

MỘT CẢI TIẾN KHẢ NĂNG THÍCH ỨNG CHO CHATBOT ĐA NGÔN NGỮ

Ngô Văn Sơn¹, Thái Thị Phương², Võ Việt Minh Nhật²

¹Trường Du lịch - Đại học Huế

²Ban Đào tạo và Công tác sinh viên Đại học Huế

ngovanson@hueuni.edu.vn, ttpuong@hueuni.edu.vn, vvmnhat@hueuni.edu.vn

TÓM TẮT: Bài báo trình bày một cải tiến khả năng thích ứng cho chatbot đa ngôn ngữ dựa trên kiến trúc RASA và mô hình tiền huấn luyện BERT. Mục tiêu là nhằm xây dựng một chatbot có khả năng hiểu và phản hồi trên nhiều ngôn ngữ mà không cần phải triển khai riêng biệt cho từng ngôn ngữ. Phương pháp tiếp cận tập trung vào hai khía cạnh chính. Đầu tiên, để chatbot có thể xử lý đa ngôn ngữ, môđun "Phát hiện ngôn ngữ" được đưa vào bộ quản lý đối thoại của RASA. Thành phần này giúp chatbot tự động nhận diện ngôn ngữ của câu hỏi người dùng mà không cần khai báo cụ thể loại ngôn ngữ. Thứ hai, dữ liệu huấn luyện và dữ liệu phản hồi của chatbot được điều chỉnh theo nhiều ngôn ngữ cần hỗ trợ. Kết quả cài đặt cho thấy chatbot có thể hiểu được nhiều ngôn ngữ một cách linh hoạt. Cải tiến khả năng thích ứng cho chatbot đa ngôn ngữ này do đó có thể ứng dụng rộng rãi trong thực tế và đáp ứng nhu cầu giao tiếp đa ngôn ngữ.

Từ khóa: Chatbot, RASA, BERT, Tiền huấn luyện, Đa ngôn ngữ.

I. GIỚI THIỆU

Chatbot đã trở thành một công cụ quan trọng trong tương tác và hỗ trợ tự động đối với người dùng. Với khả năng tiếp thu và phân tích ngôn ngữ, chatbot có thể hiểu và phản hồi lại các yêu cầu, câu hỏi của người dùng một cách tự động và nhanh chóng. Tuy nhiên, một thách thức lớn đối với chatbot là khả năng thích ứng ngôn ngữ, đặc biệt là trong môi trường đa ngôn ngữ [1, 2]. Về bản chất, chatbot là phần mềm tích hợp trí tuệ nhân tạo được thiết kế để tương tác với con người thông qua ngôn ngữ tự nhiên. Trong môi trường đa ngôn ngữ, chatbot phải có khả năng đáp ứng các yêu cầu và câu hỏi từ người dùng với nhiều ngôn ngữ khác nhau. Sự phát triển của xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) và học máy (Machine learning) đã tạo ra tiềm năng lớn cho chatbot đa ngôn ngữ [3].

Khả năng thích ứng ngôn ngữ của chatbot đa ngôn ngữ là một yếu tố quan trọng để cung cấp trải nghiệm tương tác tự nhiên và hiệu quả cho người dùng [4]. Nó cho phép chatbot hiểu, phản hồi lại các yêu cầu và câu hỏi từ người dùng với ngôn ngữ mà họ quen thuộc. Môi trường đa ngôn ngữ đặt ra những thách thức đáng kể cho chatbot. Đầu tiên, đó là sự đa dạng về ngữ cảnh văn bản, ngữ nghĩa và cấu trúc ngôn ngữ trong các ngôn ngữ khác nhau. Chatbot cần có khả năng hiểu và xử lý các biến thể ngôn ngữ này để đảm bảo tính chính xác và hiệu quả. Thứ hai, chatbot cần phải xử lý các vấn đề về phản hồi và tương tác trong thời gian thực, vì mỗi ngôn ngữ có các cấu trúc câu và văn bản riêng biệt. Cuối cùng, sự đa dạng ngôn ngữ cũng đặt ra thách thức về việc xây dựng và quản lý tập dữ liệu đa ngôn ngữ để huấn luyện chatbot [5, 6, 7].

