

**XÂY DỰNG ỨNG DỤNG CHATBOT
HỖ TRỢ TƯ VẤN ĐĂNG KÝ VÀ TỔ CHỨC THI TIẾNG ANH
THEO KHUNG NĂNG LỰC NGOẠI NGỮ 6 BẬC DÙNG CHO VIỆT NAM
(VSTEP)**

Phạm Trọng Nguyễn

Trường Đại học Sài Gòn

Email: nguyen.pt@sgu.edu.vn

Ngày nhận bài: 20/8/2024; ngày hoàn thành phần biện: 27/9/2024; ngày duyệt đăng: 01/11/2024

TÓM TẮT

Chatbot là một chương trình máy tính được thiết kế để mô phỏng cuộc trò chuyện với người dùng qua các nền tảng nhắn tin, trang web, hoặc ứng dụng di động. Ứng dụng chatbot trong tư vấn tuyển sinh có thể mang lại lợi ích đáng kể cho các trường học và tổ chức giáo dục. Trong bài báo này, chúng tôi xây dựng ứng dụng chatbot nhằm hỗ trợ công tác tuyển sinh và tổ chức thi tiếng Anh theo Khung năng lực ngoại ngữ 6 bậc dùng cho Việt Nam tại Trường Đại học Sài Gòn. Chatbot được xây dựng dựa trên Rasa, nền tảng mã nguồn mở mạnh mẽ cho phép phát triển các chatbot và trợ lý ảo thông minh, và được kết nối với Zalo, nền tảng nhắn tin phổ biến nhất tại Việt Nam. Bằng cách thu thập và huấn luyện mô hình dựa trên các câu hỏi thường gặp của thí sinh, ứng dụng của chúng tôi có thể nhận diện hơn 50 loại câu hỏi khác nhau với độ chính xác của mô hình hiểu ngôn ngữ tự nhiên là khá cao.

Từ khóa: trợ lý ảo, chatbot, tư vấn tuyển sinh, Rasa.

1. GIỚI THIỆU

Trợ lý ảo (virtual assistant) hay chatbot là một chương trình máy tính có thể tương tác người dùng bằng ngôn ngữ tự nhiên [1]. Chatbot có thể hoạt động theo các kịch bản và câu trả lời được lập trình sẵn, phù hợp với các tác vụ đơn giản như trả lời câu hỏi thường gặp hoặc có thể sử dụng các thuật toán máy học và xử lý ngôn ngữ tự nhiên, giúp hiểu sâu hơn về ngữ cảnh và trả lời linh hoạt hơn. Ngày nay, chatbot được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau như dịch vụ khách hàng, hỗ trợ kỹ thuật, thương mại điện tử, và giáo dục [2]. Trong lĩnh vực giáo dục, ứng dụng chatbot trong tư vấn tuyển sinh có thể mang lại nhiều lợi ích đáng kể cho các trường học và tổ chức giáo dục. Nó có thể cung cấp thông tin về các khóa học, quy trình tuyển sinh và học phí, giúp giảm tải công việc cho nhân viên tư vấn.

Trên thế giới, chatbot hỗ trợ tuyển sinh (admission chatbot) ngày càng trở nên phổ biến. Đại học Kỹ thuật VR Siddhartha đã phát triển một ứng dụng chatbot dựa trên nền tảng BotPress nhằm hỗ trợ học sinh lớp 12 trong quá trình đăng ký nhập học [3]. Đại học Quốc gia về Khoa học và Công nghệ Pakistan đã xây dựng chatbot hỗ trợ tuyển sinh dựa trên nền tảng DialogFlow [4]. Đại học Thành phố Manila đã phát triển một ứng dụng web chatbot trên nền tảng Microsoft Azure Bot Service có thể giúp trường giải đáp một cách hiệu quả những mối quan tâm của người nộp đơn [5]. Tương tự, Đại học NMIMS Mumbai cũng sử dụng nền tảng Microsoft Azure Bot Service cho ứng dụng chatbot hỗ trợ tuyển sinh của mình [6]. Nhằm hỗ trợ tư vấn sinh viên trong tuyển sinh cao học, Đại học Princess Nourah đã phát triển một ứng dụng chatbot dựa trên nền tảng Botsify [7]. Tại Việt Nam, “NEU-chatbot” đã được triển khai để hỗ trợ quá trình tuyển sinh và đã hoạt động trên trang web “Tư vấn tuyển sinh đại học chính quy” của Trường Đại học Kinh tế Quốc dân [8].

