**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

----🙞🕮🙜-----

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**CẤP TRƯỜNG NĂM 2022**

**“Nghiên cứu, thiết kế trợ lý ảo cá nhân trên nền tảng web kết hợp học máy (Machine Learning) và xử lý ngôn ngữ tự nhiên (Natural Language Processing)”**

**Sinh viên thực hiện : Nguyễn Đức Nam Bình 2018600020**

**Trần Minh Chiến 2018600805**

**Tăng Xuân Biên 2018606202**

**Lê Thanh Nga 2018606084**

**Trần Bá Hiến 2018605413**

**Giảng viên hướng dẫn : TS. Hà Thị Kim Duyên**

**Hà Nội, Năm 2022**

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

----🙞🕮🙜-----

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**CẤP TRƯỜNG NĂM 2022**

**“Nghiên cứu, thiết kế trợ lý ảo cá nhân trên nền tảng web kết hợp học máy (Machine Learning) và xử lý ngôn ngữ**

**tự nhiên (Natural Language Processing)”**

Sinh viên thực hiện: Nguyễn Đức Nam Bình

Sinh viên thực hiện: Trần Minh Chiến

Nam/Nữ: Nam

Ngành học: Công nghệ Thông tin

Lớp-Khóa: CNTT1-K21

Năm thứ: 3/3

Nam/Nữ: Nam

Ngành học: Điện Tử - Viễn Thông

Lớp-Khóa: Điện Tử 1-K13

Năm thứ:4/4

Sinh viên thực hiện: Tăng Xuân Biên

Nam/Nữ: Nam

Sinh viên thực hiện: Lê Thanh Nga

Nam/Nữ: Nữ

Ngành học: Điện Tử - Viễn Thông

Lớp-Khóa: Điện Tử 6-K13

Năm thứ:4/4

Ngành học: Điện Tử - Viễn Thông

Lớp-Khóa: Điện Tử 6-K13

Năm thứ:4/4

Sinh viên thực hiện: Trần Bá Hiến

Nam/Nữ: Nam

Ngành học: Điện Tử - Viễn Thông

Lớp-Khóa: Điện Tử 4-K13

Năm thứ:4/4

**Giảng viên hướng dẫn:** **TS. Hà Thị Kim Duyên**

**Hà Nội, 5/2022**

**LỜI CẢM ƠN**

Để hoàn thành đề tài nghiên cứu khoa học, nhóm nghiên cứu xin chân thành cảm ơn trường Đại học Công Nghiệp Hà Nội đã xây dựng, duy trì và phát triển một môi trường học tập, nghiên cứu hàng đầu. Nhóm nghiên cứu xin gửi lời cảm ơn đến các thầy cô là giảng viên dạy tại trường Đại Học Công nghiệp Hà Nội nói chung và các thầy cô trong khoa Điện tử nói riêng đã hướng dẫn và chỉ dạy cho các thành viên trong nhóm kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp nhóm nghiên cứu có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ các thành viên trong nhóm trong suốt quá trình học tập.

Nhóm nghiên cứu xin gửi lời cảm ơn chân thành đến **TS. Hà Thị Kim Duyên**, giảng viên khoa Điện tử - Trường ĐH Công nghiệp Hà Nội người đã tạo điều kiện cùng với hỗ trợ tốt nhất có thể từ hướng tìm hiểu lý thuyết đến trong quá trình triển khai và chỉ ra những thiếu sót trong đề tài.

Nhóm nghiên cứu xin chân thành cảm ơn **TS. Ngô Mạnh Tiến** – Viện Vật Lý, Viện Hàn Lâm Khoa học và Công nghê Việt Nam đã tạo điều kiện thuận lợi về cơ sở vật chất và hơn nữa những góp ý từ thầy đã giúp nhóm nghiên cứu hoàn thiện ý tưởng và định hướng để có hoàn thiện đề tài trong thời gian tiếp theo.

Cuối cùng mặc dù đã thực hiện hết sức có thể nhưng do hạn chế và kinh nghiệm cũng như kiến thức nên chắc chắn đề tài sẽ còn nhiều phần thiếu sót. Trên tinh thần học hỏi, nhóm nghiên cứu rất mong nhận được sự chỉ bảo, góp ý của Quý Thầy Cô cùng các bạn sinh viên.

Nhóm nghiên cứu xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | **Trân trọng** |
|  | Hà Nội, ngày 30 tháng 5 năm 2022 |
|  |  |

**MỤC LỤC**

[DANH MỤC VIẾT TẮT iii](#_Toc106179060)

[DANH MỤC HÌNH ẢNH iv](#_Toc106179061)

[DANH SÁCH BẢNG BIỂU v](#_Toc106179062)

[MỞ ĐẦU 1](#_Toc106179063)

[1. Tình hình nghiên cứu trong và ngoài nước 1](#_Toc106179064)

[1.1. Tình hình nghiên cứu ngoài nước 1](#_Toc106179065)

[1.2. Tình hình nghiên cứu trong nước 1](#_Toc106179066)

[2. Lý do chọn đề tài 2](#_Toc106179067)

[3. Đối tượng và phạm vi nghiên cứu 2](#_Toc106179068)

[3.1. Đối tượng nghiên cứu 2](#_Toc106179069)

[3.2. Phạm vi nghiên cứu 3](#_Toc106179070)

[4. Mục đích đề tài nghiên cứu khoa học 3](#_Toc106179071)

[5. Ý nghĩa khoa học và thực tiễn của đề tài 3](#_Toc106179072)

[6. Kết quả đạt được 3](#_Toc106179073)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI 5](#_Toc106179074)

[1.1. Cơ sở lý thuyết về máy học và học sâu 5](#_Toc106179075)

[1.1.1. Khái niệm về máy học và học sâu 5](#_Toc106179076)

[1.1.2. Quy trình thực hiện của máy học 6](#_Toc106179077)

[1.1.3. Phân loại giải thuật trong máy học 7](#_Toc106179078)

[1.2. Tổng quan về ngôn ngữ tự nhiên NLP 8](#_Toc106179079)

[1.2.1. Giới thiệu về xử lý ngôn ngữ tự nhiên (NLP) 8](#_Toc106179080)

[1.2.2. Khó khăn khi xử lý ngôn ngữ tự nhiên 9](#_Toc106179081)

[1.2.3. Ứng dụng 9](#_Toc106179082)

[1.3. Chatbot 10](#_Toc106179083)

[1.3.1. Phân loại chatbot 10](#_Toc106179084)

[1.3.2. Tổng quan về TODs 11](#_Toc106179085)

[1.3.3. Một số thách thức khi xây dựng TODs 15](#_Toc106179086)

[1.3.4. Một số nền tảng phát triển Chatbot 16](#_Toc106179087)

[1.4. Kết luận chương 1 16](#_Toc106179088)

[CHƯƠNG 2. XỬ LÝ BÀI TOÁN CHATBOT 18](#_Toc106179089)

[2.1. Một số kỹ thuật được sử dụng trong chatbot 18](#_Toc106179090)

[2.1.1. Xác định ý định trong mỗi câu nói của người dùng 19](#_Toc106179091)

[2.1.2. Trích xuất thông tin 20](#_Toc106179092)

[2.1.3. Quản lý hội thọai 21](#_Toc106179093)

[2.1.4. Mô hình sinh hội thoại cho chatbot (NLG) 23](#_Toc106179094)

[2.2. Rasa Framework 25](#_Toc106179095)

[2.2.1. Giới thiệu 25](#_Toc106179096)

[2.2.2. Cấu trúc chương trình của Rasa 26](#_Toc106179097)

[2.3. Kết luận chương 2 27](#_Toc106179098)

[CHƯƠNG 3. THIẾT KẾ, XÂY DỰNG HỆ THỐNG 28](#_Toc106179099)

[3.1. Phân tích bài toán thiết kế 28](#_Toc106179100)

[3.1.1. Phân tích yêu cầu bài toán 28](#_Toc106179101)

[3.1.2. Tiêu chí kỹ thuật của bài toán 28](#_Toc106179102)

[3.2. Thiết kế tổng quan 28](#_Toc106179103)

[3.3. Thiết kế hệ thống Chatbot 29](#_Toc106179104)

[3.4. Xây dựng chương trình 31](#_Toc106179105)

[3.5. Kết quả thực nghiệm 35](#_Toc106179106)

[3.5.1. Môi trường thực nghiệm 35](#_Toc106179107)

[3.5.2. Thiết kế 35](#_Toc106179108)

[3.5.3. Kết quả thực nghiệm 36](#_Toc106179109)

[3.6. Đánh giá 37](#_Toc106179110)

[3.7. Ứng dụng trên robot thông minh 38](#_Toc106179111)

[3.8. Kết luận chương 3 38](#_Toc106179112)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 39](#_Toc106179113)

[TÀI LIỆU THAM KHẢO 40](#_Toc106179114)

[PHỤ LỤC 41](#_Toc106179115)

###### DANH MỤC VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Chữ tiếng Anh** | **Chữ tiếng Việt** |
| 1 | AI | Artificial Intelligence | Trí tuệ nhân tạo |
| 2 | API | Application Programming Interface | Giao diện lập trình ứng dụng |
| 3 | ASR | Automatic Speech Recognition | Nhận dạng tự động giọng nói |
| 4 | CSDL |  | Cơ sở dữ liệu |
| 5 | CRF | Conditional Random Fields | Mô hình CRF |
| 6 | DL | Deep Learning | Học sâu |
| 7 | DM | Dialog Management | Quản lý hội thoại |
| 8 | DST | Dialogue State Tracker | Trình theo dõi trạng thái đối thoại |
| 9 | FAQ | Frequenly Asked Questions | Các câu hỏi thường gặp |
| 10 | NLG | Natural language generation | Sinh ngôn ngữ tự nhiên |
| 11 | NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| 12 | NLU | Natural Language Understanding | Hiểu ngôn ngữ tự nhiên |
| 13 | TTS | Text To Speech | Chuyển văn bản thành giọng nói |

###### DANH MỤC HÌNH ẢNH

[Hình 1.1 Sơ đồ minh họa tổng quan của AI, Machine Learning và Deep learning 5](#_Toc106179118)

[Hình 1.2 Sơ đồ minh hoạ một quy trình của máy học 6](#_Toc106179119)

[Hình 1.3 Sơ đồ minh hoạ học có giám sát 7](#_Toc106179120)

[Hình 1.4 Sơ đồ minh hoạ học không giám sát 8](#_Toc106179121)

[Hình 1.5 Kiến trúc của một hệ thống TODs 13](#_Toc106179122)

[Hình 2.1 Kiến trúc của hệ thống IVASTChatbot 18](#_Toc106179123)

[Hình 2.2 Mô hình xác định ý định của người 19](#_Toc106179124)

[Hình 2.3 Mô hình máy trạng thái hữu hạn (Finite – State Machine) 22](#_Toc106179125)

[Hình 2.4 Mô hình Frame-based 22](#_Toc106179126)

[Hình 2.5 Các thành phần của Rasa 26](#_Toc106179127)

[Hình 2.6 Cách thức Rasa phản hồi một tin nhắn 26](#_Toc106179128)

[Hình 2.7 Cấu trúc một chương trình của Rasa 27](#_Toc106179129)

[Hình 3.1 Sơ đồ khối của hệ thống 28](#_Toc106179130)

[Hình 3.2 Cấu trúc ý định cảm ơn của file intent.json 31](#_Toc106179131)

[Hình 3.3 Load file json và lấy ra các thông tin cần thiết để xử lý 32](#_Toc106179132)

[Hình 3.4. Sử dụng Toknizer của underthesea 32](#_Toc106179133)

[Hình 3.5 Xây dựng id cho mỗi từ 33](#_Toc106179134)

[Hình 3.6 Code xây dựng ma trận sparse cho dữ liệu ban đầu 33](#_Toc106179135)

[Hình 3.7 Code tạo ma trận có độ dài max là 20 33](#_Toc106179136)

[Hình 3.8 Tạo label cho các intent 33](#_Toc106179137)

[Hình 3.9 Code xây dựng mô hình 34](#_Toc106179138)

[Hình 3.10 Lưu model và chạy thử 34](#_Toc106179139)

[Hình 3.11 Chạy thử phân loại 34](#_Toc106179140)

[Hình 3.12 Kiến trúc chung của hệ thống 35](#_Toc106179141)

[Hình 3.13 Intent Confution matrix 36](#_Toc106179142)

[Hình 3.14 Ví dụ về đặt lịch hẹn 37](#_Toc106179143)

[Hình 3.15 Một số giao tiếp cơ bản khác 37](#_Toc106179144)

[Hình 3.16 Ứng dụng chatbot trên robot thông minh 38](#_Toc106179145)

# DANH SÁCH BẢNG BIỂU

[Bảng 2.1 Ưu và nhược điểm của chatbot dựa trên quy tắc 23](#_Toc106179146)

[Bảng 2.2 Ưu và nhược điểm của chatbot dựa trên AI 24](#_Toc106179147)

[Bảng 2.3 Giải thích các thành phần chương trình 27](#_Toc106179148)

[Bảng 3.1 Đánh giá trích chọn thông tin thực thể (entity) 36](#_Toc106179149)

# MỞ ĐẦU

Khoa học công nghệ ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lí được các công việc một cách tự động, phục vụ cho lợi ích của con người, khi đó mong muốn của con người tạo ra robot không chỉ để làm việc mà còn là một phần trong đời sống tinh thần của con người - là một người bạn đồng hành của con người, có thể tương tác giữa con người qua giọng nói, thị giác, cử chỉ. Để tương tác giữa người với robot một cách hiệu quả nhất, con người đã đặt ra các yêu cầu về tương tác giữa người với robot, người ta hy vọng rằng robot có thể hiểu được ngôn ngữ giọng nói của con người và thực hiện trao đổi với con người qua ngôn ngữ giọng nói. Vì vậy để góp phần tìm hiểu thêm về tương tác giữa robot với con người qua ngôn ngữ nên nhóm nghiên cứu đề xuất đề tài: ***“Nghiên cứu, thiết kế trợ lý ảo cá nhân trên nền tảng web kết hợp học máy (Machine Learning) và xử lý ngôn ngữ tự nhiên (Natural Language Processing)”.*** Đề tài thực hiện thiết kết một chatbot tích hợp thêm các dịch vụ Automatic Speech Recognition (ASR) và Text to Speed (TTS) để cung cấp tương tác bằng giọng nói cho chatbot trên nền tảng web.

Đề tài làm tiền để thực hiện ứng dụng, phát triển trên robot thông minh, giúp robot có thể thực hiện tương tác với con người qua ngôn ngữ giọng nói.

## 1. Tình hình nghiên cứu trong và ngoài nước

### 1.1. Tình hình nghiên cứu ngoài nước

Chatbot được ứng dụng phổ biến và mạnh mẽ trong nhiều lĩnh vực, tạo nên một cơn sốt công nghệ khi có nhiều hãng công nghệ nổi tiếng trên thế giới tham gia như Google, Facebook, Microsoft, IBM, … Theo Grand View Research, thị trường chatbot dự kiến sẽ đạt 1,25 tỷ đô la trên toàn cầu vào năm 2025 [1], đặc biệt là xu hướng chuyển dịch phát triển AI chatbot trong robot có khả năng hội thoại, xử lý tương tác thông tin với con người.

### 1.2. Tình hình nghiên cứu trong nước

Ở Việt nam, chatbot đã bắt đầu được áp dụng ở trong một số lĩnh vực như chăm sóc khách hàng mua sắm trực tuyến, trả lời thông tin ngân hàng, y tế… Đối với việc phát triển AI chatbot trong robot chưa được ứng dụng nhiều, mặc dù đây là lĩnh vực rất phù hợp cho cho các ứng dụng chatbot.