Trong môi trường đa ngôn ngữ, chatbot phải đáp ứng được các yêu cầu và câu hỏi từ người dùng ở nhiều ngôn ngữ khác nhau. Điều này đặt ra một thách thức về việc xây dựng chatbot có khả năng thích ứng ngôn ngữ mạnh mẽ. Mục tiêu của nghiên cứu này là nâng cao khả năng thích ứng ngôn ngữ trong chatbot dựa trên mô hình tiền huấn luyện đa ngôn ngữ [8]. Trong bài báo này, chúng tôi đề xuất một cải tiến trong xây dựng chatbot hỗ trợ đa ngôn ngữ mà không cần phải triển khai nhiều chatbot riêng biệt cho từng ngôn ngữ. Mục tiêu là nhằm tạo ra một chatbot đa ngôn ngữ linh hoạt, có khả năng hiểu và phản hồi đối với người dùng sử dụng các ngôn ngữ khác nhau.

Các đóng góp chính của bài báo bao gồm:

- Đề xuất và triển khai môđun "Phát hiện ngôn ngữ" vào bộ quản lý hội thoại của RASA. Môđun này giúp chatbot nhận biết ngôn ngữ của câu hỏi người dùng một cách tự động và linh hoạt.
- Đề xuất một mô hình triển khai chatbot đa ngôn ngữ trên một hệ thống mã nguồn duy nhất, tiết kiệm thời gian so với việc triển khai từng chatbot cho từng ngôn ngữ riêng lẻ.
- Xây dựng tập dữ liệu huấn luyện đa ngôn ngữ cho chatbot, bao gồm các câu mẫu và hành động phản hồi tương ứng với nhiều ngôn ngữ khác nhau. Điều này đảm bảo rằng mô hình có thể hiểu và xử lý các câu hỏi từ nhiều ngôn ngữ.

Các phần tiếp theo được tổ chức như sau: Mục II đánh giá các nghiên cứu liên quan trong 05 năm trở lại đây, trong đó tập trung vào chatbot đa ngôn ngữ. Trên cơ sở các phân tích, Mục III sẽ mô tả chi tiết phương pháp triển khai chatbot đa ngôn ngữ dựa trên kiến trúc RASA. Môđun "Phát hiện ngôn ngữ" được thêm vào kiến trúc RASA cùng với việc tổ chức lại dữ liệu huấn luyện và dữ liệu phản hồi theo ngôn ngữ nhằm giúp chatbot hiểu và trả lời linh hoạt theo ngôn ngữ của người dùng. Các phân tích và đánh giá về độ chính xác hiểu ngôn ngữ tự nhiên, đánh giá về kịch bản trả lời của chatbot được mô tả ở Mục IV. Cuối cùng, kết luận được trình bày trong Mục V.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Trải qua quá trình phát triển, chatbot đã cho thấy tính hữu ích và tiềm năng lớn trong nhiều lĩnh vực ứng dụng như dịch vụ khách hàng, hỗ trợ ngôn ngữ hay tư vấn giáo dục, ... Tuy nhiên, một trong những thách thức lớn mà chatbot đang phải đối diện là khả năng hoạt động đa ngôn ngữ. Việc triển khai chatbot có thể tương tác với người dùng từ nhiều quốc gia và ngôn ngữ khác nhau đòi hỏi chatbot phải có hiểu biết và phản ứng linh hoạt trên nhiều ngôn ngữ [9]. Sau đây là đánh giá đối với các nghiên cứu liên quan đến chatbot có khả năng hiểu và xử lý đa ngôn ngữ từ năm 2018 đến nay.

Khi xây dựng chatbot, tùy theo nhu cầu mà một ngôn ngữ sẽ được chọn để phát triển cho một chatbot tương ứng. Tuy nhiên, khi nhu cầu cần mở rộng sang ngôn ngữ khác, vấn đề là làm thế nào để tích hợp thêm ngôn ngữ mới vào chatbot. Cách tiếp cận phổ biến là sao chép chatbot từ ngôn ngữ đã xây dựng, dịch dữ liệu của nó sang ngôn ngữ mới và huấn luyện lại. RASA là một nền tảng phát triển chatbot hỗ trợ cách tiếp cận này. Ban đầu, RASA được thiết kế với mục tiêu chính là xây dựng các chatbot hoạt động trong một ngôn ngữ cụ thể nhưng sau đó do nhu cầu mà mở rộng bằng cách sao chép thêm chatbot với ngôn ngữ khác [10]. Với cách tiếp cận này, các chatbot ở từng ngôn ngữ vận hành độc lập, tốn nhiều tài nguyên duy trì cũng như gặp nhiều khó khăn trong việc chỉnh sửa và đồng bộ dữ liệu giữa các ngôn ngữ.