Trường Đại học Sài Gòn được Bộ Giáo dục và Đào tạo giao nhiệm vụ tổ chức thi đánh giá năng lực tiếng Anh theo khung năng lực ngoại ngữ 6 bậc dùng cho Việt Nam (VSTEP). Trong giai đoạn từ tháng 02/2020 đến nay số lượng thí sinh đăng ký dự thi ngày càng tăng, với hơn 2,600 lượt thí sinh đăng ký dự thi hàng năm, nhu cầu tư vấn của thí sinh ngày càng nhiều. Phần lớn thí sinh muốn được tư vấn từ xa trước khi đến nộp hồ sơ dự thi, do đó việc nhân viên tư vấn đáp ứng tất cả nhu cầu tìm hiểu thông tin của thí sinh gặp nhiều áp lực. Cùng với xu hướng chuyển đổi số, áp dụng công nghệ vào thực tế công việc tại đơn vị, trên cơ sở nhu cầu của thí sinh, việc xây dựng chatbot hỗ trợ giúp thí sinh có thể tiếp cận dịch vụ tư vấn mọi lúc, mọi nơi là nhu cầu thực tế và cấp thiết. Với ưu điểm phục vụ liên tục không phụ thuộc vào thời gian, địa điểm và đáp ứng nhu cầu nhiều thí sinh cùng lúc, ứng dụng chatbot có thể giúp công tác tư vấn và hỗ trợ tuyển sinh giảm phụ thuộc vào nhân viên tư vấn, giảm chi phí phát sinh trong quá trình hoạt động. Vì vậy, trong bài báo này chúng tôi xây dựng một ứng dụng chatbot hỗ trợ công tác tuyển sinh và tổ chức thi tiếng Anh VSTEP tại Trường Đại học Sài Gòn.

2. NỀN TẢNG PHÁT TRIỂN CHATBOT

Có nhiều nền tảng phát triển chatbot được phát triển bởi các cộng đồng, nhóm phát triển khác nhau. Những nền tảng này có sự khác biệt về ngôn ngữ lập trình, mô hình dữ liệu, mô hình huấn luyện cũng như luồng giao tiếp. Tuy nhiên, nhìn chung, cấu trúc của một chatbot gồm hai thành phần chính: (1) Phân tích thông điệp người dùng; (2) Tạo ra câu trả lời tương ứng với thông điệp đầu vào. Việc lựa chọn nền tảng phát triển chatbot khá quan trọng nhằm đảm bảo độ chính xác trong việc phân tích thông điệp của người dùng và tạo câu trả lời phù hợp. Chúng tôi so sánh 5 nền tảng phát triển chatbot phổ biến bao gồm Google Dialogflow [9], Wit.ai [10], IBM Watson [11], Microsoft Azure Bot Service [12], và Rasa [13]. Sự khác biệt cơ bản của các nền tảng dựa trên các

tiêu chí khác nhau được trình bày trong Bảng 1. Cả 5 nền tảng đều hỗ trợ tốt về Khả năng tích hợp (chatbot có thể được tích hợp vào các ứng dụng tin nhắn như Facebook Messenger, WhatsApp, Telegram, hoặc trên các trang web và ứng dụng di động) cũng như Phân loại ý định (xác định ý định của người dùng khi tương tác với hệ thống thông qua câu hỏi). Về mặt chi phí, IBM Watson và Microsoft Azure Bot Service là hai dịch vụ có phí được tính dựa trên khối lượng dữ liệu của người dùng; Dialogflow và Wit.ai miễn phí nhưng là mã nguồn đóng; Rasa miễn phí và là mã nguồn mở. So với các nền tảng còn lại, Dialogflow và Rasa có cộng đồng hỗ trợ lớn hơn cũng như dễ sử dụng hơn. Nói tóm lại, trong 5 nền tảng phát triển được nghiên cứu, Rasa đạt được sự cân bằng giữa các tiêu chí đánh giá.

