

**TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN**

--- 📖 ---



ĐỒ ÁN TỐT NGHIỆP

NGHIÊN CỨU KỸ THUẬT XỬ LÝ NGÔN NGỮ TỰ NHIÊN ỨNG DỤNG XÂY DỰNG CHATBOT

SINH VIÊN THỰC HIỆN : NGUYỄN TIẾN ĐẠT

MÃ SINH VIÊN : 1451020054

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN TIẾN ĐẠT

NGHIÊN CỨU KỸ THUẬT HIỂU NGÔN
NGŨ TỰ NHIÊN ỨNG DỤNG XÂY DỰNG
CHATBOT

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

MÃ SỐ: 74.80.201

NGƯỜI HƯỚNG DẪN: TS. TRẦN ĐĂNG CÔNG

HÀ NỘI - 2024

LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0, trí tuệ nhân tạo (AI) đang ngày càng trở thành một phần không thể thiếu trong nhiều lĩnh vực của cuộc sống. AI không chỉ thúc đẩy sự phát triển của các ngành công nghiệp truyền thống mà còn mở ra những hướng đi mới cho các ngành dịch vụ, giáo dục, y tế và thương mại điện tử. Một trong những ứng dụng nổi bật và tiềm năng của AI chính là xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), giúp máy tính có khả năng hiểu, phân tích và tương tác với con người thông qua ngôn ngữ tự nhiên. Trong số các ứng dụng của NLP, chatbot là một trong những công nghệ đang được quan tâm và phát triển mạnh mẽ, góp phần nâng cao trải nghiệm người dùng và tối ưu hóa quy trình làm việc.

Chatbot, một sản phẩm trực tiếp của NLP, đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Từ việc hỗ trợ chăm sóc khách hàng, trả lời các câu hỏi thường gặp, đến việc tư vấn sản phẩm và dịch vụ, chatbot đang dần thay thế con người trong những công việc lặp đi lặp lại và đòi hỏi tính chính xác cao. Với sự phát triển của các kỹ thuật học máy (machine learning) và học sâu (deep learning), các chatbot ngày nay không chỉ đơn thuần là trả lời câu hỏi theo kịch bản có sẵn mà còn có khả năng hiểu ngữ cảnh, phân tích ý định của người dùng và đưa ra các phản hồi thông minh và linh hoạt.

Xuất phát từ nhu cầu thực tiễn và tiềm năng to lớn của công nghệ này, đề tài "Nghiên cứu kỹ thuật xử lý ngôn ngữ tự nhiên ứng dụng xây dựng Chatbot" được lựa chọn nhằm mục tiêu tìm hiểu sâu về các kỹ thuật NLP hiện đại và áp dụng những kiến thức đó để phát triển một hệ thống chatbot thông minh. Hệ thống này không chỉ dừng lại ở việc trả lời các câu hỏi thông thường mà còn hướng tới việc hiểu ngữ cảnh và ý định của người dùng, từ đó đưa ra các phản hồi phù hợp, tạo ra trải nghiệm tương tác tự nhiên và hiệu quả.

Việc nghiên cứu và phát triển chatbot không chỉ giúp tối ưu hóa các quy trình làm việc, giảm thiểu chi phí vận hành mà còn mang lại những trải nghiệm mới mẻ và tiện lợi cho người dùng, góp phần thúc đẩy sự phát triển của xã hội. Em hy vọng rằng, những kết quả thu được từ nghiên cứu này sẽ là nền tảng cho các nghiên cứu và ứng dụng tiếp theo, mở ra những triển vọng mới cho việc ứng dụng AI trong cuộc sống hàng ngày.

LỜI CẢM ƠN

Em xin gửi lời cảm ơn trân thành đến thầy **TS. Trần Đăng Công** đã giúp đỡ em hoàn thiện đề tài này. Không những thế, trong quá trình học tập Thầy đã chỉ bảo và hướng dẫn tận tình cho em những kiến thức lý thuyết, cũng như các kỹ năng thực hành, cách giải quyết vấn đề. Thầy luôn là người truyền động lực cho em, giúp em hoàn thành tốt bản báo cáo. Em trân thành cảm ơn lãnh đạo cùng cán bộ, giảng viên khoa công nghệ thông tin trường Đại học Đại Nam đã tổ chức và hỗ trợ tạo mọi điều kiện để chúng em học tập và nghiên cứu tốt. Do mặt kiến thức và thời gian hạn chế nên báo cáo còn nhiều thiếu sót nhưng rất mong nhận được sự góp ý và nhận xét từ thầy cô để bài báo cáo của em được hoàn thiện tốt hơn.

Em xin trân thành cảm ơn!



LỜI CAM ĐOAN

Em cam đoan rằng tất cả các nội dung trong đồ án này là công sức của chính em và không sao chép từ bất kỳ nguồn thông tin nào khác mà không được trích dẫn. Tất cả các nguồn tài liệu, thông tin, ý kiến, hoặc dữ liệu từ bên ngoài đã được trích dẫn đầy đủ và chính xác trong phần tài liệu tham khảo. Em đã tuân thủ tất cả các quy định, quy tắc và quyền lợi về tác quyền, bản quyền và các quyền sở hữu trí tuệ khác khi sử dụng tài liệu từ nguồn bên ngoài. Tất cả các phần mềm, mã nguồn hoặc công cụ mà em đã sử dụng trong quá trình nghiên cứu và triển khai đồ án này đều được liệt kê và chỉ ra nguồn gốc một cách đầy đủ.

Hà Nội, ngày 14 tháng 6 năm 2024

Sinh viên thực hiện

This image shows a full page of primary-ruled paper. It features approximately 20 horizontal dashed lines spaced evenly down the page, providing a guide for handwriting practice. The background is white, and there are no margins or other markings present.



DANH MỤC VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết tắt đầy đủ
1	CDD	Conversation-Driven Development
2	NLU	Natural Language Understanding
3	NLP	Natural Language Processing
4	NLG	Natural Language Generation

DANH MỤC HÌNH ẢNH

Hình 1.1. Cách thức hoạt động trong Rasa (nguồn: ailab.siu.edu.vn).....	12
Hình 1.2. Kiến trúc tổng quát trong Rasa (nguồn: ailab.siu.edu.vn).....	12
Hình 3.1. Giao diện trang cài đặt thư viện hỗ trợ Visual C++	23
Hình 3.2. File cài đặt (nguồn: duythanhcse.wordpress.com)	24
Hình 3.3. Quá trình cài đặt thư viện hỗ trợ 1 (nguồn: duythanhcse.wordpress.com) ...	24
Hình 3.4. Quá trình cài đặt thư viện hỗ trợ 2 (nguồn: duythanhcse.wordpress.com) ...	24
Hình 3.5. Quá trình cài đặt thư viện hỗ trợ 3 (nguồn: duythanhcse.wordpress.com) ...	25
Hình 3.6. Cài đặt Anaconda (nguồn: duythanhcse.wordpress.com)	25
Hình 3.7. Quá trình cài đặt Anaconda 1 (nguồn: duythanhcse.wordpress.com).....	26
Hình 3.8. Quá trình cài đặt Anaconda 2 (nguồn: duythanhcse.wordpress.com).....	26
Hình 3.9. Quá trình cài đặt Anaconda 3 (nguồn: duythanhcse.wordpress.com).....	27
Hình 3.10. Quá trình cài đặt Anaconda 4 (nguồn: duythanhcse.wordpress.com).....	27
Hình 3.11. Quá trình cài đặt Anaconda 5 (nguồn: duythanhcse.wordpress.com).....	28
Hình 3.12. Quá trình cài đặt Anaconda 6 (nguồn: duythanhcse.wordpress.com).....	28
Hình 3.13. Quá trình cài đặt Anaconda 7 (nguồn: duythanhcse.wordpress.com).....	29
Hình 3.14. Quá trình cài đặt Anaconda 8 (nguồn: duythanhcse.wordpress.com).....	29
Hình 3.15. Quá trình cài đặt Anaconda 9 (nguồn: duythanhcse.wordpress.com).....	30
Hình 3.16. Quá trình cài đặt Anaconda 10 (nguồn: duythanhcse.wordpress.com).....	30
Hình 3.17. Quá trình cài đặt Anaconda 11 (nguồn: duythanhcse.wordpress.com).....	31
Hình 3.18. Quá trình khởi tạo Rasa 1 (nguồn: duythanhcse.wordpress.com).....	31
Hình 3.19. Quá trình khởi tạo Rasa 2 (nguồn: duythanhcse.wordpress.com).....	32
Hình 3.20. Quá trình khởi tạo Rasa 3 (nguồn: duythanhcse.wordpress.com).....	32
Hình 3.21. Quá trình khởi tạo Rasa 4 (nguồn: duythanhcse.wordpress.com).....	33
Hình 3.22. Quá trình khởi tạo Rasa 5 (nguồn: duythanhcse.wordpress.com).....	33
Hình 3.23. Quá trình khởi tạo Rasa 6 (nguồn: duythanhcse.wordpress.com).....	34
Hình 3.24. Quá trình khởi tạo Rasa 7 (nguồn: duythanhcse.wordpress.com).....	34
Hình 3.25. Quá trình kích hoạt Rasa 1 (nguồn: duythanhcse.wordpress.com)	35
Hình 3.26. Quá trình kích hoạt Rasa 2 (nguồn: duythanhcse.wordpress.com)	35
Hình 3.27. Quá trình cài đặt ujson 1 (nguồn: duythanhcse.wordpress.com)	36
Hình 3.28. Quá trình cài đặt ujson 2 (nguồn: duythanhcse.wordpress.com)	36
Hình 3.29. Quá trình cài đặt ujson 3 (nguồn: duythanhcse.wordpress.com)	37
Hình 3.30. Quá trình cài đặt ujson 4 (nguồn: duythanhcse.wordpress.com)	37
Hình 3.31. Quá trình cài đặt tensorflow 1 (nguồn: duythanhcse.wordpress.com)	38
Hình 3.32. Quá trình cài đặt tensorflow 2 (nguồn: duythanhcse.wordpress.com)	38
Hình 3.33. Quá trình cài đặt tensorflow 3 (nguồn: duythanhcse.wordpress.com)	39
Hình 3.34. Quá trình cài đặt tensorflow 4 (nguồn: duythanhcse.wordpress.com)	39
Hình 3.35. Quá trình cài đặt rasa 1 (nguồn: duythanhcse.wordpress.com).....	40
Hình 3.36. Quá trình cài đặt rasa 2 (nguồn: duythanhcse.wordpress.com).....	40
Hình 3.37. Quá trình cài đặt rasa 3 (nguồn: duythanhcse.wordpress.com).....	41
Hình 3.38. Quá trình cài đặt rasa 4 (nguồn: duythanhcse.wordpress.com).....	41
Hình 3.39. Quá trình cài đặt rasa 5 (nguồn: duythanhcse.wordpress.com).....	42
Hình 3.40. Cấu trúc dữ liệu (nguồn: rasa.com)	43

Hình 3.41. Kịch bản phân luồng chatbot.....	43
Hình 3.42. Ví dụ về intent của nhóm kinh doanh.....	44
Hình 3.43. Ví dụ về entity của nhóm kinh doanh.....	44
Hình 3.44. Ví dụ về responses của nhóm kinh doanh	44
Hình 3.45. pipeline	44
Hình 3.46. Ví dụ về EntitySynonymMapper.....	48
Hình 3.47. Định nghĩa phản hồi trong các tệp “nlu.yml” (rasa.com).....	49
Hình 3.48. policies.....	49
Hình 3.49. Định nghĩa các quy tắc trong tệp “rules.yml”	50
Hình 3.50. Tích hợp chatbot lên website 1 (nguồn: duythanhcse.wordpress.com)	51
Hình 3.51. Tích hợp chatbot lên website 2 (nguồn: duythanhcse.wordpress.com)	52
Hình 3.52. Tích hợp chatbot lên website 3 (nguồn: duythanhcse.wordpress.com)	52
Hình 3.53. Tích hợp chatbot lên website 4 (nguồn: duythanhcse.wordpress.com)	53
Hình 3.54. Tích hợp chatbot lên website 5 (nguồn: duythanhcse.wordpress.com)	53
Hình 3.55. Tích hợp chatbot lên website 6 (nguồn: duythanhcse.wordpress.com)	54
Hình 3.56. Tích hợp chatbot lên website 7 (nguồn: duythanhcse.wordpress.com)	54
Hình 3.57. Tích hợp chatbot lên website 8 (nguồn: duythanhcse.wordpress.com)	55
Hình 3.58. Tích hợp chatbot lên website 9 (nguồn: duythanhcse.wordpress.com)	55
Hình 3.59. Tích hợp chatbot lên website 10 (nguồn: duythanhcse.wordpress.com)	56
Hình 3.60. Tích hợp chatbot lên website 11 (nguồn: duythanhcse.wordpress.com)	56
Hình 3.61. Tích hợp chatbot lên website 12 (nguồn: duythanhcse.wordpress.com)	57
Hình 3.62. Tích hợp chatbot lên website 13 (nguồn: duythanhcse.wordpress.com)	57
Hình 3.63. Tích hợp chatbot lên website 14 (nguồn: duythanhcse.wordpress.com)	58
Hình 3.64. Tích hợp chatbot lên website 15 (nguồn: duythanhcse.wordpress.com)	58
Hình 3.65. Tích hợp chatbot lên website 16 (nguồn: duythanhcse.wordpress.com)	59
Hình 3.66. Tích hợp chatbot lên website 17 (nguồn: duythanhcse.wordpress.com)	60
Hình 3.67. Tích hợp chatbot lên website 18 (nguồn: duythanhcse.wordpress.com)	60
Hình 3.68. Tích hợp chatbot lên website 19 (nguồn: duythanhcse.wordpress.com)	61
Hình 3.69. Tích hợp chatbot lên website 20 (nguồn: duythanhcse.wordpress.com)	61
Hình 3.70. Tích hợp chatbot lên website 21 (nguồn: duythanhcse.wordpress.com)	62
Hình 3.71. Tích hợp chatbot lên website 22 (nguồn: duythanhcse.wordpress.com)	63
Hình 3.72. Tích hợp chatbot lên website 23 (nguồn: duythanhcse.wordpress.com)	63
Hình 3.73. Tích hợp chatbot lên website 24 (nguồn: duythanhcse.wordpress.com)	64
Hình 3.74. Tích hợp chatbot lên website 25 (nguồn: duythanhcse.wordpress.com)	64

MỤC LỤC

CHƯƠNG 1.	KHÁI QUÁT VỀ CHATBOT VÀ RASA OPEN SOURCE.....	1
1.1.	Tổng quan về Chatbot.....	1
1.1.1.	Bối cảnh	1
1.1.2.	Phát triển	3
1.1.3.	Phân loại chatbot.....	4
1.2.	Rasa Open Source.....	10
1.2.1.	Lịch sử hình thành và phát triển.....	10
1.2.2.	Thành phần chính của Rasa	11
1.2.3.	Kiến trúc tổng quát của Rasa	12
CHƯƠNG 2.	KỸ THUẬT XỬ LÝ NGÔN NGỮ TỰ NHIÊN TRONG CHATBOT	14
2.1.	Hiểu ngôn ngữ tự nhiên	14
2.1.1.	Kỹ thuật Tokenization	14
2.1.2.	Kỹ thuật Parsing	15
2.1.3.	Kỹ thuật Entity Recognition:	17
2.1.4.	Kỹ thuật Intent Classification	17
2.1.5.	Kỹ thuật Sentiment Analysis	18
2.2.	Xử lý ngôn ngữ tự nhiên.....	20
2.2.1.	NLP hoạt động như thế nào?.....	20
2.2.2.	Phương pháp xử lý ngôn ngữ tự nhiên	22
CHƯƠNG 3.	ỨNG DỤNG RASA OPEN SOURCE XÂY DỰNG CHATBOT	23
3.1.	Cài đặt và cấu hình Rasa Open Source.....	23
3.2.	Huấn luyện mô hình Rasa.....	42
3.2.1.	Converstation-Driven Development	42
3.2.2.	Tạo dữ liệu NLU	43
3.2.3.	Huấn luyện mô hình.....	44
3.3.	Tích hợp Chatbot với website.....	51

CHƯƠNG 1. KHÁI QUÁT VỀ CHATBOT VÀ RASA OPEN SOURCE

1.1. Tổng quan về Chatbot

Chatbot (ban đầu là chatterbot) là một ứng dụng phần mềm hoặc giao diện web được thiết kế để bắt chước cuộc trò chuyện của con người thông qua tương tác bằng văn bản hoặc giọng nói. Các chatbot hiện đại thường trực tuyến và sử dụng hệ thống trí tuệ nhân tạo tổng hợp có khả năng duy trì cuộc trò chuyện với người dùng bằng ngôn ngữ tự nhiên và mô phỏng cách con người cư xử với tư cách là một đối tác đàm thoại. Những chatbot như vậy thường sử dụng deep learning và xử lý ngôn ngữ tự nhiên, nhưng những chatbot đơn giản hơn đã tồn tại trong nhiều thập kỷ.

Kể từ cuối năm 2022, lĩnh vực này đã thu hút được sự chú ý rộng rãi nhờ sự phổ biến của ChatGPT của OpenAI, sau đó là các lựa chọn thay thế như Copilot của Microsoft và Gemini của Google. Những ví dụ như vậy phản ánh thực tiễn gần đây về việc xây dựng các sản phẩm đó dựa trên các mô hình ngôn ngữ lớn có nền tảng rộng, chẳng hạn như GPT-4 hoặc mô hình ngôn ngữ Gemini, được tinh chỉnh để nhắm mục tiêu các nhiệm vụ hoặc ứng dụng cụ thể (tức là mô phỏng cuộc trò chuyện của con người, trong trường hợp chatbot). Chatbots cũng có thể được thiết kế hoặc tùy chỉnh để nhắm mục tiêu hơn nữa vào các tình huống cụ thể hơn hoặc các miền chủ đề cụ thể.

Một lĩnh vực chính mà chatbot đã được sử dụng từ lâu là dịch vụ và hỗ trợ khách hàng, với nhiều loại trợ lý ảo khác nhau. Các công ty trải rộng trên nhiều ngành công nghiệp đã bắt đầu sử dụng các công nghệ trí tuệ nhân tạo tổng hợp mới nhất để thúc đẩy sự phát triển tiên tiến hơn trong các lĩnh vực đó.

1.1.1. Bối cảnh

Năm 1950, bài báo nổi tiếng "Máy tính và trí thông minh" của Alan Turing được xuất bản, đề xuất cái mà ngày nay gọi là phép thử Turing như một tiêu chí của trí thông minh. Tiêu chí này phụ thuộc vào khả năng chương trình máy tính mạo danh con người trong cuộc trò chuyện bằng văn bản theo thời gian thực với thẩm phán là con người đến mức thẩm phán không thể phân biệt một cách đáng tin cậy - chỉ dựa trên nội dung hội thoại - giữa chương trình và một con người thực sự. Danh tiếng của bài kiểm tra do Turing đề xuất đã kích thích sự quan tâm lớn đến chương trình ELIZA của Joseph Weizenbaum, xuất bản năm 1966, chương trình này dường như có thể đánh

lừa người dùng tin rằng họ đang trò chuyện với một con người thực sự. Tuy nhiên, bản thân Weizenbaum không khẳng định rằng ELIZA thực sự thông minh và phần giới thiệu bài báo của ông trình bày nó giống như một bài tập vạch trần:

“Trong trí tuệ nhân tạo, máy móc được tạo ra để hoạt động theo những cách kỳ diệu, thường đủ để làm choáng váng ngay cả những người quan sát giàu kinh nghiệm nhất. Nhưng một khi một chương trình cụ thể bị vạch trần, một khi hoạt động bên trong của nó được giải thích, phép thuật của nó sẽ tan biến; nó được tiết lộ chỉ là một tập hợp các thủ tục. Người quan sát tự nhủ: "Đáng lẽ tôi đã viết được điều đó". Với suy nghĩ đó, anh ấy chuyển chương trình đang được đề cập từ kệ được đánh dấu "thông minh" sang kệ dành riêng cho đồ tò mò. Mục tiêu của bài viết này là để việc đánh giá lại chương trình như vậy sắp được "giải thích". Rất ít chương trình cần nó nhiều hơn.”

Phương thức hoạt động chính của ELIZA (được các nhà thiết kế chatbot sao chép kể từ đó) liên quan đến việc nhận dạng các từ hoặc cụm từ đầu mỗi trong đầu vào và đầu ra của các phản hồi được chuẩn bị trước hoặc lập trình sẵn tương ứng có thể chuyển cuộc trò chuyện về phía trước theo cách rõ ràng có ý nghĩa (ví dụ: bằng cách trả lời bất kỳ nội dung đầu vào nào có chứa từ 'MOTHER' bằng 'TELL ME MORE about YOUR FAMILY'). Do đó, ảo tưởng về sự hiểu biết được tạo ra, mặc dù quá trình xử lý liên quan chỉ đơn thuần là bề ngoài. ELIZA đã chỉ ra rằng ảo tưởng như vậy dễ tạo ra một cách đáng ngạc nhiên bởi vì các thẩm phán của con người rất sẵn sàng đưa ra sự nghi ngờ khi các câu trả lời hội thoại có thể được hiểu là "thông minh".

Các nhà thiết kế giao diện đã đánh giá cao sự sẵn lòng của con người trong việc diễn giải đầu ra của máy tính dưới dạng đàm thoại thực sự - ngay cả khi nó thực sự dựa trên việc khớp mẫu khá đơn giản - có thể được khai thác cho các mục đích hữu ích. Hầu hết mọi người thích tham gia vào các chương trình giống con người và điều này mang lại cho các kỹ thuật kiểu chatbot một vai trò hữu ích trong các hệ thống tương tác cần lấy thông tin từ người dùng, miễn là thông tin đó tương đối đơn giản và thuộc các danh mục có thể dự đoán được. Do đó, ví dụ: hệ thống trợ giúp trực tuyến có thể sử dụng các kỹ thuật chatbot một cách hữu ích để xác định lĩnh vực trợ giúp mà người dùng yêu cầu, có khả năng cung cấp giao diện "thân thiện hơn" so với hệ thống menu hoặc tìm kiếm chính thức hơn. Cách sử dụng này có triển vọng chuyển công

nghệ chatbot từ "kệ ... dành riêng cho đồ tò mò" của Weizenbaum sang "phương pháp tính toán thực sự hữu ích" được đánh dấu đó.

1.1.2. Phát triển

Trong số những chatbot đầu tiên đáng chú ý nhất là ELIZA (1966) và PARRY (1972). Các chương trình đáng chú ý gần đây hơn bao gồm ALICE, Jabberwacky và DUDE (Agence Nationale de la Recherche và CNRS 2006). Trong khi ELIZA và PARRY chỉ được sử dụng để mô phỏng cuộc trò chuyện bằng văn bản, nhiều chatbot hiện bao gồm các tính năng chức năng khác, chẳng hạn như trò chơi và khả năng tìm kiếm trên web. Năm 1984, một cuốn sách có tên Bộ râu của cảnh sát được xây dựng một nửa đã được xuất bản, được cho là do chatbot Racter viết (mặc dù chương trình được phát hành sẽ không có khả năng làm như vậy).

Từ năm 1978 đến một thời gian sau năm 1983, dự án CYRUS do Janet Kolodner dẫn đầu đã xây dựng một chatbot mô phỏng Cyrus Vance (Ngoại trưởng thứ 57 của Hoa Kỳ). Nó sử dụng lý luận dựa trên từng trường hợp và cập nhật cơ sở dữ liệu hàng ngày bằng cách phân tích tin tức từ United Press International. Chương trình không thể xử lý các mục tin tức sau sự từ chức bất ngờ của Cyrus Vance vào tháng 4 năm 1980 và nhóm đã xây dựng một chatbot khác mô phỏng người kế nhiệm của ông, Edmund Muskie.

Một lĩnh vực nghiên cứu AI thích hợp là xử lý ngôn ngữ tự nhiên. Thông thường, các lĩnh vực AI yếu sử dụng phần mềm hoặc ngôn ngữ lập trình chuyên dụng được tạo riêng cho chức năng hẹp được yêu cầu. Ví dụ: ALICE sử dụng ngôn ngữ đánh dấu có tên AIML, dành riêng cho chức năng của nó như một tác nhân đàm thoại và từ đó đã được nhiều nhà phát triển khác của cái gọi là Alicebots áp dụng. Tuy nhiên, ALICE vẫn hoàn toàn dựa trên các kỹ thuật khớp mẫu mà không có bất kỳ khả năng suy luận nào, kỹ thuật tương tự mà ELIZA đã sử dụng vào năm 1966. Đây không phải là AI mạnh, đòi hỏi khả năng suy luận logic và nhạy bén.

Jabberwacky tìm hiểu các phản hồi và bối cảnh mới dựa trên tương tác của người dùng trong thời gian thực, thay vì được điều khiển từ cơ sở dữ liệu tĩnh. Một số chatbot gần đây hơn cũng kết hợp việc học theo thời gian thực với các thuật toán tiến hóa nhằm tối ưu hóa khả năng giao tiếp dựa trên mỗi cuộc trò chuyện được tổ chức.

Tuy nhiên, hiện tại không có trí tuệ nhân tạo đàm thoại có mục đích chung và một số nhà phát triển phần mềm tập trung vào khía cạnh thực tế, truy xuất thông tin.

Chatbots có thể sử dụng mạng lưới thần kinh nhân tạo làm mô hình ngôn ngữ. Ví dụ: máy biến áp được đào tạo trước tổng quát (GPT), sử dụng kiến trúc máy biến áp, đã trở nên phổ biến để xây dựng các chatbot phức tạp. "Đào tạo trước" trong tên của nó đề cập đến quá trình đào tạo ban đầu trên một kho văn bản lớn, cung cấp nền tảng vững chắc để mô hình thực hiện tốt các nhiệm vụ tiếp theo với lượng dữ liệu dành riêng cho nhiệm vụ hạn chế. Một ví dụ về chatbot GPT là ChatGPT. Bất chấp những lời chỉ trích về tính chính xác và xu hướng "ảo giác" - tức là tự tin đưa ra thông tin sai lệch và thậm chí trích dẫn các nguồn không tồn tại - ChatGPT đã thu hút được sự chú ý nhờ các phản hồi chi tiết và kiến thức lịch sử. Một ví dụ khác là BioGPT, được phát triển bởi Microsoft, tập trung vào việc trả lời các câu hỏi y sinh. Vào tháng 11 năm 2023, Amazon công bố một chatbot mới, có tên là "Q", để mọi người sử dụng tại nơi làm việc. DBpedia đã tạo ra một chatbot trong GSoC năm 2017. Nó có thể giao tiếp thông qua Facebook Messenger.

1.1.3. Phân loại chatbot

Có nhiều cách phân loại các loại chatbot phổ biến, phụ thuộc vào nhiều yếu tố như mục đích sử dụng, cách thức hoạt động, công nghệ sử dụng, và mức độ phát triển. Dưới đây là một phân loại phổ biến cho từng loại chatbot:

Chatbot dựa trên luật (Rule-based chatbots):

- Mô tả: Chatbot dựa trên luật, hay còn gọi là chatbot quy tắc, là loại chatbot hoạt động dựa trên hệ thống quy tắc và logic được lập trình sẵn. Loại chatbot này không sử dụng trí tuệ nhân tạo (AI) hay học máy (ML) để tự học hỏi và cải thiện khả năng tương tác.
 - Cách thức hoạt động:
 - Thiết lập quy tắc: Lập trình viên sẽ tạo ra một tập hợp các quy tắc và logic để xác định cách chatbot phản hồi với các câu hỏi và yêu cầu từ người dùng.
 - Xử lý đầu vào: Khi người dùng nhập câu hỏi hoặc yêu cầu, chatbot sẽ phân tích đầu vào và so sánh với các quy tắc đã được thiết lập.
-

- Trả lời: Chatbot sẽ lựa chọn quy tắc phù hợp nhất và trả lời người dùng dựa trên quy tắc đó.

- Ưu điểm:

- Dễ phát triển: Việc xây dựng chatbot dựa trên luật tương đối đơn giản và không yêu cầu kiến thức chuyên sâu về AI.

- Độ chính xác cao: Chatbot có thể cung cấp câu trả lời chính xác cho các câu hỏi và yêu cầu được lập trình sẵn.

- Hiệu quả: Chatbot có thể xử lý nhiều yêu cầu cùng lúc một cách nhanh chóng và hiệu quả.

- Nhược điểm:

- Thiếu tính linh hoạt: Chatbot không thể xử lý các câu hỏi hoặc yêu cầu nằm ngoài quy tắc đã được thiết lập.

- Khó mở rộng: Việc thêm mới các chức năng và khả năng cho chatbot tương đối phức tạp.

- Thiếu tính tự nhiên: Các câu trả lời của chatbot có thể thiếu tính tự nhiên và giống như máy móc.

- Ứng dụng:

- Dịch vụ khách hàng: Chatbot có thể cung cấp hỗ trợ khách hàng 24/7, trả lời các câu hỏi thường gặp và giải quyết các vấn đề đơn giản.

- Tiếp thị: Chatbot có thể thu thập thông tin leads, giới thiệu sản phẩm và dịch vụ, và thúc đẩy bán hàng.

- Giáo dục: Chatbot có thể cung cấp tài liệu học tập, trả lời các câu hỏi của học sinh và hỗ trợ giảng dạy.

