

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



**NIÊN LUẬN CHUYÊN NGÀNH  
NGÀNH KHOA HỌC MÁY TÍNH**

**Đề tài**

**CHATBOT TƯ VẤN TUYỂN SINH**

**Giáo viên hướng dẫn:  
TS. LƯU TIẾN ĐẠO**

**Sinh viên thực hiện:  
NGUYỄN VĂN LINH  
Mã số: B1609778  
Khóa: 42**

**Cần Thơ, 05/2020**

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

## LỜI CẢM ƠN

Để có được bài niên luận này, em xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến Thầy Lưu Tiến Đạo – người đã trực tiếp tận tình hướng dẫn, giúp đỡ em. Trong suốt quá trình thực hiện niên luận, nhờ những sự chỉ bảo và hướng dẫn quý giá đó mà bài niên luận này được hoàn thành một cách tốt nhất.

Em cũng xin chân thành cảm ơn bạn bè cùng với gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ trong suốt quá trình thực hiện để em có thể hoàn thành bài niên luận một cách tốt nhất.

Tuy có nhiều cố gắng trong quá trình thực hiện niên luận, nhưng không thể tránh khỏi những sai sót. Em rất mong nhận được sự đóng góp ý kiến quý báu của quý Thầy Cô và các bạn để bài niên luận hoàn thiện hơn.

Cần Thơ, ngày 15 tháng 05 năm 2020

Người viết

Nguyễn Văn Linh

---

## MỤC LỤC

<b>PHẦN GIỚI THIỆU.....</b>	<b>3</b>
1. Đặt vấn đề .....	3
2. Lịch sử giải quyết vấn đề .....	5
3. Mục tiêu đề tài.....	5
4. Phạm vi nghiên cứu .....	5
5. Phương pháp nghiên cứu .....	6
6. Bố cục .....	7
<b>PHẦN NỘI DUNG .....</b>	<b>8</b>
<b>CHƯƠNG 1 .....</b>	<b>8</b>
<b>MÔ TẢ BÀI TOÁN.....</b>	<b>8</b>
1. Mô tả chi tiết bài toán .....	8
2. Vấn đề và giải pháp liên quan đến bài toán .....	8
<b>CHƯƠNG 2 .....</b>	<b>9</b>
<b>THIẾT KẾ VÀ CÀI ĐẶT .....</b>	<b>9</b>
1. Thiết kế hệ thống .....	9
2. Xây dựng hệ thống.....	10
<b>CHƯƠNG 3 .....</b>	<b>19</b>
<b>KIỂM THỬ VÀ ĐÁNH GIÁ .....</b>	<b>19</b>
<b>PHẦN KẾT LUẬN.....</b>	<b>20</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>21</b>

---

## TÓM TẮT

Hệ thống đối thoại người máy hay còn gọi với thuật ngữ là ChatBot, ChatBot là một chương trình máy tính tiến hành cuộc trò chuyện thông qua nhắn tin nhanh, nó có thể tự động trả lời những câu hỏi hoặc xử lý tình huống. Phạm vi và sự phức tạp của ChatBot được xác định bởi thuật toán của người tạo nên chúng. ChatBot thường được ứng dụng trong nhiều lĩnh vực như thương mại điện tử, dịch vụ khách hàng, y tế, tài chính ngân hàng, các dịch vụ giải trí,...

Tiến hành xây dựng ChatBot trên máy tính có thể trả lời tự động cho sinh viên các câu hỏi liên quan đến những ngành nghề đào tạo của Đại Học Cần Thơ. Sinh viên thực hiện đặt câu hỏi cho ChatBot bằng cách chat trực tiếp. ChatBot tiếp nhận câu hỏi và sẽ truy vấn dữ liệu câu trả lời từ cơ sở dữ liệu đã được xây dựng trước đó. Các câu hỏi sẽ là dữ liệu đầu vào và nó sẽ được xây dựng trước. Các câu hỏi này là các mẫu câu hỏi phổ biến về các ngành nghề, ví dụ như “Cho em hỏi ngành CNTT cần tổ chức gì “. ChatBot này sử dụng mã nguồn mở RASA ChatBot, dữ liệu đầu vào sẽ dùng thuật toán lặp tối ưu để huấn luyện.