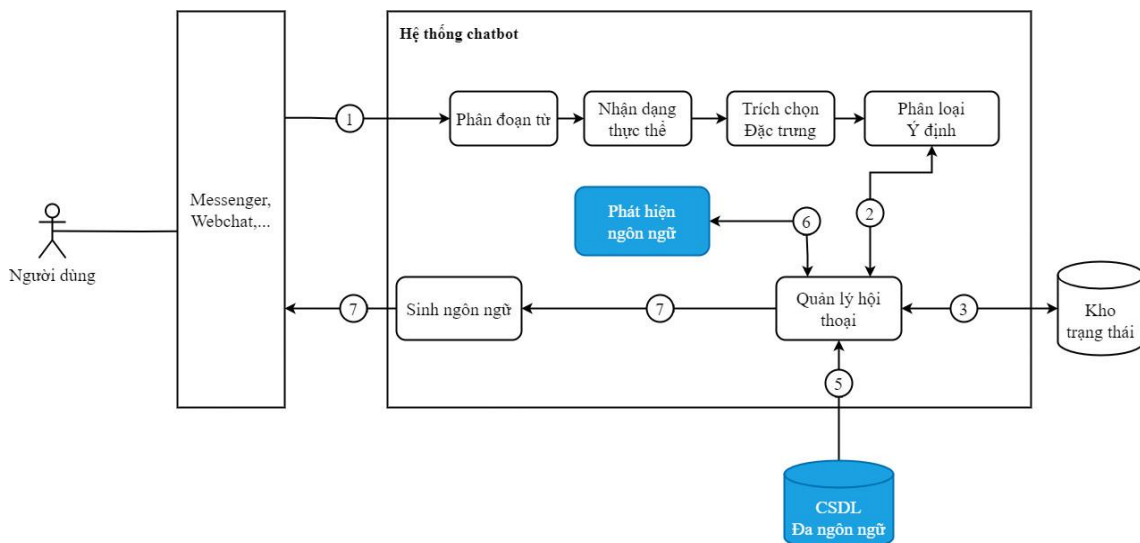
Cách tiếp cận thứ hai là tích hợp mô đun dịch ngôn ngữ vào bên trong chatbot. Cụ thể, Badlani và cs. [11] đã đề xuất xây dựng một chatbot tích hợp mô đun dịch ngôn ngữ mà không cần huấn luyện lại hay duy trì đồng thời nhiều chatbot. Tuy nhiên, ngữ nghĩa của ngôn ngữ phụ thuộc vào mô đun dịch ngôn ngữ, nên trong nhiều lĩnh vực chuyên môn, mô đun dịch ngôn ngữ có thể làm sai lệch ý nghĩa của câu hỏi người dùng cũng như câu trả lời của chatbot.

Hướng tiếp cận thứ ba, chatbot được xây dựng nhằm mục tiêu có thể hiểu và trả lời đa ngôn ngữ [12]. Hướng tiếp cận này gặp nhiều thách thức khi triển khai bởi vì chatbot phải có khả năng xác định ngôn ngữ của câu truy vấn người dùng. Chatbot sử dụng phương pháp NLP đa ngôn ngữ để có thể hiểu câu truy vấn từ người dùng, sau đó sử dụng cơ chế phân loại ngôn ngữ để xác định ngôn ngữ của câu truy vấn. Sau khi xác định được ngôn ngữ, chatbot trả lời bằng cùng ngôn ngữ đó. Theo nghiên cứu của Anshul và cs. [13], hầu hết các chatbot hiện tại là đơn ngữ, trong khi thực tế các dịch vụ người dùng cần đa ngôn ngữ. Rõ ràng có một nhu cầu về kết hợp các ngôn ngữ trong khi tương tác giữa người dùng, cũng như giữa họ với hệ thống chatbot. Các nhà ngôn ngữ học gọi hiện tượng này là trộn mã hoặc chuyển mã. Nghiên cứu trong [13] đã thực hiện với 91 người dùng song ngữ để kiểm tra cách đàm thoại kết hợp với mã hóa và các lựa chọn pha trộn qua nhiều lượt trò chuyện. Kết quả cho thấy người dùng đa ngôn ngữ rất thích chatbot có thể trộn mã hoặc chuyển mã.