## 2. Lý do chọn đề tài

Trí tuệ nhân tạo, máy học và học sâu đang trở nên rất phổ biến ngày nay. Tất cả các công nghệ này được liên kết với nhau và mục tiêu chung là bắt chước trí thông minh của con người. Có rất nhiều ứng dụng cho các lĩnh vực này như suy luận logic (Logical Reasoning), biểu diễn tri thức (Knowledge Representation), lập kế hoạch (Planning), học tập (Learning), xử lý ngôn ngữ tự nhiên (Natural Language Processing), nhận thức (Perception), trong đó đặc biệt là trí tuệ xã hội (Social intelligence), đây cũng chính là một trong những lĩnh vực ứng dụng của AI. Sử dụng AI, chúng ta có thể xây dựng trí thông minh xã hội trong một cỗ máy hoặc robot – robot xã hội (Social Robot). Nói một cách dễ hiểu, robot xã hội (Social Robot) là bạn đồng hành cá nhân hoặc robot hỗ trợ có thể tương tác với con người bằng giọng nói, thị giác và cử chỉ. Những con robot này hoạt động giống như một con người có thể hiểu được ngôn ngữ giọng nói của con người và truyền đạt thông tin qua ngôn ngữ giọng nói.

Với những yêu cầu về tương tác giữa robot với người qua ngôn ngữ giọng nói, đề tài đề xuất một hệ thống chatbot được thiết kế nhằm phát triển, ứng dụng trên robot lễ tân IVASTBot với 3 mục tiêu: một là khả năng của robot nhận dạng ra ngôn ngữ giọng nói của con người, hai là khả năng robot có thể xử lý và đưa ra kết quả, ba là robot có thể thực hiện giao tiếp truyền đạt ngôn ngữ giọng nói.

## 3. Đối tượng và phạm vi nghiên cứu

Đối với đề tài *“Nghiên cứu, thiết kế trợ lý ảo cá nhân trên nền tảng web kết hợp học máy (Machine Learning) và xử lý ngôn ngữ tự nhiên (Natural Language Processing)”*. Việc hệ thống chatbot có thể nhận dạng, xử lý tín hiệu thông tin con người truyền đạt qua ngôn ngữ giọng nói và truyền đạt lại thông tin qua giọng nói là vấn đề quan tâm của đề tài. Đối với đề tài này, nhóm thực hiện xây dựng hệ thống chatbot thực hiện nhận dạng và xử lý thông tin con người truyền đạt qua văn bản, giọng nói và thực hiện xuất ra kết quả “Trao đổi” phù hợp.

### 3.1. Đối tượng nghiên cứu

- Nghiên cứu tổng quan về trí tuệ nhân tạo, máy học và học sâu, các ứng dụng chatbot.

- Nghiên cứu tổng quan xử lý ngôn ngữ tự nhiên (Natural Language Processing).

- Nghiên cứu, tìm hiểu và thực hiện ứng dụng chatbot vào robot thông minh, thực hiện những giao tiếp cơ bản.

### 3.2. Phạm vi nghiên cứu

Phạm vi đề tài tập trung nghiên cứu ứng dụng trí tuệ nhân tạo

* Nghiên cứu tổng quan về chatbot và cách tạo ra một chatbot
* Nghiên cứu thiết kế một hệ thống chatbot có khả năng ứng dụng vào robot thông minh.
* Nghiên cứu kỹ thuật xử lý ngôn ngữ tự nhiên.

## 4. Mục đích đề tài nghiên cứu khoa học

- Nghiên cứu tiếp cận các kiến thức về trí tuệ nhân tạo, máy học và học sâu. Các ứng dụng của chatbot trong đời sống. Tìm hiểu ứng dụng trí tuệ nhân tạo thực hiện thiết kế một hệ thống chatbot cơ bản.

- Thực hiện ứng dụng hệ thống chatbot trên robot, mong muốn robot có thể hiểu được ngôn ngữ giọng nói con người và thực hiện tương tác với con người qua giọng nói.

- Đóng góp sản phẩm có giá trị tốt đáp ứng nhu cầu xã hội.

- Truyền lửa nghiên cứu thực hành cho sinh viên học các ngành Điện tử, Công nghệ thông tin, Tự động hóa...

## 5. Ý nghĩa khoa học và thực tiễn của đề tài

**Ý nghĩa khoa học:**

Đề tài trình bày vấn đề thời sự liên quan đến vấn đề tương tác giữa người và robot qua ngôn ngữ giọng nói.

Kết quả có thể làm tài liệu tham khảo cho các lập trình viên, hoặc các học viên – học sinh trong việc nghiên cứu về xử lý và nhận dạng ngôn ngữ tự nhiên, và ứng dụng tương tác giữa robot với con người qua văn bản và ngôn ngữ giọng nói.

**Ý nghĩa thực tiễn:**

Kết quả của đề tài đề xuất giải pháp góp phần tăng hiệu quả việc tương tác giữa con người với máy móc qua văn bản và ngôn ngữ giọng nói.

## 6. Kết quả đạt được

Trong thời gian thực hiện đề tài, nhóm đã tập trung nghiên cứu về các thuật toán xử lý và nhận dạng ngôn ngữ tự nhiên. Trong quá trình nghiên cứu và thực hiện, nhóm đã đạt được một số kết quả sau:

* Tiếp cận các kiến thức về trí tuệ nhân tạo, máy học và học sâu. Tìm hiểu ứng dụng trí tuệ nhân tạo thực hiện thiết kế một hệ thống chatbot cơ bản. Các ứng dụng của hệ thống chatbot trong đời sống.
* Tìm hiểu, ứng dụng vào xử lý ngôn ngữ tự nhiên, xây dựng chatbot.
* Thực hiện tìm hiểu, ứng dụng hệ thống trên robot thông minh.
* Trong quá trình thực hiện đề tài nhóm nghiên cứu đã thực hiện tìm hiểu và đã có 1 bài báo ứng dụng hệ thống chatbot vào khả năng nhận dạng nhận dạng cảm xúc thông qua biểu cảm khuôn mặt “Ứng dụng trí tuệ nhân tạo nhận dạng cảm xúc người giao tiếp cho robot dạng người” đăng trên Tạp chí Khoa học và Công nghệ, Trường Đại học Công nghiệp Hà Nội.

# CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI

## Cơ sở lý thuyết về máy học và học sâu

### Khái niệm về máy học và học sâu

Máy học (Machine learning) [2] là một ứng dụng chuyên sâu của trí tuệ nhân tạo bao gồm các thuật toán phân giải dữ liệu, học từ các dữ liệu đã phân giải, sau đó vận dụng những điều đã học từ các dữ liệu để tự đưa ra quyết định. Máy học có liên quan nhiều đến thống kê vì đều phải phân tích dữ liệu. Khác với thống kê chỉ là đưa ra các thông số và đánh giá, Máy học sẽ phải tập trung vào sự phức tạp của các giải thuật trong quá trình thực hiện tính toán. Các bài toán của máy học giải quyết thường được phân chia làm hai loại lớn là phân loại (classification) và dự đoán (prediction).

Hình 1.1 minh họa cho ta thấy phạm vi và tổng quan cách thức thực hiện việc giải quyết bài toán của các khái niệm.



Hình 1.1 Sơ đồ minh họa tổng quan của AI, Machine Learning và Deep learning

Học sâu [2] là một mảng con được phát triển từ Máy học nhằm giải quyết nhiều bài toán phi tuyến (những bài toán phức tạp) nhờ việc kết cấu lại các thuật toán để tạo ra mạng nơ-ron nhân tạo (Artificial Neural Network) để có thể tự đưa ra quyết định của mình dựa trên yêu cầu ban đầu bài toán đưa ra. Cảm hứng tạo nên mạng nơ-ron này dựa trên cấu tạo sinh học của não người là sự liên kết giữa các nơ-ron. Tuy nhiên khác với nơ-ron sinh học có thể liên kết được với nhau tự do trong cơ thể sinh học với khoảng cách vật lý nhất định và sử dụng xung điện cùng chất hóa học để truyền dẫn thì các mạng nơ-ron nhân tạo này được chia ra thành nhiều lớp rời rạc cùng với các kết nối và hướng đi của dữ liệu trong mạng.

Các kiến trúc mạng nơ-ron gồm nhiều thành phần, nơi đối tượng được biểu diễn thành các phân lớp. Các lớp bổ sung cho phép cấu thành các tính năng từ các tầng thấp hơn bồi đắp với nhau cho khả năng tạo mô hình dữ liệu phức tạp hơn. Mạng nơ ron thường là các mạng truyền thẳng, trong đó dữ liệu đi từ lớp đầu vào đến lớp đầu ra mà không lặp lại. Trong Học sâu có thể tạo ra một mô hình để thực hiện các công việc phức tạp và tỉ mỉ mà tưởng chừng chỉ có thể con người thực hiện được như phân loại văn bản, âm thanh, hình ảnh, dữ liệu, … Mô hình Học sâu khi được triển khai với mức tối ưu hóa cao thì có thể đạt mức độ chính xác cận ngưỡng con người hoặc có thể hơn con người và mang tính khách quan. Ta có thể thấy đây là một bước tiến lớn trong việc hỗ trợ sức lao động của con người khi nó có thể giúp con người làm những việc với vai trò gần như con người.

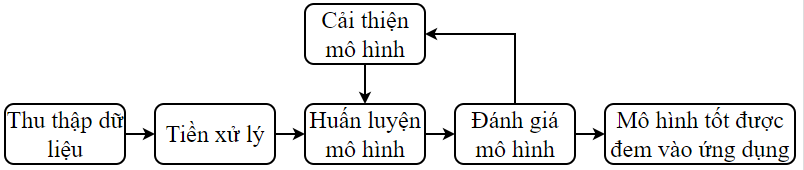
Học sâu là một dạng chuyên sâu đặc biệt hơn của Máy học. Máy học bắt đầu các dữ liệu và các thuộc tính đã được trích xuất rồi dùng các thuộc tính có sẵn đó để tạo ra mô hình giải quyết các bài toán ban đầu. Còn đối với Học sâu thì các thuộc tính đó chưa được xử lý và trích xuất thô để sử dụng “học theo quy trình đầu cuối” (end-to-end learning) thì chỉ cần cung cấp dữ liệu chưa xử lý (dữ liệu thô) và nhiệm vụ các thuật toán học sâu sẽ mang đến hiệu suất nhất định.

### Quy trình thực hiện của máy học

Hình 1.2 cho ta thấy được một quy trình thực hiện (workflow) của máy học:

Các bước thực hiện cơ bản trong một quy trình của máy học [3] theo sơ đồ trên là:

- Thu thập dữ liệu (Data collection): đây là bước ban đầu và quan trọng vì để máy tính có thể thực hiện được việc học thì ta cần một bộ dữ liệu tốt và chuẩn. Có thể tự bản thân tạo ra dữ liệu học hoặc có thể sử dụng các bộ dữ liệu phổ biến và mang tính chính thống cao trong các nghiên cứu của những người đi trước.



Hình 1.2 Sơ đồ minh hoạ một quy trình của máy học

- Tiền xử lý (Preprocessing): ở bước này là bước tốn nhiều thời gian nhất trong việc thực hiện (xấp xỉ 50% thời gian). Khi đã có dữ liệu thì ta bắt đầu rút gọn dữ liệu, xóa bỏ đi các thông số hay thuộc tính không cần thiết, giới hạn lại dữ liệu, gắn nhãn, mã hóa các đặc trưng cần thiết, … nhưng vẫn phải đảm bảo được dữ liệu đưa ra kết quả đầy ra ổn định.

- Huấn luyện mô hình (Training model): trong bước này chúng ta tạo mô hình và sử dụng dữ liệu đã được xử lý từ hai bước trên để huấn luyện mô hình.

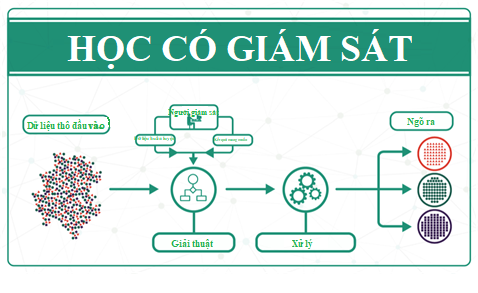
- Đánh giá mô hình (Evaluating model): bước kế tiếp sau khi huấn luyện là bước để đánh giá mô hình ta đã tạo ra. Ta dùng các thông số đo để đánh giá mô hình đã được xây dựng và tùy vào từng cách đo khác nhau sẽ cho ta biết mô hình tốt hay không tốt. Độ chính xác tùy thuộc vào ứng dụng mà ta hướng tới.

- Cải thiện mô hình (Improve): ngay sau việc đánh giá mô hình, các mô hình được tạo ra nếu có mức độ chính xác không tốt thì cần được huấn luyện lại thì ta sẽ quay lại từ bước Huấn luyện mô hình, cho đến khi mô hình đạt độ chính xác như như đã đặt ra từ trước. Mô hình tốt sẽ được đem vào các ứng dụng tương ứng đã đặt ra khi thiết kế mô hình.

### Phân loại giải thuật trong máy học

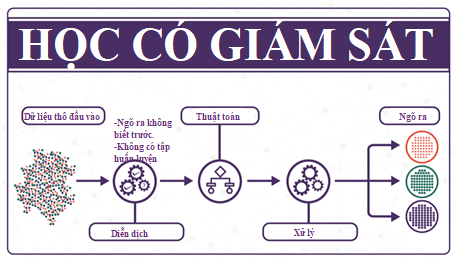
Thông thường sẽ được phân ra 2 loại chính là học có giám sát và học không giám sát.

Hình 1.3 và 1.4 minh họa 2 loại trên.



Hình 1.3 Sơ đồ minh hoạ học có giám sát

Học với việc có giám sát (Supervised learning) [3]: Cung cấp tập dữ liệu đã được dán nhãn(label) với đầu vào thì có đầu ra tương ứng (traning dataset) làm cơ sở để xây dựng ra mô hình mong muốn. Các tham số sau khi huấn luyện sẽ được dùng để thẩm định lại mô hình (testing dataset).



Hình 1.4 Sơ đồ minh hoạ học không giám sát

Học không có giám sát (Unsupervised learning) [3]: Ta chỉ cung cấp dữ liệu đầu vào, từ các dữ liệu đầu vào này máy tính phải tự tìm ra quy luật, sự tương quan, mô hình của dữ liệu, ... Có nghĩa là máy tính phải tự làm cho bản thân nó có kiến thức về dữ liệu để phân lớp, phân nhóm, tốt ưu hóa dữ liệu, … Học không giám sát có ích cho việc nén dữ liệu.

Ngoài ra máy học còn được phân thành các loại như: học bán giám sát (semi-supervised learning), học củng cố (reinforce learning), học sâu (deep learning) ...

## Tổng quan về ngôn ngữ tự nhiên NLP

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) [4] là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ - công cụ hoàn hảo nhất của tư duy và giao tiếp.

### Giới thiệu về xử lý ngôn ngữ tự nhiên (NLP)

Xử lý ngôn ngữ tự nhiên là một nhánh của trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên có thể chia thành hai nhánh lớn, không hoàn toàn độc lập, bao gồm xử lý tiếng nói (speech processing) và xử lý văn bản (text processing). Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói (dữ liệu âm thanh). Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Xử lý văn bản được chia thành hai nhánh nhỏ gồm hiểu văn bản và sinh văn bản.

