

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



KHOÁ LUẬN TỐT NGHIỆP
NGÀNH MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU

Đề tài
**XÂY DỰNG HỆ THỐNG CHATBOT CHO TRƯỜNG
ĐẠI HỌC Y DƯỢC CẦN THƠ**

Giảng viên hướng dẫn:
PGS.TS ĐỖ THANH NGHỊ

Sinh viên thực hiện:
PHẠM THÀNH TUẤN LỘC
MSSV: B2013481
KHOÁ 46

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



KHOÁ LUẬN TỐT NGHIỆP
NGÀNH MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU

Đề tài

XÂY DỰNG HỆ THỐNG CHATBOT CHO TRƯỜNG
ĐẠI HỌC Y DƯỢC CẦN THƠ

Giảng viên hướng dẫn:

PGS.TS ĐỖ THANH NGHỊ

Sinh viên thực hiện:

PHẠM THÀNH TUẤN LỘC

MSSV: B2013481

KHOÁ 46

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG
TIN VÀ TRUYỀN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA
VIỆT NAM
Độc lập – Tự do – Hạnh phúc

XÁC NHẬN CHỈNH SỬA LUẬN VĂN
THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh): Xây dựng hệ thống Chatbot cho Trường Đại Học Y Dược Cần Thơ (Developing a Chatbot System for Can Tho University of Medicine and Pharmacy)

Họ tên sinh viên: PHẠM THÀNH TUẤN LỘC MASV: B2013481

Mã lớp: DI20T9A1

Đã báo cáo tại hội đồng ngành: Mạng máy tính và truyền thông dữ liệu.

Ngày báo cáo: 11/12/2024

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày tháng năm 2024

Giáo viên hướng dẫn

(Ký và ghi họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

LỜI CẢM ƠN

Với tất cả tình cảm và lòng chân thành, tôi xin gửi lời cảm ơn sâu sắc đến quý Thầy, Cô đã giảng dạy lớp Mạng máy tính và Truyền thông dữ liệu khóa 46 (2020 - 2024) tại trường Đại học Cần Thơ. Với sự nhiệt huyết trong giảng dạy, cùng những lời khuyên, đóng góp, hỗ trợ tận tình của quý Thầy, Cô đã giúp tôi rất nhiều trong suốt quá trình học tập và nghiên cứu.

Đặc biệt, tôi xin gửi lời tri ân sâu sắc tới Thầy PGS.TS. Đỗ Thanh Nghị. Thầy không chỉ là người thầy tận tâm mà còn là người hướng dẫn nhiệt tình, luôn sẵn sàng giúp đỡ tôi vượt qua mọi khó khăn trong suốt thời gian học tập cũng như trong quá trình nghiên cứu thực hiện luận văn này.

Tôi xin cảm ơn gia đình và bạn bè đã luôn giúp đỡ, ủng hộ tôi trong suốt thời gian học tập. Tôi xin chân thành cảm ơn các anh chị và quý Thầy, Cô góp ý xây dựng và hướng dẫn tôi để có thể hoàn thành luận văn.

Mặc dù đã cố gắng hết mình, nhưng chắc chắn luận văn vẫn không tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý quý từ quý Thầy, Cô để tôi có thể bổ sung và cải tiến để tài luận văn này ngày càng hoàn thiện hơn.

Xin chân thành cảm ơn!

PHẠM THÀNH TUẤN LỘC

Lớp Mạng máy tính và truyền thông dữ liệu

DI20T9A1 Khoa 46

Mục lục

DANH MỤC HÌNH ẢNH.....	4
DANH MỤC BẢNG	5
CHƯƠNG 1: GIỚI THIỆU	9
1.1 Lý do chọn đề tài.....	9
1.2 Lịch sử giải quyết vấn đề	9
1.3 Mục tiêu đề tài.....	10
1.4 Đối tượng nghiên cứu.....	10
1.5 Phạm vi nghiên cứu.....	11
1.6 Phương pháp nghiên cứu.....	11
1.7 Ý nghĩa khoa học và thực tiễn của đề tài	12
1.8 Bố cục luận văn	13
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	14
2.1 Tổng quan về Seq2Seq.....	14
2.1.1 Encoder (Bộ Mã Hóa)	15
2.1.2 Decoder (Bộ Giải Mã).....	16
2.1.3 RNN (Recurrent neural network).....	16
2.1.4 LSTM (Long Short-Term Memory)	17
2.2 Tổng Quan Về Rasa	19
2.3 Rasa NLU (Natural Language Understanding).....	20
2.3.1 Tokenization.....	20
2.3.2 Intent Classification (Phân Loại Ý Định)	21
2.3.3 Entity Recognition (Nhận Diện Thực Thể).....	21
2.3.4 Synonyms and Regex Features (Từ Đồng Nghĩa và Đặc Trung Biểu Thức Chính Quy).....	21
2.3.5 Pipeline (Đường Ống Xử Lý)	22
2.4 Rasa Core	24
2.4.1 Dialogue Management (Quản Lý Hội Thoại)	24

2.4.2 Stories (Kịch Bản Hội Thoại)	25
2.4.3 Policies (Chính Sách Ra Quyết Định).....	25
2.4.4 Action (Hành Động).....	25
2.4.5 Slot Filling (Điền Giá Trị Cho Slots).....	26
2.4.6 Event Tracking (Theo Dõi Sự Kiện).....	26
CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN	27
3.1 Tổng quan về bài toán Seq2Seq.....	28
3.1.1 Thu thập và xử lý dữ liệu	28
3.1.2 Ứng dụng LSTM huấn luyện mô hình sửa lỗi chính tả	33
3.2 Tổng quan bài toán Rasa.....	37
3.2.1 Xử lý thông điệp (NLU).....	38
3.2.2 Theo dõi cuộc hội thoại (Tracker).....	45
3.2.3 Custom action.....	46
Chương 4: Kết quả thực nghiệm	48
4.1 Thu thập và xây dựng tập dữ liệu.....	48
4.2 Môi trường thực nghiệm	49
4.3 Kết quả thực nghiệm	49
4.4 Đánh giá độ chính xác của hệ thống	53
Chương 5: Kết luận và hướng phát triển	54
5.1 Kết luận	54
5.2 Hướng phát triển	54
Phụ lục	56
Tài liệu tham khảo	66

DANH MỤC HÌNH ẢNH

Hình 1. Mô hình Encoder-Decoder.....	15
Hình 2. Các dạng bài toán RNN	17
Hình 3. Kiến trúc LSTM	18
Hình 4. Sơ đồ của một tế bào LSTM	19
Hình 5. Sơ đồ hoạt động hệ thống sửa lỗi chính tả và chatbot	27
Hình 6. Sơ đồ hoạt động hệ thống sửa lỗi chính tả	28
Hình 7. Sơ đồ hệ thống chatbot Rasa	37
Hình 8. Lược đồ phân loại ý định và trích xuất thực thể sử dụng Rasa NLU ...	38
Hình 9. Biểu đồ thống kê dữ liệu	48
Hình 10. Ma trận nhầm lẫn ý định	51
Hình 11. Biểu đồ tự tin của ý định	52

DANH MỤC BẢNG

Bảng 1: Thông số kĩ thuật phần cứng thực hiện nghiên cứu	49
Bảng 2: Bảng mô tả số lần thử nghiệm với người dùng	50
Bảng 3: Bảng mô tả hội thoại test với chatbot rasa.....	65

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Viết đầy đủ
AI	Artificial Intelligence
RAG	Retrieval-Augmented Generation
RNN	Recurrent neural network
LSTM	LSTM (Long Short-Term Memory)
GRU	Gated Recurrent Unit
NLU	Natural Language Understanding
STT	Số thứ tự
Seq2Seq	Sequence-to-Sequence
NLP	Natural language processing
sos	Start of Sequence
eos	End of Sequence

TÓM TẮT

Luận văn trình bày chi tiết các bước xây dựng hệ thống chatbot dựa trên nền tảng Rasa, từ việc thu thập dữ liệu huấn luyện, thiết kế và huấn luyện mô hình, đến việc thử nghiệm và triển khai. Hệ thống trả lời tự động này được đánh giá qua nhiều ứng dụng thực tiễn, bao gồm dịch vụ khách hàng, hệ thống chatbot có thể xử lý các câu hỏi phổ biến, tự động hóa các quy trình hỗ trợ khách hàng, giúp giảm tải công việc cho đội ngũ nhân viên, đồng thời nâng cao sự hài lòng của khách hàng nhờ thời gian phản hồi nhanh chóng. Trong giáo dục, chatbot có thể hỗ trợ học tập, trả lời các câu hỏi về nội dung học tập hoặc giải đáp các thắc mắc của sinh viên, từ đó giúp nâng cao hiệu quả học tập và trải nghiệm giáo dục số. Ngoài ra, luận văn cũng đánh giá các thách thức khi phát triển hệ thống chatbot bằng Rasa, chẳng hạn như quản lý hội thoại trong các kịch bản phức tạp, duy trì tính chính xác và cập nhật liên tục dữ liệu huấn luyện để đảm bảo phản hồi luôn phù hợp với ngữ cảnh thực tế. Hệ thống Rasa không chỉ hỗ trợ việc quản lý hội thoại phức tạp mà còn dễ dàng tích hợp với các công cụ và cơ sở dữ liệu hiện có, giúp tăng cường độ chính xác của các phản hồi và đảm bảo rằng hệ thống luôn cập nhật thông tin mới nhất.

Một trong những thách thức đáng chú ý khi phát triển hệ thống chatbot là xử lý các lỗi chính tả trong đầu vào của người dùng. Lỗi chính tả có thể dẫn đến sự hiểu lầm về ý định của người dùng, từ đó ảnh hưởng đến chất lượng của phản hồi mà hệ thống chatbot cung cấp. Để giải quyết vấn đề này, luận văn áp dụng mô hình Seq2Seq (sequence-to-sequence) như một giải pháp cơ bản cho việc sửa các lỗi chính tả phổ biến.

Việc tích hợp Seq2Seq vào hệ thống Rasa mang lại lợi ích rõ rệt, đặc biệt trong các tình huống mà người dùng thường xuyên nhập sai do nhầm lẫn hoặc tốc độ gõ phím nhanh. Seq2Seq giúp chatbot nhận diện và sửa các lỗi chính tả trước khi phân tích ý định của người dùng, cải thiện quá trình xử lý ngôn ngữ tự nhiên (NLP) của hệ thống. Điều này đảm bảo rằng ngay cả khi người dùng mắc lỗi gõ phím, hệ thống vẫn có thể cung cấp phản hồi phù hợp và chính xác.

Kết quả nghiên cứu cho thấy, hệ thống trả lời tự động dựa trên Rasa kết hợp với mô hình Seq2Seq đã cải thiện đáng kể khả năng xử lý ngôn ngữ và phản hồi tự động trong các trường hợp đầu vào chứa lỗi chính tả. Không chỉ giúp tăng cường độ chính xác của hệ thống trong việc nhận diện ý định và quản lý hội thoại phức tạp, mà sự kết hợp này còn mang lại trải nghiệm người dùng tốt hơn nhờ khả năng phản hồi nhanh chóng. Việc áp dụng mô hình Seq2Seq vào chatbot phát triển trên nền tảng Rasa cho thấy tiềm năng lớn trong việc giảm thiểu sai sót do lỗi chính tả, từ đó cải thiện hiệu quả tổng thể của hệ thống.

ABSTRACT

This thesis details the development of a chatbot system using Rasa, covering steps from data collection, model design and training, to testing and deployment. The automated response system is evaluated across practical applications, such as customer service, where it handles common queries, automates support processes, reduces staff workload, and enhances customer satisfaction through fast response times. In education, the chatbot supports learning by answering questions related to course content, improving educational effectiveness and the digital learning experience. The thesis also addresses challenges in chatbot development with Rasa, such as managing complex dialogues and continuously updating training data to maintain response relevance in real-world scenarios. Rasa's system excels at managing complex conversations and easily integrates with existing tools and databases, ensuring accuracy and up-to-date information.

One notable challenge in chatbot development is handling user input spelling errors, which can distort intent recognition and response quality. The thesis employs a Seq2Seq (sequence-to-sequence) model to correct spelling errors, significantly improving the system's natural language processing (NLP) capabilities. Seq2Seq integration enables the chatbot to detect and correct user input errors before analyzing intent, ensuring accurate and relevant responses, even when input contains typos.

Results show that integrating Seq2Seq with Rasa-based chatbots significantly enhances language processing and automatic responses when dealing with input errors. This combination improves accuracy in intent recognition, conversation management, and overall user experience by providing fast. The use of Seq2Seq demonstrates its potential in reducing errors due to typos, thereby improving the overall system's efficiency.

CHƯƠNG 1: GIỚI THIỆU

1.1 Lý do chọn đề tài

Trong bối cảnh trí tuệ nhân tạo (AI) phát triển mạnh mẽ, các hệ thống chatbot thông minh ngày càng đóng vai trò quan trọng trong nhiều lĩnh vực như chăm sóc khách hàng, thương mại điện tử và giáo dục. Chatbot không chỉ đáp ứng nhu cầu tương tác 24/7 mà còn hỗ trợ tối ưu quy trình làm việc và nâng cao trải nghiệm người dùng. Tuy nhiên, để chatbot hoạt động hiệu quả, việc hiểu ngữ cảnh và phản hồi chính xác, đặc biệt khi xử lý lỗi ngôn ngữ, là yếu tố then chốt.

Đề tài này ứng dụng Seq2Seq và Rasa để xây dựng một hệ thống chatbot thông minh, kết hợp khả năng sửa lỗi chính tả, ngữ pháp với luồng hội thoại tự nhiên và chính xác. Seq2Seq đảm bảo chatbot có thể xử lý tự động các lỗi đầu vào của người dùng, trong khi Rasa cung cấp nền tảng linh hoạt để phát triển và quản lý hội thoại.

Việc chọn đề tài này xuất phát từ nhu cầu phát triển hệ thống chatbot tiên tiến, đáp ứng yêu cầu ngày càng cao về tính tương tác, hiệu quả và chính xác trong giao tiếp. Ứng dụng công nghệ không chỉ nâng cao chất lượng dịch vụ mà còn thúc đẩy việc triển khai AI vào thực tế, mở ra cơ hội cải tiến các giải pháp công nghệ hiện đại trong tương lai.

1.2 Lịch sử giải quyết vấn đề

Tại Việt Nam ứng dụng chatbot thu hút nhiều nhà khoa học và sinh viên của các trường đại học nghiên cứu và phát triển, một số dự án nổi bật đó là:

Đoàn Thị Hồng Phước, Lê Văn Tường Lân, Nguyễn Văn Trung Khoa Công nghệ Thông tin, Trường Đại học Khoa học, Đại học Huế, 77 Nguyễn Huệ, Huế, Việt Nam. XÂY DỰNG KHUNG ỨNG DỤNG AI CHATBOT TRONG LĨNH VỰC QUY CHẾ ĐÀO TẠO. Kết quả chạy thử nghiệm chương trình cho thấy rằng độ chính xác của mô hình hiểu ngôn ngữ tự nhiên là 84,6% và độ chính xác theo đánh giá của các chuyên gia là 89,3%. [1]

SV.Tăng Xuân Biên1 , SV. Trần Minh Chiến 1 , SV.Lê Thanh Nga1 , SV.Trần Bá Hiển 1 , SV. Nguyễn Đức Nam Bình 1 , TS. Hà Thị Kim Duyên1 , TS. Ngô Mạnh Tiến 2 Khoa Điện tử, Trường Đại học Công nghiệp Hà Nội 2Viện Vật lý, Viện hàn lâm Khoa học và Công nghệ Việt Nam NGHIÊN CỨU, THIẾT KẾ TRỢ LÝ ẢO CÁ NHÂN TRÊN NỀN TẢNG WEB KẾT HỢP HỌC MÁY (MACHINE LEARNING) VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN (NATURAL LANGUAGE PROCESSING) . Kết quả test trên tập dữ liệu cho độ chính xác khoảng 81%. [2]

1.3 Mục tiêu đề tài

Mục tiêu chính của đề tài này là xây dựng một hệ thống trả lời tự động (chatbot) thông minh, tích hợp mô hình Seq2Seq để xử lý các lỗi chính tả, giúp hiểu chính xác đầu vào từ người dùng. Hệ thống này được phát triển dựa trên nền tảng mã nguồn mở Rasa, với khả năng hiểu và phản hồi ngôn ngữ tự nhiên. Cụ thể, các mục tiêu của đề tài bao gồm:

Ứng dụng Seq2Seq: Tích hợp Seq2Seq để tự động sửa các lỗi chính tả hoặc lỗi đánh máy trong đầu vào từ người dùng, giúp chatbot hiểu chính xác ý định, ngay cả khi có lỗi ngữ pháp. Điều này đảm bảo hiệu quả và độ chính xác trong phản hồi, đặc biệt quan trọng khi chatbot được thiết kế để tư vấn và hỗ trợ người dùng tại Đại học Y Dược Cần Thơ.

Ứng dụng Rasa: Nghiên cứu cách sử dụng nền tảng Rasa, từ quy trình xử lý ngôn ngữ tự nhiên (NLP) đến việc xây dựng mô hình hội thoại thông minh, để tạo ra chatbot có khả năng phản hồi hiệu quả và phù hợp với nhu cầu thực tế.

1.4 Đối tượng nghiên cứu

Nền tảng Rasa:

- Rasa NLU (Natural Language Understanding): Tìm hiểu về khả năng phân tích và xử lý ngôn ngữ tự nhiên, cách trích xuất ý định (intent) và thực thể (entity) từ câu hỏi của người dùng. Rasa NLU đóng vai trò quan trọng trong việc hiểu và phân loại các yêu cầu mà chatbot nhận được.
- Rasa Core: Tìm hiểu về cơ chế điều hướng hội thoại và xử lý các luồng hội thoại phức tạp. Rasa Core giúp xây dựng các mô hình phản hồi thông minh dựa trên ngữ cảnh và lịch sử cuộc trò chuyện.

Mô hình Seq2Seq (Sequence-to-Sequence):

Là một mô hình học sâu được thiết kế để chuyển đổi một chuỗi đầu vào thành một chuỗi đầu ra tương ứng, rất phù hợp cho việc sửa các lỗi chính tả phổ biến và ngữ pháp từ người dùng.

Dữ liệu cho hệ thống chatbot là thông tin chi tiết về Trường Đại học Y dược Cần Thơ, bao gồm: học phí, chương trình đào tạo, tuyển sinh, điểm chuẩn và các hoạt động của nhà trường.

1.5 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài này không chỉ dừng lại ở việc phát triển hệ thống chatbot thông minh dựa trên nền tảng Rasa, mà còn tích hợp mô hình Seq2Seq để xử lý lỗi chính tả, nâng cao độ chính xác trong việc hiểu và phản hồi của chatbot. Các nội dung nghiên cứu chính bao gồm:

Thiết kế và triển khai hệ thống:

Rasa NLU (Natural Language Understanding): Nghiên cứu sâu về cách Rasa NLU phân tích và xử lý ngôn ngữ tự nhiên (NLP), bao gồm việc trích xuất ý định (intents) và thực thể (entities) từ các câu hỏi của người dùng. Đây là bước quan trọng giúp chatbot có thể hiểu rõ yêu cầu của người dùng trong các tình huống thực tế. Mục tiêu của phạm vi nghiên cứu này tăng khả năng nhận diện ý định và thực thể từ dữ liệu thực tế lấy từ Đại học Y Dược Cần Thơ, giúp chatbot cung cấp phản hồi chính xác dựa trên thông tin tuyển sinh, chương trình đào tạo, học phí, và các thủ tục hành chính.

Rasa Core: Phạm vi nghiên cứu sẽ tập trung vào việc xây dựng các luồng hội thoại phức tạp dựa trên ngữ cảnh và lịch sử cuộc trò chuyện. Rasa Core giúp điều hướng hội thoại thông minh và mạch lạc, đảm bảo rằng chatbot có thể quản lý tốt các cuộc hội thoại kéo dài, thích ứng với những yêu cầu đa dạng từ người dùng.

Phát hiện và sửa các lỗi chính tả phổ biến: Seq2Seq sẽ xử lý đầu vào từ người dùng, phát hiện các từ hoặc câu có lỗi chính tả và sửa chữa chúng trước khi gửi đến Rasa NLU để phân tích. Điều này không chỉ đảm bảo chatbot hiểu đúng yêu cầu của người dùng mà còn cải thiện trải nghiệm giao tiếp tổng thể.

Ứng dụng dữ liệu thực tế: Nghiên cứu sẽ sử dụng dữ liệu từ Trường Đại học Y Dược Cần Thơ, bao gồm các thông tin liên quan đến tuyển sinh, chương trình đào tạo, học phí và thủ tục hành chính. Hệ thống chatbot sẽ được phát triển để có thể hỗ trợ người dùng truy xuất thông tin nhanh chóng và hiệu quả.

Phạm vi nghiên cứu không chỉ tập trung vào việc xây dựng một chatbot thông minh mà còn đảm bảo tích hợp các công nghệ tiên tiến để đáp ứng tốt nhất nhu cầu cụ thể của Trường Đại học Y Dược Cần Thơ.