Như vậy, việc lựa chọn hướng tiếp cận tùy thuộc vào mức độ phức tạp và tương tác mà nhà phát triển chatbot muốn đạt được. Trong bài báo này, chúng tôi tiếp cận theo hướng thứ ba là đề xuất một cải tiến trong xây dựng chatbot hỗ trợ đa ngôn ngữ (tiếng Anh và tiếng Việt). Cải tiến của chúng tôi có thể được mở rộng thêm ngôn ngữ mới một cách dễ dàng nhưng không cần tạo thêm chatbot mới. Do đó, chatbot này thể hiện được tính thông minh và hiểu nhiều ngôn ngữ để trả lời khách hàng.

III. PHƯƠNG PHÁP XÂY DỰNG CHATBOT THÍCH ỨNG ĐA NGÔN NGỮ

Trong nghiên cứu này, chúng tôi sử dụng kiến trúc RASA [14] như là cơ sở để xây dựng chatbot đa ngôn ngữ và bổ sung thêm mô đun tùy chỉnh nhằm nâng cao khả năng xử lý đa ngôn ngữ của hệ thống (Hình 1). Cải thiện dữ liệu huấn luyện và phát triển mô đun tùy chỉnh “Phát hiện ngôn ngữ” là các điểm cốt lõi để đạt được mục tiêu này.



Hình 1. Mô hình chatbot đa ngôn ngữ được phát triển dựa trên kiến trúc đơn ngôn ngữ của RASA

Môđun “*Phát hiện ngôn ngữ*” được tạo ra để chatbot tự động xác định ngôn ngữ của câu hỏi từ người dùng mà không yêu cầu khai báo loại ngôn ngữ cụ thể. Điều này giúp chatbot dễ dàng xác định ngôn ngữ của câu hỏi và đưa câu hỏi tới môđun hiểu ngôn ngữ tự nhiên (Natural Language Understanding - NLU) tương ứng để xử lý. Việc điều chỉnh tổ chức dữ liệu huấn luyện và dữ liệu phản hồi để hỗ trợ đa ngôn ngữ cũng được chú ý đến. Dữ liệu đa ngôn ngữ thu thập bao gồm các bộ câu hỏi và câu trả lời trong nhiều ngôn ngữ khác nhau, nhằm đảm bảo sự đa dạng và đại diện cho các ngôn ngữ khác nhau. Điều này giúp chatbot học cách xử lý đa ngôn ngữ một cách chính xác và linh hoạt, cung cấp khả năng tương tác đa ngôn ngữ đáng tin cậy.

Như vậy, cải tiến của chúng tôi tập trung vào sử dụng kiến trúc RASA làm cơ sở và bổ sung các thành phần tùy chỉnh để xây dựng chatbot đa ngôn ngữ mà không cần triển khai các chatbot cho riêng từng ngôn ngữ. Việc cải thiện dữ liệu huấn luyện và thêm môđun “*Phát hiện ngôn ngữ*” sẽ giúp tăng cường khả năng xử lý đa ngôn ngữ của hệ thống NLU và quản lý đối thoại một cách hiệu quả.

A. Phát hiện ngôn ngữ

Môđun phát hiện ngôn ngữ “*action_detect_language*” được thiết kế với mục đích phân loại ngôn ngữ của văn bản đầu vào từ người dùng và gán giá trị cho hai biến quan trọng là “*langcode*” (mã ngôn ngữ) và “*langname*” (tên ngôn ngữ). Điều này cho phép chatbot sử dụng thông tin ngôn ngữ để trả lời một cách phù hợp và chính xác hơn. Để đạt được mục tiêu này, chúng tôi sử dụng thư viện Python “*langid*” [15], một công cụ mạnh mẽ để phát hiện ngôn ngữ trong một đoạn văn bản. Thư viện này sử dụng một mô hình học máy được huấn luyện trên một tập dữ liệu lớn với các văn bản có mã ngôn ngữ đã biết trước. Khi nhận được một đoạn văn bản, mô hình sẽ phân loại các đặc trưng của nó và đưa ra dự đoán về mã ngôn ngữ tương ứng. Vì *langid* chỉ cung cấp mã ngôn ngữ, chúng tôi đã bổ sung thư viện “*iso639*” [16] để tìm ra tên ngôn ngữ tương ứng dựa vào mã ngôn ngữ. Điểm mạnh của *langid* nằm ở tốc độ phát hiện ngôn ngữ nhanh chóng và hiệu quả, tiết kiệm tài nguyên tính toán. Tuy nhiên, *langid* chỉ phát hiện được các ngôn ngữ có trong tập dữ liệu huấn luyện, nên có thể dẫn đến kết quả phát hiện không chính xác nếu đoạn văn bản có ngôn ngữ hiếm hoặc không xuất hiện trong tập dữ liệu đã được sử dụng để huấn luyện. Trong cải tiến của chúng tôi, “*action_detect_language*” là một công cụ quan trọng trong việc giúp chatbot xác định và làm việc với nhiều ngôn ngữ khác nhau. Tuy nhiên, việc sử dụng *langid* với dữ liệu đầu vào có quá ít từ (thường ít hơn 3) có thể làm cho việc xác định ngôn ngữ không chính xác.