### Khó khăn khi xử lý ngôn ngữ tự nhiên

Có nhiều lý do khiến xử lý ngôn ngữ tự nhiên là một nhiệm vụ khó như tập từ vựng rộng lớn và được cập nhật thường xuyên, cấu trúc ngữ pháp linh hoạt và đôi khi khá lỏng lẻo, ngôn ngữ đôi khi thể hiện cảm xúc, ẩn ý của người viết. Tuy nhiên có hai lý do cơ bản là tính nhập nhằng của ngôn ngữ (ambiguty) và sự cần thiết của tri thức nền (background knowledge).

### Ứng dụng

Có thể kể đến một loạt ứng dụng của NLP [4]:

* Nhận dạng chữ viết: nhận dạng chữ viết của con người có ứng dụng trong khoa học hình sự và bảo mật thông tin (nhận dạng chữ ký điện tử).
* Nhận dạng tiếng nói (Speech to text): nhận dạng tiếng nói rồi chuyển chúng thành văn bản tương ứng. Giúp thao tác của con người trên các thiết bị nhanh hơn và đơn giản hơn, chẳng hạn thay vì gõ một tài liệu nào đó bạn đọc nó lên và trình soạn thảo sẽ tự ghi nó ra. Đây cũng là bước đầu tiên cần phải thực hiện trong ước mơ thực hiện giao tiếp giữa con người với robot.
* Tổng hợp tiếng nói (Text to speech): tổng hợp tiếng nói tự động từ một đoạn văn bản cho trước, đây cũng là bước cuối trong giao tiếp giữa người và robot.
* Trả lời tin nhắn tự động (Text to text): ứng dụng phổ biến trong thời đại 4.0 hiện nay khi được sử dụng nhiều trong việc trả lời tin nhắn khách hàng tự động, mail tự động. Đây cũng là bước cầu nối giữa nhận dạng tiếng nói và tổng hợp tiếng nói, thực hiện hóa giao tiếp giữa người và robot, các trợ lý ảo (Google assitant, Siri, Bixby…)
* Chatbot là việc chương trình máy tính có khả năng trò chuyên (chat), hỏi đáp với con người qua hình thức hội thoại dưới dạng văn bản (text). Chatbot được sử dụng trong ứng dụng hỗ trợ khách hàng, giúp người dùng tìm kiếm thông tin sản phẩm hoặc giải đáp thắc mắc.
* Ngoài ra có thể kể thêm các ứng dụng khác như: dịch tự động, tìm kiếm thông tin, tóm tắt văn bản…

## Chatbot

“Chatbot” là thuật ngữ được mọi người biết đến nhất khi nói về một chương trình hoặc phần mềm máy tính có khả năng trò chuyện với con người bằng ngôn ngữ tự nhiên. Sẽ chính xác hơn nếu ta gọi chương trình hoặc phần mềm này là “Conversational systems” hoặc “Dialogue Systems” tuy nhiên vì tính phổ cập của thuật ngữ “Chatbot” nên trong đề tài này nhóm nghiên cứu sử dụng “Chatbot” với tên gọi chung cho tất cả các hệ thống có khả năng “đối thoại” với con người bằng ngôn ngữ tự nhiên.

Những hệ thống “Chatbot” này bao gồm [5]: Task-oriented Dialogue Systems (TODs), Intelligent Personal Assistants (IPAs), Chit-chat Dialogue Systems (CCDs). Trong đề tài này sử dụng hệ thống TODs vì đây là loại Chatbot có tính ứng dụng cao, hai loại chatbot còn lại có độ phức tạp cao và cần nhiều nguồn nhân lực mà chỉ có một số công ty lớn như Google, Apple, Microsoft, Facebook mới có khả năng triển khai.

### Phân loại chatbot

#### Task-oriented dialogue systems

TODs [5] là loại Chatbot được thiết kế để thực hiện các tác vụ cụ thể trong một lĩnh vực nhất định như: mua vé máy bay, cung cấp thông tin về đại dịch Covid-19... Chúng thường được các doanh nghiệp phát triển để cung cấp dịch vụ hỗ trợ khách hàng 24/7.

Các hệ thống TODs hiện nay đang được các doanh nghiệp ứng dụng mạnh mẽ trong nhiều lĩnh vực như tài chính, y tế, thương mại điện tử, viễn thông...chúng thường được tích hợp trên các nền tảng Chat như Messenger, WhatsApp, Slack hoặc Chat platform do doanh nghiệp tự phát triển. Một số hệ thống có khả năng giao tiếp bằng giọng nói thường được doanh nghiệp tích hợp trên các tổng đài trả lời điện thoại tự động để cung cấp các dịch vụ chăm sóc khách hàng của mình.

#### Intelligent Personal Assistans

IPAs [5] là loại Chatbot đóng vai trò như một trợ lý cá nhân để giúp người dùng thực hiện một số công việc thường ngày của họ. Ví dụ trợ lý cá nhân Siri của Apple có thể thực hiện các tác vụ như: thực hiện cuộc gọi, gửi tin nhắn, tìm quán ăn, quản lý lịch hẹn.... Tương đồng với Siri còn có một số trợ lý cá nhân khác như Cortana của Microsoft, Google Assistant, Alexa của Amazon. Đối với những yêu cầu nằm ngoài khả năng của mình, các trợ lý thường thực thi hành động Web search như một hành động thay thế.

IPAs có khả năng thực hiện nhiều tác vụ khác nhau mà không bị giới hạn trong một lĩnh vực cụ thể và nó mang lại cho người dùng cảm giác luôn bên cạnh để giúp đỡ họ khi cần. Để làm được điều này IPAs thường được tích hợp trên các thiết bị cá nhân như điện thoại thông minh, máy tính cá nhân, thiết bị smart home, đồng hồ thông minh và được cho phép quyền truy cập vào thông tin cá nhân của người dùng như: danh bạ, email, lịch, hành vi của người dùng trên thiết bị...

#### Chit-chat dialogue systems

Không giống như TODs và IPAs được thiết kế để giúp người dùng hoàn thành tác vụ một cách hiệu quả và nhanh chóng. Chit-chat dialogue systems (CDDs) được thiết kế để trở thành một người “*bạn ảo”* đồng hành lâu dài với người dùng, lắng nghe họ tâm sự và cùng thảo luận về mọi chủ đề trong cuộc sống.

Ví dụ Xiaoice Chatbot được Microsoft phát triển cho thị trường Trung Quốc và ra mắt năm 2014. Xiaoice được thiết kế mô phỏng theo tính cách của một cô gái 18 tuổi với khả năng ứng xử khéo léo trong giao tiếp (chỉ số EQ cao), hiểu biết về nhiều lĩnh vực kể cả các sự kiện vừa mới diễn ra (chỉ số IQ cao). Chính vì vậy mà Xiaoice đã trở thành người bạn tâm sự của giới trẻ tại Trung Quốc. Hiện nay Xiaoice có hơn 650 triệu người dùng và được triển khai trên hơn 450 triệu thiết bị thông minh. Đây là một trong những Chatbot thành công nhất trên thị trường hiện nay khi Xiaoice tách ra thành công ty riêng năm 2020 và được định giá lên tới 1 tỉ đô la [6].

### Tổng quan về TODs

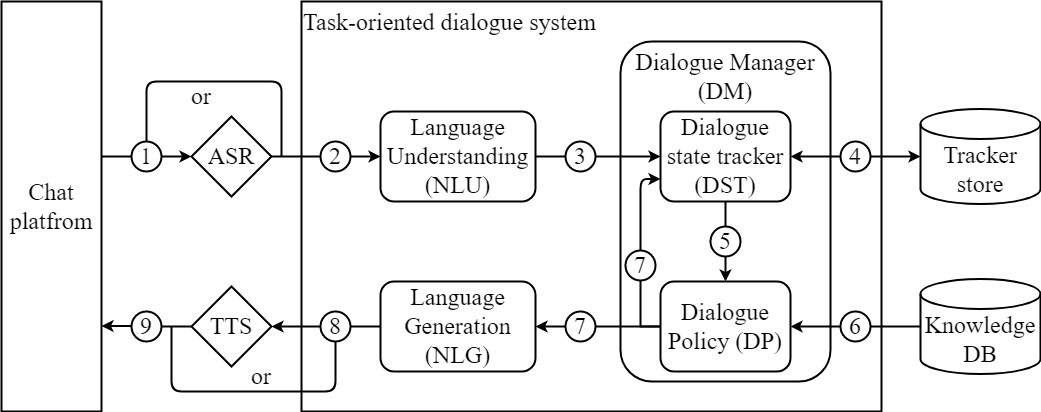
Với hướng ứng dụng của đề tài là xây dựng một hệ thống chatbot cho robot lễ tân IVASTBot thì phương pháp tiếp cận chatbot dựa trên kiến trúc Task-oriented Dialogue Systems (TODs) là phù hợp với yêu cầu “đối thoại” cơ bản giữa người và robot IVASTBot.

Có hai hướng tiếp cận cho các hệ thống TODs đó là pipeline và end-to-end. Pipeline nghĩa là hệ thống TODs sẽ được chia thành các module nhỏ, mỗi module sẽ thực hiện nhiệm vụ khác nhau, tiến trình xử lý của hệ thống sẽ tuần tự đi qua từng module. Ngược lại end-to-end nghĩa là hệ thống TODs được thiết kế với một module duy nhất để xử lý yêu cầu của người dùng. Mỗi hướng tiếp cận đều có những ưu, nhược và thách thức riêng, hầu hết các Chatbot dạng TODs hiện nay được triển khai theo hướng pipeline vì tính dễ kiểm soát và dễ dàng tích hợp vào các hệ thống đã có của doanh nghiệp hơn. End-to-end TODs là hướng nghiên cứu có nhiều tiềm năng và đang trong quá trình nghiên cứu với kỳ vọng có thể khắc phục một số nhược điểm của pipeline TODs.

Với hệ thống chatbot cho robot lễ tân IVASTBot nhóm thực hiện triển khai TODs theo hướng pipeline (Bạn đọc nếu quan tâm về end-to-end TODs có thể xem thêm tài liệu tham khảo ở mục 5).

Một số thuật ngữ quan trọng:

* Intent: ý định trong mỗi câu nói của người dùng (ví dụ: chào hỏi, tìm quán cà phê...).
* Entity: các thông tin có thể trích xuất trong câu nói của người dùng (ví dụ: số điện thoại, địa chỉ, tên quán cà phê...).
* Slot: đối tượng để ánh xạ giá trị các trường trong Database với với các thông tin mà người dùng cung cấp, ví dụ thông tin người dùng cung cấp là tìm quán cà phê để “học bài” thì mục đích “học bài” sẽ được ánh xạ vào cặp key-value là “purpose: HB” để tiện cho việc truy xuất dữ liệu.
* Action: là hành động mà Chatbot có thể làm, thể hiện khả năng của Chatbot (ví dụ: tìm quán cà phê, tìm nhà ăn...).
* Domain: là một tập các intents, entities, slots, actions. Chúng mang ý nghĩa thể hiện cho tất cả những gì Chatbot có thể hiểu (thông qua intents, entities, slots) và có thể làm(actions).
* Tracker Store: là cơ sở dữ liệu để lưu trữ lịch sử cuộc hội thoại của Chatbot với người dùng. Một số hệ thống thiết kế đơn giản sẽ lưu lịch sử trò chuyện trong RAM hoặc gộp chung với Knowledge DB.
* Knowledge DB: là cơ sở dữ liệu lưu trữ tri thức của Chatbot. Ví dụ như: thông tin về các quán cà phê, nhà hàng...



Hình 1.5 Kiến trúc của một hệ thống TODs

Kiến trúc của một pipeline TODs được mô tả trong hình 1.5, nó gồm 4 module đảm nhận các vai trò khác nhau, đầu ra của module trước sẽ là đầu vào của module sau. Ngoài ra một số hệ thống TODs có thể được tích hợp thêm các dịch vụ Automatic Speech Recognition (ASR) và Text to Speed (TTS) để cung cấp tương tác bằng giọng nói cho chatbot.

Để hiểu thêm chúng ta sẽ đi vào ví dụ về Chatbot Sani [5]:

Sani là Chatbot có khả năng đưa ra gợi ý và cung cấp thông tin về các quán cà phê ở Hà Nội. Dựa trên khả năng này ta có thể giả định Sani được thiết kế với domain như sau:

* Intents: greet (chào hỏi), request\_coffee\_shops (tìm quán cà phê), ask\_for\_review (đánh giá về quán cà phê), thanks (tạm biệt).
* Entities: location (địa điểm hoặc khu vực mà người dùng đang tìm quán cà phê), purpose (mục đích tìm quán cà phê), coffee\_shop\_name (tên quán cà phê).
* Slots: coordinates (toạ độ địa lý của entity location), purpose (mục đích tìm quán cà phê), coffee\_shop\_name (tên quán cà phê).
* Actions: utter\_greet (hành đọng đưa ra lời chào), suggest\_coffee\_shops (đưa ra gợi ý về quán cà phê), veview\_coffee\_shop (đưa ra các đánh giá về quán cà phê), utter\_bye (chào tạm biệt).

#### Natural Language Understanding (NLU)

NLU [5] là module chịu trách nhiệm chuyển đổi tin nhắn văn bản của người dùng thành dạng dữ liệu có cấu trúc đã được định nghĩa từ trước. Dạng dữ liệu có cấu trúc này chính là các Intents, Entities. Để làm được điều này NLU sẽ phải giải quyết 2 bài toán trong lĩnh vực xử lý ngôn ngữ tự nhiên là Intent Classification (IC) và Named Entity Recognition (NER) hay còn được gọi là Entity Extraction. Bài toán IC giải quyết vấn đề ánh xạ tin nhắn của người dùng vào một bộ các Intents đã định nghĩa trước. Bài toán NER giải quyết vấn đề trích xuất các thông tin có trong tin nhắn của người dùng. Ví dụ với tin nhắn “*tìm quán cà phê để hẹn hò khu vực hồ tây*” NLU sẽ ánh xạ vào intent *request\_coffee\_shops* và các entities được trích xuất: *(purpose, hẹn hò), (location, hồ tây)*. Các Intents, Entities này sau đó được tổng hợp thành định dạng như “*request\_coffee\_shops (purpose = hẹn hò, location = hồ tây)*” sau đó gửi đến module DST.

#### Dialogue State Tracking (DST)

DST [5] là module chịu trách nhiệm theo dõi và cập nhật trạng thái của cuộc hội thoại. Có 2 luồng xử lý riêng biệt trong module này bao gồm: luồng 1 cấp nhật trạng thái được kích hoạt bởi module NLU, luồng 2 cập nhật trạng thái kích hoạt bởi Dialogue Policy.

Ở luồng thứ nhất DST sẽ thực hiện các tiến trình sau:

* Cập nhật Intent và Entities được gửi tới từ NLU vào Tracker Store.
* Truy xuất lịch sử cuộc trờ chuyện từ Tracker Store bao gồm Intents, Entities, Slots, Actions trong những lượt trò chuyện trước đó.
* Dựa trên Intent và Entities tại trạng thái hiện tại và lịch sử cuộc trò chuyện thực hiện cập nhật giá trị của các Slots. Ví dụ trạng thái hiện tại có entity location = “hồ tây” cần phải cập nhật giá trị cho slot coordinates = (21.0580311,105.8063123) là toạ độ địa lý của “hồ tây" để tiện cho việc truy xuất thông tin từ Knowledge DB. Chatbot Sani làm điều này bằng cách sử dụng Google maps API.
* Cập nhật giá trị của Slots mới vào Tracker Store. Sau đó gửi trạng thái hiện tại của hệ thống cho module Dialogue Policy. Trạng thái của hệ thống được gửi tới Dialogue Policy bao gồm: Intent, Entities, Slots và lịch sử cuộc trò chuyện.