1.6 Phương pháp nghiên cứu

Nghiên cứu tài liệu: Tìm hiểu nền tảng Rasa (Rasa NLU và Rasa Core) thông qua tài liệu kỹ thuật và bài báo khoa học, tập trung vào cách xử lý ngôn ngữ tự nhiên và điều hướng luồng hội thoại dựa trên ngữ cảnh. Tham khảo các tài liệu về mô hình

Seq2Seq để hiểu cách phát hiện và sửa lỗi chính tả trong chuỗi văn bản, từ đó áp dụng vào việc nâng cao khả năng của chatbot.

Thu thập và xử lý dữ liệu: Dữ liệu từ Trường Đại học Y Dược Cần Thơ: Lấy từ website, tài liệu tuyển sinh, thông tin chương trình đào tạo, học phí và thủ tục hành chính. Dữ liệu huấn luyện Seq2Seq: Thu thập từ các hội thoại thực tế chứa lỗi chính tả để đào tạo mô hình xử lý lỗi.

Kiểm thử chatbot: Đánh giá khả năng nhận diện ý định, thực thể, và xử lý hội thoại phức tạp. Kiểm thử Seq2Seq: Đánh giá hiệu quả sửa lỗi chính tả và cải thiện khả năng hiểu ngôn ngữ tự nhiên của chatbot. Điều chỉnh hiệu suất dựa trên thời gian phản hồi, độ chính xác, và khả năng duy trì hội thoại.

Phân tích và đánh giá: Đo lường hiệu quả dựa trên độ chính xác câu trả lời, thời gian phản hồi, và khả năng duy trì mạch lạc trong hội thoại. Đánh giá sự cải thiện từ việc tích hợp Seq2Seq vào hệ thống chatbot.

1.7 Ý nghĩa khoa học và thực tiễn của đề tài

Ý nghĩa khoa học:

- Góp phần phát triển lĩnh vực trí tuệ nhân tạo (AI) và xử lý ngôn ngữ tự nhiên (NLP) thông qua việc ứng dụng nền tảng Rasa và mô hình Seq2Seq.
- Seq2Seq giúp nâng cao khả năng tự động sửa lỗi chính tả, cải thiện độ chính xác trong phân tích và phản hồi của chatbot.
- Hỗ trợ cho các nghiên cứu ứng dụng AI trong các hệ thống hội thoại tự động.

Ý nghĩa thực tiễn:

- Đơn giản hóa quy trình hỗ trợ người dùng tại Trường Đại học Y Dược Cần Thơ, giúp truy xuất thông tin nhanh chóng và chính xác.
- Tự động hóa các câu trả lời lặp lại, giảm tải công việc, tiết kiệm chi phí và tăng hiệu quả hoạt động.
- Cải thiện trải nghiệm người dùng nhờ khả năng sửa lỗi chính tả, tăng độ chính xác và sự hài lòng.
- Mở rộng ứng dụng chatbot sang các lĩnh vực như chăm sóc khách hàng, thương mại điện tử, và dịch vụ công, hỗ trợ chuyển đổi số trong giáo dục và nhiều ngành khác.

1.8 Bố cục luận văn

Nội dung luận văn gồm có 5 chương:

Chương 1: Giới thiệu

Chương 2: Cơ sở lý thuyết

Chương 3: Phương pháp thực hiện

Chương 4: Kết quả thực nghiệm

Chương 5: Kết luận và hướng phát triển

Tổng kết chương 1: Phương pháp nghiên cứu của đề tài này tập trung vào việc xây dựng một hệ thống chatbot thông minh dựa trên nền tảng Rasa, kết hợp với mô hình Seq2Seq để sửa các lỗi chính tả phổ biến. Việc nghiên cứu tài liệu kỹ thuật, thu thập và xử lý dữ liệu từ môi trường thực tế, kiểm thử, đánh giá hệ thống sẽ giúp đảm bảo chatbot có khả năng phản hồi chính xác, nhanh chóng, và hiệu quả trong các tình huống hội thoại phức tạp.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trong chương này, luận văn sẽ tập trung vào việc trình bày các cơ sở lý thuyết liên quan đến việc xây dựng và triển khai hệ thống chatbot sử dụng Rasa. Nội dung bao gồm:

Tổng quan về Seq2Seq

- Encoder
- Decoder
- RNN
- LTSM

Tổng Quan Về Rasa

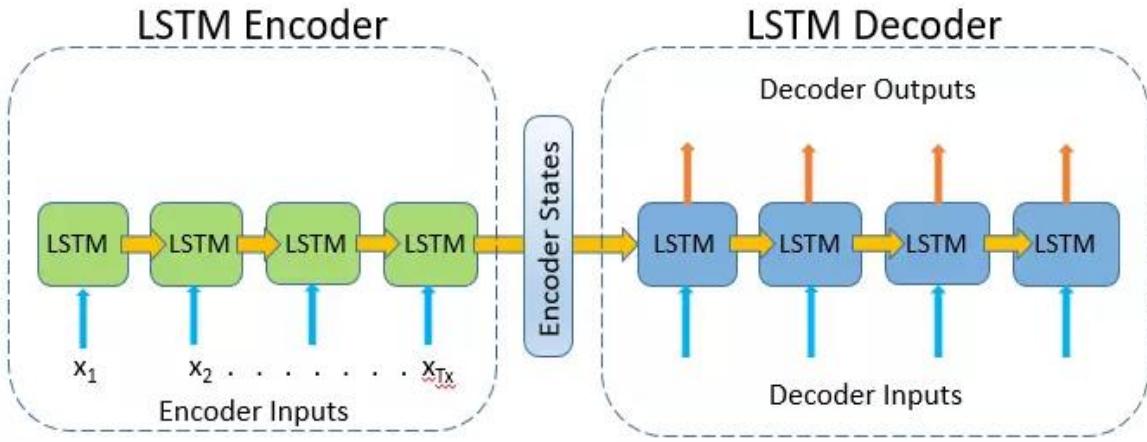
- Rasa NLU
- Rasa core

2.1 Tổng quan về Seq2Seq

Mô hình Seq2Seq, viết tắt của "Sequence to Sequence," là một kiến trúc mạnh mẽ được sử dụng rộng rãi trong nhiều ứng dụng xử lý ngôn ngữ tự nhiên (NLP), đặc biệt là trong việc dịch máy, tóm tắt văn bản, trả lời câu hỏi và sửa các lỗi chính tả phổ biến. Seq2Seq sử dụng phép biến đổi chuỗi, nó biến một chuỗi thành một chuỗi khác, Seq2Seq đã trở thành nền tảng cho nhiều tiến bộ trong lĩnh vực NLP nhờ khả năng xử lý các chuỗi dữ liệu có độ dài biến đổi.

Seq2Seq là một cách tiếp cận dịch máy (hay nói chung hơn là chuyển đổi trình tự) có nguồn gốc từ lý thuyết thông tin, trong đó giao tiếp được hiểu là một quá trình mã hóa-truyền-giải mã, và dịch máy có thể được nghiên cứu như một trường hợp đặc biệt của giao tiếp.

Mô hình Seq2Seq chủ yếu bao gồm hai thành phần chính: Encoder và Decoder



Hình 1. Mô hình Encoder-Decoder

2.1.1 Encoder (Bộ Mã Hóa)

Quá trình hoạt động:

1. Nhập Dữ Liệu: Chuỗi đầu vào $X = (x_1, x_2, \dots, x_T)$ được đưa vào Encoder.
2. Xử Lý Chuỗi: Mỗi phần tử x_t được xử lý tuần tự, cập nhật trạng thái ẩn h_t .
3. Biểu Diễn Ngữ Nghĩa: Sau khi xử lý toàn bộ chuỗi, trạng thái ẩn cuối cùng h_t được sử dụng làm biểu diễn ngữ nghĩa của toàn bộ chuỗi đầu vào.

Chức năng: Nhiệm vụ chính của Encoder là nhận đầu vào (chuỗi văn bản có lỗi chính tả) và chuyển đổi nó thành một vector ngữ nghĩa (context vector) đại diện cho thông tin cần thiết để Decoder có thể sinh ra đầu ra tương ứng.

Cách thức hoạt động:

Nhập Dữ Liệu: Chuỗi đầu vào thường được xử lý thành các vector nhúng (embedding vectors) thông qua một lớp nhúng (embedding layer) để chuyển đổi các từ thành các vector số.

Cấu Trúc Mạng: Encoder thường sử dụng LSTM (Long Short-Term Memory) hoặc GRU (Gated Recurrent Unit) để giữ lại thông tin qua các bước thời gian. Cả LSTM và GRU đều là các mạng nơ-ron hồi quy (RNN) có khả năng học và duy trì thông tin trong các chuỗi dài, điều này rất quan trọng cho các nhiệm vụ như sửa các lỗi chính tả phổ biến, nơi ngữ cảnh từ có thể ảnh hưởng đến quyết định của mô hình.

LSTM: LSTM sử dụng ba cổng (input gate, output gate, forget gate) để điều khiển thông tin nào sẽ được lưu lại hoặc loại bỏ, giúp giảm thiểu vấn đề vanishing gradient thường gặp trong các RNN truyền thống.

Kết quả: Sau khi xử lý, Encoder sẽ tạo ra một vector ngữ nghĩa đại diện cho toàn bộ chuỗi đầu vào. Vector này sẽ được truyền đến Decoder để tạo ra đầu ra mong muốn.

2.1.2 Decoder (Bộ Giải Mã)

Decoder cũng thường được xây dựng từ RNN tương tự như Encoder. Nó sử dụng biểu diễn ngữ nghĩa từ Encoder để sinh ra chuỗi đầu ra $Y=(y_1, y_2, \dots, y_{T'})$

Quá trình hoạt động:

1. Khởi Tạo Trạng Thái: Trạng thái ẩn ban đầu của Decoder được khởi tạo bằng trạng thái ẩn cuối cùng của Encoder h_T .
2. Sinh Từ Mỗi Bước: Tại mỗi bước thời gian t , Decoder nhận vào từ hiện tại y_{t-1} (hoặc token bắt đầu $\langle \text{sos} \rangle$) và trạng thái ẩn hiện tại để sinh ra từ tiếp theo y_t .
3. Kết Thúc Chuỗi: Quá trình tiếp tục cho đến khi sinh ra token kết thúc $\langle \text{eos} \rangle$ hoặc đạt độ dài tối đa.

Chức năng: Nhiệm vụ của Decoder là nhận vector ngữ nghĩa từ Encoder và sinh ra chuỗi đầu ra (câu đã được sửa lỗi) từng bước một.

Cách thức hoạt động:

Nhận Vector Ngữ Nghĩa: Decoder bắt đầu quá trình sinh đầu ra bằng cách nhận vector ngữ nghĩa từ Encoder. Nó cũng có thể sử dụng một từ bắt đầu đặc biệt (start token) để khởi động chuỗi đầu ra.

Tạo Đầu Ra: Mỗi lần sinh ra một từ, Decoder sẽ:

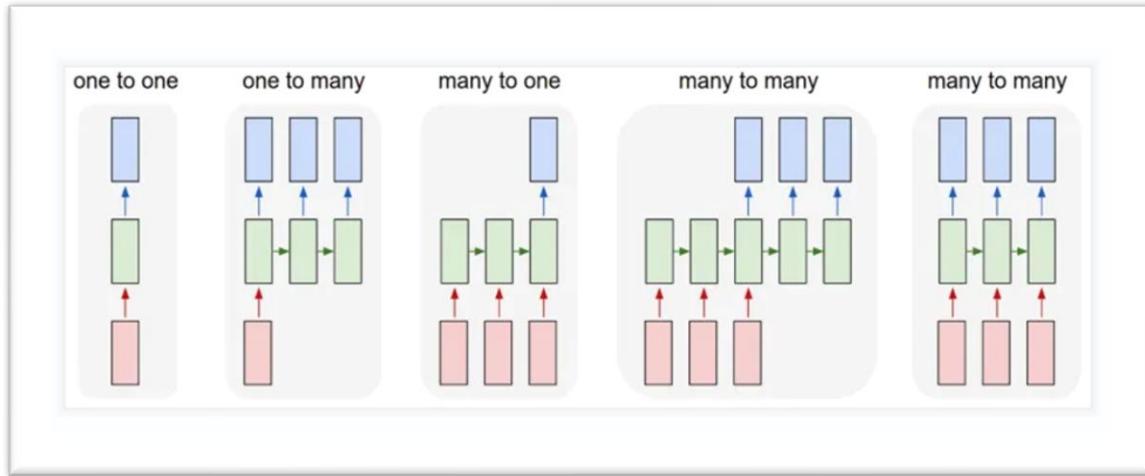
- Dựa vào vector ngữ nghĩa và từ đã sinh ra trước đó để dự đoán từ tiếp theo.
- Sử dụng các kiến trúc LSTM hoặc GRU để duy trì thông tin và trạng thái trong quá trình sinh.

Kết quả: Sau khi hoàn tất quá trình sinh, Decoder sẽ tạo ra một chuỗi đầu ra, có thể là một câu hoàn chỉnh đã được sửa lỗi.

2.1.3 RNN (Recurrent neural network)

Mạng nơ-ron hồi quy (RNN) là một mô hình học sâu được huấn luyện để xử lý và chuyển đổi đầu vào dữ liệu tuần tự thành đầu ra dữ liệu tuần tự cụ thể. Dữ liệu tuần tự là dữ liệu, chẳng hạn như từ, câu hoặc dữ liệu chuỗi thời gian, trong đó các thành phần tuần tự tương quan với nhau dựa trên ngữ nghĩa phức tạp và quy tắc cú pháp. RNN là một hệ thống phần mềm gồm nhiều thành phần được kết nối với nhau theo cách con người thực hiện chuyển đổi dữ liệu tuần tự, chẳng hạn như dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác. Phần lớn RNN đang được thay thế bằng trí tuệ nhân tạo (AI) dựa trên công cụ biến đổi và các mô hình ngôn ngữ lớn (LLM), hiệu quả hơn nhiều trong việc xử lý dữ liệu tuần tự.

Phân loại bài toán RNN:



Hình 2. Các dạng bài toán RNN

One to one: mẫu bài toán cho Neural Network (NN) và Convolutional Neural Network (CNN), 1 input và 1 output, ví dụ với CNN input là ảnh và output là ảnh được segment.

One to many: bài toán có 1 input nhưng nhiều output, ví dụ: bài toán caption cho ảnh, input là 1 ảnh nhưng output là nhiều chữ mô tả cho ảnh đó, dưới dạng một câu.

Many to one: bài toán có nhiều input nhưng chỉ có 1 output, ví dụ bài toán phân loại hành động trong video, input là nhiều ảnh (frame) tách ra từ video, output là hành động trong video

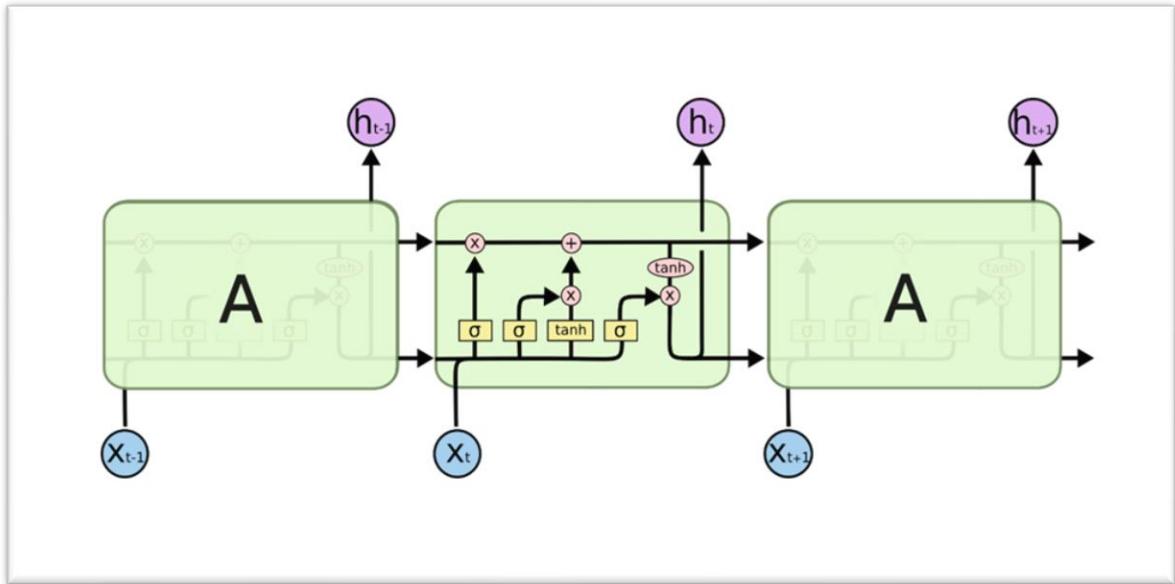
Many to many: bài toán có nhiều input và nhiều output, ví dụ bài toán dịch từ tiếng anh sang tiếng việt, input là 1 câu gồm nhiều chữ: “I love Vietnam” và output cũng là 1 câu gồm nhiều chữ “Tôi yêu Việt Nam”.

Trong mô hình Seq2Seq (sequence-to-sequence) này, cả đầu vào và đầu ra đều là chuỗi có độ dài tương đối tương tự nhau, và mô hình cần học cách ánh xạ từ một chuỗi ký tự này sang chuỗi ký tự khác. Điều này phù hợp với mô tả của Many to Many, nơi mà nhiều phần của chuỗi đầu vào được sử dụng để tạo ra nhiều phần của chuỗi đầu ra.

2.1.4 LSTM (Long Short-Term Memory)

Long Short-Term Memory (LSTM) là một biến thể của RNN, là một kiến trúc mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được thiết kế để giải quyết vấn đề học các phụ thuộc dài hạn trong dữ liệu chuỗi. LSTM được giới thiệu bởi Sepp Hochreiter và Jürgen Schmidhuber vào năm 1997 và đã trở thành một trong những

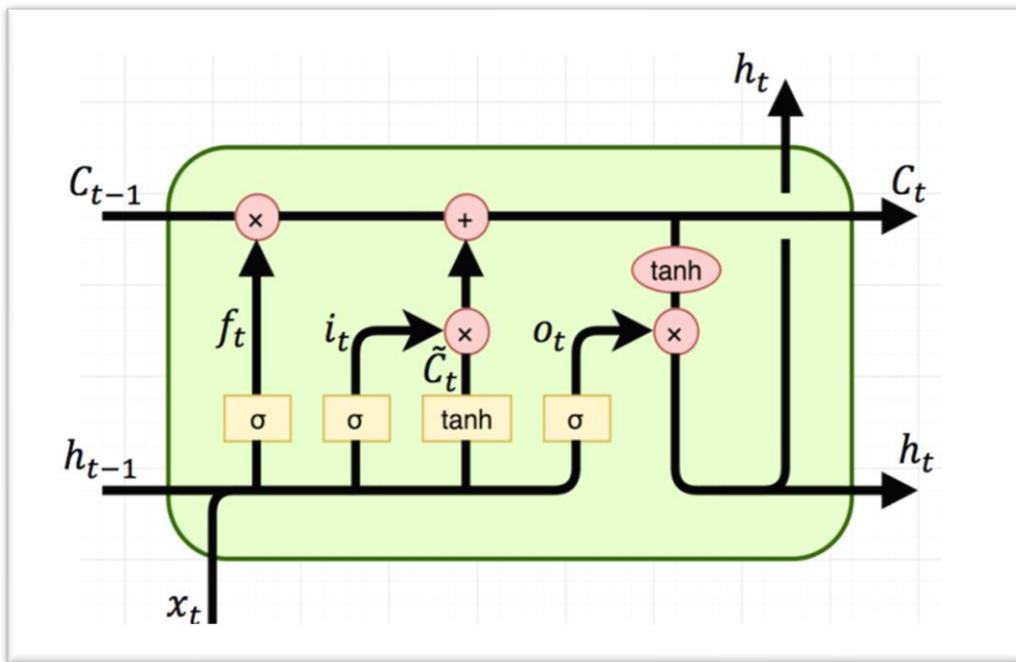
kiến trúc phổ biến nhất trong xử lý ngôn ngữ tự nhiên, dịch máy, nhận diện giọng nói, và nhiều ứng dụng khác liên quan đến dữ liệu chuỗi.



Hình 3. Kiến trúc LSTM

Một tế bào LSTM điển hình bao gồm các thành phần chính sau:

1. Cell State (Trạng Thái Tế Bào): Đây là thành phần chính lưu trữ thông tin dài hạn. Nó chạy xuyên suốt qua các bước thời gian và có khả năng giữ thông tin cần thiết hoặc loại bỏ thông tin không cần thiết.
2. Hidden State (Trạng Thái Ân): Đây là đầu ra ngắn hạn của LSTM tại mỗi bước thời gian, chứa thông tin về trạng thái hiện tại của chuỗi.
3. Cổng (Gates): LSTM sử dụng ba loại cổng để kiểm soát luồng thông tin:
 - Forget Gate (Cổng Lãng Quên): Quyết định thông tin nào từ cell state cần được loại bỏ.
 - Input Gate (Cổng Nhập): Quyết định thông tin nào từ đầu vào hiện tại và hidden state cũ cần được cập nhật vào cell state.
 - Output Gate (Cổng Xuất): Quyết định thông tin nào từ cell state sẽ được xuất ra như hidden state mới.



Hình 4. Sơ đồ của một tế bào LSTM

Ở state thứ t của mô hình LSTM:

- Output: c_t, h_t , ta gọi c là cell state, h là hidden state.
- Input: c_{t-1}, h_{t-1}, x_t . Trong đó x_t là input ở state thứ t của model. c_{t-1}, h_{t-1} là output của layer trước. h đóng vai trò khá giống như s ở RNN, trong khi c là điểm mới của LSTM.