---

## PHẦN GIỚI THIỆU

### 1. Đặt vấn đề

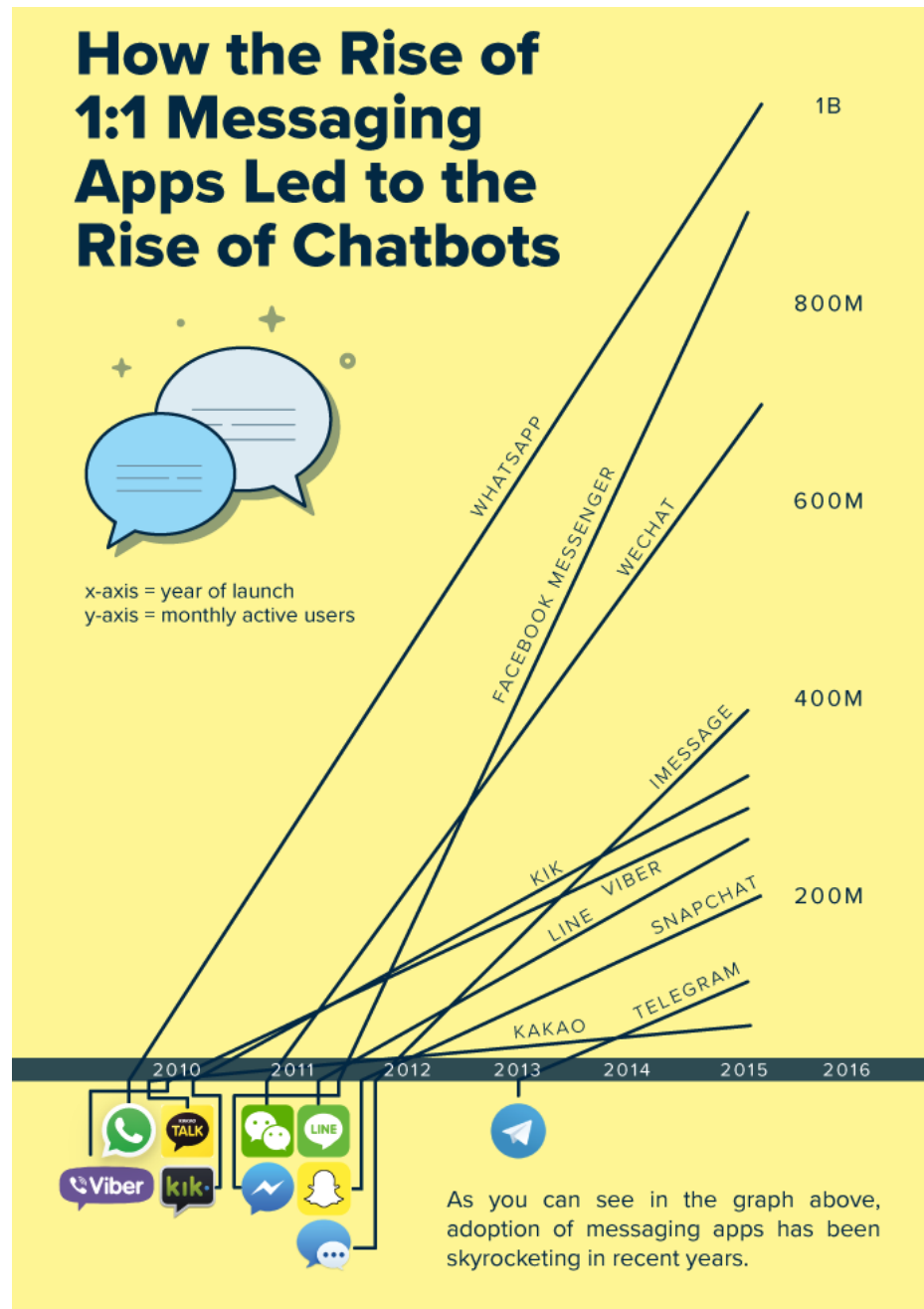
ChatBot chính là một phần mềm của trí tuệ nhân tạo. Nó có thể hoạt động độc lập và trả lời một số câu hoặc xử lý tình huống càng giống thật càng tốt. ChatBot hiện nay có thể kết hợp giữa các kịch bản có trước và sự tự học qua quá trình tương tác. Nếu câu hỏi được đặt ra ChatBot sẽ dự đoán được câu trả lời chính xác nhất có thể qua sự tự học của mình. Nói tóm lại ChatBot là một phần mềm mà bạn có thể chat với nó với những mục đích khác nhau.

Về cơ bản, ChatBot là một chương trình máy tính tiến hành cuộc trò chuyện thông qua nhắn tin nhanh hoặc âm thanh, nó có thể tự động trả lời những câu hỏi hoặc xử lý tình huống. Phạm vi và sự phức tạp của ChatBot được xác định bởi thuật toán của người tạo nên chúng. Trên thực tế, ChatBot nhanh hơn hẳn con người về tốc độ đưa ra câu trả lời. Do vậy, ChatBot thường được sử dụng rộng rãi cho nhiều mục đích khác nhau bao gồm: giải trí, nghiên cứu, dịch vụ khách hàng, quảng bá sản phẩm hoặc thu thập thông tin,... và giáo dục cũng không phải là ngoại lệ.

Với ChatBot, sinh viên có thể nhận được đáp án một cách đơn giản, chính xác và nhanh chóng trong vài giây thay vì tốn thời gian tra cứu quy chế học vụ và tìm câu trả lời.

Tại sao ChatBot lại phát triển mạnh như vậy ? do các tác nhân như :

## 1. Thời đại bùng nổ của tin nhắn



Hình 1. Sự phát triển tin nhắn

## 2. Sự phát triển của NLP và NLU

Đột nhiên cuộc “Cách mạng công nghiệp 4.0” đến và soi sáng con đường của ChatBot. Chỉ khác là bây giờ chúng ta không dùng “Búa” và “Liềm” nữa mà thay vào đó là AI và ML.