Bảng 1

Nền tảng	Khả năng tích hợp	Phân loại ý định	Chi phí	Hỗ trợ từ cộng đồng	Dễ sử dụng
Google Dialogflow	✓	✓	Miễn phí	Cao	Dễ
Wit.ai	✓	✓	Miễn phí	Trung bình	Khó
IBM Watson	✓	✓	Dựa trên dữ liệu	Trung bình	Khó
Microsoft Azure Bot Service	✓	✓	Dựa trên dữ liệu	Trung bình	Khó
Rasa	✓	✓	Miễn phí (Nguồn mở)	Cao	Dễ

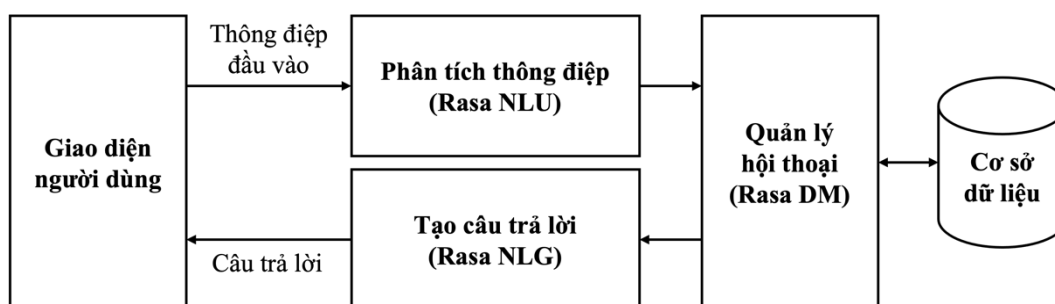
So sánh các nền tảng phát triển chatbot

Việc xây dựng chatbot dựa trên nền tảng sẵn có giúp tiết kiệm thời gian và kinh phí. Trong bài báo này, chúng tôi chọn sử dụng nền tảng Rasa [13] để xây dựng ứng dụng chatbot vì một số lý do như sau. Thứ nhất, Rasa là một nền tảng phát triển chatbot nguồn mở; dễ tích hợp và tùy chỉnh các thành phần; hỗ trợ kết nối với các ứng dụng nhắn tin khác và triển khai trong nhiều môi trường. Thứ hai, Rasa hỗ trợ tích hợp các mô hình máy học (machine learning) trong quản lý đối thoại, giúp xây dựng các mô hình xử lý ngôn ngữ tự nhiên (natural language processing) một cách hiệu quả, ngay cả với dữ liệu huấn luyện ban đầu tối thiểu. Cuối cùng, Rasa là một thư viện lập trình Python, là lựa chọn phù hợp cho các dự án bao gồm các yêu cầu xử lý ngôn ngữ tự nhiên vì cú pháp đơn giản và ngữ nghĩa minh bạch của nó.

3. CÁCH TIẾP CẬN XÂY DỰNG CHATBOT

Dựa trên nền tảng Rasa, chúng tôi xây dựng chatbot bao gồm các thành phần chính được trình bày trong Hình 1. Hệ thống gồm 5 thành phần chính: (1) **Giao diện người dùng** được triển khai trên nền tảng nhắn tin Zalo dùng để nhận thông điệp từ người dùng và hiển thị phản hồi của hệ thống; (2) **Phân tích thông điệp (Rasa NLU)** dùng để xác định chủ đề, trích xuất thông tin từ thông điệp đầu vào của người dùng; (3) **Tạo câu trả lời (Rasa NLG)** có trách nhiệm đưa ra câu trả lời tương ứng với thông điệp từ người dùng; (4) **Quản lý hội thoại (Rasa DM)** quyết định cách chatbot phản hồi và điều phối cuộc trò chuyện; (5) **Cơ sở dữ liệu** dùng để truy vấn thông tin cần thiết để tạo ra câu trả lời cho người dùng.