Các đọc biểu đồ trên: bạn nhìn thấy kí hiệu σ , tanh ý là bước đáy dùng sigma, tanh activation function. Phép nhân ở đây là element-wise multiplication, phép cộng là cộng ma trận.

f_t, i_t, o_t tương ứng với forget gate, input gate và output gate.

- Forget gate: $f_t = \sigma(U_f * x_t + W_f * h_{t-1} + b_f)$
- Input gate: $i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i)$
- Output gate: $o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o)$

2.2 Tổng Quan Về Rasa

Rasa là một nền tảng mã nguồn mở chuyên dụng để phát triển các hệ thống chatbot thông minh và các ứng dụng xử lý ngôn ngữ tự nhiên (NLP). Được thiết kế để cung cấp các công cụ và khả năng tùy chỉnh mạnh mẽ, Rasa cho phép các nhà phát triển xây dựng chatbot có khả năng hiểu và tương tác với người dùng một cách tự nhiên, phù hợp với nhiều ngữ cảnh khác nhau. Điều này giúp tăng trải nghiệm người dùng và cải thiện hiệu suất của các hệ thống tự động. Rasa hỗ trợ tích hợp nhiều dịch

vụ bên ngoài và có khả năng mở rộng, tạo điều kiện cho việc phát triển chatbot trong môi trường sản xuất quy mô lớn. Rasa bao gồm hai thành phần chính là Rasa NLU và Rasa Core

2.3 Rasa NLU (Natural Language Understanding)

Rasa NLU (Natural Language Understanding) và Rasa Core, phối hợp với nhau để tạo ra một hệ thống chatbot hoàn chỉnh và linh hoạt: Rasa NLU: Được sử dụng để phân tích và hiểu các ý định (intent) và thực thể (entity) trong các câu hỏi của người dùng. Đây là bước quan trọng trong việc giải mã thông tin từ ngôn ngữ tự nhiên, giúp chatbot nhận diện các yếu tố quan trọng từ văn bản và hiểu rõ mục đích của người dùng. Rasa NLU có thể sử dụng các mô hình máy học hoặc các phương pháp dựa trên quy tắc để cải thiện độ chính xác trong việc nhận diện ý định và thực thể.

Rasa NLU bao gồm một số thành phần chính như sau:

2.3.1 Tokenization

Tokenization là bước đầu tiên trong quá trình xử lý ngôn ngữ tự nhiên (NLP). Đây là quá trình chia câu hoặc văn bản của người dùng thành các đơn vị nhỏ hơn, gọi là tokens. Các tokens có thể là từ, cụm từ, hoặc ký tự đơn lẻ. Quá trình này rất quan trọng vì nó giúp các mô hình xử lý và phân tích văn bản một cách dễ dàng hơn.

Các loại token: Từ (Word Tokens): Là các đơn vị cơ bản nhất có nghĩa trong văn bản. Ví dụ, trong câu "Tôi thích học lập trình", các tokens là "Tôi", "thích", "học", "lập", "trình".

Cụm từ (Phrase Tokens): Các nhóm từ được coi là đơn vị cơ bản. Ví dụ, "lập trình Python" có thể được coi là một cụm từ token.

Ký tự đơn lẻ (Character Tokens): Trong một số ngôn ngữ hoặc ứng dụng, mỗi ký tự trong văn bản có thể được xem là một token riêng biệt. Ví dụ, trong từ "NLP", các tokens có thể là "N", "L", "P".

Các kỹ thuật tokenization trong Rasa:

- WhitespaceTokenizer: Tách các từ dựa trên khoảng trắng. Phương pháp này đơn giản và thường dùng cho các ngôn ngữ có khoảng trắng phân tách rõ ràng các từ, như tiếng Anh.
- RegexFeaturizer: Sử dụng biểu thức chính quy (regex) để nhận diện các token cụ thể. Phương pháp này linh hoạt và có thể tùy chỉnh theo các quy tắc cụ thể.
- LexicalSyntacticFeaturizer: Tạo các đặc trưng từ các tokens dựa trên cấu trúc cú pháp và từ vựng. Phương pháp này giúp hiểu rõ hơn về cấu trúc ngữ pháp và mối quan hệ giữa các tokens.

- Tokenization là một bước quan trọng trong xử lý ngôn ngữ tự nhiên, giúp chuyển đổi văn bản từ dạng thô thành dạng có thể xử lý và phân tích dễ dàng hơn. Điều này không chỉ giúp cải thiện hiệu suất của các mô hình NLP mà còn tạo điều kiện thuận lợi cho việc thực hiện các nhiệm vụ phân tích và sinh dữ liệu hiệu quả.

2.3.2 Intent Classification (Phân Loại Ý Định)

Phân loại ý định là quá trình xác định mục tiêu hoặc hành động mà người dùng muốn chatbot thực hiện. Các intent có thể là những mục tiêu khác nhau mà người dùng mong đợi, ví dụ như "đặt vé", "hỏi thời tiết", hoặc "hủy đơn hàng".

Rasa NLU sử dụng các mô hình học máy để phân loại ý định. Các kỹ thuật phổ biến được sử dụng bao gồm:

SVM (Support Vector Machines): Một thuật toán phân loại hiệu quả. Random Forests: Thuật toán cây quyết định được sử dụng để tăng độ chính xác.

Neural Networks (Mạng Nơ-ron): Mô hình mạnh mẽ trong việc học và phân loại từ dữ liệu. Quá trình này dựa vào việc huấn luyện mô hình trên một tập dữ liệu bao gồm các câu hỏi mẫu tương ứng với từng ý định. Khi mô hình đã được huấn luyện, nó sẽ phân tích và xác định ý định của người dùng dựa trên văn bản đầu vào.

2.3.3 Entity Recognition (Nhận Diện Thực Thể)

Nhận diện thực thể là quá trình trích xuất các thông tin cụ thể từ văn bản người dùng, như tên, địa chỉ, số điện thoại, ngày giờ, hoặc các thông tin quan trọng khác. Entities là các thành phần quan trọng giúp chatbot có thể hiểu ngữ cảnh chi tiết và cung cấp các phản hồi chính xác.

Rasa hỗ trợ nhiều loại entities khác nhau, chẳng hạn:

- Predefined entities (thực thể định sẵn): Các thực thể phổ biến như ngày tháng, số lượng, tiền tệ, và địa chỉ.
- Custom entities (thực thể tùy chỉnh): Các thực thể do người dùng định nghĩa dựa trên ngữ cảnh hoặc yêu cầu cụ thể của chatbot.
- Rasa NLU sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên, như:
- Conditional Random Fields (CRF), để nhận diện và gán nhãn cho các thực thể trong câu.

2.3.4 Synonyms and Regex Features (Từ Đồng Nghĩa và Đặc Trưng Biểu Thức Chính Quy)

Từ Đồng Nghĩa (Synonyms): Từ đồng nghĩa là các từ hoặc cụm từ khác nhau nhưng có ý nghĩa giống nhau hoặc tương tự trong một ngữ cảnh cụ thể. Trong hệ

thống xử lý ngôn ngữ tự nhiên, việc nhận diện và xử lý các từ đồng nghĩa là rất quan trọng để hiểu được ý nghĩa của người dùng một cách chính xác hơn. Ví dụ, trong một ngữ cảnh liên quan đến công nghệ, từ "laptop" và "máy tính xách tay" có thể được coi là từ đồng nghĩa.

Chức Năng của Từ Đồng Nghĩa: Chuẩn Hóa Dữ Liệu: Khi hệ thống Rasa NLU nhận diện một từ hoặc cụm từ, nó có thể chuyển đổi từ đó thành một giá trị chuẩn hóa. Điều này giúp hệ thống dễ dàng xử lý và phân tích các yêu cầu của người dùng mà không bị nhầm lẫn giữa các cách diễn đạt khác nhau.

Cải Thiện Hiệu Suất: Việc sử dụng từ đồng nghĩa giúp cải thiện độ chính xác của hệ thống trong việc nhận diện ý định và thực thể, bằng cách giảm sự phân mảnh và giúp hệ thống hiểu đúng ngữ nghĩa của các từ trong văn bản.

Regex features (Đặc trưng biểu thức chính quy) là các mẫu dựa trên biểu thức chính quy được sử dụng để nhận diện các kiểu dữ liệu cụ thể trong văn bản. Ví dụ: biểu thức chính quy có thể được sử dụng để nhận diện địa chỉ email, số điện thoại hoặc mã bưu điện. Rasa cho phép người dùng định nghĩa các biểu thức chính quy để cải thiện việc nhận diện thực thể.

2.3.5 Pipeline (Đường Ống Xử Lý)

Một thành phần quan trọng khác của Rasa NLU là pipeline, là tập hợp các bước xử lý và mô hình giúp phân tích và hiểu văn bản đầu vào. Pipeline bao gồm các bước từ tokenization đến intent classification và entity recognition. Mỗi bước trong pipeline có thể được tùy chỉnh để phù hợp với nhu cầu cụ thể của ứng dụng chatbot.

Rasa cung cấp nhiều thành phần sẵn có để xây dựng pipeline như:

WhitespaceTokenizer:

- **Chức Năng:** Tách các từ trong văn bản dựa trên khoảng trắng. Đây là bước đầu tiên trong pipeline giúp phân chia văn bản thành các đơn vị cơ bản gọi là tokens.
- **Lý Thuyết:** Tokenization là quá trình chia văn bản thành các đơn vị nhỏ hơn như từ hoặc cụm từ. WhitespaceTokenizer đơn giản và hiệu quả, đặc biệt với các ngôn ngữ mà khoảng trắng là dấu phân cách chính giữa các từ.

RegexFeaturizer:

- **Chức Năng:** Sử dụng biểu thức chính quy để nhận diện và trích xuất các mẫu cụ thể từ văn bản. Ví dụ, nhận diện số điện thoại, địa chỉ email, hoặc mã bưu điện.
- **Lý Thuyết:** Featurization là quá trình chuyển đổi tokens thành các đặc trưng mà mô hình học máy có thể sử dụng. RegexFeaturizer giúp nhận diện các kiểu dữ liệu đặc biệt bằng cách áp dụng các mẫu biểu thức chính quy để trích xuất thông tin từ văn bản.

LexicalSyntacticFeaturizer:

- Chức Năng: Tạo các đặc trưng từ các tokens dựa trên cấu trúc cú pháp và từ vựng. Điều này bao gồm việc phân tích ngữ nghĩa và cấu trúc của câu để tạo ra các đặc trưng có ích cho việc phân loại và nhận diện thực thể.
- Lý Thuyết: Các đặc trưng ngữ nghĩa và cú pháp giúp mô hình hiểu được ngữ cảnh và mối quan hệ giữa các từ trong câu. LexicalSyntacticFeaturizer sử dụng thông tin về cấu trúc ngữ pháp và từ vựng để tạo ra các đặc trưng bổ sung cho mô hình học máy.

CountVectorsFeaturizer:

- Chức Năng: Chuyển đổi các tokens thành các vector đặc trưng dựa trên số lượng và n-grams. Đây là bước giúp mô hình có thể xử lý các chuỗi ký tự và phân tích mối quan hệ giữa các từ.
- Lý Thuyết: CountVectorsFeaturizer sử dụng các phương pháp phân tích n-grams để tạo ra các đặc trưng từ văn bản.

char_wb (character-level word boundaries)

- Chức năng: Tạo đặc trưng từ ký tự: Giúp mô hình hiểu và phân tích cấu trúc ký tự của từ, đặc biệt với từ có cấu trúc phức tạp. Xử lý từ hiếm: Giúp nhận diện và phân loại từ mới hoặc hiếm bằng cách tạo đặc trưng từ ký tự. Giảm ảnh hưởng lỗi chính tả: Mô hình vẫn có thể nhận diện từ bị sai chính tả dựa trên n-gram ký tự. Cải thiện phân tích cú pháp: Tăng cường khả năng phân tích ngữ nghĩa và cú pháp của văn bản.
- Lý thuyết: Mô hình hóa ngôn ngữ: Cung cấp cách tiếp cận linh hoạt để phân tích ngôn ngữ thông qua ký tự. N-gram: Dựa trên lý thuyết n-gram, cho phép tính toán xác suất và phân phối ký tự. Xử lý ngôn ngữ tự nhiên: Cải thiện khả năng hiểu ngữ nghĩa của hệ thống, hữu ích trong các ứng dụng như chatbot và dịch máy.

DIETClassifier:

- Chức Năng: Mô hình phân loại đa nhiệm có khả năng phân loại ý định và nhận diện thực thể từ văn bản. DIETClassifier (Dual Intent and Entity Transformer) là một mô hình mạnh mẽ sử dụng các mạng nơ-ron để thực hiện cả hai nhiệm vụ này.
- Lý Thuyết: DIETClassifier là một mô hình học sâu được thiết kế để xử lý cả phân loại ý định và nhận diện thực thể trong một bước. Nó sử dụng kiến trúc transformer để học từ dữ liệu huấn luyện và cải thiện khả năng phân loại và nhận diện.

EntitySynonymMapper:

- Chức Năng: Chuyển đổi các thực thể thành các giá trị chuẩn hóa dựa trên danh sách từ đồng nghĩa. Điều này giúp hệ thống hiểu rằng các từ khác nhau có thể đại diện cho cùng một thực thể.
- Lý Thuyết: Nhận diện thực thể là quá trình trích xuất các thông tin cụ thể từ văn bản. EntitySynonymMapper giúp chuẩn hóa các giá trị thực thể bằng cách sử dụng các từ đồng nghĩa, từ đó cải thiện khả năng nhận diện thực thể và cung cấp phản hồi chính xác.

ResponseSelector:

- Chức Năng: Chọn phản hồi phù hợp từ một tập hợp các phản hồi có sẵn dựa trên ý định và ngữ cảnh của người dùng. Đây là bước cuối cùng trong pipeline, giúp hệ thống đưa ra các phản hồi chính xác và phù hợp.
- Lý Thuyết: ResponseSelector sử dụng các mô hình học máy để lựa chọn phản hồi từ một danh sách dự kiến dựa trên các đặc trưng đầu vào. Mô hình này cải thiện khả năng tương tác của chatbot bằng cách chọn các phản hồi phù hợp và tự nhiên hơn.

Ngoài các thành phần sẵn có, Rasa cho phép người dùng định nghĩa và thêm các thành phần tùy chỉnh vào pipeline. Điều này cho phép tích hợp các mô hình hoặc thư viện xử lý ngôn ngữ riêng để đáp ứng các yêu cầu đặc thù của doanh nghiệp.

2.4 Rasa Core

Rasa Core: Đảm nhận vai trò quản lý luồng hội thoại và đưa ra các phản hồi phù hợp dựa trên ngữ cảnh của cuộc trò chuyện. Rasa Core sử dụng các mô hình học máy, chẳng hạn như các thuật toán dự đoán hành động, để điều khiển cách thức chatbot phản hồi và tương tác với người dùng qua các bước trong cuộc hội thoại. Điều này giúp chatbot có khả năng xử lý các tình huống phức tạp, duy trì mạch hội thoại hợp lý và tạo ra trải nghiệm tương tác gần gũi hơn.

Dưới đây là những thành phần chính của Rasa Core

2.4.1 Dialogue Management (Quản Lý Hội Thoại)

Dialogue management là chức năng cốt lõi của Rasa Core, cho phép chatbot quản lý và điều khiển các luồng hội thoại phức tạp. Nó sử dụng các mô hình học máy để học hỏi từ dữ liệu huấn luyện về cách đối thoại và quyết định hành động tiếp theo dựa trên ngữ cảnh của cuộc trò chuyện. Dialogue management đảm bảo chatbot không chỉ phản hồi từng câu riêng lẻ mà còn có thể giữ ngữ cảnh của toàn bộ cuộc hội thoại, giúp trải nghiệm người dùng trở nên tự nhiên và mượt mà hơn.

2.4.2 Stories (Kịch Bản Hội Thoại)

Stories là các kịch bản hội thoại mẫu được sử dụng để huấn luyện Rasa Core. Mỗi story mô tả một chuỗi các tương tác giữa người dùng và chatbot, bao gồm ý định của người dùng và hành động của chatbot. Những kịch bản này giúp Rasa Core học cách quản lý các tình huống hội thoại khác nhau và quyết định hành động nào cần thực hiện dựa trên ngữ cảnh.

2.4.3 Policies (Chính Sách Ra Quyết Định)

Policies là các mô hình ra quyết định được Rasa Core sử dụng để dự đoán hành động tiếp theo dựa trên ngữ cảnh hội thoại hiện tại. Mỗi policy có một vai trò riêng và có thể được cấu hình để đáp ứng các yêu cầu cụ thể của hệ thống chatbot. Các loại policies chính bao gồm:

MemoizationPolicy: Ghi nhớ chính xác các chuỗi hội thoại đã học trong dữ liệu huấn luyện. Nếu hội thoại của người dùng khớp hoàn toàn với một kịch bản đã học, policy này sẽ đưa ra hành động chính xác mà nó đã ghi nhớ.

RulePolicy: Quy tắc cụ thể do người dùng định nghĩa, giúp xử lý các tình huống cố định trong hội thoại, chẳng hạn như kết thúc hội thoại khi người dùng yêu cầu hủy đơn hàng.

UnspecTEDIntentPolicy là một chính sách trong Rasa được sử dụng để xử lý các ý định (intents) không mong đợi hoặc không xác định. Chính sách này giúp chatbot phản ứng một cách hợp lý khi người dùng đưa ra yêu cầu mà hệ thống không nhận diện được.

TEDPolicy (Transformer Embedding Dialogue Policy): Đây là mô hình học sâu được sử dụng để ra quyết định dựa trên chuỗi các ý định và hành động trong hội thoại. TED sử dụng kiến trúc Transformer để học các mẫu phức tạp trong hội thoại, giúp chatbot đưa ra hành động chính xác ngay cả khi kịch bản hội thoại không giống hoàn toàn với dữ liệu huấn luyện.

2.4.4 Action (Hành Động)

Actions là các phản hồi hoặc thao tác mà chatbot thực hiện để phản hồi người dùng.

Có hai loại actions chính trong Rasa:

- Utter Actions: Các câu trả lời đơn giản mà chatbot gửi tới người dùng, chẳng hạn như utter_greet (chào hỏi), utter_farewell (chào tạm biệt). Các hành động này được định nghĩa trước trong kịch bản và không yêu cầu xử lý phức tạp.
- Custom Actions (Hành Động Tùy Chỉnh): Các hành động phức tạp hơn, yêu cầu tính toán hoặc truy xuất dữ liệu từ cơ sở dữ liệu hoặc API bên ngoài. Ví dụ, khi người dùng yêu cầu thông tin cụ thể (như số dư tài khoản), custom action có thể truy xuất dữ liệu từ cơ sở dữ liệu và gửi phản hồi phù hợp.

2.4.5 Slot Filling (Điền Giá Trị Cho Slots)

Slots trong Rasa là các biến lưu trữ thông tin từ cuộc hội thoại. Chúng có vai trò quan trọng trong việc duy trì ngữ cảnh và thực hiện hành động phù hợp. Slot filling là quá trình điền giá trị vào các slots từ thông tin người dùng cung cấp. Ví dụ, nếu người dùng cung cấp thông tin về ngày tháng, chatbot sẽ điền giá trị vào slot tương ứng để sử dụng trong các bước tiếp theo của hội thoại.

Các loại slots phổ biến:

- Categorical Slots: Lưu trữ các giá trị có giới hạn (ví dụ: lựa chọn "nam" hoặc "nữ")
- Text Slots: Lưu trữ bất kỳ chuỗi văn bản nào mà người dùng nhập.
- Boolean Slots: Lưu trữ giá trị đúng hoặc sai. List Slots: Lưu trữ nhiều giá trị cùng một lúc (ví dụ: danh sách các sản phẩm mà người dùng quan tâm).

2.4.6 Event Tracking (Theo Dõi Sự Kiện)

Event tracking cho phép Rasa Core theo dõi mọi sự kiện diễn ra trong suốt cuộc hội thoại, bao gồm các ý định của người dùng, hành động của chatbot, và các thay đổi về giá trị slots. Dựa trên lịch sử sự kiện này, chatbot có thể đưa ra quyết định tốt hơn trong tương lai, duy trì ngữ cảnh cuộc trò chuyện, và cung cấp các phản hồi nhất quán.