Sự phát triển vượt bậc của ứng dụng tin nhắn (Xem Hình 1).

## 2. Lịch sử giải quyết vấn đề

ChatBot đầu tiên ra đời năm 1965, tên là Eliza [1], là một chương trình máy tính của Joseph Weizenbaum [2] (Viện Công nghệ Massachusetts [3] (Mỹ)). Ban đầu Eliza được tạo ra nhằm một trị pháp điều trị tâm lý mới bằng máy tính

Trong những năm gần đây thì ChatBot thông minh nhất năm 2013 là Mitsuku ChatBot (Vô địch Loebner prize [4]).

Cùng với sự đa dụng của ChatBot, không ít dự án ChatBot hỗ trợ giáo dục đã ra đời. Trong đó phải kể đến nghiêm cứu “ChatBot for information technology students” của thầy Đỗ Thanh Nghị - trường Đại học Cần Thơ và thầy Hoàng Tùng – trường Đại học Nguyễn Tất Thành, TP.Hồ Chí Minh. Trong dự án này, ChatBot được xây dựng trên máy tính Raspberry Pi có thể trả lời tự động cho sinh viên ngành Công nghệ thông tin (CNTT) các câu hỏi liên quan đến môi trường học tập và phương pháp học tập bậc đại học, kỹ năng nghề nghiệp, và xu hướng công nghệ. Sau khi biên soạn tập dữ liệu gồm 986 câu hỏi của 213 câu trả lời, các câu hỏi trải qua quá trình tách từ, tiền xử lý và biểu diễn dưới dạng mô hình túi từ (bag-of-words [5]). Tiếp đó, tác giả huấn luyện cách bộ phân lớp như máy học véc-tơ hỗ trợ (support vector machine [6]), mạng nơ-ron (neural networks [7]), rừng ngẫu nhiên (random forest [8]), k láng giềng (k nearest neighbors [9]). Đối với phương pháp huấn luyện sử dụng giải thuật k láng giềng, độ chính xác khi dự đoán câu trả lời ở mức 65.66%.

## 3. Mục tiêu đề tài

Đề xuất xây dựng ChatBot trên máy tính có thể trả lời tự động cho học sinh trường Đại học Cần Thơ các câu hỏi liên quan đến các ngành nghề đào tạo của Đại Học Cần Thơ

## 4. Phạm vi nghiên cứu

✚ Phạm vi nghiên cứu : Đại Học Cần thơ

✚ Đối tượng nghiên cứu : Học sinh



## 5. Phương pháp nghiên cứu

Dựa trên kết quả tìm hiểu về các bộ phân lớp máy học véc-tơ hỗ trợ, mạng nơ-ron, rừng ngẫu nhiên, k láng giềng và tập câu hỏi – câu trả lời dùng làm tập dữ liệu ChatBot. Khi xét về phương diện độ phức tạp giải thuật, k láng giềng là giải thuật có thể triển khai đơn giản nhất và giải thuật k láng giềng không có bước huấn luyện dữ liệu nên tốc độ dự đoán cũng nhanh hơn. Hạn chế của k láng giềng là độ chính xác khi áp dụng trên tập dữ liệu lớn

Technique	Sức mạnh	Độ phức tạp giải thuật	Tốc độ huấn luyện dữ liệu
Máy học véc-tơ hỗ trợ	Cao	Rất khó	Rất chậm
Mạng nơ-ron	Rất cao	Rất khó	Rất chậm
Rừng ngẫu nhiên	Cao	Khó	Rất chậm
K láng giềng	Trung bình	Dễ	Nhanh

Như ta đã thấy ở trên là ưu khuyết điểm của việc sử dụng thuần các giải thuật để xây dựng con ChatBot. Ta thấy là thực khó để so sánh việc sử dụng giải thuật thuần hay mô hình như RASA , cái nào là tốt hơn .Theo ý kiến cá nhân của tôi thì tùy vào từng trường của ChatBot mà chúng ta , có thể lựa chọn giải pháp khác nhau.

Và đối với con ChatBot tư vấn tuyển sinh mà tôi xây dựng thì dữ liệu đầu vào sẽ rất đa dạng , nó cũng sẽ bị thay đổi qua các năm và dữ liệu thì cũng rất lớn . Như vậy việc sử dụng thuần giải thuật để xử lý , phải xử lý thêm NLP thì việc đó rất phức tạp và tốn nhiều thời gian . Dẫn đến độ chính xác của ChatBot cũng giảm xuống rất nhiều.

Nhưng khi dùng RASA ChatBot, độ chính xác của nó sẽ cao hơn rất nhiều đối với dữ liệu lớn. Nó sẽ huấn luyện dựa trên mô hình và phần này tôi sẽ rõ hơn trong Chương 2.

## **6. Bố cục**

### **Phần giới thiệu**

Giới thiệu tổng quát về đề tài.