- Chăm sóc sức khỏe: Chatbot có thể cung cấp thông tin về sức khỏe, tư vấn triệu chứng và đặt lịch hẹn khám.

Chatbot Dựa trên Machine Learning (ML-based chatbots):

- Mô tả: Chatbot dựa trên Machine Learning (ML) là thế hệ tiên tiến hơn so với chatbot theo luật (rule-based). Chúng tận dụng sức mạnh của học máy để tự học hỏi và cải thiện khả năng tương tác với người dùng theo thời gian.

- Cách thức hoạt động:



- Huấn luyện: Chatbot ML được huấn luyện trên một lượng lớn dữ liệu hội thoại. Dữ liệu này có thể bao gồm các cuộc trò chuyện thực tế giữa người với người, các câu hỏi thường gặp và câu trả lời, hoặc các tập dữ liệu được tạo riêng.

- Xác định ý định: Khi người dùng nhập câu hỏi hoặc yêu cầu, chatbot ML sử dụng các thuật toán học máy để xác định ý định đằng sau đầu vào.

- Tạo phản hồi: Dựa trên ý định được xác định, chatbot ML sẽ tạo ra phản hồi phù hợp cho người dùng. Quá trình này có thể liên quan đến việc truy xuất thông tin từ cơ sở dữ liệu bên ngoài hoặc tạo văn bản mới.

- Cải thiện theo thời gian: Với mỗi tương tác với người dùng, chatbot ML học hỏi và cải thiện khả năng xác định ý định và tạo ra phản hồi phù hợp.

- Ưu điểm:

- Tính linh hoạt cao: Chatbot ML có thể xử lý các câu hỏi và yêu cầu phức tạp, ngay cả những câu hỏi nằm ngoài phạm vi dữ liệu huấn luyện ban đầu.

- Tính tự nhiên: Chatbot ML có thể tạo ra các phản hồi giống người hơn, giúp cải thiện trải nghiệm người dùng.

- Khả năng tự học: Chatbot ML liên tục học hỏi và cải thiện theo thời gian, mà không cần lập trình lại thủ công.

- Nhược điểm:

- Phức tạp để phát triển: Xây dựng và huấn luyện chatbot ML đòi hỏi chuyên môn về học máy và xử lý ngôn ngữ tự nhiên (NLP).

- Yêu cầu nhiều dữ liệu: Hiệu quả của chatbot ML phụ thuộc rất nhiều vào chất lượng và số lượng dữ liệu huấn luyện.

- Tính minh bạch hạn chế: Khó có thể giải thích chính xác tại sao chatbot ML đưa ra một phản hồi cụ thể.

- Ứng dụng:

- Dịch vụ khách hàng: Giải quyết các vấn đề phức tạp, cung cấp hỗ trợ theo ngữ cảnh và cá nhân hóa trải nghiệm người dùng.

- Trợ lý ảo: Hỗ trợ người dùng trong các tác vụ hàng ngày, chẳng hạn như đặt lịch hẹn, đặt hàng và tìm kiếm thông tin.

- Giải trí: Cung cấp trải nghiệm trò chuyện thú vị và tương tác, tạo bạn đồng hành ảo.

- Giáo dục cá nhân hóa: Hỗ trợ học tập theo nhịp độ riêng của từng người, giải đáp thắc mắc chi tiết.

Chatbot dựa trên AI (AI-based chatbots):

- Mô tả: Chatbot dựa trên AI là một thuật ngữ chung để chỉ các chatbot sử dụng trí tuệ nhân tạo (AI) để tương tác với người dùng. Loại chatbot này bao gồm chatbot dựa trên luật (rule-based) và chatbot dựa trên học máy (ML), cũng như các chatbot sử dụng các kỹ thuật AI tiên tiến hơn như xử lý ngôn ngữ tự nhiên (NLP) và hiểu ngôn ngữ tự nhiên (NLU).

- Cách thức hoạt động:

- Chatbot AI sử dụng các thuật toán AI để hiểu ý định của người dùng và tạo ra phản hồi phù hợp.
- Quá trình này có thể liên quan đến việc phân tích văn bản, xác định các từ khóa và ngữ cảnh, truy xuất thông tin từ cơ sở dữ liệu và tạo văn bản mới.
- Chatbot AI có thể học hỏi và cải thiện theo thời gian bằng cách sử dụng các kỹ thuật như học máy và tăng cường học tập.

- Ưu điểm:

- Tính linh hoạt cao: Chatbot AI có thể xử lý các câu hỏi và yêu cầu phức tạp, ngay cả những câu hỏi nằm ngoài phạm vi dữ liệu huấn luyện ban đầu.
- Tính tự nhiên: Chatbot AI có thể tạo ra các phản hồi giống người hơn, giúp cải thiện trải nghiệm người dùng.
- Khả năng tự học: Chatbot AI liên tục học hỏi và cải thiện theo thời gian, mà không cần lập trình lại thủ công.
- Cá nhân hóa: Chatbot AI có thể cá nhân hóa trải nghiệm cho từng người dùng dựa trên sở thích, hành vi và lịch sử tương tác của họ.

- Nhược điểm:

- Phức tạp để phát triển: Xây dựng và huấn luyện chatbot AI đòi hỏi chuyên môn về AI, NLP và NLU.
 - Yêu cầu nhiều dữ liệu: Hiệu quả của chatbot AI phụ thuộc rất nhiều vào chất lượng và số lượng dữ liệu huấn luyện.
-

- Tính minh bạch hạn chế: Khó có thể giải thích chính xác tại sao chatbot AI đưa ra một phản hồi cụ thể.

- Vấn đề đạo đức: Chatbot AI có thể bị sử dụng để lan truyền thông tin sai lệch, phân biệt đối xử và các hành vi có hại khác.

- Ứng dụng:

- Dịch vụ khách hàng: Giải quyết các vấn đề phức tạp, cung cấp hỗ trợ theo ngữ cảnh và cá nhân hóa trải nghiệm người dùng.

- Trợ lý ảo: Hỗ trợ người dùng trong các tác vụ hàng ngày, chẳng hạn như đặt lịch hẹn, đặt hàng và tìm kiếm thông tin.

- Giải trí: Cung cấp trải nghiệm trò chuyện thú vị và tương tác, tạo bạn đồng hành ảo.

- Giáo dục cá nhân hóa: Hỗ trợ học tập theo nhịp độ riêng của từng người, giải đáp thắc mắc chi tiết.

- Chăm sóc sức khỏe: Cung cấp thông tin về sức khỏe, tư vấn triệu chứng và đặt lịch hẹn khám.

Chatbot Dựa trên Xử lý Ngôn ngữ Tự Nhiên (Natural Language Processing - NLP chatbots):

- Mô tả: Chatbot dựa trên Xử lý Ngôn ngữ Tự Nhiên (NLP) là một nhánh của chatbot AI, tập trung vào khả năng hiểu và xử lý ngôn ngữ của con người một cách tự nhiên. Chúng sử dụng các kỹ thuật NLP tiên tiến để phân tích văn bản, hiểu ngữ nghĩa và tạo ra phản hồi phù hợp với ý định của người dùng.

- Cách thức hoạt động:

- Xử lý đầu vào: Khi người dùng nhập câu hỏi hoặc yêu cầu, chatbot NLP sẽ sử dụng các kỹ thuật NLP để phân tích đầu vào.

- Phân tích cú pháp và ngữ nghĩa: Chatbot phân tích cấu trúc câu (cú pháp) và ý nghĩa của các từ ngữ (ngữ nghĩa) để hiểu ý định của người dùng.

- Xác định thực thể: Chatbot xác định các thực thể quan trọng trong câu, chẳng hạn như người, địa điểm, ngày tháng, tổ chức, v.v.

- **Tìm kiếm thông tin:** Dựa trên ý định được xác định và các thực thể được trích xuất, chatbot có thể tìm kiếm thông tin từ các nguồn khác nhau như cơ sở dữ liệu nội bộ, web hoặc API bên ngoài.

- **Tạo phản hồi:** Cuối cùng, chatbot NLP sẽ tạo ra phản hồi phù hợp với ý định của người dùng, sử dụng ngôn ngữ tự nhiên và ngữ cảnh phù hợp.

- **Ưu điểm:**

- **Hiểu ngôn ngữ tự nhiên:** Chatbot NLP có thể hiểu được các câu hỏi phức tạp, câu hỏi ngoài ngữ liệu và cách diễn đạt khác nhau.

- **Cải thiện trải nghiệm người dùng:** Các cuộc trò chuyện với chatbot NLP trở nên tự nhiên hơn, giống như tương tác với người thật.

- **Cá nhân hóa:** Chatbot NLP có thể sử dụng thông tin từ các cuộc trò chuyện trước đó để cá nhân hóa trải nghiệm cho từng người dùng.

- **Nhược điểm:**

- **Phức tạp để phát triển:** Xây dựng và huấn luyện chatbot NLP đòi hỏi chuyên môn về NLP và kỹ thuật xử lý ngôn ngữ.

- **Yêu cầu nhiều dữ liệu huấn luyện:** Hiệu quả của chatbot NLP phụ thuộc rất nhiều vào chất lượng và số lượng dữ liệu huấn luyện được cung cấp.

- **Giới hạn về kiến thức:** Chatbot NLP có thể gặp khó khăn với các chủ đề phức tạp hoặc các câu hỏi yêu cầu kiến thức nền tảng rộng.

- **Ứng dụng:**

- **Dịch vụ khách hàng:** Giải đáp các thắc mắc phức tạp của khách hàng, cung cấp hỗ trợ theo ngữ cảnh và cá nhân hóa trải nghiệm.

- **Trợ lý ảo thông minh:** Hỗ trợ người dùng trong các tác vụ hàng ngày, chẳng hạn như đặt lịch hẹn, đặt hàng, tìm kiếm thông tin và thậm chí cả giải trí.

- **Giáo dục cá nhân hóa:** Hỗ trợ học tập theo nhịp độ riêng của từng học sinh, trả lời các câu hỏi phức tạp và giải thích các khái niệm khó.

- **Y tế và chăm sóc sức khỏe:** Cung cấp thông tin về bệnh tật, triệu chứng, tư vấn chăm sóc sức khỏe cơ bản và hỗ trợ đặt lịch khám bác sĩ.

1.2. Rasa Open Source

Với hơn 25 triệu lượt tải xuống, Rasa Open Source là mã nguồn mở phổ biến nhất để xây dựng trợ lý AI dựa trên giọng nói và trò chuyện.

Rasa Open Source là một nền tảng AI đàm thoại nguồn mở cho phép bạn hiểu và tổ chức các cuộc hội thoại cũng như kết nối với các kênh nhắn tin và hệ thống của bên thứ ba thông qua một bộ API. Nó cung cấp các khối xây dựng để tạo trợ lý ảo (kỹ thuật số) hoặc chatbot.

1.2.1. Lịch sử hình thành và phát triển

Năm 2016:

- Thành lập công ty: Alan Nichol và Alexander Weidauer thành lập Rasa Technologies GmbH.
- Ra mắt Rasa NLU và Rasa Core: Phát hành phiên bản đầu tiên của Rasa NLU và Rasa Core. Rasa NLU được sử dụng để xử lý và hiểu ngôn ngữ tự nhiên, trong khi Rasa Core được sử dụng để quản lý luồng trò chuyện và hành vi của bot.

Năm 2017:

- Phát triển và cải thiện: Rasa Open Source tiếp tục phát triển và cải thiện với sự đóng góp từ cộng đồng người dùng.
- Tăng cường tính năng: Các tính năng mới được thêm vào, bao gồm hỗ trợ cho multiple intents, các actions đa bước, và khả năng huấn luyện dữ liệu từ đầu.

Năm 2018:

- Ra mắt Rasa 1.0: Rasa công bố phiên bản 1.0 của Rasa Open Source, với nhiều cải tiến về hiệu suất và tính ổn định, đánh dấu một bước quan trọng trong sự phát triển của dự án.
- Hỗ trợ ngôn ngữ khác: Rasa bắt đầu hỗ trợ ngôn ngữ khác ngoài tiếng Anh, mở rộng khả năng sử dụng cho người dùng toàn cầu.

Năm 2019:

- Tích hợp với các nền tảng: Rasa tiếp tục cải thiện tích hợp với các nền tảng phổ biến như Facebook Messenger, Slack, và các kênh trò chuyện khác.
- Học tăng cường: Sự ra mắt của tính năng học tăng cường giúp chatbot cải thiện hiệu suất thông qua phản hồi từ người dùng.

Năm 2020:

- Tăng cường học tập tự động: Rasa tiếp tục cải thiện khả năng học tập tự động của chatbot thông qua việc tích hợp các thuật toán học máy mới nhất.
- Cộng đồng người dùng phát triển mạnh mẽ: Số lượng người dùng và nhà phát triển sử dụng Rasa ngày càng tăng, tạo ra một cộng đồng lớn và đa dạng.

Năm 2021:

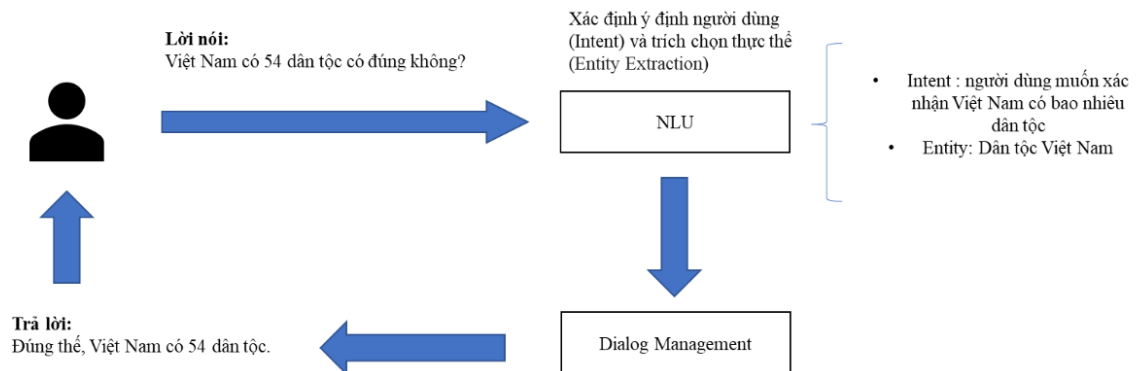
- Cải tiến và tối ưu hóa: Rasa tiếp tục cải tiến và tối ưu hóa hiệu suất của nền tảng, bao gồm cải thiện trong việc xử lý ngôn ngữ tự nhiên và tăng cường tính ổn định của hệ thống.
- Mở rộng tích hợp và tính năng: Các tính năng mới được thêm vào để mở rộng khả năng tích hợp và linh hoạt của Rasa, giúp người dùng xây dựng các ứng dụng trò chuyện phức tạp hơn.

1.2.2. Thành phần chính của Rasa

Rasa gồm 2 phần chính là Natural Language Understanding và dialogue management :

- Natural Language Understanding (NLU): Đây là một module đóng vai trò xử lý đầu vào của cuộc hội thoại (tokenizer, featurize). NLU sẽ xác định ý định của người dùng muốn nói (Intent Classification), trích xuất và chọn ra các thông tin chính. (Entity Extraction).
 - Dialogue management: Sau khi hiểu được câu nói của người dùng, module Dialogue management sẽ quyết định hành động tiếp theo trong hội thoại dựa trên các thông tin chính đã được trích xuất thông các chính sách (Policy) sau:
 - RulePolicy: Sử dụng những quy tắc(Rule) đã được định nghĩa trước và đưa ra quyết định.
-

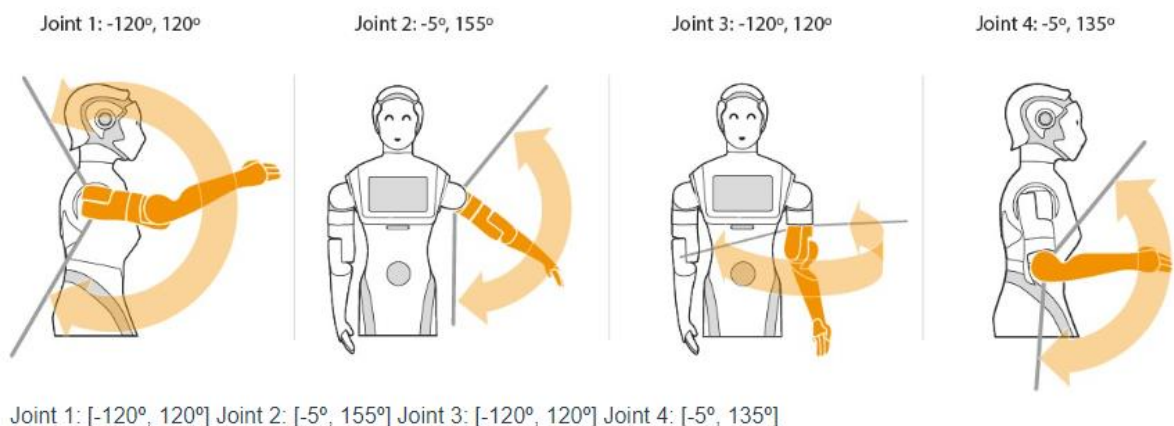
- TEDPolicy (Transformer Embedding Dialogue Policy): Sử dụng học sâu để xác định các hành động tiếp theo



Hình 1.1. Cách thức hoạt động trong Rasa (nguồn: ailab.siu.edu.vn)

1.2.3. Kiến trúc tổng quát của Rasa

Ngoài hai thành phần chính đã được nêu ở mục trên thì hệ thống cần thêm phần để xử lý bộ nhớ, kết nối các thành phần lại với nhau để hoạt động hiệu quả.



Hình 1.2. Kiến trúc tổng quát trong Rasa (nguồn: ailab.siu.edu.vn)

- NLU Pipeline: Tập hợp các module NLU dùng để phân loại các ý định (intent) và trích xuất thực thể (Entity).
- Dialog Policies: Tập hợp các chính sách (policy) hỗ trợ đưa ra các quyết định hành động tiếp theo.
- Action Server: Cho phép người dùng có thể tùy chỉnh và chạy một đoạn code python bất kỳ như gọi API, truy vấn cơ sở dữ liệu,... hay bất kì việc gì tùy vào mục đích của người dùng.

- Tracker Store: Module lưu trữ các giá trị của chatbot.
- Lock Store: Cơ chế đảm bảo rằng cuộc trò chuyện diễn ra đúng thứ tự để tránh tình trạng khi một hệ thống cố gắng thực hiện nhiều hành động cùng một lúc hành động.
- Filesystem: Lưu trữ các file của một dự án.
- Agent: Module xử lý chung và kết nối các thành phần khác.

RASA là framework rất mạnh mẽ cho phát triển chatbot, ngay cả với người mới chưa từng có kinh nghiệm qua lĩnh vực Conversational AI hoặc lập trình, có thể hoạt động tốt với lượng dữ liệu ít. Tuy nhiên, để chatbot RASA thực hiện được những bài toán khó, đòi hỏi chúng ta phải nắm rõ, hiểu rõ từng thành phần để xây dựng hoặc điều chỉnh lại theo mục đích sử dụng riêng.



CHƯƠNG 2. KỸ THUẬT XỬ LÝ NGÔN NGỮ TỰ NHIÊN TRONG CHATBOT

2.1. Hiểu ngôn ngữ tự nhiên

Hiểu ngôn ngữ tự nhiên (Natural Language Understanding - NLU) là một lĩnh vực trong trí tuệ nhân tạo (AI) tập trung vào việc máy tính hiểu và xử lý ngôn ngữ tự nhiên của con người. Nó bao gồm các phương pháp và thuật toán để giải quyết các vấn đề như nhận dạng từ ngữ, phân tích cú pháp, xác định ý định và trích xuất thông tin từ văn bản hoặc giọng nói.

Các phương pháp NLU thường bao gồm:

- Tokenization: Phân tích văn bản thành các phần tử nhỏ hơn như từ, câu hoặc đoạn văn.
- Parsing: Phân tích cú pháp của câu để hiểu cấu trúc và mối quan hệ giữa các phần tử.
- Entity Recognition: Nhận dạng và phân loại các thực thể như người, địa điểm, thời gian, số lượng, v.v.
- Intent Recognition: Xác định ý định hoặc mục đích của người dùng dựa trên văn bản hoặc giọng nói.
- Sentiment Analysis: Phân tích cảm xúc hoặc quan điểm được diễn đạt trong văn bản.

2.1.1. Kỹ thuật Tokenization

Tokenization là quá trình chia một đoạn văn bản thành các phần tử nhỏ hơn gọi là "tokens". Tokens có thể là các từ, dấu câu, hoặc các đơn vị khác như số, ký tự đặc biệt, hoặc thậm chí là các đoạn văn ngắn. Quá trình tokenization là bước quan trọng đầu tiên trong xử lý ngôn ngữ tự nhiên và NLP, vì nó giúp máy tính hiểu được cấu trúc của văn bản để có thể xử lý và phân tích nó một cách hiệu quả.

Một số phương pháp tokenization phổ biến:

- Tokenization dựa trên dấu cách (Whitespace Tokenization):
-

- Phương pháp đơn giản nhất là chia văn bản thành các token bằng cách sử dụng dấu cách hoặc khoảng trắng làm dấu phân tách.

- Ví dụ: "Tôi yêu học máy" sẽ được chia thành ["Tôi", "yêu", "học", "máy"].

- Tokenization dựa trên từ (Word Tokenization):

- Phương pháp này chia văn bản thành các token dựa trên các từ cụ thể.
- Trong trường hợp này, dấu câu được coi là các token riêng biệt.
- Ví dụ: "Điều này là một câu." sẽ được chia thành ["Điều", "này", "là", "một", "câu", "."].

- Tokenization dựa trên ký tự (Character Tokenization):

- Văn bản được chia thành các token dựa trên các ký tự riêng biệt.
- Đây là phương pháp hữu ích khi muốn xử lý các đoạn văn bản ngắn hoặc khi muốn giữ nguyên các thông tin về ký tự đặc biệt.
- Ví dụ: "hello" sẽ được chia thành ["h", "e", "l", "l", "o"].

- Tokenization dựa trên biểu thức chính quy (Regex Tokenization):

- Sử dụng biểu thức chính quy để xác định các quy tắc tokenization.
- Có thể tùy chỉnh để xử lý các trường hợp đặc biệt như email, địa chỉ web, số điện thoại, v.v.

- Tokenization n-gram:

- Chia văn bản thành các n-gram, tức là các cặp hoặc nhóm gồm n từ liên tiếp.
- Ví dụ: "Tôi thích học máy" có thể được chia thành ["Tôi thích", "thích học", "học máy"].

Quá trình tokenization không chỉ đơn giản là chia câu thành các từ, mà còn phụ thuộc vào ngữ cảnh và mục tiêu cụ thể của ứng dụng. Đôi khi, cùng một đoạn văn bản có thể được tokenized theo nhiều cách khác nhau, tùy thuộc vào mục đích sử dụng cụ thể của mô hình hoặc ứng dụng NLP.

2.1.2. Kỹ thuật Parsing

Parsing là quá trình phân tích và phân loại cú pháp của một chuỗi ký tự hoặc một câu để hiểu cấu trúc và mối quan hệ giữa các thành phần của nó. Trong ngữ cảnh

của ngôn ngữ tự nhiên và NLP, parsing thường được sử dụng để phân tích cú pháp của câu văn để hiểu ý nghĩa của nó. Một số khía cạnh quan trọng của parsing:

- **Cây cú pháp (Parse Tree):** Trong parsing, một cây cú pháp được tạo ra để biểu diễn cấu trúc cú pháp của câu. Mỗi nút trong cây đại diện cho một phần của câu, ví dụ như từ, cụm từ hoặc mệnh đề, và mối quan hệ giữa chúng được biểu diễn bằng các cạnh.

- **Loại Parsing:** Có nhiều phương pháp parsing khác nhau được sử dụng trong NLP, bao gồm:

- **Top-down Parsing:** Bắt đầu từ nguyên tắc và cố gắng tạo ra cây cú pháp từ trên xuống dưới.

- **Bottom-up Parsing:** Bắt đầu từ các thành phần cơ bản và kết hợp chúng thành cấu trúc lớn hơn.

- **Chart Parsing:** Sử dụng một bảng phương pháp để lưu trữ các phần của câu và các cấu trúc có thể có của chúng.

- **Dependency Parsing:** Tập trung vào xác định các mối quan hệ phụ thuộc giữa các từ trong câu.

- **Ứng dụng của Parsing:** Parsing là một phần quan trọng của nhiều ứng dụng NLP, bao gồm:

- **Hiểu ngôn ngữ tự nhiên (NLU):** Để hiểu ý nghĩa của một câu văn, parsing được sử dụng để xác định cấu trúc cú pháp và mối quan hệ giữa các phần của câu.

- **Dịch máy:** Trong dịch máy, parsing giúp phân tích cấu trúc câu nguồn và xây dựng câu đích tương ứng.

- **Xử lý ngôn ngữ tự nhiên (NLP):** Parsing được sử dụng để trích xuất thông tin, phân tích cú pháp, và hiểu ý nghĩa của văn bản.

- **Thách thức của Parsing:** Mặc dù có các công cụ và thuật toán parsing phức tạp, việc parsing vẫn đối mặt với một số thách thức, bao gồm đa nghĩa từ ngữ, ngữ cảnh phức tạp và sự đa dạng của ngôn ngữ tự nhiên.

2.1.3. *Kỹ thuật Entity Recognition:*

Entity Recognition (hay còn gọi là Named Entity Recognition - NER) là quá trình trong xử lý ngôn ngữ tự nhiên (NLP) nhằm nhận dạng và phân loại các thực thể quan trọng trong văn bản như tên riêng, địa chỉ, ngày tháng, số lượng, tổ chức, và các loại thực thể khác. Mục tiêu của NER là xác định các phần của văn bản mà có ý nghĩa hoặc giá trị đặc biệt cho ứng dụng hoặc mục đích cụ thể.

- Loại thực thể: Các loại thực thể cần được nhận dạng trong một văn bản có thể rất đa dạng, bao gồm những thực thể như người, địa điểm, tổ chức, thời gian, tiền tệ, số lượng, sự kiện, v.v.
- Phân loại đa dạng: Mỗi thực thể có thể thuộc vào một hoặc nhiều loại khác nhau. Ví dụ, "Barack Obama" có thể được phân loại là một người và cũng là một tên riêng.
- Contextual Understanding: Việc hiểu ngữ cảnh là rất quan trọng trong việc nhận dạng thực thể. Ví dụ, trong câu "Tôi đang sống ở Paris", "Paris" có thể là một địa danh, nhưng trong câu "Paris là một thành phố đẹp", "Paris" là một địa danh hoặc một tổ chức.
- Phụ thuộc vào ngữ cảnh: Phân tích thực thể thường phụ thuộc vào ngữ cảnh của câu hoặc đoạn văn. Ví dụ, trong một cuộc hội thoại về du lịch, "Paris" có thể được nhận dạng là một địa danh, trong khi trong một cuộc trò chuyện về chính trị, "Paris" có thể được nhận dạng là một tổ chức.
- Kỹ thuật phân loại: Có nhiều kỹ thuật để thực hiện NER, bao gồm các phương pháp dựa trên quy tắc, học có giám sát, học không giám sát, và kết hợp cả hai.

Công nghệ Entity Recognition đóng vai trò quan trọng trong nhiều ứng dụng NLP như tóm tắt văn bản, phân loại văn bản, trích xuất thông tin, dịch máy, và chatbot, giúp máy tính hiểu và xử lý ngôn ngữ tự nhiên một cách hiệu quả và tự động.

2.1.4. *Kỹ thuật Intent Classification*

Intent Classification là quá trình trong Natural Language Understanding (NLU) mà hệ thống phân loại các yêu cầu hoặc ý định của người dùng dựa trên các câu hoặc đoạn văn họ nhập vào. Mục tiêu của Intent Classification là xác định mục đích hoặc ý

định cụ thể của người dùng khi họ tương tác với hệ thống, giúp hệ thống hiểu rõ và phản hồi đúng đắn.

- Tập dữ liệu đào tạo: Một tập dữ liệu đào tạo lớn và đa dạng là cần thiết để huấn luyện mô hình Intent Classification hiệu quả. Tập dữ liệu này cần chứa các câu mẫu đã được gán nhãn với các ý định tương ứng.
- Biểu diễn câu (Sentence Representation): Để huấn luyện mô hình, các câu cần được biểu diễn thành các vector số học. Phương pháp phổ biến là sử dụng word embeddings hoặc các phương pháp biểu diễn câu phức tạp hơn như Transformers.
- Kiến trúc mô hình: Các mô hình được sử dụng cho Intent Classification có thể là các mô hình học máy cơ bản như máy học tuyến tính hoặc mạng nơ-ron như LSTM, CNN, hoặc các mô hình biến thể như BERT, GPT.
- Tối ưu hóa và đánh giá mô hình: Sau khi xây dựng mô hình, quá trình tối ưu hóa nhằm cải thiện độ chính xác và hiệu suất của mô hình là cần thiết. Đánh giá mô hình được thực hiện thông qua các phương pháp như cross-validation hoặc sử dụng tập dữ liệu kiểm tra độc lập.
- Xử lý class imbalance: Trong thực tế, các nhóm ý định có thể có số lượng mẫu không cân bằng. Việc xử lý class imbalance là một phần quan trọng của quá trình huấn luyện để đảm bảo rằng mô hình không bị thiên vị theo các nhóm nhiều mẫu.
- Tinh chỉnh và cải thiện: Sau khi xây dựng mô hình ban đầu, quá trình tinh chỉnh và cải thiện liên tục là cần thiết để đảm bảo rằng mô hình hoạt động tốt trên các dữ liệu mới và trong các tình huống thực tế.