Luồng thứ 2 được kích hoạt sau khi module Dialogue Policy hoàn thành các tác vụ, chi tiết sẽ được mô tả trong phần Dialogue Policy.

#### Dialogue Policy (DB)

DB [5] là module thực hiện dự đoán hành động kế tiếp mà Chatbot cần thực hiện dựa trên trạng thái cuộc hội thoại được gửi tới từ DST. Ta có thể xem đây là bài toán Classification tương tự như bài toán IC của module NLU.

Ví dụ chatbot sani từ trạng trái của cuộc hội thoại bao gồm:

* Intent: request\_coffee\_shops
* Entities: purpose = hẹn hò, location = hồ tây
* Slots: coordinates = (21.0580311,105.8063123), purpose = HH, coffee\_shop\_name = None
* History: Intents, Entities, Actions của những lượt trước đó.

DP sẽ đưa ra dự đoán hành động kế tiếp là suggest\_coffee\_shops với Slots: coordinates = (21.0580311, 105.8063123), purpose = HH, coffee\_shop\_name = None. DP sau đó sẽ thực thi hành động suggesy\_coffee\_shops, hành động này ở đây là tìm những quán cà phê có coordinates = (21.0580211, 105.8063123) và purpose = HH trong Knowledge DB.

Sau khi thực hiện xong hành động *suggest\_coffee\_shops*, kết quả sẽ được gửi về module NLG để tạo ra câu trả lời bằng ngôn ngữ tự nhiên hiển thị cho người dùng. Đồng thời hành động mới được thực thi cũng sẽ được gửi về module DST để kích hoạt luồng 2 cập nhật trạng thái mới là hành động mà Chatbot vừa mới thực thi.

#### Natural Language Generation (NLG)

NLG [5] là module chịu trách nhiệm tạo ra câu trả lời bằng ngôn ngữ tự nhiên từ kết quả của module DP. Phương pháp truyền thống là sử dụng một bộ các mẫu câu có sẵn kết hợp với kết quả từ DP để tạo ra câu trả lời.

Hiện nay các phương pháp tiên tiến hơn sử dụng Deep learning để tạo ra câu trả lời bằng ngôn ngữ tự nhiên bạn đọc quan tâm có thể tìm hiểu thêm về phương pháp này.

### Một số thách thức khi xây dựng TODs

Có rất nhiều thách thức trong việc xây dựng Chatbot để giải quyết các vấn đề trong thế giới thực.

* Một là vấn đề thiếu dữ liệu để phân tích thiết kế domain và dữ liệu đào tạo mô hình học máy (mô hình Intent Classification, Named Entity Recognition, Dialogue Policy). Một số doanh nghiệp xây dựng Chatbot mà trước đó họ chưa có kênh chat cho người dùng. Nghĩa là không có lịch sử chat để phân tích hành vi của người dùng và nhu cầu của họ. Sẽ rất khó để có thể thiết kế Chatbot đáp ứng nhu cầu của người dùng.
* Hai là, vấn đề xây dựng Chatbot ở các lĩnh vực khác nhau. Rất khó để tận dụng những module đã xây dựng trong lĩnh vực này cho các lĩnh vực khác.
* Ba là, các vấn đề liên quan đến module NLU như vấn đề sai lỗi chính tả, viết tắt trong tin nhắn của người dùng. Hay khi người dùng spam 2,3 tin nhắn một lúc hoặc trong cùng một tin nhắn có nhiều ý định (ví dụ: xin chào, mình muốn tìm quán cà phê ở hồ tây).
* Bốn là, thiếu công cụ để theo dõi, phân tích hoạt động của chatbot trong kịch bản thực tế để phát hiện vấn đề và phát triển nó đáp ứng đúng nhu cầu của người dùng.

### Một số nền tảng phát triển Chatbot

#### Dialogflow

* Cung cấp bởi Google, được sử dụng rộng rãi trong cộng đồng chatbot
* Cung cấp cho người cách mới để tương tác bằng cách xây dựng giao diện trao đổi giựa trên giọng nói và văn bản.
* Kết nối với người dùng trên Google Assitant, Amazon, Facebook
* Có khả năng phân tích và hiểu ý định của người dùng để giúp bạn phản hồi theo cách hữu ích nhất.

#### Rasa

* Rasa Open Source là một nền tảng để tự động hóa các trợ lý dựa trên văn bản và giọng nói sử dụng học máy.
* Có thể thực hiện các hành động mà bot có thể thực hiện bằng mã python.
* Thay vì một loạt các câu lệnh if…else khác, logic của bot dựa trên một mô hình học máy được đào tạo trên các ví dụ hội thoại.

#### Chatfuel

* Chatfuel ra đời với mục tiêu làm cho việc xây dựng bot trở nên dễ dàng với bất kỳ ai.
* Không cần lập trình cũng có thể tạo ra chatbot
* Đây là nền tảng hàng đầu để xây dựng bot trên Facebook Messenger.

## Kết luận chương 1

Chương 1 đã trình bày tổng quan về cơ sở lý thuyết trí tuệ nhân tạo, máy học và học sâu, xử lý ngôn ngữ tự nhiên. Trình bảy tổng quan về chatbot - bài toán có tính ứng dụng cao và đang đạt được sự quan tâm rất lớn hiện nay, phân loại chatbots và đưa ra hướng tìm hiểu dựa trên kiến trúc Task-oriented Dialogue Systems (TODs), đưa ra một số thách thức khi thực hiện tạo ra một hệ thống chatbot.

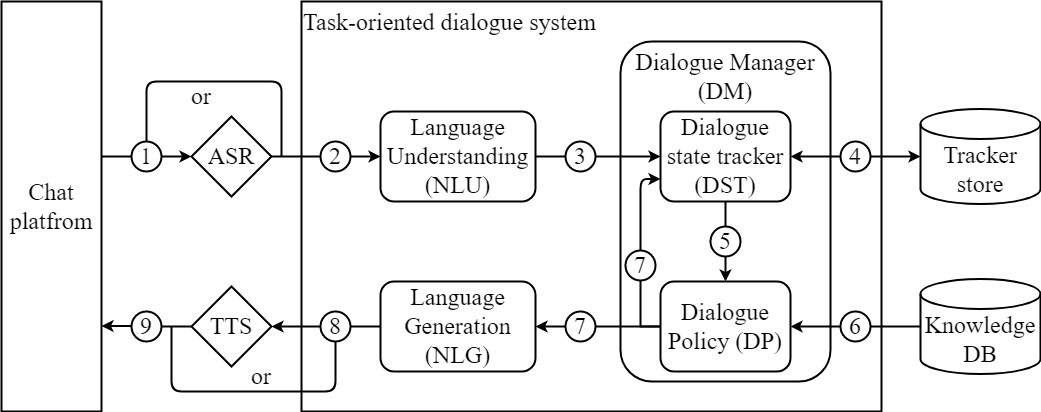
Nội dung chương 1 đề cập đến các nền tảng hỗ trợ xây dụng chatbot, trong đề tài này nhóm lựa chọn Rasa framework để xây dựng, thực nghiệm công cụ chatbot với những thuận lợi sau:

* Rasa hoạt động khá tốt và mạnh mẽ, đặc biệt là trong vấn đề xác định ý định người dùng (intent) và xác định thực thể (entity).
* Mã nguồn của Rasa là mã nguồn mở, do đó Rasa giúp ta biết chính xác được đang làm gì với chatbot của mình, thậm chí có thể tùy biến chatbot theo mong muốn của bản thân.
* Hầu hết các nền tảng lớn như Dialogflow (Google), Microsoft, … do các hãng lớn cung cấp, có các công cụ xây dựng và tích hợp thông qua API nhưng các thuật toán và dữ liệu người dùng sẽ lưu trên nền tảng này nên khó làm chủ hệ thống.

# XỬ LÝ BÀI TOÁN CHATBOT

Với mục đích của đề tài là một hệ thống Chatbot được ứng dụng trên Robot thông minh IVASTBot, từ những lý do được đề cập ở chương 1 nhóm nghiên cứu đã đề xuất thực hiện thiết kế một hệ thống IVASTChatbot dựa trên kiến trúc Task-oriented Dialogue Systems (TODs) và phát triển trên nền tảng mã nguồn mở Rasa framework. Tiếp theo đây nhóm nghiên cứu sẽ đề cập tới một số kỹ thuật sử dụng trong đề tài.

## Một số kỹ thuật được sử dụng trong chatbot



Hình 2.1 Kiến trúc của hệ thống IVASTChatbot

Chatbot giả định có khả năng đưa ra gợi ý và cung cấp thông tin về các quán karaoke ở Hà Nội. Dựa trên khả năng này ta có thể giả định Bot được thiết kế với domain như sau:

* Intents: greet (chào hỏi), request\_karaoke (tìm quán karaoke), ask\_for\_review (đánh giá về quán karaoke), thanks (tạm biệt).
* Entities: location (địa điểm hoặc khu vực mà người dùng đang tìm quán karaoke), purpose (mục đích tìm quán karaoke), karaoke\_name (tên quán karaoke).
* Slots: coordinates (toạ độ địa lý của entity location), purpose (mục đích tìm quán karaoke), karaoke\_name (tên quán karaoke).
* Actions: utter\_greet (hành động đưa ra lời chào), suggest\_karaoke (đưa ra gợi ý về quán karaoke), review\_karaoke (đưa ra các đánh giá về quán karaoke), utter\_bye (chào tạm biệt).

### Xác định ý định trong mỗi câu nói của người dùng

#### Xác định ý định (intent)

Intent: ý định trong mỗi câu nói của người dùng.

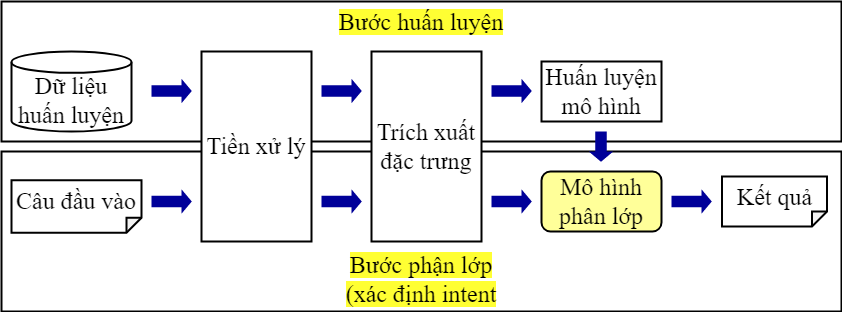
Intent được xác định sẽ quyết định cấu trúc (frame) và kịch bản (script) của đoạn hội thoại tiếp theo. Việc xác định intent là rất quan trọng đối với hệ thống Chatbot. Nếu Chatbot xác định sai intent sẽ dẫn đến phản hồi không thích hợp, dẫn đến người dùng không hài lòng và có thể rời đi.

#### Các vấn đề khi xác định ý định intent

* Thiếu nguồn dữ liệu: trong quá trình thực tế phát hiện ý định, có rất ít văn bản có mục đích với chú thích và chúng rấ khó lấy, điều này mang lại những thách thức đối với nghiên cứu và phát hiện ý định.
* Sự bất quy tắc trong diễn đạt người dùng: Trong hệ thống chat, mục đích của người dùng nói chung được đặc trưng bởi bởi cách biểu đạt giọng nói thông thường, câu ngắn và nội dung rộng, điều này gây khó khăn cho việc xác định mục đích của người dùng
* Phát hiện đa ý định: Trong câu thoại của con người sẽ có những nội dung với đa ý định.

#### Xác định ý định dựa trên máy học (machine learning)

Các bước xác định ý định của người trong câu dựa trên máy học [8] được minh họa trong hình 2.2



Hình . Mô hình xác định ý định của người

Hệ thống phân lớp ý định người dùng có một số bước cơ bản sau [8]:

* Tiền xử lý dữ liệu: Bước tiền xử lý dữ liệu là thao tác làm sạch dữ liệu như loại bỏ các thông tin thừa, chuẩn hóa dữ liệu, … Bước tiền xử lý dữ liệu có vai trò quan trọng trong hệ thống chatbot. Nếu dữ liệu đầu vào được tiền xử lý tốt sẽ làm tăng độ chính xác của hệ thống.
* Trích xuất đặc trưng (feature extraction) từ những dữ liệu đã được tiền xử lý. Bước trích xuất đặc trưng ảnh hưởng lớn đến độ chính xác của mô hình. Để trích xuất được đặc trưng tốt, chúng ta cần phân tích dữ liệu tỉ mỉ. Đây là bước biểu diễn ngôn ngữ loài người dưới dạng một số sao cho có nghĩa và máy có thể hiểu được.

Một số kỹ thuật trích dẫn đặc trưng

+ Word2Vec

+ One-hot Encoding

+ Bag of Words

+TD/IDF

* Huấn luyện mô hình: Bước huấn luyện mô hình nhận đầu vào là các đặc trưng đã được trích xuất và đầu ra áp dụng các thuật toán máy học để học ra một mô hình phân lớp. Các mô hình phân lớp có thể là các thuật toán phân lớp (nếu sử dụng decision tree) hoặc là các vector trọng số tương ứng với các đặc trưng được trích xuất (như trong các mô hình logistic regression, SVM, hay mạng Neural)

Một số kỹ thuật phân lớp:

+ Support Vector Machine (SVM)

+ Random Forest

+ Neural Network (LSTM)

Sau khi có một mô hình phân lớp intent, chúng tra có thể sử dụng nó trên một câu hội thoại mới. Câu hội thoại này cũng đi qua các bước tiền xử lý và trích xuất đặc trưng, sau đó mô hình phân lớp sẽ xác định “Độ chính xác” cho nhưng intent trong tập các intent và đưa ra intent có độ chính xác cao nhất.

### Trích xuất thông tin

#### Trích xuất thông tin thực thể (NER)

Named Entity Recognition (NER) còn gọi là nhận dạng thực thể, vai trò chính của tác vụ này là nhận dạng các cụm từ trong văn bản và phân loại chúng vào các nhóm đã được định trước.

NER là bước rất quan trọng trong chatbot, việc tách chi tiết từ vựng ra giúp ích cho bot có thể hiểu được ý nghĩa, mục đích của câu văn.

Các loại thực thể mà chatbot thường hỗ trợ: Vị trí (Location), thời gian (Datetime), Số (Number), địa chỉ liên lạc (Contact), khoảng cách (Distance), khoảng thời gian (Durantion)

#### Trích xuất thông tin thực thể dựa trên học máy

Có một số phương pháp trích xuất thông tin thực thể [8] như phương pháp tiếp cận từ vựng (lexicon approach), hệ thống dựa trên quy tắc (rula-based systems) và phương pháp dựa trên học máy cũng rất phổ biến.

Các hệ thống dựa trên học máy học cách nhận biết các thực thể trong văn bản dựa trên các ví dụ (examples) trước đây mà chúng đã thấy.

