

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



**BÁO CÁO NIÊN LUẬN CƠ SỞ NGÀNH
KHOA HỌC MÁY TÍNH
(CT20105)**

**ĐỀ TÀI
TÌM HIỂU RASA FRAMEWORK,
XÂY DỰNG CHATBOT HỖ TRỢ THÔNG TIN
LIÊN QUAN ĐẾN COVID-19**

**Giảng viên hướng dẫn:
Trần Nguyễn Dương Chi**

Sinh viên thực hiện:
Trần Anh Khoa B1913240

Cần Thơ, 05/2022

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN	2
DANH MỤC HÌNH ẢNH	4
GIỚI THIỆU	5
I. ĐẶT VẤN ĐỀ.....	5
II. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ	6
III. MỤC TIÊU CỦA ĐỀ TÀI	6
IV. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	6
V. PHƯƠNG PHÁP NGHIÊN CỨU	6
VI. KẾT QUẢ ĐẠT ĐƯỢC.....	7
MÔ TẢ CHI TIẾT BÀI TOÁN.....	8
I. MÔ TẢ CHI TIẾT BÀI TOÁN	8
II. VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN	8
III. GIẢI PHÁP CÓ LIÊN QUAN ĐẾN BÀI TOÁN	8
1. Cách tìm API cung cấp số liệu ca nhiễm:	8
2. Tìm các câu trả lời liên quan đến câu hỏi đã được hoạch định từ trước:.....	9
CÀI ĐẶT	10
I. CÀI ĐẶT CÁC THƯ VIỆN CẦN THIẾT	10
II. TIẾN HÀNH XÂY DỰNG MÔ HÌNH CHATBOT	10
PHẦN KẾT LUẬN	18
I. KẾT QUẢ ĐẠT ĐƯỢC.....	18
II. HẠN CHẾ	18
III. HƯỚNG PHÁT TRIỂN	18
TÀI LIỆU THAM KHẢO:.....	19

DANH MỤC HÌNH ẢNH

Hình 1. Demo kết quả đạt được.....	7
Hình 2. Demo kết quả đạt được.....	7
Hình 3. Demo kết quả đạt được.....	7
Hình 4. Ảnh minh họa về API cung cấp số liệu ca nhiễm covid-19	8
Hình 5. Tìm dữ liệu cho các câu hỏi về covid-19	9
Hình 6. Cấu trúc tổ chức của Rasa Framework.....	10
Hình 7. Giao diện tương tác với chatbot sau khi đã được xây dựng xong	17

GIỚI THIỆU

I. ĐẶT VẤN ĐỀ

Trong những năm gần đây, Chatbots mặc dù còn rất mới trong lĩnh vực công nghệ nhưng đang được các trung tâm nghiên cứu, các trường đại học, cao đẳng nhanh chóng nghiên cứu và phát triển, có nhiều ứng dụng trong các lĩnh vực khác nhau như giải trí, chăm sóc y tế, kinh doanh, tự động hóa,... Kể từ năm 2016, chatbots đã nhanh chóng trở thành một trong những xu hướng công nghệ phổ biến nhất.

Chatbots phát triển dựa trên sự kết hợp của các kịch bản có trước và tự học trong quá trình tương tác. Ta sẽ tương tác với chatbots qua nền tảng tin nhắn. Với các câu hỏi được đặt ra, Chatbots sử dụng các hệ thống xử lý ngôn ngữ tự nhiên (Natural Language Processing) để phân tích dữ liệu sau đó chúng lựa chọn các thuật toán học máy để đưa ra các loại phản hồi khác nhau, chúng sẽ dự đoán và phản hồi chính xác nhất có thể. Chatbots sử dụng nhiều hệ thống quét các từ khóa bên trong đầu vào, sau đó bot khởi động một hành động, kéo một câu trả lời với các từ khóa phù hợp nhất và trả lời thông tin từ một cơ sở dữ liệu / API, hoặc bàn giao cho con người. Nếu tình huống đó chưa xảy ra (không có trong dữ liệu). Chatbot sẽ bỏ qua nhưng sẽ đồng thời tự học để áp dụng cho các cuộc trò chuyện về sau.

Một trong các yếu tố làm nên sức mạnh của Chatbot là khả năng tự học hỏi. Càng được sử dụng, tương tác với người dùng nhiều, nền tảng Chatbot càng “thông minh”. Chatbot thông minh có khả năng tự học hỏi dựa trên các dữ liệu đưa vào mà không cần phải được lập trình cụ thể (đó được gọi là phương pháp máy học - Machine Learning. Chính điều này làm cho các nhà phát triển dễ dàng tạo các chương trình trò chuyện và tự động hoá các cuộc trò chuyện với người dùng.