Hình 1



Khung ứng dụng chatbot dựa trên Rasa

3.1 Giao diện người dùng

Rasa cung cấp khả năng kết nối với nhiều nền tảng tin nhắn khác nhau như Facebook Messenger, Slack, Telegram,... Với sự phổ biến rộng rãi của Zalo tại Việt Nam, chúng tôi quyết định sử dụng Zalo thay vì các nền tảng tin nhắn khác. Zalo là một nền tảng nhắn tin và mạng xã hội được phát triển bởi công ty VNG Corporation, một trong những tập đoàn công nghệ lớn tại Việt Nam. Ra mắt vào năm 2012, Zalo nhanh chóng trở thành ứng dụng nhắn tin phổ biến nhất tại Việt Nam, với hàng triệu người dùng sử dụng hàng ngày. Zalo cung cấp Zalo Official Account (OA), cho phép các doanh nghiệp, tổ chức, và cơ quan chính phủ tương tác trực tiếp với khách hàng thông qua Zalo. Các doanh nghiệp có thể sử dụng OA để cung cấp dịch vụ khách hàng, quảng cáo sản phẩm, và gửi thông báo. Kết nối Rasa với Zalo đòi hỏi phải tùy chỉnh nhiều do Zalo không có hỗ trợ chính thức từ Rasa. Đầu tiên, chúng tôi tạo một ứng dụng trên Zalo Developer, lấy App ID và Secret Key. Sau đó, cấu hình webhook trên Zalo bằng cách cung cấp URL mà Rasa sẽ lắng nghe, đồng thời xác thực webhook này. Tiếp theo, chúng tôi tạo một custom connector trong Rasa bằng cách viết một lớp Python mới, kế thừa từ InputChannel, để xử lý các sự kiện từ Zalo và chuyển chúng đến Rasa. Sau đó, cấu hình kết nối Zalo trong tập tin credentials.yml của Rasa.

3.2 Phân tích thông điệp (Rasa NLU)

Rasa NLU (Natural Language Understanding) giúp hệ thống có thể hiểu được yêu cầu của người dùng, bao gồm 2 chức năng: Phân loại ý định (Intent classification) và Trích xuất thực thể (Entity extraction). Phân loại ý định giúp nhận diện ý định của người dùng từ văn bản nhập vào. Ví dụ, phân biệt được liệu người dùng muốn hỏi về thủ tục đăng ký thi VSTEP, hỏi về lệ phí thi hay hỏi về kết quả thi. Trích xuất thực thể có nhiệm vụ trích xuất các thông tin ngữ cảnh thông qua các thực thể (entities) của thông điệp người dùng. Ví dụ, trong câu “Tôi muốn biết kết quả thi của Trần Minh Anh vào ngày thi 20/6/2024”, các thực thể như họ tên thí sinh (“Trần Minh Anh”) và ngày thi (“20/6/2024”) sẽ được nhận diện dùng để truy vấn thông tin trong các bước sau.

3.2.1 Cấu hình NLU pipeline

Trong Rasa, các văn bản đầu vào được xử lý bởi một chuỗi các thành phần được thực thi lần lượt theo luồng (pipeline) được định nghĩa trong tập tin config.yml. Pipeline được chúng tôi sử dụng trong chatbot được mô tả trong Hình 2. Đầu tiên, thành phần Preprocessor đã được tùy chỉnh nhằm thực hiện các bước tiền xử lý như làm sạch văn bản hay sửa lỗi chính tả (các bước tiền xử lý văn bản được mô tả rõ hơn trong phần 3.2.2). Sau đó, thành phần SpacyTokenizer sử dụng thư viện SpaCy để thực hiện việc tách từ (tokenization) trong quá trình xử lý ngôn ngữ tự nhiên. Tokenization là bước phân chia một câu hoặc đoạn văn bản thành các đơn vị nhỏ hơn, gọi là token, thường là các từ hoặc dấu câu. Thành phần RegexFeaturizer sử dụng các biểu thức chính quy (regex) để tạo ra các đặc trưng (features) từ văn bản. Điều này rất hữu ích để phát hiện các mẫu ngôn ngữ đặc biệt như số điện thoại, email, mã số, v.v. Thành phần EntitySynonymMapper ánh xạ các thực thể tương đương về một giá trị chuẩn. Ví dụ, “VSTEP bậc 3” và “VSTEP B1” có thể được ánh xạ thành một thực thể duy nhất. Đây là một bước quan trọng trong việc xử lý thực thể để đảm bảo tính nhất quán trong dữ liệu. Tiếp theo, thành phần CountVectorsFeaturizer chuyển văn bản thành các vector đặc trưng sử dụng phương pháp “Count Vectors” (tần suất xuất hiện của các từ trong văn bản). Vector này sẽ là đầu vào cho các mô hình học máy sau đó. Thành phần LanguageModelFeaturizer sẽ sử dụng mô hình ngôn ngữ (language model) để tạo ra các đặc trưng dạng dense (đặc trưng dày đặc). Ở đây, mô hình được sử dụng là BERT với trọng số từ rasa/LaBSE. LaBSE là một mô hình ngôn ngữ mạnh mẽ được sử dụng để tạo ra các biểu diễn vector cho văn bản. Thành phần chính trong pipeline là DIETClassifier [14], một mô hình đa tác vụ (multi-task learning) trong Rasa, có thể thực hiện đồng thời việc phân loại ý định (intent classification) và nhận diện thực thể (entity recognition). Mô hình được huấn luyện qua 100 vòng với việc hạn chế tương tự giữa các vector ý định để cải thiện độ chính xác. Thành phần tiếp theo RegexEntityExtractor sử dụng regex để trích xuất thực thể từ văn bản, trong đó các tùy chọn như use_lookup_tables, use_regexes, và use_word_boundaries cho phép kiểm soát cách thức trích xuất thực thể dựa trên các

