

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

BÁO CÁO TỔNG KẾT

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM 2021**

**NGHIÊN CỨU RASA FRAMEWORK ỨNG DỤNG
CHATBOT TƯ VẤN SẢN PHẨM DINH DƯỠNG**

Sinh viên thực hiện

Nguyễn Văn An	Lớp: CNTT 1 – k59	Khoa: Công Nghệ Thông Tin
Nguyễn Đức Phú	Lớp: CNTT 1 – k59	Khoa: Công Nghệ Thông Tin
Lê Quang Thọ	Lớp: CNTT 1 – k59	Khoa: Công Nghệ Thông Tin

Người hướng dẫn: ThS.Nguyễn Thu Hường

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN

NĂM 2021

**NGHIÊN CỨU RASA FRAMEWORK ỨNG DỤNG
CHATBOT TƯ VẤN SẢN PHẨM DINH DƯỠNG**

Sinh viên thực hiện

Nguyễn Văn An Nam, Nữ: Nam/Nữ Dân tộc: Kinh

Lớp: CNTT 1-k59 Khoa: Công Nghệ Thông Tin Năm thứ:3/4

Ngành học:Công Nghệ Thông Tin

Nguyễn Đức Phú Nam, Nữ: Nam/Nữ Dân tộc: Kinh

Lớp: CNTT 1-k59 Khoa: Công Nghệ Thông Tin Năm thứ:3/4

Ngành học:Công Nghệ Thông Tin

Lê Quang Thọ Nam, Nữ: Nam/Nữ Dân tộc: Kinh

Lớp: CNTT 1-k59 Khoa: Công Nghệ Thông Tin Năm thứ:3/4

Ngành học:Công Nghệ Thông Tin

Người hướng dẫn: ThS.Nguyễn Thu Hường

MỤC LỤC

MỞ ĐẦU.....	4
1. Lý do lựa chọn đề tài	4
2. Mục tiêu đề tài	4
3. Cách tiếp cận, phương pháp nghiên cứu	4
4. Đối tượng và phạm vi nghiên cứu.	4
CHƯƠNG 1 : TỔNG QUAN CÁC HỆ THỐNG CHATBOT	5
1.1. Giới thiệu	5
1.2. Các mô hình chatbot bán hàng tiêu biểu hỗ trợ Tiếng Việt hiện nay	6
1.2.1. Chatbot theo kịch bản (menu/button).	6
1.2.2. Chatbot nhận dạng từ khóa.	7
1.2.3. Mô hình Chatbot bán hàng theo ngữ cảnh.....	8
CHƯƠNG 2: TỔNG QUAN VỀ CHƯƠNG TRÌNH RASA	9
2.1. Giới thiệu Rasa là gì.....	9
2.2. Ưu điểm của Rasa.....	9
2.3 Các thành phần của Rasa.	9
2.3.1. Rasa NLU.	10
2.3.2. Rasa Core.....	12
CHƯƠNG 3: XÂY DỰNG CHATBOT TƯ VẤN SẢN PHẨM DINH DƯỠNG	15
3.1. Phát biểu bài toán.	15
3.2. Phân tích nghiệp vụ tư vấn và bán sản phẩm của Chatbot.	16
3.3. Xây dựng bộ dữ liệu Chatbot	17
3.3.1. Intent.....	18
3.3.2. Entities.	20
3.3.3. Slot.....	21
3.3.4. Actions.....	22
3.3.5. Forms.	24
3.3.6. Responses.	25
3.3.7. Stories.	26
3.4. Thử nghiệm.....	27
3.5. Đánh giá.....	28
CHƯƠNG 4: KẾT LUẬN.....	29
4.1. Kết quả đạt được.	29
4.2. Hạn chế.....	29
4.3. Hướng phát triển.....	29
TÀI LIỆU THAM KHẢO.....	30
PHỤ LỤC	31

THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

1. Thông tin chung:

- Tên đề tài: Nghiên cứu Rasa Framework ứng dụng Chatbot tư vấn sản phẩm dinh dưỡng

- Sinh viên thực hiện:

Nguyễn Văn An

Lớp:CNTT 1-k59

Khoa: Công Nghệ Thông Tin

Năm thứ:3/4

Nguyễn Đức Phú

Lớp:CNTT 1-k59

Khoa: Công Nghệ Thông Tin

Năm thứ:3/4

Lê Quang Thọ

Lớp:CNTT 1-k59

Khoa: Công Nghệ Thông Tin

Năm thứ:3/4

- Người hướng dẫn: ThS.Nguyễn Thu Hường

2. Mục tiêu đề tài:

- Nâng cao khả năng lập trình của sinh viên.
- Nắm được các kỹ thuật và cấu trúc của Rasa framework.
- Ứng dụng xây dựng ChatBot trên Facebook Messenger hỗ trợ cho việc tư vấn và đặt hàng về sản phẩm bánh dinh dưỡng Hebi và men vi khuẩn sống Việt Nhật.

3. Tính mới và sáng tạo:

- Tìm hiểu về Chatbot
- Nghiên cứu về Rasa framework
- Xây dựng Chatbot tự động tương tác với khách hàng trên Facebook

Messenger hỗ trợ người dùng trong việc bán hàng và tư vấn sản phẩm dựa vào Rasa framework.

4. Kết quả nghiên cứu:

Nhóm chúng em đã tổng quan về một hệ thống Chatbot, các mô hình Chatbot bán hàng hiện nay. Xây dựng được Chatbot trên Facebook Messenger dựa trên Rasa framework với các chức năng như: Tư vấn về sản phẩm cụ thể, tư vấn sản phẩm với từng lứa tuổi, hỏi thông tin của khách hàng,...

5. Đóng góp về mặt kinh tế - xã hội, giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài:

Rasa framework có tính ứng dụng cao trong việc không chỉ dừng lại ở lĩnh vực tư vấn sản phẩm, bán hàng mà Rasa có thể áp dụng vào các lĩnh vực khác như giải trí, thời tiết, hoạt động xã hội, du lịch & khách sạn, y tế, hàng không,...

6. Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài (*ghi rõ họ tên tác giả, nhan đề và các yếu tố về xuất bản nếu có*) hoặc nhận xét, đánh giá của cơ sở đã áp dụng các kết quả nghiên cứu (*nếu có*):

.....
.....
.....

Ngày tháng năm

Sinh viên chịu trách nhiệm chính

thực hiện đề tài

(ký, họ và tên)

Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài (*phần này do người hướng dẫn ghi*):

.....
.....
.....
.....
.....

Ngày tháng năm

Người hướng dẫn

(ký, họ và tên)

MỞ ĐẦU

Ngày nay, cùng với sự phát triển của công nghệ, Chatbot đang được sử dụng mạnh mẽ trong và phổ biến trong nhiều lĩnh vực khác nhau như: Chăm sóc khách hàng, mua sắm trực tuyến, trả lời thông tin thời tiết, tin tức, xã hội,... Chatbot có thể thay thế con người trong việc trả lời câu hỏi của các khách hàng khác nhau có khả năng lặp đi lặp lại nhiều hoặc là ý định giống nhau giữa các khách hàng, hỗ trợ khách hàng 24/24.

1. Lý do lựa chọn đề tài

Hiện nay việc trả lời các câu hỏi của bộ phận tư vấn & chăm sóc khách hàng qua tin nhắn đang được ưa chuộng. Việc này còn thực hiện một cách thủ công và gặp nhiều khó khăn như: tốn rất nhiều thời gian và chi phí chi trả cho nhân viên chỉ để trả lời những câu hỏi đơn giản và có mục đích giống nhau. Chính vì vậy Chatbot là một sự lựa chọn hoàn hảo thay thế cho một phần công việc của nhân viên và có thể nhắn tin với nhiều khách hàng cùng lúc.

2. Mục tiêu đề tài

Mục tiêu của chúng em tập chung vào xây dựng ứng dụng Chatboot bằng Rasa framework để tư vấn sản phẩm dinh dưỡng là Bánh Hebi của viện dinh dưỡng Quốc Gia và men vi khuẩn sống Việt Nhật.

3. Cách tiếp cận, phương pháp nghiên cứu

Cách tiếp cận là sử dụng Rasa framework và áp dụng những kiến thức nền tảng để có thể làm chủ và tùy chỉnh mã nguồn mở này sẽ tập trung một hệ thống Chatbot hỗ trợ người dùng.

4. Đối tượng và phạm vi nghiên cứu.

Đối tượng nghiên cứu là Rasa framework và khái quát về Chatbot. Phạm vi nghiên cứu là ứng dụng hệ thống Chatbot trên Facebook và được tư vấn về sản phẩm dinh dưỡng cho khách hàng.

CHƯƠNG 1 : TỔNG QUAN CÁC HỆ THỐNG CHATBOT

Chương này sẽ giới thiệu những khái niệm cơ bản, một số hệ thống và các mô hình về Chatbot hiện nay đang phổ biến và những vấn đề gặp phải khi xây dựng Chatbot.

1.1. Giới thiệu

Theo Wikipedia thì Chatbot là một phần mềm ứng dụng dùng để thực hiện một cuộc trò chuyện trực tuyến thông qua một đoạn văn bản hoặc chuyển văn bản thành giọng nói, thay vì cung cấp liên hệ trực tiếp với một nhân viên có thật. Do được thiết kế để mô phỏng cách một người cư xử như một đối tác thảo luận như thật, các hệ thống chatbot thường phải điều chỉnh và thử nghiệm liên tục, và nhiều ứng dụng thực tế vẫn không đạt đủ tiêu chuẩn vượt qua bài kiểm tra Turing. Thuật ngữ "ChatterBot" có nguồn gốc từ Michael Loren Mauldin vào năm 1994 để mô tả các chương trình đối thoại này.