Để xây dựng một trình trích xuất thực thể, chứng ta cần cung cấp cho mô hình một khối lượng lớn dữ liệu huấn luyện có chú thích, để nó có thể tìm hiểu thực thể là gì. Chẳng hạn, nếu ta muốn xây dựng một mô hình để trích xuất các vị trí “Locations” chúng ta cần gán nhãn thủ công tên của các thành phố, quốc gia, địa điểm, vv. Càng nhiều ví dụ được gán nhãn, mô hình sẽ càng chính xác. Với phương thức này, mô hình trở nên thông minh hơn theo thời gian khi nó học hỏi từ các ví dụ mới.

Các kỹ thuật thường được sử dụng: Hidden Markow Model, Maximum Entropy, Conditional Random Fields – CRFs…

### Quản lý hội thọai

#### Vai trò của quản lý hội thoại

Trong các phiên trao đổi dài (long conversation) giữa người và chatbot, chatbot sẽ cần ghi nhớ những thông tin về ngữ cảnh (context) hay quản lý các trạng thái họi thoại (dialog state). Vấn đề quản lý hội thoại (dialoge management) khi đó là quan trọng để đảm bảo việc trao đổi giữa người và máy thông suốt [7].

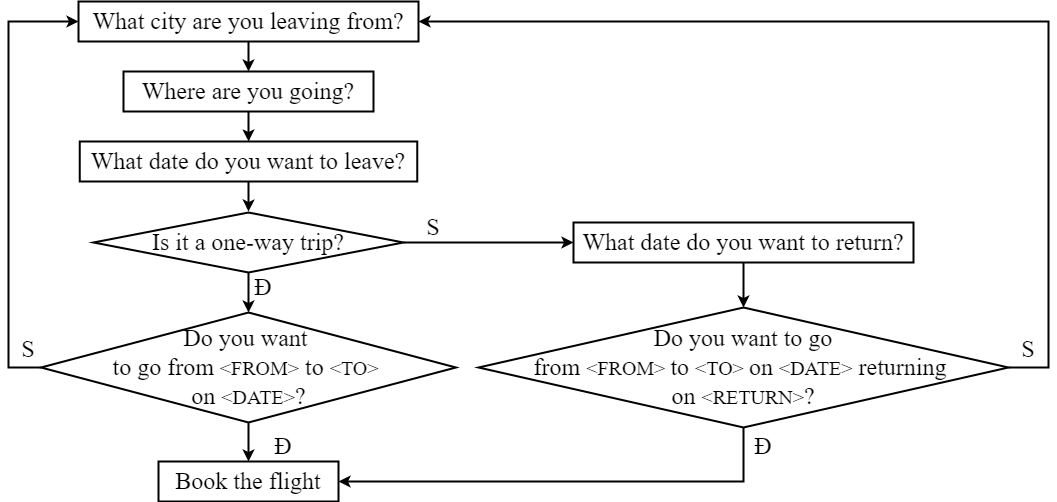
Chức năng của thành phần quản lý hội thoại là nhận đầu vào từ các thành phần NLU, quản lý các trạng thái hội thoại (dialogue state), ngữ cảnh đoạn hội thoại (dialogue context), và truyền ra cho thành phần sinh ngôn ngữ (Natural Language Generation - NLG).

Trạng thái của hội thoại (dialog state) được lưu trữ lại và dựa vào tập luật hội thoại (dialog policcy) để quyết định hành động tiếp theo cho câu trả lời của bot trong một kịch bản hội thoại, hay là hành động (action) chỉ phụ thuộc vào trạng thái (dialog state) trước của nó.

#### Các mô hình quản lý hội thoại phổ biến

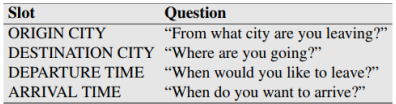
Hiện nay, các chatbot thường dùng mô hình máy trạng thái hữu hạn (Finite State Machines - FSM), mô hình Frame-based (Slot Filling) hoặc kết hợp cả hai mô hình này.

* Mô hình máy trạng thái hữu hạn (Finite – State Machines) [7]: Các FSM đặt một trình tự được xác định trước các bước đại diện cho trạng thái của cuộc hội thoại tại bất cứ điểm nào. Các FSM là phương pháp thiết kế thủ công của DM. Các trạng thái của FSM tương ứng với các câu hỏi mà DM hỏi người dùng. Các cung nói giữa các trạng thái tưng ứng với các hành động của chatbot sẽ thực hiện. Các hành động này phụ thuộc phản hồi của người dùng cho các câu hỏi.



Hình . Mô hình máy trạng thái hữu hạn (Finite – State Machine)

* Mô hình Frame-based (hoặc tên khác là Form-base) có thể được giải giải quyết vấn đề mà mô hình FSM gặp phải. Mô hình Frame-based dựa trên các frame định sẵn để định hướng cuộc hội thoại. Mỗi frame sẽ bao gồm các thông tin (slot) cần điền và các câu hỏi tương ứng mà dialogue manager hỏi người dùng. Mô hình này cho phép người dùng điền thông tin vào nhiều slot khác nhau trong frame.



Hình . Mô hình Frame-based

### Mô hình sinh hội thoại cho chatbot (NLG)

Trên thực tế, có các loại ứng dụng NLG chính: NLG dựa trên quy tắc (rule-based), dựa trên truy xuất (retrieval-based) và dựa trên cơ sở sáng tạo (generative - based)[8].

* Các chatbot dựa trên quy tắc (rule-based): Đây là kiểu kiến trúc mà hầu hết các chatbot đầu tiên đã xây dựng. Chúng chọn câu trả lời của hệ thống dựa trên một tập hợp các quy tắc được xác định trước cố định, dựa trên việc nhận ra dạng từ vựng của văn bản đầu vào mà không tạo bất kỳ câu trả lời văn bản mới nào. Kiến thức được sử dụng trong chatbot được con người viết mã bằng tay (hard-coded) và được sắp xếp và trình bày bằng các mẫu hội thoại. Cơ sở dữ liệu quy tắc toàn diện cho phép chatbot trả lời nhiều loại đầu vào của người dùng hơn. Tuy nhiên, loại mô hình này không mạnh đối với lỗi chính tả và ngữ pháp trong đầu vào của người dùng. Hầu hết các nghiên cứu hiện có về chatbots dựa trên các quy tắc đều nghiên cứu lựa chọn phản hồi cho cuộc trò chuyện một lượt, chỉ xem xét tin nhắn cuối cùng. Trong các chatbot thông minh hơn, lựa chọn phản hồi nhiều lượt xem xét tin nhắn cuối cùng. Trong các chatbot thông minh hơn, lựa chọn phản hồi nhiều lượt xem xét các phần trước của cuộc trò chuyện để chọn phản hồi phù hợp với toàn bộ bối cảnh hội thoại. Các hệ thống như vậy chủ yếu dựa vào quy tắc, điều này khiến chúng kém linh hoạt hơn NLG nâng cao.
* Cách tiếp cận dựa trên quy tắc, thông thường còn được gọi là cách tiếp cận theo hướng menu (menu-driven), trong đó các hành động được thực hiện bằng menu. Chatbot cố găng hiểu câu hỏi người dùng và sau đó trình bay một menu để chọn hành động tiếp theo. Danh sách này đảm bảo backend biết hoạt động chính xác nào cần được thực hiện yêu cầu.

Bảng 2.1 Ưu và nhược điểm của chatbot dựa trên quy tắc

|  |  |
| --- | --- |
| Ưu điểm | Nhược được |
| * Độ chính xác của phản hồi được xác nhận bởi thiết kế. * Dựa trên heuristics chứ không phải NLP phức tạp, dễ hiểu và dễ thực hiện. * Dễ mở rộng sang các mục menu mới mà không phải đào tạo lại core. | * Chức năng được giới hạn trong các mã lệnh. * Việc hoàn thành gồm hai bước: hiểu bối cạnh và đưa menu lên. Sau đó nhấp vào menu, thực hiện yêu cầu. * Nó cung cấp hạn chế các cuộc trò chuyện theo ngôn ngữ tự nhiên bởi vì chatbot không hiều các tình huống viết mã lệnh. |

* Mô hình dựa trên truy xuất (retrieval-based): một chút khác biệt so với mô hình dựa trên quy tắc là mô hình dựa trên truy xuất (retrieval-based), cung cấp tính linh hoạt hơn vì nó truy vấn và phân tích các tài nguyên có sẵn bằng cách sử dụng các API. Một chatbot dựa trên truy xuất lấy một số lựa chọn phản hồi từ một chỉ mục trước khi nó áp dụng phương pháp matching cho lựa chọn phản hồi.
* Mô hình sáng tạo (generative - based): Tạo ra câu trả lời theo cách tốt hơn so với các mô hình còn lại, dựa trên các tin nhắn hội thoại của người dùng hiện tại và trước đó. Các chatbot này giống như con người hơn và sử dụng các thuật toán máy học (machine leaning) hoặc kỹ thuật học sâu (deep learning) nên linh hoạt hơn. NLG nâng cao cho phép dự đoạn khả năng xuất hiện từ này đến từ khác và sửa các lỗi ngôn ngữ, chẳng hạn như lỗi chính tả, Các thuật toán được sử dụng trong NLG nâng cao cũng tốt hơn trong việc xử lý các từ và biểu thức mới không có trong các mẫu đào tạo ban đầu.

Phương pháp dựa trên AI này dựa trên một công cụ NLP nâng cao để hỗ trợ ngôn ngữ tự nhiên đáp và đáp ứng yêu cầu dựa trên thuật toán NLP nâng cao để hỗ trợ ngôn ngữ tự nhiên và đáp ứng yêu cầu dựa trên các thuật toán ML và tích hợp hệ thống truy xuất thông tin di động. Độ chinh xác của chatbot thấp hơn khi bắt đầu và tăng theo thời gian.

Sự khác biệt quan trong giữa cách tiếp cận dựa trên menu và dưa trên AI là NLP engine. Engine này chịu trách nhiệm trích xuất thông tin có trong đầu vào của người dùng. Hơn nữa, dựa trên thông tin trích xuất, chatbot cần quyết định các bước tiếp theo [7]

Bảng 2.2 Ưu và nhược điểm của chatbot dựa trên AI

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| * Các cuộc trò chuyện nâng cao có thể xảy ra mà không cần người dùng thực hiện nhiều bước cho các hành động. * NLP engine có thể giải quyết với các tình huống chưa biết trước và nhiều chữ. * Chatbot có thể học cách tạo ra phản hồi tùy chỉnh từ đầu (tạo ngôn ngữ tự nhiên) | * NLP engine rất phức tạp để đào tạo, bảo trì và cải tiến. * Độ chính xác của phản hồi bị ảnh hướng bởi vì đầu ra NLP không chính xác 100% * Yêu cầu một lượng lớn dữ liệu cho một NLP engine |

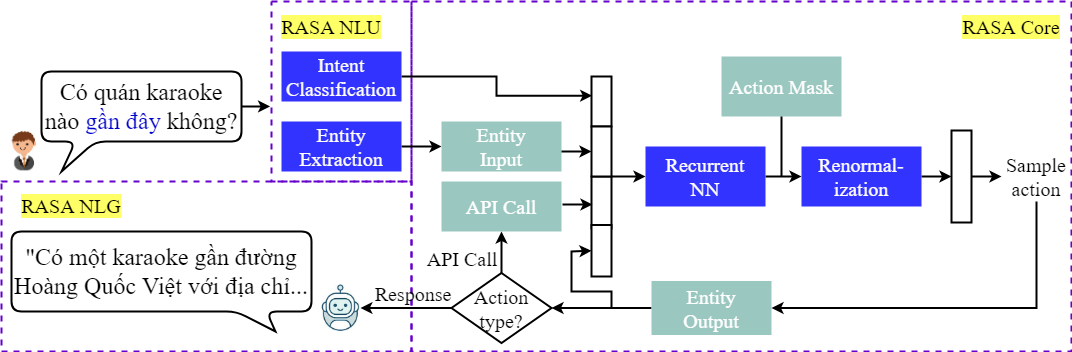
## Rasa Framework

### Giới thiệu

Rasa Open source là một nền tảng máy học để tạo ra các trợ lý ảo dựa trên văn bản và giọng nói, có khả năng hiểu ngôn ngữ tự nhiên, quản lý đối thoại và tích hợp. Rasa là bộ công cụ miễn phí được sử dụng để cải thiện các trợ lý theo ngữ cảnh được xây dựng bằng Rasa Open Source. Cùng với nhau, chúng bao gồm rất nhiều các tính năng để tạo ra các trợ lý và chatbot dựa trên văn bản và giọng nói[8]:

* Hiểu được thông điêp (Understand messages): Biến văn bản dạng tự do ở bất kỳ ngôn ngữ nào thành dữ liệu có cấu trúc. Hỗ trợ đơn và đa ý định (multiple intent) và cả các thực thể được đào tạo trước tùy chỉnh (pre-trained and custom entities).
* Duy trì cuộc trò chuyện (Hold conversations): ghi nhớ ngữ cảnh bằng cách sử dụng quản lý hội thoại dựa trên máy học.
* Học tập tương tác (Interative learning): tạo dữ liệu đào tạo bằng cách nói chuyện với chatbot của bạn và cung cấp phản hồi khi nó mắc lỗi.
* Kết nối với các nền tảng nhắn tin thường dùng (Connect): tích hợp chatbot của bạn trên Slack, Facebook, Google Home, …
* Tích hợp các lệnh gọi API (Integrate): sử dụng các hành động tùy chỉnh của Rasa để tương tác với các API và các hệ thống khác.
* Xem và chú thích cuộc hội thoại (View and annotate conversation): lọc, gán cờ và sửa các cuộc trò chuyền để liên tục cải thiện chatbot của bạn.
* Triển khai mọi nơi (Deploy): có khả năng triển khai Doker containers và điều phối để chạy Rasa on-premise hoặc thông qua nhà cung cấp đám mây ưa thích (cloud).

Rasa có đầy đủ các thành phần cơ bản của hệ thống chatbot bao gồm: NLU (hiểu ngôn ngữ tự nhiên), Dialogue Management (Quản lý hội thoại) và NLG (sinh ngôn ngữ tự nhiên) [8].



Hình . Các thành phần của Rasa

Rasa có hỗ trợ chế độ học tương tác. Trong chế độ này, ta cung cấp một phản hồi cho chatbot của mình trong khi nói chuyện với nó. Đây là một cách hiệu quả để khám phá những gì chatbot có thể làm và là cách dễ nhất để sửa chữa bất kỳ lỗi nào mà nó mắc phải. Một lợi thế của đối thoại dựa trên máy học là khi chatbot chưa biết cách làm điều gì, ta có thể dạy cho nó.

Sơ đồ hình 2.6 dưới đây cho thấy các bước cơ bản về cách một trợ lý được xây dựng bằng Rasa phản hồi một thông báo:

Diagram

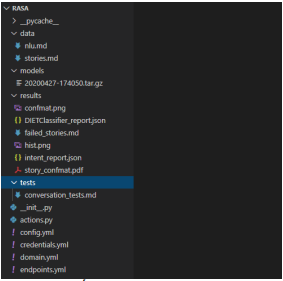
Description automatically generated

Hình . Cách thức Rasa phản hồi một tin nhắn

* Tin nhắn được nhận và chuyển để trình thông dịch (Interpreter), chuyển nó thành từ điển bao gồm văn bản gốc, ý định và cách thực thể được tìm thấy. Phần này do NLU đảm nhận.
* Tracker là đối tượng theo dõi trạng thái hội thoại. Nó nhận được thông tin rằng một tin nhắn mới đã đến.
* Chính sách (Policy) nhận trạng thái hiện tại của Tracker.
* Chính sách chọn hành động (Action) tiếp theo để thực hiện.
* Hành đông đã chọn được ghi lại bởi Tracker.
* Một phản hồi được gửi đến người dùng.

