tiếp đó là bước kiểm tra thông tin mới, xem thông tin nào sẽ được lưu vào trạng thái tế bào C_{t-1} . Ở bước này ta sử dụng một tầng sigmoid được gọi là input layer gate kết hợp với tầng hàm tanh để cập nhật trạng thái mới.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$

Bước kế tiếp là cập nhật tế bào mới C_t , dựa vào sơ đồ hình 2 ta suy ra được:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Bước cuối cùng là bước quyết định xem đầu ra là gì, kết quả đó sẽ là đầu vào cho các hidden layer tiếp theo nên sẽ được sản lọc thông tin lần cuối cùng dựa vào hàm sigmoid và sau đó nhân với hàm tanh để đưa ra giá trị đầu ra mong muốn

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

Các tập dữ liệu được thu thập bao gồm tập câu hỏi mẫu phục vụ cho quá trình hiệu chỉnh và tập dữ liệu cho quá trình huấn luyện mạng LSTM. Cấu trúc dữ liệu huấn luyện mạng bao gồm một câu hỏi làm đầu vào cho quá trình mã hóa (Encoder) của mạng LSTM và câu trả lời cho quá trình giải mã (Decoder) [8]. Mạng LSTM được thiết kế sử dụng thư viện Keras. Kiến trúc mạng LSTM được tóm tắt trong hình 3.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, None)	0	
input_2 (InputLayer)	(None, None)	0	
embedding_1 (Embedding)	(None, None, 200)	168000	input_1[0][0]
embedding_2 (Embedding)	(None, None, 200)	168000	input_2[0][0]
lstm_1 (LSTM)	((None, 200), (None, 320800))	320800	embedding_1[0][0]
lstm_2 (LSTM)	((None, None, 200), (None, 320800))	320800	embedding_2[0][0] lstm_1[0][1] lstm_1[0][2]
dense_1 (Dense)	(None, None, 840)	168840	lstm_2[0][0]

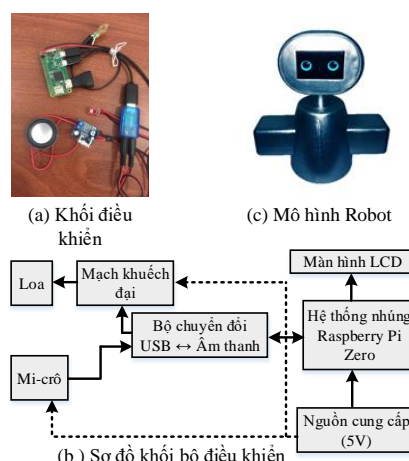
Total params: 1,146,440
 Trainable params: 1,146,440
 Non-trainable params: 0

Hình 3. Kiến trúc mô hình LSTM sử dụng thư viện Keras

Cấu trúc mạng trong hình 3 gồm: 2 input layer, 2 embedding và 2 LSTM với số nơ-ron lớp ẩn là 200. Số lượng nơ-ron trong lớp ẩn được lựa chọn tối ưu sao cho mạng được triển khai hiệu quả trên cấu hình phần cứng tương đối thấp của Raspberry Pi Zero. Với tập dữ liệu cụ thể cho một môn học kỹ thuật, số lượng nơ-ron lớp ẩn là 200 đảm bảo độ chính xác và tốc độ đáp ứng khi được triển khai trên phần cứng Raspberry Pi Zero. Dựa vào tập dữ liệu, có khoảng 840 từ khác nhau, do đó, lựa chọn lớp đầu vào (Input layer) của encoder_input và decoder_input với số mẫu (num_sample) là 840. Embedding layer là lớp chuyển đổi không gian vector của encoder_embedding và decoder_embedding với kích thước $840 \times 200 = 168000$. LSTM layer nhận đầu vào từ các embedding layer,

mỗi ô nhớ LSTM yêu cầu đầu vào là mảng 3 chiều, khi một LSTM xử lý một chuỗi đầu vào của các bước thời gian mỗi ô nhớ sẽ xuất một giá trị duy nhất cho toàn bộ chuỗi dưới dạng một mảng 2 chiều.