Hiện nay Chatbot thường được sử dụng trong các hệ thống trên ứng dụng tin nhắn nhằm giao tiếp với người dùng. Khả năng xử lý các tình huống, tính toán được xác định bằng thuật toán do con người tạo ra. Chatbot được ứng dụng rất nhiều trong bán hàng, tư vấn và chăm sóc khách hàng, giải trí, giáo dục, ...vv

Chatbot có thể chia thành 3 loại:

- **Chatbot dựa theo quy tắc (Rule-Based chatbots):** Loại Chatbot này khả năng rất hạn chế. Chỉ có khả năng phản hồi chính xác những lệnh cụ thể mà ta đã quy định từ trước hoặc người dùng không được phép tùy trình phản hồi mà phải lựa chọn các câu trả lời hoặc ý kiến được lập trình ra từ trước đó. Sự thông minh và linh hoạt của loại này không cao phải phụ thuộc vào người lập trình ra Chatbot.

- **Chatbot độc lập về mặt trí tuệ (Intellectually-Independent chatbots):** Các chatbot độc lập về mặt trí tuệ là các bot dựa trên việc máy học (machine learning). Những Chatbot này được tạo ra dựa vào việc đào tạo một mạng nơ-ron tự "suy nghĩ" bằng cách cung cấp cho nó hàng nghìn hoặc đôi khi hàng triệu ví dụ về những gì bot cần để có thể hiểu được. Chúng tự cải thiện theo thời gian và hầu hết được sử dụng rộng rãi cho mục đích giải trí và khoa học.

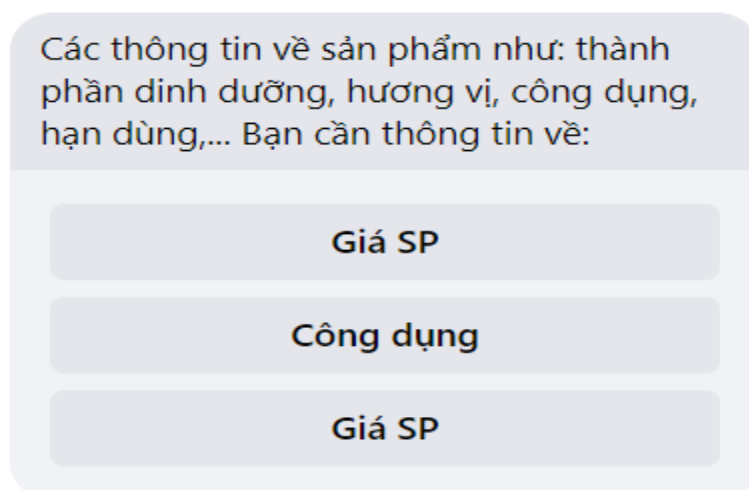
– **Được hỗ trợ bởi AI:** Loại này có khả năng hiểu được ngôn ngữ. Nghĩa là Chatbot không bị giới hạn bởi các quy định được xác định từ trước mà có thể hiểu ở phạm vi rộng hơn. Tất cả các Chatbot này vẫn hỏi học các dữ liệu được cung cấp sẵn từ trước đó, nhưng nó có khả năng đoán được ý định và ngữ cảnh của cuộc trò chuyện với người dùng. Chatbot được hỗ trợ bởi AI còn có khả năng học thêm, nghĩa là ta đưa càng nhiều câu chuyện với ngữ cảnh khác nhau thì bot có thể ghi nhớ ngữ cảnh của một cuộc trò chuyện và sở thích của người dùng. Từ đó xen kẽ từ cuộc trò chuyện này sang cuộc trò chuyện khác, xác suất Chatbot phản hồi lại người dùng chính xác càng cao.

1.2. Các mô hình chatbot bán hàng tiêu biểu hỗ trợ Tiếng Việt hiện nay

1.2.1. Chatbot theo kịch bản (menu/button).

Menu/Button là dạng chatbot cơ bản nhất là các hệ thống phân cấp cây và trình bày cho khách hàng dưới dạng các nút (button). Loại Chatbot này hoạt động thông qua dữ liệu được lập trình sẵn, đây là loại được sử dụng phổ biến nhất hiện nay. Khi khách truy cập đặt câu hỏi, bot sẽ đưa ra các tùy chọn liên quan. Khách hàng sẽ nhấp chọn một mục tương ứng có thể đáp ứng mục đích tìm kiếm. Sau đó, bot sẽ đưa ra câu trả lời liên quan nhất với thông tin khách hàng vừa nhấp vào.

Tuy nhiên, đôi khi người dùng sẽ phải trả lời nhiều câu hỏi, nhấp nhiều tùy chọn mới có thể tìm được chính xác những gì mình cần. Yêu cầu của khách hàng thường sẽ được giải đáp khá chậm. Đối với một số câu hỏi không được lập trình sẵn, Robot có thể không trả lời được, hoặc giải đáp thiếu chính xác. Sau đây là một số ví dụ về Chatbot này.



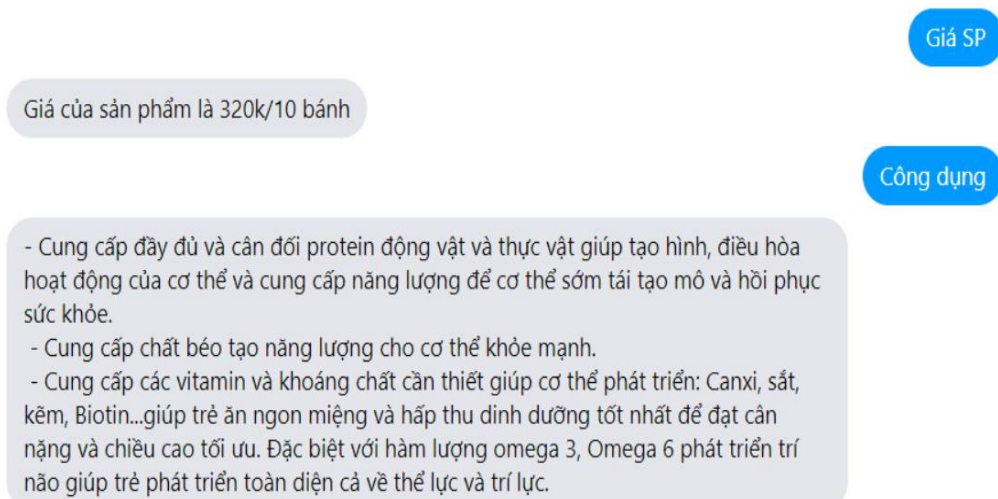
Hình 1: Ví dụ về dạng Chatbot (menu/button).

Ở ví dụ hình 1 Chatbot đã xây dựng sẵn ba button để tương tác với khách hàng với ba yêu cầu khác nhau, chỉ cần chọn một button thì Chatbot sẽ đưa ra câu trả lời tương ứng với button đấy.

- Ưu điểm của Chatbot này dễ dàng xây dựng, độ chính xác cao với từng yêu cầu mà khách hàng đưa ra mà đã được xây dựng từ trước.
- Nhược điểm là khách hàng sẽ không được chủ động với những lựa chọn mà mình đề cập thêm, mà phải phụ thuộc vào menu/button của Chatbot đã được xây dựng từ trước không tùy biến được theo ý khách hàng.

1.2.2. Chatbot nhận dạng từ khóa.

Khác với các Chatbot dạng menu/button, loại này dùng học máy (Machine Learning) để xử lý linh hoạt các truy vấn của người dùng. Chatbot này dựa trên nhận dạng từ khóa hoặc cụm từ có liên quan và đã được huấn luyện từ trước để trả lời các câu hỏi nhất định. Nhờ đó bot có thể hiểu được mục đích khách hàng khi gặp phải những câu hỏi có chứa từ khóa đã được huấn luyện sẵn và trả lời câu hỏi phù hợp nhất. Sau đây là ví dụ về loại Chatbot này:



Hình 2: Ví dụ về Chatbot theo keywords.

Trong ví dụ trên người ta hỏi “Giá của Bánh Hebi ạ” thì bot sẽ hiểu được là khách hàng đang xin giá phải trả lời về là “Giá của sản phẩm là 320k/10 bánh”. Tương tự khi khách hàng hỏi công dụng cũng vậy bot sẽ trả về các công dụng.

Ưu điểm của mô hình Chatbot này là không đưa ra những tùy chọn có sẵn như của menu/button giúp người dùng chủ động hơn trong việc đưa ra yêu cầu, làm cho cuộc trò chuyện tự nhiên hơn.

Nhược điểm là nó vẫn còn khá nhiều như khi người dùng sử dụng các từ đồng nghĩa với các từ khoá thì Chatbot không thể phát hiện được để trả lời phù hợp, và không thể nắm bắt được ngữ cảnh cuộc trò chuyện.

1.2.3. Mô hình Chatbot bán hàng theo ngữ cảnh

Những phân tích trên về ưu nhược điểm của 2 mô hình Chatbot dựa trên menu/button và nhận dạng từ khoá. Xây dựng Chatbot dựa trên Machine Learning (ML) và Trí tuệ nhân tạo (AI). Nó có thể hoạt động dựa trên việc ghi nhớ lại sở thích, bối cảnh trò chuyện với khách hàng hoặc truy cập từ các cuộc trò chuyện khác trước đó, Điều này cho phép bot đưa ra phản hồi phù hợp nhất với khách hàng.

Ví dụ:

Với mô hình này thì mô hình này thì Chatbot thông minh hơn, có thể hiểu được các ý định và có thể trích chọn được các thông tin từ yêu cầu của khách hàng, lưu được ngữ cảnh cuộc trò chuyện và đưa ra câu trả lời phù hợp nhất, giúp cho trải nghiệm của khách hàng được tự nhiên hơn

CHƯƠNG 2: TỔNG QUAN VỀ CHƯƠNG TRÌNH RASA

2.1. Giới thiệu Rasa là gì.

Rasa framework là một công cụ Machine Learning mã nguồn mở để phát triển các chatbot, được hỗ trợ bởi AI để cho các cuộc trò chuyện dựa trên văn bản và giọng nói tự động. Hiểu tin nhắn giữa cuộc trò chuyện theo ngữ cảnh và kết nối với các kênh nhắn tin và API.