### Cấu trúc chương trình của Rasa

Cấu trúc của một chương trình của Rasa như hình dưới:



Hình . Cấu trúc một chương trình của Rasa

Các thành phần chương trình được diễn giải như sau [8]:

Bảng 2.3 Giải thích các thành phần chương trình

|  |  |
| --- | --- |
| \_\_init\_\_.py | Một file trống giúp python tìm thấy hành động của chatbot |
| actions.py | Mã cho các hành động tùy chỉnh của chatbot |
| Config.yml | Cấu hình NLU và các mô hình Core của chatbot |
| Credentials.yml | Chi tiết để kết nối với các dịch vụ khác |
| Data/nlu.md | Dữ liệu đào tạo NLU của chatbot |
| Data/stories.md | Các stories |
| Domain.yml | Miền của chatbot |
| Endpoints.yml | Chi tiết để kết nối với các kênh như facebook messenger |
| Models/<timestamp>.tar.gz | Mô hình (model) ban đầu của bạn |

## Kết luận chương 2

Trong chương 2 của đề tài, nhóm nghiên cứu đã giới thiệu các kỹ thuật quan trọng được sử dụng trong chatbot, ngoài ra cũng đề cập đến các thành phần cơ bản của Rasa Framework. Đây là các cơ sở để áp dụng xây dựng bài toán chatbot cho robot IVASTBot.

# THIẾT KẾ, XÂY DỰNG HỆ THỐNG

## Phân tích bài toán thiết kế

### Phân tích yêu cầu bài toán

Với mục đích ứng dụng của đề tài là xây dựng một hệ thống Chatbot được ứng dụng trên Robot thông minh IVASTBot, từ những lý do và một số kỹ thuật sử dụng được đề cập ở các chương trên nhóm nghiên cứu đã đề xuất thực hiện thiết kế một hệ thống IVASTChatbot dựa trên kiến trúc Task-oriented Dialogue Systems(TODs) và phát triển trên nền tảng mã nguồn mở Rasa framework. Hệ thống được thiết kế với 3 tính năng: một là khả năng của robot nhận dạng ra ngôn ngữ giọng nói của con người, hai là khả năng robot có thể xử lý và đưa ra kết quả, ba là robot có thể thực hiện giao tiếp truyền đạt ngôn ngữ giọng nói.

Thực hiện xây dựng một hệ thống chatbot dựa vào những lý thuyết và kỹ thuật được đề xuất ở các chương trên, ...

### Tiêu chí kỹ thuật của bài toán

Hệ thống IVASTChatbot có các khả năng cơ bản của các hệ thống chatbot hiện nay là nhận dạng xử lý ngôn ngữ văn bản text nhập vào từ người dùng và thực hiện phản hôi kết quả dưới dạng văn bản text, bên cạnh đó hệ thống IVASTChatbot còn thực hiện tích hợp thêm các dịch vụ Automatic Speech Recognition (ASR) và Text to Speed (TTS) để cung cấp tương tác bằng giọng nói cho IVASTChatbot.

## Thiết kế tổng quan

Robot thông minh có khả năng nhận thức thế giới bằng camera, tương tác với con người bằng giọng nói, cử chỉ và đưa ra các quyết định bằng thuật toán trí tuệ nhân tạo. Robot này có thiết kế như sơ đồ khối sau:

Diagram

Description automatically generated

Hình 3.1 Sơ đồ khối của hệ thống

Khối phần cứng gồm máy ảnh có độ phân giải cao được sử dụng để chụp các bức ảnh thời gian thực về nét mặt và cử chỉ cơ thể của người dùng; micrô được sử dụng để thu thập tín hiệu lời nói, loa để truyền đạt tín hiệu giọng nói; thiết bị truyền động giúp robot chuyển động đầu, cơ thể, cánh tay; động cơ và bánh đa hướng omni được tích hợp để điều hướng; màn hình LCD hiển thị giao diện và tương tác với người dùng.

Bên trong khối phần mềm, gồm các mô-đun nhận thức giúp xử lý dữ liệu camera và tìm kiếm các đối tượng cần thiết từ hiện trường; mô-dun nhận dạng/tổng hợp giọng nói giúp giao tiếp với người dùng; mô-đun trí tuệ nhân tạo; mô-đun điều khiển robot để điều khiển thiết bị truyền động; nút quyết định kết hợp tất cả dữ liệu từ các cảm biến và đưa ra quyết định cuối cùng về việc phải làm tiếp theo Hệ điều hành ROS kết nối tới các cảm biến, bộ truyền động. Khối GUI giúp giao tiếp giữa người dùng với robot thông qua thao tác với hình ảnh trên bảng điều khiển LCD.

## Thiết kế hệ thống Chatbot

Nhòm nghiên cứu thực hiện xây dựng hệ thống chatbot với chức năng như một lễ tân, đề tài tập trung vào một số nội dung sau.

**Mô hình huấn luyện cho chatbot**

Trong Rasa, các messages được xử lý bởi một chuỗi các thành phần (component). Các thành phần này được thực thi lần lượt trong “pipeline” được xác định trong file config.yml. Việc chọn một NLU pipeline cho phép ta tùy chỉnh mô hình của mình.

Có các component để trích xuất thực thể, để phân loại ý định, lựa chọn phản hồi, tiền xử lý và các thành phần khác. Nếu muốn thêm thành phần của riêng mình, chẳng hạn như để chạy kiểm tra chính tả (spell-check) hoặc để phân tích quan điểm (sentiment analysis), có thể thực hiện custom component.

Một pipeline bao gồm 3 phần chính:

* Tokenization:

Tách mỗi câu thành một danh sách từ tố (token), mỗi câu được tách ra thành một danh sách các từ có nghĩa. Đối với ngôn ngữ tiếng việt, các từ được phân tách bằng dấu cách, ở đây nhóm nghiên cứu lựa chọn tách bằng WhutespaceTokenizer.

* Featurization:

Ta cần quyết định xem có nên sử dụng các thành phần cung cấp tính năng nhúng từ được đào tạo từ trước (pre-trained word embedding) hay nhúng được giám sát (supervising Embeddings).

+Pre-trained Embeddings: Phân loại ý định người dùng sẽ dựa trên các tập dữ liệu lọc trước, sau đó sử dụng để hiện tương ứng từng thông điệp người dùng dưới dạng từ nhúng hay biểu diễn ngôn ngữ dưới dạng vector. Lợi thế của việc sử dụng tính năng nhúng từ được đào tạo trước là nếu ta có một ví dụ đào tạo như: “Tôi muốn mua táo” và Rasa được yêu cầu dự đoán ý định cho “lấy lê”, thì mô hình của ta đã biết rằng từ “táo” và “lê” rất giống nhau. Điều này đặc biệt hữu ích nếu không có đủ dữ liệu đào tạo.

+ Supervised Embedding: Với phương pháp nhúng được giám sát này thì ta sẽ tự tạo tập dữ liệu training riêng của mình từ đầu. Với phương pháp nhúng được giám sát, Rasa có khả năng huấn luyện với bất kì ngôn ngữ nào (bao gồm tiếng việt), vì training lại mọi thứ từ đầu, chỉ phụ thuộc vào dữ liệu huấn luyện. Với việc khó tìm ra được mô hình đào tạo trước cho ngôn ngữ tiếng việt, cùng với bài toán trong một miền lĩnh vực đóng như trả lời thông tin thì nó sẽ đảm bảo rính chính xác hơn nhiều và tránh dư thừa dữ liệu. Do đó, ở đây nhóm nghên cứu lựa chọn phương pháp này cùng một số thành phần cụ thể là

+ CountVectorsFeaturizer: Trích xuất đặc trưng cho phân loại ý định và lựa chọn phản hồi, tạo túi từ (BoW: bag-of-words) đại diện cho tin nhắn người dùng, ý định và phản hồi.

RegexFeaturizer: Tạo biểu diễn vecto của các thông điệp người dùng bằng cách sử dụng biểu thức chính quy (regular expressions)

+LexicalSyntacticFeaturizer: Tạo các đặc trưng từ vựng và cú pháp cho tin nhắn của người dùng để hỗ trợ và trích xuất thực thể.

* Entity Recognition/ Intent Classification/ Response Selectors:

Tùy thuộc vào dữ liệu, ta có thể chỉ muốn thực hiện phân loại ý định, nhận dạng thực thể hoặc lựa chọn phản hồi. Hoặc ta có thể muốn kết hợp nhiều nhiệm vụ đó. Rasa hỗ trợ một số thành phần cho mỗi nhiệm vụ. Trong Rasa, việc nhận diện ý định thông thường sử dụng mô hình máy vector hỗ trợ (Support Vector Machines – SVM), trích xuất thông tin thực thể sử dụng mô hình ngẫu nhiên có điều khiện (Conditional Random Fields - CRF). Ở đây nhóm nghiên cứu lựa chọn các thành phần như sau:

+ DIETClassifier: DIET (Dual Intent and Entity Transformer) là một kiến trúc đa tác vụ để phân loại ý định và nhận dạng thực thể. Kiến trúc dựa trên một bộ chuyển đổi đươc chia sẻ cho cả hai nhiệm vụ. Mỗi chuỗi các nhãn thực thể được dự đoán thông qua một lớp gán thẻ trường ngẫu nhiên có điều kiện (Conditional Random Field - CRF) tương úng với chuỗi đầu vào của tokens. Đối với nhãn ý định, đầu ra bộ chuyển đổi cho \_\_CLS\_\_ token và nhãn ý định được nhúng vào một không gian vector ngữ nghĩa duy nhất. DIETClassifier cũng hỗ trợ đa ý định (multi-intent) tách các Intent thành nhiều nhãn.

+ResponseSelector: Thành phần này có thể được sử dụng để xây dựng mô hình truy xuất phản hồi nhằm dự đoán trực tiếp các phản hồi của bot từ tập hợp các phản hồi. Dự đoán cỏa mô hình này được sử dụng bời Retrieval Actions. Nó nhúng đầu vào của người dùng và nhãn phản hồi vào cùng một không gian và tuân theo cùng một kiến trúc và tối ưu hóa mạng thần kinh giống hệt như DIETClassifier.

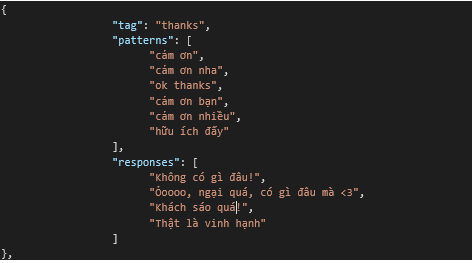
Sau khi xây dựng xong mô hình và tạo một số dữ liệu đào tạo NLU, có thể huấn luyện (train) mô hình với rasa train nlu. Sau khi quá trình huấn luyện kết thúc, ta có thể kiểm tra (test) khả năng của mô hình diễn giải các thông điệp đầu vào khác nhau qua rasa shell nlu.

## Xây dựng chương trình

#### Xây dựng dữ liệu

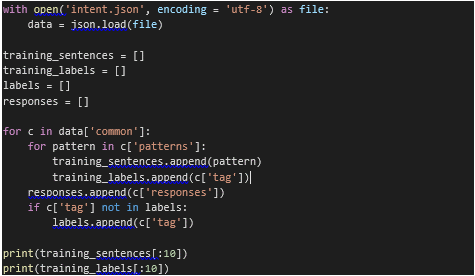
Tạo một file json có các thành phần chính là tag, patterns và reponses. Trong đó tag là ý định, patterns là các ví dụ về các câu nói thuộc ý định đó, reponses là các câu có thể trả lời nếu người dùng hỏi một câu có ý định tương ứng.

Đây là cấu trúc ý định cảm ơn của file intent.json:



Hình 3.2 Cấu trúc ý định cảm ơn của file intent.json

Khi có được file json tổng hợp các dữ liệu cần thiết, ta cần phải load file và lấy ra những thông tin cần thiết trước khi tiếp tục xử lý.



Hình 3.3 Load file json và lấy ra các thông tin cần thiết để xử lý

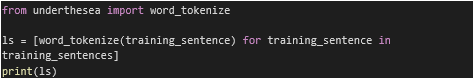
#### Lựa chọn Tokenizer

Vì đặc thù của từng ngôn ngữ nên không thể dùng chung Tokenizer cùng các ngôn ngữ khác, nên mỗi ngôn ngữ nên có một bộ Tokenizer thích hợp cho nó. Nhiệm vụ của Tokenizer sẽ cắt các từ hoặc cụm từ trong câu.

Phần này nhóm nghiên cứu sử dụng Tokenizer của underthesea để được hiệu quả tốt, bạn có thể sử dụng các Tokenizer khác như VNcoreNLP, pyvi, coccoc-tokenizer. Một câu sau khi được tokenize sẽ thành một danh sách như sau:

**“tôi cảm thấy cực kỳ khỏe mạnh”**

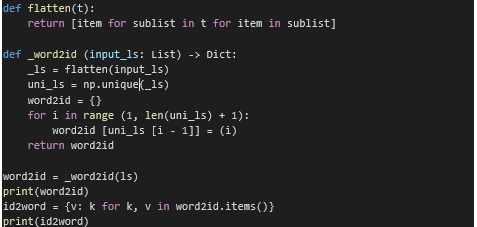
**['tôi', 'cảm thấy', 'cực kỳ', 'khỏe mạnh']**



Hình 3.4. Sử dụng Toknizer của underthesea

#### Xây dựng id cho mỗi từ

Để có thể train mô hình, trước hết ta cần phải những từ trong dữ liệu có được thành những số để có thể đưa vào mô hình:



Hình 3.5 Xây dựng id cho mỗi từ

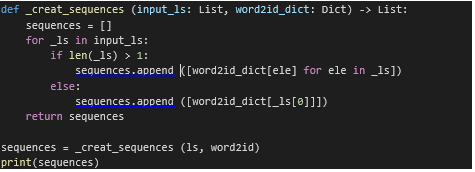
Kết quả của 2 Dictionary word2id và id2word có dạng như sau:

{',': 1, 'ai': 2, 'anh': 3, 'bao giờ': 4, 'biết': 5, …}

{1: ',', 2: 'ai', 3: 'anh', 4: 'bao giờ', 5: 'biết', …}

#### Xây dựng ma trận sparse cho dữ liệu ban đầu

Khi đã có id tương ứng với từng từ, ta thay thế vào dữ liệu ban đầu:

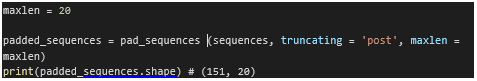


Hình 3.6 Code xây dựng ma trận sparse cho dữ liệu ban đầu

Kết quả:

[[15], [15, 10], [128, 15], [15, 3], [15, 17], [15, 28], …]

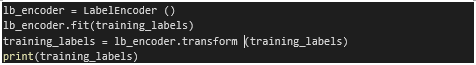
Tạo ma trận có độ dài max là 20



Hình 3.7 Code tạo ma trận có độ dài max là 20

#### Tạo label cho các intent

Sử dụng Label Encoder trong thư viện của Keras

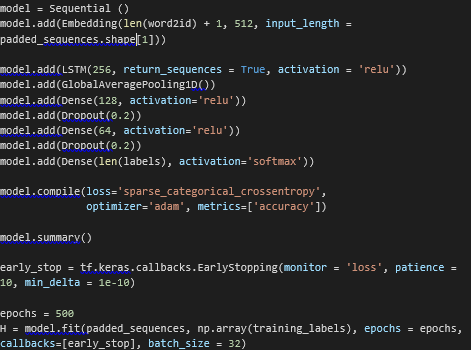


Hình 3.8 Tạo label cho các intent

#### Xây dựng mô hình

Xây dựng một model LSTM đơn giản với đầu vào là Sparse matrix đi qua lớp Embedding và kết thúc ở lớp FullyConnected có hàm kích hoạt là softmax