Intent Classification đóng vai trò quan trọng trong việc xây dựng các hệ thống trò chuyện tự động như chatbot, ứng dụng hỗ trợ khách hàng, và các ứng dụng NLP khác, vì nó là khâu cơ bản nhất trong việc hiểu ý định của người dùng từ ngôn ngữ tự nhiên.

2.1.5. Kỹ thuật Sentiment Analysis

Sentiment Analysis là quá trình tự động phân tích và nhận diện cảm xúc hoặc quan điểm được diễn đạt trong văn bản hoặc dữ liệu đầu vào khác. Mục tiêu của Sentiment Analysis là xác định xem một phần của văn bản có tính tích cực, tiêu cực hoặc trung lập.

Có ba cấp độ chính trong Sentiment Analysis:

- Sentiment Polarity: Phân loại cảm xúc thành tích cực, tiêu cực hoặc trung tính.
- Sentiment Subjectivity: Đo lường mức độ chủ quan của cảm xúc trong văn bản.
- Sentiment Intensity: Xác định mức độ mạnh mẽ của cảm xúc được diễn đạt.

Các phương pháp thường được sử dụng trong Sentiment Analysis bao gồm:

- Machine Learning: Sử dụng các mô hình máy học để huấn luyện từ dữ liệu được gán nhãn về cảm xúc.
- Lexicon-based: Dựa trên từ điển để gán điểm cảm xúc cho các từ hoặc cụm từ trong văn bản.
- Rule-based: Sử dụng các quy tắc và luật ngữ pháp để xác định cảm xúc.

Sentiment Analysis có nhiều ứng dụng thực tế, bao gồm:

- Social Media Monitoring: Theo dõi và phân tích cảm xúc của người dùng trên các nền tảng mạng xã hội như Twitter, Facebook, và Instagram.
- Customer Feedback Analysis: Phân tích phản hồi của khách hàng từ các đánh giá sản phẩm hoặc dịch vụ trên các trang web thương mại điện tử.
- Brand Reputation Management: Theo dõi và đánh giá cảm nhận của công chúng đối với thương hiệu và sản phẩm của mình.
- Financial Market Analysis: Dự đoán xu hướng thị trường dựa trên cảm xúc và quan điểm của nhà đầu tư được diễn đạt trên các diễn đàn tài chính.

Tuy nhiên, Sentiment Analysis cũng có những hạn chế như:

- Ngôn ngữ phức tạp: Một số văn bản có ngữ cảnh phức tạp hoặc sử dụng ngôn ngữ hài hước, nghệ thuật, điều này có thể làm cho việc phân tích cảm xúc trở nên khó khăn.
 - Đa nghĩa: Một từ hoặc cụm từ có thể có nhiều ý nghĩa, điều này có thể dẫn đến kết quả phân tích không chính xác.
 - Đánh giá không chính xác: Mô hình Sentiment Analysis có thể không hiệu quả khi đối mặt với ngữ cảnh mới hoặc từ ngôn ngữ chưa được huấn luyện đầy đủ.
-

Khi áp dụng vào hệ thống chatbot, NLU giúp chatbot hiểu và phản hồi một cách tự nhiên và linh hoạt hơn. Nhờ vào NLU, chatbot có thể:

- Nhận diện ý định: NLU giúp chatbot nhận biết và hiểu rõ mục đích hoặc ý muốn của người dùng từ những gì họ viết hoặc nói. Bằng cách phân tích cú pháp và từ vựng của câu, NLU có thể xác định ý định của người dùng, từ đó giúp chatbot cung cấp phản hồi chính xác và đáp ứng nhu cầu của họ.
- Xử lý ngôn ngữ phức tạp: NLU có khả năng hiểu và xử lý các cấu trúc ngôn ngữ phức tạp như ngôn ngữ tự nhiên. Điều này cho phép chatbot hiểu được các câu trả lời phức tạp, các yêu cầu phức tạp hoặc ngôn ngữ không chuẩn và vẫn có thể cung cấp phản hồi chính xác và hữu ích.
- Trích xuất thông tin: NLU có thể nhận biết và trích xuất thông tin quan trọng từ văn bản hoặc giọng nói của người dùng. Điều này giúp chatbot tự động hóa quá trình trả lời câu hỏi hoặc cung cấp thông tin một cách nhanh chóng và chính xác.
- Phản hồi một cách linh hoạt: Dựa trên ý định của người dùng được nhận dạng bởi NLU, chatbot có thể cung cấp câu trả lời phù hợp hoặc yêu cầu thêm thông tin một cách linh hoạt. Điều này tạo ra trải nghiệm tương tác tự nhiên và thoải mái cho người dùng khi tương tác với chatbot.

2.2. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (NLP) là một công nghệ máy học, cung cấp cho máy tính khả năng diễn giải, tương tác và hiểu được ngôn ngữ của con người. Các tổ chức ngày nay có khối lượng lớn dữ liệu thoại và văn bản từ nhiều kênh liên lạc khác nhau như email, tin nhắn văn bản, bài đăng trên mạng xã hội, tệp video, tệp âm thanh và nhiều hơn nữa. Họ sử dụng phần mềm NLP để tự động xử lý dữ liệu này, phân tích ý định hoặc cảm xúc trong tin nhắn và phản hồi bằng người thật theo thời gian thực.

2.2.1. NLP hoạt động như thế nào?

Xử lý ngôn ngữ tự nhiên (NLP) kết hợp ngôn ngữ học điện toán, máy học và các mô hình học sâu để xử lý ngôn ngữ của con người.

- Ngôn ngữ học điện toán là lĩnh vực khoa học nghiên cứu và xây dựng các mô hình ngôn ngữ con người bằng những công cụ máy tính và phần mềm. Các nhà nghiên

cứ sử dụng những phương pháp ngôn ngữ học điện toán, chẳng hạn như phân tích cú pháp và ngữ nghĩa, để tạo ra các khung giúp máy móc hiểu được ngôn ngữ hội thoại của con người. Các công cụ như trình chuyển ngữ, trình tổng hợp văn bản thành giọng nói và phần mềm nhận dạng giọng nói đều dựa trên ngôn ngữ học điện toán.

- Máy học là một công nghệ đào tạo máy tính bằng dữ liệu mẫu để nâng cao hiệu quả của nó. Ngôn ngữ con người có một số đặc điểm như âm điệu, ẩn dụ, biến thể trong cấu trúc câu, cộng với các ngoại lệ về ngữ pháp và ngữ cảnh sử dụng mà con người phải mất nhiều năm để học. Các lập trình viên sử dụng những phương pháp máy học để dạy các ứng dụng NLP nhận biết và hiểu chính xác các đặc điểm này ngay từ đầu.

- Deep learning: Học sâu là một lĩnh vực cụ thể của máy học, tập trung vào việc dạy máy tính học và suy nghĩ như con người. Lĩnh vực này liên quan đến một mạng nơ-ron bao gồm các nút xử lý dữ liệu được cấu trúc để phỏng theo bộ não người. Với học sâu, các máy tính nhận biết, phân loại và tìm mối tương quan giữa các mẫu phức tạp trong dữ liệu đầu vào.

Các bước triển khai NLP: Thông thường, việc triển khai NLP bắt đầu bằng cách thu thập và chuẩn bị dữ liệu văn bản hoặc giọng nói phi cấu trúc từ các nguồn như kho dữ liệu đám mây, khảo sát, email hoặc các ứng dụng quy trình kinh doanh nội bộ.

- Xử lý trước: Phần mềm NLP sử dụng các kỹ thuật xử lý trước như token hóa, rút gọn từ, phục hồi nguyên thể từ và loại bỏ từ dừng để chuẩn bị dữ liệu cho các ứng dụng khác nhau. Dưới đây là mô tả về các kỹ thuật này:

- Token hóa chia nhỏ một câu thành các đơn vị từ hoặc cụm từ riêng lẻ.
- Rút gọn từ và phục hồi nguyên thể từ đơn giản hóa các từ về dạng gốc của chúng. Ví dụ, các quá trình này biến từ “starting” về thành “start”.
- Loại bỏ từ dừng đảm bảo rằng các từ không bổ sung thêm ý nghĩa đáng kể cho câu, chẳng hạn như “cho” và “với,” sẽ bị loại bỏ.

- Đào tạo: Các nhà nghiên cứu sử dụng dữ liệu được xử lý trước và máy học để đào tạo các mô hình NLP nhằm thực hiện các ứng dụng cụ thể dựa trên thông tin văn bản được cung cấp. Đào tạo các thuật toán NLP đòi hỏi cung cấp cho phần mềm các mẫu dữ liệu lớn để tăng độ chính xác của thuật toán.



- Triển khai và suy luận: Sau đó, các chuyên gia máy học sẽ triển khai mô hình hoặc tích hợp nó vào một môi trường sản xuất hiện có. Mô hình NLP nhận dữ liệu đầu vào và đưa ra dự đoán cho trường hợp sử dụng cụ thể mà mô hình được thiết kế. Có thể chạy ứng dụng NLP trên dữ liệu trực tiếp và thu được dữ liệu đầu ra theo yêu cầu.

2.2.2. Phương pháp xử lý ngôn ngữ tự nhiên

NLP có giám sát:

Các phương pháp NLP có giám sát đào tạo phần mềm với một tập dữ liệu đầu vào và đầu ra được gắn nhãn hoặc đã biết. Trước tiên, chương trình sẽ xử lý khối lượng lớn dữ liệu đã biết và học cách tạo ra dữ liệu đầu ra chính xác từ bất kỳ dữ liệu đầu vào chưa xác định nào. Ví dụ: các công ty đào tạo những công cụ NLP để phân loại tài liệu theo nhãn cụ thể.

NLP không giám sát:

NLP không giám sát sử dụng một mô hình ngôn ngữ thống kê để dự đoán mẫu xảy ra khi nó được cung cấp dữ liệu đầu vào không gắn nhãn. Ví dụ: tính năng tự động hoàn thành trong nhắn tin văn bản đề xuất các từ có liên quan tạo thành câu có nghĩa bằng cách theo dõi phản hồi của người dùng.

Hiểu ngôn ngữ tự nhiên:

Hiểu ngôn ngữ tự nhiên (NLU) là một nhánh con của NLP, tập trung vào việc phân tích ý nghĩa đằng sau các câu. NLU cho phép phần mềm tìm kiếm các câu đồng nghĩa khác nhau hoặc xử lý các từ có nhiều nghĩa khác nhau.

Tạo ngôn ngữ tự nhiên:

Tạo ngôn ngữ tự nhiên (NLG) tập trung vào việc tạo ra văn bản hội thoại giống như con người dựa trên các từ khóa hoặc chủ đề cụ thể. Ví dụ: một chatbot thông minh với khả năng NLG có thể trò chuyện với khách hàng theo những cách tương tự như nhân viên hỗ trợ khách hàng.

CHƯƠNG 3. ỨNG DỤNG RASA OPEN SOURCE XÂY DỰNG CHATBOT

3.1. Cài đặt và cấu hình Rasa Open Source

Phần này sẽ là các bước chi tiết cách cài đặt AI Rasa trên Windows 64bit như thế nào.

Bước 1:

Cài đặt Visual Studio 2019 sau đó cài Python 3.8.

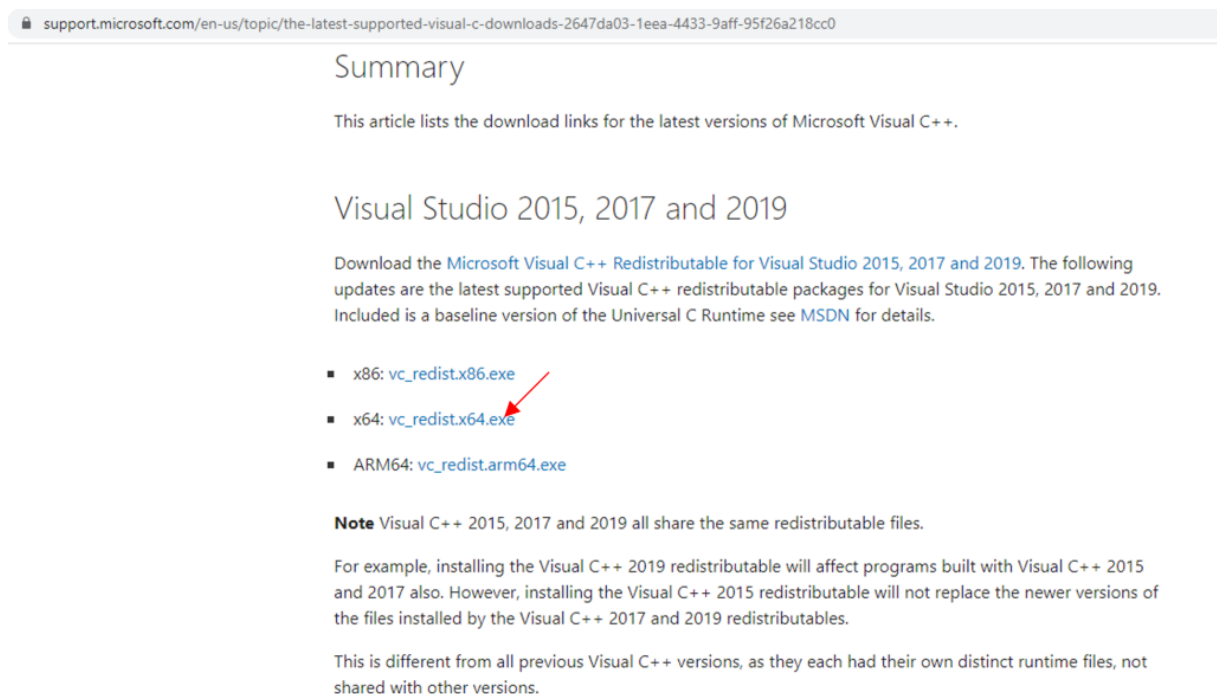
Bước 2:

Sau khi cài xong Visual Studio 2019 và Python 3.8. Ta tiến hành cài đặt thư viện hỗ trợ visual C++.

Link download thư viện hỗ trợ visual C++ ở đây:

<https://support.microsoft.com/en-us/topic/the-latest-supported-visual-c-downloads-2647da03-1eea-4433-9aff-95f26a218cc0>

khi vào link trên, bạn sẽ thấy giao diện hiển thị như dưới đây:



Hình 3.1. Giao diện trang cài đặt thư viện hỗ trợ Visual C++

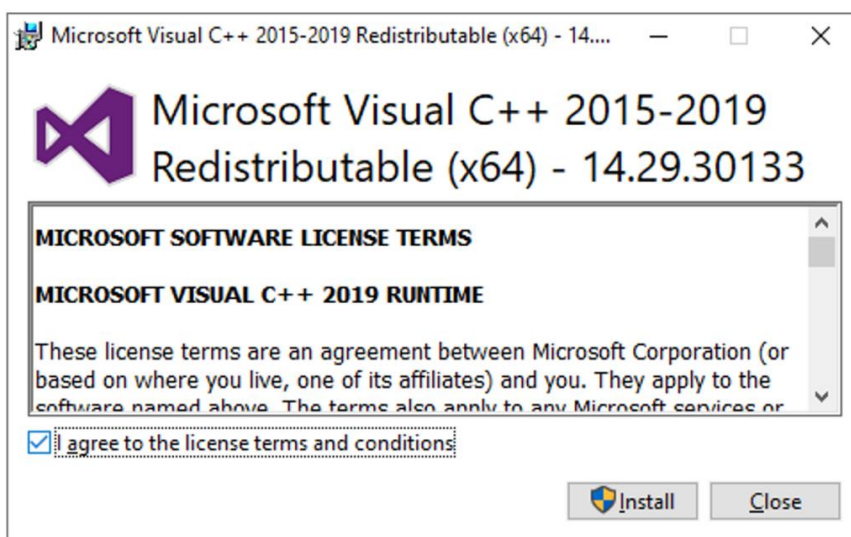
(nguồn: duythanhcse.wordpress.com)

Ta bấm vào link file **VC_redist.x64.exe** để tải, dung lượng khoảng 24.578 KB.

Data (D:) > RASA			
Name	Date modified	Type	Size
Help-Installing-RASA-Windows.docx	8/23/2021 11:33 AM	Microsoft Word D...	101 KB
VC_redist.x64.exe	8/23/2021 11:31 AM	Application	24,578 KB

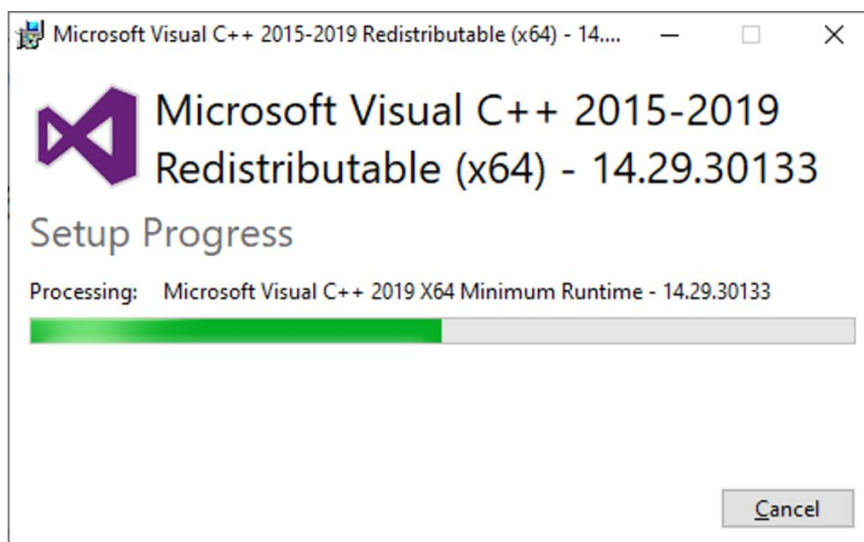
Hình 3.2. File cài đặt (nguồn: duythanhcse.wordpress.com)

Double Click vào VC_redist.x64.exe để cài đặt.



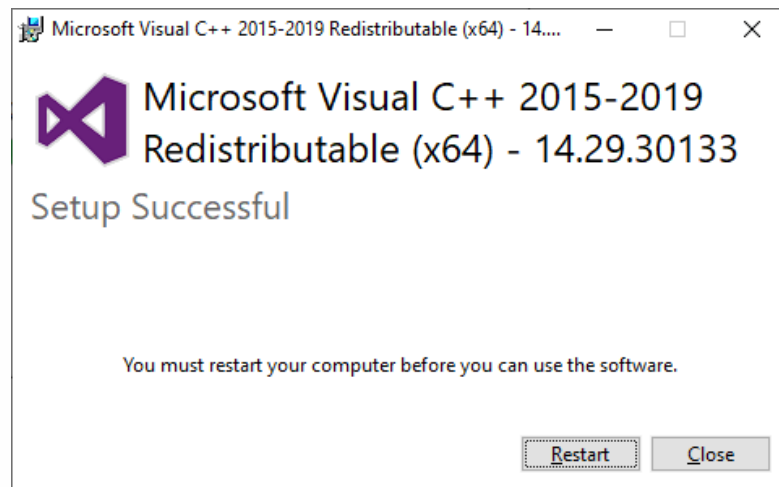
Hình 3.3. Quá trình cài đặt thư viện hỗ trợ 1 (nguồn: duythanhcse.wordpress.com)

Check “I agree to the license terms and conditions” và sau đó click install



Hình 3.4. Quá trình cài đặt thư viện hỗ trợ 2 (nguồn: duythanhcse.wordpress.com)

Ta chờ cho nó cài đặt hoàn tất, sau đó nhấn nút Restart để khởi động lại máy:



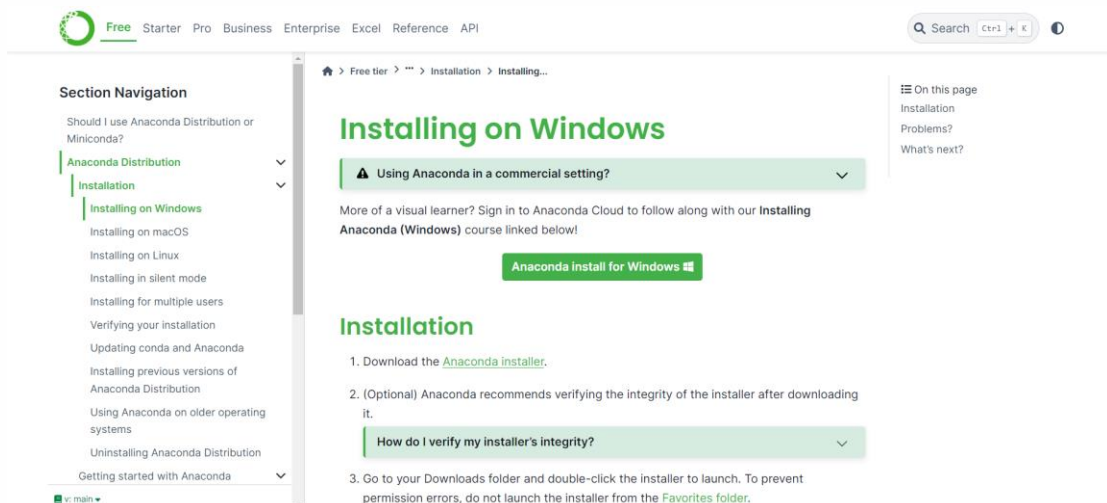
Hình 3.5. Quá trình cài đặt thư viện hỗ trợ 3 (nguồn: duythanhcse.wordpress.com)

Bước 3:

Sau khi khởi động máy xong, ta tiến hành cài Anaconda.

Ta vào link: <https://docs.anaconda.com/anaconda/install/windows/>

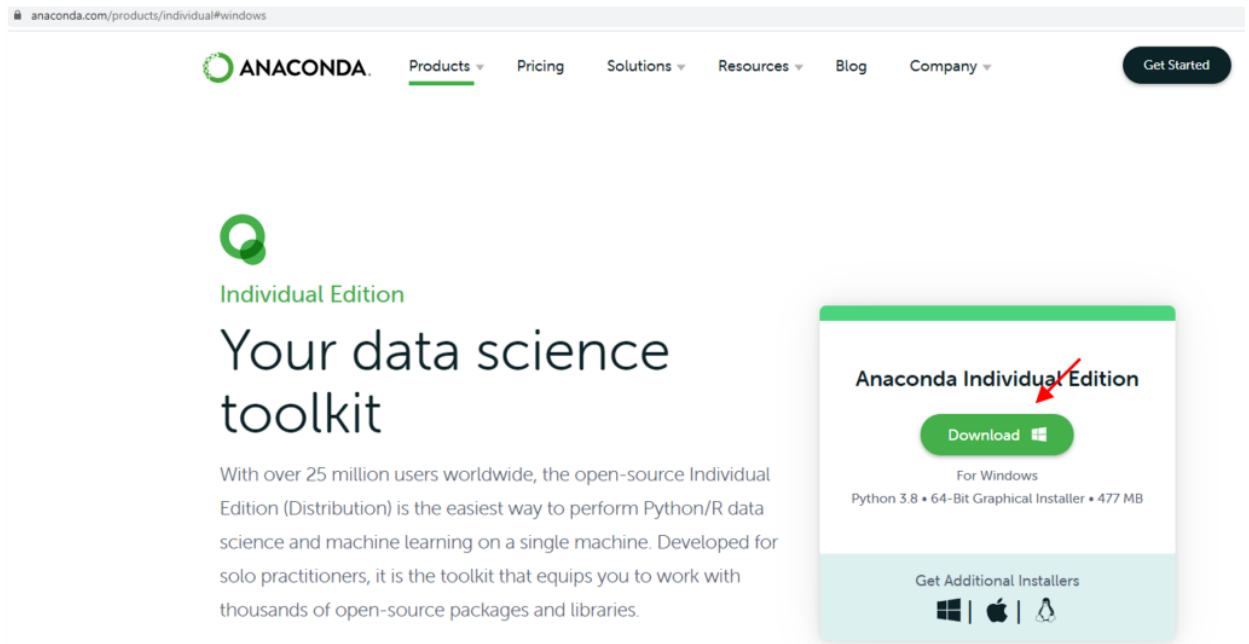
Màn hình dưới đây xuất hiện:



Hình 3.6. Cài đặt Anaconda (nguồn: duythanhcse.wordpress.com)

Ta nhấn vào link **Download the Anaconda installer** ở trên.

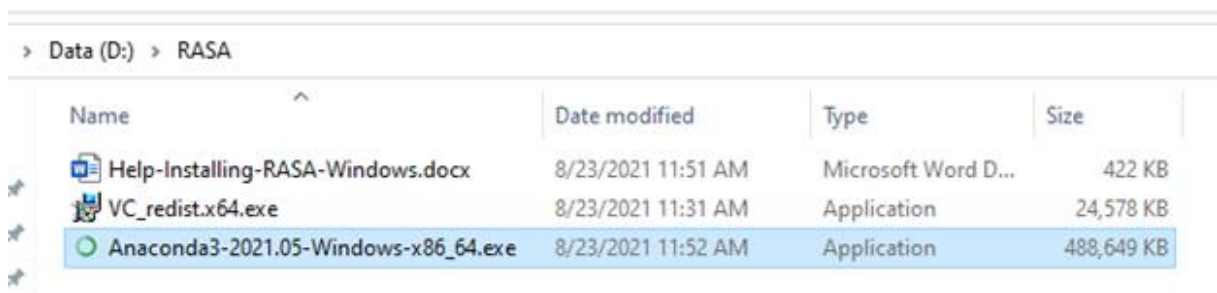
Sau đó nó sẽ hiển thị phiên bản để tải, bạn chọn Download:



Hình 3.7. Quá trình cài đặt Anaconda 1 (nguồn: duythanhcse.wordpress.com)

Chọn thư mục để lưu Anaconda installer

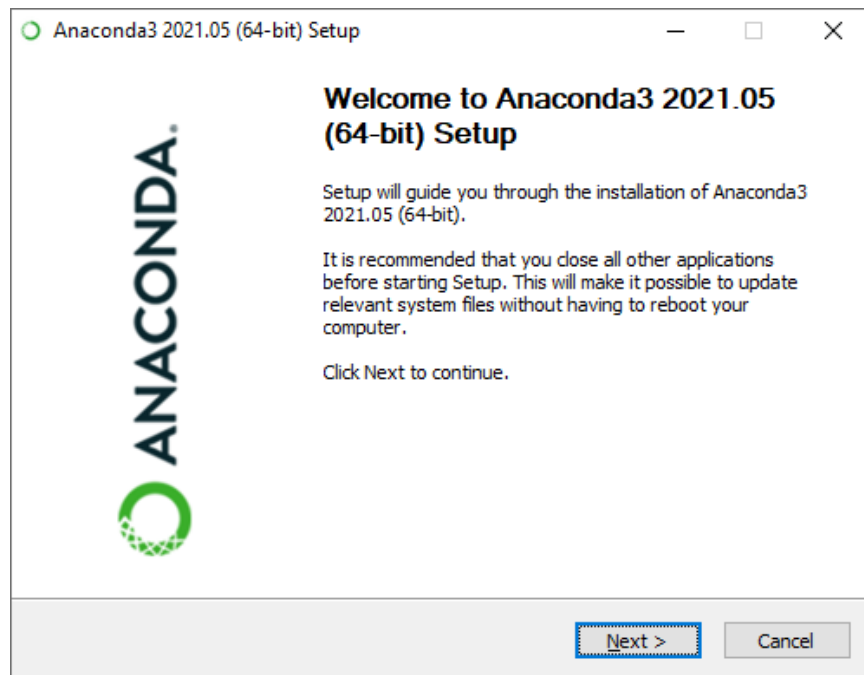
Click save và chờ nó tải....



Hình 3.8. Quá trình cài đặt Anaconda 2 (nguồn: duythanhcse.wordpress.com)

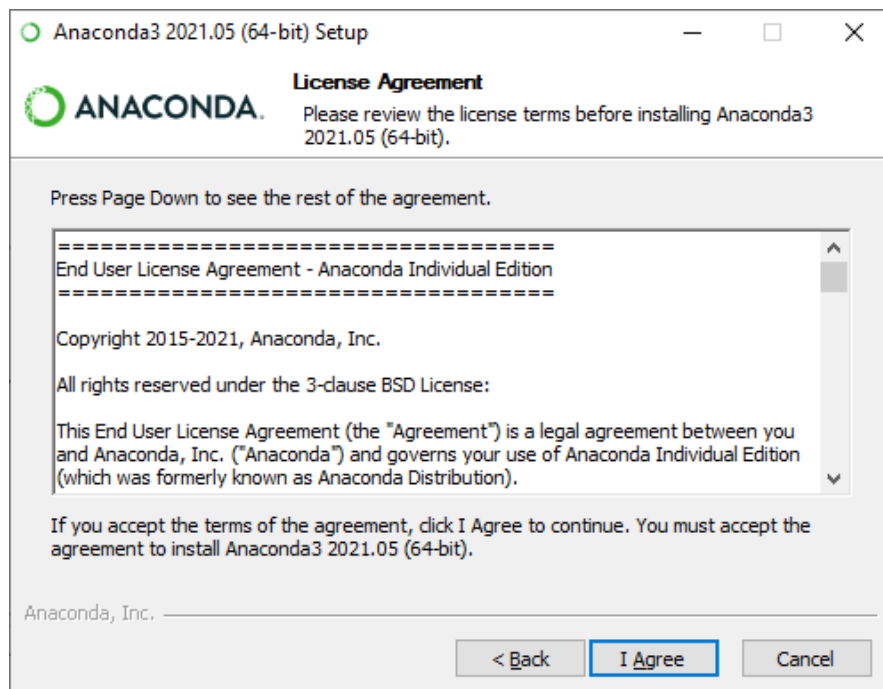
Ở trên là đã tải hoàn tất.