Hiện nay, Trong bối cảnh đại dịch Covid-19 đang hoành hành, mọi người phải hạn chế đi lại và phải tự cách ly tại nhà thì Chatbot lại đóng một vai trò quan trọng trong việc hỗ trợ cung cấp thông tin về covid-19 đến với mọi người một cách tự động và nhanh chóng, giúp mọi người có thể cập nhật thông tin về tình hình dịch bệnh một cách chính xác và kịp thời mà không cần phải tiếp xúc với bất kỳ ai. Từ đó không chỉ tiết kiệm được thời gian mà còn góp phần bảo vệ sức khỏe cho chính bản thân mình và người thân trong gia đình.

Và Rasa là một framework máy học mã nguồn mở hỗ trợ mạnh mẽ cho việc xây dựng Chatbot, giúp người dùng xây dựng được những hệ thống tự động và tối ưu. Xuất phát từ mục tiêu tiếp cận, bổ sung các kiến thức mới, cũng như củng cố lại những kỹ năng và kiến thức trong suốt quá trình học tập tại trường, đồng thời muốn nghiên cứu sâu hơn về Rasa Framework và ứng dụng Rasa cho việc tự động hóa các quy trình. Do đó, tôi quyết định chọn đề tài: **“Xây dựng chatbot hỗ trợ thông tin covid-19.”**

II. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ

Phương pháp được dựa vào các kiến thức được chia sẻ từ nguồn uy tín:

- Tài liệu từ trang chủ của Rasa:
<https://rasa.com/docs/rasa/>
- Video hỗ trợ từ kênh youtube của Rasa:
https://www.youtube.com/watch?v=Ap62n_YAVZ8&list=PL75e0qA87dIEjGAc9j9v3a5h1mxI2Z9fi

III. MỤC TIÊU CỦA ĐỀ TÀI

Ứng dụng các kiến thức sau khi tìm hiểu về Rasa Framework để xây dựng hệ thống chatbot nhằm hỗ trợ thông tin về covid-19. Chatbot phải trả lời được tối thiểu những nội dung sau:

- Covid-19 là gì?
- Các triệu chứng của bệnh covid-19 là gì?
- Cách điều trị covid-19 tại nhà?
- Cách phòng ngừa lây nhiễm covid-19?
- Tiêm Vaccine có an toàn không?
- Lợi ích khi tiêm vắc xin là gì?
- Thời gian vaccine có tác dụng kéo dài bao lâu?
- Vắc xin có tác dụng phụ không?
- Hậu covid là gì?
- Những biểu hiện của hậu covid là gì?
- Thú cưng có lây covid-19 cho người được hay không?
- Số ca nhiễm covid ở Việt Nam hôm nay?
- Số ca nhiễm covid trên thế giới hôm nay?

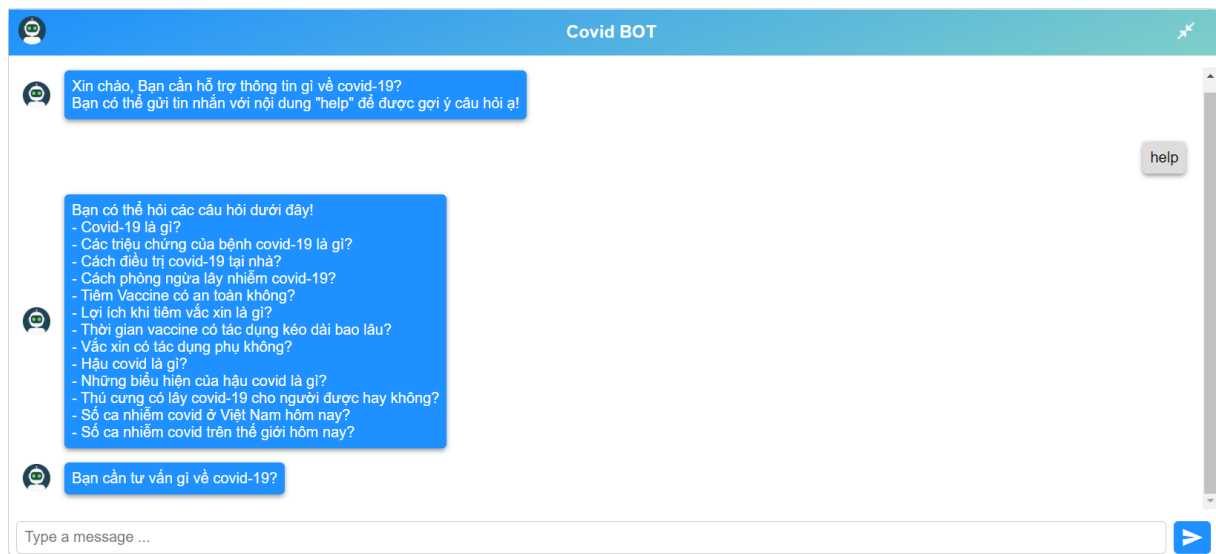
IV. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Chủ đề và phạm vi nghiên cứu: Hệ thống chatbot được xây dựng với mục tiêu học tập và nghiên cứu về nền tảng Rasa.