Trong bài toán này, nhóm chúng em sẽ tìm hiểu một số khía cạnh cơ bản nhất trong Rasa và phát triển chatbot. Sau khi hoàn thành bài toán này, nhóm có thể tạo Chatbot đơn giản được hỗ trợ bởi AI của riêng mình để hỗ trợ cho việc tư vấn sản phẩm dinh dưỡng là Bánh dinh dưỡng Hebi.

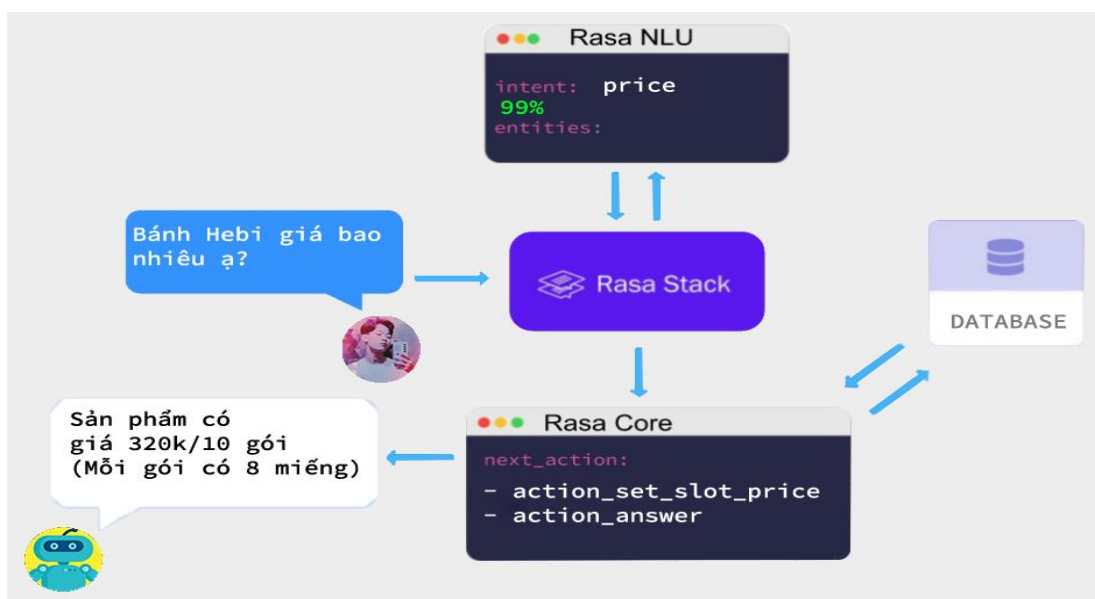
2.2. Ưu điểm của Rasa.

Rasa framework thực sự dễ tiếp cận cho người mới bắt đầu ngay cả khi bạn chưa biết gì về Chatbot cũng có thể sử dụng và làm quen. Hầu hết công việc của người dùng là tập trung đi xây dựng nội dung kịch bản cũng có thể tạo ra được một con Bot đơn giản.

Rasa framework hoạt động khá mạnh mẽ, đặc biệt là trong vấn đề xác định ý định (intent) của người dùng và xác định thực thể (entity). Mã nguồn của Rasa là mã nguồn mở, do đó Rasa giúp ta biết chính xác mình đang làm gì với Chatbot của mình thậm chí có thể tùy biến theo ý thích của bản thân.

2.3 Các thành phần của Rasa.

Rasa bao gồm 2 thành phần chính: Rasa NLU và Rasa Core



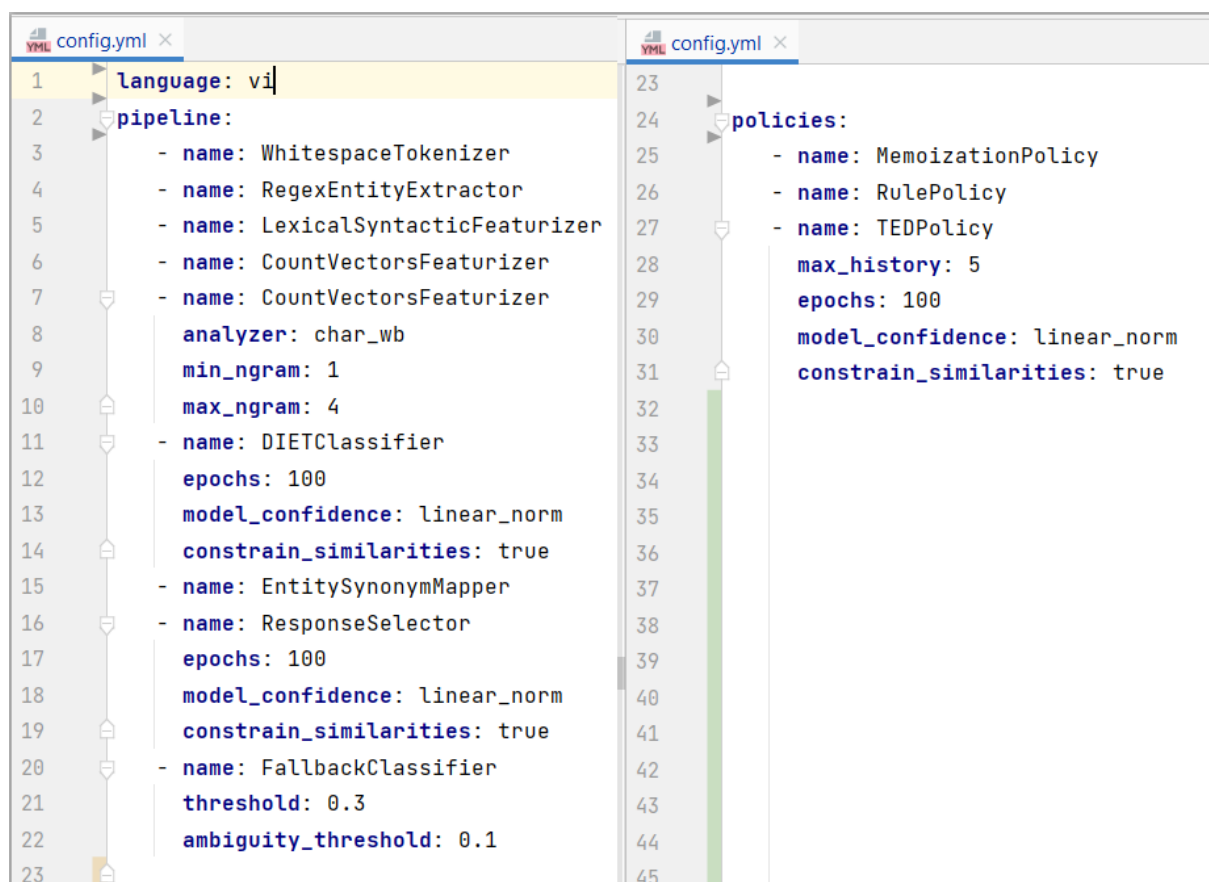
Hình 3: Cách hoạt động của 1 luồng.

Rasa NLU: (Natural Language Understanding) một thư viện để hiểu ngôn ngữ tự nhiên (NLU) thực hiện phân loại ý định và trích xuất thực thể từ đầu vào của người dùng và giúp bot hiểu những gì người dùng đang nói.

Rasa Core là nơi thực hiện quản lý luồng hội thoại. Dựa vào các intent, entity đã được tìm ra ở phần NLU, Rasa Core tiến hành lấy các kết quả này làm đầu vào, rồi quyết định message đầu ra.

2.3.1. Rasa NLU.

File config.yml đây là nơi ta cấu hình cho NLU, chúng ta có thể chọn ngôn ngữ, model cần thiết.



Hình 4: File config.yml để cấu hình Rasa NLU.

Ở hình 4 là ta chọn ngôn ngữ là vi (tiếng việt), tiếp là pipeline là một quy trình lựa chọn Tokenizer, Featurizer, Extractor đến Classifier. Trong một dự án Rasa, pipeline NLU xác định các bước xử lý để chuyển đổi tin nhắn người dùng thành ý định và thực thể. Nó bao gồm một loạt các thành phần, có thể được cấu hình và tùy chỉnh bởi người lập trình.

```

1  version: 2.0
2  nlu:
3    - intent: greet
4      examples: - xin chào- Xin chào
22   - intent: bye
23     examples: - bye - g...iệt em nhé
31   - intent: confirm
32     examples: - Đồng ý ... Đồng ý ạ!
52   - intent: unconfirm
53     examples: - Hủy đơn...Hủy đơn ạ!
72   - intent: thanks
73     examples: - thanks ... nhiều nhé
82   - intent: composition
83     examples: - thành p... phần nào?
86   - intent: taste
87     examples: - hương v... có vị gì?
92   - intent: effects
93     examples: |
94       - tác dụng chính
95       - có lợi
96       - tốt cho
97       - công dụng
98       - công dụng của sản phẩm là gì vậy?
99   - intent: contraindications
100  examples: |
101    - lưu ý
102    - chống chỉ định
103    - không dùng cho
104    - ko dùng cho
105    - có lưu ý gì khi dùng sản phẩm không?
106  - intent: user_manual
107  examples: |
108    - liều dùng
109    - dùng làm sao
110    - cách dùng
111    - hướng dẫn sử dụng
112    - có thể dùng theo những cách nào
113    - dùng sản phẩm này thế nào ạ
114    - bé nhà mình [3](age) tuổi thì dùng bao nhiêu
115  - intent: storage
116  examples: |
117    - hạn dùng
118    - hạn sử dụng
119    - sản phẩm có thể bảo quản trong bao lâu?
120  - intent: made_in
121  examples: |

```

Hình 5: File nlu.yml xây dựng ý định khách hàng.