Tổng kết chương 2:

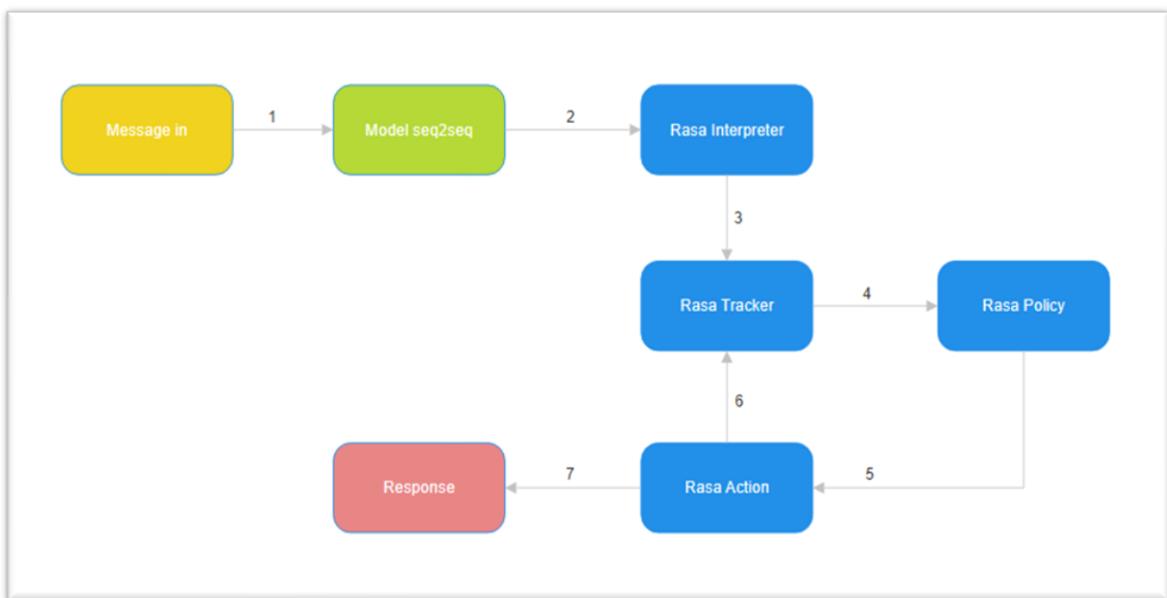
Rasa NLU là một thành phần mạnh mẽ của hệ thống chatbot, cung cấp khả năng phân tích ngữ nghĩa và ngữ cảnh của các yêu cầu người dùng. Nhờ vào các thành phần như tokenization, intent classification, entity recognition, và khả năng tùy chỉnh linh hoạt trong pipeline, Rasa NLU có thể hiểu sâu sắc văn bản và đưa ra các phản hồi chính xác, đáp ứng nhu cầu đa dạng của doanh nghiệp.

Rasa Core là một thành phần mạnh mẽ, đảm bảo rằng chatbot không chỉ hiểu được ý định của người dùng mà còn có thể phản hồi hợp lý, giữ ngữ cảnh trong các cuộc hội thoại phức tạp. Thông qua các thành phần như stories, policies, actions, và slot filling, Rasa Core giúp xây dựng các hệ thống chatbot thông minh, linh hoạt, và có thể mở rộng, đáp ứng tốt các yêu cầu khác nhau của doanh nghiệp.

CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN

Trong chương này, chúng ta sẽ tìm hiểu quy trình xây dựng và triển khai một hệ thống chatbot sử dụng nền tảng Rasa, đặc biệt chú trọng vào việc ứng dụng mô hình Sequence-to-Sequence (Seq2Seq) trong việc xử lý và chỉnh sửa văn bản tiếng Việt. Mô hình Seq2Seq là một kiến trúc mạnh mẽ, được thiết kế để biến đổi một chuỗi đầu vào thành một chuỗi đầu ra, rất hữu ích trong nhiều tác vụ như dịch máy, tóm tắt văn bản và chỉnh sửa văn bản.

Dưới đây là sơ đồ hoạt động từ khi nhận tin nhắn từ người dùng đưa vào mô hình sửa lỗi chính tả sau đó chuyển vào rasa để đưa ra phản hồi phù hợp với yêu cầu người dùng.

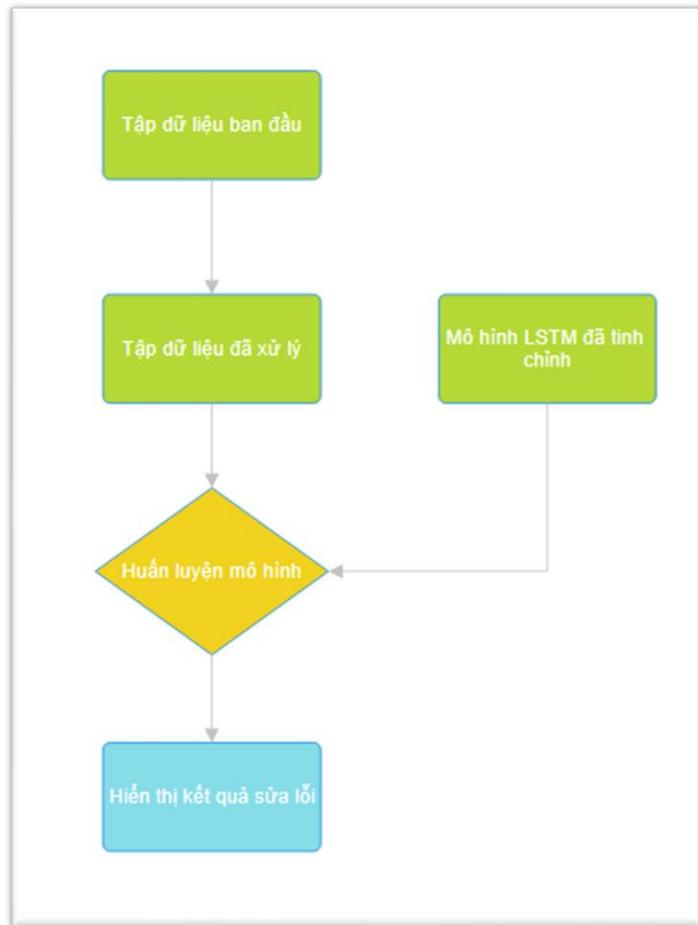


Hình 5. Sơ đồ hoạt động hệ thống sửa lỗi chính tả và chatbot

Trong Hình 5, ngoại trừ bước 1 và 2 được thực hiện bởi Rasa NLU và Seq2Seq, thì tất cả các bước đều được xử lý bởi Rasa Core. Sau khi nhận được tin nhắn từ người dùng và chuyển tiếp đến trình thông dịch để trích xuất ý định, thực thể và các thông tin cần thiết, Rasa NLU và tracker sẽ theo dõi, phát hiện, và duy trì trạng thái của ngữ cảnh hội thoại thông qua các thông báo tin nhắn đã nhận được. Sau đó, đầu ra của Tracker (trạng thái ngữ cảnh) sẽ đi vào trình quản lý chính sách và chính sách sẽ quyết định đến hành động tiếp theo. Ở bước 6, tracker sẽ ghi lại tất cả hành động trước khi chúng được thực thi và gửi đến người dùng. Các phản hồi đến người dùng đều được định trước trong tệp domain.yml, nơi xác định mọi thứ bao gồm ý định, thực thể, slots và hành động. Nếu người dùng bỏ qua các hành động đã thực hiện, quy trình sẽ quay trở lại bước thứ 4.

3.1 Tổng quan về bài toán Seq2Seq

Trong nghiên cứu này, hệ thống sửa lỗi chính tả tiếng việt được thực hiện qua các giai đoạn chính theo sơ đồ sau:



Hình 6. Sơ đồ hoạt động hệ thống sửa lỗi chính tả

3.1.1 Thu thập và xử lý dữ liệu

Thu thập dữ liệu: Điều này bao gồm việc thu thập một tập dữ liệu gồm các câu văn bản từ các trang web. Dữ liệu này phải được làm sạch và chuẩn hóa trước khi sử dụng cho mô hình.

Dưới đây là ví dụ về mã nguồn sử dụng BeautifulSoup để thu thập các bài báo từ trang VnExpress và lưu trữ dữ liệu dưới dạng câu.

```

from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
# Đường dẫn đến trang web gốc
URL = "https://vnexpress.net"

def get_links(url):
    links = []
    html = urlopen(url).read()
    soup = BeautifulSoup(html, 'html.parser')
    # Lấy tất cả đường dẫn trong trang web hiện tại
    urls = set(re.findall(
        r'^(?:http|https|ftp)://(?:[\.]+\.(?:[\.]+\.)+)+)(?:[^\w@?^=%&/~#+-]*[\w@?^=%&/~#+-])?', str(soup)))

    for url in urls:
        # Chỉ lấy đường dẫn là trang web chưa bài báo
        if url != URL and url.startswith(URL) and not url.startswith(URL + '/tac-gia') and url.endswith('.html'):
            links.append(url)
    return links

```

Xử lý dữ liệu: Hàm `_get_content` được sử dụng để trích xuất nội dung từ một bài báo và làm sạch nội dung, xóa các ký tự thừa và khoảng trắng không cần thiết.

```

def _get_content(url):
    # Nội dung bài báo
    content = ''
    html = urlopen(url).read()
    soup = BeautifulSoup(html, 'html.parser')

    # Lấy phần chứa nội dung bài báo
    div_content = soup.select('.page-detail .container')
    if len(div_content) > 0:
        div_content = div_content[0]

    # Lấy phần mô tả
    description = div_content.find_all('p', {'class': 'description'})
    if len(description) > 0:
        description = description[-1]
        text_description = description.get_text()
        location = description.find('span', {'class': 'location-stamp'})
        if location is not None:
            content = text_description[len(location.get_text()):]
        else:
            content = text_description
    # Lấy phần chứa nội dung chi tiết bài báo
    detail = div_content.find('article', {'class': 'fck_detail'})
    if detail is not None:
        p_normal = detail.find_all('p', {'class': 'Normal'})
        if len(p_normal) > 0:
            for p in p_normal:
                p_text = p.get_text()
                if p_text != '' and not p_text.startswith('>>') and p_text[-1] in ['.', '!', '?']:
                    content += ' ' + p.get_text()
    # Xóa ký tự ngắt dòng và trả về
    return re.sub(r'\n', ' ', content)

```

Lưu nội dung được tách vào file corpus.txt là một tệp văn bản chứa tập dữ liệu được sử dụng để huấn luyện mô hình. Trong bối cảnh sửa lỗi chính tả, corpus thường bao gồm các câu, đoạn văn, hoặc từ ngữ được sử dụng làm đầu vào cho mô hình để mô phỏng quá trình học tập.

```
# Loại bỏ các câu trùng lặp
sentence = list(set(sentence))
print('Tổng số câu tách được:', len(sentence))
corpus_file_path = os.path.join(drive_path, 'corpus.txt')
file_corpus = open(corpus_file_path, 'w', encoding='utf-8')
with open(corpus_file_path, 'w', encoding='utf-8') as file_corpus:
    count = 0
    for sent in sentence:
        # Xóa bỏ ký tự thừa
        sent = re.sub(r'\u200b', '', sent)
        file_corpus.write(sent.strip() + '\n')
        count += 1
print('Số lượng câu trong file ngữ liệu: ', count)
```

Xử lý văn bản để tách thành các cụm từ (phrases) phù hợp để huấn luyện mô hình.

Chi tiết:

- Duyệt qua từng câu trong data.
- Thay thế hoặc loại bỏ các ký tự không thuộc alphabet:
- Sử dụng unidecode để chuyển các ký tự không thuộc alphabet thành dạng không dấu hoặc loại bỏ chúng.
- Sử dụng biểu thức chính quy re.findall(r'\w[\w\s]+', text) để tìm các cụm từ chứa các ký tự chữ cái và khoảng trắng.
- Loại bỏ các cụm từ có ít hơn 2 từ.
- In ra số lượng cụm từ sau khi xử lý.

```

phrases = []
for text in data:
    # Thay thế hoặc xóa bỏ các ký tự thừa
    for c in set(text):
        if re.match('\w', c) and c not in alphabet:
            uc = unidecode(c)
            if re.match('\w', uc) and uc not in alphabet:
                text = re.sub(c, ' ', text)
            else:
                text = re.sub(c, uc, text)
    phrases += re.findall(r'\w[\w\s]+', text)

phrases = [p.strip() for p in phrases if len(p.split()) > 1]
print("số đoạn: ", len(phrases))

```

Dữ liệu huấn luyện được xây dựng bằng cách chia các câu văn thành các n-grams có độ dài tối đa là 5 từ. Việc tạo n-gram giúp mô hình học được ngữ cảnh ngắn hạn trong các chuỗi từ để xử lý lỗi chính tả hiệu quả hơn.

```

NGRAM = 5
MAXLEN = 39
list_ngrams = []
for p in phrases:
    list_p = p.split()
    if len(list_p) >= NGRAM:
        for ngr in ngrams(p.split(), NGRAM):
            if len(' '.join(ngr)) <= MAXLEN:
                list_ngrams.append(' '.join(ngr))
    elif len(' '.join(list_p)) <= MAXLEN:
        list_ngrams.append(' '.join(list_p))

list_ngrams = list(set(list_ngrams))
print(len(list_ngrams))

```

Chuyển đổi văn bản thành định dạng one-hot encoding để sử dụng trong mô hình học máy.

Chi tiết:

- Tạo một mảng numpy x có kích thước (MAXLEN, len(alphabet)), khởi tạo bằng 0.
- Duyệt qua từng ký tự trong văn bản (tối đa MAXLEN ký tự):

- Đánh dấu vị trí của ký tự đó trong alphabet bằng cách đặt giá trị 1 tại vị trí tương ứng.
- Nếu số ký tự trong văn bản ít hơn MAXLEN, điền vào các vị trí còn lại bằng cách đặt 1 ở cột đầu tiên ('\x00').
- Trả về mảng mã hóa x.

```
def _encoder_data(text):
    x = np.zeros((MAXLEN, len(alphabet)))
    for i, c in enumerate(text[:MAXLEN]):
        x[i, alphabet.index(c)] = 1
    if i < MAXLEN - 1:
        for j in range(i+1, MAXLEN):
            x[j, 0] = 1
    return x
```

Chuyển đổi định dạng one-hot encoding trở lại thành văn bản.

Chi tiết:

- Tìm chỉ số có giá trị lớn nhất (1) trong mỗi hàng của mảng x bằng argmax.
- Chuyển các chỉ số này thành ký tự tương ứng trong alphabet.
- Nối các ký tự lại thành một chuỗi văn bản.

```
def _decoder_data(x):
    x = x.argmax(axis=-1)
    return ''.join(alphabet[i] for i in x)
```

3.1.2 Ứng dụng LSTM huấn luyện mô hình sửa lỗi chính tả

Xây dựng một mô hình mạng nơ-ron sâu để xử lý chuỗi ký tự và học cách chỉnh sửa văn bản.

Encoder:

- Sử dụng một lớp LSTM với 256 đơn vị.
- input_shape=(MAXLEN, len(alphabet)): Mỗi đầu vào có độ dài MAXLEN và mỗi ký tự được mã hóa bằng one-hot với kích thước len(alphabet).
- return_sequences=True: Trả về toàn bộ chuỗi đầu ra để có thể tiếp nối với decoder.

```
encoder = LSTM(256, input_shape=(MAXLEN, len(alphabet)), return_sequences=True)
```

Decoder:

- Sử dụng một lớp LSTM hai chiều (Bidirectional) với 256 đơn vị.
- dropout=0.2: Áp dụng dropout để giảm overfitting.
- return_sequences=True: Trả về toàn bộ chuỗi đầu ra.
- Activation('softmax'): Hàm kích hoạt Softmax để xác định xác suất của mỗi ký tự trong alphabet.

```
decoder = Bidirectional(LSTM(256, return_sequences=True, dropout=0.2))
```

Các lớp tiếp theo:

- TimeDistributed(Dense(256)): Áp dụng một lớp Dense cho mỗi thời điểm trong chuỗi.
- Activation('relu'): Hàm kích hoạt ReLU.
- TimeDistributed(Dense(len(alphabet))): Lớp Dense để chuyển đổi lại thành kích thước của alphabet.

```
model.add(TimeDistributed(Dense(256)))
model.add(Activation('relu'))
model.add(TimeDistributed(Dense(len(alphabet))))
model.add(Activation('softmax'))
```

Chia dữ liệu thành hai tập: huấn luyện (80%) và kiểm tra (20%) để đánh giá hiệu suất mô hình.

Chi tiết:

- Sử dụng train_test_split từ thư viện sklearn để chia dữ liệu.
- test_size=0.2: 20% dữ liệu dành cho tập kiểm tra.
- random_state=42: Đặt seed để đảm bảo kết quả chia dữ liệu có thể tái lập.

```
from sklearn.model_selection import train_test_split
train_data, valid_data = train_test_split(list_ngrams, test_size=0.2, random_state=42)
```

Tạo các batch dữ liệu để huấn luyện mô hình mà không cần tải toàn bộ dữ liệu vào bộ nhớ cùng một lúc.

Chi tiết:

- data: Dữ liệu nguồn (tập huấn luyện hoặc kiểm tra).
- batch_size: Kích thước của mỗi batch.
- Vòng lặp vô hạn (while True) để liên tục cung cấp dữ liệu cho mô hình.
- Trong mỗi batch:
 - y: Đối tượng mục tiêu (dữ liệu gốc được mã hóa).
 - x: Đầu vào (dữ liệu đã được thêm nhiễu và mã hóa).
 - Nếu đạt đến cuối dữ liệu, quay lại đầu danh sách.

```

BATCH_SIZE = 512
EPOCHS = 10
def _generate_data(data, batch_size):
    current_index = 0
    while True:
        x, y = [], []
        for i in range(batch_size):
            y.append(_encoder_data(data[current_index]))
            x.append(_encoder_data(_add_noise(data[current_index])))
            current_index += 1
            if current_index > len(data) - 1:
                current_index = 0
        yield (np.array(x), np.array(y))

```

Tạo các generators để cung cấp dữ liệu cho quá trình huấn luyện và kiểm tra.
Chi tiết:

- BATCH_SIZE = 512: Mỗi batch chứa 512 mẫu dữ liệu.
- EPOCHS = 10: Số lần lặp qua toàn bộ dữ liệu huấn luyện.

```

train_generator = _generate_data(train_data, batch_size = BATCH_SIZE)
validation_generator = _generate_data(valid_data, batch_size = BATCH_SIZE)

```

model.fit(): Đây là hàm dùng để huấn luyện mô hình dựa trên dữ liệu đầu vào. Hàm này sẽ thực hiện quá trình học bằng cách cập nhật trọng số mô hình dựa trên thuật toán tối ưu hóa đã chọn.

Các tham số chính trong model.fit():

- train_generator: Là bộ sinh (generator) dữ liệu huấn luyện. Thay vì đưa trực tiếp một tập dữ liệu lớn vào bộ nhớ, generator này sẽ sinh lần lượt các batch dữ liệu để huấn luyện mô hình. Điều này hữu ích khi làm việc với tập dữ liệu lớn.
- epochs = EPOCHS: Số lần lặp lại toàn bộ dữ liệu huấn luyện. Mỗi epoch là một vòng lặp mà mô hình sẽ đi qua toàn bộ tập dữ liệu huấn luyện một lần. EPOCHS là một biến chứa số lượng epoch mong muốn.
- steps_per_epoch=len(train_data) // BATCH_SIZE: Số bước (steps) trong mỗi epoch. Mỗi step tương ứng với một batch dữ liệu được sinh ra từ train_generator. Giá trị này được tính bằng cách lấy tổng số mẫu trong tập huấn luyện chia cho kích thước batch (BATCH_SIZE), nghĩa là số lượng batch cần thiết để hoàn thành một epoch.

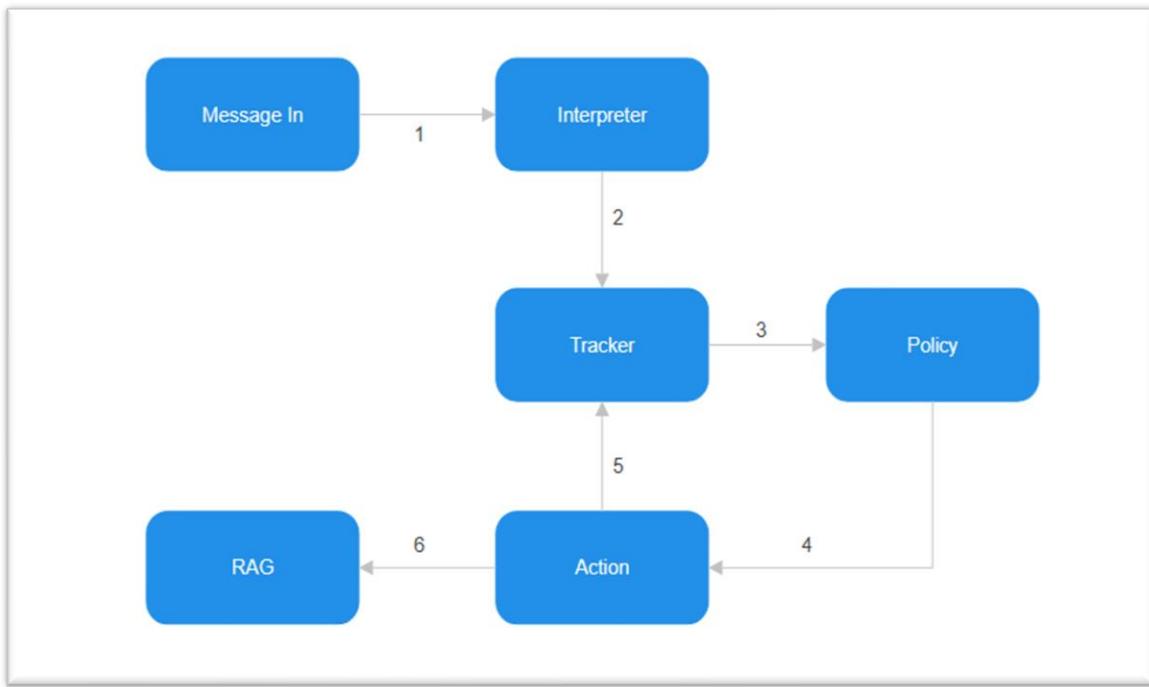
- validation_data=validation_generator: Bộ sinh dữ liệu để đánh giá mô hình sau mỗi epoch, từ đó kiểm tra mô hình có cải thiện qua từng epoch hay không. Generator này tương tự như train_generator, nhưng dành cho dữ liệu kiểm thử.
- validation_steps=len(valid_data) // BATCH_SIZE: Số bước tương tự như steps_per_epoch, nhưng áp dụng cho dữ liệu kiểm thử. Nó cho biết số batch cần thiết để đánh giá mô hình trên toàn bộ dữ liệu kiểm thử.

```
H = model.fit(  
    train_generator, epochs = EPOCHS,  
    steps_per_epoch=len(train_data) // BATCH_SIZE,  
    validation_data=validation_generator,  
    validation_steps=len(valid_data) // BATCH_SIZE  
)
```

```
model.save('./my_model.h5')
```

Khi hoàn tất quá trình huấn luyện, mô hình sẽ được đánh giá dựa trên tập dữ liệu kiểm tra.