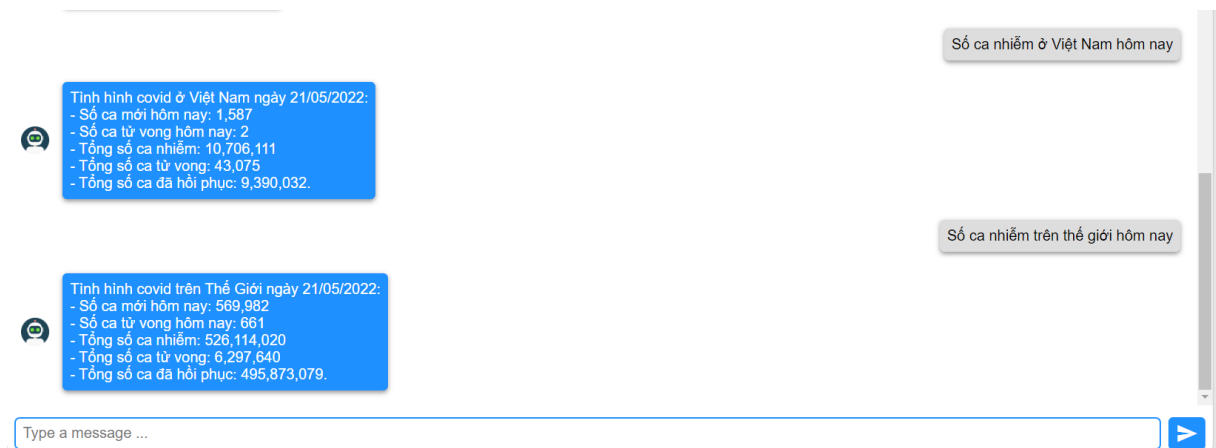
V. PHƯƠNG PHÁP NGHIÊN CỨU

- Nghiên cứu về cách tổ chức và hoạt động của nền tảng Rasa.
- Cài đặt Rasa và tìm hiểu các thành phần bên trong gói được cài đặt.
- Tìm kiếm thông tin từ cdc Việt Nam để tìm câu trả lời cho các câu hỏi đã đặt ra.
- Tìm kiếm API về số liệu ca nhiễm covid-19 ở Việt Nam cũng như trên Thế Giới.

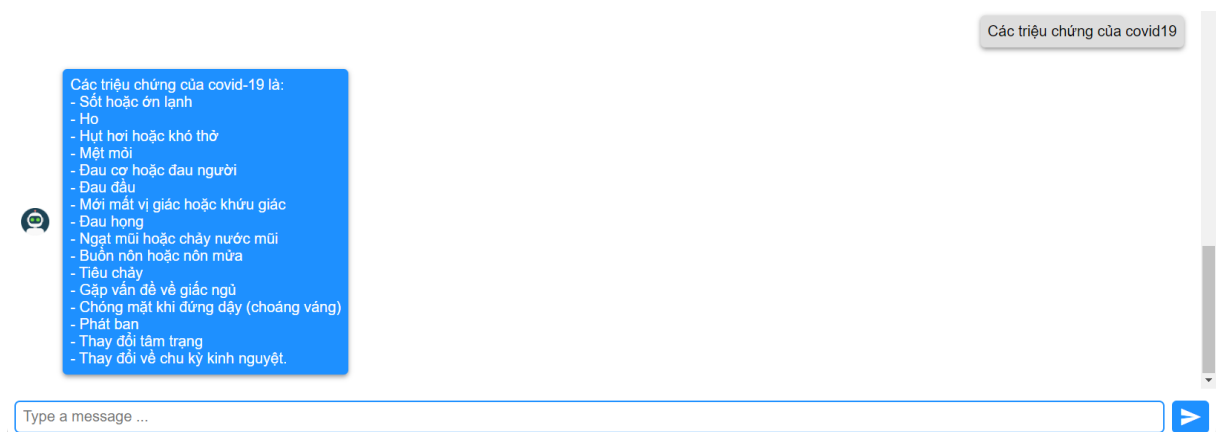
VI. KẾT QUẢ ĐẠT ĐƯỢC



Hình 1. Demo kết quả đạt được



Hình 2. Demo kết quả đạt được



Hình 3. Demo kết quả đạt được

MÔ TẢ CHI TIẾT BÀI TOÁN

I. MÔ TẢ CHI TIẾT BÀI TOÁN

Đề tài: “Xây dựng chatbot hỗ trợ thông tin covid-19” thuộc dạng bài toán máy học, sử dụng nguồn dữ liệu được cung cấp từ người dùng nhập vào hoặc lấy từ API của website được liên kết để học. Để có thể cung cấp thông tin một cách chính xác nhất chúng ta cần lấy dữ liệu từ trang chủ của cdc Việt Nam và sử dụng API về số liệu về ca nhiễm covid từ nguồn uy tín.

II. VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN

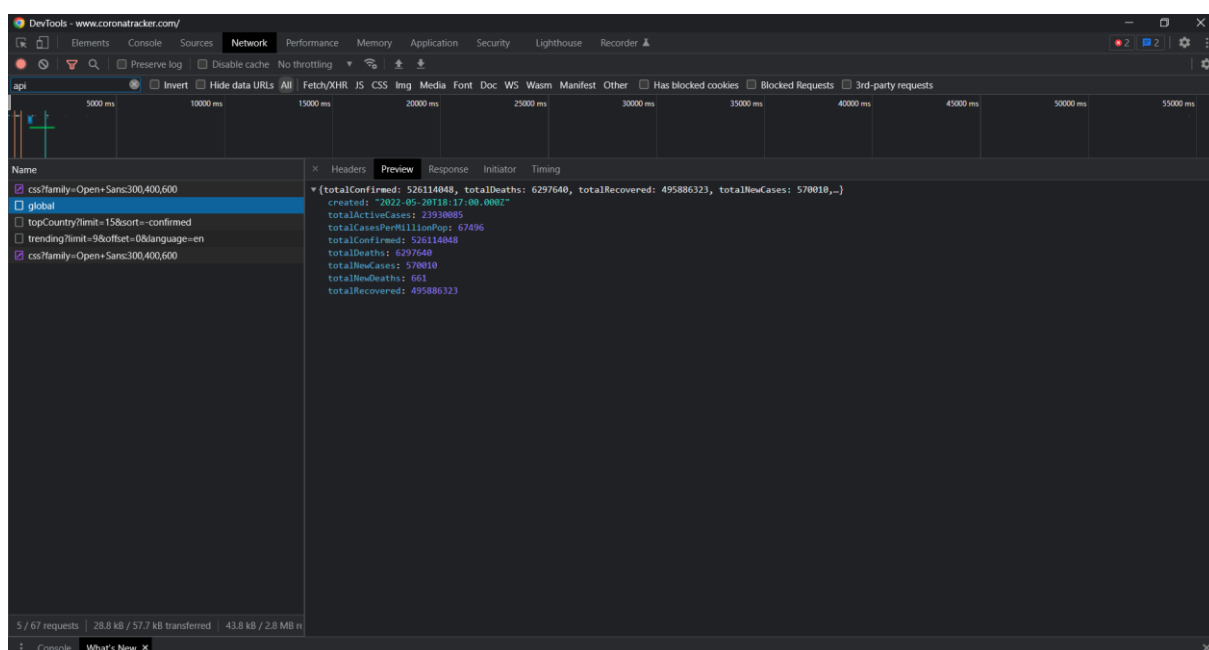
Các vấn đề liên quan đến việc tìm dữ liệu cho chatbot:

- Tìm API cung cấp số liệu về ca nhiễm
- Tìm các câu trả lời liên quan đến câu hỏi đã hoạch định phía trên.

III. GIẢI PHÁP CÓ LIÊN QUAN ĐẾN BÀI TOÁN

1. Cách tìm API cung cấp số liệu ca nhiễm:

Để lấy được API của một website ta cần sử dụng công cụ devtool của trình duyệt Google Chrome hoặc Microsoft Edge để xem những truy vấn từ máy khách lên máy chủ. Công cụ này sẽ hiển thị tất cả các truy vấn khi ta bắt đầu truy cập bất kỳ một trang web nào, từ đó ta có thể tìm thấy được đường dẫn của API. Để có thể lấy được request của trang web coronatracker.com một website cung cấp số liệu về ca nhiễm covid-19 khá uy tín, chúng ta cần mở thanh công cụ devtool từ một trong hai trình duyệt phía bên trên bằng cách ấn phím F12 sau đó bật tab network và truy cập vào trang web <https://www.coronatracker.com/> sau khi trang web đã được load đầy đủ tất cả request đều được hiển thị như hình.



Hình 4. Ảnh minh họa về API cung cấp số liệu ca nhiễm covid-19

Dựa vào request và response ta có thể thấy rằng để lấy được các số liệu về ca nhiễm trên toàn thế giới của website coronatracker.com thì truy vấn đến API sau:

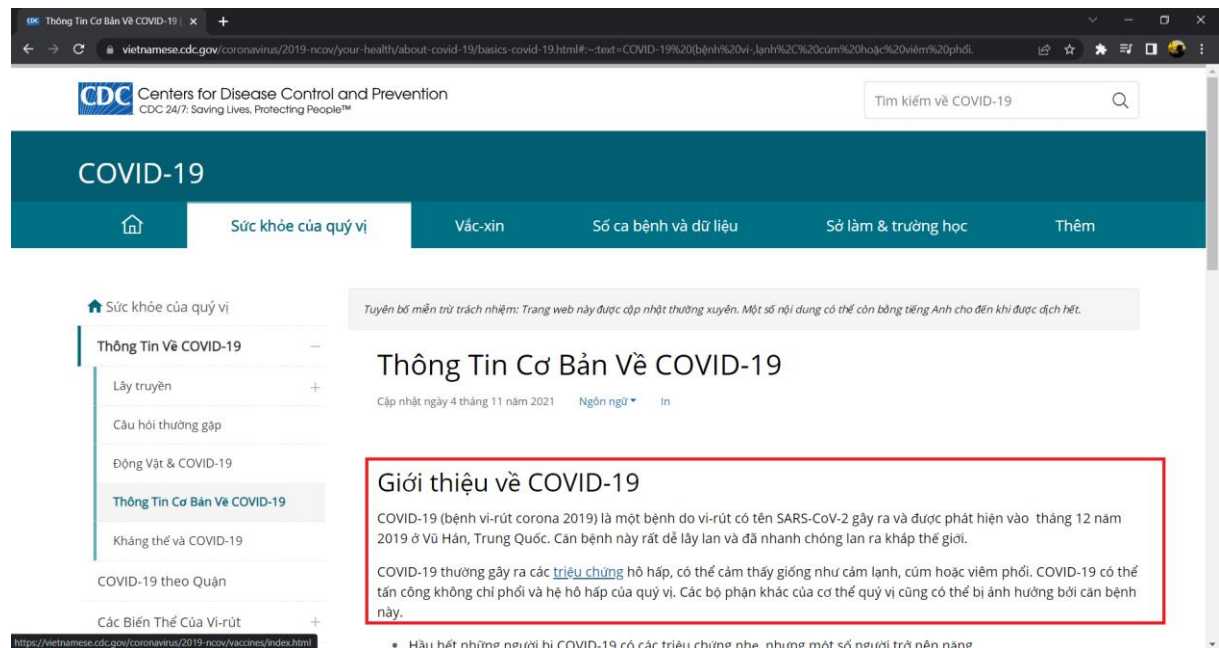
<https://api.coronatracker.com/v3/stats/worldometer/global>

Tương tự như vậy ta cũng lấy được API về số liệu ca nhiễm ở Việt Nam:

<https://api.coronatracker.com/v3/stats/worldometer/country?countryCode=VN>

2. Tìm các câu trả lời liên quan đến câu hỏi đã được hoạch định từ trước:

Ví dụ: Đối với câu hỏi “Covid-19 là gì?” ta có thể lấy câu trả lời ở khung màu đỏ để mô hình có thể học từ nó. Và đối với các câu hỏi khác cũng tương tự như vậy.



Hình 5. Tìm dữ liệu cho các câu hỏi về covid-19

CÀI ĐẶT

I. CÀI ĐẶT CÁC THƯ VIỆN CẦN THIẾT

Cài đặt Rasa Framework và thư viện requests:

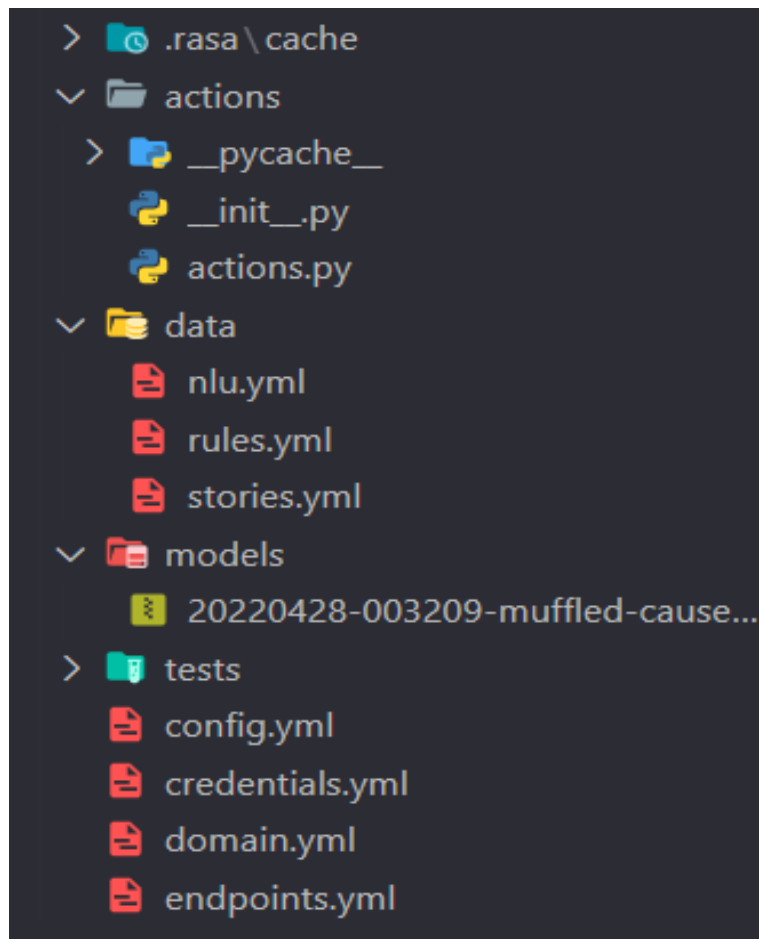
```
!pip install rasa  
!pip install requests
```