B. Dữ liệu đa ngôn ngữ

Chúng tôi đề xuất thay đổi cấu trúc dữ liệu sử dụng trong RASA để giúp chatbot hiểu và trả lời được nhiều ngôn ngữ. Các thay đổi này tập trung vào việc chuẩn bị dữ liệu huấn luyện và cung cấp thông tin ngôn ngữ cho chatbot để nó có khả năng làm việc với nhiều ngôn ngữ khác nhau. Dưới đây là chi tiết về các thay đổi đối với 2 ngôn ngữ tiếng Anh và tiếng Việt.

Tập tin *Nlu.yml* chứa dữ liệu huấn luyện cho môđun NLU của chatbot [17]. *Nlu.yml* bao gồm các mẫu câu huấn luyện cho các ý định của chatbot và được viết bằng hai ngôn ngữ là tiếng Anh và tiếng Việt. Mỗi ý định có một danh sách các câu mẫu tương ứng để mô tả các cách khác nhau mà người dùng có thể hỏi về cùng một ý định, như ví dụ trong Bảng 1.

Bảng 1. Tập tin *nlu.yml* mở rộng với tiếng Anh và tiếng Việt

| Tiếng Anh | Tiếng Anh và tiếng Việt |
|--|---|
| - intent: faqlatecheckin examples: - is it possible for late check in? - late check-in - will you allow for the late check in? | - intent: faqlatecheckin examples: - is it possible for late check in? - Có thể nhận phòng muộn không? - late check-in - Nhận phòng muộn - will you allow for the late check in? - Bạn cho phép nhận phòng muộn không? |

Cấu trúc dữ liệu này giúp cho việc bổ sung dễ dàng thêm ngôn ngữ mới sau mỗi ý định (intent). Khi muốn hỗ trợ thêm ngôn ngữ khác, chỉ cần thêm câu mẫu tương ứng vào danh sách mẫu của ý định đó trong tập tin *Nlu.yml*.

Trong tập tin *stories.yml* [17], “*action_detect_language*” được thêm vào mỗi bước trong hội thoại mẫu để phát hiện ngôn ngữ của từng câu hỏi và đáp án trong chatbot.

Như được chỉ ra trong Bảng 2, phân tích cụ thể cho mỗi bước là như sau:

Bước 1: Intent: greet (người dùng chào hỏi).

Action: *action_detect_language* (hành động để phát hiện ngôn ngữ của câu hỏi).

Bước 2: Action: *utter_greet* (chatbot phản hồi chào hỏi).

Bước 3: Intent: *book_rooms* (người dùng yêu cầu đặt phòng).

Action: action_detect_language (hành động để phát hiện ngôn ngữ của câu hỏi).

Bước 4: Action: action_book_rooms (chatbot xử lý yêu cầu đặt phòng).

Bước 5: Intent: dulex_room_details (người dùng yêu cầu thông tin về loại phòng duplex).

Action: action_detect_language (hành động để phát hiện ngôn ngữ của câu hỏi).

Bước 6: Action: utter_dulexroom_details (chatbot cung cấp thông tin về loại phòng duplex).

Với việc thêm "action_detect_language" vào mỗi bước trong hội thoại mẫu, chatbot sẽ nhận biết ngôn ngữ của từng câu hỏi và đáp án một cách tự động. Điều này giúp chatbot có thể xử lý và trả lời chính xác cho người dùng từ nhiều ngôn ngữ khác nhau, tăng cường khả năng đa ngôn ngữ của chatbot và cải thiện trải nghiệm người dùng.