bảng tra cứu, biểu thức chính quy và ranh giới từ (các bước tiền xử lý văn bản được mô tả rõ hơn trong phần 3.2.3). Thành phần FallbackClassifier đảm bảo rằng hệ thống sẽ đưa ra quyết định dự phòng nếu độ tin cậy của dự đoán thấp hơn ngưỡng 0.7, hoặc khi có sự mơ hồ giữa các ý định. Cuối cùng, thành phần ResponseSelector chọn phản hồi từ các câu trả lời có sẵn, cũng được huấn luyện qua 100 vòng, đảm bảo rằng câu trả lời phù hợp nhất được chọn dựa trên ý định của người dùng.

Hình 2

```
language: vi

pipeline:
- name: pipelines.Preprocessor.Preprocessor
- name: pipelines.Spacy.SpacyTokenizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: EntitySynonymMapper
- name: CountVectorsFeaturizer
- name: LanguageModelFeaturizer
  model_name: bert
  model_weights: rasa/LaBSE
- name: DIETClassifier
  epochs: 100
  constrain_similarities: true
- name: RegexEntityExtractor
  use_lookup_tables: False
  use_regexes: True
  use_word_boundaries: True
- name: EntitySynonymMapper
- name: FallbackClassifier
  threshold: 0.7
  ambiguity_threshold: 0.1
- name: ResponseSelector
  epochs: 100
  constrain_similarities: true
```

Rasa core pipeline

3.2.2 Tiền xử lý văn bản

Tiền xử lý văn bản có ảnh hưởng đáng kể đến độ chính xác của mô hình phân tích thông điệp (NLU). Trong bài báo này, chúng tôi áp dụng một số kỹ thuật tiền xử lý văn bản bao gồm làm sạch văn bản (clean text) và sửa lỗi chính tả (spell correction). Mục đích của clean text là loại bỏ nhiễu khỏi dữ liệu, có nghĩa là loại bỏ các ký hiệu vô nghĩa ra khỏi văn bản đầu vào. Cách đơn giản nhất để giải quyết vấn đề này là sử dụng bộ lọc dựa trên biểu thức chính quy (regular expressions). Spell correction nhằm mục đích sửa lỗi chính tả tiếng Việt, cũng như sửa các từ viết tắt được sử dụng phổ biến trên mạng như “ko, dc, trc” thay cho “không, được, trước”. Chúng tôi đã xây dựng một từ điển chứa các từ sai lỗi chính tả hoặc từ viết tắt và từ đúng tương ứng, bất kỳ từ sai nào có trong văn bản đầu vào sẽ được thay thế. Từ điển các từ sai lỗi chính tả hoặc từ viết tắt được thu thập từ dữ liệu văn bản đầu vào của người dùng. Trong quá trình vận hành, nếu có từ sai lỗi chính tả hoặc từ viết tắt mới được phát hiện sẽ được cập nhật vào từ điển. Việc xây dựng từ điển được thực hiện thủ công và có tham khảo ý kiến của chuyên gia nên có thể đảm bảo về độ chính xác.

3.2.3 Trích xuất thực thể

Trích xuất thực thể (**entity**) từ thông điệp của người dùng là một phần quan trọng của NLU. **Entity** đại diện cho các thông tin cụ thể mà chatbot cần nắm bắt từ cuộc trò chuyện để có thể thực hiện những tác vụ đặc biệt như tra cứu lịch thi, kết quả thi của một thí sinh cụ thể. Trong các tác vụ này, các thực thể quan trọng cần trích xuất là thông tin số báo danh dự thi, cụ thể là số Căn cước công dân (CCCD) hoặc số Chứng minh nhân dân (CMND), và ngày dự thi trong trường hợp một thí sinh đăng ký thi nhiều lần. Trong NLU pipeline, chúng tôi lựa chọn thành phần **RegexEntityExtractor** (trích xuất entity dựa trên các biểu thức chính quy regex) vì nó phù hợp với các thực thể có cấu trúc và định dạng cụ thể như ngày tháng hoặc mã định danh. Ví dụ chúng tôi định nghĩa biểu thức chính quy cho số CCCD là trích xuất các chuỗi gồm 12 chữ số liên tiếp.