Giờ ta Click vào file Anaconda vừa tải về để install.



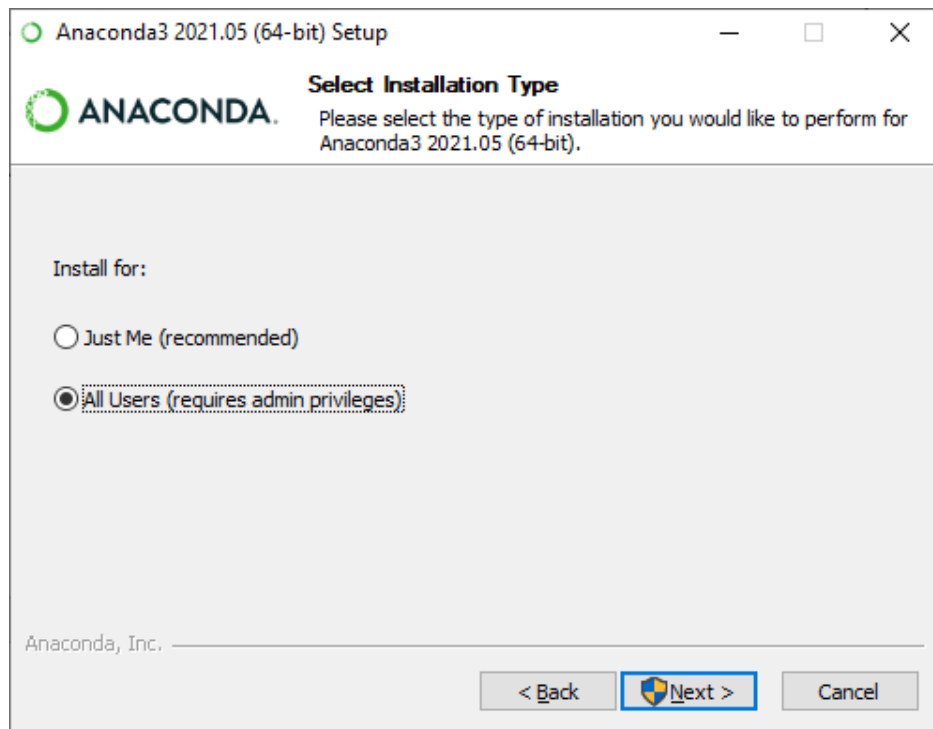
Hình 3.9. Quá trình cài đặt Anaconda 3 (nguồn: duythanhcse.wordpress.com)

Click Next để tiếp tục



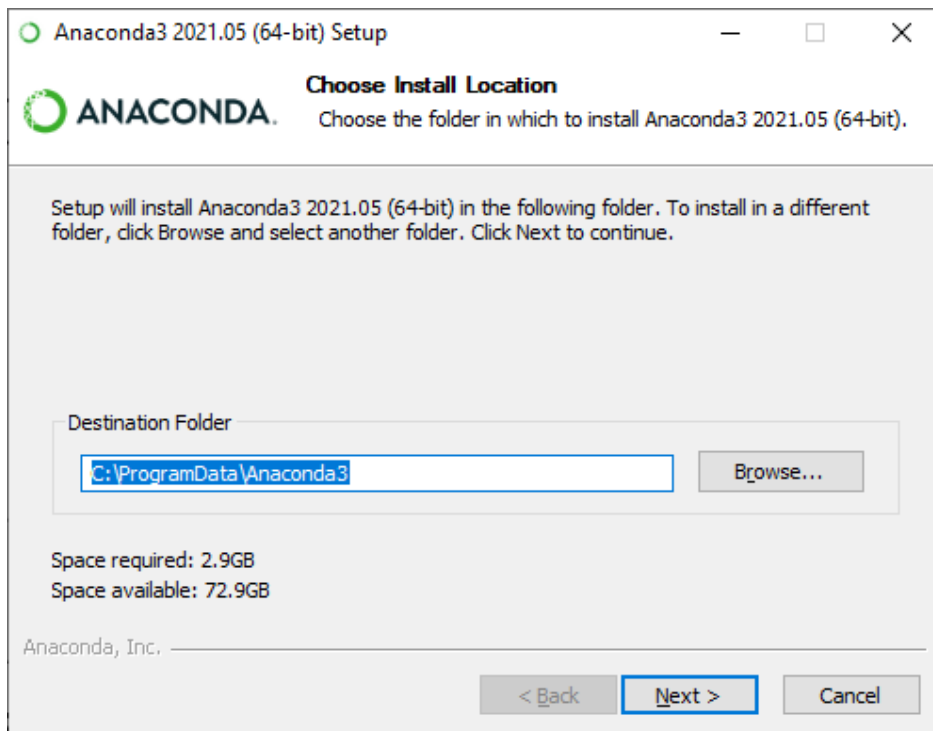
Hình 3.10. Quá trình cài đặt Anaconda 4 (nguồn: duythanhcse.wordpress.com)

Click I agree để đồng ý cài.



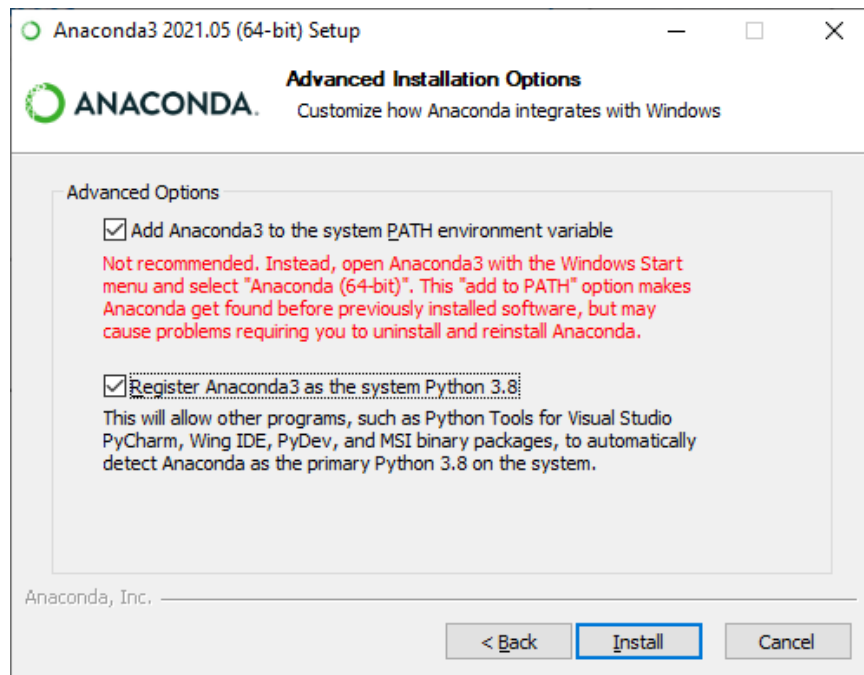
Hình 3.11. Quá trình cài đặt Anaconda 5 (nguồn: duythanhcse.wordpress.com)

Check All users và click Next



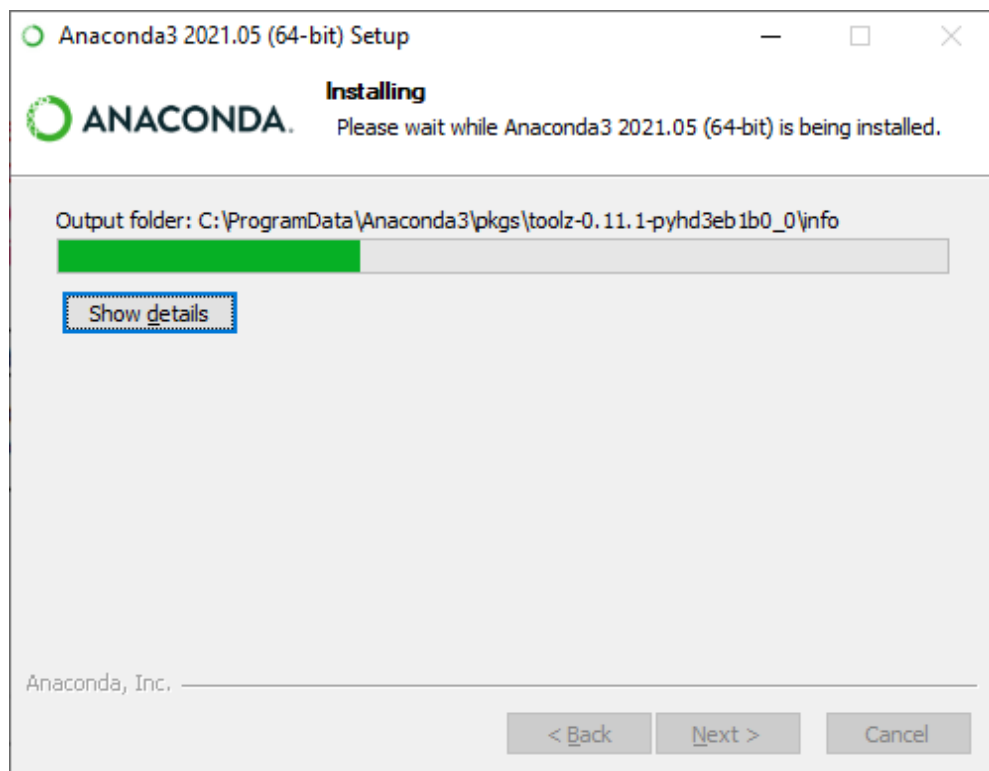
Hình 3.12. Quá trình cài đặt Anaconda 6 (nguồn: duythanhcse.wordpress.com)

Chọn nơi cài đặt (để mặc định như trên OK) rồi bấm Next



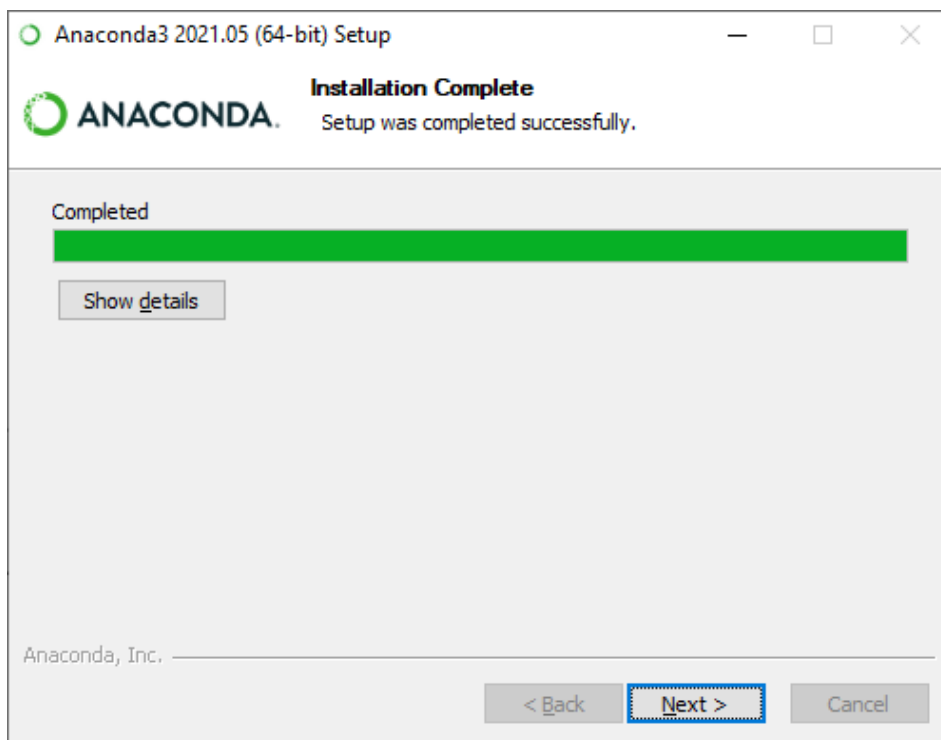
Hình 3.13. Quá trình cài đặt Anaconda 7 (nguồn: duythanhcse.wordpress.com)

Checked 2 lựa chọn ở trên rồi bấm Install.



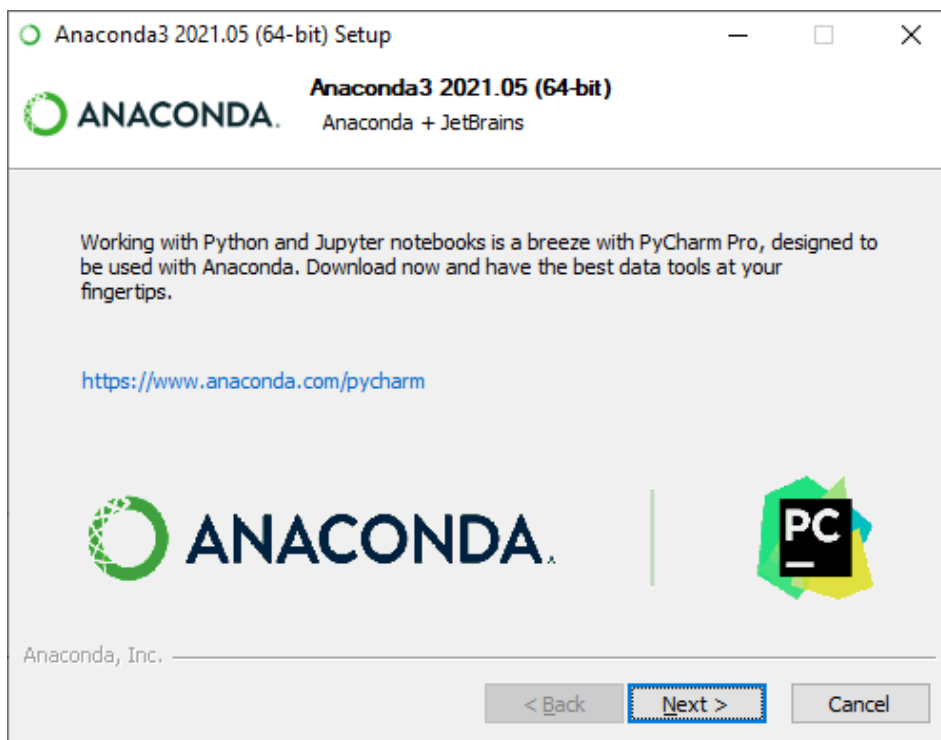
Hình 3.14. Quá trình cài đặt Anaconda 8 (nguồn: duythanhcse.wordpress.com)

Ta chờ cho nó cài đặt khi nào tới màn hình Completed dưới đây thì bấm Next:



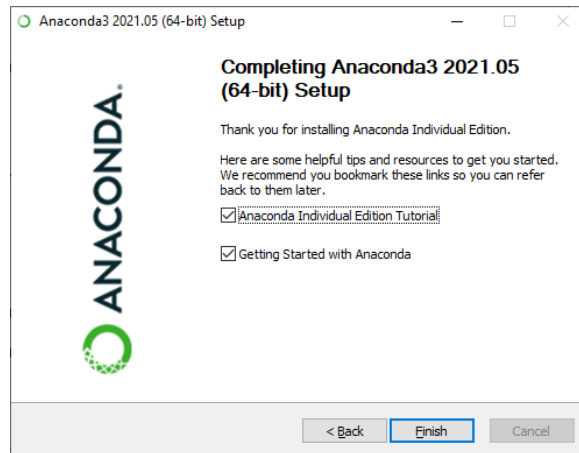
Hình 3.15. Quá trình cài đặt Anaconda 9 (nguồn: duythanhcse.wordpress.com)

Nhấn Next màn hình sau sẽ xuất hiện:



Hình 3.16. Quá trình cài đặt Anaconda 10 (nguồn: duythanhcse.wordpress.com)

Tiếp tục nhấn Next



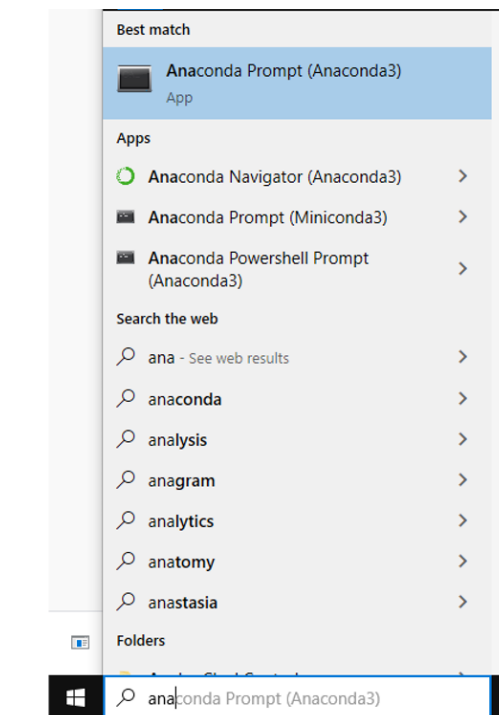
Hình 3.17. Quá trình cài đặt Anaconda 11 (nguồn: duythanhcse.wordpress.com)

Nhấn Finish để hoàn tất quá trình cài đặt Anaconda.

Bước 4:

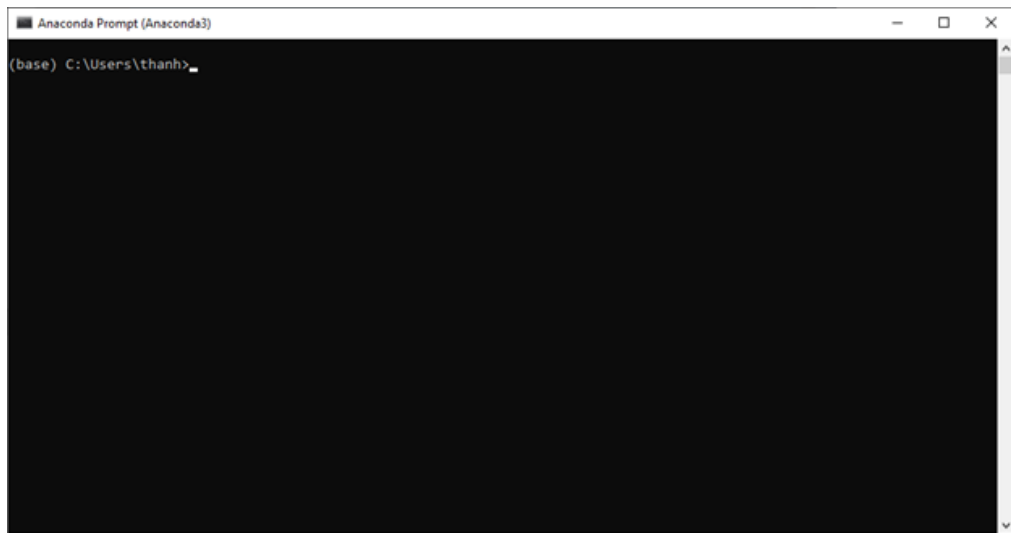
Cài đặt Rasa.

Ta mở màn hình Anaconda Prompt



Hình 3.18. Quá trình khởi tạo Rasa 1 (nguồn: duythanhcse.wordpress.com)

Trong màn hình start, tìm anaconda rồi chọn Anaconda Prompt, nhấn Enter, màn hình lệnh Anaconda sẽ xuất hiện như dưới đây:



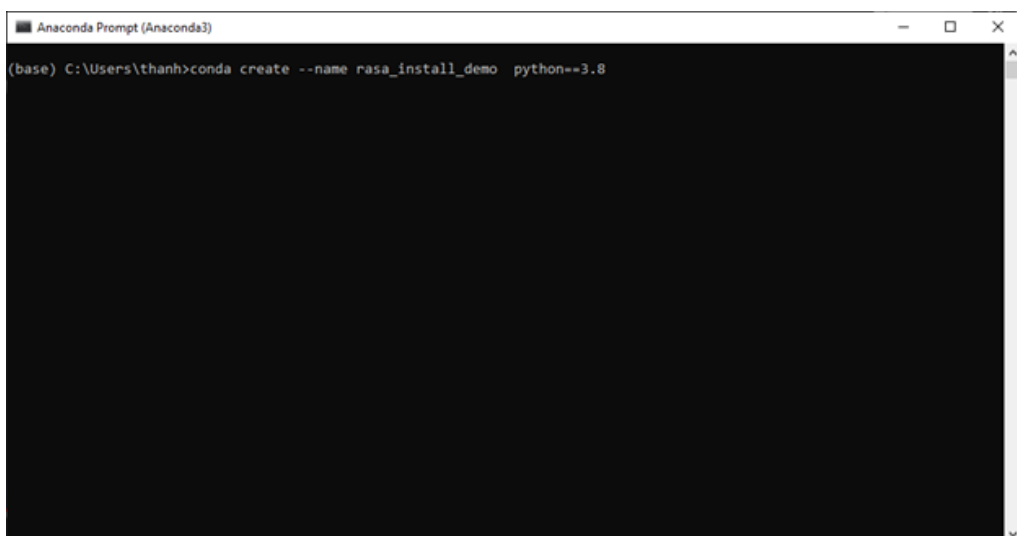
Hình 3.19. Quá trình khởi tạo Rasa 2 (nguồn: duythanhcse.wordpress.com)

Bước 4.1: Tạo rasa_install_demo (có thể đặt tên khác)

Gõ lệnh sau vào dấu nhắc lệnh của prompt:

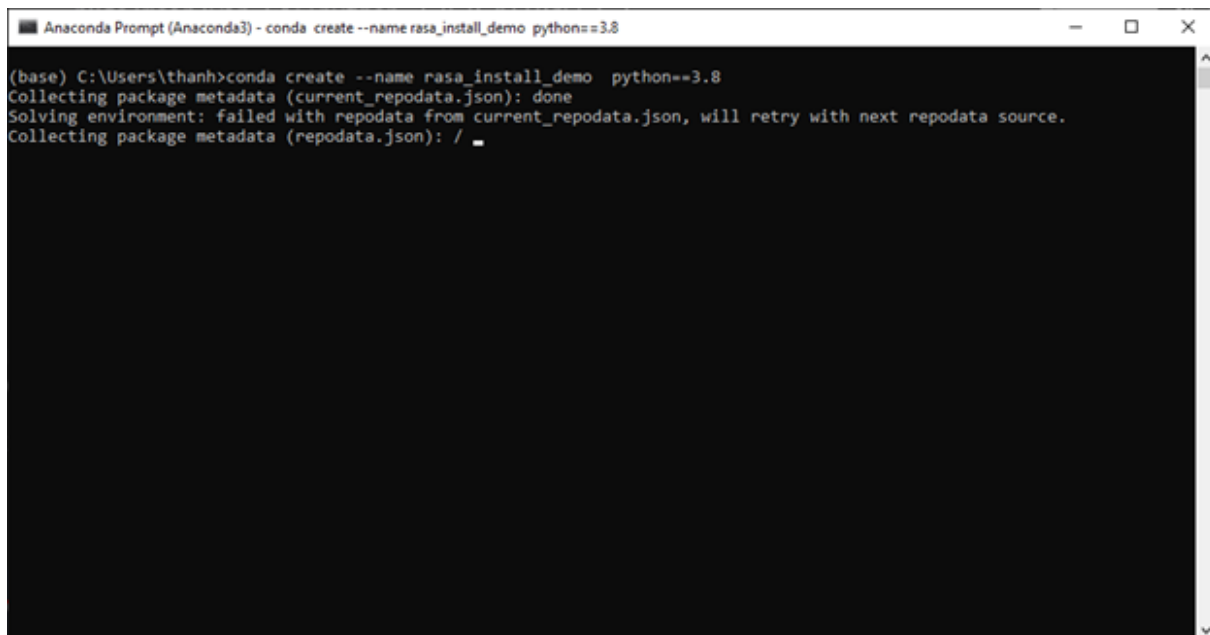
```
1 | conda create --name rasa_install_demo python==3.8
```

xem hình:



Hình 3.20. Quá trình khởi tạo Rasa 3 (nguồn: duythanhcse.wordpress.com)

Nhấn phím Enter để hệ thống thực hiện:



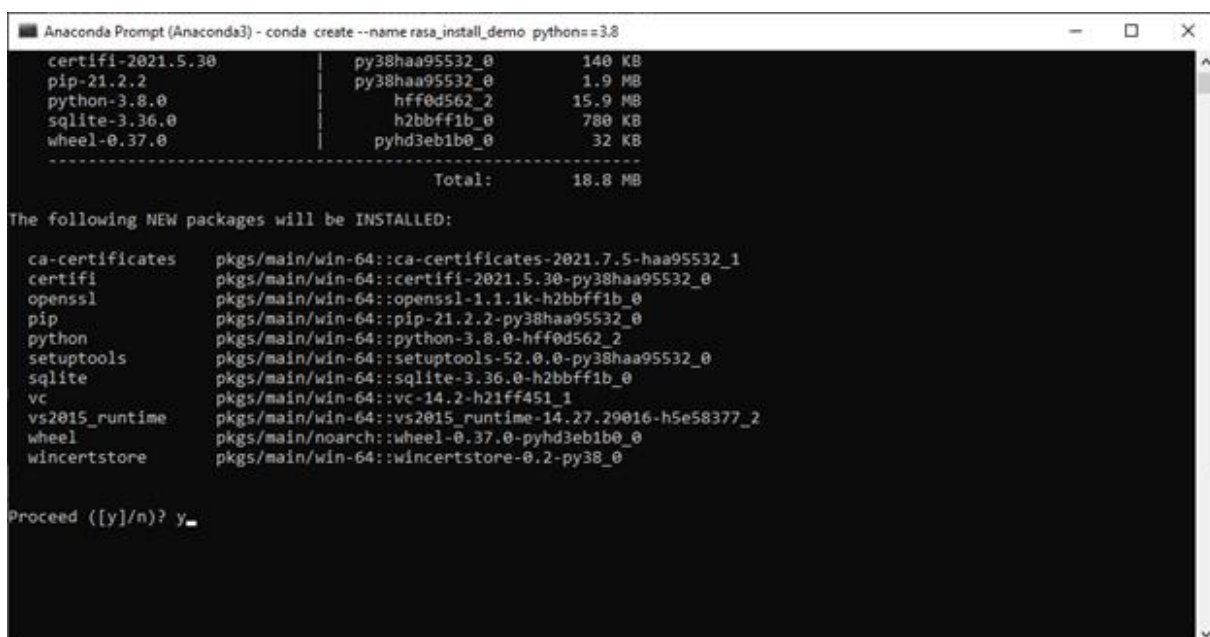
```

Anaconda Prompt (Anaconda3) - conda create --name rasa_install_demo python==3.8

(base) C:\Users\thanh>conda create --name rasa_install_demo python==3.8
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): /
  
```

Hình 3.21. Quá trình khởi tạo Rasa 4 (nguồn: duythanhcse.wordpress.com)

Chờ nó thực hiện tiếp..., sau đó nó hiện thị ra màn hình dưới đây:



```

Anaconda Prompt (Anaconda3) - conda create --name rasa_install_demo python==3.8

certifi-2021.5.30      | py38haa95532_0      | 140 KB
pip-21.2.2            | py38haa95532_0      | 1.9 MB
python-3.8.0          | hff0d562_2          | 15.9 MB
sqlite-3.36.0         | h2bbff1b_0          | 780 KB
wheel-0.37.0          | pyhd3eb1b0_0        | 32 KB
-----
Total:                | 18.8 MB

The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2021.7.5-haa95532_1
certifi              pkgs/main/win-64::certifi-2021.5.30-py38haa95532_0
openssl              pkgs/main/win-64::openssl-1.1.1k-h2bbff1b_0
pip                  pkgs/main/win-64::pip-21.2.2-py38haa95532_0
python               pkgs/main/win-64::python-3.8.0-hff0d562_2
setuptools            pkgs/main/win-64::setuptools-52.0.0-py38haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.36.0-h2bbff1b_0
vc                   pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                pkgs/main/noarch::wheel-0.37.0-pyhd3eb1b0_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py38_0

Proceed ([y]/n)? y_
  
```

Hình 3.22. Quá trình khởi tạo Rasa 5 (nguồn: duythanhcse.wordpress.com)

gõ y và nhấn enter, chờ nó cài đặt tiếp:

```

Anaconda Prompt (Anaconda3) - conda create --name rasa_install_demo python==3.8
pip                pkgs/main/win-64::pip-21.2.2-py38haa95532_0
python             pkgs/main/win-64::python-3.8.0-hff0d562_2
setuptools         pkgs/main/win-64::setuptools-52.0.0-py38haa95532_0
sqlite             pkgs/main/win-64::sqlite-3.36.0-h2bbff1b_0
vc                 pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel              pkgs/main/noarch::wheel-0.37.0-pyhd3eb1b0_0
wincertstore       pkgs/main/win-64::wincertstore-0.2-py38_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
sqlite-3.36.0           | 780 KB | ##### | 100%
certifi-2021.5.30      | 140 KB | ##### | 100%
wheel-0.37.0           | 32 KB  | ##### | 100%
python-3.8.0           | 15.9 MB | ##### | 98%

```

Hình 3.23. Quá trình khởi tạo Rasa 6 (nguồn: duythanhcse.wordpress.com)

Tiếp tục chờ cho tới khi nó ra màn hình sau:

```

Anaconda Prompt (Anaconda3)
wincertstore       pkgs/main/win-64::wincertstore-0.2-py38_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
sqlite-3.36.0           | 780 KB | ##### | 100%
certifi-2021.5.30      | 140 KB | ##### | 100%
wheel-0.37.0           | 32 KB  | ##### | 100%
python-3.8.0           | 15.9 MB | ##### | 100%
ca-certificates-2021   | 113 KB | ##### | 100%
pip-21.2.2             | 1.9 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate rasa_install_demo
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
(base) C:\Users\thanh>

```

Hình 3.24. Quá trình khởi tạo Rasa 7 (nguồn: duythanhcse.wordpress.com)

Như vậy bước tạo `rasa_install_demo` đã hoàn tất.