3.2 Tổng quan bài toán Rasa



Hình 7. Sơ đồ hệ thống chatbot Rasa

1. Xử lý thông điệp (NLU)

- Đầu vào: Tin nhắn của người dùng được gửi đến Interpreter (bộ xử lý ngôn ngữ tự nhiên của Rasa - Rasa NLU).
- Chuyển đổi: Bộ xử lý NLU sẽ chuyển đổi tin nhắn của người dùng thành một đầu ra có cấu trúc, bao gồm văn bản gốc, ý định (intents) và thực thể (entities).

2. Theo dõi cuộc hội thoại (Tracker)

- Theo dõi: Bộ Tracker theo dõi trạng thái của cuộc hội thoại và nhận biết sự xuất hiện của tin nhắn mới.

3. Chuyển sang chính sách (Policy)

- Đầu ra của Tracker: Đầu ra từ Tracker sẽ được chuyển sang Policy (chính sách), nơi tiếp nhận trạng thái hiện tại của Tracker.

4. Chọn hành động tiếp theo

- Chính sách chọn hành động: Dựa trên trạng thái hiện tại của Tracker, Policy sẽ quyết định hành động tiếp theo mà chatbot cần thực hiện.

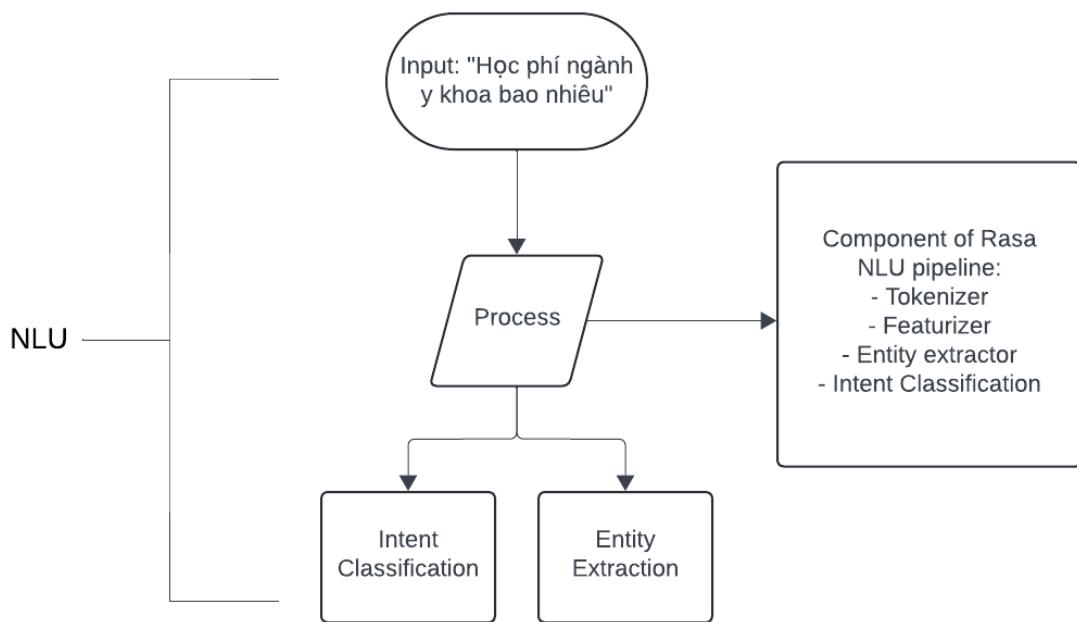
5. Ghi lại hành động

- Ghi log: Tracker sẽ ghi lại hành động đã chọn để duy trì lịch sử hội thoại và trạng thái của nó.

6. Phản hồi cho người dùng

- Gửi phản hồi: Hệ thống sẽ gửi phản hồi đến người dùng, sử dụng các câu trả lời được định nghĩa trước (utterances) trong tệp domain.yml hoặc các tệp câu hình khác.

3.2.1 Xử lý thông điệp (NLU)



Hình 8. Lược đồ phân loại ý định và trích xuất thực thể sử dụng Rasa NLU

```

language: vi
pipeline:
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: char_wb
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 100
    constrain_similarities: true
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
    constrain_similarities: true

```

Cấu hình tiếng Việt nhờ thành phần trong pipeline

Cấu hình tiếng Việt nhờ thành phần trong pipeline (đặc biệt là tokenizer, featurizer và classifier) đều hoạt động trên các cấu trúc cơ bản của văn bản. Với cách thức xử lý dựa trên n-gram ký tự, mô hình DIETClassifier đa ngôn ngữ, và huấn luyện bằng dữ liệu tiếng Việt, Rasa có thể nhận diện ý định và thực thể tiếng Việt tương đối hiệu quả.

Trước khi huấn luyện mô hình, ta cần chuẩn bị dữ liệu ngôn ngữ tự nhiên bao gồm các câu ví dụ (examples) cùng với ý định (intents) và các thực thể (entities) liên quan. Đây là bước quan trọng để giúp mô hình học cách nhận diện ý định và thực thể từ các câu đầu vào của người dùng.

File nlu.yml

```

intent: ask_hocphi_nganh
examples:
  - [học phí]{"entity":"hocphi", "role":"hoi_hocphi"} của ngành [y khoa]{"entity":"khoa_phong", "role":"ten_nganh", "value":"Y khoa"}
  - [chi phí]{"entity":"hocphi", "role":"hoi_hocphi"} [bác sĩ yk]{"entity":"khoa_phong", "role":"ten_nganh", "value":"Y khoa"}
  - [giá tiền]{"entity":"hocphi", "role":"hoi_hocphi"} [y khoa]{"entity":"khoa_phong", "role":"ten_nganh", "value":"Y khoa"}
  - [lệ phí]{"entity":"hocphi", "role":"hoi_hocphi"} [bac sĩ y khoa]{"entity":"khoa_phong", "role":"ten_nganh", "value":"Y khoa"}
  - [kính phí]{"entity":"hocphi", "role":"hoi_hocphi"} ngành [y]{"entity":"khoa_phong", "role":"ten_nganh", "value":"Y khoa"}
  - [chi phí]{"entity":"hocphi", "role":"hoi_hocphi"} [bs y khoa]{"entity":"khoa_phong", "role":"ten_nganh", "value":"Y khoa"}

```

Các thành phần chính trong file nlu.yml:

- Intent: Đây là ý định mà người dùng muốn thực hiện. Ví dụ: hỏi_học_phí, hỏi_chương_trình_đào_tạo, hỏi_thủ_tục,....
- Examples: Là các ví dụ thực tế về cách người dùng có thể nhập thông tin. Những câu ví dụ này sẽ được dùng để huấn luyện mô hình phân loại ý định và nhận diện thực thể.
- Entities: Thực thể trong câu ví dụ, thường là những thông tin quan trọng như địa điểm, thời gian. Trong ví dụ trên, [học phí](hocphi) có role là “hỏi học phí” và [y khoa](khoa_phong) có role là “ten_nganh” là các thực thể đã được gán nhãn.
- Role (vai trò) hoặc Group (nhóm) của các thực thể trong câu. Điều này đặc biệt hữu ích khi các thực thể có cùng loại nhưng đảm nhiệm các vai trò khác nhau trong các ngữ cảnh khác nhau.

Tạo thêm các intent và examples: Bạn cần cung cấp nhiều ví dụ về các câu mà người dùng có thể sử dụng cho mỗi ý định để mô hình có đủ dữ liệu để học. Thực thể cần được đánh dấu rõ ràng trong câu.

Lưu ý khi chuẩn bị dữ liệu:

- Đảm bảo dữ liệu đầu vào bao gồm đủ số lượng câu ví dụ cho từng intent (ít nhất từ 10-20 câu cho mỗi intent).
- Đánh nhãn thực thể chính xác để mô hình có thể học được cách nhận diện.
- Phải sử dụng ngôn ngữ tự nhiên của người dùng (trong trường hợp này là tiếng Việt) để mô hình có thể phản ánh chính xác cách người dùng thực sự giao tiếp.
- Chỉ định ý định, thực thể, vị trí, phản hồi, biểu mẫu và hành động mà bot của bạn nên biết. Nó cũng xác định cấu hình cho các phiên hội thoại.

File doomain.yml

Các phần chính gồm có:

- Intents
- Entities
- Actions
- Slots

```
intents:  
  - ask_hocphi  
  - ask_hocphi_nganh  
  - ask_CTDT  
  - ask_CTDT_nganh  
  - ask_daihoc_lienthong_saudaihoc  
  - ask_thong_tin_tuyen_sinh  
  - ask_thong_tin_tuyen_sinh_tung_nganh  
  - ask_diem_chuan_tung_nganh
```

```
entities:  
  - khoa_phong:  
    |   roles:  
    |     - ten_khoa_phong  
    |     - ten_nganh
```

```
actions:
  - action_goi_y_hocphi
  - action_hocphi_nganh
  - action_goi_y_CTDT
  - action_CTDT_nganh
  - action_goi_y_thong_tin_tuyen_sinh
  - action_thong_tin_tuyen_sinh_tung_nganh
  - action_diem_chuan_tung_nganh
```

```
slots:
  hocphi_nganh:
    type: text
    influence_conversation: true
    mappings:
      - type: from_entity
        entity: khoa_phong
        role: ten_nganh
```

File stories.yml

Tạo liên kết các ý định (intents) với các hành động (actions) hoặc phản hồi (responses) cụ thể mà chatbot cần thực hiện.

```
- story: Hỏi thông tin học phí chung
  steps:
    - intent: ask_hocphi
    - action: action_goi_y_hocphi

- story: Hỏi học phí ngành cụ thể
  steps:
    - intent: ask_hocphi_nganh
    - slot_was_set:
      - hocphi_nganh
    - action: action_hocphi_nganh
```

Sau khi chuẩn bị xong dữ liệu trong tệp nlu.yml, domain.yml, stories.yml bạn tiến hành huấn luyện mô hình. Mô hình sẽ học từ dữ liệu bạn cung cấp để nhận diện ý định và thực thể trong các câu mới.

sử dụng lệnh sau để bắt đầu huấn luyện:

rasa train

Lệnh này sẽ bắt đầu quá trình huấn luyện dựa trên các tệp dữ liệu bạn đã chuẩn bị, bao gồm tệp nlu.yml (dữ liệu ý định và thực thể), stories.yml (dữ liệu câu chuyện) và domain.yml (tệp cấu hình các phản hồi và hành động).

Mô tả quá trình huấn luyện:

- Bước đầu tiên: Rasa sẽ phân tích và tải dữ liệu từ các tệp nlu.yml, stories.yml và domain.yml. Nếu có lỗi trong tệp YAML hoặc cấu hình, quá trình sẽ dừng lại và yêu cầu bạn sửa chữa.
- Bước tiếp theo: Mô hình sẽ được huấn luyện trên các ví dụ đã cung cấp. Các thành phần của pipeline trong tệp cấu hình (như DIETClassifier, ResponseSelector, v.v.) sẽ học cách phân loại ý định và nhận diện thực thể.
- Kết quả huấn luyện: Khi quá trình hoàn tất, một mô hình mới sẽ được tạo ra trong thư mục models/. Tệp mô hình này chứa tất cả các tham số đã học để giúp hệ thống phân tích câu và nhận diện ý định trong tương lai.

Thời gian huấn luyện:

- Trong quá trình huấn luyện mô hình, không tránh khỏi việc thử nghiệm mô hình nhiều lần để sửa lỗi, cải thiện độ chính xác của mô hình. Mỗi lần chạy thì phải duyệt lại hết các tập dữ liệu thì sẽ mất rất nhiều thời gian, thời gian huấn luyện mô hình có thể thay đổi tùy thuộc vào kích thước dữ liệu và sức mạnh xử lý của máy tính. Quá trình này có thể mất từ vài phút đến vài giờ.

```

NLU model loaded. Type a message and press enter to parse it.
Next message:
học phí ngành y khoa bao nhiêu?
{
    "text": "học phí ngành y khoa bao nhiêu?",  

    "intent": {  

        "name": "ask_hocphi_nganh",  

        "confidence": 1.0
    },  

    "entities": [  

        {  

            "entity": "hocphi",  

            "start": 0,  

            "end": 7,  

            "confidence_entity": 0.9999498128890991,  

            "role": "hoi_hocphi",  

            "confidence_role": 0.9999589920043945,  

            "value": "học phí",  

            "extractor": "DIETClassifier",  

            "processors": [  

                "EntitySynonymMapper"
            ]
        },
        {
            "entity": "khoa_phong",  

            "start": 14,  

            "end": 20,  

            "confidence_entity": 0.9999643564224243,  

            "role": "ten_nganh",  

            "confidence_role": 0.9998739957809448,  

            "value": "Y khoa",  

            "extractor": "DIETClassifier",  

            "processors": [  

                "EntitySynonymMapper"
            ]
        }
    ]
}

```

Kiểm thử mô hình rasa shell nlu

Mục tiêu chính của lệnh rasa shell nlu là kiểm tra xem mô hình đã được huấn luyện có thể hiểu đúng các câu đầu vào của người dùng hay không. Cụ thể, lệnh này sẽ kiểm tra:

- Mô hình có thể phân loại đúng ý định (intent) của câu nói.
- Mô hình có thể trích xuất chính xác các thực thể (entity) trong câu.

Kết quả trả về:

Khi bạn nhập một câu vào giao diện dòng lệnh, mô hình sẽ phân tích và trả về kết quả dựa trên quá trình huấn luyện. Kết quả sẽ được hiển thị ở dạng JSON, trong đó bao gồm các thành phần chính:

Intent (Ý định):

Mô hình sẽ dự đoán ý định của câu đầu vào và trả về tên của intent cùng với độ tin cậy (confidence). Độ tin cậy thể hiện mức độ tự tin của mô hình về dự đoán này (giá trị từ 0 đến 1).

Entities (Thực thể):

Nếu có thực thể nào được nhận diện từ câu, mô hình sẽ trích xuất và trả về danh sách các thực thể cùng với vị trí và giá trị của chúng trong câu.

3.2.2 Theo dõi cuộc hội thoại (Tracker)

Theo dõi: Bộ Tracker đóng vai trò là bộ nhớ của chatbot, theo dõi và lưu trữ trạng thái của cuộc hội thoại với người dùng. Mỗi khi có một tin nhắn mới từ người dùng, Tracker sẽ cập nhật trạng thái hiện tại dựa trên thông tin đã được xử lý từ NLU.

Chức năng chính:

- Lưu trữ lịch sử hội thoại: Tracker ghi lại tất cả các sự kiện trong cuộc hội thoại, bao gồm các tin nhắn của người dùng, hành động của chatbot, và các biến trạng thái.
- Quản lý trạng thái: Tracker duy trì trạng thái hiện tại của cuộc hội thoại, bao gồm các thông tin như ý định gần đây, thực thể đã trích xuất, và các biến tùy chỉnh khác.
- Nhận biết sự xuất hiện của tin nhắn mới: Khi có tin nhắn mới từ người dùng, Tracker sẽ nhận biết và cập nhật trạng thái dựa trên tin nhắn đó.

Câu lệnh rasa shell - -debug

```
2024-09-29 14:48:01 DEBUG rasa.core.policies.memoization - [debug ] memoization.predict.actions  tracker_states=[{}], {'user': {'intent': 'ask_hocphi_nganh', 'entities': ['hocphi', 'hocphi@hoi_hocphi', 'khoa_phong'], 'slots': {'hocphi_nganh': (1.0,), 'CTDT_nganh': (1.0,), 'ten_nganh': (1.0,)}, 'prev_ac cy_predict_action_probabilities': ''}
2024-09-29 14:48:01 DEBUG rasa.core.policies.ted_policy - TED predicted 'action_listen' based on user intent.
2024-09-29 14:48:01 DEBUG rasa.engine.graph - Node 'run_UnexpectedIntentPolicy' running 'UnexpectedIntentPolicy.predict_action_probabilities'.
2024-09-29 14:48:01 DEBUG rasa.core.policies.unexpected_intent_policy - Skipping predictions for UnexpectedIntentPolicy as either there is no event of type 'Useruttered', event's intent is new and not in domain or there is an event of type 'ActionExecuted' after the last 'UserUttered'.
2024-09-29 14:48:01 DEBUG rasa.engine.graph - Node 'select_prediction' running 'DefaultPolicyPredictionEnsemble.combine_predictions_from_kwargs'.
2024-09-29 14:48:01 DEBUG rasa.core.policies.ensemble - Predicted next action using TEDPolicy.
2024-09-29 14:48:01 DEBUG rasa.core.processor - Predicted next action 'action_listen' with confidence 1.00.
2024-09-29 14:48:01 DEBUG rasa.core.processor - [debug ] processor.actions.policy_prediction prediction_events={}
2024-09-29 14:48:01 DEBUG rasa.core.processor - [debug ] processor.actions.log      action_name=action_listen rasa_events={}
2024-09-29 14:48:01 DEBUG rasa.core.tracker_store - No event broker configured. Skipping streaming events.
2024-09-29 14:48:01 DEBUG rasa.core.lock_store - Deleted lock for conversation '6e56b70c3ef54a4ca3004dc26044ae'.
<br><b>Học phí của ngành Y khoa : <br><b>Hệ chính quy :<br>- Đào tạo trung tuyển chính quy : 1.371.000/tin chi<br>- Đào tạo theo nhu cầu xã hội (theo đặt hàng ) : 1.645.000/tin chi<br>- Đào tạo hệ quốc tế tiếng anh : 4.000.000/tin chi<br><b>Hệ liên thông :<br>- Đào tạo liên thông hệ chính quy : 1.371.000/tin chi<br>- Đào tạo liên thông theo nhu cầu xã hội : 1.371.000/tin chi
Your input -> ■
```

3.2.3 Custom action

Sau mỗi tin nhắn của người dùng, mô hình sẽ dự đoán hành động mà trợ lý sẽ thực hiện tiếp theo. Trong Rasa, custom actions cho phép bạn thực hiện các tác vụ phức tạp mà không thể đạt được chỉ với các phản hồi định nghĩa sẵn. Custom actions có thể bao gồm gọi API bên ngoài, truy vấn cơ sở dữ liệu, hoặc thực hiện các logic tùy chỉnh khác để tạo ra phản hồi động cho người dùng.

```
from typing import Any, Text, Dict, List
from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher
from rasa_sdk.events import SlotSet
```

typing: Cung cấp các kiểu dữ liệu cho các tham số và giá trị trả về, giúp mã rõ ràng và dễ hiểu hơn.

rasa_sdk.Action: Lớp cơ sở để định nghĩa các custom actions.

rasa_sdk.Tracker: Lưu trữ trạng thái của cuộc hội thoại, bao gồm các thông điệp, intent, entities, slots, v.v.

rasa_sdk.executor.CollectingDispatcher: Dùng để gửi phản hồi từ custom action đến người dùng.

rasa_sdk.events.SlotSet: Sử dụng để thiết lập giá trị cho các slot trong Tracker.

```
class actionGoiYHocPhi(Action):
    def name(self):
        return "action_goi_y_hocphi"
    def run(self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        print("action Gọi ý học phí")
        user_message = tracker.latest_message.get('text')
        print("user_ask: " + user_message)
        entities = tracker.latest_message['entities']
        print(f"Entity từ input : {entities}")
        role = []
        for entity in entities:
            if entity.get('role') not in role:
                role.append(entity.get('role'))
        print("mảng role: ", role)
        response = "Tôi có thông tin học phí của các ngành:  
- Y khoa,  
- Răng hàm mặt,  
- Y học dự phòng  
- Y học cổ truyền  
- Điều dưỡng  
- Y tế công cộng  
- Kỹ thuật xét nghiệm y học  
- Hộ sinh  
- Kỹ thuật hình ảnh y học"
        dispatcher.utter_message(text=response)
        return [SlotSet("hocphi_nganh", None)]
```

class actionGoiYHocPhi(Action): Định nghĩa một lớp mới tên là actionGoiYHocPhi kế thừa từ lớp Action của Rasa.

def name(self): Phương thức bắt buộc trong custom action để trả về tên của hành động. Tên này phải trùng khớp với tên được định nghĩa trong tệp domain.yml.

return "action_goi_y_hocphi": Trả về tên của hành động. Đây là tên sẽ được sử dụng trong các câu chuyện (stories) hoặc rules để gọi hành động này.

def run(self, dispatcher, tracker, domain): Phương thức chính thực hiện logic của custom action. Nó nhận vào ba tham số:

- dispatcher: Đối tượng để gửi phản hồi trở lại người dùng.
- tracker: Đối tượng theo dõi trạng thái cuộc hội thoại, bao gồm các thông tin như intent, entities, slots, v.v.
- domain: Từ điển chứa thông tin về domain của chatbot, bao gồm intents, entities, slots, actions, v.v.
- List[Dict[Text, Any]]: Chỉ định kiểu dữ liệu trả về, trong trường hợp này là một danh sách các từ điển chứa các sự kiện (events).

tracker.latest_message.get('text'): Lấy văn bản của tin nhắn gần nhất mà người dùng đã gửi.

dispatcher.utter_message(text=response): Sử dụng dispatcher để gửi phản hồi đã định nghĩa đến người dùng. Tham số text=response chỉ định nội dung của tin nhắn phản hồi.

return [SlotSet("hocphi_nganh", None)]: Trả về một danh sách các sự kiện. Trong trường hợp này, sự kiện SlotSet được sử dụng để đặt giá trị của slot hocphi_nganh thành None, tức là xóa giá trị hiện tại của slot này. Điều này có thể được sử dụng để reset trạng thái hoặc chuẩn bị cho cuộc hội thoại tiếp theo.