3.3 Tạo câu trả lời (Rasa NLG)

Rasa NLG (Natural Language Generation) có trách nhiệm phát sinh câu trả lời tương ứng với các thông điệp của người dùng. Câu trả lời có thể là một phát biểu (utterance) hoặc một hành động tùy biến (custom action). Phát biểu là một câu trả lời được định nghĩa trước cho một ý định của người dùng, ví dụ “Lệ phí thi VSTEP là 1,800,000VNĐ” cho câu hỏi liên quan đến lệ phí thi; trong khi, hành động tùy biến là một hành động được định nghĩa lại để thực hiện các tác vụ đặc biệt như gọi API, truy vấn cơ sở dữ liệu, hoặc xử lý logic phức tạp dựa trên yêu cầu của người dùng, ví dụ chatbot phải truy vấn cơ sở dữ liệu để trả về kết quả thi của thí sinh.

Trong nghiên cứu này, chatbot cần truy vấn dữ liệu trong cơ sở dữ liệu dựa trên đầu vào của người dùng trong nhiều tình huống. Ví dụ, với ý định hỏi về kết quả thi:

chatbot sẽ phản hồi yêu cầu thí sinh cung cấp số CCCD đăng ký khi dự thi; hệ thống sẽ nhận diện thực thể “CCCD” dựa vào phản hồi của thí sinh; dựa vào thông tin này, hệ thống sẽ truy vấn cơ sở dữ liệu để lấy kết quả thi và trả kết quả cho thí sinh.

3.4 Quản lý hội thoại (Rasa DM)

Rasa DM (Dialogue Manager) chịu trách nhiệm quyết định cách chatbot phản hồi và điều phối cuộc trò chuyện với người dùng. Cụ thể, nó quản lý trạng thái cuộc trò chuyện, chọn hành động tiếp theo mà chatbot cần thực hiện và đảm bảo sự liên tục, mạch lạc trong các luồng hội thoại. Chính sách (Policy) là thành phần quyết định cách chatbot phản hồi trong các bước đối thoại tiếp theo, dựa trên ngữ cảnh cuộc trò chuyện [15]. Có nhiều loại policies khác nhau như MemoizationPolicy, ghi nhớ và lặp lại các mẫu đối thoại đã xảy ra; TEDPolicy, sử dụng mạng nơ-ron để dự đoán hành động tiếp theo dựa trên lịch sử đối thoại; và RulePolicy, xác định các quy tắc cụ thể mà chatbot phải tuân theo. FallbackPolicy được kích hoạt khi mô hình không tự tin về hành động tiếp theo, thường yêu cầu người dùng làm rõ ý định, trong khi FormPolicy quản lý các cuộc trò chuyện đòi hỏi thu thập thông tin có hệ thống. Mỗi bước trong cuộc trò chuyện, Rasa xem xét toàn bộ ngữ cảnh và các chính sách sẽ tính toán xác suất của các hành động có thể xảy ra, chọn hành động có xác suất cao nhất. Việc cấu hình đúng các policies là rất quan trọng để chatbot hoạt động hiệu quả, xử lý các tình huống đa dạng và cải thiện trải nghiệm người dùng. Tùy thuộc vào mục đích của mình, chúng tôi đã chọn nhiều chính sách trong cấu hình được định nghĩa trong tập tin config.yml.

3.5 Cơ sở dữ liệu

Hiện tại, hệ thống quản lý kỳ thi VSTEP tại Trường Đại học Sài Gòn đã được xây dựng sẵn trên nền tảng web. Thí sinh có thể truy cập vào website chính thức để tra cứu thông tin như lịch thi, điểm thi. Hệ thống chatbot mà chúng tôi xây dựng không xây dựng cơ sở dữ liệu riêng biệt mà sử dụng cơ sở dữ liệu của hệ thống có sẵn. Khi cần thông tin phản hồi dạng hành động tùy biến (ví dụ tra cứu điểm thi bằng CCCD), hệ thống chatbot sẽ gọi API được cung cấp bởi hệ thống web đã xây dựng sẵn.