### **Phần nội dung**

**Chương 1** : Mô tả bài toán.

**Chương 2** : Thiết kế, cài đặt.

**Chương 3** : Kiểm thử hệ thống và đánh giá độ chính xác.

### **Phần kết luận**

Trình bày kết quả đạt được và hướng phát triển hệ thống.

---

# PHẦN NỘI DUNG

## CHƯƠNG 1

### MÔ TẢ BÀI TOÁN

#### 1. Mô tả chi tiết bài toán

- Vấn đề quan trọng nhất của ChatBot chính là khả năng hiểu được các câu hỏi của các bạn, để có thể trả lời chính xác câu hỏi. Vậy đầu vào là gì ? để có thể đưa ra các câu trả lời đúng với yêu cầu của người hỏi.
- Đầu vào sẽ là các tập câu hỏi mẫu sẽ được xây dựng trước , các câu hỏi này chứa các ý đồ của người hỏi và ta sẽ tìm ra ý đồ của người hỏi . Để đưa ra câu trả lời đúng với yêu cầu của người hỏi
- Sau khi ta đã có dữ liệu đầu vào và xử lý được dữ liệu , ta sẽ sang bước huấn luyện dữ liệu để đưa ra đầu ra đúng với yêu cầu của người hỏi.
- Các bạn có thể xem qua ví dụ này , để hiểu tổng thể về chi tiết của bài :
  - + ChatBot sẽ trả lời được câu hỏi như :  
Người hỏi: “Cho em hỏi mã ngành của ngành công nghệ thông tin là gì ?”  
Bot trả lời: “Mã ngành của ngành Công nghệ thông tin là 123456”

#### 2. Vấn đề và giải pháp liên quan đến bài toán

##### a) Vấn đề:

- Vấn đề trọng tâm mà ta phải giải quyết được trong ChatBot, để ChatBot có thể trả lời đúng nhất câu hỏi mà người dùng hỏi. Chính là giúp ChatBot hiểu được ý đồ của câu hỏi, vậy vấn đề đó được giải quyết như thế nào ?

##### b) Giải pháp:

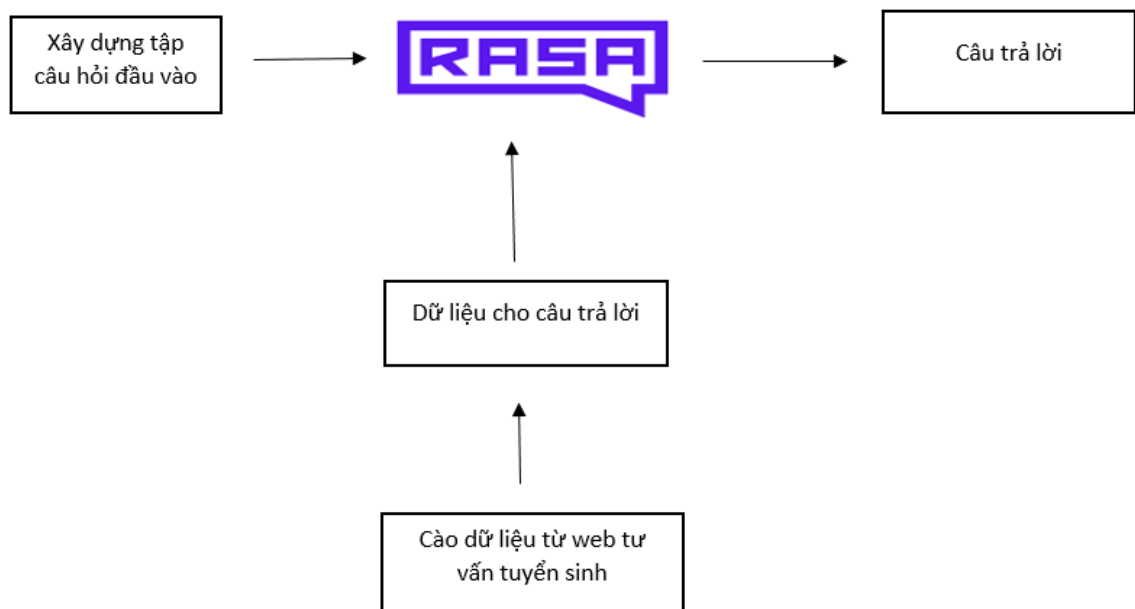
- Như ta thấy phần ví dụ của phần mô tả chi tiết bài toán ở trên câu hỏi đặt ra là “Cho em hỏi mã ngành của ngành công nghệ thông tin là gì ?”, vậy với câu hỏi này vấn đề đặt ra lớn nhất chính là làm sao để ChatBot có thể hiểu được ý đồ của câu hỏi. Để đưa ra câu trả lời đúng với yêu cầu của người dùng. Để giải quyết vấn đề này,ta sẽ xây dựng tập các câu hỏi đầu vào là các câu hỏi chứa những “entity” . Xin nói thêm “entity” trong RASA ChatBot đóng vai trò như một ý đồ , dành cho câu đầu vào ví dụ như: “ cho em hỏi mã ngành của ngành công nghệ thông tin là gì ? ”. Phân tích câu trên ta thấy rằng nó sẽ mang hai ý đồ. Thứ nhất là hỏi mã ngành, thứ hai là hỏi của ngành nào vậy ta chỉ cần lấy được hai ý đồ của câu trên thì coi như ta đã giúp ChatBot hiểu được ý đồ của câu đó. Vấn đề này được gọi là xử lý ngôn ngữ tự nhiên.