Mô hình LSTM được huấn luyện trên máy chủ sử dụng bộ tăng tốc đồ họa (Graphic Processing Unit) GTX1080 để rút ngắn thời gian huấn luyện. Tập dữ liệu huấn luyện cho Robot được tác giả tự tạo, bao gồm các câu hỏi kèm theo câu trả lời được lưu dưới dạng các tập tin văn bản (.txt). Cấu trúc tập dữ liệu bao gồm một câu hỏi cho ngõ vào quá trình Encoder và một câu trả lời cho quá trình Decoder của mạng LSTM. Các câu hỏi và câu trả lời được biên soạn bám theo nội dung một môn học kỹ thuật, cụ thể là môn Ngôn ngữ lập trình C. Mô hình sau khi huấn luyện và chương trình xử lý được cài đặt xuống cho phần cứng nhúng Raspberry Pi. Để Robot có thể trả lời các câu hỏi từ người dùng bằng giọng nói, tác giả thiết kế khối chuyên đổi từ văn bản sang giọng nói. Dữ liệu giọng nói được ghi âm cho từng từ riêng lẻ và được lưu dưới dạng các tệp âm thanh (.wav) trong thẻ nhớ. Kết quả đưa ra từ mô hình LSTM sẽ được ánh xạ sang các tệp âm thanh. Chương trình sẽ gọi thực thi các tệp âm thanh tương ứng để tạo ra câu trả lời. Chương trình được triển khai trên phần cứng Raspberry Pi Zero và được lắp đặt vào mô hình Robot như hình 4.



Hình 4. (a) Kết nối bộ điều khiển, (b) sơ đồ khối bộ điều khiển, (c) thiết kế phần thân Robot

Hình 4 mô tả kết nối của khối điều khiển, sơ đồ khối bộ điều khiển và phần thân

của robot. Khối điều khiển được sử dụng là hệ thống nhúng Raspberry Pi Zero có kích thước nhỏ và giá thành thấp, phù hợp với các thiết kế robot di động. Hệ thống sử dụng một Microphone có tích hợp bộ khuếch đại với hệ số tín hiệu trên nhiễu (SNR) là 62 dBA để tăng khoảng cách thu và loại bỏ nhiễu. Tín hiệu âm thanh được khuếch đại và đưa đến ngõ ra loa. Hệ thống sử dụng pin với dung lượng 4200 mAh cho phép Robot có thể hoạt động liên tục trong thời gian hơn 2 giờ. Phần thân robot được chế tạo bằng nhựa như hình 4(c). Phần mặt robot được gắn thêm một màn hình cho phép hiển thị một số thông tin, hình ảnh trong quá trình giao tiếp.

3. KẾT QUẢ VÀ THẢO LUẬN

Sau khi hoàn thiện, Robot được vận hành và kiểm tra khả năng nhận dạng câu hỏi và khả năng trả lời. Qua thực nghiệm cho thấy, Robot trả lời các câu hỏi sai khi nhận dạng sai. Trong trường hợp nhận dạng giọng nói đúng, Robot trả lời đúng với kịch bản trong tập huấn luyện. Tác giả tiến hành thực nghiệm bằng cách giao tiếp với robot qua 100 câu hỏi trong điều kiện phòng thí nghiệm, không có tiếng ồn từ môi trường. Để đánh giá hiệu quả của khối hiệu chỉnh được đề xuất, tác giả đo độ chính xác trong trường hợp có khối hiệu chỉnh và không có khối hiệu chỉnh. Khoảng cách từ người nói đến vị trí robot là 1m. Kết quả được thể hiện trong bảng 2.

Bảng 2. Kiểm tra độ chính xác hoạt động của Robot.

Mô hình	Độ chính xác (%)
Không sử dụng khối hiệu chỉnh	62.5
Sử dụng khối hiệu chỉnh	87.2

Kết quả thực nghiệm được đo trong điều kiện phòng, không có tiếng ồn. Kết quả được thực nghiệm 5 lần và lấy giá trị trung bình. Trong mỗi lần thực nghiệm sử dụng một giọng nói khác nhau và giao tiếp với Robot qua 100 câu hỏi liên quan đến môn học Ngôn ngữ lập trình. Bảng 2 liệt kê kết quả trung

bình trong 5 lần thực nghiệm với 2 mô hình; không sử dụng khối hiệu chỉnh và sử dụng khối hiệu chỉnh. Khi không sử dụng khối hiệu chỉnh, tỷ lệ nhận dạng và trả lời câu hỏi đúng là 62.5%. Trong trường hợp sử dụng khối hiệu chỉnh, độ chính xác được nâng lên 87.2%.

Sử dụng khối hiệu chỉnh cho phép tăng độ chính xác của robot khi robot hoạt động ngoại tuyến (offline). Các mô đun được cài đặt trên phần cứng có cấu hình thấp, Raspberry Pi Zero, giúp tiết kiệm năng lượng khi hoạt động. Robot có khả năng giao tiếp với người thông qua giọng nói và trả lời một số câu hỏi liên quan đến môn học Ngôn ngữ lập trình. Thời gian hoạt động liên tục của Robot khoảng 2 giờ vì hệ thống nhúng Raspberry Pi Zero tiêu tốn ít năng lượng.

Thời gian đáp ứng của Robot được đo trong quá trình kiểm thử 100 câu hỏi và câu trả lời. Các câu hỏi và câu trả lời có độ dài từ 3 đến 15 từ. Trung bình thời gian cho quá trình chuyển đổi từ giọng nói sang văn bản là 2.7s. Thời gian trung bình cho quá trình trả lời từ mạng LSTM là 1.2s. Như vậy, tổng thời gian cho quá trình trả lời câu hỏi là 3.9s. Thời gian đáp ứng của hệ thống đã được tối ưu nhờ tắt các dịch vụ không cần thiết của hệ điều hành cho hệ thống nhúng Raspberry Pi Zero.