4. THỰC NGHIỆM

4.1. Thiết lập thực nghiệm

Tập dữ liệu được sử dụng cho quá trình học mô hình xác định chủ đề và thực thể của thông điệp đầu vào. Việc thu thập dữ liệu huấn luyện được tiến hành như sau. Chúng tôi đã thiết kế mẫu thu thập các câu hỏi thường gặp của sinh viên liên quan đến công tác tuyển sinh và tổ chức thi tiếng Anh VSTEP tại Trường Đại học Sài Gòn. Kết quả thu được hơn 1600 mẫu câu hỏi liên quan đến 50 chủ đề (intents) khác nhau (ví dụ về chủ đề và câu hỏi có thể được tham khảo ở Bảng 2). Chúng tôi chia bộ dữ liệu thành hai

phần với 80% là dữ liệu huấn luyện (training dataset) và 20% là dữ liệu kiểm tra (validation dataset). Dữ liệu huấn luyện được chia thành năm nhóm có kích thước bằng nhau để áp dụng phương pháp k-fold cross-validation. Sau khi chia dữ liệu thành năm phần, trong lần lặp đầu tiên, phần đầu tiên được dùng để kiểm tra mô hình, còn các phần còn lại dùng để huấn luyện. Ở lần lặp thứ hai, phần thứ hai được sử dụng làm tập kiểm tra, và các phần còn lại dùng để huấn luyện. Quá trình này tiếp tục lặp lại cho đến khi mỗi trong năm phần đều đã được dùng để kiểm tra.

Bảng 2

Chủ đề	Ví dụ câu hỏi
Lệ phí thi	Cho em hỏi lệ phí thi A2 là bao nhiêu ạ?
Hồ sơ đăng ký thi	Hồ sơ đăng ký thi bao gồm gì?
Hình thức thi	Làm bài trên máy tính hay trên giấy?

Ví dụ chủ đề và câu hỏi

4.2. Kết quả thực nghiệm

Chúng tôi thực hiện đánh giá hiệu suất của mô hình chatbot hoặc các thành phần xử lý ngôn ngữ tự nhiên (NLP) sau khi đã được huấn luyện, giúp xác định mức độ hiệu quả của mô hình trong việc hiểu và phản hồi đúng ý định của người dùng cũng như nhận diện chính xác các thực thể. Độ đo được sử dụng để đánh giá mô hình bao gồm:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Trong đó, TP (True Positive) là tổng số trường hợp mà mô hình dự đoán đúng lớp dương, trong khi TN (True Negative) là tổng số trường hợp mà mô hình dự đoán đúng lớp âm. FP (False Positive) là số lượng các quan sát thuộc lớp âm nhưng lại bị mô hình dự đoán là lớp dương, còn FN (False Negative) là số lượng các quan sát thuộc lớp dương nhưng bị mô hình dự đoán là lớp âm.

Việc phân loại ý định (intent classification) được đánh giá bằng độ đo Precision, Recall, F1 và Accuracy thông qua phương pháp cross-validation. Bảng 3 hiển thị giá trị trung bình của các độ đo của mô hình được chọn.

Bảng 3

Độ đo	Giá trị
Precision	91.25%
Recall	91.34%
F1	90.71%
Accuracy	91.34%

Kết quả phân loại ý định

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi xây dựng hệ thống chatbot hỗ trợ công tác tuyển sinh và tổ chức thi tiếng Anh VSTEP tại Trường Đại học Sài Gòn. Sau khi khảo sát các nền tảng phát triển chatbot, chúng tôi lựa chọn nền tảng Rasa để xây dựng chatbot vì những thế mạnh của nó như miễn phí, mã nguồn mở, hỗ trợ các mô hình NLP tiên tiến, dễ tích hợp vào các nền tảng nhắn tin và có cộng đồng hỗ trợ lớn. Bài báo trình bày khung ứng dụng chatbot tổng quan cũng như cài đặt cụ thể của từng thành phần. Thực nghiệm cho thấy hệ thống chatbot đã đạt được độ chính xác cao (accuracy = 91.34%) trong việc phân loại ý định và nhận diện thực thể. Bên cạnh đó, hệ thống cung cấp các phản hồi tùy biến cho các tác vụ chuyên biệt cần tương tác với hệ thống bên ngoài dựa trên nhu cầu cụ thể của người dùng.