Hình 5 là file nlu.yml là nơi chúng ta cần thêm data để train model, file nlu.yml sẽ đảm nhận phần data đó. Ở đây ta xây dựng các intent (ý định) tương ứng với keywords. Khi khách hàng chat những câu tương tự như vậy hoặc trong câu có các keywords đã được định nghĩa trong intent thì bot sẽ hiểu được ý định của khách hàng. Dưới đây là tin nhắn mà khách hàng hỏi đã được gắn nhãn và trích xuất intent tương ứng:

```

Next message:
cho mình đặt hàng ạ
{
  "text": "cho mình đặt hàng ạ",
  "intent": {
    "id": 2991736532844968976,
    "name": "order",
    "confidence": 0.6172868013381958
  },
  "entities": [],
  "intent_ranking": [
    {
      "id": 2991736532844968976,
      "name": "order",
      "confidence": 0.6172868013381958
    },
    {
      "id": 4157844525059155904,
      "name": "year_old",
      "confidence": 0.17465445399284363
    }
  ]
}

```

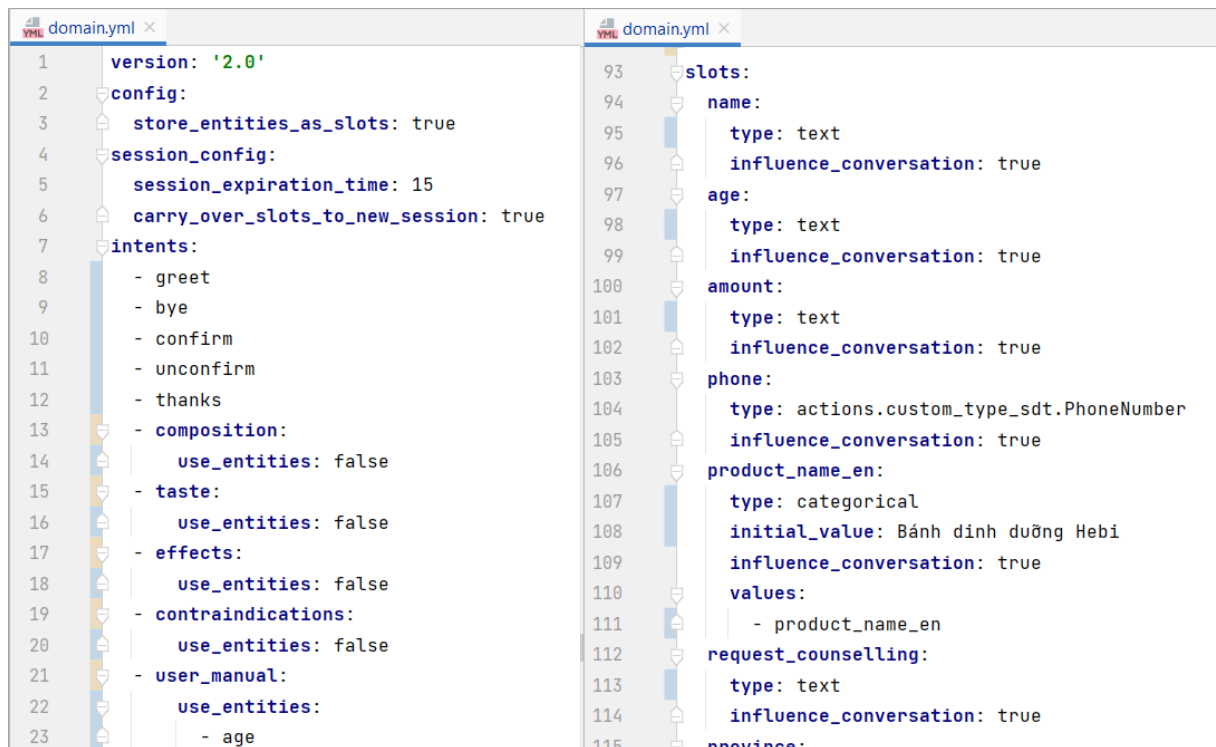
2.3.2. Rasa Core.

Rasa Core nơi tiếp nhận thông tin từ khách hàng rồi xử lý cuộc hội thoại, dựa vào các intent và entities được phát hiện và phân tích ở phần NLU, Rasa Core lấy các kết quả này làm đầu vào rồi quyết định kết quả đầu ra phù hợp để phản hồi với khách.

```
# Configuration for Rasa Core.  
# https://rasa.com/docs/rasa/core/policies/  
policies:  
- name: MemoizationPolicy  
- name: TEDPolicy  
  max_history: 5  
  epochs: 100  
  constrain_similarities: true  
- name: RulePolicy
```

Hình 6 : File config.yml để cấu hình Rasa Core.

Ở hình 6 là file config.yml, mình dùng một số policies như: MemoizationPolicy (quyết định message đầu ra dựa vào thông tin của đoạn hội thoại trước đó), TEDPolicy (một thuật toán của Rasa giúp dự đoán hành động tiếp theo của người dùng mà không cần dựa vào hành động trước đó), RulePolicy (đưa ra dự đoán giữa trên những quy tắc được đào tạo từ trước). Tiếp theo là đến file domain.



Hình 7: File domain.yml

Hình 7 trên là file domain.yml nơi khai báo các thông tin cần thiết:

- Intent là liệt kê các ý định (intent) của người dùng có trong file nlu.yml
- Entities là các thực thể thông tin đặc trưng quan trọng được trích xuất theo các ý định khách hàng.
- Slots là bộ nhớ bot của bạn. Chúng hoạt động như một kho lưu trữ key-value có thể được sử dụng để lưu trữ thông tin mà người dùng cung cấp
- Responses là phần chúng ta định nghĩa các message dạng text, hoặc hình ảnh, ... (các response này thường có dạng utter_{}).
- Actions là những thứ mà bot có thể làm được như: truy vấn cơ sở dữ liệu, trả lời người dùng, thực hiện lệnh gọi API bên ngoài, ...
- Forms là một loại action đặc biệt nhằm giúp bot của bạn thu thập thông tin từ người dùng. Xác định forms với các key trong file domain.yml.

Sau khi đã thiết lập và khai báo trong file domain thì tiếp theo chúng ta xây dựng một số kịch bản cần thiết cho việc train model.

```
stories:
- story: chao hoi 1
  steps:
  - intent: greet
  - action: utter_greet
  - intent: bye
  - action: utter_bye
- story: hoi gia
  steps:
  - intent: greet
  - action: utter_greet
  - intent: price
  - action: action_set_slot_price
  - action: action_answer
  - intent: bye
  - action: utter_bye
```

Hình 8: File stories.yml

File stories.md ta viết một số kịch bản có thể xảy ra khi tương tác với Bot, việc dự tính trước các trường hợp có thể xảy ra trong quá trình nhắn tin sẽ giúp bot xử lý các trường hợp một cách mượt hơn và có vẻ thông minh hơn. Vậy nên cần phải cố gắng tìm thật nhiều các trường hợp mà khách hàng có thể hỏi bot.

CHƯƠNG 3: XÂY DỰNG CHATBOT TƯ VẤN SẢN PHẨM DINH DƯỠNG

Trong chương này, chúng em tập chung vào xây dựng ứng dụng Chatboot để tư vấn sản phẩm dinh dưỡng là Bánh Hebi của viện ươm dưỡng Quốc Gia và men vi khuẩn sống Việt Nhật. Trong quá trình xây dựng và kiểm thử chúng em sẽ đưa ra được đánh giá về khả năng phục vụ của Chatboot, từ đó chúng em đưa ra hướng khắc phục nó....

Chương trình này sẽ mô tả từng bước xây dựng bài toán bằng mã nguồn mở Rasa Fframework. Kiểm tra và đánh giá sẽ cho ta biết khả năng phục vụ của Chatbot với khách hàng tới mức độ nào và tìm ra điểm hạn chế của Chatbot rồi khắc phục và cải tiến nó, tìm ra hướng đi tốt nhất nhằm phục vụ cho việc tư vấn sản phẩm cho khách hàng.

3.1. Phát biểu bài toán.

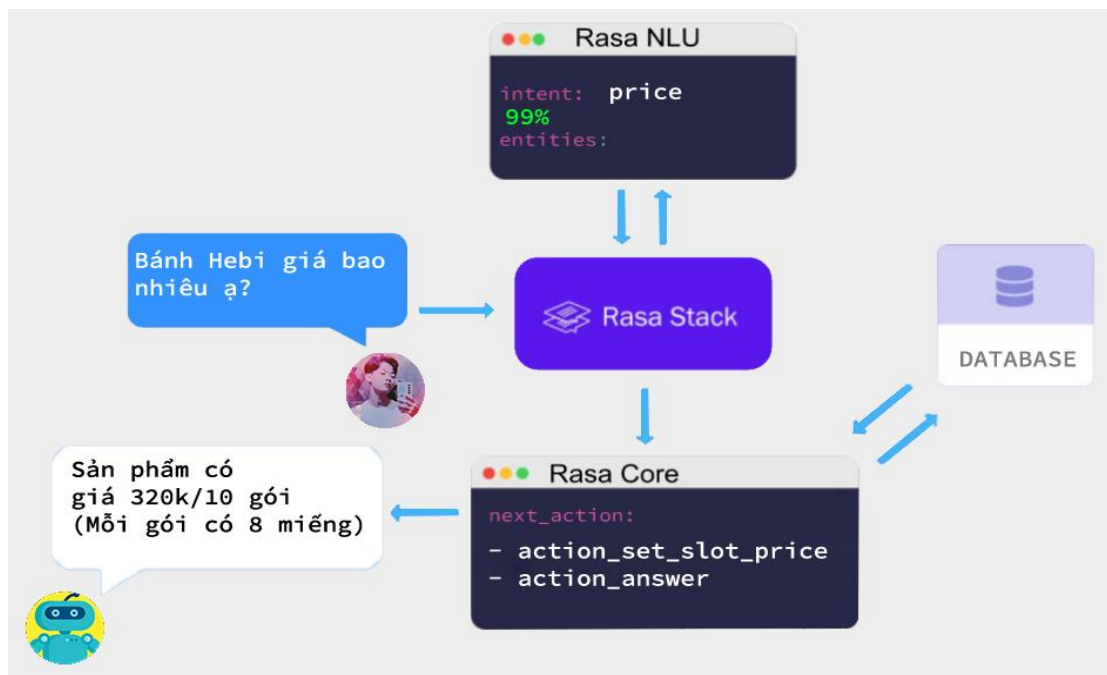
Chatbot được xây dựng lên với mục đích tư vấn, bán hàng, có thể gắn lên trên page bán hàng bất kỳ nào và bot sẽ thay thế nhân viên tương tác với khách hàng.