Bước 4.2: Activate rasa_install_demo

Dùng lệnh sau:

1 | `conda activate rasa_install_demo`

Xem hình:

```

Anaconda Prompt (Anaconda3)
Proceed ([y]/n)? y

Downloading and Extracting Packages
sqlite-3.36.0      | 780 KB | ##### | 100%
certifi-2021.5.30 | 140 KB | ##### | 100%
wheel-0.37.0      | 32 KB  | ##### | 100%
python-3.8.0      | 15.9 MB | ##### | 100%
ca-certificates-2021 | 113 KB | ##### | 100%
pip-21.2.2        | 1.9 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate rasa_install_demo
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

(base) C:\Users\thanh>conda activate rasa_install_demo_

```

Hình 3.25. Quá trình kích hoạt Rasa 1 (nguồn: duythanhcse.wordpress.com)

Gõ lệnh xong thì nhấn enter để activate

```

Anaconda Prompt (Anaconda3)
wheel-0.37.0      | 32 KB  | ##### | 100%
python-3.8.0      | 15.9 MB | ##### | 100%
ca-certificates-2021 | 113 KB | ##### | 100%
pip-21.2.2        | 1.9 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate rasa_install_demo
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

(base) C:\Users\thanh>conda activate rasa_install_demo
(rasa_install_demo) C:\Users\thanh>

```

Hình 3.26. Quá trình kích hoạt Rasa 2 (nguồn: duythanhcse.wordpress.com)

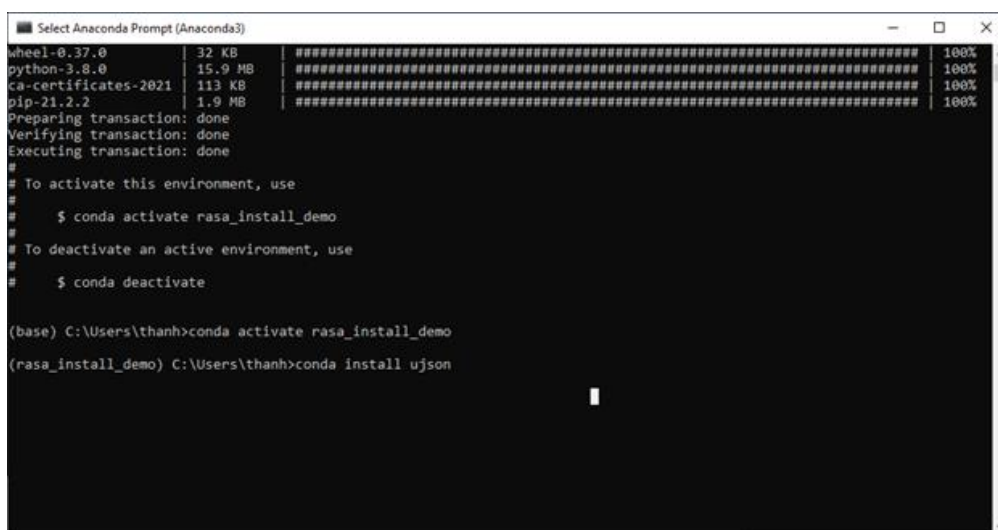
Như vậy đã activate thành công.

Bước 4.3: Cài đặt ujson

gõ lệnh sau:

1 | `conda install ujson`

Xem hình:

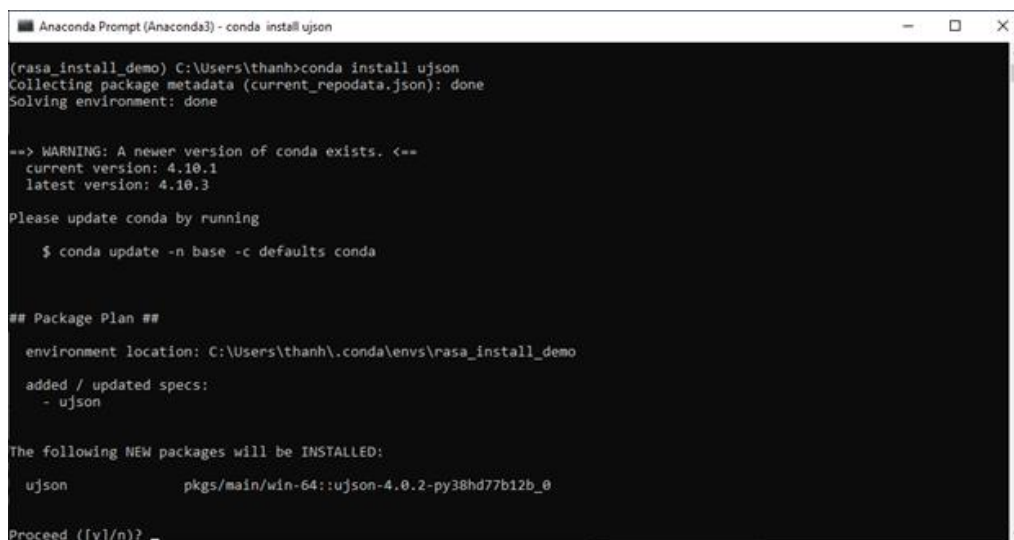


```

Select Anaconda Prompt (Anaconda3)
wheel-0.37.0      | 32 KB | 100%
python-3.8.0      | 15.9 MB | 100%
ca-certificates-2021 | 113 KB | 100%
pip-21.2.2       | 1.9 MB | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate rasa_install_demo
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) C:\Users\thanh>conda activate rasa_install_demo
(rasa_install_demo) C:\Users\thanh>conda install ujson
  
```

Hình 3.27. Quá trình cài đặt ujson 1 (nguồn: duythanhcse.wordpress.com)

Gõ lệnh xong thì nhấn enter để cài đặt:



```

Anaconda Prompt (Anaconda3) - conda install ujson
(rasa_install_demo) C:\Users\thanh>conda install ujson
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\thanh\.conda\envs\rasa_install_demo
  added / updated specs:
    - ujson

The following NEW packages will be INSTALLED:

  ujson                pkgs/main/win-64::ujson-4.0.2-py38hd77b12b_0

Proceed ([y]/n)?
  
```

Hình 3.28. Quá trình cài đặt ujson 2 (nguồn: duythanhcse.wordpress.com)

Ta nhập y rồi gõ enter:



```
Anaconda Prompt (Anaconda3) - conda install ujson
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\thanh\.conda\envs\rasa_install_demo

  added / updated specs:
    - ujson

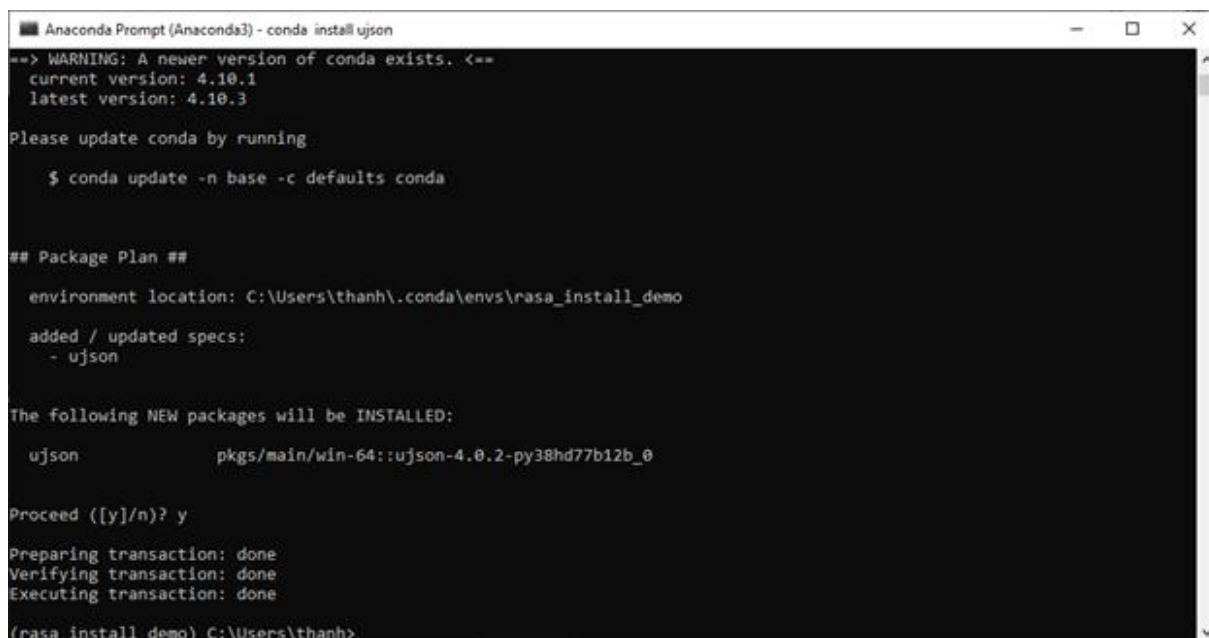
The following NEW packages will be INSTALLED:

  ujson                pkgs/main/win-64::ujson-4.0.2-py38hd77b12b_0

Proceed ([y]/n)? y
```

Hình 3.29. Quá trình cài đặt ujson 3 (nguồn: duythanhcse.wordpress.com)

Nhấn enter để cài đặt, chờ nó cài hoàn tất, dưới đây là màn hình sau khi cài ujson thành công:



```
Anaconda Prompt (Anaconda3) - conda install ujson
==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\thanh\.conda\envs\rasa_install_demo

  added / updated specs:
    - ujson

The following NEW packages will be INSTALLED:

  ujson                pkgs/main/win-64::ujson-4.0.2-py38hd77b12b_0

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(rasa_install_demo) C:\Users\thanh>
```

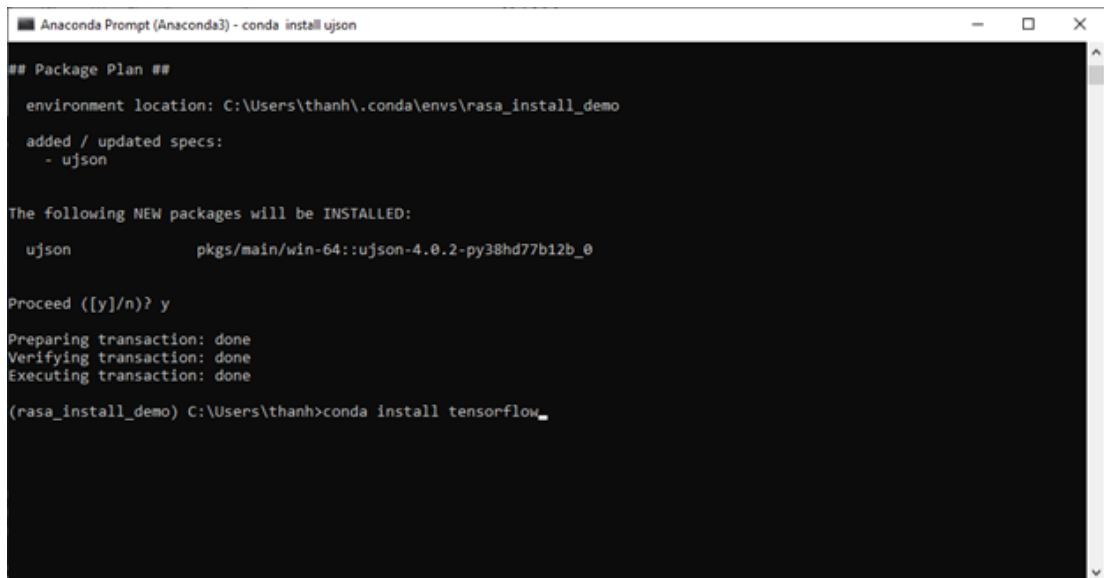
Hình 3.30. Quá trình cài đặt ujson 4 (nguồn: duythanhcse.wordpress.com)

Bước 4.4: Cài đặt tensorflow

Dùng lệnh sau:

1 | `conda install tensorflow`

Xem hình:



```

Anaconda Prompt (Anaconda3) - conda install ujson

## Package Plan ##

environment location: C:\Users\thanh\.conda\envs\rasa_install_demo

added / updated specs:
- ujson

The following NEW packages will be INSTALLED:

ujson                pkgs/main/win-64::ujson-4.0.2-py38hd77b12b_0

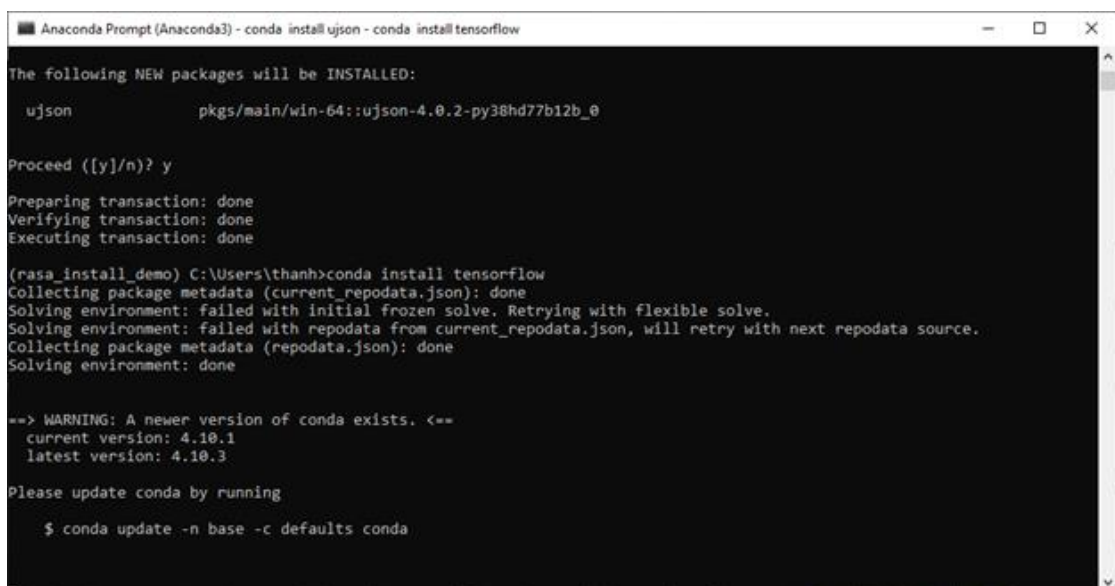
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(rasa_install_demo) C:\Users\thanh>conda install tensorflow_

```

Hình 3.31. Quá trình cài đặt tensorflow 1 (nguồn: duythanhcse.wordpress.com)

Nhấn Enter và chờ nó cài đặt:



```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow

The following NEW packages will be INSTALLED:

ujson                pkgs/main/win-64::ujson-4.0.2-py38hd77b12b_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(rasa_install_demo) C:\Users\thanh>conda install tensorflow
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

Please update conda by running

  $ conda update -n base -c defaults conda

```

Hình 3.32. Quá trình cài đặt tensorflow 2 (nguồn: duythanhcse.wordpress.com)

Thấy màn hình trên xuất hiện thì nhập **y** và nhấn enter.

```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow

tensorflow-base pkgs/main/win-64::tensorflow-base-2.3.0-eigen_py38h75a453f_0
tensorflow-estima pkgs/main/noarch::tensorflow-estimator-2.5.0-pyh7b7c402_0
termcolor pkgs/main/win-64::termcolor-1.1.0-py38haa95532_1
typing-extensions pkgs/main/noarch::typing-extensions-3.10.0.0-hd3eb1b0_0
typing_extensions pkgs/main/noarch::typing_extensions-3.10.0.0-pyh06a4308_0
urllib3 pkgs/main/noarch::urllib3-1.26.6-pyhd3eb1b0_1
werkzeug pkgs/main/noarch::werkzeug-1.0.1-pyhd3eb1b0_0
win_inet_pton pkgs/main/win-64::win_inet_pton-1.1.0-py38haa95532_0
wrapit pkgs/main/win-64::wrapit-1.12.1-py38he774522_1
yarl pkgs/main/win-64::yarl-1.6.3-py38h2bbff1b_0
zipp pkgs/main/noarch::zipp-3.5.0-pyhd3eb1b0_0
zlib pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

Proceed ([y]/n)? y

Downloading and Extracting Packages
keras-applications-1 29 KB | 100%
zipp-3.5.0 13 KB | 100%
multidict-5.1.0 61 KB | 100%
tensorflow-2.3.0 6 KB | 100%
cachetools-4.2.2 13 KB | 100%
_tflow_select-2.3.0 4 KB | 100%
google-pasta-0.2.0 46 KB | 100%
oauthlib-3.1.1 90 KB | 100%
astunparse-1.6.3 17 KB | 100%
chardet-3.0.4 194 KB | 100%
coverage-5.5 272 KB | 100%
cython-0.29.24 1.8 MB | 46%

```

Hình 3.33. Quá trình cài đặt tensorflow 3 (nguồn: duythanhcse.wordpress.com)

```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow

tensorflow-base 8.8 MB | 100%
markdown-3.3.4 144 KB | 100%
pyasn1-0.4.8 57 KB | 100%
rsa-4.7.2 28 KB | 100%
astor-0.8.1 47 KB | 100%
intel-openmp-2021.3 2.0 MB | 100%
blinker-1.4 23 KB | 100%
async-timeout-3.0.1 14 KB | 100%
attrs-21.2.0 46 KB | 100%
mkl-service-2.4.0 51 KB | 100%
grpcio-1.36.1 1.7 MB | 100%
pyasn1-modules-0.2.8 72 KB | 100%
opt_einsum-3.3.0 57 KB | 100%
keras-preprocessing 35 KB | 100%
tensorflow-base-2.3 49.5 MB | 100%
absl-py-0.13.0 175 KB | 100%
mkl_random-1.2.2 225 KB | 100%
click-8.0.1 79 KB | 100%
typing-extensions-3 8 KB | 100%
protobuf-3.17.2 257 KB | 100%
termcolor-1.1.0 9 KB | 100%
tensorflow-plugin-w 630 KB | 100%
six-1.16.0 18 KB | 100%
numpy-1.20.3 23 KB | 100%
google-auth-oauthlib 18 KB | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(rasa_install_demo) C:\Users\thanh>

```

Hình 3.34. Quá trình cài đặt tensorflow 4 (nguồn: duythanhcse.wordpress.com)

Màn hình trên xuất hiện tức là đã hoàn tất bước cài đặt Tensorflow.

Bước 4.5: Đây là bước thực sự cài đặt rasa

gõ lệnh sau:

1 | `pip install rasa`

Xem hình:

```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow
rsa-4.7.2 28 KB 100%
astor-0.8.1 47 KB 100%
intel-openmp-2021.3 2.0 MB 100%
blinker-1.4 23 KB 100%
async-timeout-3.0.1 14 KB 100%
attrs-21.2.0 46 KB 100%
mkl-service-2.4.0 51 KB 100%
grpcio-1.36.1 1.7 MB 100%
pyasn1-modules-0.2.8 72 KB 100%
opt_einsum-3.3.0 57 KB 100%
keras-preprocessing-2.3.0 35 KB 100%
tensorflow-base-2.3.0 49.5 MB 100%
absl-py-0.13.0 175 KB 100%
mkl_random-1.2.2 225 KB 100%
click-8.0.1 79 KB 100%
typing-extensions-3.7.4 8 KB 100%
protobuf-3.17.2 257 KB 100%
termcolor-1.1.0 9 KB 100%
tensorboard-plugin-wit-1.8.0 630 KB 100%
six-1.16.0 18 KB 100%
numpy-1.20.3 23 KB 100%
google-auth-oauthlib-0.4.6 18 KB 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(rasa_install_demo) C:\Users\thanh>pip install rasa

```

Hình 3.35. Quá trình cài đặt rasa 1 (nguồn: duythanhcse.wordpress.com)

Sau khi gõ lệnh, nhấn Enter để tiến hành cài rasa.

```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow - pip install rasa
grpcio-1.36.1 1.7 MB 100%
pyasn1-modules-0.2.8 72 KB 100%
opt_einsum-3.3.0 57 KB 100%
keras-preprocessing-2.3.0 35 KB 100%
tensorflow-base-2.3.0 49.5 MB 100%
absl-py-0.13.0 175 KB 100%
mkl_random-1.2.2 225 KB 100%
click-8.0.1 79 KB 100%
typing-extensions-3.7.4 8 KB 100%
protobuf-3.17.2 257 KB 100%
termcolor-1.1.0 9 KB 100%
tensorboard-plugin-wit-1.8.0 630 KB 100%
six-1.16.0 18 KB 100%
numpy-1.20.3 23 KB 100%
google-auth-oauthlib-0.4.6 18 KB 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(rasa_install_demo) C:\Users\thanh>pip install rasa
Collecting rasa
  Downloading rasa-2.8.3-py3-none-any.whl (766 kB)
    | 766 kB 1.6 MB/s
Collecting mattermostwrapper<2.3,>=2.2
  Using cached mattermostwrapper-2.2.tar.gz (2.5 kB)
Collecting oauth2client==4.1.3
  Using cached oauth2client-4.1.3-py2.py3-none-any.whl (98 kB)
Collecting scikit-learn<0.25,>=0.22
  Downloading scikit-learn-0.24.2-cp38-cp38-win_amd64.whl (6.9 MB)
    | 501 kB 6.8 MB/s eta 0:00:01

```

Hình 3.36. Quá trình cài đặt rasa 2 (nguồn: duythanhcse.wordpress.com)


```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow
Attempting uninstall: scipy
Found existing installation: scipy 1.6.2
Uninstalling scipy-1.6.2:
Successfully uninstalled scipy-1.6.2
Attempting uninstall: gast
Found existing installation: gast 0.4.0
Uninstalling gast-0.4.0:
Successfully uninstalled gast-0.4.0
Successfully installed SQLAlchemy-1.4.23 aio-pika-6.8.0 aiofiles-0.7.0 aiormq-3.3.1 apscheduler-3.7.0 async-generator-1.10.0 bidict-0.21.2 boto3-1.18.26 botocore-1.21.26 cloudpickle-1.6.0 colorama-0.4.4 colorclass-2.2.0 coloredlogs-15.0.1 colrhash-1.0.3 cycler-0.10.0 decorator-4.4.2 dm-tree-0.1.6 dnspython-1.16.0 docopt-0.6.2 fbmessage-6.0.0 future-0.18.2 gast-0.3.3 greenlet-1.1.1 h11-0.9.0 httpcore-0.11.1 httplib2-0.19.1 httptools-0.3.0 httpx-0.15.4 humanfriendly-9.2 jmespath-0.10.0 joblib-1.0.1 jsonpickle-2.0.0 jsonschema-3.2.0 kafka-python-2.0.2 kiwisolver-1.3.1 matplotlib-3.3.4 mattermost-wrapper-2.2 networkx-2.5.1 numpy-1.18.5 oauth2client-4.1.3 packaging-20.9 pamqp-2.3.0 pillow-8.3.1 prompt-toolkit-2.0.10 psycopg2-binary-2.9.1 pyTelegramBotAPI-3.8.3 pydot-1.4.2 pykwalify-1.8.0 pymongo-3.10.1 pyparsing-2.4.7 pyrsistent-0.18.0 python-crfsuite-0.9.7 python-dateutil-2.8.2 python-engineio-4.2.1 python-engineio-client-4.2.1 python-socketio-5.4.0 pytz-2021.1 questionnaire-1.9.0 rasa-2.8.3 rasa-sdk-2.8.1 redis-3.5.3 regex-2021.7.6 requests-toolbelt-0.9.1 rfc3986-1.5.0 rocketchat-API-1.16.0 ruamel-yaml-0.16.13 ruamel-yaml-clib-0.2.6 s3transfer-0.5.0 sanic-20.12.3 sanic-cors-0.10.0.post3 sanic-jwt-1.7.0 sanic-plugin-framework-0.9.5 scikit-learn-0.24.2 scrapy-1.4.1 sentry-sdk-1.2.0 sklearn-crfsuite-0.3.6 slackclient-2.9.3 sniffio-1.2.0 tabulate-0.8.9 tensorflow-addons-0.13.0 tensorflow-estimator-2.3.0 tensorflow-hub-0.12.0 tensorflow-probability-0.13.0 terminaltables-3.1.0 threadpoolctl-2.2.0 tqdm-4.62.1 twilio-6.50.1 typeguard-2.12.1 tzlocal-2.1 wcwidth-0.2.5 webexteamssdk-1.6 websockets-8.1

(rasa_install_demo) C:\Users\thanh>

```

Hình 3.37. Quá trình cài đặt rasa 3 (nguồn: duythanhcse.wordpress.com)

Màn hình trên xuất hiện tức là đã hoàn tất bước cài đặt **rasa**.

Bước 4.6: Double check để đảm bảo rằng rasa đã cài hoàn tất và không bị lỗi.

dùng lệnh sau:

1 | `rasa -h`

Xem hình:

```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow
Attempting uninstall: scipy
Found existing installation: scipy 1.6.2
Uninstalling scipy-1.6.2:
Successfully uninstalled scipy-1.6.2
Attempting uninstall: gast
Found existing installation: gast 0.4.0
Uninstalling gast-0.4.0:
Successfully uninstalled gast-0.4.0
Successfully installed SQLAlchemy-1.4.23 aio-pika-6.8.0 aiofiles-0.7.0 aiormq-3.3.1 apscheduler-3.7.0 async-generator-1.10.0 bidict-0.21.2 boto3-1.18.26 botocore-1.21.26 cloudpickle-1.6.0 colorama-0.4.4 colorclass-2.2.0 coloredlogs-15.0.1 colrhash-1.0.3 cycler-0.10.0 decorator-4.4.2 dm-tree-0.1.6 dnspython-1.16.0 docopt-0.6.2 fbmessage-6.0.0 future-0.18.2 gast-0.3.3 greenlet-1.1.1 h11-0.9.0 httpcore-0.11.1 httplib2-0.19.1 httptools-0.3.0 httpx-0.15.4 humanfriendly-9.2 jmespath-0.10.0 joblib-1.0.1 jsonpickle-2.0.0 jsonschema-3.2.0 kafka-python-2.0.2 kiwisolver-1.3.1 matplotlib-3.3.4 mattermost-wrapper-2.2 networkx-2.5.1 numpy-1.18.5 oauth2client-4.1.3 packaging-20.9 pamqp-2.3.0 pillow-8.3.1 prompt-toolkit-2.0.10 psycopg2-binary-2.9.1 pyTelegramBotAPI-3.8.3 pydot-1.4.2 pykwalify-1.8.0 pymongo-3.10.1 pyparsing-2.4.7 pyrsistent-0.18.0 python-crfsuite-0.9.7 python-dateutil-2.8.2 python-engineio-4.2.1 python-engineio-client-4.2.1 python-socketio-5.4.0 pytz-2021.1 questionnaire-1.9.0 rasa-2.8.3 rasa-sdk-2.8.1 redis-3.5.3 regex-2021.7.6 requests-toolbelt-0.9.1 rfc3986-1.5.0 rocketchat-API-1.16.0 ruamel-yaml-0.16.13 ruamel-yaml-clib-0.2.6 s3transfer-0.5.0 sanic-20.12.3 sanic-cors-0.10.0.post3 sanic-jwt-1.7.0 sanic-plugin-framework-0.9.5 scikit-learn-0.24.2 scrapy-1.4.1 sentry-sdk-1.2.0 sklearn-crfsuite-0.3.6 slackclient-2.9.3 sniffio-1.2.0 tabulate-0.8.9 tensorflow-addons-0.13.0 tensorflow-estimator-2.3.0 tensorflow-hub-0.12.0 tensorflow-probability-0.13.0 terminaltables-3.1.0 threadpoolctl-2.2.0 tqdm-4.62.1 twilio-6.50.1 typeguard-2.12.1 tzlocal-2.1 wcwidth-0.2.5 webexteamssdk-1.6 websockets-8.1

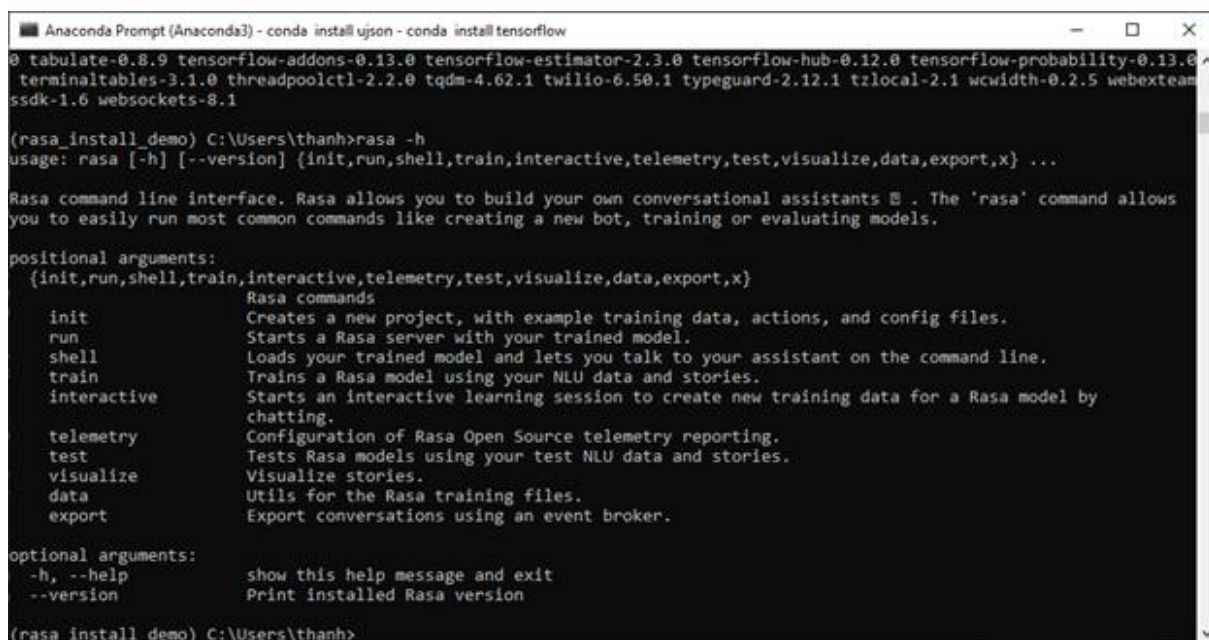
(rasa_install_demo) C:\Users\thanh>rasa -h

```

Hình 3.38. Quá trình cài đặt rasa 4 (nguồn: duythanhcse.wordpress.com)

Gõ lệnh xong thì nhấn Enter để double check.