Tổng kết chương 3:

Chương 3 đã trình bày cách cấu hình, huấn luyện mô hình Rasa để trả lời tự động bằng tiếng Việt, tích hợp mô hình sửa lỗi chính tả Seq2Seq và xây dựng giao diện chat trên website, tạo nền tảng cho một chatbot hỗ trợ hiệu quả và chính xác hơn.

Chương tiếp theo, luận văn giới thiệu tập dữ liệu sử dụng để nghiên cứu, môi trường thực nghiệm và kết quả thực nghiệm của mô hình Seq2Seq và chatbot Rasa.

Chương 4: Kết quả thực nghiệm

4.1 Thu thập và xây dựng tập dữ liệu

Thu thập dữ liệu huấn luyện mô hình sửa lỗi chính tả phổ biến:

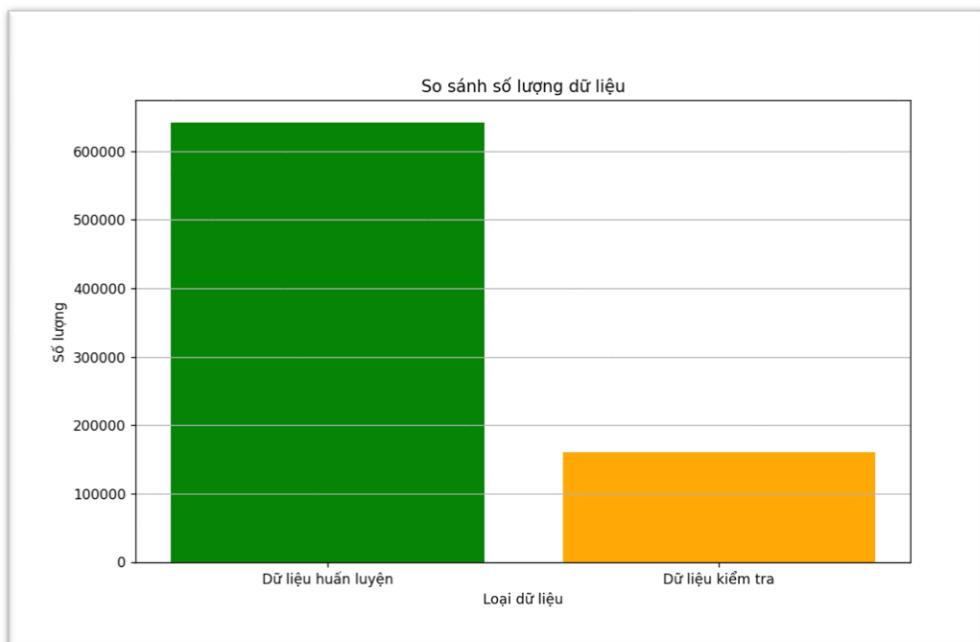
Trong đề tài này, dữ liệu được thu thập từ 2 nguồn chính:

1. thu thập một tập dữ liệu gồm các dữ liệu thu thập từ trang web “<https://vnexpress.net>”: với số lượng là 1390921 từ vựng, được gom nhóm thành các N-gram để phục vụ cho việc huấn luyện mô hình.
2. danh sách các nguyên âm tiếng Việt bao gồm cả chữ hoa và chữ thường, cũng như các dấu phụ với danh sách các phụ âm tiếng Việt và các từ viết tắt (teencode)

Tổng số lượng các N-gram được xây dựng là 802,773.

- Số dữ liệu dùng cho huấn luyện: 642,218
- Số dữ liệu dùng cho kiểm tra: 160,555

BIỂU ĐỒ THÔNG KÊ DỮ LIỆU



Hình 9. Biểu đồ thống kê dữ liệu

Thu thập dữ liệu huấn luyện Rasa

Thu thập dữ liệu huấn luyện Rasa NLU: thu thập và tự tạo các mẫu câu hỏi thường gặp từ sinh viên liên quan đến quy chế đào tạo đại học. Với khoảng hơn 1500 mẫu

câu hỏi phong phú, phản ánh nhiều chủ đề (intents) chính: học phí, học bổng, chương trình đào tạo, thông tin tuyển sinh đại học, liên thông, sau đại học, quỹ hỗ trợ và những vấn đề liên quan khác. Những câu hỏi này không chỉ đại diện cho mối quan tâm thực tế của sinh viên mà còn thể hiện những thách thức mà họ đang gặp phải trong quá trình học tập.

Tạo câu trả lời: Trong bài luận văn này, chúng tôi định nghĩa các câu trả lời cố định là các phát biểu (utter) dựa vào văn bản quy chế đào tạo đại học của Trường Đại học Y Dược Cần Thơ, dựa vào chủ đề và ngữ cảnh của thông điệp đầu vào, hệ thống sẽ đưa ra câu trả lời thích hợp.

4.2 Môi trường thực nghiệm

Luận văn được thực hiện trên hệ điều hành Window 11 và được viết bằng ngôn ngữ lập trình Python phiên bản 3.10.10 với các thư viện cần thiết Keras (3.6.0), TensorFlow (3.6.0), Numpy (1.26.4), nltk (3.9.1),...

Phần cứng	Thông số kỹ thuật
CPU	Intel core i5
RAM	8GB
GPU	Intel(R) Iris(R)
Hard disk	256GB SSD

Bảng 1: Thông số kỹ thuật phần cứng thực hiện nghiên cứu

4.3 Kết quả thực nghiệm

Mục tiêu chính của thực nghiệm này là đánh giá hiệu quả của mô hình Seq2Seq trong việc sửa lỗi chính tả tiếng Việt. Để đạt được điều này, một tập dữ liệu tiếng Việt với các lỗi chính tả phổ biến đã được xây dựng. Các lỗi chính tả này bao gồm:

- Lỗi về dấu (ví dụ: thiếu hoặc nhầm dấu).
- Lỗi âm vị (các âm có cách phát âm gần nhau hoặc dễ bị nhầm lẫn).
- Lỗi nhập sai ký tự trên bàn phím do vị trí phím gần nhau.

Tập dữ liệu được xây dựng gồm nhiều câu tiếng Việt có lỗi chính tả, được chọn lọc để phản ánh các lỗi phổ biến và đa dạng về ngữ cảnh. Tập dữ liệu này được chia thành

các tập huấn luyện và tập kiểm tra nhằm đánh giá khả năng của mô hình trong việc sửa lỗi.

- Tập huấn luyện: Sử dụng X câu với các biến thể khác nhau của lỗi chính tả, giúp mô hình học được cách nhận diện và sửa lỗi.
- Tập kiểm tra: Sử dụng hơn một trăm câu hỏi ngẫu nhiên để đánh giá hiệu quả của mô hình sau khi huấn luyện.

Kết quả thực nghiệm cho thấy mô hình Seq2Seq đạt độ chính xác khoảng 90% trong việc sửa lỗi chính tả tiếng Việt cơ bản, chứng tỏ tiềm năng của mô hình trong ứng dụng thực tế.

Tiến hành thử nghiệm chat với chatbot ngẫu nhiên mỗi lần 30 câu trên tổng số 150 câu hỏi trong phạm vi kịch bản đã huấn luyện cho chatbot thì ta được kết quả theo bảng đánh giá sau.

Lần thử nghiệm	Số câu đúng	Độ chính xác
1	22/30	73%
2	23/30	76%
3	23/30	76%
4	25/30	83%
5	26/30	86%

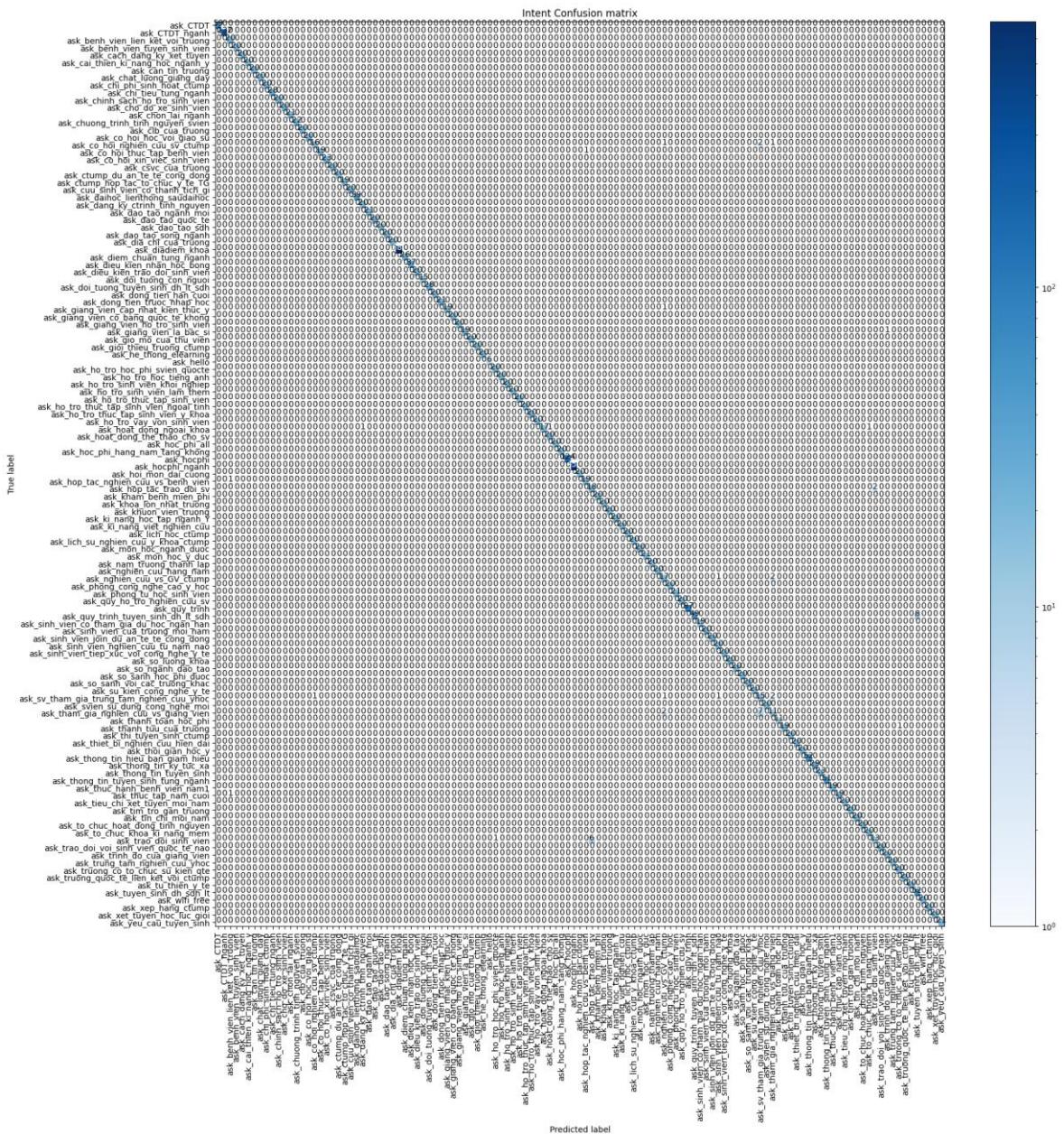
Bảng 2: Bảng mô tả số lần thử nghiệm với người dùng

Bảng mô tả số lần thử nghiệm

Thử nghiệm tương tác với chatbot cho thấy độ chính xác trung bình trên 70% cà chỉ tính các câu hỏi xung quanh kịch bản đã huấn luyện cho mô hình chatbot. Đối với những câu hỏi chưa được training (out of scop) thì bot sẽ không phản hồi ra nội dung để không làm nhiễu dữ liệu khi kết hợp với công nghệ RAG.

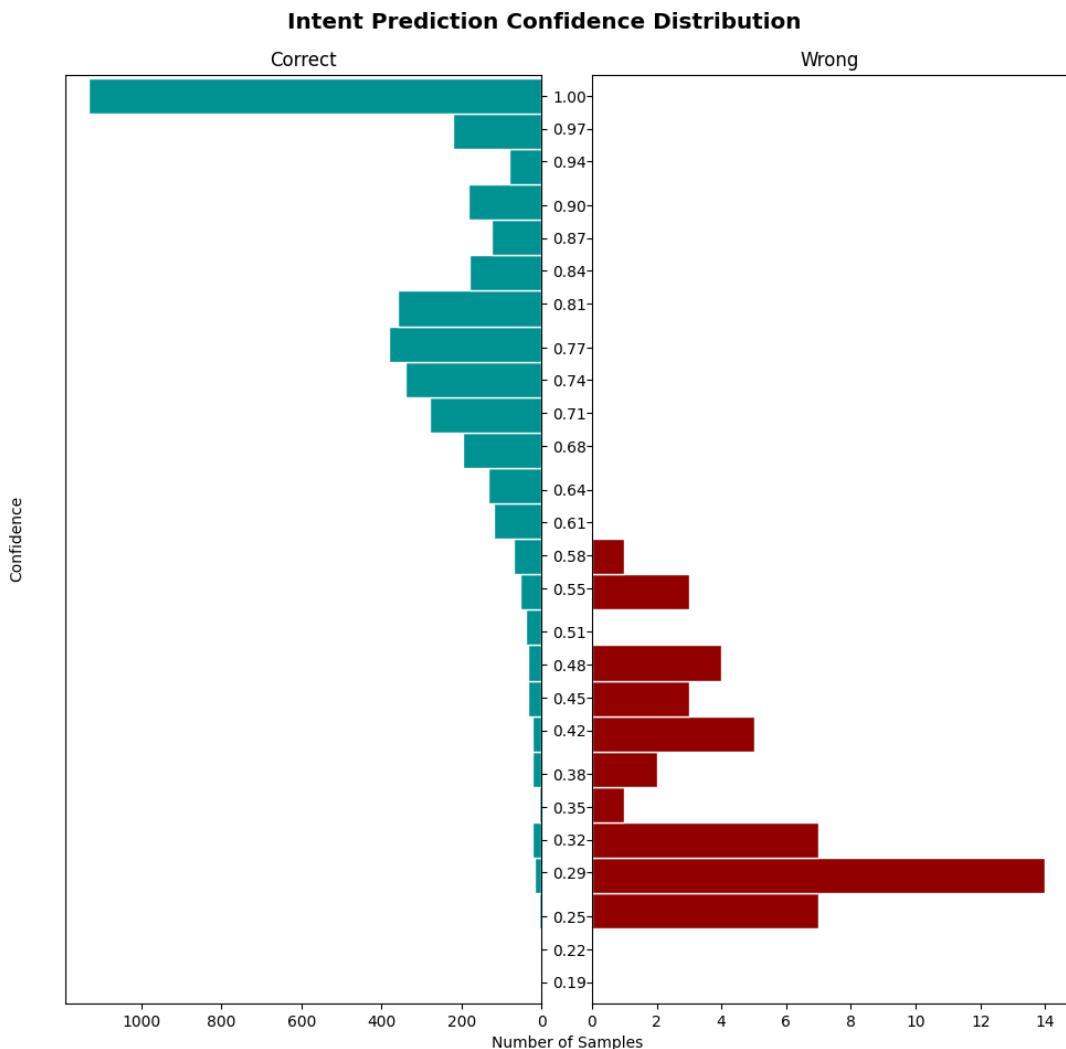
Kiểm thử NLU: RASA cung cấp các công cụ kiểm thử cho phép người đưa vào các ý định và câu chuyện để kiểm thử NLU. Trong giai đoạn kiểm thử này, ma trận nhầm lẫn (confusion matrix) và Biểu đồ độ tự tin của các dự đoán ý định (Intent Prediction Confidence Distribution) được dùng để đánh giá thành phần NLU: Ma trận nhầm lẫn (confusion matrix) với trực ngang là ý định do hệ thống dự đoán (predicted labels) và cột dọc là ý định thực tế (true labels), mỗi ô trong ma trận thể hiện số lần một intent thực tế được dự đoán là một intent cụ thể, mỗi ô thuộc đường chéo thể hiện intent dự đoán và intent thực tế khớp với nhau, còn mỗi ô ngoài đường chéo thể hiện hệ thống dự đoán nhầm lẫn, tương tự cũng có ma trận nhầm lẫn thực thể.

Biểu đồ độ tự tin của các dự đoán ý định (Intent Prediction Confidence Distribution) Hiển thị mức độ tự tin của các dự đoán và số lượng mẫu, với bên Correct là dự đoán đúng, cột Wrong là dự đoán sai



Hình 10. Ma trận nhầm lẫn ý định

Biểu đồ ma trận nhầm lẫn cho thấy mô hình dự đoán chính xác phần lớn các intent, với giá trị lớn tập trung trên đường chéo chính. Tuy nhiên, vẫn xảy ra nhầm lẫn giữa một số intent, đặc biệt là các intent có nội dung tương đồng như: trao đổi sinh viên, các sự kiện nghiên cứu, quy trình học tập. Để cải thiện, cần phân tích các điểm nhầm lẫn lớn ngoài đường chéo và bổ sung dữ liệu huấn luyện giúp phân biệt rõ ràng hơn giữa các intent.



Hình 11. Biểu đồ tự tin của ý định

Biểu đồ hiển thị phân phối độ tự tin của mô hình Rasa trong việc dự đoán ý định (intent). Các dự đoán đúng (Correct) chiếm phần lớn, với mức độ tự tin từ 0,97 đến 1,00, hơn 1000 mẫu tập trung ở mức cao nhất. Khi độ tự tin giảm, số lượng dự đoán đúng cũng giảm.

Ngược lại, các dự đoán sai (Wrong) chủ yếu xuất hiện ở mức độ tự tin thấp, đặc biệt trong khoảng 0,29 đến 0,35, với số lượng dự đoán sai thấp hơn rất nhiều so với đúng.

Kết quả cho thấy mô hình hoạt động hiệu quả với tỷ lệ dự đoán đúng cao, nhưng cần cải thiện các trường hợp dự đoán sai, đặc biệt ở mức độ tự tin trung bình (0,4-0,6), để nâng cao hiệu năng tổng thể.

Trong luận văn này hệ thống được kiểm tra với bộ 150 câu hỏi, kết quả phản hồi từ Chatbot được kết hợp từ mô hình sửa lỗi chính tả Seq2seq với RASA và phương pháp Retrieval-Augmented Generation được thể hiện như bảng sau:

Tổng số câu	Số câu đúng	Độ chính xác
150	119	79.33%

4.4 Đánh giá độ chính xác của hệ thống

Thử nghiệm tương tác với Chatbot 150 câu hỏi với nhiều kịch bản gồm các câu hỏi liên quan trong tài liệu, các câu hỏi bên ngoài, các câu hỏi sai chính tả. Đối với các câu hỏi Chatbot trả lời bằng kiến thức được huấn luyện trước đó hoặc từ bên ngoài sẽ được đánh giá là sai. Đối với câu trả lời đúng được đánh giá bằng cách kiểm tra với tài liệu truy xuất và có phù hợp ngữ cảnh khi truy xuất tài liệu hay không. Tỷ lệ đúng của chatbot trong thử nghiệm này là 79.33%.

Chương 5: Kết luận và hướng phát triển

5.1 Kết luận

Trong luận văn này, chúng tôi đã sử dụng nền tảng Rasa để xây dựng và triển khai hệ thống chatbot thông minh, phục vụ việc giao tiếp và cung cấp thông tin tự động cho người dùng. Rasa được chọn vì khả năng xử lý ngôn ngữ tự nhiên (NLP) và điều hướng hội thoại thông minh, đáp ứng nhu cầu giao tiếp đa dạng trong các tình huống thực tế.

Ngoài ra, mô hình Seq2Seq đã được tích hợp để cải thiện hiệu quả của chatbot bằng cách phát hiện và sửa lỗi chính tả trong các câu văn. Seq2Seq được lựa chọn nhờ khả năng học các chuỗi đầu vào và đầu ra phức tạp, giúp tăng độ chính xác trong việc xử lý ngôn ngữ tự nhiên. Qua quá trình huấn luyện và thử nghiệm, hệ thống đã đạt kết quả khả quan, đảm bảo độ chính xác cao trong nhận diện và sửa lỗi chính tả, góp phần nâng cao trải nghiệm người dùng.