Bài toán đặt ra ở đây là ứng dụng Chatbot tập trung vào việc hỗ trợ và tư vấn sản phẩm dinh dưỡng cho khách hàng trên các Fanpage Facebook. Các tính năng chính bao gồm:

- Tư vấn thông tin của sản phẩm
 - + Tư vấn sử dụng sản phẩm đối với từng đối tượng
 - + Hỏi giá sản phẩm
 - + Xuất xứ và quy cách đóng của sản phẩm
 - + Tư vấn ship hàng
- Đặt hàng
 - + Trích xuất thông tin khách hàng
 - + Trích xuất thông tin đơn hàng
 - + Lưu hóa đơn

3.2. Phân tích nghiệp vụ tư vấn và bán sản phẩm của Chatbot.

Cấu trúc hệ thống Chatbot phục vụ cho tư vấn sản phẩm dinh dưỡng được thiết kế như sau:



Hình 9: Quá trình xử lý của Chatbot.

Mô hình trên là ví dụ về một luồng đầu vào của hệ thống là một câu hỏi của khách hàng, đầu ra là câu trả lời của Chatbot các thành phần sẽ có các từng chức năng riêng như sau.

Rasa NLU: Phân tích ý định của người dùng và trích xuất ra các thông tin. Ví dụ khách hàng hỏi “Cho tôi xin giá của sản phẩm ?” thì hệ thống sẽ thu thập dữ liệu và đối chiếu với các tập dữ liệu đã training đã được gán nhãn để đưa ra intent là “gia” và entities là none vì chưa có thực thể nào cần trích xuất thông tin vì đây là câu hỏi giá đơn giản thông thường

Rasa Core: Sau khi nhận được intent và entities từ khách hàng rồi. Tại đây Rasa Core sẽ xây dựng model tương ứng với intent của khách hàng và phản hồi lại khách hàng. Thì bot sẽ nhận biết next_action (hành động tiếp theo): utter_gia 100% là sẽ trả lời khách hàng liên quan đến giá “Giá của sản phẩm là 320/10 gói”.

3.3. Xây dựng bộ dữ liệu Chatbot

Nguồn dữ liệu được xây dựng Chatbot tư vấn thông qua việc train model và thu thập qua việc nhắn tin với khách hàng qua Fanpage Facebook.

Một đoạn tin nhắn giữa Chatbot và khách hàng, đây là cách Chatbot phản hồi lại yêu cầu của khách hàng.

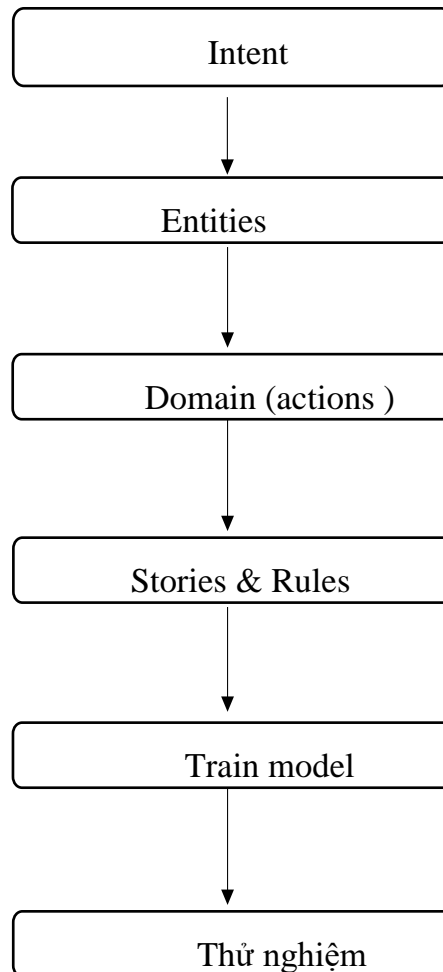
Khách Hàng	Chatbot
Xin chào	Chào bạn mình có thể giúp gì được cho bạn?
Cho mình xin giá của bánh Hebi	Giá của sản phẩm là 320k/10 bánh
Cho mình đặt hàng ạ.	Tôi là trợ lí ảo 🤖 của page Bánh Dinh Dưỡng HEBI, nguồn từ Rasa. tôi được tạo ra để có thể hỗ trợ cô chủ tôi nhận thông tin đặt hàng từ quý khách nếu có bất cứ điều gì sai sót mang quý khách thông cảm! Tên của bạn là gì?
Mình tên Thọ.	Cho mình xin số điện thoại của bạn ạ...
0973271289.	Cho mình xin địa chỉ giao hàng.
Mình ở số 3 đường Cầu Giấy, Láng Hạ, Đống Đa, Hà Nội	Đơn hàng bạn Thọ đã được lưu lại! Số điện thoại: 0973271289 Địa chỉ: số 3 đường Cầu Giấy, Láng Hạ, Đống Đa, Hà Nội Sản phẩm: Bánh dinh dưỡng Hebi Bạn sẽ nhận được cuộc gọi xác nhận trong vòng 24h Cảm ơn Thọ đã quan tâm đến sản phẩm của chúng tôi

Như vậy với việc xây dựng dữ liệu và train model cho Chatbot tư vấn sản nói riêng là ta phải đi xây dựng danh sách các khung kịch bản như ví dụ bên trên.

Ý định của khách hàng có thể diễn đạt bằng nhiều cách nói khác nhau nhưng chung một thông điệp, ứng với mỗi một câu hỏi, yêu cầu trong đoạn trò chuyện là một ý định của người dùng. Ví dụ người dùng có thể diễn đạt mục đích chào hỏi của mình bằng nhiều câu nói như “xin chào”, “chào bạn”, “chào shop”, “hi shop”,..vv.

Khi đã nắm được ý định của người dùng trong câu nói của họ qua cuộc trò chuyện thì ta cần phải trích xuất ra được những thông tin đặc trưng của người dùng. Đó chính là các thực thể (entity), các thực thể này sẽ được lưu lại vào các slot để lưu được ngữ cảnh tránh việc phải hỏi lại nhiều lần trong cuộc trò chuyện, việc này sẽ phục vụ cho các hành động sau của Chatbot.

Quy trình xây dựng hệ thống Chatbot sẽ thực hiện qua các bước chính như sau:



3.3.1. *Intent.*

Nhiệm vụ xây dựng các tập ý định sẽ theo nguyên tắc là những mẫu câu hỏi mà người dùng khi có ý định đó và hay sử dụng. Cần định nghĩa các ý định khớp với ngôn ngữ tự nhiên nhất, trong đoạn hội thoại trên nhóm định nghĩa các intent như sau:

- greet: ý định liên quan đến chào hỏi
- price: ý định liên quan đến yêu cầu tư vấn về giá sản phẩm
- order: ý định liên quan đến yêu cầu lập đơn
- customer_name: ý định liên quan đến cung cấp tên khách
- customer_phone_number: ý định cung cấp số điện thoại của khách
- customer_address: ý định cung cấp địa chỉ giao hàng của khách

Ví dụ cho một ý định muốn hỏi về giá sản phẩm thì khách hàng có thể hỏi nhiều câu như sau:

```
- intent: price
examples: |
- xin giá sản phẩm [Bánh dinh dưỡng Hebi](product_name_en)
- xin giá [Bánh dinh dưỡng Hebi](product_name_en)
- [Bánh dinh dưỡng Hebi](product_name_en) giá
- xin giá bán
- giá
- bán như thế nào?
- bán ra sao đây
- cho mk xin giá?
- Bánh bn 1 thanh vậy shop
- Bánh bao nhiêu 1 thanh vậy shop
- Bánh bao nhiêu 1 thanh
- Bánh bán theo hộp hay như như thế nào
- Bánh bán theo hộp hay như ntn
- cho mình xin giá sản phẩm
```

Hình 10: Mẫu câu trả lời của Chatbot cho ý định tư vấn.#

Từ đồng nghĩa chuẩn hóa dữ liệu đào tạo bằng cách ánh xạ một thực thể được trích xuất sang một giá trị khác với văn bản theo nghĩa đen được trích xuất. Ta có thể xác định các từ đồng nghĩa bằng cách sử dụng định dạng các định dạng khác.

```
- synonym: Bánh dinh dưỡng Hebi
examples: |
- Bánh Hebi
- Bánh dinh dưỡng
- Bánh
- cao năng lượng hebi
- synonym: Men vi khuẩn sống Việt Nhật
examples: |
- Men vi khuẩn sống
- Men Việt Nhật
- Men
```

Hình 11: Từ đồng nghĩa.