Nếu màn hình sau xuất hiện tức là bạn đã thực sự cài đặt con AI Rasa thành công:



```

Anaconda Prompt (Anaconda3) - conda install ujson - conda install tensorflow
tabulate-0.8.9 tensorflow-addons-0.13.0 tensorflow-estimator-2.3.0 tensorflow-hub-0.12.0 tensorflow-probability-0.13.0
termintables-3.1.0 threadpoolctl-2.2.0 tqdm-4.62.1 twilio-6.50.1 typeguard-2.12.1 tzlocal-2.1 wcwidth-0.2.5 webexteam
ssdk-1.6 websockets-8.1

(rasa_install_demo) C:\Users\thanh> rasa -h
usage: rasa [-h] [--version] {init,run,shell,train,interactive,telemetry,test,visualize,data,export,x} ...

Rasa command line interface. Rasa allows you to build your own conversational assistants. The 'rasa' command allows
you to easily run most common commands like creating a new bot, training or evaluating models.

positional arguments:
  {init,run,shell,train,interactive,telemetry,test,visualize,data,export,x}
    Rasa commands
    init                Creates a new project, with example training data, actions, and config files.
    run                 Starts a Rasa server with your trained model.
    shell               Loads your trained model and lets you talk to your assistant on the command line.
    train               Trains a Rasa model using your NLU data and stories.
    interactive          Starts an interactive learning session to create new training data for a Rasa model by
                        chatting.
    telemetry            Configuration of Rasa Open Source telemetry reporting.
    test                Tests Rasa models using your test NLU data and stories.
    visualize            Visualize stories.
    data                Utils for the Rasa training files.
    export              Export conversations using an event broker.

optional arguments:
  -h, --help            show this help message and exit
  --version              Print installed Rasa version

(rasa_install_demo) C:\Users\thanh>

```

Hình 3.39. Quá trình cài đặt rasa 5 (nguồn: duythanhcse.wordpress.com)

3.2. Huấn luyện mô hình Rasa

3.2.1. Conversation-Driven Development

Conversation-Driven Development (CDD) là quá trình lắng nghe người dùng của bạn và sử dụng những hiểu biết sâu sắc đó để cải thiện trợ lý AI của bạn. Đây là phương pháp thực hành tốt nhất bao quát để phát triển chatbot.

CDD bao gồm các hành động sau:

- Chia sẻ trợ lý của bạn với người dùng càng sớm càng tốt
- Xem lại các cuộc trò chuyện một cách thường xuyên
- Chú thích tin nhắn và sử dụng chúng làm dữ liệu đào tạo NLU
- Kiểm tra xem trợ lý của bạn có luôn cư xử như bạn mong đợi không
- Theo dõi khi trợ lý của bạn thất bại và đo lường hiệu suất của nó theo thời gian
- Khắc phục cách trợ lý của bạn xử lý các cuộc trò chuyện không thành công

CDD không phải là một quá trình tuyến tính; bạn sẽ lặp đi lặp lại những hành động tương tự khi bạn phát triển và cải thiện bot của mình.

3.2.2. Tạo dữ liệu NLU

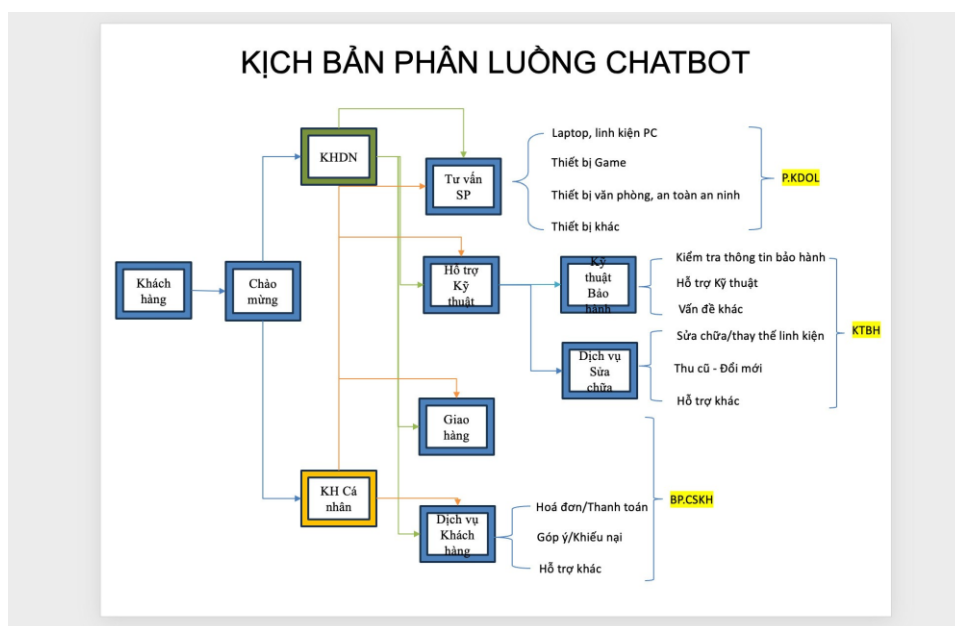
NLU (Natural Language Understanding) là một phần của Rasa thực hiện phân loại ý định, trích xuất thực thể và truy xuất phản hồi.

NLU sẽ tiếp nhận một câu như “I am looking for a French restaurant in the center of town” và trả về dữ liệu có cấu trúc như:

```
{
  "intent": "search_restaurant",
  "entities": {
    "cuisine": "French",
    "location": "center"
  }
}
```

Hình 3.40. Cấu trúc dữ liệu (nguồn: rasa.com)

Conversation-Driven Development (CDD) cho NLU có nghĩa là để các cuộc hội thoại của người dùng thực sự hướng dẫn sự phát triển của trợ lý AI. Theo đó, dựa vào dữ liệu thực tế và kịch bản phân luồng chatbot:



Hình 3.41. Kịch bản phân luồng chatbot

Ta có bộ dữ liệu huấn luyện cho chatbot được chia thành 3 phần chính là ý định (Intent), thực thể (Entity) và phản hồi (Responses). Dữ liệu tiếp tục được chia nhỏ thành 4 nhóm bao gồm: Chào mừng, Kinh doanh, Kỹ thuật, Giao hàng, Dịch vụ, Vị trí.

- Ví dụ về dữ liệu của nhóm kinh doanh:

Intent		
Group	Name	Text
Kinh doanh	kd.san_pham_chuot	[Chuột laptop>{"entity": "kd.san_pham_chuot"}
		Tôi muốn mua [chuột>{"entity": "kd.san_pham_chuot"}
		tôi muốn mua [chuột>{"entity": "kd.san_pham_chuot"}
		[chuột>{"entity": "kd.san_pham_chuot"}
		[chuột] máy tính{"entity": "kd.san_pham_chuot"}

Hình 3.42. Ví dụ về intent của nhóm kinh doanh

Entity	
Name	Keyword
kd.san_pham_chuot	chuột

Hình 3.43. Ví dụ về entity của nhóm kinh doanh

Responses	
Name	Text
utter_kd.san_pham_chuot	"Đây là danh sách {kd.san_pham_chuot} bên mình đang quan tâm: https://hacom.vn/chuot-may-tinh . Bạn vui lòng cung cấp mã sản phẩm bạn đang quan tâm để mình tư vấn thêm cho bạn."

Hình 3.44. Ví dụ về responses của nhóm kinh doanh

3.2.3. Huấn luyện mô hình

Từ dữ liệu đó, các tin nhắn đến được xử lý theo một chuỗi các thành phần. Các thành phần này được thực thi lần lượt trong cái gọi là xử lý **pipeline** được xác định trong tệp config.yml.

```
pipeline:
- name: "WhitespaceTokenizer"
- name: "RegexFeaturizer" |
- name: "CountVectorsFeaturizer"
- name: "DIETClassifier"
  epochs: 100 # giảm vòng lặp
- name: "EntitySynonymMapper"
- name: "ResponseSelector"
  epochs: 100 # giảm vòng lặp
```

Hình 3.45. pipeline

Theo đó:

a) `WhitespaceTokenizer`:

- Là một công cụ xử lý văn bản trong thư viện NLP (Natural Language Processing). Nó được sử dụng để tách một chuỗi văn bản thành các token dựa trên khoảng trắng. Cách thức hoạt động của `WhitespaceTokenizer` rất đơn giản: nó chỉ tìm kiếm các khoảng trắng trong chuỗi và sử dụng chúng làm điểm phân chia để tách các từ.

- Đây là một bước tiền xử lý đơn giản để tách văn bản thành các token (từ ngữ) bằng cách sử dụng khoảng trắng như dấu phân cách.

b) `RegexFeaturizer`

- Là một thành phần trong Rasa NLU, một nền tảng mã nguồn mở dành cho xử lý ngôn ngữ tự nhiên. `RegexFeaturizer` sử dụng biểu thức chính quy (regular expressions) để tạo ra các đặc trưng (features) từ văn bản. Những đặc trưng này sau đó được sử dụng trong các mô hình máy học để cải thiện khả năng nhận diện các thực thể (entities) và ý định (intent).

Cách hoạt động của `RegexFeaturizer`:

- Định nghĩa các biểu thức chính quy (regex patterns):
 - Trước tiên, bạn cần định nghĩa các mẫu biểu thức chính quy mà bạn muốn hệ thống nhận diện. Các mẫu này có thể được định nghĩa trong tệp cấu hình của dự án (thường là tệp `nlu.yml` hoặc `config.yml`).
 - Ví dụ, bạn có thể muốn nhận diện các số điện thoại, địa chỉ email, hoặc các từ khóa cụ thể trong văn bản.
 - Áp dụng các biểu thức chính quy:
 - `RegexFeaturizer` sẽ áp dụng các mẫu regex lên văn bản đầu vào để kiểm tra xem có khớp với bất kỳ mẫu nào không.
 - Nếu một phần của văn bản khớp với một mẫu regex, `RegexFeaturizer` sẽ tạo ra một đặc trưng (feature) tương ứng cho mẫu đó.
 - Tạo ra các đặc trưng:
-

- Mỗi lần một mẫu regex được khớp, `RegexFeaturizer` sẽ thêm một đặc trưng nhị phân (binary feature) vào biểu diễn đặc trưng của văn bản. Giá trị của đặc trưng này sẽ là 1 nếu mẫu regex khớp và 0 nếu không khớp.

- Các đặc trưng này sau đó sẽ được kết hợp với các đặc trưng khác từ các thành phần featurizer khác (như `CountVectorsFeaturizer`) để tạo ra biểu diễn cuối cùng của văn bản.

c) `CountVectorsFeaturizer`:

- Là một thành phần quan trọng trong pipeline của Rasa, được sử dụng để chuyển đổi văn bản thành các đặc trưng số học (feature vectors) dựa trên sự xuất hiện của từ (word counts) trong văn bản. Nó dựa trên kỹ thuật Bag-of-Words (BoW), một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên (NLP).

Cách hoạt động của `CountVectorsFeaturizer`:

- Tạo từ điển (Vocabulary Creation):
 - `CountVectorsFeaturizer` tạo một từ điển của tất cả các từ xuất hiện trong tập dữ liệu huấn luyện.
 - Mỗi từ trong từ điển sẽ được ánh xạ đến một chỉ số duy nhất.
- Tính toán tần suất từ (Term Frequency Calculation):
 - Đối với mỗi văn bản đầu vào, `CountVectorsFeaturizer` đếm số lần xuất hiện của mỗi từ trong từ điển.
 - Một vector đặc trưng (feature vector) được tạo ra, trong đó mỗi phần tử của vector đại diện cho tần suất xuất hiện của một từ trong văn bản đó.
- Biểu diễn văn bản dưới dạng vector (Vector Representation of Text):
 - Văn bản được biểu diễn dưới dạng vector số học, với độ dài bằng số từ trong từ điển.
 - Nếu một từ trong từ điển không xuất hiện trong văn bản, giá trị tương ứng trong vector sẽ là 0.

d) `DIETClassifier`:

- (Dual Intent and Entity Transformer) là một mô hình học sâu mạnh mẽ được sử dụng trong Rasa để thực hiện đồng thời hai nhiệm vụ chính: phân loại ý định (intent classification) và nhận diện thực thể (entity recognition). Dựa trên kiến trúc

transformer, DIETClassifier cải thiện hiệu suất và độ chính xác so với các mô hình truyền thống bằng cách tận dụng cơ chế tự chú ý (self-attention mechanism).

Cách thức hoạt động của DIETClassifier:

- Tiền xử lý dữ liệu:
 - Tokenization: Văn bản đầu vào được tách thành các token (thường sử dụng các thành phần như `WhitespaceTokenizer`).
 - Featurization: Các đặc trưng (features) được tạo ra từ văn bản bằng cách sử dụng các thành phần như `CountVectorsFeaturizer` và `RegexFeaturizer`. Những đặc trưng này có thể bao gồm cả các n-gram và các mẫu regex khớp với văn bản.
 - Embedding Layer:
 - Các token được chuyển đổi thành các vector embedding, là các biểu diễn số học của từ ngữ. DIETClassifier hỗ trợ cả các embedding được học trong quá trình huấn luyện và các embedding tiền huấn luyện (pre-trained embeddings) như GloVe hoặc FastText.
 - Transformer Encoder:
 - Các vector embedding sau đó được đưa qua một hoặc nhiều lớp transformer encoder. Các lớp này sử dụng cơ chế tự chú ý để học các mối quan hệ và ngữ cảnh của các từ trong câu.
 - Song song xử lý ý định và thực thể:
 - Nhận diện ý định (Intent Classification): Đầu ra của transformer encoder được sử dụng để dự đoán ý định của câu thông qua một lớp fully connected và hàm softmax để tính toán xác suất của các ý định.
 - Nhận diện thực thể (Entity Recognition): Đầu ra của transformer encoder cũng được sử dụng để dự đoán các thực thể trong câu. Các token được gán nhãn thực thể thông qua một lớp fully connected và cơ chế CRF (Conditional Random Field) để đảm bảo tính nhất quán trong gán nhãn.
 - Huấn luyện và tối ưu hóa:
 - Mô hình được huấn luyện bằng cách sử dụng một hàm mất mát kết hợp, bao gồm mất mát cho nhận diện ý định và mất mát cho nhận diện thực thể.
-

- Các thuật toán tối ưu hóa như Adam hoặc SGD được sử dụng để cập nhật trọng số của mô hình.

e) EntitySynonymMapper

- Là một thành phần trong Rasa NLU, được sử dụng để ánh xạ các thực thể (entities) sang các từ đồng nghĩa tương ứng. Điều này giúp cải thiện độ chính xác của mô hình bằng cách chuẩn hóa các giá trị thực thể khác nhau về một giá trị thống nhất. Khi các thực thể được nhận diện trong văn bản, EntitySynonymMapper sẽ thay thế chúng bằng từ đồng nghĩa đã được định nghĩa, đảm bảo rằng tất cả các biến thể của một thực thể đều được xử lý nhất quán trong các bước tiếp theo của pipeline.

```
- intent: kd.san_pham_PC
examples: |
  - [Máy tính>{"entity": "kd.san_pham_PC"} mini đơn giản
  - Tôi cần [máy mini>{"entity": "kd.san_pham_PC"} chạy ít tốn điện nhất
  - tôi cần mua [PC>{"entity": "kd.san_pham_PC"}
  - [máy tính>{"entity": "kd.san_pham_PC"}
  - [máy tính cây>{"entity": "kd.san_pham_PC"}
  - [PC>{"entity": "kd.san_pham_PC"}
```

Hình 3.46. Ví dụ về EntitySynonymMapper

- Trong ví dụ trên, “máy mini”, “PC” và “Máy tính” được định nghĩa là từ đồng nghĩa với nhau và ánh xạ sang thực thể “kd.san_pham_PC”.

f) ResponseSelector:

- Là một thành phần trong Rasa NLU, được thiết kế để lựa chọn phản hồi phù hợp nhất từ một tập hợp các phản hồi tiềm năng dựa trên ý định và nội dung của tin nhắn người dùng. Nó rất hữu ích trong các ứng dụng chatbot và trợ lý ảo, nơi có nhiều câu trả lời khả thi cho các câu hỏi hoặc yêu cầu của người dùng.

- Giả sử người dùng nhập: "How can I track my order?"

- Nhận diện ý định: DIETClassifier nhận diện ý định faq.
- Dự đoán phản hồi: ResponseSelector xác định rằng phản hồi phù hợp nhất cho câu hỏi này là utter_faq/track_order.
- Trả lời người dùng: Bot trả lời người dùng với nội dung: "You can track your order using the tracking number sent to your email."

```

nlu:
- intent: faq
  examples: |
    - What is your return policy?
    - How can I track my order?
    - What are your shipping options?

responses:
  utter_faq/return_policy:
  - text: "Our return policy lasts 30 days."

  utter_faq/track_order:
  - text: "You can track your order using the tracking number sent to your email."

  utter_faq/shipping_options:
  - text: "We offer standard, express, and next-day shipping options."

```

Hình 3.47. Định nghĩa phản hồi trong các tệp “nlu.yml” (rasa.com)

Tiếp theo sẽ là Policy: Các thành phần Rasa dự đoán hành động tiếp theo của hệ thống đối thoại. Các chính sách đưa ra quyết định về cách tiến hành luồng hội thoại. Một cấu hình điển hình bao gồm nhiều chính sách và chính sách có độ tin cậy cao nhất sẽ quyết định hành động tiếp theo sẽ được thực hiện trong cuộc trò chuyện.

```

policies:
- name: "MemoizationPolicy"
- name: RulePolicy
- name: "TEDPolicy"
  max_history: 70
  epochs: 100 # giảm vòng lặp

```

Hình 3.48. policies

Các chính sách bao gồm:

- MemoizationPolicy: là một chính sách trong Rasa có nhiệm vụ ghi nhớ và sử dụng lại các mẫu hội thoại đã gặp trong quá trình huấn luyện để quyết định phản hồi của bot. Chính sách này giúp bot nhanh chóng phản hồi khi các cuộc hội thoại khớp chính xác với những gì đã được học, làm tăng hiệu quả và tốc độ của bot trong các kịch bản đã biết.

- RulePolicy: trong Rasa là một thành phần cho phép bạn xác định các quy tắc cụ thể mà bot sẽ tuân theo khi phản hồi người dùng. Điều này rất hữu ích trong việc xử lý các quy trình định hướng rõ ràng hoặc các hành động cụ thể mà bạn muốn bot thực hiện dựa trên các điều kiện nhất định. RulePolicy sử dụng các quy tắc do người dùng xác định để định hướng hành vi của bot trong các cuộc hội thoại.

```
- rule: Xin chào
  steps:
  - intent: grt.greet
  - action: utter_grt.greet

- rule: Giới thiệu HACOM...

- rule: Hỏi hỗ trợ khác
  steps:
  - intent: grt.ho_tro_khac
  - action: utter_grt.ho_tro_khac

- rule: Hỏi kết nối với chăm sóc khách hàng
  steps:
  - intent: grt.contact_cskh
  - action: utter_grt.contact_cskh

- rule: Hỏi nhân viên kinh doanh
  steps:
  - intent: grt.contact_kdol
  - action: utter_grt.contact_kdol
```

Hình 3.49. Định nghĩa các quy tắc trong tập “rules.yml”

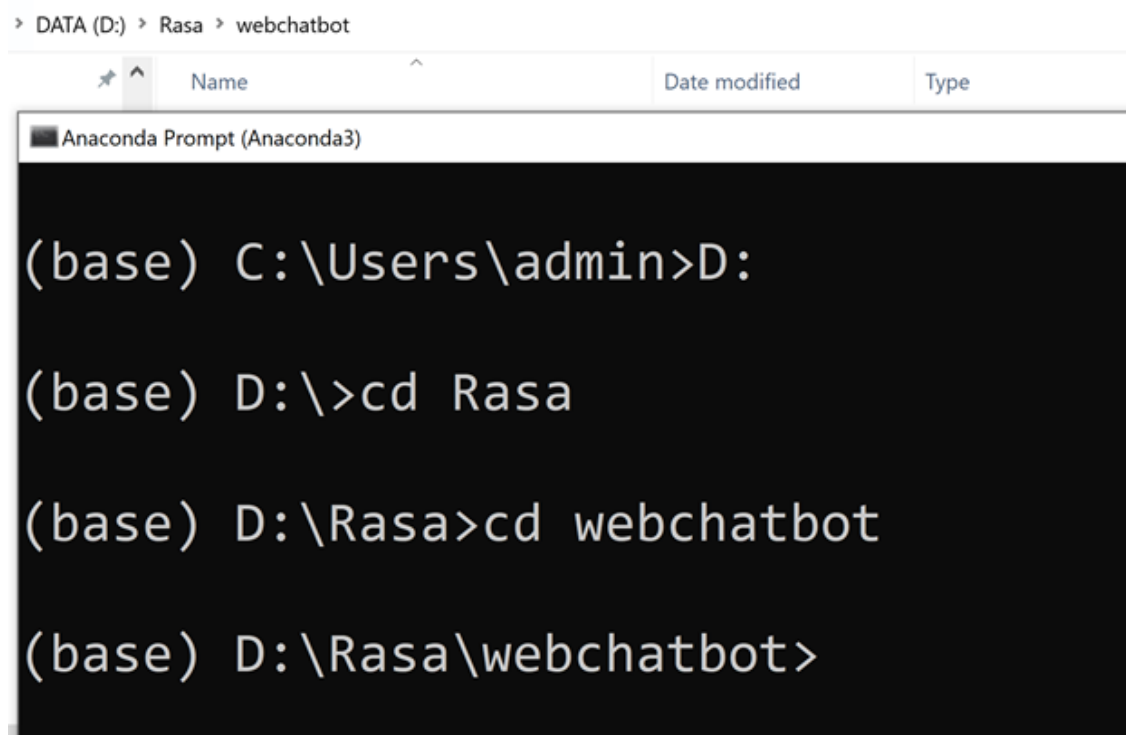
- TEDPolicy:
 - Một mô hình huấn luyện sử dụng phương pháp học tăng cường để xác định phản hồi dựa trên lịch sử hội thoại.
 - max_history: 70 chỉ ra rằng mô hình sẽ xem xét tối đa 70 bước trước đó của hội thoại để quyết định phản hồi tiếp theo.

- epochs: 100 chỉ ra rằng mô hình sẽ được huấn luyện trong 100 vòng lặp, giảm so với mặc định để tiết kiệm thời gian huấn luyện.

3.3. Tích hợp Chatbot với website

Tạo thư mục sau trong ổ D: D:/Rasa/webchatbot

Sau đó mở Anaconda prompt lên để di chuyển lệnh vào thư mục này. Xem hình và gõ theo:

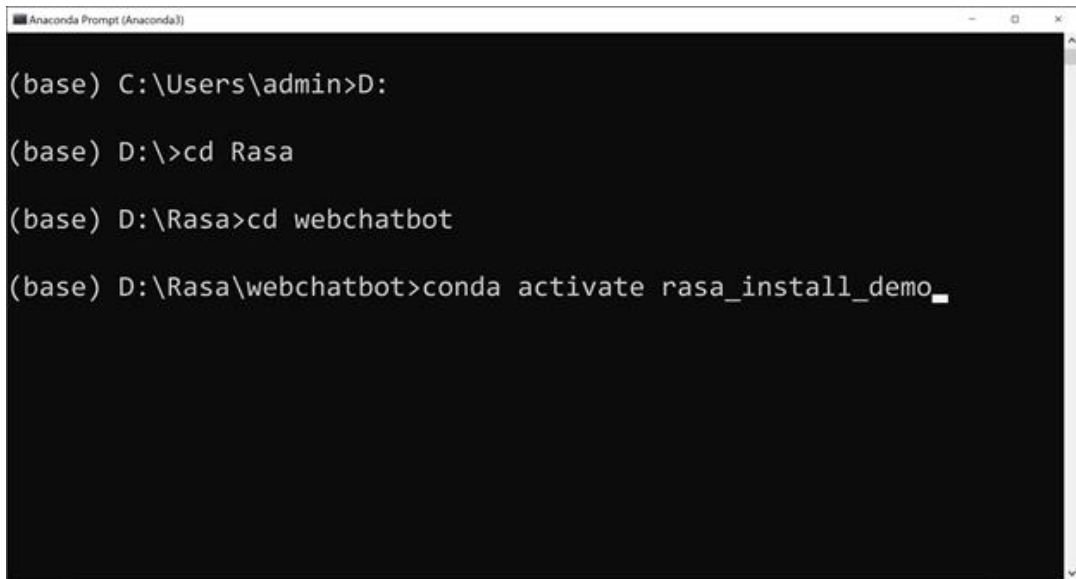


Hình 3.50. Tích hợp chatbot lên website 1 (nguồn: duythanhcse.wordpress.com)

Sau đó activate rasa_install_demo bằng lệnh:

```
1 | conda activate rasa_install_demo
```

Xem hình minh họa:



```

Anaconda Prompt (Anaconda3)

(base) C:\Users\admin>D:

(base) D:\>cd Rasa

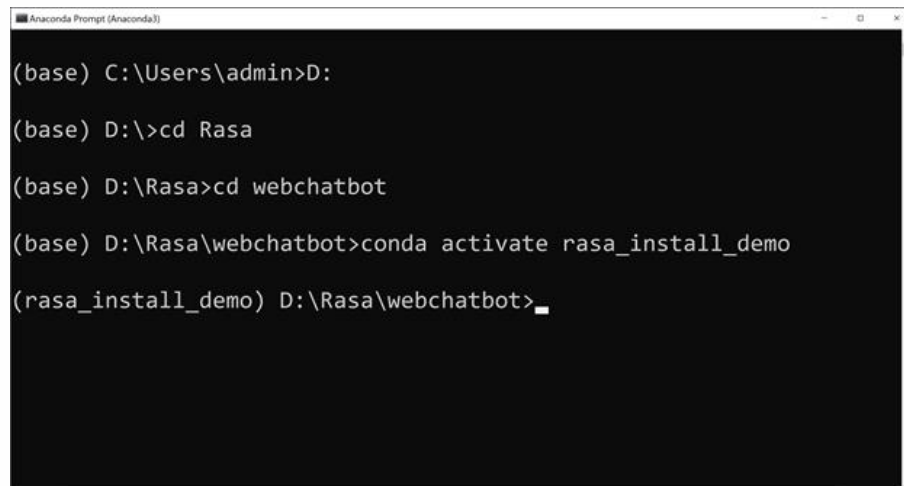
(base) D:\Rasa>cd webchatbot

(base) D:\Rasa\webchatbot>conda activate rasa_install_demo_
  
```

Hình 3.51. Tích hợp chatbot lên website 2 (nguồn: duythanhcse.wordpress.com)

Gõ lệnh cho đúng như trên rồi nhấn enter

Sau khi Nhấn Enter, xem hình kết quả:



```

Anaconda Prompt (Anaconda3)

(base) C:\Users\admin>D:

(base) D:\>cd Rasa

(base) D:\Rasa>cd webchatbot

(base) D:\Rasa\webchatbot>conda activate rasa_install_demo

(rasa_install_demo) D:\Rasa\webchatbot>
  
```

Hình 3.52. Tích hợp chatbot lên website 3 (nguồn: duythanhcse.wordpress.com)

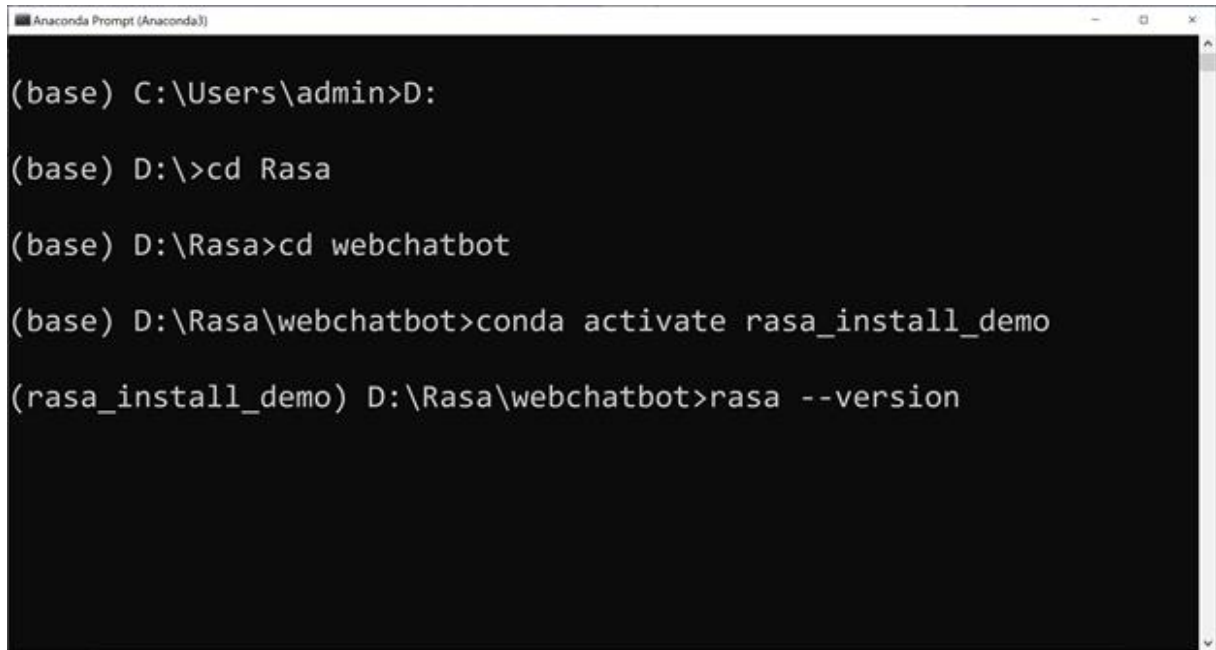
Như vậy nó đã được activated.