II. TIẾN HÀNH XÂY DỰNG MÔ HÌNH CHATBOT

Sau khi cài đặt Rasa Framework phía trên xong, ta bắt đầu khởi tạo dự án đầu tiên với Rasa bằng lệnh:

```
rasa init
```

Rasa sẽ lập tức sinh ra các file cần thiết cho dự án



Hình 6. Cấu trúc tổ chức của Rasa Framework

Đầu tiên chúng ta sẽ đến file config.yml cũng là phần mà Rasa sử dụng nhiều nhất. Đây là phần cấu hình cho NLU (Natural Language Understanding), nơi chúng ta lựa chọn ngôn ngữ, model cần thiết. Chúng ta sẽ lựa chọn ngôn ngữ ở đây là vi (vietnam). Tiếp theo là đến pipeline, có thể thấy pipeline chỉ là một dòng đơn giản, vậy thực chất là nó làm gì? Đó là một quy trình hoàn chỉnh từ lựa chọn Tokenizer, Featurizer, Extractor đến Classifier. Tất nhiên chúng ta hoàn toàn có thể lựa chọn thay thế bất cứ một công đoạn nào trong pipeline này nếu cảm thấy nó sẽ đạt hiệu quả tốt hơn. Việc lựa chọn phương án thay thế có thể là trong chính Rasa hoặc một package nào đó bên ngoài mà chúng ta thấy phù hợp.

Bên cạnh supervised_embeddings thì Rasa cũng cung cấp sẵn thêm một vài pipeline template khác là pretrained_embeddings_spacy, pretrained_embeddings_convert, ... (ở phần đọc thêm [Docs](#)). Trong trường hợp này của chúng ta, với ngôn ngữ tiếng việt thì supervised_embeddings là lựa chọn tốt nhất vì nó thực hiện train lại từ đầu, dựa vào file nlu.yml mà chúng ta cung cấp.

Đầu tiên là lựa chọn pipeline trong file config.yml:

```
recipe: default.v1
language: vi

pipeline:
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: char_wb
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 100
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
```

Đề nạp vào những câu hỏi từ người dùng có thể sẽ tương tác với Rasa chúng ta vào thư mục data/nlu.yml để thêm vào nội dung mới với cú pháp như sau:

```
version: "3.0"

nlu:
- intent: what_is_the_covid_19
  examples: |
    - Covid-19 là gì?
    - Thế nào là covid-19?
    - Covid-19 là bị gì?
    - Covid là gì?
```

Sau khi đã thêm các intent cần thiết, chúng ta sẽ tiến hành vào file domain.yml để thêm vào các câu trả lời cho những câu hỏi đã thêm ở bước trên.

Đầu tiên trong file domain.yml chúng ta cần phải khai báo các intents ở phía trên như sau:

```
version: "3.0"

intents:
- help
- what_is_the_covid_19 #Covid 19 là gì?
- manifestations_of_covid_19_disease # Các triệu chứng của covid 19?
- treat_covid_19_at_home # Cách điều trị covid 19 tại nhà?
- preventing_the_spread_of_covid_19 # Cách phòng ngừa lây nhiễm covid 19?
- vaccine_time # Thời gian vaccine có tác dụng?
- what_is_the_post_covid_19 # Hậu covid 19 là gì?
- symptoms_of_post_covid_19 # Biểu hiện của hậu covid?
- pet_and_human # Thú cưng có lây covid cho người hay không?
- benefits_of_vaccination_against_covid_19 # Lợi ích của tiêm vắc xin
- vaccine_safety # Tiêm vắc xin có an toàn không?
- vaccine_side_effects # Vắc xin có tác dụng phụ không?
- thanks # Lời cảm ơn
- goodbye # Tạm biệt
- bot_challenge # Bot trả lời thông tin.
- how_many_case_in_day # Số ca nhiễm ở Việt Nam hôm nay?
- how_many_case_on_the_world # Số ca nhiễm trên Thế giới hôm nay?
```

Sau khi đã khai báo tất cả các intents cần thiết chúng ta đến bước cài đặt phản hồi cho chatbot bằng cách thêm câu trả lời tương ứng ở phần response trong file domain.yml:

```
responses:
  utter_what_is_the_covid_19:
    - text: " COVID-19 (bệnh vi-rút corona 2019) là một bệnh do vi-rút có tên SARS-CoV-2 gây ra<br> và được phát hiện vào tháng 12 năm 2019 ở Vũ Hán, Trung Quốc.<br> Căn bệnh này rất dễ lây lan và đã nhanh chóng lan ra khắp thế giới.<br> COVID-19 thường gây ra các triệu chứng hô hấp,<br> có thể cảm thấy giống như cảm lạnh, cúm hoặc viêm phổi.<br> COVID-19 có thể tấn công không chỉ phổi và hệ hô hấp của quý vị.<br> Các bộ phận khác của cơ thể quý vị cũng có thể bị ảnh hưởng bởi căn bệnh này."
```