Ta có thể sử dụng biểu thức chính quy để cải thiện phân loại ý định và trích xuất thực thể.

```
- regex: age
examples: |
  - \d{1,2}[ ]*(tuổi|tháng|thg|T|th|Th)
  - \d{1,2}[ ]*(tuổi|tháng|thg|t)
  - \d{1,2}[ ]*(tuổi|tháng|thg|T|th)
- regex: amount
examples: |
  - \d{1,3}[ ]*(bánh|túi|gói|b|B|hộp|H|h|Túi)
  - \d{1,3}[ ]*(bánh|túi|gói|b|B|hộp|H)
  - \d{1,3}[ ]*(bánh|túi|gói|b|B|hộp|H|h)
- regex: phone
examples: |
  - \d{10,12}
  - \d{10,12}
  - \d{10,12}
```

Hình 12: Biểu thức chính quy

3.3.2. Entities.

Entity là các thực thể thông tin đặc trưng quan trọng được trích xuất theo các ý định khách hàng. Các thông tin được trích chọn trong câu nói của khách hàng được hệ thống lưu lại trong bộ nhớ hệ thống để dùng trong các hành động hoặc để đưa ra các câu trả lời phù hợp theo ngữ cảnh, tránh việc phải hỏi lại những thông tin mà khách hàng đã cung cấp từ trước.

```
entities:
- name
- phone
- age
- product_name_en
- amount
- province
- district
- ward
```

Hình 12: Danh sách các entities.

3.3.3. Slot.

Slots là bộ nhớ bot của bạn. Chúng hoạt động như một kho lưu trữ key-value có thể được sử dụng để lưu trữ thông tin mà người dùng cung cấp như:

```
slots:
  name:
    type: text
    influence_conversation: false
  age:
    type: text
    influence_conversation: false
```

Hình 13: Slot kiểu text

Slot kiểu text để lưu trữ các giá trị dạng văn bản. Nếu `influence_conversation` : `false` để lưu trữ thông tin mà không ảnh hưởng tới cuộc hội thoại.

```
slots:
  amount:
    type: float
    min_value: 1.0
    max_value: 999.0
    influence_conversation: false
```

Hình 14: Slot kiểu float

Slot kiểu float để lưu trữ dạng số thực, `min_value` và `max_value` là các giá trị tối thiểu và tối đa để lưu trữ dữ liệu.

```
slots:|
  phone:
    type: actions.custom_type_sdt.PhoneNumber
    influence_conversation: true
```

Hình 15: Slot kiểu custom

Slot kiểu custom để lưu trữ dạng số điện thoại, được gọi đến từ `custom_type_sdt`.

```
slots:
  product_name_en:
    type: categorical
    influence_conversation: false
    values:
      - product_name_en
  province:
    type: categorical
    influence_conversation: true
    values:
      - province
```

Hình 16: Slot kiểu categorical

3.3.4. Actions.

Khai báo các actions trong file domain: là miền xác định mà bot của bạn hoạt động nó chỉ định intent, entities, slot, responses, forms và actions mà bot của bạn nên biết. Nó cũng xác định cấu hình cho các phiên hội thoại khác nhau.

```
actions:
- action_answer
- validate_action_order_form
- action_submit
- action_confirm_order
- action_reset_slot
- action_set_slot_composition
- action_set_slot_taste
- action_set_slot_effects
- action_set_slot_contraindications
- action_set_slot_user_manual
- action_set_slot_storage
- action_set_slot_made_in
- action_set_slot_sale
- action_set_slot_price
- action_set_slot_nominations
- action_set_slot_recognizing_signs
- action_set_slot_ship
- action_set_slot_user_object
```

Hình 17 : Danh sách các actions.

Khi chatbot cần thu thập nhiều thông tin từ khách hàng để tiến hành một tác vụ như: thông tin đặt hàng, thông tin khách hàng, ... Đó là lúc cần đến 2 Action đặc biệt để là Form Action và Follow up Action để đảm bảo các Action được thực hiện liên mạch. Ở hình 18 và hình 19 là ví dụ về các Action.


```

# ----- Trả lời yêu cầu của khách về thông tin sản phẩm -----#
class ActionAnswer(Action):
    def name(self) -> Text:
        return "action_answer"
    def run(self,
        dispatcher: CollectingDispatcher,
        tracker: Tracker,
        domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        if tracker.get_slot("product_name_en") is None:
            # nếu chưa nhận được tên sản phẩm sẽ yêu cầu khách hàng chọn sản phẩm cần tư vấn.
            dispatcher.utter_message(response="utter_ask_product_name_en")
        elif tracker.get_slot("request_counselling") is None:
            # Nếu chưa nhận được yêu cầu tư vấn từ khách sẽ hỏi khách cần tư vấn gì?
            dispatcher.utter_message(response="utter_ask_request_counselling")
        else:
            # Xử lý yêu cầu của khách
            requestCustom = str(tracker.get_slot("request_counselling"))
            productName = tracker.get_slot("product_name_en")
            content = "Rất xin lỗi!!!\nShop hiện chưa có thông tin về vấn đề này."
            if "bánh" in str(productName).lower():
                # Tư vấn theo sản phẩm Bánh định dưỡng Hebi
                if sheet["B" + requestCustom].value is not None:
                    content = str(sheet["B" + requestCustom].value)
            elif "men" in str(productName).lower():
                # Tư vấn theo sản phẩm Men vi khuẩn sống Việt Nhật
                if sheet["C" + requestCustom].value is not None:
                    content = str(sheet["C" + requestCustom].value)
            # Trả về thông tin khách yêu cầu
            dispatcher.utter_message(text=content)
            # reset slot request_counselling
            return [SlotSet("request_counselling", None)]
    return []

```

Hình 18: Custom action xử lý slot tenSP.

```

# ----- Lưu đơn hàng -----#
class ActionConfirm(Action):
    def name(self) -> Text:
        return "action_confirm_order"
    def run(self,
        dispatcher: CollectingDispatcher,
        tracker: Tracker,
        domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        # lấy thông tin đơn hàng từ slot
        name = str(tracker.get_slot("name"))
        phone = str(tracker.get_slot("phone"))
        product_name = str(tracker.get_slot("product_name_en"))
        amount = str(tracker.get_slot("amount"))
        total = str(tracker.get_slot("total"))
        date = str(datetime.datetime.today())
        address = str(tracker.get_slot("ward")) + ', ' + str(tracker.get_slot("district")) + ', ' + str(
            tracker.get_slot("province"))
        # Thêm 1 hàng giá trị vào file
        saveSuccess = save_order([name, phone, product_name, amount, total, date, address])
        # Hiện thị xác nhận đã lưu đơn
        if saveSuccess:
            dispatcher.utter_message(
                text="Đơn hàng của bạn đã được lưu lại✓\nShop sẽ liên hệ xác nhận trong vòng 24h, "
                "vui lòng chú ý điện thoại của bạn.\n")
        else:
            dispatcher.utter_message(
                text="Hệ thống đang bảo trì chức năng này. Xin lỗi vì sự bất tiện này!"
                "\nKhách hàng vui lòng trở lại sau 30 phút. \n")
        return [FollowupAction('action_reset_slot', None)]

```

Hình 19: Followup Action để lưu hoá đơn

3.3.5. Forms.

Forms là một kiểu đặc biệt của actions nhằm giúp Bot thu thập thông tin từ khách hàng. Định nghĩa Forms bằng các key trong file domain.yml.

```
forms:|
  action_order_form:
    required_slots:
      phone:
        - type: from_entity
          entity: phone
          intent: customer_phone_number
      name:
        - type: from_entity
          entity: name
          intent: customer_name
      product_name_en:
        - type: from_entity
          entity: product_name_en
      amount:
        - type: from_entity
          entity: amount
          intent: number_of_products
      province:
        - type: from_entity
          entity: province
          intent: answer_province
      district:
        - type: from_entity
          entity: district
          intent: answer_district
      ward:
        - type: from_entity
          entity: ward
          intent: answer_ward
```

Hình 20: Forms trong file domain.yml

Forms sẽ thực hiện thu thập các thông tin từ người dùng trong Forms đã được khai báo trước cho đến khi thu thập đủ thông tin để lấy dữ liệu vào các Slot ở trên. Các form_entity sẽ ánh xạ đủ vào các Slots dựa trên các entities đã được trích xuất.