5.2 Hướng phát triển

Trong tương lai, mô hình có thể được phát triển và cải tiến thêm theo các hướng sau:

1. Cải thiện mô hình Seq2Seq: Hiện tại, mô hình Seq2Seq đang đạt được độ chính xác tốt nhưng vẫn có thể cải tiến về hiệu quả tính toán và dung lượng bộ nhớ. Các kỹ thuật tối ưu hóa như giảm số tham số, tinh chỉnh siêu tham số hoặc sử dụng các kiến trúc tiên tiến hơn như Transformers có thể được áp dụng để nâng cao hiệu suất của mô hình.
2. Nâng cao chất lượng dữ liệu huấn luyện: Bổ sung thêm dữ liệu đa dạng sẽ giúp mô hình cải thiện khả năng sửa lỗi chính tả. Việc tạo ra một tập dữ liệu lớn hơn và phong phú hơn về ngữ cảnh sẽ giúp mô hình xử lý tốt hơn trong các trường hợp phức tạp và tăng cường độ chính xác trong các tình huống thực tế.
3. Tích hợp khả năng học hỏi theo ngữ cảnh người dùng: Triển khai các phương pháp học tập trực tuyến (online learning) giúp mô hình có thể học hỏi từ các phản hồi người dùng thực tế. Điều này sẽ giúp mô hình tự điều chỉnh để phù hợp với các cách viết và lỗi chính tả phổ biến của người dùng cụ thể, từ đó nâng cao tính cá nhân hóa.
4. Áp dụng các mô hình ngôn ngữ lớn (LLMs): Việc tích hợp các mô hình ngôn ngữ lớn hiện đại như BERT, GPT hoặc các phiên bản tối ưu hơn của Rasa sẽ giúp chatbot xử lý tốt hơn các ngữ cảnh phức tạp, tăng cường khả năng sửa lỗi chính tả trong các văn bản có nội dung đa dạng.
5. Cải tiến quy trình xử lý và hiệu suất hệ thống: Một hướng phát triển khác là tối ưu hóa hệ thống tổng thể, từ việc giảm độ trễ trong thời gian phản hồi cho đến việc xử lý dữ liệu một cách hiệu quả hơn. Điều này sẽ giúp hệ thống hoạt

động ổn định và phản hồi nhanh chóng ngay cả khi có lượng lớn người dùng cùng lúc.

Việc tiếp tục phát triển và tối ưu hóa mô hình sửa lỗi chính tả kết hợp với nền tảng Rasa không chỉ giúp nâng cao độ chính xác mà còn mang lại trải nghiệm sử dụng tốt hơn, đáp ứng nhu cầu đa dạng của người dùng trong giao tiếp và tìm kiếm thông tin.

Phụ lục

Sau khi huấn luyện mô hình sửa lỗi chính tả tiếng Việt phổ biến. Chúng tôi thực hiện viết chương trình và sau đó kiểm thử

*Chú ý: Các câu hỏi được tạo ra dựa trên các lỗi sai chính tả đánh máy phổ biến ở người dùng, nếu lỗi chính tả trong một câu quá nhiều hoặc có tình đánh chữ ngẫu nhiên sẽ ảnh hưởng đến độ chính xác của hệ thống

Câu 1: Câu hỏi ban đầu: Điểm chuẩn đầu vafo các ngafnh của truwongf là bao nhiêu?
Câu đã sửa lỗi: Điểm chuẩn đầu vào các ngành của trường là bao
Câu 2: Câu hỏi ban đầu: Khu vực xung quanh truwongwff có nhieefu chỗ ở trợ không?
Câu đã sửa lỗi: Khu vực xung quanh trường có nhiều chỗ ở trợ không?
Câu 3: Câu hỏi ban đầu: Em cos thể tham gia các caau lạc bộ sinh viên nào ở truwongfw ?
Câu đã sửa lỗi: Em có thể tham gia các câu lạc bộ sinh viên nào ở trường
Câu 4: Câu hỏi ban đầu: trường cos các hoajt động ngoại khoá khoong?
Câu đã sửa lỗi: trường có các hoạt động ngoại khoá không?
Câu 5: Câu hỏi ban đầu: Sinh viên cos thể đi lafm thêm khi học khoong?
Câu đã sửa lỗi: Sinh viên có thể đi làm thêm khi học không?
Câu 6: Câu hỏi ban đầu: Truwongfw có tổ chức các chuowng trình trao đổi sinh vieen khoong?
Câu đã sửa lỗi: Trường có tổ chức các chương trình trao đổi sinh viên không?
Câu 7: Câu hỏi ban đầu: Cow hội thực tâjp ở các bệnh viện lớn thees nafo?
Câu đã sửa lỗi: Cơ hội thực tập ở các bệnh viện lớn thế nào?
Câu 8: Câu hỏi ban đầu: Có quán ăn hay căng tin troong khuoon vien trường không?
Câu đã sửa lỗi: Có quán ăn hay căng tin trong khuôn viên trường không?
Câu 9: Câu hỏi ban đầu: Casc hoajt động thể thao cho sinh viên nhuw thế nafo?
Câu đã sửa lỗi: Các hoạt động thể thao cho sinh viên như thế nào?
Câu 10: Câu hỏi ban đầu: Trường co trung taam nghiên cuu y hoc không?
Câu đã sửa lỗi: Trường có trung tâm nghiên cứu y học không?
Câu 11: Câu hỏi ban đầu: Sinh viên có dduowjcw tham gia vào các dự án nghiên cứu không?
Câu đã sửa lỗi: Sinh viên có được tham gia vào các dự án nghiên cứu không?
Câu 12: Câu hỏi ban đầu: Trường có chương trình hoox trowj khởi nghiệp cho sinh vieen không?
Câu đã sửa lỗi: Trường có chương trình hỗ trợ khởi nghiệp cho sinh viên không?
Câu 13: Câu hỏi ban đầu: Lafm sao để tham gia các dự asn nghiên cứu của giảng viên?
Câu đã sửa lỗi: Làm sao để tham gia các dự án nghiên cứu của giảng viên?
Câu 14: Câu hỏi ban đầu: Có thể xin quyx hỗ trowj nghieen cứu từ trường khoong?
Câu đã sửa lỗi: Có thể xin quỹ hỗ trợ nghiên cứu từ trường không?
Câu 15: Câu hỏi ban đầu: Trường cos hợp tasc với beejnh vien nào để nghieen cứu khoong?
Câu đã sửa lỗi: Trường có hợp tác với bệnh viện nào để nghiên cứu không?
Câu 16: Câu hỏi ban đầu: Các công trifnh nghiên cứu của sinh vieen có đươjc công boos không?
Câu đã sửa lỗi: Các công trình nghiên cứu của sinh viên có được công bố không?
Câu 17: Câu hỏi ban đầu: Sinh viên có theer thực hiện nghiên cuws tuwf năm nao?
Câu đã sửa lỗi: Sinh viên có thể thực hiện nghiên cứu từ năm nào?
Câu 18: Câu hỏi ban đầu: Các thiết bij phong this nghiệm của trường có hieejn đại khoong?
Câu đã sửa lỗi: Các thiết bị phòng thí nghiệm của trường có hiện đại không?
Câu 19: Câu hỏi ban đầu: Trường có tổ chuwsc hội thao nghieen cuu hàng năm không?
Câu đã sửa lỗi: Trường có tổ chức hội thao nghiên cứu hàng năm không?
Câu 20: Câu hỏi ban đầu: Thuw vien của trường có mờ cà cuosi tuân khoong?
Câu đã sửa lỗi: Thư viện của trường có mờ cà cuối tuần không?

- Hệ thống đã nhận dạng và sửa chính xác 18/20 câu chứa các lỗi chính tả phổ biến, tuy nhiên vẫn còn hạn chế ở câu 1 và câu 6, nơi một số lỗi nhỏ khiến hệ thống chưa thể nhận diện hoặc sửa đúng hoàn toàn.

Câu hỏi sửa lỗi: Trường có tổ chức ngày trao giải thưởng của trường không?

Câu 20: Câu hỏi ban đầu: Thư viện của trường có mở cả cuối tuần không?
 Câu đã sửa lỗi: Thư viện của trường có mở cả cuối tuần không?

Câu 21: Câu hỏi ban đầu: Có phòng tuwj học cho sinh viên không?
 Câu đã sửa lỗi: Có phòng tự học cho sinh viên không?

Câu 22: Câu hỏi ban đầu: Trường có heej thống e-learning không?
 Câu đã sửa lỗi: Trường có hệ thống elerrning không?

Câu 23: Câu hỏi ban đầu: Các phòng thí nghiệm có đủ thieest bị cho sinh viên khoong?
 Câu đã sửa lỗi: Các phòng thí nghiệm có đủ thiết bị cho sinh viên không?

Câu 24: Câu hỏi ban đầu: trường có wifi miễn phis không?
 Câu đã sửa lỗi: trường có wifi miễn phí không?

Câu 25: Câu hỏi ban đầu: Khuon viên trường có rộng khoong vaajy?
 Câu đã sửa lỗi: Khuôn viên trường có rộng không vậy?

Câu 26: Câu hỏi ban đầu: Có chôx để xe miễn phí cho sinh viên không?
 Câu đã sửa lỗi: Có chỗ để xe miễn phí cho sinh viên không?

Câu 27: Câu hỏi ban đầu: Trường có các phofng họp nhosm không?
 Câu đã sửa lỗi: Trường có các phòng họp nhóm không?

Câu 28: Câu hỏi ban đầu: Cawng tin trong trường có basn thức ăn gias sinh viên không?
 Câu đã sửa lỗi: Cảng tin trong trường có bán thức ăn giá sinh viên không?

Câu 29: Câu hỏi ban đầu: Trường có hệ thoonsgs kí túc xá hiện đại không?
 Câu đã sửa lỗi: Trường có hệ thống kí túc xá hiện đại không?

Câu 30: Câu hỏi ban đầu: Học phis ngafnh Y đa khoa là bao nhieeu?
 Câu đã sửa lỗi: Học phí ngành Y đa khoa là bao nhiêu?

Câu 31: Câu hỏi ban đầu: Trường có chính sasch hỗ trợ hojc phí cho sinh viên khó khawn không?
 Câu đã sửa lỗi: Trường có chính sách hỗ trợ học phí cho sinh viên khó khăn không?

Câu 32: Câu hỏi ban đầu: Học phis có tăng theo từng nawm không?
 Câu đã sửa lỗi: Học phí có tăng theo từng năm không?

Câu 33: Câu hỏi ban đầu: Sinh viên có được giarm học phí nếu có hoàn carnh khó khawn không?
 Câu đã sửa lỗi: Sinh viên có được giảm học phí nếu có hoàn cảnh khó khăn không?

Câu 34: Câu hỏi ban đầu: Trường có học boorng khuyến khích học tập không?
 Câu đã sửa lỗi: Trường có học bổng khuyến khích học tập không?

Câu 35: Câu hỏi ban đầu: hojc bổng cura trường có dành cho sinh viên quoosc tees không?
 Câu đã sửa lỗi: học bổng của trường có dành cho sinh viên quốc tế không?

Câu 36: Câu hỏi ban đầu: Học phí nganh Được so với casc trường khác có cao không không?
 Câu đã sửa lỗi: Học phí nganh Được so với các trường khác có cao không vậy?

Câu 37: Câu hỏi ban đầu: Trường có yêu caafu dosng hojc phi trước khi nhập hojc khoong?
 Câu đã sửa lỗi: Trường có yêu cầu đóng học phí trước khi nhập học không?

Câu 38: Câu hỏi ban đầu: Thowfi gian đóng hojc phi là khi nào vâjy?
 Câu đã sửa lỗi: Thời gian đóng học phí là khi nào vậy?

Câu 39: Câu hỏi ban đầu: Truwongf cos hỗ trowj sinh viên vay vốn hojc tập khoong?
 Câu đã sửa lỗi: Trường có hỗ trợ sinh viên vay vốn học tập không?

Câu 40: Câu hỏi ban đầu: trường có tổ chức thực tập cho sinh viên khong?
 Câu đã sửa lỗi: trường có tổ chức thực tập cho sinh viên không?

- Hệ thống đã nhận dạng và sửa chính xác 15/20 câu chứa các lỗi chính tả phổ biến, tuy nhiên vẫn còn hạn chế ở câu 26, 36, 38, 39 và 40 nơi một số lỗi nhỏ khiến hệ thống chưa thể nhận diện hoặc sửa đúng hoàn toàn.

Câu 41: Câu hỏi ban đầu: Sinh viên thực tập owr ddaau khi hojc ngafnh Y khoa?
 Câu đã sửa lỗi: Sinh viên thực tập ở đâu khi học ngành Y khoa?

Câu 42: Câu hỏi ban đầu: Có theer thuwjc tập owr các bệnh veejn ngoafi tinh khong?
 Câu đã sửa lỗi: Có thể thực tập ở các bệnh viện ngoài tinh khong?

Câu 43: Câu hỏi ban đầu: trường có hợp tác với nhuwngx bệnh viện nào cho sinh viên thuwjc tập?
 Câu đã sửa lỗi: trường có hợp tác với nhung bệnh viện cho sinh viên thực tập

Câu 44: Câu hỏi ban đầu: Sinh vileen có theer thuwjc tap owr bệnh viện nafo trong nawm cuoosi?
 Câu đã sửa lỗi: Sinh viên có thể thực tap ở bệnh viện nào trong năm cuối?

Câu 45: Câu hỏi ban đầu: Cow hooji việc lafm sau khi tốt nghiệp ngafnh Y thế nào?
 Câu đã sửa lỗi: Cơ hội việc làm sau khi tốt nghiệp ngành Y thế nào?

Câu 46: Câu hỏi ban đầu: trường cos chuongw trifnh ho trowj viejc làm sau tốt nghiệp khong?
 Câu đã sửa lỗi: trường có chương trình hỗ trợ việc làm sau tốt nghiệp khong?

Câu 47: Câu hỏi ban đầu: Các bệnh viện có thuowngfw xuyeen tuyển dụng sinh vileen trường khong?
 Câu đã sửa lỗi: Các bệnh viện có thường xuyên tuyển dụng sinh viên trường khong?

Câu 48: Câu hỏi ban đầu: Sinh vileen co theer xin việc lafm theem trong ngành khoong?
 Câu đã sửa lỗi: Sinh viên co thể xin việc làm thêm trong ngành không?

Câu 49: Câu hỏi ban đầu: Lafm sao deer tijm được việc lafm sau khi ra truongf?
 Câu đã sửa lỗi: Làm sao để tìm được việc làm sau khi ra trường?

Câu 50: Câu hỏi ban đầu: trường cos chương trình hojc quoc tees khoong?
 Câu đã sửa lỗi: trường có chương trình học quốc tế không?

Câu 51: Câu hỏi ban đầu: Lafm sao để ddawng kys hojc trao đổi quoc te
 Câu đã sửa lỗi: Làm sao để đăng ký học trao đổi quốc tế

Câu 52: Câu hỏi ban đầu: Điều kiện để được tham gia chương trình trao ddooris sinh vileen là gì?
 Câu đã sửa lỗi: Điều kiện để được tham gia chương trình trao đổi sinh viên là gì?

Câu 53: Câu hỏi ban đầu: Các nước nafo trường có howjp tác trao ddooris sinh viên?
 Câu đã sửa lỗi: Các nước nào trường có hợp tác trao đổi sinh viên?

Câu 54: Câu hỏi ban đầu: trường có liien kết với trường ddaji hojc quốc tế nafo khoong?
 Câu đã sửa lỗi: trường có liên kết với trường đại học quốc tế nào không?

Câu 55: Câu hỏi ban đầu: Sinh viên có thể du hoc ngawnss hajin trong thời gian học tại trường khoong?
 Câu đã sửa lỗi: Sinh viên có thể du học ngắn hạn trong thời gian học tại trường không?

Câu 56: Câu hỏi ban đầu: Các chuongw trình trao đổi sinh vileen thường keso dài bao laau?
 Câu đã sửa lỗi: Các chương trình trao đổi sinh viên thường kéo dài bao lâu?

Câu 57: Câu hỏi ban đầu: Truwngvfw có hỗ trợ sinh viên hojc tiếng Anh không?
 Câu đã sửa lỗi: Trường có hỗ trợ sinh viên học tiếng Anh không?

Câu 58: Câu hỏi ban đầu: Sinh vileen trao đổi quoosc tees có được hỗ trowj hojc phí không?
 Câu đã sửa lỗi: Sinh viên trao đổi quốc tế có được hỗ trợ học phí không?

Câu 59: Câu hỏi ban đầu: Dooji ngux giảng viên của trường có kinh nghieejm như thế nào?
 Câu đã sửa lỗi: Đội ngũ giảng viên của trường có kinh nghiệm như thế nào?

Câu 60: Câu hỏi ban đầu: Giarng viên có banwgfw cấp quốc tees khoong?
 Câu đã sửa lỗi: Giảng viên có bằng cấp quốc tế không?

- Hệ thống đã nhận dạng và sửa chính xác 14/20 câu chứa các lỗi chính tả phổ biến, tuy nhiên vẫn còn hạn chế ở câu 43, 44, 46, 47, 48 và 49 nơi một số lỗi nhỏ khiến hệ thống chưa nhận diện hoặc sửa đúng hoàn toàn.

Câu 61: Câu hỏi ban đầu: Các giảng viên có thường xuyên cập nhật kiến thức y học mới không?
 Câu đã sửa lỗi: Các giảng viên có thường xuyên cập nhật kiến thức y học mới không?

Câu 62: Câu hỏi ban đầu: Có nhiều giảng viên là bác sĩ không?
 Câu đã sửa lỗi: Có nhiều giảng viên là bác sĩ không?

Câu 63: Câu hỏi ban đầu: trường có tổ chức các khóa học ngắn hạn không?
 Câu đã sửa lỗi: trường có tổ chức các khóa học ngắn hạn không?

Câu 64: Câu hỏi ban đầu: Có giảng viên ngoài tham gia giảng dạy học không?
 Câu đã sửa lỗi: Có giảng viên nước ngoài tham gia giảng dạy học không?

Câu 65: Câu hỏi ban đầu: Các giảng viên có hỗ trợ sinh viên ngoài giờ học không?
 Câu đã sửa lỗi: Các giảng viên có hỗ trợ sinh viên ngoài giờ học không?

Câu 66: Câu hỏi ban đầu: Sinh viên có theo học với các giáo sư nổi tiếng không?
 Câu đã sửa lỗi: Sinh viên có thể học với các giáo sư nổi tiếng không?

Câu 67: Câu hỏi ban đầu: Trường có chất lượng giảng dạy tốt không?
 Câu đã sửa lỗi: Trường có chất lượng giảng dạy tốt không?

Câu 68: Câu hỏi ban đầu: Có cơ hội học từ giảng viên quốc tế không?
 Câu đã sửa lỗi: Có cơ hội học từ giảng viên quốc tế không?

Câu 69: Câu hỏi ban đầu: Trường có tổ chức chương trình khám chữa bệnh miễn phí không?
 Câu đã sửa lỗi: Trường có tổ chức chương trình khám chữa bệnh miễn phí không?

Câu 70: Câu hỏi ban đầu: Sinh viên có được tham gia các chương trình từ thiện y tế không?
 Câu đã sửa lỗi: Sinh viên có được tham gia chương trình từ thiện y tế không?

Câu 71: Câu hỏi ban đầu: Trường có liên kết với các tổ chức y tế quốc tế không?
 Câu đã sửa lỗi: trường có liên kết với các tổ chức y tế quốc tế không?

Câu 72: Câu hỏi ban đầu: Các dự án y tế công cộng thường tập trung vào lĩnh vực gì?
 Câu đã sửa lỗi: Các dự án y tế công cộng thường tập trung vào lĩnh vực gì?

Câu 73: Câu hỏi ban đầu: Sinh viên có được tham gia vào các dự án y tế công cộng không?
 Câu đã sửa lỗi: Sinh viên có được tham gia vào các dự án y tế công cộng không?

Câu 74: Câu hỏi ban đầu: Có chương trình tình nguyện y tế không?
 Câu đã sửa lỗi: Có chương trình tình nguyện y tế không?

Câu 75: Câu hỏi ban đầu: Trường có hợp tác với các tổ chức phi chính phủ ko?
 Câu đã sửa lỗi: trường có hợp tác với các tổ chức phi chính phủ không?

Câu 76: Câu hỏi ban đầu: Có các chương trình hỗ trợ sức khỏe cộng đồng không?
 Câu đã sửa lỗi: Có các chương trình hỗ trợ sức khỏe cộng đồng không?

Câu 77: Câu hỏi ban đầu: Các dự án y tế cộng đồng của trường có tầm ảnh hưởng lớn ko?
 Câu đã sửa lỗi: Các dự án y tế cộng đồng của trường có tầm ảnh hưởng lớn không?

Câu 78: Câu hỏi ban đầu: Trường đã đạt được những thành tựu nào nổi bật?
 Câu đã sửa lỗi: Trường đã đạt được những thành tựu nào nổi bật?

Câu 79: Câu hỏi ban đầu: Đại học Y Dược Cần Thơ có nằm trong bảng xếp hạng nào không?
 Câu đã sửa lỗi: Đại học Y Dược Cần Thơ có nằm trong bảng xếp hạng nào không?

Câu 80: Câu hỏi ban đầu: trường có truyền thống nghiên cứu y khoa không?
 Câu đã sửa lỗi: trường có truyền thống nghiên cứu y khoa không?

- Hệ thống đã nhận dạng và sửa chính xác 13/20 câu chứa các lỗi chính tả phổ biến, tuy nhiên vẫn còn hạn chế ở câu 61, 64, 66, 67, 71 và 75 nơi một số lỗi nhỏ khiến hệ thống chưa nhận diện hoặc sửa đúng hoàn toàn.