Sau khi đã hoàn thành việc thêm câu hỏi và câu trả lời tương ứng. Chúng ta sẽ xây dựng các câu chuyện để mô hình có thể học nhằm nâng cao khả năng nhận biết câu trả lời thông qua các mẫu chuyện có thể xảy ra trong đời sống thực tế do đó chúng ta sẽ đưa chúng vào file data/stories.yml như sau:

```
xversion: "3.0"

stories:
- story: about pet and human
  steps:
    - intent: pet_and_human
    - action: utter_pet_and_human

- story: cases in Viet Nam
  steps:
    - intent: how_many_case_in_day
    - action: action_corona_tracker_viet_nam

- story: cases on the world
  steps:
    - intent: how_many_case_on_the_world
    - action: action_corona_tracker_on_the_world
```

Trong một số trường hợp chúng ta sẽ muốn chatbot trả lời theo đúng duy nhất một hướng đã hoạch định mà không phụ thuộc vào quá trình học chúng ta sẽ thêm các intents và action vào file data/rules.yml theo đúng cú pháp ở bên dưới để chatbot có thể dựa vào đó mà trả lời theo đúng hoạch định đã đưa ra:

```
version: "3.0"

rules:

- rule: Support question anytime the user say help
  steps:
  - intent: help
  - action: utter_help
  - action: utter_question

- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye

- rule: Say 'I am a bot' anytime the user challenges
  steps:
  - intent: bot_challenge
  - action: utter_iamabot

- rule: Say No problem anytime user say thanks
  steps:
  - intent: thanks
  - action: utter_thanks
```

Do chúng ta có sử dụng đến API để thực hiện một số hành động riêng nên cần phải chỉnh sửa file actions/actions.py như sau:

```
from typing import Any, Text, Dict, List
from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher
import requests
from datetime import date, timedelta

class ActionCoronaTrackerVietNam(Action):

    def name(self) -> Text:
        return "action_corona_tracker_viet_nam"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        today = date.today()
        # yesterday = today - timedelta(days = 1)

        # url =
        f"https://api.covid19api.com/live/country/vietnam/status/confirmed/date/{yesterday}"
        url =
        "https://api.coronatracker.com/v3/stats/worldometer/country?countryCode=VN"

        payload={}
        headers = {}

        response = requests.request("GET", url, headers=headers,
data=payload).json()
        response = response[0]
        new_day = today.strftime("%d/%m/%Y")
        str = f"Tình hình covid ở Việt Nam ngày {new_day}:<br>\
        - Số ca mới hôm nay: {response['dailyConfirmed']:,}<br>\
        - Số ca tử vong hôm nay: {response['dailyDeaths']:,}<br>\
        - Tổng số ca nhiễm: {response['totalConfirmed']:,}<br>\
        - Tổng số ca tử vong: {response['totalDeaths']:,}<br>\
        - Tổng số ca đã hồi phục: {response['totalRecovered']:,}."
        dispatcher.utter_message(text=str)

        return []
```

Sau khi đã chỉnh sửa file actions/actions.py, chúng ta tiếp tục vào file domain.yml để khai báo hành động mới:

```
version: "3.0"

intents:
  - help
  - what_is_the_covid_19 #Covid 19 là gì?
  - manifestations_of_covid_19_disease # Các triệu chứng của covid 19?
  - treat_covid_19_at_home # Cách điều trị covid 19 tại nhà?
  - preventing_the_spread_of_covid_19 # Cách phòng ngừa lây nhiễm covid 19?
  - vaccine_time # Thời gian vaccine có tác dụng?
  - what_is_the_post_covid_19 # Hậu covid 19 là gì?
  - symptoms_of_post_covid_19 # Biểu hiện của hậu covid?
  - pet_and_human # Thú cưng có lây covid cho người hay không?
  - benefits_of_vaccination_against_covid_19 # Lợi ích của tiêm vắc xin
  - vaccine_safety # Tiêm vắc xin có an toàn không?
  - vaccine_side_effects # Vắc xin có tác dụng phụ không?
  - thanks # Lời cảm ơn
  - goodbye # Tạm biệt
  - bot_challenge # Bot trả lời thông tin.
  - how_many_case_in_day # Số ca nhiễm ở Việt Nam hôm nay?
  - how_many_case_on_the_world # Số ca nhiễm trên Thế giới hôm nay?

actions:
  - action_corona_tracker_viet_nam
  - action_corona_tracker_on_the_world
```