4. KẾT LUẬN

Robot trợ lý giảng dạy được thiết kế có khả năng giao tiếp với con người bằng giọng nói và có thể trả lời các câu hỏi liên quan đến các nội dung của môn học được cài đặt trước. Robot đã được thiết kế hoàn thiện về phần cứng và phần mềm. Trong đó, bộ điều khiển robot sử dụng hệ thống nhúng Raspberry Pi Zero thực thi 3 khối chức năng chính bao gồm khối nhận dạng giọng nói, khối xử lý ngôn ngữ tự nhiên và khối chuyển đổi từ văn bản sang giọng nói. Kết quả thực nghiệm, robot có khả năng nhận dạng giọng nói và trả lời các câu hỏi liên quan đến nội dung môn học Ngôn ngữ lập trình C. Hơn nữa, việc sử dụng khối hiệu chỉnh đặt sau khối Pocketsphinx đã cải thiện tỷ lệ nhận dạng và trả lời chính xác câu hỏi từ 62.5% lên 87.2%.

LỜI CẢM ƠN

Kết quả nghiên cứu và ứng dụng là sản phẩm của Đề tài Nghiên cứu Khoa học Cấp Bộ, mã số B2019-SPK-05, được hỗ trợ bởi Bộ Giáo dục và Đào tạo và chủ trì bởi Trường Đại học Sư phạm Kỹ thuật TP.HCM.

TÀI LIỆU THAM KHẢO

- [1] B. Setiaji and F. W. Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling," *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Bangkok, 2016, pp. 72-77.
- [2] G. M. D'silva, S. Thakare, S. More and J. Kuriakose, "Real world smart chatbot for customer care using a software as a service (SaaS) architecture," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, 2017, pp. 658-664.
- [3] M. Bates, "Health Care Chatbots Are Here to Help", *IEEE Pulse*, vol. 10, no. 3, pp. 12–14, May 2019.
- [4] D. Madhu, C. J. N. Jain, E. Sebastain, S. Shaji and A. Ajayakumar, "A novel approach for medical assistance using trained chatbot," *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, 2017, pp. 243-246.
- [5] A. Mondal, M. Dey, D. Das, S. Nagpal and K. Garda, "Chatbot: An automated conversation system for the educational domain," *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Pattaya, Thailand, 2018, pp. 1-5
- [6] B. R. Ranoliya, N. Raghuvanshi and S. Singh, "Chatbot for university related FAQs," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, 2017, pp. 1525-1530
- [7] S. J. du Preez, M. Lall and S. Sinha, "An intelligent web-based voice chat bot," *IEEE EUROCON 2009*, St.-Petersburg, 2009, pp. 386-391.
- [8] J. Liu and B. Zhu, "An intelligent personal assistant robot: BoBi secretary," *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, Hefei, 2017, pp. 402-407.
- [9] J. Han et al., "A trial English class with a teaching assistant robot in elementary school," *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Osaka, 2010, pp. 335-335.
- [10] J. Han, S. Ji and S. Lee, "Development of collective assistant teaching for r-learning in a kindergarten", *9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Daejeon, 2012, pp. 136-139.
- [11] G. Boza-Quispe, J. Montalvan-Figueroa, J. Rosales-Huamaní and F. Puente-Mansilla, "A friendly speech user interface based on Google cloud platform to access a tourism semantic website", *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, Pucon, 2017, pp. 1-4.
- [12] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar and A. I. Rudnicky, "Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices", *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Toulouse, 2006, pp. I-I.
- [13] C. Patel and S. Koppurapu, "Reusing automatic speech recognition platform for resource deficient languages," *2014 8th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Gold Coast, QLD, 2014, pp. 1-5.

- [14] A. Ene and A. Ene, “An application of Levenshtein algorithm in vocabulary learning”, *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Targoviste, 2017, pp. 1-4.
- [15] S. Zhang, S. Liu and M. Liu, “Natural language inference using LSTM model with sentence fusion”, *2017 36th Chinese Control Conference (CCC)*, Dalian, 2017, pp. 11081-11085.
- [16] Q. Zhan, L. Zhang, H. Deng and X. Xie, “An Improved LSTM For Language Identification”, *2018 14th IEEE International Conference on Signal Processing (ICSP)*, Beijing, China, 2018, pp. 609-612.

Tác giả chịu trách nhiệm bài viết:

TS. Trương Ngọc Sơn

Trường Đại học Sư phạm Kỹ thuật TP.HCM

Email: sontn@hcmute.edu.vn