3.3.6. Responses.

Responses là tin nhắn mà Bot gửi cho người dùng. Một phản hồi thường chỉ là văn bản, nhưng cũng có thể bao gồm nội dung như hình ảnh và các buttons.

```
responses:
| utter_greet:
|   - text: Xin chào Shop có thể giúp gì cho bạn?
| utter_bye:
|   - text: Bye👋
|   - text: Xin chào! Hẹn gặp lại👋
| utter_thanks:
|   - text: Cảm ơn {name} đã quan tâm đến các sản phẩm của Shop!☺
| utter_unconfirm:
|   - buttons:
|     - payload: /#
|       title: Sửa thông tin
|     - payload: /#
|       title: Xóa đơn
|     text: |-
|       Bạn cần sửa thông tin nào ạ?
|       Hoặc muốn xóa đơn hàng?
| utter_advisory:
|   - buttons:
|     - payload: /product_name{"product_name_en":"Bánh dinh dưỡng Hebi"}
|       title: Bánh dinh dưỡng Hebi
|     - payload: /product_name{"product_name_en":"Men vi khuẩn sống Việt Nhật"}
|       title: Men vi khuẩn sống Việt Nhật
|     text: 'Shop hiện có 2 sản phẩm là:'
| utter_ask_product_name_en:
|   - buttons:
|     - payload: /product_name{"product_name_en":"Bánh dinh dưỡng Hebi"}
|       title: Bánh Hebi 🍪
|     - payload: /product_name{"product_name_en":"Men vi khuẩn sống Việt Nhật"}
|       title: Men Việt Nhật 🦠
|     text: 'Shop có 2 sản phẩm:\n🍪Bánh dinh dưỡng Hebi 🍪
|       \n🦠Men vi khuẩn sống Việt Nhật 🦠\nBạn đang cần sản phẩm nào?'
| utter_ask_amount:
|   - text: Bạn {name} mua bao nhiêu sản phẩm {product_name_en} ạ?
| utter_ask_request_counselling:
|   - buttons:
|     - payload: /price{"request_counselling":"price"}
|       title: Giá SP
|     - payload: /effects{"request_counselling":"effects"}
|       title: Công dụng SP
|     text: Bạn cần biết gì về sản phẩm {product_name_en}?

utter_order_details:
|   - buttons:
|     - payload: /confirm
|       title: Đặt hàng
|     - payload: /unconfirm
|       title: Hủy đơn
|   text: |-
|     Đơn hàng bạn {name}:
|     📞 SĐT: {phone}
|     📦 SP: {product_name_en}
|     ✍ SL: {amount}
|     📍 Địa chỉ: {ward}, {district}, {province}
|     💰 Tổng tiền: {total} VNĐ
|     Vui lòng xác nhận đặt hàng!!
|     (Chat: Đồng ý hoặc Hủy đơn)

utter_ask_age:
|   - buttons:
|     - payload: /year_old{"age":"1"}
|       title: Dưới 2 tuổi 🍼
|     - payload: /year_old{"age":"3"}
|       title: 2 đến 15 tuổi 👦
|     - payload: /year_old{"age":"20"}
|       title: Trên 15 tuổi 👤
|   text: Người dùng ở độ tuổi nào trong 3 lựa chọn?

utter_ask_name:
|   - text: Cho Shop hỏi bạn tên gì?👋
|   - text: Bạn cho Shop xin tên👋
|   - text: Tên của bạn là gì?👋

utter_ask_phone:
|   - text: 📞 Số điện thoại của bạn là...
|   - text: 📞 Cho Shop xin số điện thoại của bạn a...

utter_nlu_fallback:
|   - text: |-
|     Xin lỗi! Shop chưa hiểu ý của bạn.
|     Vui lòng cho Shop biết chính xác bạn cần gì.

utter_ask_province:
|   - text: Bạn cần giao hàng đến Tỉnh/Thành phố nào?

utter_ask_district:
|   - text: Tên Quận/Huyện?

utter_ask_ward:
|   - text: Tên Phường/Xã?

utter_list_district:
|   - text: 'Các Huyện: {list_district}'

utter_list_ward:
|   - text: 'Các Xã: {list_ward}'
```

Hình 21: responses trong file domain.yml

Responses sẽ được khai báo trong domain.yml. Mỗi responses được khai báo thường bắt đầu bằng “utter_” và mỗi response sẽ đặt theo tên intent tương ứng.

3.3.7. Stories.

Stories là một loại dữ liệu đào tạo được sử dụng để đào tạo mô hình quản lý đối thoại của Bot. Các stories có thể được sử dụng để đào tạo các mô hình có khả năng khái quát hóa thành các hội có sẵn theo một hướng nhất định nào đó.

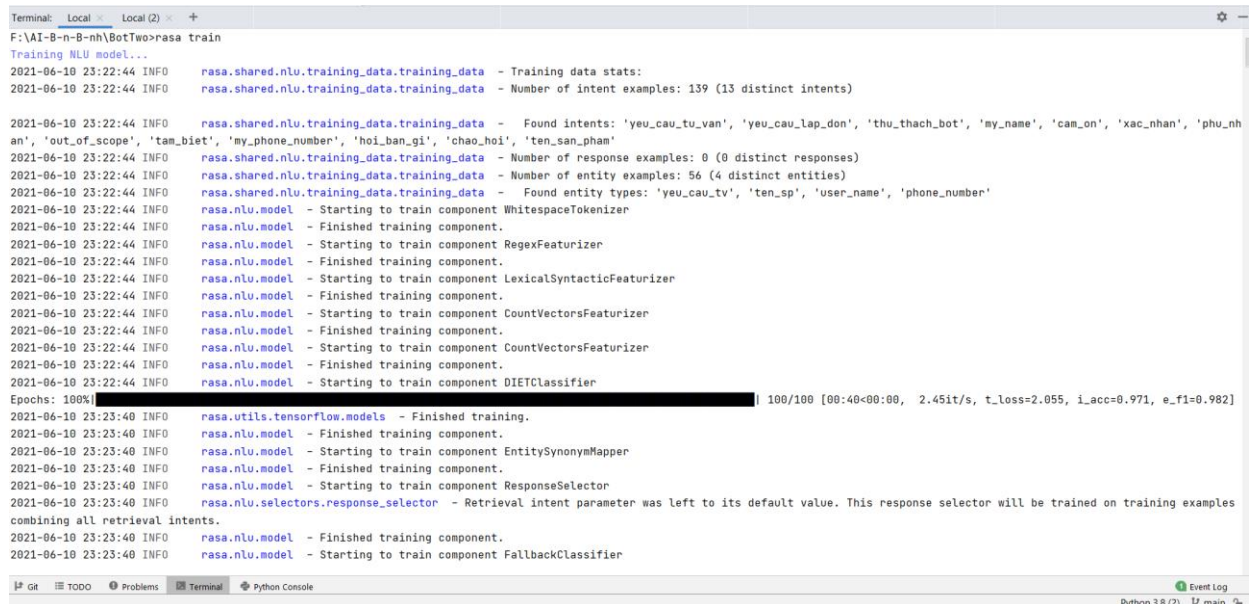
```
stories:
- story: chao hoi 1
  steps:
  - intent: greet
  - action: utter_greet
  - intent: bye
  - action: utter_bye
- story: hoi gia
  steps:
  - intent: greet
  - action: utter_greet
  - intent: price
  - action: action_set_slot_price
  - action: action_answer
  - intent: bye
  - action: utter_bye
```

Hình 22: story về tư vấn.

Stories là bản trình bày cuộc hội thoại giữa người dùng và Bot, được chuyển đổi thành một định dạng cụ thể trong đó thông tin người dùng nhập vào được thể hiện trong intent và Bot sẽ phản hồi ở action và responses.

3.4. Thử nghiệm

Tiến hành thử nghiệm tương tác với Chatbot với một số câu thì thấy độ chính xác của Chatbot đạt khoảng 80% và chỉ tính các câu hỏi xung quanh kịch bản đã đào tạo cho Chatbot. Còn đối với các câu hỏi mà chưa training thì bot sẽ đưa ra một số mẫu câu trả lời điều hướng người dùng tới một số câu hỏi sẵn có mà Chatbot có thể đáp ứng được.



```
Terminal: Local (2) x +
F:\AI-B-N-B-nh\BotTwo>rasa train
Training NLU model...
2021-06-10 23:22:44 INFO rasa.shared.nlu.training_data.training_data - Training data stats:
2021-06-10 23:22:44 INFO rasa.shared.nlu.training_data.training_data - Number of intent examples: 139 (13 distinct intents)

2021-06-10 23:22:44 INFO rasa.shared.nlu.training_data.training_data - Found intents: 'yeu_cau_tu_van', 'yeu_cau_lap_don', 'thu_thach_bot', 'my_name', 'cam_on', 'xac_nhan', 'phu_nh
an', 'out_of_scope', 'tam_biet', 'my_phone_number', 'hoi_ban_gi', 'chao_hoi', 'ten_san_pham'
2021-06-10 23:22:44 INFO rasa.shared.nlu.training_data.training_data - Number of response examples: 0 (0 distinct responses)
2021-06-10 23:22:44 INFO rasa.shared.nlu.training_data.training_data - Number of entity examples: 56 (4 distinct entities)
2021-06-10 23:22:44 INFO rasa.shared.nlu.training_data.training_data - Found entity types: 'yeu_cau_tv', 'ten_sp', 'user_name', 'phone_number'
2021-06-10 23:22:44 INFO rasa.nlu.model - Starting to train component WhitespaceTokenizer
2021-06-10 23:22:44 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:22:44 INFO rasa.nlu.model - Starting to train component RegexFeaturizer
2021-06-10 23:22:44 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:22:44 INFO rasa.nlu.model - Starting to train component LexicalSyntacticFeaturizer
2021-06-10 23:22:44 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:22:44 INFO rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2021-06-10 23:22:44 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:22:44 INFO rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2021-06-10 23:22:44 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:22:44 INFO rasa.nlu.model - Starting to train component DIETClassifier
Epochs: 100% | 100/100 [00:40<00:00, 2.45it/s, t_loss=2.055, i_acc=0.971, e_f1=0.982]
2021-06-10 23:23:40 INFO rasa.utils.tensorflow.models - Finished training.
2021-06-10 23:23:40 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:23:40 INFO rasa.nlu.model - Starting to train component EntitySynonymMapper
2021-06-10 23:23:40 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:23:40 INFO rasa.nlu.model - Starting to train component ResponseSelector
2021-06-10 23:23:40 INFO rasa.nlu.selectors.response_selector - Retrieval intent parameter was left to its default value. This response selector will be trained on training examples
combining all retrieval intents.
2021-06-10 23:23:40 INFO rasa.nlu.model - Finished training component.
2021-06-10 23:23:40 INFO rasa.nlu.model - Starting to train component FallbackClassifier
```

Hình 23: Huấn luyện cho Chatbot.

```
Your input -> xin chào
Chào bạn mình có thể giúp gì cho bạn?
Your input -> em muốn đặt hàng
Tôi là trợ lý ảo 💎 của page Bánh Dinh Dưỡng HEBI, nguồn từ Rasa.
tôi được tạo ra để có thể hỗ trợ cô chủ tôi nhận thông tin đặt hàng
từ quý khách
nếu có bất cứ điều gì sai sót mang quý khách thông cảm!💎
Tên của bạn là gì?💎
Your input -> mình tên Thọ
☎ Số điện thoại của bạn là...
Your input -> 097327193
? Vui lòng chọn sản phẩm: 1: Bánh dinh dưỡng Hebi (/ten_san_pham{"ten_sp":
Đơn hàng bạn Thọ đã được lưu lại!
☎ Số điện thoại: 097327193
💎 Sản phẩm: Bánh dinh dưỡng Hebi
Bạn sẽ nhận được cuộc gọi xác nhận trong vòng 24h
Cảm ơn Thọ đã quan tâm đến sản phẩm của chúng tôi :v
```

Hình 24: Thử nghiệm Chatbot.