## CHƯƠNG 2

### THIẾT KẾ VÀ CÀI ĐẶT

#### 1. Thiết kế hệ thống

- Mô hình thiết kế ChatBot tư vấn tuyển sinh như sau:



Hình 2. Mô hình hoạt động ChatBot

- Xây dựng tập câu hỏi đầu vào: đây là những câu hỏi mẫu để ChatBot dựa theo những câu hỏi này để đưa ra những câu trả lời.
- RASA: đây là phần mềm mã nguồn mở hỗ trợ xây dựng ChatBot
- Dữ liệu câu trả lời: Là cơ sở dữ liệu được lấy từ phần cào dữ liệu từ trang web tư vấn tuyển sinh.
- Cào dữ liệu từ trang web tư vấn tuyển sinh: thu thập dữ liệu, để dùng xây dựng câu trả lời cho người dùng.
- Câu trả lời: đầu ra của ChatBot. (Xem Hình 2)
- Ta có thể thấy câu hỏi khi được đặt ra sẽ qua ChatBot.

## 2. Xây dựng hệ thống

### a) Yêu cầu hệ thống

- Trước khi bắt đầu thực hiện xây dựng ChatBot, máy tính cần cài đặt các thư viện python cần thiết:
  - + Yêu cầu Python 3.6 hoặc 3.7
- Cài đặt RASA:
  - + pip install RASA
  - + RASA init: Tạo dự án RASA mới
  - + sau khi chạy lệnh trên bạn sẽ có các tệp:

<code>__init__.py</code>	an empty file that helps python find your actions
<code>actions.py</code>	code for your custom actions
<code>config.yml</code> <small>“*”</small>	configuration of your NLU and Core models
<code>credentials.yml</code>	details for connecting to other services
<code>data/nlu.md</code> <small>“*”</small>	your NLU training data
<code>data/stories.md</code> <small>“*”</small>	your stories
<code>domain.yml</code> <small>“*”</small>	your assistant’s domain
<code>endpoints.yml</code>	details for connecting to channels like fb messenger

+ Như vậy là bạn đã cơ bản tạo ra một con ChatBot với sự hỗ trợ của RASA ChatBot.

b) Thực hiện

- Sau khi đã xây dựng được ChatBot cơ bản, ta sẽ tiếp tục hướng tới chức năng chính của ChatBot đó là tư vấn tuyển sinh.
- Các bước thực hiện :
  - + Bước 1: Xây dựng intent trong file data/nlu.md, các intent sẽ chứa các câu hỏi mẫu để RASA ChatBot huấn luyện dữ liệu. Nó được xem như tập dữ liệu đầu vào. Đây là ví dụ về intent mã ngành và các câu này sẽ lấy mã ngành theo tên ngành.

```
## intent:tuvan_mn
- cho em hỏi học mã ngành của ngành [Công nghệ thông tin](mn) là gì ?
- mã ngành của ngành [CNTT](mn:Công nghệ thông tin) là gì ?
- mã ngành [Sư phạm toán](mn) là gì ?
- ngành [Công nghệ kỹ thuật hóa học](mn) có mã ngành là gì ?
- ngành [Thú y](mn) thì có mã ngành là gì ?
- mã ngành của ngành [Quản trị kinh doanh](mn) là gì ?
- mã ngành [Cơ khí](mn) là gì ?
- mã ngành của ngành [Kinh tế](mn) là gì ?
- mã ngành [ck](mn) là gì ?
- mã ngành của ngành [kt hóa học](mn) là gì ?
- mã ngành của ngành [kt vat lieu](mn) là gì ?
- mã ngành của ngành [quản lý công nghiệp](mn) là gì ?
- mã ngành của ngành [quan ly cong nghiep](mn) là gì ?
```

+ Bước 2: Khai báo những “entity” đã được định trong tập câu hỏi trong intent : tuvan\_mn. Ví dụ như “mã ngành [Sư phạm toán](mn) là gì ? ”, ta có thể thấy “entity” của câu này chính là “Sư phạm toán” và nó được gán là “mn”. Ngoài ra ta cần tạo thêm “slot” để lưu trữ cái “entity” và khai báo intent “tuvan\_mn” trong intents, chúng ta sẽ xử lý những thứ nói trên trong tập tin domain.yml.

```
intents:
- tuvan_mn

entities:
- mn

slots:
mn:
auto_fill: true
type : text
```