Bảng 2. Bổ sung thêm action_detect_language vào stories.yml

| Tiếng Anh | Tiếng Anh và tiếng Việt |
|--|--|
| - story: book_room_juniorsuite_details steps: - intent: greet - action: utter_greet - intent: book_rooms - action: action_book_rooms - intent: junior_suite_details - action: utter_juniorsuite_details | - story: book_room_juniorsuite_details steps: - intent: greet - action: action_detect_language - action: utter_greet - intent: book_rooms - action: action_detect_language - action: action_book_rooms - intent: junior_suite_details - action: action_detect_language - action: utter_juniorsuite_details |

Tập tin *domain.yml* [17] chứa thông tin về domain (lĩnh vực) của chatbot và các cấu trúc dữ liệu đi kèm. Trong trường hợp này, chúng ta thêm điều kiện mã ngôn ngữ cho mỗi mẫu trả lời. Điều này giúp cho chatbot có thể hiểu ngôn ngữ của người dùng và phản hồi lại bằng ngôn ngữ tương ứng (Bảng 3).

Bảng 3. Mở rộng tập tin domain.yml

| Tiếng Anh | Tiếng Anh và tiếng Việt |
|---|--|
| utter_ask_room: text: "How many rooms would you like to book?" buttons: - payload: "1" title: "1" - payload: "2" title: "2" - payload: "3" title: "3" | utter_ask_room: - condition: - type: slot name: langcode value: en text: "How many rooms would you like to book?" buttons: - payload: "1" title: "1" - payload: "2" title: "2" - payload: "3" title: "3" - condition: - type: slot name: langcode value: vi text: "Bạn muốn đặt bao nhiêu phòng?" buttons: - payload: "1" title: "1" - payload: "2" title: "2" - payload: "3" title: "3" - text: "I'm sorry, I don't recognize the language you're using. Currently, I can only provide assistance in English and Vietnamese. If you're comfortable with either of these languages, please feel free to continue. If not, please try to rephrase your question or provide more context, and I'll do my best to assist you." |

Sau khi hoàn thành quá trình tinh chỉnh (*fine-tuning*) mô hình, chúng tôi thu được mô hình ngôn ngữ đa ngôn ngữ đã được tùy chỉnh dành riêng cho chatbot. Dựa trên dữ liệu huấn luyện từ tập tin *nlu.yml*, mô hình đã học cách hiểu các yêu cầu và trích xuất thông tin về khách sạn từ người dùng trong cả hai ngôn ngữ tiếng Anh và tiếng Việt. Ma trận nhầm lẫn ý định (Intent Confusion Matrix) là công cụ quan trọng để đánh giá hiệu suất của khả năng hiểu ngôn ngữ của chatbot. Ma trận này thể hiện mức độ chính xác của việc dự đoán các ý định từ câu hỏi người dùng.

Như mô tả trong Hình 2, hàng thứ *i* và cột thứ *j* đại diện cho ý định *i* và ý định *j*. Giá trị tại vị trí (*i, j*) cho biết số lượng câu hỏi có ý định *i* được dự đoán là ý định *j*. Ví dụ, giá trị tại ô (*book_room, book_room*) là 40, có nghĩa là có 40 câu hỏi có ý định *book_room* đã được dự đoán đúng là ý định *book_room*. Giá trị tại ô (*dulux_room_details, club_suite_details*) là 1, có nghĩa là có 1 câu hỏi có ý định *dulux_room_details* đã bị dự đoán sai thành ý định *club_suite_details*.

Bảng 4. So sánh kết quả dự đoán của mô hình với ý định thực tế

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---------------|---------------|---------------|---------------|
| bert-base-multilingual-cased (Tiếng Anh và Tiếng Việt) | 0,9884 | 0,9897 | 0,9878 | 0,9883 |
| bert-base-multilingual-cased (Tiếng Anh) | 0,8563 | 0,8590 | 0,8595 | 0,8563 |
| bert-base-multilingual-cased (Tiếng Việt) | 0,8537 | 0,8550 | 0,8531 | 0,8516 |