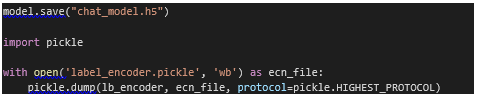
Mô hình được xây dựng với loss là sparse\_categorical\_crossentropy và sử dụng EarlyStopping để lưu lại mô hình tốt nhất



Hình 3.9 Code xây dựng mô hình

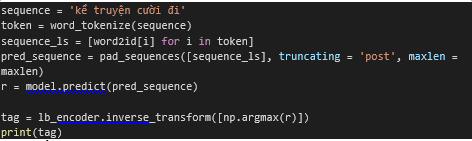
#### Lưu model và chạy thử

Lưu model và LabelEncoder lại



Hình 3.10 Lưu model và chạy thử

Chạy thử phân loại một câu xem đã đúng lớp intent chưa



Hình 3.11 Chạy thử phân loại

Kết quả:

1/1 [==============================] - 0s 56ms/step

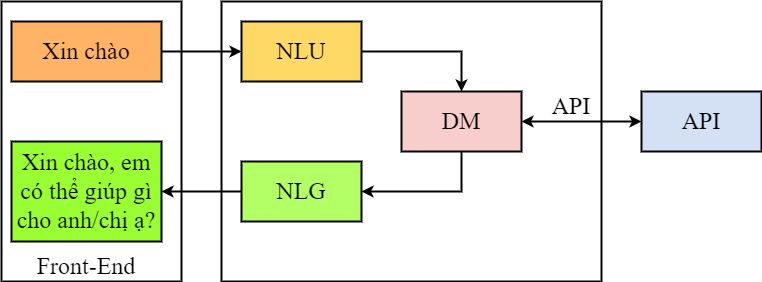
['jokes']

## Kết quả thực nghiệm

### Môi trường thực nghiệm

Chương trình thử nghiệm được thiết kế, xây dựng và thực hiện trên môi trường hệ điều hành Windows với nền tảng framework Rasa, dựa trên ngôn ngữ lập trình python. Giao diện người dùng sử dụng nền tảng web/ứng dựng chat.

### Thiết kế giao diện tương tác với người dùng



Hình 3.12 Kiến trúc chung của hệ thống

Front-end sử dụng giao diện web hoặc các trình nhắn tin phổ biến message. Với mục tiêu minh họa, ở đây nhóm nghiên cứu sử dụng giao diện web.

* Mỗi khi có một người dùng gửi tin nhắn cho chatbot thì nội dung tin nhắn này sẽ gửi một POST request để webhook được sử dụng để lắng nghe sự kiện. Webhook này sẽ chuyển đển bộ NLU của RASA
* RASA nhận diện ý định, sau khi đã thu được message của người dùng thì sử dụng RASA để hiểu ý định của người dùng cùng các thông tin thực thể
* Thông tin này tiếp tục chuyển đến DM của RASA, tại đây tùy theo ý định và thông tin thực thể cùng với các thông tin theo dõi của cuộc trò chuyện đã xảy ra cho đến nay, để dự đoán một phản ứng thích hợp, bao gồm cả việc gọi API để lấy thông tin trả lời.
* NLG sinh ra câu trả lời dựa vào dữ liệu từ thành phần DM theo các mẫu câu template đã được xây dựng trước hoặc là kết qua của API.
* Gửi tin nhắn qua phản hồi trả về cho người dùng.

### Kết quả thực nghiệm

#### Kết quả thử nghiệm

Kết quả đánh giá NLU model và Rasa Core sau khi thực hiện đào tạo chatbot và kiểm tra trên dữ liệu test, dữ liệu người dùng nhập vào:

Chart

Description automatically generated

Hình 3.13 Intent Confution matrix

Bảng 3.1 Đánh giá trích chọn thông tin thực thể (entity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precition | Recall | F1-score | support |
| Room\_type | 0.85 | 0.75 | 0.79 | 8 |
| location | 1.00 | 0.58 | 0.74 | 17 |

Tính chung, kết quả test trên tập dữ liệu cho độ chính xác khoảng 81%

Correct: 131/214

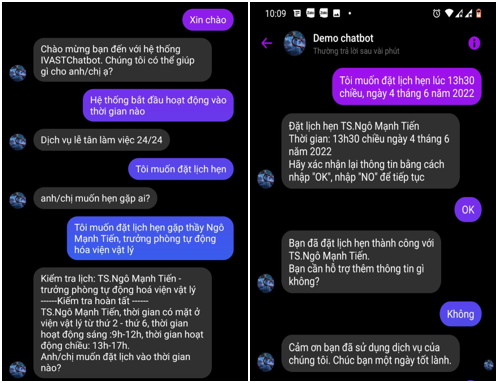
F1-Score: 0.906

Precision: 0.926

Accuracy: 0.899

#### Kiểm thử trên giao diện người dùng

Thực hiện thử nghiệm tương tác với chatbot trên nền tảng web messenger qua một số câu hỏi gần với kịch bản đã đào tạo cho chatbot.



Hình . Ví dụ về đặt lịch hẹn

Người dùng cũng có thể thực hiện “trao đổi” những thông tin khác với chatbot

Text

Description automatically generated

Hình . Một số giao tiếp cơ bản khác

## Đánh giá

Từ kết quả thực nghiệm rút ra một số đánh giá như sau:

* Xác định đúng được ý định (intent) có ý nghĩa quan trọng nhất đối với  
  chatbot. Đối với bài toán trong miền đóng cần xác định rõ ràng các intent, xây dựng tập dữ liệu đủ lớn, gán nhãn và tiến hành training.
* Xây dựng dữ liệu đào tạo, training cho chatbot với các kịch bản là rất cần  
  thiết để cho độ chính xác cao của chatbot.
* Chatbot ứng dụng AI có khả năng đáp ứng tốt với các kịch bản dựng sẵn, và  
  được đào tạo. Đối với các kịch bản nằm ngoài kịch bản dựng sẵn, có thể tăng cường khả năng cho chatbot bằng cách điều hướng người dùng về các câu mặc định hoặc các dạng giao diện menu lựa chọn.
* Việc xác định và phản hồi đa ý định có thể thực hiện bằng việc kết hợp các ý định.
* Qua bài toán thực nghiệm có thể thấy rằng áp dụng bài toán Chatbot cho việc  
  hỗ trợ trả lời thông tin khách sạn là khả thi, có tính thực tiễn cao, và hoàn toàn áp dụng được ngay trong thực tiễn.

## Ứng dụng trên robot thông minh

Graphical user interface

Description automatically generated with low confidence

Hình . Ứng dụng chatbot trên robot thông minh

Kết quả của đề tài có thể được sử dụng, tiếp tục phát triển để có thể triển khai trên robot, robot có thể giao tiếp, tương tác với con người qua ngôn ngữ văn bản và giọng nói.

## Kết luận chương 3

Trong chương 3 nhóm nghiên cứu đã mô tả các bước chính trong quá trình thiết kế và xây dựng một hệ thống chatbot ứng dụng cho robot, từ phần cứng đến chương trình phần mềm từ việc xây dựng dữ liệu, xây dựng ý định, thực thể đến các kịch bản trả lời. Kết quả của việc huấn luyện và test dữ liệu cho kết quả khá cao trên tập dữ liệu huấn luyện và test. Nếu có được nguồn dữ liệu lớn hơn để đào tạo thì độ chính xác sẽ cao hơn và hệ thống sẽ thông minh hơn.

###### KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* **Kết luận**

Sau khoảng thời gian làm đề tài dưới sự hướng dẫn tận tình của TS. Hà Thị Kim Duyên và cơ sở vật chất thiết bị được TS. Ngô Mạnh Tiến cung cấp, nhóm nghiên cứu đã hoàn thành được đề tài nghiên cứu của mình và có thể thực hiện ứng dụng, phát triển chatbot trên robot lễ tân IVASTBot. Đề tài được thực hiện là hệ thống ứng dụng trí tuệ nhân tạo xây dựng một hệ thống chatbot với khả năng “trò chuyện” với con người thông qua ngôn ngữ văn bản, giọng nói. Kiến trúc Task-oriented Dialogue Systems(TODs) được sử dụng để xây dựng Chatbot và phát triển trên nền tảng mã nguồn mở Rasa framework, bên cạnh đó hệ thống Chatbot còn thực hiện tích hợp thêm các dịch vụ Automatic Speech Recognition (ASR) và Text to Speed (TTS) để cung cấp tương tác bằng giọng nói cho IVASTChatbot. Ứng dụng chatbot trên Robot thông minh, mong muốn Robot có khả năng nhận biết thông tin người dùng cung cấp qua ngôn ngữ văn bản, giọng nói; thực hiện xử lý và “trò chuyện” với con người qua văn bản, giọng nói.

Trong quá trình thực hiện nghiên cứu nhóm nghiên cứu đã đạt được một số kết quả sau đây:

* Cài đặt môi trường, sử dụng linh hoạt các thư viện cần thiết và ngôn ngữ lập trình Python.
* Tiếp cận các kiến thức về trí tuệ nhân tạo, máy học và học sâu, xử lý ngôn ngữ tự nhiên. Tìm hiểu ứng dụng trí tuệ nhân tạo thực hiện thiết kế một hệ thống chatbot cơ bản. Các ứng dụng của hệ thống chatbot trong đời sống.
* Tìm hiểu về mạng nơron nhân tạo và ứng dụng vào xử lý ngôn ngữ tự nhiên, xây dựng chatbot.
* Thực hiện tìm hiểu, ứng dụng hệ thống trên robot dạng người thông minh.
* **Hướng phát triển**

**-** Xây dựng thêm đa dạng hóa các câu trả lời ngẫu nhiên theo các ý định.

- Tích hợp speech to text và text to speech cho chatbot để hỗ trợ voice.

- Xây dụng bot có khả năng trả lời các câu hỏi phức tạp hơn.

- Xây dựng giao diện quản trị người dùng.

###### TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Nguyen Thi Mai Trang, Maxim Shchebakov, “Enhancing Rasa NLU model for Vietnamese chatbot”, *International Journal of Open Information Technologies ISSN: 2307 – 8162 vol.9, no.1, 2021* |
| [2] | Wei-Lun Chao, “Machine Learning Tutorial”. *National Taiwan DISP Lab, National Taiwan University. 2011*: - *ngày truy cập 20/5/2022.* |
| [3] | Siwar Chibani, François-Xavier Coudert, “Machine learning approaches for the prediction of materials properties*”, HAL Id: hal-02911837,2020. pp 4-5: -ngày truy cập 20/5/2022.* |
| [4] | https://vi.wikipedia.org/wiki/Xử\_lý\_ngôn\_ngữ\_tự\_nhiên -*ngày truy cập 30/5/2022.* |
| [5] | Pham Nam, “Tổng quan về Chatbot,”Viblo, 5-Nov-2021.[Online].Available: https://viblo.asia/p/tong-quan-ve-chatbot-yMnKMByaZ7P.[Accessed:3-Jun-2022]. |
| [6] | P. H. Quang, “Rasa chatbot: Tăng khả năng chatbot với custom component và custom tokenization(tiếng Việt tiếng Nhật),” Viblo, 16-Mar-2020. [Online]. Available:<https://viblo.asia/p/rasa-chatbottang-kha-nang-chatbot-voi-custom-component-va> customtokenizationtiengviet-tieng-nhat-Qbq5QN4mKD8.[Accessed:3-Jun-2022] |
| [7] | T. Nguyen and M. Shcherbakov, “A Neural Network based Vietnamese Chatbot,” in 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), 2018 |
| [8] | Chu Le Long, “Nghiên cứu, xây dựng chatbot hỏi đáp thông tin khách sạn sử dụng rasa framework”, *ngày truy cập 20/5/2022.* |
| [9] | Nguyễn Thị Duyên, Ngô Mạnh Tiến, Hà Thị Kim Duyên, Bùi Quang Tuấn, Trần Bá Hiến, Nguyễn Minh Đông, Đỗ Quang Hiệp, "Xây dựng hệ điều hướng trên bản đồ, định vị SLAM cho Robot tự hành trong nhà kính nông nghiệp dựa trên hệ điều hành ROS," *Hội nghị - Triển lãm quốc tế lần thứ 4 về Điều khiển và Tự động hoá (VCCA),* 2021 |

###### PHỤ LỤC

**Phụ lục 1**

Code nguồn của toàn bộ đề tài:

https://github.com/ThanhNgaLe/NCKH\_2022\_Chatbot\_NhomCoKimDuyen.git

\*File Config.yml

version: "3.0"

language: vi

pipeline:

- name: vietnamese\_tokenizer.VietnameseTokenizer

- name: RegexFeaturizer

- name: LexicalSyntacticFeaturizer

- name: CountVectorsFeaturizer

- name: CountVectorsFeaturizer

analyzer: char\_wb

min\_ngram: 1

max\_ngram: 4

# - name: rasa\_nlu\_examples.featurizers.dense.FastTextFeaturizer

# cache\_dir: cmp/

# cache\_path: cmp/fasttext/cc.vi.300.bin

- name: DIETClassifier

epochs: 100

constrain\_similarities: true

model\_cofidence: linear\_norm

- name: EntitySynonymMapper

- name: FallbackClassifier

threshold: 0.2

policies:

- name: MemoizationPolicy

- name: TEDPolicy

epochs: 100

max\_history: 5

constrain\_similarities: true

- name: RulePolicy

\*Một số actions

actions:

- utter\_ask\_name

- utter\_ask\_time

- utter\_ask\_support

- utter\_ask\_continue

- utter\_xin\_chao

- utter\_tam\_biet

- utter\_hoi\_suc\_khoe

- utter\_hoi\_ten

- utter\_cam\_on

- utter\_khen

- utter\_ke\_chuyen

- utter\_ke\_chuyen\_cuoi

- utter\_hat

- utter\_giup

- utter\_is\_bot

- validate\_dat\_lich\_form

- utter\_submit

- utter\_dat\_lich\_slots

- action\_reset\_slot

**Phụ lục 2**

**ỨNG DỤNG TRÍ TUỆ NHÂN TẠO NHẬN DẠNG CẢM XÚC NGƯỜI GIAO TIẾP CHO ROBOT DẠNG NGƯỜI**

APPLICATION OF HUMAN EMOTION RECOGNITION ARTIFICIAL INTELLIGENCE FOR HUMAN-ROBOT

***SV. Tăng Xuân Biên1, SV. Nguyễn Đức Nam Bình1, SV. Lê Thanh Nga1,***

***TS. Hà Thị Kim Duyên1, TS. Trần Đình Thông1. TS. Ngô Mạnh Tiến2***

*1Khoa Điện tử, Trường Đại học Công nghiệp Hà Nội*

*2Viện Vật lý, Viện hàn lâm Khoa học và Công nghệ Việt Nam*

*Email:* [*ha.duyen@haui.edu.vn*](mailto:ha.duyen@haui.edu.vn)

*Số điện thoại: 0988901420*

**TÓM TẮT**

Bài báo trình bày về ứng dụng trí tuệ nhân tạo nhận dạng cảm xúc người giao tiếp với robot dạng người FER-HRI. Hệ thống FER-HRI cho phép robot không chỉ nhận ra cảm xúc của con người mà còn tạo ra được biểu cảm trên khuôn mặt để thích ứng với cảm xúc của con người. Một phương pháp nhận dạng cảm xúc khuôn mặt dựa trên mạng nơ ron tích chập CNN được trình bày, sử dụng để nhận dạng nét mặt theo thời gian thực cho robot. Biểu cảm khuôn mặt của robot được thể hiện bằng các ký hiệu đơn giản và hiển thị bằng màn hình LED được trang bị trên robot, con người có thể dễ dàng hiểu được. Các kết quả mô phỏng và quá trình chạy thử nghiệm đã cho thấy khả năng giao tiếp linh hoạt, chính xác được thực hiện bằng nhận dạng cảm xúc của con người và tạo biểu cảm trên khuôn mặt của robot trong một khoảng thời gian ngắn. Kết quả của bài báo là một ứng dụng có thể được áp dụng trong dịch vụ gia đình, điều trị y tế, lái xe an toàn, phát hiện nói dối,...