+ Bước 3: Sau khi ta đã xây dựng xong “entity ” thì chúng ta sẽ trích xuất thực thể (entity). Nhưng khi bạn muốn trích xuất các thực thể dành riêng cho ứng dụng của mình, chẳng hạn như tên ngành, rất có thể không có mô hình được đào tạo trước. Trong trường hợp này, một giải pháp là cung cấp vô số dữ liệu đào tạo và hy vọng rằng mô hình học

cách chọn ra các thực thể tùy chỉnh của bạn. Cách tiếp cận này có nhược điểm, bởi vì việc tạo ra một loạt các ví dụ có thể sẽ tạo ra dữ liệu rất nhiều. Để giải quyết vấn đề này, ở RASA đã thêm một tính năng mới trong phiên bản 0.13.3 của RASA NLU cho phép bạn thêm các bảng tra cứu (lookup) vào dữ liệu đào tạo của mình. Các bảng tra cứu này được thiết kế để chứa tất cả các giá trị đã biết mà bạn mong muốn các thực thể của mình đảm nhận. Xin nói thêm để ChatBot có thể trả lời đa dạng các câu hỏi của người dùng, thì tôi đã xây dựng thêm các từ đồng nghĩa cho “tên ngành”.

```
Công nghệ kỹ thuật hóa học
công nghệ kỹ thuật hóa học
cong nghe ky thuat hoa hoc
kỹ thuật hóa học
kt hóa học
Kỹ thuật vật liệu
kỹ thuật vật liệu
ky thuat vat lieu
kt vat lieu
vật liệu kỹ thuật
Quản lý công nghiệp
quản lý công nghiệp
quan ly cong nghiep
ql công nghiệp
ql cong nghiep
Cơ khí
cơ khí
co khi
ck
```

+ Bước 4 : Xây dựng thêm các intent khác để hoàn thiện phần xử lý đầu vào của ChatBot. Xây dựng thêm các intent chào hỏi trong data/nlu.md, xây dựng file data/stories.md để xây dựng kịch bản chat cho ChatBot và trong file domain.yml ta xây dựng các “responses” để xây dựng các câu trả lời sẵn cho các câu hỏi đơn giản như chào hỏi.

File data/nul.md:

```
## intent:greet
- xin chào
- hi
- good morning
- good evening
- hey there
- hello
- Hello
- Hi
- chào bạn
- Xin chào

## intent:goodbye
- bye
- goodbye
- see you around
- see you later
- tạm biệt
- tạm biệt
- Tạm biệt
- hẹn gặp lại
```

File data/stories.md :

```
## happy path
* greet
  - utter_greet
* mood_great
  - utter_happy

## sad path 1
* greet
  - utter_greet
* mood_unhappy
  - utter_cheer_up
  - utter_did_that_help
* affirm
  - utter_happy
```

+ Bước 5: Sau khi chúng ta đã xử lý đủ hết phần đầu vào, chúng ta sẽ chuyển sang phần xây dựng cấu hình để RASA ChatBot có thể chạy.

- Cấu hình cho RASA ChatBot : với các file credentials.yml, endpoints.yml chúng ta lần lượt thêm như sau :

```
socketio:
  user_message_evt: user_uttered
  bot_message_evt: bot_uttered
  session_persistence: true

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

- Riêng đối với file config.yml, ta thấy nó chính là file cấu hình chính cho RASA ChatBot và nó được viết theo dạng pipeline (xử lý theo dạng đường ống) giống như bạn đã



dữ liệu trượt qua một cái ống vậy, dữ liệu sẽ được xử lý theo từng giai đoạn của cái ống.