Để đánh giá ý định, cần xác định xem mô hình đã phân loại ý định từ người dùng đúng hay sai và đo lường độ chính xác của việc phân loại này. Dựa trên tập dữ liệu kiểm tra (test data) có các câu mẫu chứa các ý định khác nhau, Bảng 4 so sánh kết quả dự đoán của mô hình với ý định thực tế của từng câu mẫu. Khi kết hợp cả tiếng Anh và tiếng Việt trong dữ liệu huấn luyện, mô hình có cơ hội tiếp xúc với nhiều ngôn ngữ và cấu trúc câu khác nhau. Điều này giúp mô hình học được đặc điểm và quy luật phân loại ý định trong nhiều tình huống khác nhau, từ đó cải thiện khả năng dự đoán. Thêm vào đó, thông tin từ tiếng Anh và tiếng Việt có thể bổ sung cho nhau, đặc biệt khi có sự tương đồng trong các ý định giữa hai ngôn ngữ. Mô hình có thể học từ dữ liệu tiếng Anh để cải thiện khả năng dự đoán ở dữ liệu tiếng Việt và ngược lại. Hơn nữa, việc kết hợp nhiều ngôn ngữ trong dữ liệu huấn luyện giúp mô hình tránh overfitting hiệu quả hơn bằng cách tìm hiểu các biểu đồ phân phối của nhiều ngôn ngữ khác nhau.

Sử dụng các độ đo như độ chính xác (accuracy), độ phủ (recall), độ chính xác dương tính (precision) và F1-score để đánh giá hiệu suất của mô hình trong việc phân loại ý định. Mô hình ngôn ngữ *bert-base-multilingual-cased* đã cho thấy hiệu suất tốt trong việc phát hiện và hiểu ý định từ người dùng trong cả hai ngôn ngữ. Điều này cho thấy tính linh hoạt và khả năng đa ngôn ngữ của mô hình, cho phép chatbot hỗ trợ người dùng từ nhiều ngôn ngữ khác nhau. Thời gian tiền huấn luyện mô hình đã được kiểm tra và xác định là tương đối nhanh. Kết quả dự kiến cho mô hình đã được tích hợp vào chatbot là tích cực và đáp ứng tốt nhu cầu của người dùng. Chatbot có khả năng hiểu và phản hồi với độ chính xác cao với các yêu cầu đặt phòng và cung cấp thông tin về khách sạn trong cả hai ngôn ngữ. Quá trình tinh chỉnh mô hình tiền huấn luyện đa ngôn ngữ cho chatbot khách sạn đã đạt được những kết quả tích cực. Hiệu suất và tính linh hoạt của mô hình giúp chatbot hỗ trợ thông tin khách sạn một cách hiệu quả và tương tác với người dùng bằng nhiều ngôn ngữ, nâng cao trải nghiệm người dùng và giúp cung cấp dịch vụ tốt hơn trong lĩnh vực du lịch và khách sạn.

Sau khi tinh chỉnh mô hình và huấn luyện các luồng hội thoại, chúng tôi đánh giá hiệu suất của mô hình trong việc xử lý các tình huống thực tế (stories). Điều này thường được thực hiện bằng cách sử dụng tập dữ liệu kiểm tra gồm các story đã được chuẩn bị trước đó. Mỗi tình huống thực tế (story) sẽ bao gồm một loạt các ý định và hành động được dự định từ người dùng và mô hình sẽ phản hồi tương ứng. Đánh giá story sẽ giúp xác định xem mô hình có thể đáp ứng chính xác các yêu cầu và thông tin từ người dùng hay không. Nếu có bất kỳ sai sót hoặc không khớp nào trong quá trình xử lý, cần điều chỉnh và cải thiện mô hình và luồng hội thoại để đảm bảo chatbot hoạt động chính xác và hiệu quả.

Ma trận nhầm lẫn hành động (action confusion matrix) thường được sử dụng để phân tích sự phân phối, so khớp giữa các hành động thực tế và các hành động được dự đoán bởi chatbot. Việc theo dõi các hành động này giúp cải thiện và tối ưu hóa tương tác giữa chatbot và người dùng. Như được mô tả trong Hình 3, *action_listen* thể hiện rằng chatbot đang ở trạng thái chờ đợi đầu vào tiếp theo từ người dùng, không có hành động cụ thể nào được thực hiện. Ô (*action_listen, action_listen*) có giá trị 94 cho thấy rằng hành động này đã được dự đoán đúng 94 lần trong dữ liệu kiểm tra hoặc dữ liệu đánh giá. Với hành động phát hiện ngôn ngữ *action_detect_language*, ô (*action_detect_language, action_detect_language*) có giá trị 95 thể hiện rằng hành động *action_detect_language* đã được thực hiện 95 lần sau khi người dùng nhập văn bản và trước khi hành động tiếp theo của chatbot nhằm giúp chatbot nhận ra được ngôn ngữ của người dùng đang sử dụng. Hình 3 cũng cho thấy chatbot dự đoán hành động không có sai lệch, giá trị không nằm trên đường chéo chính đều có giá trị 0.