(rasa_install_demo) D:/Rasa/webchatbot>

Bây giờ ta thử Kiểm tra version của Rasa, gõ lệnh:

```
1 | rasa --version
```

xem hình gõ lệnh:

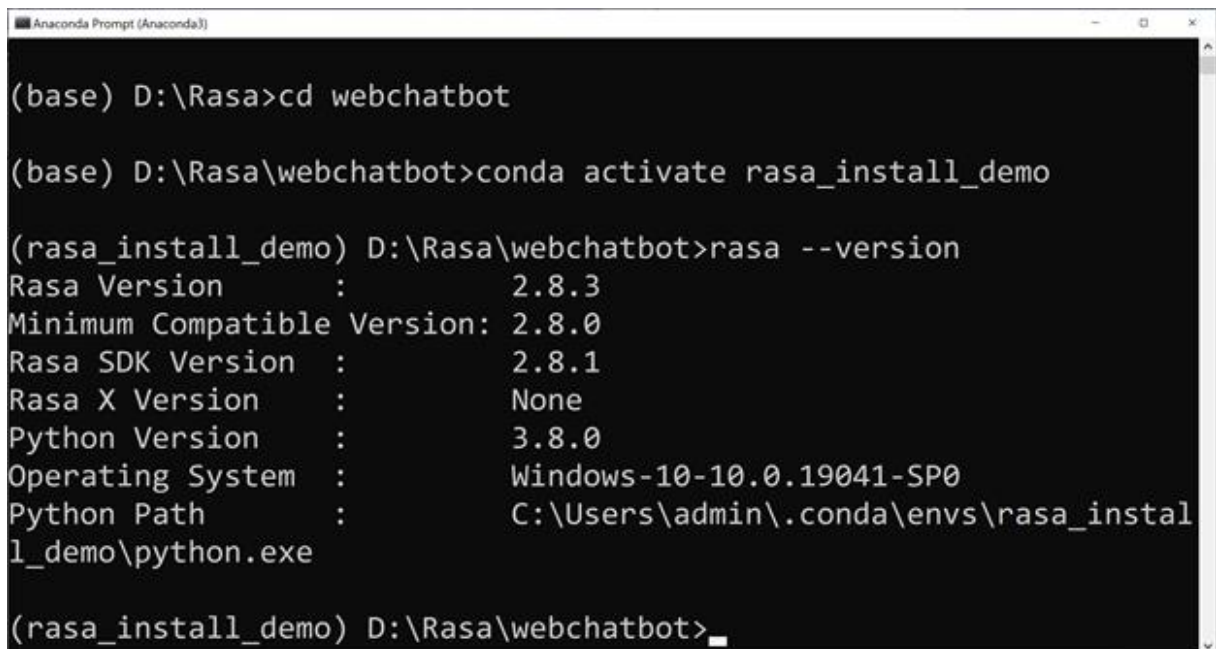


```

Anaconda Prompt (Anaconda3)
(base) C:\Users\admin>D:
(base) D:\>cd Rasa
(base) D:\Rasa>cd webchatbot
(base) D:\Rasa\webchatbot>conda activate rasa_install_demo
(rasa_install_demo) D:\Rasa\webchatbot>rasa --version
  
```

Hình 3.53. Tích hợp chatbot lên website 4 (nguồn: duythanhcse.wordpress.com)

Gõ Enter, xem kết quả:



```

Anaconda Prompt (Anaconda3)
(base) D:\Rasa>cd webchatbot
(base) D:\Rasa\webchatbot>conda activate rasa_install_demo
(rasa_install_demo) D:\Rasa\webchatbot>rasa --version
Rasa Version      :      2.8.3
Minimum Compatible Version: 2.8.0
Rasa SDK Version  :      2.8.1
Rasa X Version    :      None
Python Version    :      3.8.0
Operating System  :      Windows-10-10.0.19041-SP0
Python Path       :      C:\Users\admin\.conda\envs\rasa_install_demo\python.exe
(rasa_install_demo) D:\Rasa\webchatbot>
  
```

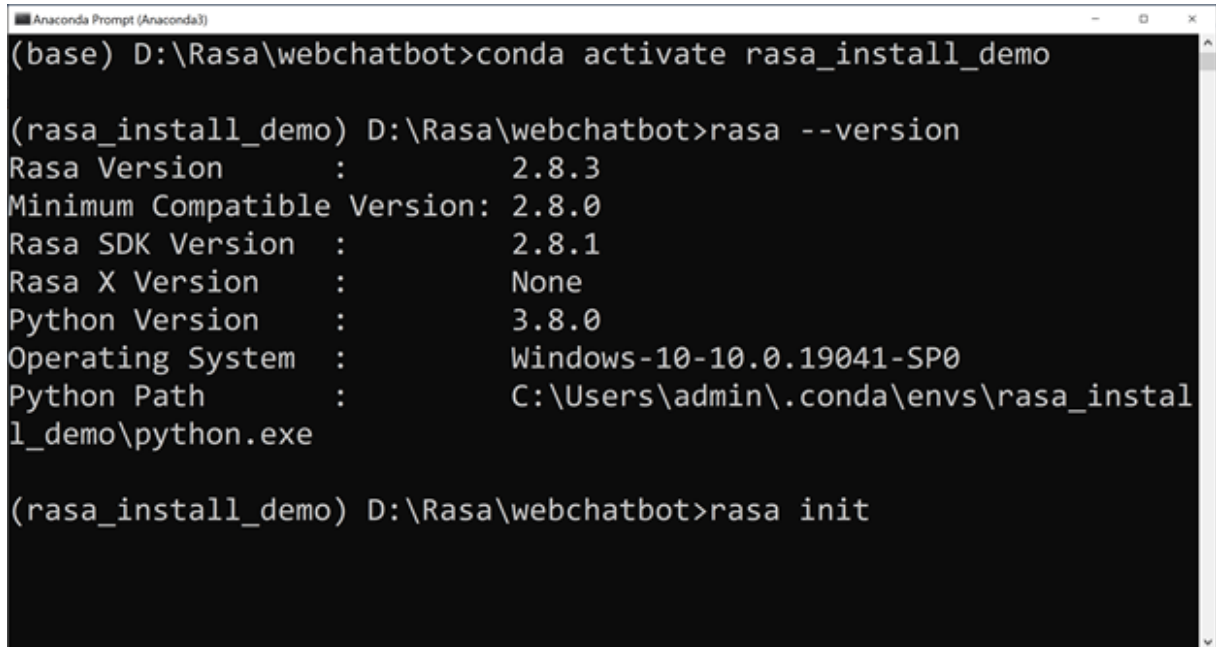
Hình 3.54. Tích hợp chatbot lên website 5 (nguồn: duythanhcse.wordpress.com)

Màn hình trên cho ta biết các thông số về phiên bản con AI Rasa.

Tiếp theo: Khởi tạo Rasa bằng cách gõ lệnh:

1 `| rasa init`

Xem hình:



```

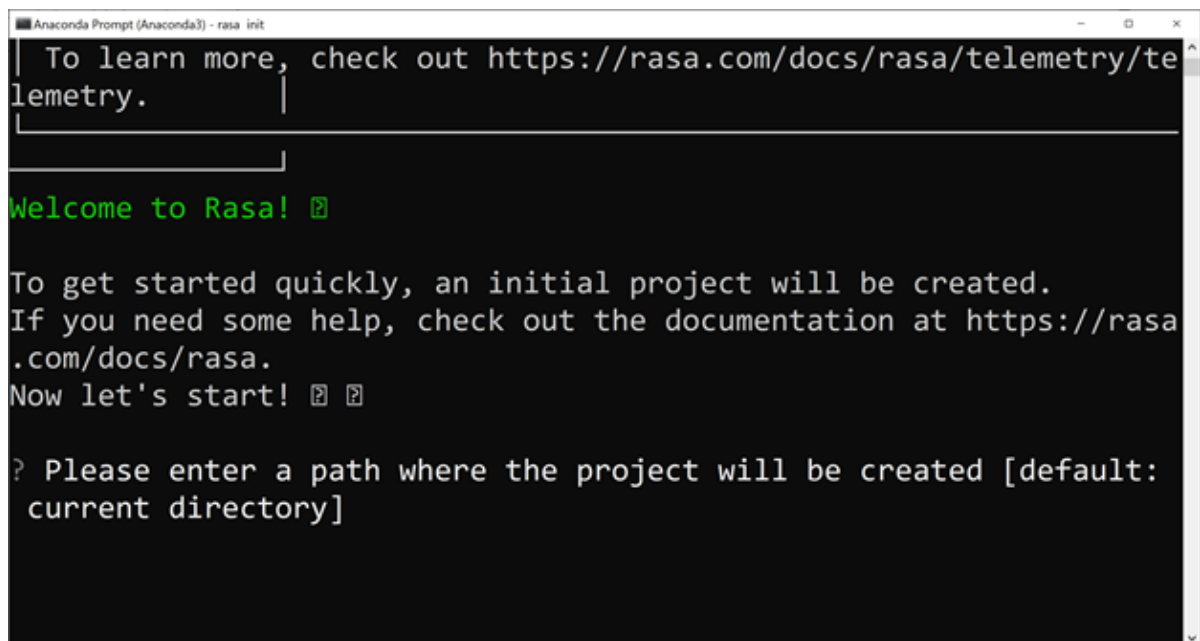
Anaconda Prompt (Anaconda3)
(base) D:\Rasa\webchatbot>conda activate rasa_install_demo

(rasa_install_demo) D:\Rasa\webchatbot>rasa --version
Rasa Version      :      2.8.3
Minimum Compatible Version: 2.8.0
Rasa SDK Version  :      2.8.1
Rasa X Version    :      None
Python Version    :      3.8.0
Operating System  :      Windows-10-10.0.19041-SP0
Python Path       :      C:\Users\admin\.conda\envs\rasa_install_demo\python.exe

(rasa_install_demo) D:\Rasa\webchatbot>rasa init
  
```

Hình 3.55. Tích hợp chatbot lên website 6 (nguồn: duythanhcse.wordpress.com)

Nhấn Enter, xem kết quả:



```

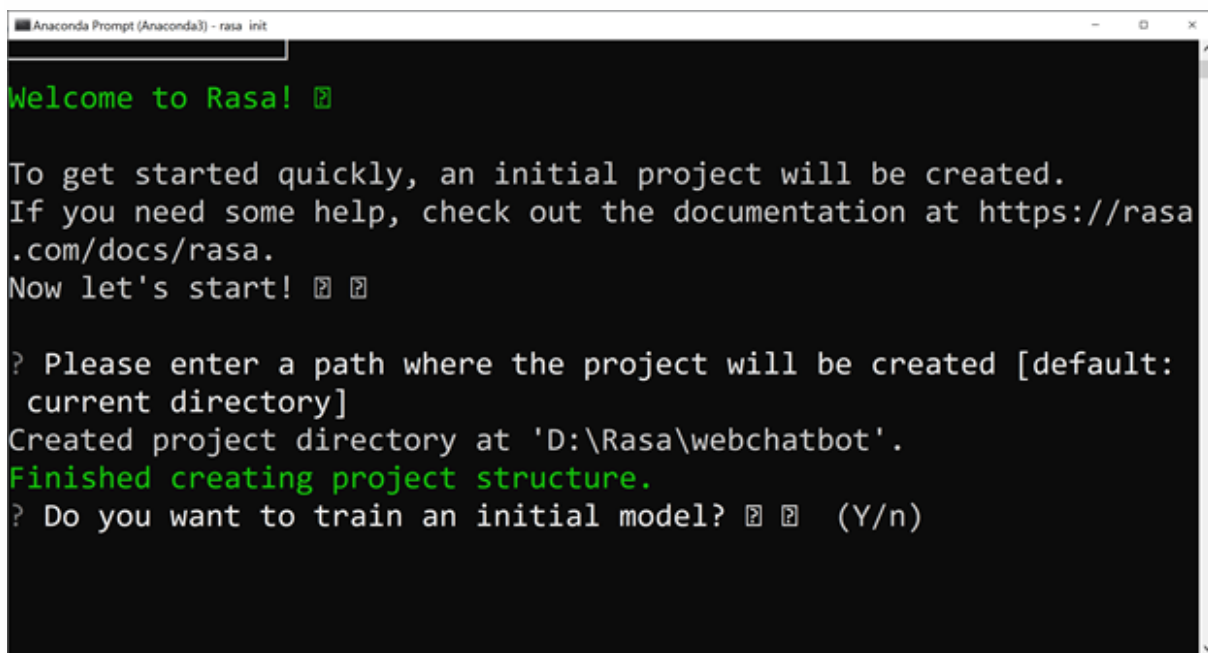
Anaconda Prompt (Anaconda3) - rasa init
| To learn more, check out https://rasa.com/docs/rasa/telemetry/telemetry.
|
|
Welcome to Rasa! 🐘

To get started quickly, an initial project will be created.
If you need some help, check out the documentation at https://rasa.com/docs/rasa.
Now let's start! 🐘 🐘

? Please enter a path where the project will be created [default: current directory]
  
```

Hình 3.56. Tích hợp chatbot lên website 7 (nguồn: duythanhcse.wordpress.com)

Ở màn hình trên là hỏi ta Tạo nơi lưu trữ dự án, vì đang trỏ lệnh ở D:/rasa/webchatbot nên ta nhấn enter luôn để tạo mặc định. Xem kết quả:



```
Anaconda Prompt (Anaconda3) - rasa init

Welcome to Rasa! 🐼

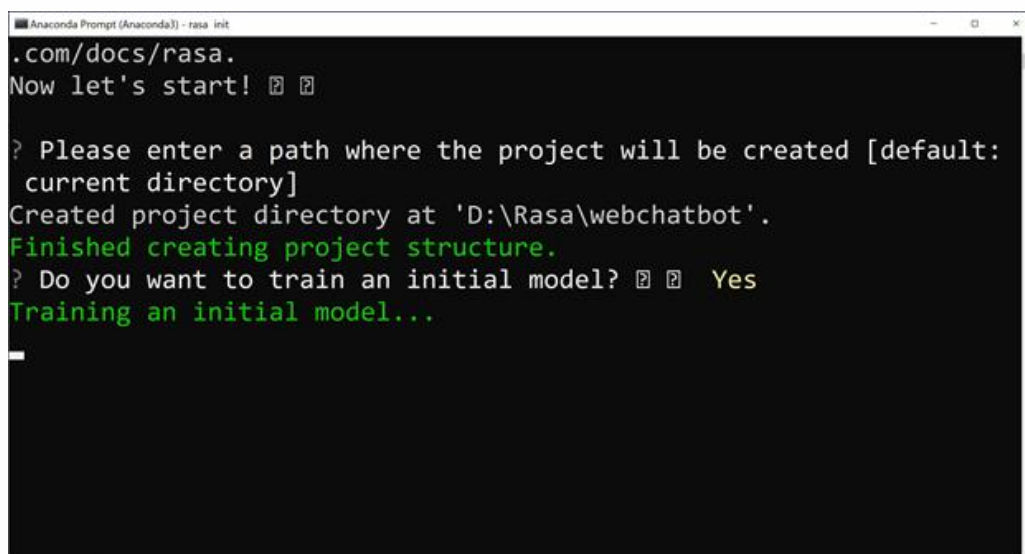
To get started quickly, an initial project will be created.
If you need some help, check out the documentation at https://rasa.com/docs/rasa.
Now let's start! 🐼 🐼

? Please enter a path where the project will be created [default:
  current directory]
Created project directory at 'D:\Rasa\webchatbot'.
Finished creating project structure.
? Do you want to train an initial model? 🐼 🐼 (Y/n)
```

Hình 3.57. Tích hợp chatbot lên website 8 (nguồn: duythanhcse.wordpress.com)

Nó hỏi có train một model lúc khởi tạo không, ta chọn Y.

và chờ nó thực hiện:

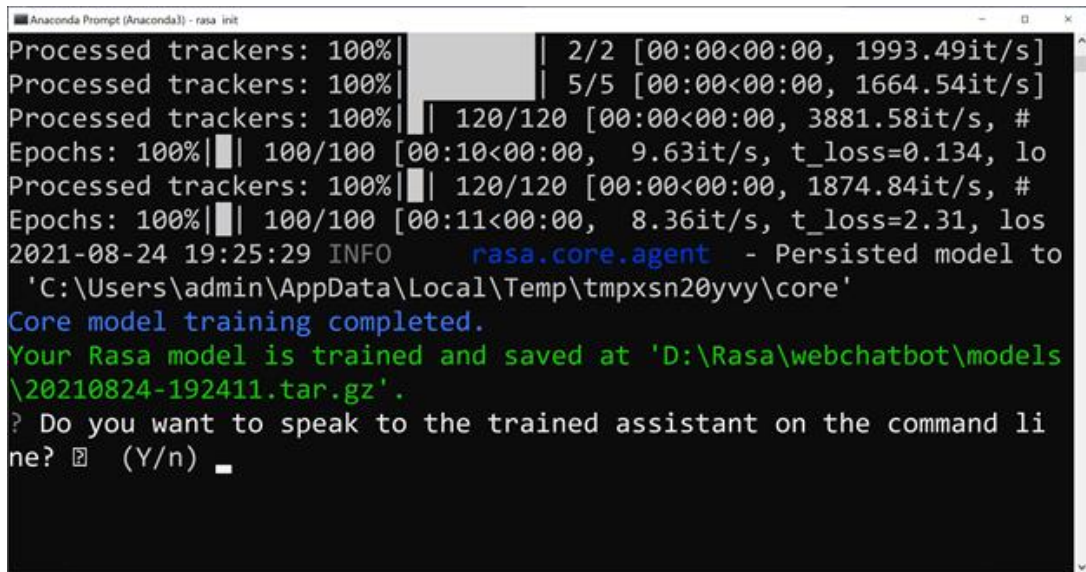


```
.com/docs/rasa.
Now let's start! 🐼 🐼

? Please enter a path where the project will be created [default:
  current directory]
Created project directory at 'D:\Rasa\webchatbot'.
Finished creating project structure.
? Do you want to train an initial model? 🐼 🐼 Yes
Training an initial model...
█
```

Hình 3.58. Tích hợp chatbot lên website 9 (nguồn: duythanhcse.wordpress.com)

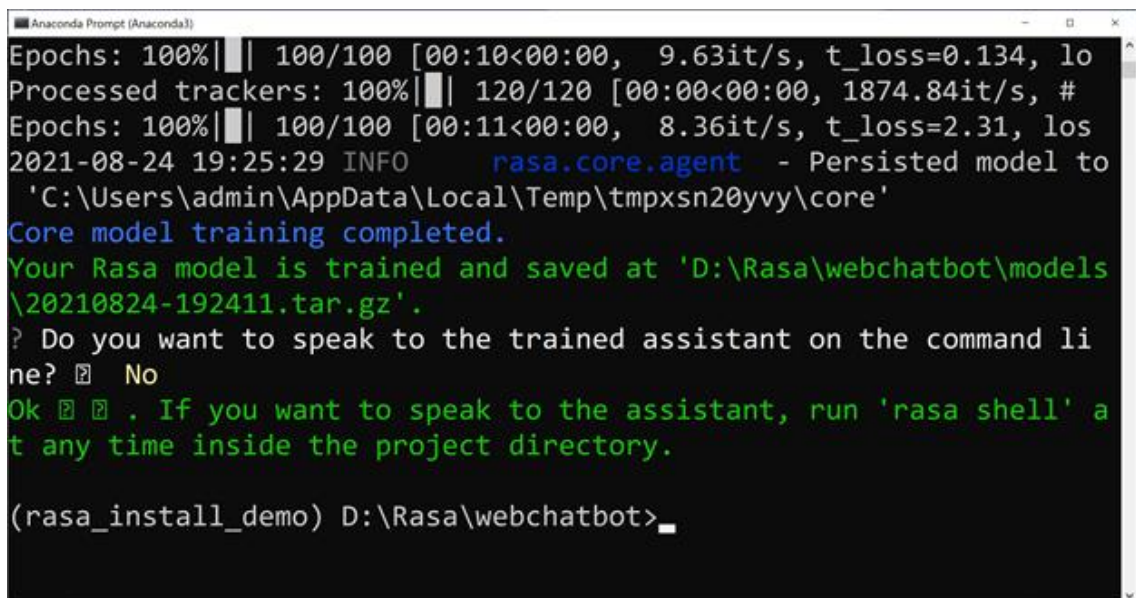
Tùy vào độ khỏe của máy mà bạn thấy màn hình dưới đây lâu hay mau:



```
Anaconda Prompt (Anaconda3) - rasa init
Processed trackers: 100%|██████████| 2/2 [00:00<00:00, 1993.49it/s]
Processed trackers: 100%|██████████| 5/5 [00:00<00:00, 1664.54it/s]
Processed trackers: 100%|██████████| 120/120 [00:00<00:00, 3881.58it/s, #
Epochs: 100%|██████████| 100/100 [00:10<00:00, 9.63it/s, t_loss=0.134, lo
Processed trackers: 100%|██████████| 120/120 [00:00<00:00, 1874.84it/s, #
Epochs: 100%|██████████| 100/100 [00:11<00:00, 8.36it/s, t_loss=2.31, los
2021-08-24 19:25:29 INFO      rasa.core.agent - Persisted model to
'C:\Users\admin\AppData\Local\Temp\tmpxsn20yvy\core'
Core model training completed.
Your Rasa model is trained and saved at 'D:\Rasa\webchatbot\models
\20210824-192411.tar.gz'.
? Do you want to speak to the trained assistant on the command li
ne? [Y/n] _
```

Hình 3.59. Tích hợp chatbot lên website 10 (nguồn: duythanhcse.wordpress.com)

Tạm thời ta chưa xử lý giọng nói, nên ta nhập **n** rồi nhấn enter, và chờ nó thực hiện tiếp:



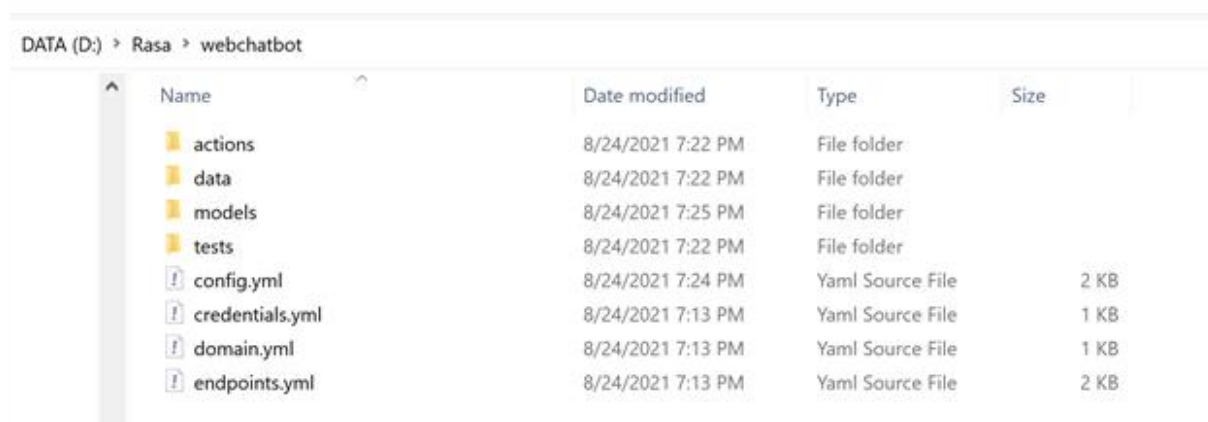
```
Epochs: 100%|██████████| 100/100 [00:10<00:00, 9.63it/s, t_loss=0.134, lo
Processed trackers: 100%|██████████| 120/120 [00:00<00:00, 1874.84it/s, #
Epochs: 100%|██████████| 100/100 [00:11<00:00, 8.36it/s, t_loss=2.31, los
2021-08-24 19:25:29 INFO      rasa.core.agent - Persisted model to
'C:\Users\admin\AppData\Local\Temp\tmpxsn20yvy\core'
Core model training completed.
Your Rasa model is trained and saved at 'D:\Rasa\webchatbot\models
\20210824-192411.tar.gz'.
? Do you want to speak to the trained assistant on the command li
ne? [Y] No
Ok [Y] [Y]. If you want to speak to the assistant, run 'rasa shell' a
t any time inside the project directory.

(rasa_install_demo) D:\Rasa\webchatbot>_
```

Hình 3.60. Tích hợp chatbot lên website 11 (nguồn: duythanhcse.wordpress.com)

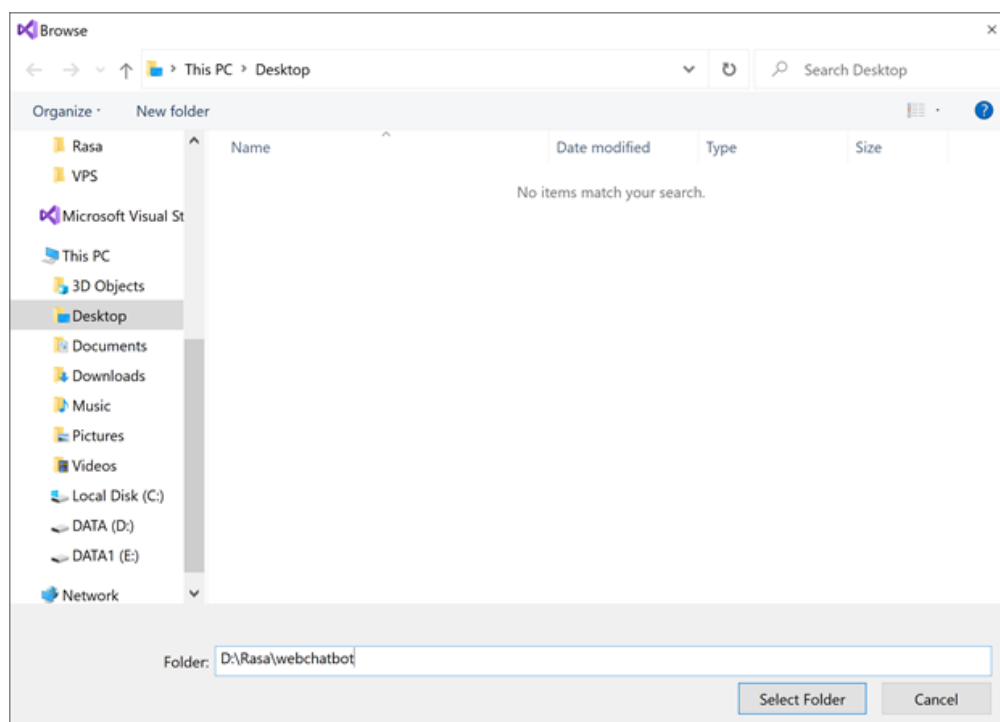
Như vậy nó đã tạo ra dự án cho bạn, vào thư mục để kiểm tra.