```
# Configuration for Rasa NLU.
# https://rasa.com/docs/rasa/nlu/components/
language: en
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
```

- Xây dựng dữ liệu cho các câu trả lời, như đã nói ở phần mô hình ChatBot dữ liệu của chúng ta sẽ được cào từ trang web tư vấn tuyển sinh của Đại Học Cần Thơ. Đây là hình ảnh minh họa việc lấy dữ liệu và cơ sở dữ liệu chúng ta sẽ lưu vào HEROKU POSTGRES.

DANH MỤC NGÀNH VÀ CHỈ TIÊU TUYỂN SINH ĐẠI HỌC CHÍNH QUY NĂM 2020  
Mã trường: TCT; Tổng chỉ tiêu tuyển sinh: 8.900

1. Chương trình đào tạo đại trà

(\*) ngành đào tạo giáo viên chỉ xét tuyển theo phương thức 1 và 2

Mã ngành	Tên Ngành - chuyên ngành	Mã tổ hợp xét tuyển (Phương thức 2 và 3)	Chỉ tiêu	Tham khảo điểm trúng tuyển		
				2019	2018	2017
Nhóm ngành Công nghệ						
7510401	Công nghệ kỹ thuật hóa học	A00, A01, B00, D07	170	15,00	17,25	21,25
7520114	Kỹ thuật cơ điện tử	A00, A01	100	16,25	17,00	20,50
7520103	Kỹ thuật cơ khí, có 2 chuyên ngành: - Cơ khí chế tạo máy - Cơ khí đúc	A00, A01	240	18,75	17,50	20,50
7520201	Kỹ thuật điện	A00, A01, D07	140	16,00	16,50	20,50
7520207	Kỹ thuật điện tử - viễn thông	A00, A01	100	15,00	15,00	18,25
7520216	Kỹ thuật điều khiển và tự động hóa	A00, A01	100	16,00	16,50	19,00
7480106	Kỹ thuật máy tính	A00, A01	100	15,00	15,25	16,50
7580201	Kỹ thuật xây dựng	A00, A01	180	16,00	16,00	19,25
7520309	Kỹ thuật vật liệu	A00, A01, B00, D07	60	14,00	14,00	14,00
7580205	Kỹ thuật xây dựng công trình giao thông	A00, A01	60	14,00	14,00	18,00
7580202	Kỹ thuật xây dựng công trình thủy	A00, A01	60	14,00	14,00	15,50

Cào Dữ Liệu Và Lưu Vào  
HEROKU POSTGRES



Ta Được Bảng Dữ Liệu

**Mã Ngành :** 7510401  
**Tên Ngành :** Công nghệ kỹ thuật hóa học  
**Mã Tổ Hợp :** A00, B00, A01, D07  
**Chỉ Tiêu :** 170  
**Điểm Trúng Tuyển :** 15.00

Hình 3. Mô hình thu thập cơ sở dữ liệu

- Ngoài ra bảng dữ liệu còn được bổ sung thêm chi tiết về các ngành như: giới thiệu ngành ,ví trí việc làm, nơi làm việc, từ đồng nghĩa.

**Giới Thiệu Ngành :** Ngành Công nghệ Kỹ thuật Hóa học đào tạo Kỹ sư hóa học có kiến thức về: KT Hóa vô cơ, KT Hóa hữu cơ, KT Hóa phân tích, KT xúc tác và KT điện hóa, QT và thiết bị Công nghệ hóa học

**Vị Trí làm Việc :** Kỹ sư, cán bộ kỹ thuật: thiết kế, vận hành sửa chữa, tư vấn kỹ thuật trong lĩnh vực công nghệ hóa học, vật liệu và môi trường ở các nhà máy, công ty, xí nghiệp.

**Nơi Làm Việc :** Các Sở KH&CN, Sở tài nguyên môi trường, TT Kiểm định & Đảm bảo chất lượng, cơ quan tư vấn kỹ thuật trong lĩnh vực công nghệ hóa học, vật liệu và môi trường.

**Từ Đồng Nghĩa :** công nghệ kỹ thuật hóa học

**Từ Đồng Nghĩa :** công nghệ kỹ thuật hóa học

**Từ Đồng Nghĩa :** kỹ thuật hóa học

**Từ Đồng Nghĩa :** kt hóa học

- Như vậy việc chuẩn bị cơ sở dữ liệu đã xong và dữ liệu đầu vào (Xem Hình 3). Bây giờ chúng ta sẽ xây dựng file actions.py để kết nối dữ liệu để đưa ra câu trả lời cho người dùng, đầu tiên là kết nối với PostgreSQL để lấy dữ liệu. Tiếp theo, chúng ta sẽ xử lý dữ liệu lấy được và so với câu hỏi để tìm ra câu trả lời cho người dùng.

```
def con():
    connection = psycopg2.connect
        (user="xdjbggulyhzhck"
        ,password="d8aa4a5f3335d5489970c10428d5adbecd7c7ad5d215ec54b8d0fd8afdaf9b763"
        ,host="ec2-52-72-65-76.compute-1.amazonaws.com"
        ,port="5432"
        ,database="db3kr1a2ptuutr")
    return connection

class Actiontuvan(Action):

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

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        x = tracker.get_slot("mn")
        print(x)
        x_main = "SELECT manganh,tennganh,tdn0,tdn1,tdn2,tdn3 FROM indexjob WHERE tennganh = '"
        x_OR_0 = "' OR tdn0 = '"
        x_OR_1 = "' OR tdn1 = '"
        x_OR_2 = "' OR tdn2 = '"
        x_OR_3 = "' OR tdn3 = '"
        x_over = "';"
        x_result = x_main + x + x_OR_0 + x + x_OR_1 + x + x_OR_2 + x + x_OR_3 + x + x_over
        print(x_result)

        cursor = con().cursor()
        cursor.execute(x_result)
        records = cursor.fetchall()
        for lis in records:
            print(lis[1])
            print(lis[0])
            dispatcher.utter_message("Mã ngành của ngành {} là {}".format(lis[1],lis[0]))
        return []
```

- Và cuối cùng ta sẽ chạy các lệnh cần thiết để kiểm tra cho ChatBot. Chúng ta sẽ chạy trên localhost để kiểm tra, xin nói thêm chúng ta sẽ sử dụng trang index.html để xây dựng hộp hội thoại để người dùng nhập vào câu hỏi.
- Các lệnh cần thiết để khởi động RASA ChatBot :
  - + RASA huấn luyện: huấn luyện dữ liệu
  - + RASA run actions: để chạy file actions.py
  - + RASA run -m mô hình --enable-api --cors "\*" --debug : khởi động localhost và chạy trang index.html.

- Kết quả cuối cùng mà ta đạt được: có hai câu hỏi được đặt ra một là “hello” và “mã ngành của ngành cơ khí là gì ?” ChatBot đều trả lời được.



## CHƯƠNG 3

### KIỂM THỬ VÀ ĐÁNH GIÁ

#### A .Kiểm Thử Và Đánh Giá

- Khi ta đặt ra những câu hỏi đúng với mẫu câu hỏi đã xây dựng ,với câu hỏi: “mã ngành của ngành cơ khí là gì ?” thì ta thấy rằng độ chính xác tìm ra câu trả lời rất cao ,với gần như RASA ChatBot đều tìm ra được intent, entity, slot.