Để file actions.py có thể nạp vào model ta cần vào file endpoints.yml để thêm thuộc tính:

```
action_endpoint:
  url: "http://localhost:5055/webhook"
```


Sau khi đã hoàn tất các bước trên ta tiến hành huấn luyện mô hình bằng lệnh:

```
rasa train
```

Quá trình huấn luyện kết thúc ta kiểm tra kết quả sau khi huấn luyện bằng lệnh:

```
rasa shell
```

Sau khi đã kiểm tra mô hình hoạt động ổn định ta bắt đầu kết nối với giao diện chatbot được hỗ trợ sẵn ở đường link sau và đưa giao diện vào cùng folder của dự án Rasa:

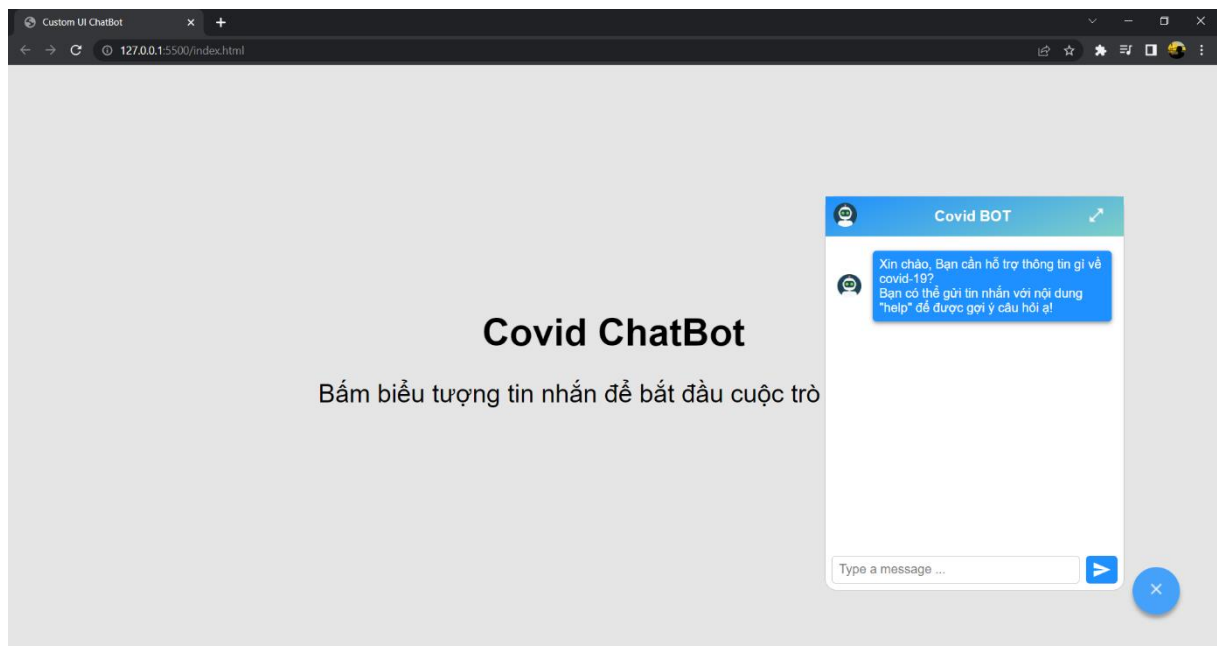
<https://elysian01.github.io/Rasa-Chatbot-UI/>

Chúng ta tiến hành chạy giao diện và kết nối với mô hình Rasa đã được huấn luyện bằng 2 lệnh như sau:

```
rasa run --enable-api --cors="*"
```

```
rasa run actions
```

Và mở phần giao diện index.html lên ta được kết quả như sau:



Hình 7. Giao diện tương tác với chatbot sau khi đã được xây dựng xong

PHẦN KẾT LUẬN

I. KẾT QUẢ ĐẠT ĐƯỢC

- Xây dựng được hệ thống chatbot hỗ trợ thông tin về covid-19.
- Hiểu thêm được về cách tổ chức và hoạt động của nền tảng Rasa.

II. HẠN CHẾ

- Số lượng câu hỏi còn quá ít, cần phải bổ sung thêm.
- Đôi khi chatbot còn trả lời nhầm câu hỏi của người dùng.

III. HƯỚNG PHÁT TRIỂN

- Cần nâng cao về số lượng và chất lượng câu hỏi.
- Cải thiện độ chính xác khi nhận diện câu hỏi.
- Có thể chuẩn đoán được người dùng có bị nhiễm covid-19 hay không thông qua các triệu chứng được cung cấp từ người dùng.

TÀI LIỆU THAM KHẢO:

- [1] Trang chủ Rasa: <https://rasa.com/docs/rasa/>
- [2] Thư viện request làm việc với API: <https://docs.python-requests.org/en/latest>