Câu 81: Câu hỏi ban đầu: Các cựu sinh viên của trường có thành tựu gì?
Câu đã sửa lỗi: Các cựu sinh viên của trường có thành tựu gì?

Câu 82: Câu hỏi ban đầu: Uy tín của trường so với các trường khác thế nào?
Câu đã sửa lỗi: Uy tín của trường so với các trường khác thế nào?

Câu 83: Câu hỏi ban đầu: Trường có từng có chức các sự kiện quốc tế không?
Câu đã sửa lỗi: Trường có từng tổ chức các sự kiện quốc tế không?

Câu 84: Câu hỏi ban đầu: Các giáo sư nổi tiếng nào đang giảng dạy tại trường?
Câu đã sửa lỗi: Các giáo sư nổi tiếng nào đang giảng dạy tại trường?

Câu 85: Câu hỏi ban đầu: Trường có mối quan hệ hợp tác với các cơ quan nước ngoài?
Câu đã sửa lỗi: Trường có mối quan hệ hợp tác với các cơ quan nước ngoài?

Câu 86: Câu hỏi ban đầu: Độ uy tín của trường trong khu vực như thế nào?
Câu đã sửa lỗi: Độ uy tín của trường trong khu vực như thế nào?

Câu 87: Câu hỏi ban đầu: Trường có tuyển sinh liên thông không?
Câu đã sửa lỗi: Trường có tuyển sinh liên thông không?

Câu 88: Câu hỏi ban đầu: Bao nhiêu ngành đang được đào tạo tại trường?
Câu đã sửa lỗi: Bao nhiêu ngành đang được đào tạo tại trường?

Câu 89: Câu hỏi ban đầu: Trường có học bổng cho sinh viên khốn khó không?
Câu đã sửa lỗi: Trường có học bổng cho sinh viên khó khăn không?

Câu 90: Câu hỏi ban đầu: Trường có phong masy tính khoong
Câu đã sửa lỗi: Trường có phong masy tính không

Câu 91: Câu hỏi ban đầu: Học phí tăng theo từng năm không?
Câu đã sửa lỗi: Học phí có tăng theo từng năm không?

Câu 92: Câu hỏi ban đầu: Sinh viên có được học tiếng Anh ko?
Câu đã sửa lỗi: Sinh viên có được học tiếng Anh không?

Câu 93: Câu hỏi ban đầu: Trường có cho phép sinh viên xin học lại các môn không?
Câu đã sửa lỗi: Trường có cho phép sinh viên xin học lại các môn không?

Câu 94: Câu hỏi ban đầu: Điểm chuẩn của ngành y
Câu đã sửa lỗi: Điểm chuẩn của ngành y

Câu 95: Câu hỏi ban đầu: Trường có những hoạt động ngoại khóa nào cho sinh viên vui chơi?
Câu đã sửa lỗi: Trường có những hoạt động ngoại khóa nào cho sinh viên vui chơi?

Câu 96: Câu hỏi ban đầu: Trường có tổ chức các buổi thảo luận về đức ko?
Câu đã sửa lỗi: Trường có tổ chức các buổi thảo luận về đức không?

Câu 97: Câu hỏi ban đầu: Sinh viên có được tham gia các hoạt động ngoại khóa vui chơi ko?
Câu đã sửa lỗi: Sinh viên có được tham gia các hoạt động ngoại khóa vui chơi không?

Câu 98: Câu hỏi ban đầu: Trường có trung tâm tư vấn tư vấn lý cho sinh viên ko?
Câu đã sửa lỗi: Trường có trung tâm tư vấn tư vấn lý cho sinh viên không?

Câu 99: Câu hỏi ban đầu: Sinh viên có được hỗ trợ về sức khỏe tinh thần ko?
Câu đã sửa lỗi: Sinh viên có được hỗ trợ về sức khỏe tinh thần không?

Câu 100: Câu hỏi ban đầu: Có chương trình hỗ trợ sinh viên gặp khó khăn về tài chính ko?
Câu đã sửa lỗi: Có chương trình hỗ trợ sinh viên gặp khó khăn về tài chính không?

- Hệ thống đã nhận dạng và sửa chính xác 15/20 câu chứa các lỗi chính tả phổ biến, tuy nhiên vẫn còn hạn chế ở câu 82, 83, 84, 90 và 93 nơi một số lỗi nhỏ khiến hệ thống chưa thể nhận diện hoặc sửa đúng hoàn toàn.

Nhận xét:

Trong quá trình xây dựng và triển khai hệ thống nhận dạng và sửa lỗi chính tả, một số hạn chế đã được nhận thấy liên quan đến khả năng sửa lỗi trong các câu chứa nhiều từ sai hoặc các lỗi nằm ngoài phạm vi dữ liệu huấn luyện. Khi câu văn có quá nhiều từ sai chính tả, hệ thống có thể gặp khó khăn trong việc xác định chính xác từng lỗi và đưa ra các gợi ý sửa phù hợp, dẫn đến nguy cơ sửa sai hoặc không sửa được toàn bộ lỗi.

Ngoài ra, hệ thống dựa vào dữ liệu huấn luyện để học cách nhận diện và sửa các lỗi chính tả phổ biến. Với độ chính xác đạt khoảng 74% trên tập kiểm tra, hệ thống cho thấy khả năng sửa lỗi hiệu quả với các lỗi phổ biến đã có trong dữ liệu mẫu. Tuy nhiên, nếu câu chứa các từ sai chính tả không xuất hiện trong dữ liệu huấn luyện hoặc thuộc về các ngữ cảnh quá khác biệt, hệ thống sẽ khó có thể phát hiện và sửa lỗi chính xác. Điều này cho thấy nhu cầu cần mở rộng và đa dạng hóa tập dữ liệu

huấn luyện để đảm bảo hệ thống có thể xử lý tốt hơn trong các tình huống đa dạng và phức tạp hơn.

Sau khi hoàn tất quá trình huấn luyện mô hình chatbot trên nền tảng Rasa, chúng tôi tiến hành xây dựng chương trình và thực hiện kiểm thử. Kết quả chi tiết được thể hiện trong bảng dưới đây.

STT	Người dùng	ChatBot	Đánh giá
1	Xin chào	Xin chào, bạn cần hỗ trợ gì?	đúng
2	Trường Đại học Y Dược Cần Thơ thành lập năm nào?	Trường Đại học Y Dược Cần Thơ được thành lập năm 2002.	đúng
3	Khoa nào ở trường là lớn nhất?	Khoa Y là khoa lớn nhất của trường Đại học Y Dược Cần Thơ.	đúng
4	Trường có bao nhiêu khoa đào tạo?	Trường Đại học Y Dược Cần Thơ có tổng cộng 07 khoa đào tạo, bao gồm: khoa Y, khoa Khoa học cơ bản, khoa Điều dưỡng, khoa Dược, khoa Răng hàm mặt (RHM), và khoa Y tế công cộng (YTCC), khoa Kỹ thuật y học	đúng
5	Đại học Y Dược Cần Thơ có ký túc xá không?	Tôi có thông tin về ký túc xá xem chi tiết tại đây: https://www.facebook.com/KytucxaTDHYDCT/	đúng
6	Trường có bao nhiêu sinh viên mỗi năm?	Hiện tại, trường Đại học Y Dược Cần Thơ có quy mô đào tạo gần 3.000 sinh viên.	đúng
7	điểm chuẩn ngành y khoa là bao nhiêu	Để vào học ngành Y đa khoa tại Trường Đại học Y Dược Cần Thơ, bạn cần đạt được điểm chuẩn là 25,52.	đúng
8	Em có thể tham gia các câu lạc bộ sinh viên nào ở trường?	Trường Đại học Y Dược Cần Thơ có nhiều câu lạc bộ dành cho sinh viên tham gia, bao gồm: - Đoàn Thanh Niên - Hội Sinh Viên - Các Câu Lạc Bộ, Đội Nhóm Trực Thuộc Cấp Khoa, Trường - Các hoạt động văn hóa, văn nghệ, thể thao lành mạnh - Thanh niên tình nguyện - Các hoạt động xã hội khác từ cấp Khoa, Trường hoặc tương đương Bạn có thể tham khảo thêm thông tin về các câu lạc bộ này và đăng ký tham gia phù hợp với sở thích và năng lực của	đúng

		mình.	
9	Các tiêu chí xét tuyển của trường có thay đổi hằng năm không?	Các điều kiện xét tuyển của trường Đại học Y Dược Cần Thơ có thể thay đổi theo từng năm, tùy thuộc vào các chính sách và quy định của Bộ Giáo dục và Đào tạo cũng như của trường. Do đó, để có thông tin chính xác và cập nhật nhất về các yêu cầu xét tuyển, bạn có thể tham khảo trang web chính thức của trường tại địa chỉ: http://www.ctump.edu.vn/default.aspx?tabid=1037 .	đúng
10	Có phòng tự học cho sinh viên không?	Có, Trường Đại học Y Dược Cần Thơ có phòng tự học cho sinh viên.	đúng
11	điểm chuẩn các ngành đào tạo	Thông tin điểm chuẩn các ngành tuyển sinh:,Tên ngành: Y khoa,Mức điểm chuẩn: 25,52,Tên ngành: Răng hàm mặt,Mức điểm chuẩn: 25,4,Tên ngành: Y học dự phòng,Mức điểm chuẩn: 22,35,Tên ngành: Y học cổ truyền,Mức điểm chuẩn: 23,9,Tên ngành: Dược,Mức điểm chuẩn: 24,7,Tên ngành: Điều dưỡng,Mức điểm chuẩn: 23,2,Tên ngành: Y tế công cộng,Mức điểm chuẩn: 20,Tên ngành: Kỹ thuật xét nghiệm y học,Mức điểm chuẩn: 24,05,Tên ngành: Hộ sinh,Mức điểm chuẩn: 22,25,Tên ngành: Kỹ thuật hình ảnh y học,Mức điểm chuẩn: 23,5,Tên ngành: kỹ thuật y sinh,Mức điểm chuẩn: 23,5	đúng
12	Các phòng thí nghiệm có đủ thiết bị cho sinh viên không	Có, các phòng thí nghiệm tại Trường Đại học Y Dược Cần Thơ được trang bị đầy đủ thiết bị cần thiết để sinh viên có thể thực hành và thực tập.	đúng
13	Trường có wifi miễn phí không?	Trường có cung cấp wifi miễn phí cho sinh viên ở một số nơi nhất định để phục vụ việc học tập	đúng
14	Có chỗ để xe miễn phí cho sinh viên không vậy?	Sinh viên phải trả phí gửi xe cho trường với giá là 2000vnđ/lượt	đúng
15	Trường có hỗ trợ sinh viên vay vốn học tập không?	Có, Trường Đại học Y Dược Cần Thơ có hỗ trợ sinh viên vay vốn học tập. Đây là một trong những chính sách nhằm tạo điều kiện cho sinh viên có hoàn cảnh khó khăn về kinh tế được tiếp tục học tập và hoàn thành chương trình đào tạo.	đúng
16	Cơ hội việc làm sau khi tốt nghiệp ngành Y thế nào?	Cơ hội việc làm sau khi tốt nghiệp ngành Y khá rộng mở. Sinh viên tốt nghiệp ngành Y có thể làm việc tại các bệnh viện, trung tâm y tế, phòng khám, hoặc các cơ sở y tế khác	đúng

17	Trường có chương trình học quốc tế không?	Trường Đại học Y Dược Cần Thơ có liên kết với Đại học Y Dược Thành phố Hồ Chí Minh để đào tạo Chuyên khoa cấp I Y học Gia đình. Ngoài ra, trường cũng mở rộng hợp tác đào tạo sau đại học và liên kết với các chương trình, dự án hợp tác quốc tế trong nghiên cứu khoa học. Do đó, có thể nói rằng trường có hợp tác với các tổ chức quốc tế để đào tạo.	đúng
18	Có nhiều giảng viên là bác sĩ không?	Có, đội ngũ giảng viên của Trường Đại học Y Dược Cần Thơ bao gồm nhiều bác sĩ. Cụ thể, trong bộ môn nội có 33 cán bộ, trong đó có 7 bác sĩ chuyên khoa cấp II.	đúng
19	Có các chương trình hỗ trợ sức khỏe cộng đồng không?	Các dự án y tế cộng đồng của Trường Đại học Y Dược Cần Thơ thường tập trung vào các lĩnh vực như chăm sóc sức khỏe ban đầu, phòng chống dịch bệnh, giáo dục sức khỏe, nâng cao nhận thức về sức khỏe cộng đồng, và cải thiện điều kiện vệ sinh môi trường.	đúng
20	Thông tin tuyển sinh đại học	Thông tin tuyển sinh đại học chính quy gồm: - Y khoa + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Răng hàm mặt + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Y học dự phòng + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Y học cổ truyền + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Điều dưỡng + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Y tế công cộng + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Kỹ thuật xét nghiệm y học + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Hộ sinh + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh - Kỹ thuật hình ảnh y học + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT,	đúng

		+ Tổ hợp: Toán hóa sinh - kỹ thuật y sinh + Mã xét tuyển: 100, + Phương thức tuyển: kết quả tốt nghiệp THPT, + Tổ hợp: Toán hóa sinh, toán lý hóa	
21	học phí ngành điều dưỡng	Học phí của ngành Điều dưỡng : Hệ chính quy : - Đào tạo trúng tuyển chính quy : 1.047.400/tín chỉ - Đào tạo theo nhu cầu xã hội (theo đặt hàng) : 1.077.000/tín chỉ - Đào tạo hệ quốc tế tiếng anh : 4.000.000/tín chỉ Hệ liên thông : - Đào tạo liên thông hệ chính quy : 1.047.400/tín chỉ - Đào tạo liên thông theo nhu cầu xã hội : 1.047.400/tín chỉ	đúng
22	Địa chỉ chính của trường nằm ở đâu?	Trụ sở chính của Trường Đại học Y Dược Cần Thơ hiện nay được đặt tại địa chỉ 179 đường Nguyễn Văn Cừ, quận Ninh Kiều, thành phố Cần Thơ.	đúng
23	Trường có các chương trình học bổng không?	Bạn cần đáp ứng các điều kiện sau: - Đạt điều kiện trúng tuyển của chương trình, ngành đào tạo trong cùng khóa tuyển sinh. - Trường phải có đủ các điều kiện bảo đảm chất lượng và chưa vượt quá năng lực đào tạo đối với chương trình, ngành đào tạo đó theo quy định của Bộ Giáo dục và Đào tạo. - Không thuộc diện sinh viên dự bị. - Tích lũy đủ học phần, số tín chỉ và hoàn thành các nội dung bắt buộc khác theo yêu cầu của chương trình đào tạo, đạt chuẩn đầu ra của chương trình. - Điểm trung bình tích lũy của toàn khóa học đạt từ trung bình trở lên. - Đóng học phí và bảo hiểm y tế đầy đủ, đúng thời hạn.	đúng
24	Sinh viên có thể chọn ngành sau khi nhập học không?	Không, sinh viên không thể chọn ngành sau khi nhập học. Sinh viên được xem xét chuyển sang học một chương trình, một ngành đào tạo khác khi có đủ các điều kiện sau: - Không đang là sinh viên trình độ năm thứ nhất hoặc năm cuối khóa, không thuộc diện bị xem xét buộc thôi học và còn đủ thời gian học tập theo quy định	đúng
25	Các môn đại cương có bắt buộc không?	Có, các môn đại cương là bắt buộc đối với sinh viên.	đúng
26	Có được thay đổi chuyên ngành sau khi đã vào học không?	Không, sinh viên không thể chọn ngành sau khi nhập học. Sinh viên được xem xét chuyển sang học một chương trình, một ngành đào tạo khác khi có đủ các điều kiện sau: - Không đang là sinh viên trình độ năm thứ nhất hoặc năm cuối khóa, không thuộc diện bị xem xét buộc thôi học và còn đủ thời gian học tập theo quy định	đúng
27	Sinh viên có	Sinh viên năm nhất có cơ hội thực hành tại bệnh	đúng

	được thực hành ở bệnh viện ngay từ năm nhất không?	viện.	
28	Trường có trung tâm nghiên cứu y học không?	Trường Đại học Y Dược Cần Thơ có thành lập trung tâm nghiên cứu y học. Cụ thể, trường có 4 trung tâm và 1 viện trực thuộc.	đúng
29	Thư viện của trường có mở cả cuối tuần không?	Thư viện trường Đại học Y Dược Cần Thơ có mở cửa vào thứ Bảy và Chủ nhật.	đúng
30	học phí ngành rhm	Học phí của ngành Răng hàm mặt : Hệ chính quy : - Đào tạo trúng tuyển chính quy : 1.385.400/tín chỉ - Đào tạo theo nhu cầu xã hội (theo đặt hàng) : 1.478.000/tín chỉ - Đào tạo hệ quốc tế tiếng anh : 4.000.000/tín chỉ	đúng

Bảng 3: Bảng mô tả hội thoại test với chatbot rasa

- Chatbot Rasa phản hồi các câu trả lời một cách nhanh chóng và chính xác trong hầu hết các trường hợp. Tuy nhiên, ở một số tình huống đặc biệt hoặc khi câu hỏi không rõ ràng, chatbot có thể đưa ra phản hồi chưa phù hợp. Điều này cho thấy tiềm năng cần cải thiện khả năng xử lý ngôn ngữ tự nhiên, đặc biệt đối với các câu hỏi phức tạp hoặc chứa lỗi chính tả. Việc tích hợp mô hình sửa lỗi và mở rộng tập dữ liệu huấn luyện có thể giúp tăng cường chất lượng phản hồi.

Tài liệu tham khảo

- [1] L. V. T. L. N. V. T. Đoàn Thị Hồng Phước, "XÂY DỰNG KHUNG ỨNG DỤNG AI CHATBOT TRONG LĨNH VỰC QUY CHẾ ĐÀO TẠO," [Online]. Available: https://www.researchgate.net/publication/371818126_XAY_DUNG_KHUNG_UNG_DUNG_AI_CHATBOT_TRONG_LINH_VUC_QUY_CHE DAO_TAO.
- [2] S. T. M. C. S. T. N. S. B. H. S. N. Đ. N. B. T. H. T. K. D. T. N. M. T. SV. Tăng Xuân Biên, "NGHIÊN CỨU, THIẾT KẾ TRỌNG LÝ ÁO CÁ NHÂN TRÊN NỀN TẢNG WEB KẾT HỢP HỌC MÁY (MACHINE LEARNING) VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN (NATURAL LANGUAGE PROCESSING)," [Online]. Available: <https://fee.hau.edu.vn/media/30/uffile-upload-no-title30611.pdf>.
- [3] J. F. N. P. a. A. N. T. Bocklisch, "Rasa: Open Source Language Understanding and Dialogue Management," [Online]. Available: <https://arxiv.org/pdf/1712.05181>.
- [4] T.-H. W. e. al, "A network-based end-to-end trainable task-oriented dialogue system," [Online]. Available: <https://aclanthology.org/E17-1042.pdf>.
- [5] A. Jiao, "An intelligent chatbot system based on entity extraction using RASA NLU and neural network," [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1487/1/012014/pdf>.
- [6] M. S. T.M.T Nguyen, "Целевой чат-бот на основе машинного обучения [A goal-oriented chatbot based on machine learning]," [Online]. Available: <https://moitvivt.ru/ru/journal/pdf?id=784>.
- [7] P. H. Quang, "Rasa chatbot: Tăng khả năng chatbot với custom component và custom tokenization(tiếng Việt tiếng Nhật)," [Online]. Available: <https://viblo.asia/p/rasa-chatbottang-kha-nang-chatbot-voi-custom-component-va-customtokenizationtieng-viet-tieng-nhat-Qbq5QN4mKD8>.
- [8] D.-T. Do, "Vsec: Transformer-based model for vietnamese spelling correction," [Online]. Available: <https://arxiv.org/pdf/2111.00640>.
- [9] L. V. T. L. N. V. T.] Đoàn Thị Hồng Phước, "Xây dựng khung ứng dụng ai chatbot trong lĩnh vực quy chế đào tạo," [Online]. Available: https://csdlkhoaahoc.hueuni.edu.vn/data/2023/6/3-Phuoc-6919-Article_Text-Morat-Tr39-52-Layout-Proof.pdf.
- [10] B. G. Bob Strahan, "Building a multi-channel Q&A chatbot at Saint Louis University using the open source QnABot," [Online]. Available:

<https://aws.amazon.com/vi/blogs/publicsector/building-a-multi-channel-qa-chatbot-at-saint-louis-university-using-the-open-source-qnabot/>.

- [11] L. Đ. A. H. T. H. N. T. Nguyễn Thành Trung, "NEU-Chatbot: Chatbot hỗ trợ tư vấn tuyển sinh Đại học Chính quy tại Trường," [Online]. Available: <https://daotao.neu.edu.vn/vi/hoat-dong-tu-van-tuyen-sinh/neu-chatbot-chatbot-ho-tro-tu-van-tuyen-sinh-dai-hoc-chinh-quy-tai-truong-dai-hoc-kinh-te-quoc-dan>.
- [12] B. Q. Manh, "Thử áp dụng mô hình dịch máy vào bài toán tự động sửa lỗi tiếng Việt," [Online]. Available: <https://viblo.asia/p/thu-ap-dung-mo-hinh-dich-may-vao-bai-toan-tu-dong-sua-loi-tieng-viet-maGK7vJB5j2>.