```
2020-06-11 11:17:15 DEBUG rasa.core.processor - Current slot values:
mn: cơ khí
2020-06-11 11:17:15 DEBUG rasa.core.policies.memoization - Current tracker state [{'intent_tuvan_mn': 1.0, 'entity_mn': 1.0, 'slot_mn_0': 1.0}
```

- Và khi ta đặt câu hỏi ngoài mẫu mà ta đã xây dựng, với câu hỏi: “cho tao hỏi mã ngành của thú y là như thế nào ?” thì ta thấy rằng kết quả vẫn rất chính xác, nhưng tốc độ tìm ra câu trả lời của RASA ChatBot là chậm hơn rất nhiều.

```
2020-06-11 11:39:08 DEBUG rasa.core.processor - Current slot values:
mn: thú y
2020-06-11 11:39:08 DEBUG rasa.core.processor - Logged UserUtterance - tracker now has 49 events.
2020-06-11 11:39:08 DEBUG rasa.core.policies.memoization - Current tracker state [{'intent_tuvan_mn': 1.0, 'entity_mn': 1.0, 'slot_mn_0': 1.0}
```

---

## PHẦN KẾT LUẬN

- Với ChatBot được xây dựng trên nền tảng mã nguồn mở RASA ChatBot, đã xử lý được rất nhiều về vấn đề xử lý ngôn ngữ tự nhiên.
- Về giải thuật, RASA ChatBot sử dụng pipeline cho phép bạn hiểu sâu hơn về phương pháp học máy. Nói thêm về huấn luyện dữ liệu của RASA ChatBot sử dụng keras [10], tensorflow [11] giúp cho khả năng tự học của máy tốt hơn. Như ta đã thấy ở phần kiểm thử và đánh giá, mặc dù dữ liệu đầu vào khác rất nhiều so với mẫu câu hỏi đặt ra nhưng ChatBot vẫn trả lời tốt.
- Diễn hình khi chạy lệnh “rasa train”, RASA ChatBot sử dụng tensorflow [11] mô hình để huấn luyện dữ liệu với 100 Epochs [12] : ta thấy hàm loss [13] giảm dần khá ổn định.



```
Epochs: 100% | 100/100 [00:45:00.00, 2.17it/s, t_loss=2.554, i_loss=0.489, e_loss=0.388, i_acc=1.000, e_f1=0.882]
```

---

## TÀI LIỆU THAM KHẢO

- [1] B. a. E. A. C. y. S. AbuShawar, ALICE chatbot: trials and outputs, 19.4 (2015): 625-632.
- [2] J. Weizenbaum, ELIZA—a computer program for the study of natural language communication between man and machine, Communications of the ACM 9.1, (1966): 36-45.
- [3] R. T. a. R. D. D. Forman, The ecological road-effect zone of a Massachusetts (USA) suburban highway." Conservation biology 14.1, (2000): 36-46.
- [4] M. L. Mauldin, Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition, AAAI. Vol. 94. 1994..
- [5] H. M. Wallach, Topic modeling: beyond bag-of-words, Proceedings of the 23rd international conference on Machine learning 2006.
- [6] J. A. a. J. V. Suykens, Least squares support vector machine classifiers, Neural processing letters 9.3 (1999): 293-300.
- [7] L. O. a. L. Y. Chua, Cellular neural networks: Theory, IEEE Transactions on circuits and systems 35.10 (1988): 1257-1272.
- [8] M. Pal, Random forest classifier for remote sensing classification, International journal of remote sensing 26.1 (2005): 217-222.
- [9] K. a. P. M. N. Fukunaga, A branch and bound algorithm for computing k-nearest neighbors, IEEE transactions on computers 100.7 (1975): 750-753.
- [10] F. Chollet, Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek, MITP-Verlags GmbH & Co. KG, 2018..
- [11] M. e. a. Abadi, Tensorflow: A system for large-scale machine learning, 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016.
- [12] K. Kreiner, The Postmodern Epoch of Organization Theory, International Studies of Management & Organization 22.2 (1992): 37-52.
- [13] A. e. a. Culotta, Author disambiguation using error-driven machine learning with a ranking loss function, Vancouver, Canada, Sixth International Workshop on Information Integration on the Web (IIWeb-07), 2007.