Chương trình huấn luyện và thử nghiệm Chatbot là trực tiếp trên nền tảng của Rasa framework dựa trên ngôn ngữ lập trình Python.

3.5. Đánh giá

Trong quá trình tìm hiểu, học hỏi và xây dựng từ nội dung nghiên cứu thì nhóm có những đánh giá sau:

1. Thu thập và xây dựng dữ liệu đào tạo, huấn luyện cho Chatbot và các cuộc hội thoại với khách hàng. Đây là yếu tố quan trọng có sự ảnh hưởng rất lớn đến tính linh hoạt và khả năng phản hồi của Chatbot.
2. Việc định nghĩa và xây dựng các intent cũng khá quan trọng vì phải phân tích với các chủ đề khác nhau mà khách hàng có thể đưa ra từ đó có thể tổng hợp lại các intent tốt nhất. Đối với các intent không rõ ràng hay gần nhau về mặt ngữ nghĩa sẽ khiến cho độ chính xác của Chatbot giảm. Nên việc định nghĩa các intent và slot là vô cùng quan trọng.
3. Với các đoạn hội thoại nằm trong story dựng sẵn thì Chatbot đáp ứng rất tốt khi trả lời đúng cho người dùng. Tuy nhiên việc xây dựng story cho Chatbot rất khó khăn trong các đoạn hội thoại vì nó có thể xảy ra rất nhiều trường hợp khác nhau phát sinh thêm khi khách hàng hỏi.
4. Bot có khả năng hỏi và thu thập thông tin khách hàng từ đó cuộc trò chuyện từ đó khách hàng cảm thấy thân thiện hơn. Tuy nhiên với việc xưng hô theo tên của khách hàng đôi lúc có tên đặc biệt sẽ hơi khó khăn vì dữ liệu train chưa được nhiều.

Qua bài toán trên nhóm đã áp dụng Rasa framework vào Chatbot cho việc tư vấn cho khách hàng về sản phẩm dinh dưỡng là rất khả thi, có tính thực tiễn cao do đáp ứng được một số vấn đề trong nghiệp vụ bán hàng.

CHƯƠNG 4: KẾT LUẬN

4.1. Kết quả đạt được.

1. Trình bày kiến thức tổng quan về một hệ thống Chatbot, tìm hiểu về một số mô hình Chatbot bán hàng hiện nay đang sử dụng.
2. Nắm được các luồng hoạt động và các bước xử lý của các thành phần trong Rasa framework.
3. Trong quá trình xây dựng tập dữ liệu đào tạo, huấn luyện cho Chatbot đã giúp nhóm có được những kinh nghiệm trong việc xử lý và gán nhãn dữ liệu. Từ đó có thể xây dựng bộ dữ liệu huấn luyện bot tốt hơn đem lại độ chính xác cao hơn mô hình.
4. Xây dựng thành công chương trình thử nghiệm hệ thống Chatbot phục vụ tư vấn các sản phẩm dinh dưỡng qua các kênh Fanpage Facebook đang bán hàng thật. Tuy nhiên, việc đánh giá kết quả vẫn còn thực hiện thủ công .

4.2. Hạn chế.

1. Nội dung cuộc trò chuyện có giới hạn nhất định: Mặc dù Chatbot có thể phản hồi lại khách hàng rất nhanh, nhưng không có nghĩa là Chatbot sẽ trả lời mọi câu hỏi mà khách hàng đưa ra. Với mỗi tình huống và câu hỏi khác nhau khi chưa được gặp thì Bot sẽ không thể đưa ra câu trả lời với khách hàng đang tìm kiếm. Vậy nên ChatBot chỉ hoạt động tốt nhất trong một lĩnh vực cụ thể nào đó.
2. Khách hàng trò chuyện với Bot đôi lúc sẽ cảm thấy nhàm chán và trả lời cứng nhắc khi Chatbot chỉ lặp đi lặp lại cuộc trò chuyện.
3. Không thể tự lôi kéo khách hàng. Đôi khi khách hàng chỉ mua một sản phẩm mà Bot không biết tư vấn lỗi khéo khách mua thêm sản phẩm kèm theo và không thể tự đưa ra quyết định khi khách hàng muốn điều kiện kèm theo.

4.3. Hướng phát triển

1. Cung cấp các dịch vụ tư vấn cho khách hàng một cách nhanh chóng.
2. Có thể tương tác 24/7 với hệ thống một cách hiệu quả.
3. Phân tích và đưa ra những câu trả lời thấu hiểu hành vi của khách hàng.
4. Xây dựng Chatbot trở nên giống người hơn khi trò chuyện.

TÀI LIỆU THAM KHẢO

1. Tài liệu trên trang chủ của Rasa - <https://rasa.com/docs/rasa/>
2. Blog MiAI của anh Nguyễn Chiến Thắng - <https://miai.vn/2019/09/03/rasa-series-1-ai-cung-co-the-lam-chatbot-sieu-ngon-khong-lo/>
3. Tài liệu của anh Bui Tien Tun trên VIBLO - <https://viblo.asia/u/tungbuitien>
4. Chatbot – Wikipedia:
https://en.wikipedia.org/wiki/Chatbot#Usage_in_dialog_systems
5. Building Bots with the Rasa Framework - <https://discover.bot/bot-talk/guide-to-bot-buiding-frameworks/rasa/>
6. What Are Chatbots: 5 Things to Know to Prevent Your Business from Falling Behind - <https://botscrew.com/blog/what-are-chatbots/>
7. What are Chatbots? (How do They Work?) - <https://www.kommunicate.io/blog/what-are-chatbots/>
8. Enterprise-grade open source platform for Rasa teams - <https://botfront.io/>

PHỤ LỤC

Tôi là trợ lí ảo 🤖 của page Bánh Dinh Dưỡng HEBI, nguồn từ Rasa. tôi được tạo ra để có thể hỗ trợ cô chủ tôi nhận thông tin đặt hàng từ quý khách nếu có bất cứ điều gì sai sót mang quý khách thông cảm! 🙏🙏

Cho mình hỏi bạn tên gì? 🙏🙏

☎ Cho mình xin số điện thoại của bạn ạ...

Đơn hàng bạn An đã được lưu lại!
☎ Số điện thoại: 1848658655566666666665
🍰 Sản phẩm: Bánh dinh dưỡng Hebi
Bạn sẽ nhận được cuộc gọi xác nhận trong vòng 24h
Cảm ơn An đã quan tâm đến bánh dinh dưỡng HEBI

Mình muốn mua 10 Bánh

Mk tên An

1848658655566666666665

Aa

Chào bạn mình có thể giúp gì cho bạn?

👍 320k/10 bánh

Tôi là trợ lí ảo 🤖 của page Bánh Dinh Dưỡng HEBI, nguồn từ Rasa. tôi được tạo ra để có thể hỗ trợ cô chủ tôi nhận thông tin đặt hàng từ quý khách nếu có bất cứ điều gì sai sót mang quý khách thông cảm! 🙏🙏

Tên của bạn là gì? 🙏🙏

☎ Cho mình xin số điện thoại của bạn ạ...

Đơn hàng bạn Thọ đã được lưu lại!
☎ Số điện thoại: 0973271289
🍰 Sản phẩm: Bánh dinh dưỡng Hebi
Bạn sẽ nhận được cuộc gọi xác nhận trong vòng 24h
Cảm ơn Thọ đã quan tâm đến sản phẩm của chúng tôi 🙏

Xin chào

Giá của Bánh Hebi a

cho mình đặt hàng a

Mình tên Thọ

0973271289

Aa

xin chào

Xin chào Shop có thể giúp gì cho bạn?

Sản phẩm có giá 320k/10 gói (Một gói bánh có 8 miếng)

Xin lỗi ! Tôi chưa hiểu ý của bạn.
Vui lòng cho tôi biết chính xác bạn cần gì.

Cho Shop hỏi bạn tên gì?

Cho Shop xin số điện thoại của bạn ạ...

bạn ở tỉnh/thành nào

bạn ở quận/huyện nào

bạn ở phường/xã nào

Đơn hàng bạn An:

☎ SĐT: 0987191143

🍪 SP: Bánh dinh dưỡng Hebi

✂ SL: 10

📍 Địa chỉ: Xã Yên Thái, Huyện Yên Mô, Tỉnh Ninh Bình

💰 Tổng tiền: 3200000 VNĐ

Vui lòng xác nhận đặt hàng!!
(Chat: Đồng ý hoặc Hủy đơn)

Đặt hàng

Hủy đơn

bạn ở phường/xã nào

Đơn hàng bạn An:

☎ SĐT: 0987191143

🍪 SP: Bánh dinh dưỡng Hebi

✂ SL: 10

📍 Địa chỉ: Xã Yên Thái, Huyện Yên Mô, Tỉnh Ninh Bình

💰 Tổng tiền: 3200000 VNĐ

Vui lòng xác nhận đặt hàng!!
(Chat: Đồng ý hoặc Hủy đơn)

Đặt hàng

Hủy đơn

Đơn hàng của bạn đã được lưu lại
Shop sẽ liên hệ xác nhận trong vòng 24h, vui lòng chú ý điện thoại của bạn.

Bánh Hebi giá bao nhiêu ạ

cho mình đặt 10 gói nhé

mình muốn đặt hàng

An

0987191143

Tỉnh Ninh Bình

Tỉnh Ninh Bình

Huyện Yên Mô

Xã Yên Thái

Xã Yên Thái

Đặt hàng