Vào ổ D bạn sẽ thấy dự án được tạo ra như dưới đây:



Hình 3.61. Tích hợp chatbot lên website 12 (nguồn: duythanhcse.wordpress.com)

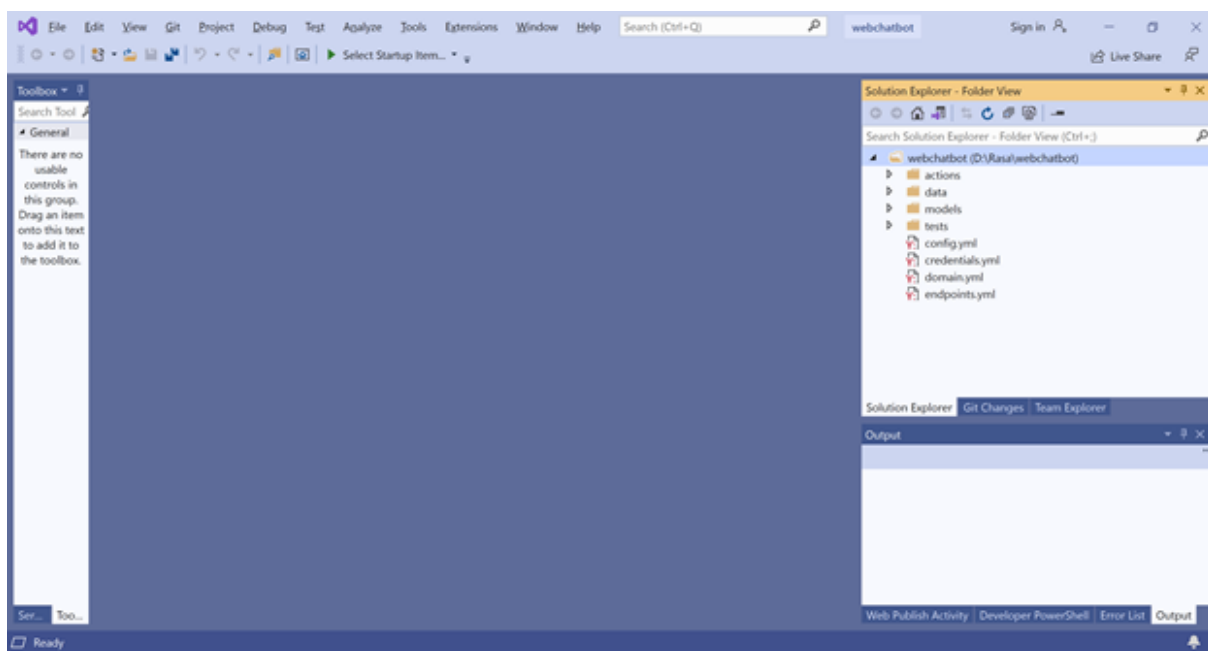
Từ Visual Studio. Chọn “**Open a local folder**”, chọn thư mục của dự án :



Hình 3.62. Tích hợp chatbot lên website 13 (nguồn: duythanhcse.wordpress.com)

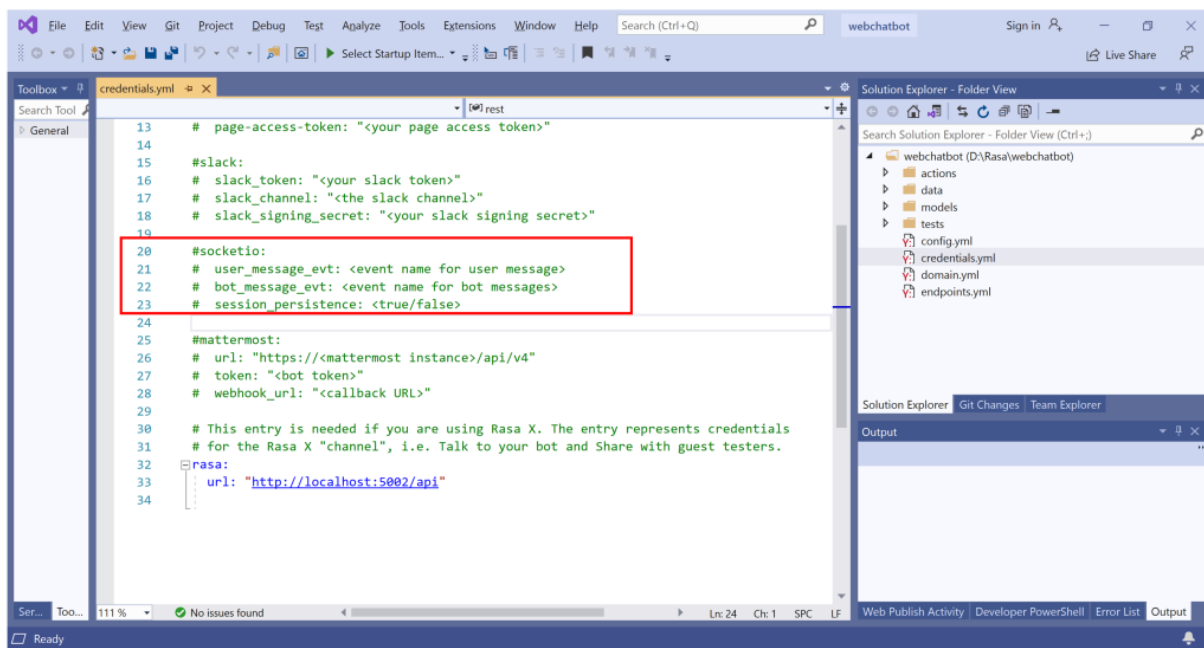
Bấm “Select Folder”

Kết quả mở trong Visual Studio:



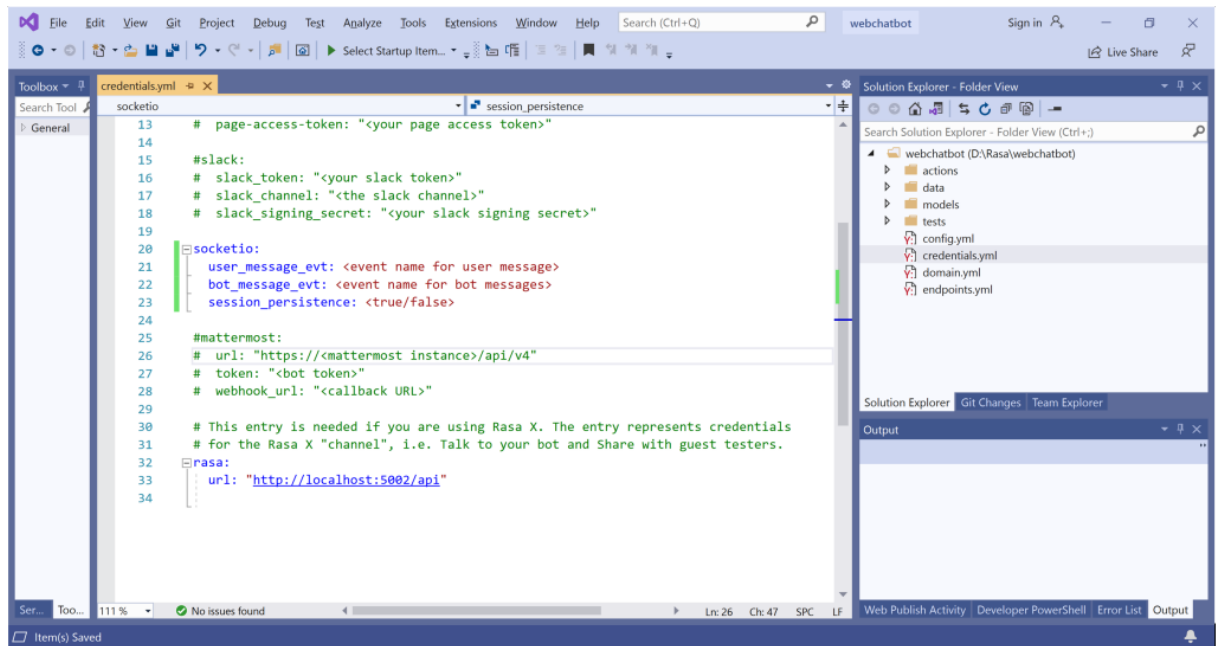
Hình 3.63. Tích hợp chatbot lên website 14 (nguồn: duythanhcse.wordpress.com)

Bây giờ mở file “credentials.yml” và chỉnh sửa:



Hình 3.64. Tích hợp chatbot lên website 15 (nguồn: duythanhcse.wordpress.com)

Dòng 20 đến 23 mở nó ra:



Hình 3.65. Tích hợp chatbot lên website 16 (nguồn: duythanhcse.wordpress.com)

Sửa chỗ này:

```

1 socketio:
2   user_message_evt:
3   bot_message_evt:
4   session_persistence:

```

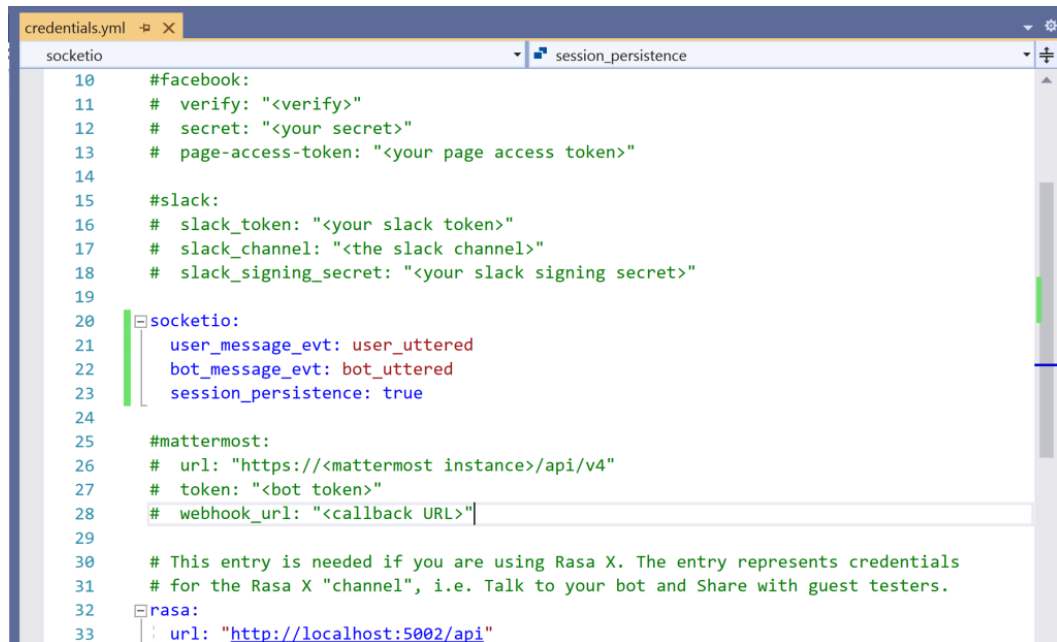
Thành:

```

1 socketio:
2   user_message_evt: user_uttered
3   bot_message_evt: bot_uttered
4   session_persistence: true

```

Xem hình:



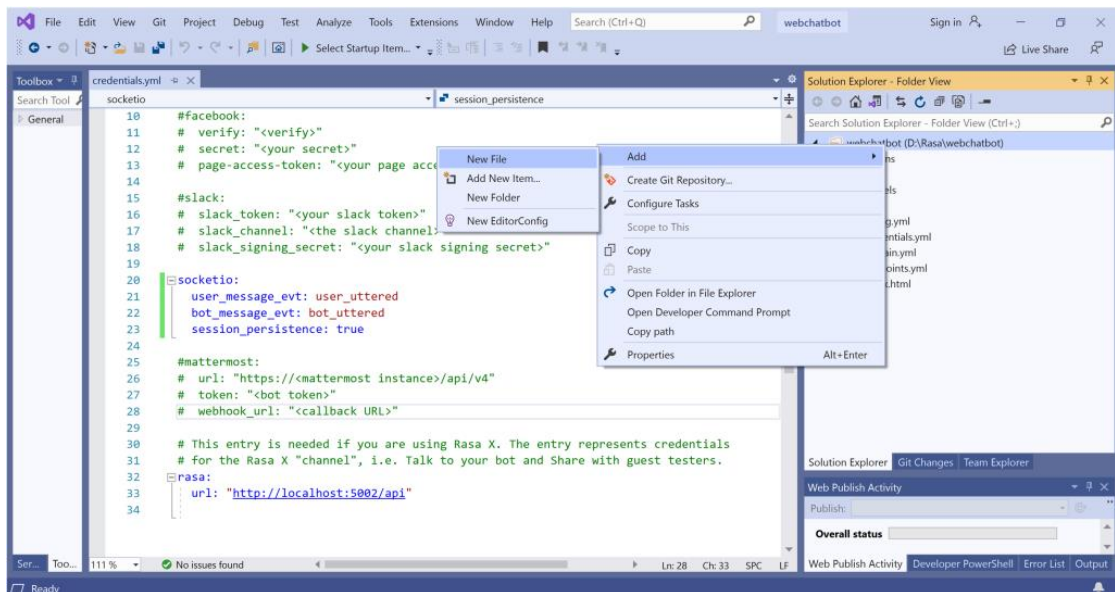
```

10 #facebook:
11 #  verify: "<verify>"
12 #  secret: "<your secret>"
13 #  page-access-token: "<your page access token>"
14
15 #slack:
16 #  slack_token: "<your slack token>"
17 #  slack_channel: "<the slack channel>"
18 #  slack_signing_secret: "<your slack signing secret>"
19
20 socketio:
21   user_message_evt: user_uttered
22   bot_message_evt: bot_uttered
23   session_persistence: true
24
25 #mattermost:
26 #  url: "https://<mattermost instance>/api/v4"
27 #  token: "<bot token>"
28 #  webhook_url: "<callback URL>"
29
30 # This entry is needed if you are using Rasa X. The entry represents credentials
31 # for the Rasa X "channel", i.e. Talk to your bot and Share with guest testers.
32 rasa:
33   url: "http://localhost:5002/api"
  
```

Hình 3.66. Tích hợp chatbot lên website 17 (nguồn: duythanhcse.wordpress.com)

Bây giờ tạo một trang index.html, trang này để tích hợp AI RASA vào để làm tool chatbot:

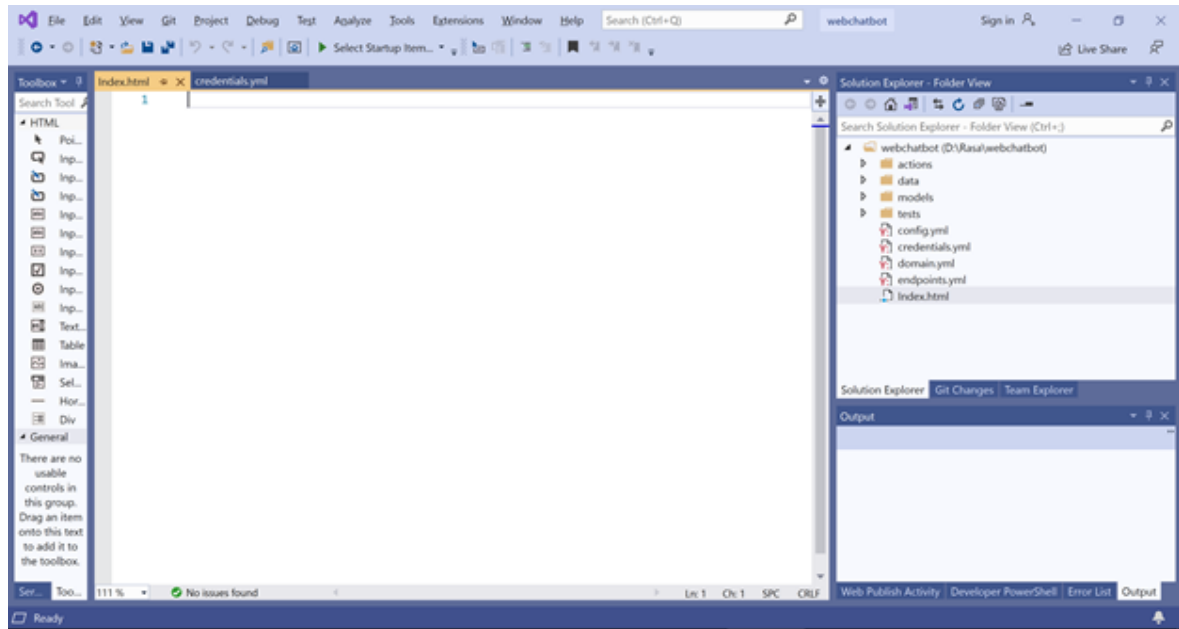
Bấm chuột phải vào dự án / chọn Add/ chọn New File:



Hình 3.67. Tích hợp chatbot lên website 18 (nguồn: duythanhcse.wordpress.com)

Lúc này trong mục Solution sẽ ra 1 tên file cho ta, ta sử dụng nó thành: **Index.html**

xem hình:



Hình 3.68. Tích hợp chatbot lên website 19 (nguồn: duythanhcse.wordpress.com)

Bây giờ ta code HTML, Javascript cho Index.html như sau:

```
<div id="webchat" />

!(function () {
    let e = document.createElement("script"),
        t = document.head || document.getElementsByTagName("head")[0];
    (e.src =
        "https://cdn.jsdelivr.net/npm/rasa-webchat@1.x.x/lib/index.js"),
        // Replace 1.x.x with the version that you want
        (e.async = !0),
        (e.onload = () => {
            window.WebChat.default(
                {
                    initPayload: '/greet',
                    customData: { language: "en" },
                    socketUrl: "http://localhost:5005",
                    // add other props here
                },
                null
            );
        }),
        t.insertBefore(e, t.firstChild);
})();
```

Hình 3.69. Tích hợp chatbot lên website 20 (nguồn: duythanhcse.wordpress.com)

ở trên:

```
1 | socketUrl: "http://localhost:5005"
```

Nghĩa là ta muốn start 1 con AI RASA server ở cổng 5005

Bây giờ ta quay lại Anaconda gõ lệnh sau để start con AI RASA server này:

```
1 | rasa run -m models --enable-api --cors "*"
```

Xem hình:

```
Anaconda Prompt (Anaconda3)
Processed trackers: 100%| 2/2 [00:00<00:00, 1993.49it/s]
Processed trackers: 100%| 5/5 [00:00<00:00, 1664.54it/s]
Processed trackers: 100%| 120/120 [00:00<00:00, 3881.58it/s, #
Epochs: 100%| 100/100 [00:10<00:00, 9.63it/s, t_loss=0.134, lo
Processed trackers: 100%| 120/120 [00:00<00:00, 1874.84it/s, #
Epochs: 100%| 100/100 [00:11<00:00, 8.36it/s, t_loss=2.31, los
2021-08-24 19:25:29 INFO rasa.core.agent - Persisted model to 'C:\Users\admin\AppData\Local\Temp\tmpxs2
0yvy\core'
Core model training completed.
Your Rasa model is trained and saved at 'D:\Rasa\webchatbot\models\20210824-192411.tar.gz'.
? Do you want to speak to the trained assistant on the command li
ne? [Y] No
Ok [Y] . If you want to speak to the assistant, run 'rasa shell' at any time inside the project directory.
(rasa_install_demo) D:\Rasa\webchatbot>rasa run -m models --enable-api --cors "*"

```

Hình 3.70. Tích hợp chatbot lên website 21 (nguồn: duythanhcse.wordpress.com)

Chờ nó chạy và xem kết quả:

```

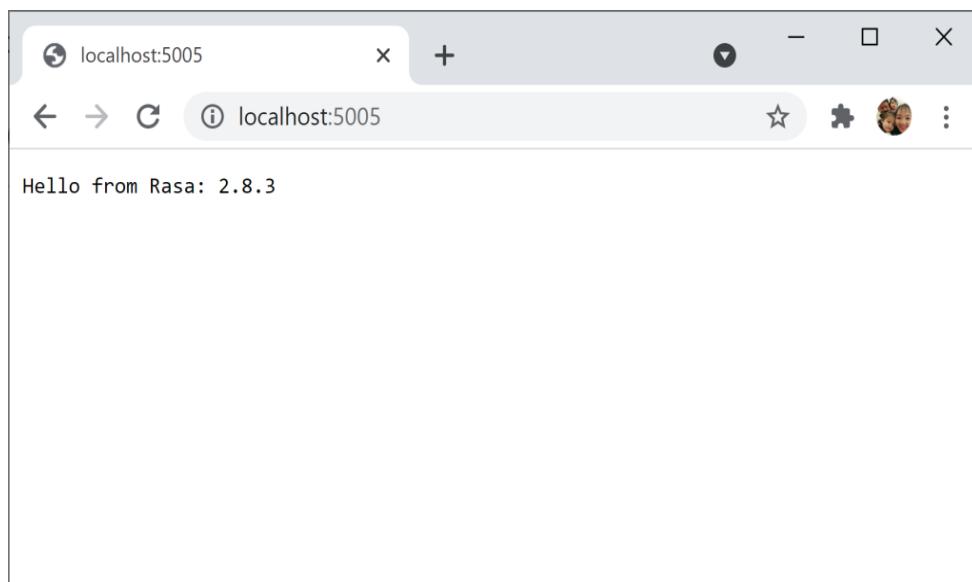
Anaconda Prompt (Anaconda3) - rasa run -m models --enable-api --cors ""
2021-08-24 19:25:29 INFO     rasa.core.agent - Persisted model to 'C:\Users\admin\AppData\Local\Temp\tmpxsn2
0yyv\core'
Core model training completed.
Your Rasa model is trained and saved at 'D:\Rasa\webchatbot\models\20210824-192411.tar.gz'.
? Do you want to speak to the trained assistant on the command li
ne? [Y/n] No
Ok [Y/n] . If you want to speak to the assistant, run 'rasa shell' at any time inside the project directory.

(rasa_install_demo) D:\Rasa\webchatbot>rasa run -m models --enable-api --cors ""
2021-08-24 20:09:30 INFO     root - Starting Rasa server on http://localhost:5005
2021-08-24 20:09:30 INFO     rasa.model - Loading model models\20210824-192411.tar.gz...
2021-08-24 20:09:51 WARNING  rasa.shared.utils.common - The Unexpected Intent Policy is currently experiment
al and might change or be removed in the future. Please share your feedback on it in the forum (https://for
um.rasa.com) to help us make this feature ready for production.
2021-08-24 20:09:59 INFO     root - Rasa server is up and running.
  
```

Hình 3.71. Tích hợp chatbot lên website 22 (nguồn: duythanhcse.wordpress.com)

Như vậy nó đã start RASA server tại <http://localhost:5005>

Ta có thể kiểm tra :

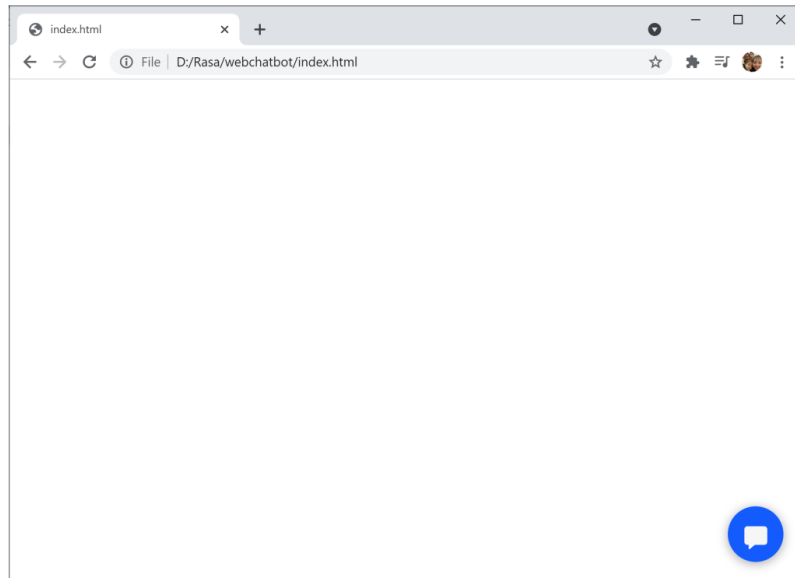


Hình 3.72. Tích hợp chatbot lên website 23 (nguồn: duythanhcse.wordpress.com)

Như vậy sau khi kiểm tra trên trình duyệt thì rõ ràng con Server AI Rasa đã chạy thành công.

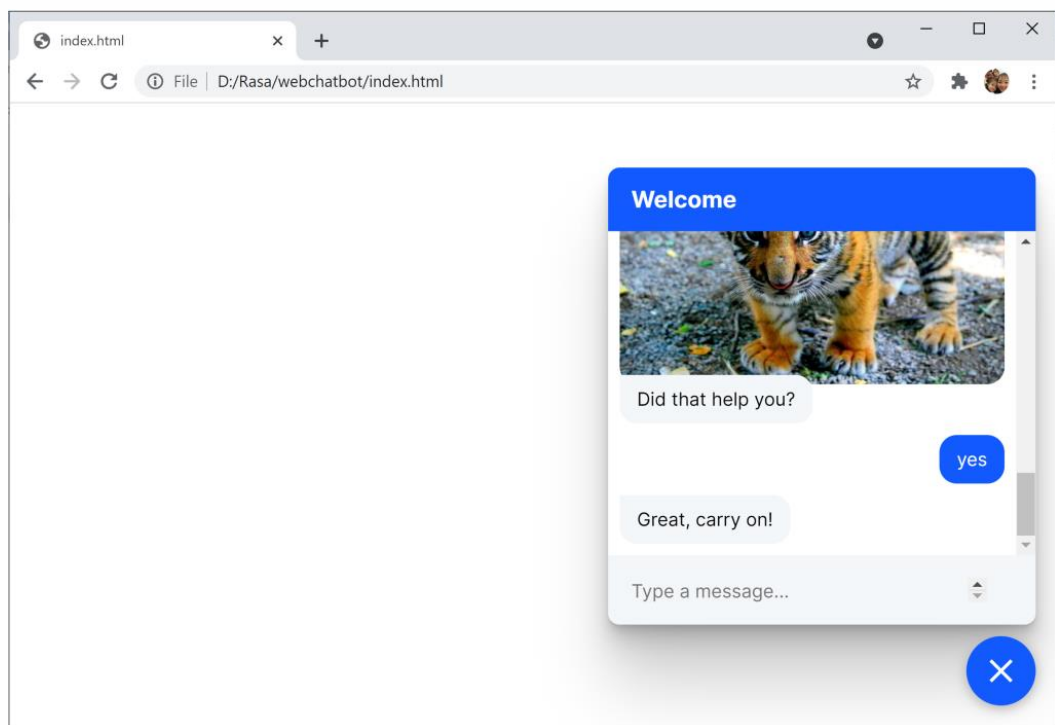
Để chạy Web Index.html thử nghiệm Chatbot ta:

Rồi gõ đường dẫn: <D:/Rasa/webchatbot/index.html> Vào trình duyệt, xem kết quả:



Hình 3.73. Tích hợp chatbot lên website 24 (nguồn: duythanhcse.wordpress.com)

Góc phải dưới cùng nhấn vào biểu tượng chat rồi thử nghiệm:



Hình 3.74. Tích hợp chatbot lên website 25 (nguồn: duythanhcse.wordpress.com)

KẾT LUẬN

Kết quả đạt được

Hiện tại chatbot đã thực hiện được các chức năng cơ bản đáp ứng nhu cầu người dùng. Em đã xây dựng được chatbot dựa trên nền tảng mã nguồn mở Rasa. Với một số chức năng cơ bản như:

- Hiểu ngôn ngữ tự nhiên (NLU)
- Tương tác tự nhiên
- Quản lý hội thoại
- Tích hợp với nền tảng khác

Hạn chế

Hiện tại chatbot còn một số hạn chế sau:

- Xử lý ngôn ngữ chưa chặt chẽ khiến đôi khi bot trả lời sai kịch bản.
- Chưa tối ưu tốc độ xử lý.
- Giao diện ứng dụng chưa được thiết kế kỹ càng còn khá đơn giản.
- Chưa đánh giá được độ chịu tải của hệ thống.
- Chưa xây dựng cơ sở dữ liệu lưu trữ tài liệu.

Hướng phát triển

Để phát triển chatbot cần thực hiện các công việc sau:

- Tối ưu tốc độ xử lý.
 - Thiết kế giao diện thân thiện đẹp mắt hơn.
 - Đánh giá độ chịu tải của hệ thống
 - Thêm chức năng gửi hình ảnh, âm thanh, video, tài liệu.
 - Dựa vào lượng dữ liệu câu hỏi và câu trả lời ngày càng lớn đến một mức độ có thể sử dụng dữ liệu đào tạo được một con Chatbot thông minh sau này.
-

TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1]. *Nguyễn Quốc Huy*, Trí tuệ nhân tạo học máy và ứng dụng, Nhà xuất bản Thanh Niên, 2014

Tiếng Anh:

- [1]. *Rudolph Russell*, Machine Learning Step-by-Step Guide To Implement Machine Learning Algorithms with Python, 2018
- [2]. *Andriy Burkov*, The Hundred-Page Machine Learning Book

Danh mục các Website tham khảo:

- [1]. Rasa, *Rasa Open Source*. Truy cập ngày 24/02/2024 tại: <https://rasa.com/docs/rasa/>
- [2]. Trần Duy Thanh, *Tích hợp Trí Tuệ Nhân Tạo Rasa để làm Web Chatbot*. Truy cập ngày 18/03/2024 tại: <https://duythanhcse.wordpress.com/2021/08/24/tich-hop-tri-tue-nhan-cao-rasa-de-lam-web-chatbot/>
- [3]. *Xử lý ngôn ngữ tự nhiên (NLP) là gì?* Truy cập ngày 10/02/2024 tại: <https://aws.amazon.com/vi/what-is/nlp/>
-