Tuy nhiên, dữ liệu huấn luyện cần được cập nhật thủ công với các ý định và thông tin mới cũng như giải quyết các hiểu nhầm về ý định thường xảy ra trong quá trình tư vấn. Trong phạm vi nghiên cứu trong tương lai, chúng tôi sẽ tận dụng các mô hình ngôn ngữ lớn đã được huấn luyện (pre-trained large language models) để hỗ trợ quá trình này.

TÀI LIỆU THAM KHẢO

- [1] Følstad, Asbjørn, and Petter Bae Brandtzæg (2017). "Chatbots and the new world of HCI." *Interactions* 24.4: 38-42.
- [2] Adamopoulou, Eleni, and Lefteris Moussiades (2020). "Chatbots: History, technology, and applications." *Machine Learning with applications* 2: 100006.
- [3] Bhharathee, A., et al (2023). "AI-Powered Student Assistance Chatbot." *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE, 2023.

-
- [4] Ali, Muhammad Shahroze, et al (2022). "Intelligent agents in educational institutions: NEdBOT-NLP-based chatbot for administrative support using DialogFlow." *2022 IEEE International Conference on Agents (ICA)*. IEEE, 2022.
- [5] Usigan, Alvin Gorom, et al (2022). "Implementation of an Undergraduate Admission Chatbot Using Microsoft Azure's Question Answering and Bot Framework." *Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference*. 2022.
- [6] Patel, Dhruv, et al (2023). "College enquiry chatbot using conversational AI." *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* 11.5
- [7] Aloqayli, Alaa, and Hoda Abdelhafez (2023). "Intelligent Chatbot for Admission in Higher Education." *International Journal of Information and Education Technology* 13.9: 1348-1357.
- [8] Nguyen, Trung Thanh, et al (2021). "NEU-chatbot: Chatbot for admission of National Economics University." *Computers and Education: Artificial Intelligence* 2: 100036.
- [9] Fernández Ortega, L. (2018). Introduction to IBM's Watson and its services.
- [10] Biswas, M., & Biswas, M. (2018). Microsoft bot framework. *Beginning AI Bot Frameworks: Getting Started with Bot Development*, 25-66.
- [11] Sabharwal, N., Agrawal, A., Sabharwal, N., & Agrawal, A. (2020). Introduction to Google dialogflow. *Cognitive virtual assistants using google dialogflow: develop complex cognitive bots using the google dialogflow platform*, 13-54.
- [12] Mitrevski, M., & Mitrevski, M. (2018). Getting started with wit. ai. *Developing Conversational Interfaces for iOS: Add Responsive Voice Control to Your Apps*, 143-164.
- [13] Bocklisch, Tom, et al (2017). "Rasa: Open source language understanding and dialogue management." *arXiv preprint arXiv:1712.05181*.
- [14] Mady Mantha (2020). "Introducing DIET: state-of-the-art architecture that outperforms fine-tuning BERT and is 6X faster to train." <https://rasa.com/blog/introducing-dual-intent-and-entity-transformer-diet-state-of-the-art-performance-on-a-lightweight-architecture/>
- [15] Sharma, R. K., & Joshi, M. (2020). An analytical study and review of open source chatbot framework, rasa. *Int. J. Eng. Res*, 9(06), 1011-1014.

BUILDING A CHATBOT FOR SUPPORTING ADMISSION AND ORGANIZATION OF VSTEP TEST

Pham Trong Nguyen

Saigon university

Email: nguyen.pt@sgu.edu.vn

ABSTRACT

A chatbot is a computer program designed to simulate conversations with users via messaging platforms, websites, or mobile applications. Using chatbots in admissions can bring significant benefits to schools and educational institutions. In this paper, we develop a chatbot application based on the Rasa platform to support the admission process and organization of the VSTEP English test at Saigon university. Our application can recognize more than 50 different types of user questions with relatively high accuracy in natural language understanding.

Keywords: virtual assistant, chatbot, admission, Rasa.



Phạm Trọng Nguyễn sinh ngày 11/7/1986 tại thành phố Hồ Chí Minh. Năm 2016, ông tốt nghiệp Thạc sĩ chuyên ngành Hệ thống thông tin tại Học viện Kỹ thuật Quân sự. Hiện nay, ông đang công tác tại Phòng Giáo dục thường xuyên, trường Đại học Sài Gòn.

Lĩnh vực nghiên cứu: Hệ thống thông tin và Khoa học máy tính.