- [4] M. Y. Murata, M. de Souza Monteiro, and L. C. de Castro Salgado, "Omotenashi: A Study about the Cultural Adaptation of an Intelligent Agent on Blip," 2021, DOI: 10.1145/3472301.3484372.
- [5] M. Marge *et al.*, "Spoken language interaction with robots: Recommendations for future research," *Comput. Speech Lang.*, vol. 71, p. 101255, Jan. 2022, DOI: 10.1016/J.CSL.2021.101255.
- [6] L. Coheur, "From Eliza to Siri and Beyond," *Commun. Comput. Inf. Sci.*, vol. 1237 CCIS, pp. 29-41, 2020, DOI: 10.1007/978-3-030-50146-4_3/FIGURES/1.
- [7] P. Cavalin, V. H. A. Ribeiro, M. Vasconcelos, C. Pinhanez, J. Nogima, and H. Ferreira, "Towards a Method to Classify Language Style for Enhancing Conversational Systems," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2021-July, Jul. 2021, DOI: 10.1109/IJCNN52387.2021.9534090.
- [8] J. Stanley, R. ten Brink, A. Valiton, T. Bostic, and R. Scollan, "Chatbot Accessibility Guidance: A Review and Way Forward," in *Proceedings of Sixth International Congress on Information and Communication Technology*, 2022, pp. 919-942.
- [9] A. Følstad and C. Taylor, "Investigating the user experience of customer service chatbot interaction: a framework for qualitative analysis of chatbot dialogues," *Qual. User Exp.*, vol. 6, no. 1, p. 6, 2021, DOI: 10.1007/s41233-021-00046-5.
- [10] "Multilingual DemoBot | Rasa Community Showcase | Rasa." <https://rasa.community/showcase/multilingual-demobot/> (accessed Jul. 30, 2023).
- [11] S. Badlani, T. Aditya, M. Dave, and S. Chaudhari, "Multilingual Healthcare Chatbot Using Machine Learning," in *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1-6, DOI: 10.1109/INCET51464.2021.9456304.
- [12] V. Kasinathan, A. Mustapha, and C. Khai Bin, "A Customizable Multilingual Chatbot System for Customer Support," *Annals of Emerging Technologies in Computing (AETiC)*, *Print A Customizable Multilingual Chatbot System for Customer Support*, vol. 5, no. 5, pp. 51-59, 2021, DOI: 10.33166/AETiC.2021.05.006.
- [13] A. Bawa, P. Khadpe, P. Joshi, K. Bali, and M. Choudhury, "Do Multilingual Users Prefer Chat-Bots That Code-Mix? Let's Nudge and Find Out!," *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. CSCW1, May 2020, DOI: 10.1145/3392846.
- [14] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," *arXiv Prepr. arXiv1712.05181*, 2017.
- [15] "langid · PyPI." <https://pypi.org/project/langid/> (accessed Aug. 01, 2023).
- [16] "iso-639 · PyPI." <https://pypi.org/project/iso-639/> (accessed Aug. 03, 2023).
- [17] M. Arevalillo-Herraez, P. Arnau-Gonzalez, and N. Ramzan, "On Adapting the DIET Architecture and the Rasa Conversational Toolkit for the Sentiment Analysis Task," *IEEE Access*, vol. 10, no. September, pp. 107477-107487, 2022, DOI: 10.1109/ACCESS.2022.3213061.
- [18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "{BERT}: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019, pp. 4171-4186, DOI: 10.18653/v1/N19-1423.

AN IMPROVEMENT OF THE ADAPTABILITY FOR MULTILINGUAL CHATBOTS

Ngô Văn Sơn, Thái Thị Phương, Võ Việt Minh Nhật

ABSTRACT: The paper presents an improvement of the adaptability for multilingual chatbot based on RASA architecture and BERT pre-training model. The goal is to build a chatbot capable of understanding and responding in multiple languages having to deploy separate chatbots for each language. Our approach focuses on two main aspects. First, a language detection module is included in RASA's dialogue manager so that the chatbot can handle multiple languages. This module helps the chatbot automatically recognize the language from the user's question without specifying the language. Second, the chatbot's training and response data are tailored to the multiple languages that need to be supported. The implementation results show that the chatbot can understand many languages flexibly. Our improvement of the adaptability for multilingual chatbot therefore can be widely applied in practice and meet the needs of multilingual communication.