***Từ khóa:*** *Tương tác giữa người và robot (HRI – human-robot interaction); nhận diện; nhận dạng cảm xúc trên khuôn mặt (FER – facial emotion recognition); tạo biểu cảm; mạng nơ ron tích chập (CNN – convolution Neural Network).*

**ABSTRACT**

This paper presents the application of artificial intelligence to recognize human emotions to communicate with the humanoid robot FER-HRI. The FER-HRI system allows the robot to not only recognize human emotions but also generate facial expressions to adapt to human emotions. A facial emotion recognition method based on convolutional neural network (CNN) is used for robots. The robot's facial expressions are represented by simple symbols and displayed by an LED screen that can be easily understood by humans. Simulation results and test runs have shown that the flexible and accurate communication is made possible by human facial recognition and through the robot's facial expressions over a period of short time. The research results can be applied in home service, medical treatment, safe driving, lie detection, etc

***Keywords:*** *human-robot interaction (HRI); recognition; facial emotion recognition(FER); Emotion generation; convolution neural network (CNN).*

**CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| *FER-HRI* | *Facial emotion recognition human-robot interaction* |
| *CNN* | *Convolution Neural Network* |
| *AI* | *Artificial intelligence* |
| *ROS* | *Robot Operating System* |
| *HOG* | *Histogram of Oriented Gradient* |

**1. GIỚI THIỆU**

Trí tuệ nhân tạo, máy học và học sâu đang trở nên rất phổ biến ngày nay. Tất cả các công nghệ này được liên kết với nhau và mục tiêu chung là bắt chước trí thông minh của con người. Có rất nhiều ứng dụng cho các lĩnh vực này như suy luận logic (Logical Reasoning), biểu diễn tri thức (knowledge representation), lập kế hoạch (Planning), học tập (learning), xử lý ngôn ngữ tự nhiên (natural language processing), nhận thức (perception), trong đó đặc biệt là trí tuệ xã hội (social intelligence), đây cũng chính là một trong những lĩnh vực ứng dụng của AI[1]. Sử dụng AI, chúng ta có thể xây dựng trí thông minh xã hội trong một cỗ máy hoặc robot – robot xã hội (social robot). Nói một cách dễ hiểu, robot xã hội (social robot) là bạn đồng hành cá nhân hoặc robot hỗ trợ có thể tương tác với con người bằng giọng nói, thị giác và cử chỉ. Những con robot này hoạt động giống như một người bạn có thể biểu lộ cảm xúc như chúng ta và có thể truyền đạt cảm xúc của chúng bằng lời nói hoặc cử chỉ.

Để tương tác với robot dễ dàng và tự nhiên hơn, nghiên cứu [2], [3] đã đặt ra những yêu cầu mới về tương tác giữa người và robot (HRI), mong muốn robot có thể nhận dạng ra nét mặt người, hiểu cảm xúc và đưa ra phản ứng thích hợp [4], [5]. Robot trí tuệ cảm xúc đã thu hút sự chú ý lớn trong vài năm gần đây. Tuy nhiên, các nghiên cứu hiện tại vẫn ở giai đoạn sơ bộ, chỉ có một số hệ thống dịch vụ thông minh sử dụng nhận dạng cảm xúc. Một robot xã hội được gọi là Kismet từ MIT của tiến sĩ Cyntia Breazeal và nhóm được chế tạo vào năm 1990 có thể xác định được con người, đồ vật và mô phỏng các cảm xúc khác nhau[1]. Robot Jibo có một cái đầu xoay với màn hình có thể giao tiếp với mọi người bằng cách sử dụng nhận dạng giọng nói và có thể nhận dạng người bằng kỹ thuật nhận thức. Pepper là một robot xã hội hình người từ softbank, không giống các robot xã hội khác, robot này có hai cánh tay và chân để di động tương tự như robot hình người, nó có thể giao tiếp với con người và có các cảm biến xúc giác trên cơ thể[1]. Mặc dù có một số hệ thống tương tác giữa người và robot lên quan đến cảm xúc của robot, nhưng chỉ có một số nghiên cứu về cả nhận dạng cảm xúc và biểu hiện cảm xúc của robot để tạo điều kiện giao tiếp thông suốt giữa người và robot.

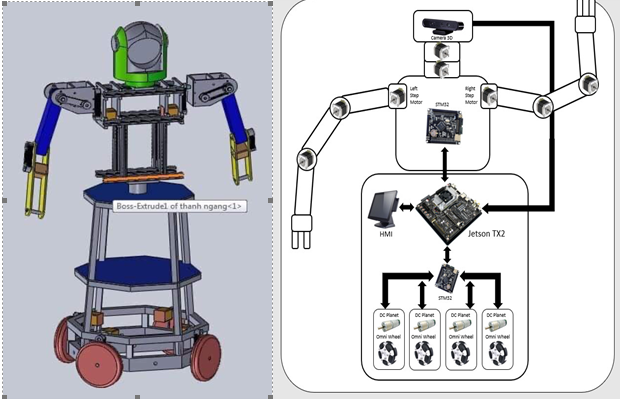
Một hệ thống nhận dạng cảm xúc dựa trên sự tương tác giữa người và robot (FER-HRI) được trình bày trong bài báo là một hệ thống con của “Robot dạng người thông minh IVASTBot ứng dụng trong giao tiếp, phục vụ con người ”[11]. Hệ thống FER-HRI được thiết kế chủ yếu cho hai mục tiêu: một là khả năng của robot nhận ra cảm xúc của con người dựa trên nét mặt, hai là khả năng của robot tạo ra biểu cảm khuôn mặt để giao tiếp tình cảm với con người thay vì giao tiếp vô cảm như truyền thống.

Quy trình vận hành của hệ thống gồm ba bước. Đầu tiên, robot thu thập dữ liệu hình ảnh khuôn mặt người thông qua Kinect và truyền đến bộ phân loại. Thứ hai, phương pháp nhận dạng nét mặt dựa trên mạng nơ ron tích chập (CNN) được sử dụng để nhận dạng cảm xúc người dùng và sau đó hệ thống sẽ tạo ra biểu cảm khuôn mặt của robot thích ứng với người dùng. Thứ ba, hệ thống truyền tín hiệu điều khiển cảm xúc đến robot và robot có thể phản hồi lại người dùng bằng một số biểu cảm cơ bản.

**2.** **KIẾN TRÚC TỔNG QUAN ROBOT DẠNG NGƯỜI THÔNG MINH IVASTBot**

Robot dạng người thông minh IVASTBot có thể di chuyển tự hành trên bốn bánh đa hướng dạng Omni [12], tránh vật cản động và tĩnh dựa trên thuật toán trí tuệ nhân tạo học tăng cường (Deep Reinforcement Learning). Được nhúng trên mạch điện tử xử lý hiệu năng cao Nvidia Jetson TX2 chuyên dụng, được lập trình trên hệ điều hành ROS (Robot Operating System). Robot có hai tay với 3 khớp, phần đầu robot có 2 khớp, có thể tương tác chuyển động cơ thể với người. Robot có thể nhận ra giao tiếp cảm xúc đa phương tiện và khuôn mặt người qua camera Kinect, và hiển thị giao diện cũng như tương tác cảm xúc với người dùng qua màn hình LCD[10].

Cấu trúc hệ thống điện tử điều khiển và chuyển động cơ khí được thể hiện trong hình 1.



**Hình 1. Cấu trúc hệ thống điện tử điều khiển và truyền động cơ khí của IVASTBot[11]**

Robot thông minh IVASTBot có khả năng nhận thức thế giới bằng camera, tương tác với con người bằng giọng nói, cử chỉ và đưa ra các quyết định bằng thuật toán trí tuệ nhân tạo. Robot này có thiết kế như sơ đồ khối sau:



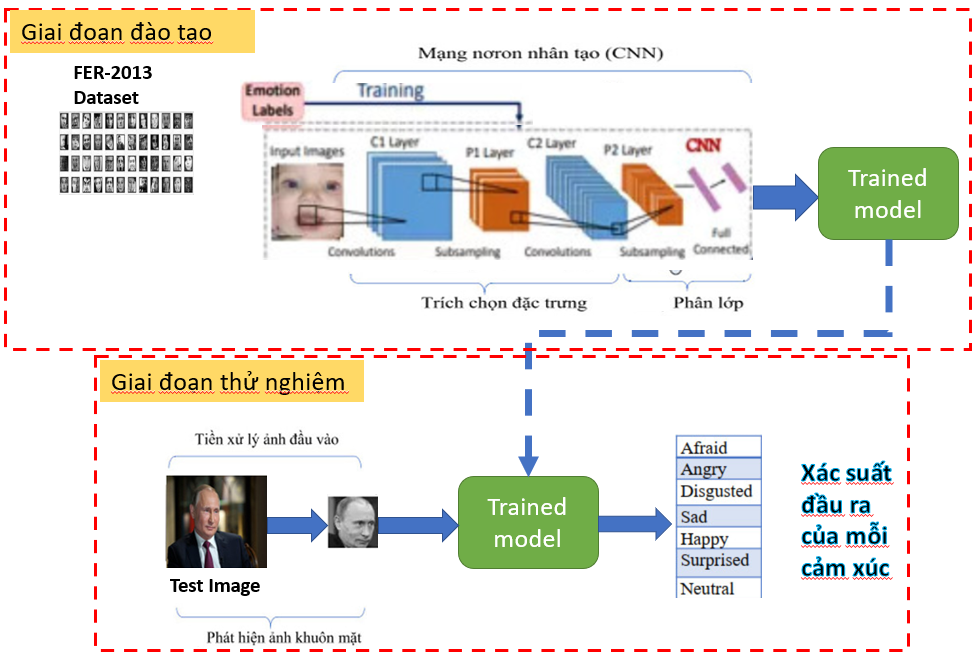
**Hình 3. Sơ đồ khối của IVASTBot [11]**

Khối phần cứng được sử dụng để nắm bắt các tín hiệu cảm xúc của con người và thể hiện cảm xúc của robot, trong đó máy ảnh có độ phân giải cao được sử dụng để chụp các bức ảnh thời gian thực về nét mặt và cử chỉ cơ thể của người dùng; micrô được sử dụng để thu thập tín hiệu lời nói; thiết bị truyền động giúp robot chuyển động đầu, cơ thể, cánh tay; động cơ và bánh đa hướng omni được tích hợp để điều hướng; màn hình LCD hiển thị giao diện và tương tác với người dùng.

Bên trong khối phần mềm, gồm các mô-đun nhận thức giúp xử lý dữ liệu camera và tìm kiếm các đối tượng cần thiết từ hiện trường; mô-dun nhận dạng / tổng hợp giọng nói giúp giao tiếp với người dùng; mô-đun trí tuệ nhân tạo; mô-đun điều khiển robot để điều khiển thiết bị truyền động; nút quyết định kết hợp tất cả dữ liệu từ các cảm biến và đưa ra quyết định cuối cùng về việc phải làm tiếp theo Hệ điều hành ROS kết nối tới các cảm biến, bộ truyền động. Khối GUI giúp giao tiếp giữa người dùng với robot thông qua thao tác với hình ảnh trên bảng điều khiển LCD.

**3.** **ỨNG DỤNG CNN NHẬN DẠNG BIỂU CẢM KHUÔN MẶT**

Để giao tiếp với người dùng, hệ thống FER-HRI cần thu thập và phân tích thông tin trên khuôn mặt. Hình ảnh khuôn mặt người dùng có thể được thu thập thông qua Kinect được trang bị trên robot thông qua cổng USB, sau đó ảnh được đưa vào “bộ phân loại” để xử lý, cuối cùng có thể thu được trạng thái cảm xúc. Các biểu cảm phổ biến trên khuôn mặt của con người chia thành 7 loại, bao gồm: tức giận (angry), khó chịu (disgusted), sợ hãi (fearful), vui (happy), buồn (sad), bất ngờ (surprised) và bình thường (neutral). Phương pháp tiếp cận nhận dạng nét mặt bằng cách sử dụng mạng nơ ron tích chập có thể thúc đẩy tỷ lệ chính xác của bộ phân loại, bao gồm 3 bước: bước 1 tiền xử lý để phát hiện và trích chọn đúng vùng ảnh chứa khuôn mặt và tăng cường chất lượng ảnh, bước 2 phân lớp sử dụng mạng CNN, bước 3 cho ra cảm xúc đầu ra dựa trên xác suất lớn nhất. Quá trình này được tóm tắt trong hình 3. Các bước chính là thu thập đặc điểm, phân lớp sử dụng CNN và nhận dạng cảm xúc.

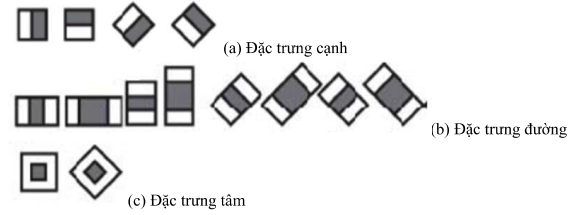


**Hình 3. Quy trình huấn luyện, nhận dạng cảm xúc bằng biểu cảm khuôn mặt**

Hình 3 tổng quan về quy trình huấn luyện, nhận dạng cảm xúc bằng biểu cảm khuôn mặt. Đầu tiên là quá giai đoạn huấn luyện, tập dữ liệu mẫu gồm các hình ảnh khuôn mặt người với các cảm xúc khác nhau đã được dán nhãn, sau đó đưa vào mạng nơ ron tích chập để trích chọn đặc trưng và phân lớp. Kết thúc của quá trình huấn luyện sẽ đưa ra mô hình đã huấn luyện. Giai đoạn thử nghiệm, hình ảnh đầu vào lấy được từ webcam sẽ được tiền xử lý và đưa vào mô hình đã huấn luyện, kết thúc giai đoạn thử nghiệm sẽ trả về xác suất đầu ra cuả mỗi cảm xúc.

**3.1. Tiền xử lý ảnh đầu vào**

Phần này áp dụng một số phương pháp tiền xử lý trên hình ảnh đầu vào, bao gồm phát hiện và cắt vùng ảnh chứa khuôn mặt, cải thiện chất lượng ảnh. Trong thực tế ứng dụng, ảnh đầu vào thường được trích xuất từ camera nên bao gồm cả không gian nền, do đó chúng ta phải thực hiện giai đoạn tìm kiếm và phát hiện khuôn mặt nhằm xác định vùng ảnh chứa khuôn mặt cần xử lý và cắt bỏ không gian nền của ảnh. Để thực hiện điều này, chúng tôi sử dụng phương pháp phát hiện vùng ảnh có chứa khuôn mặt dựa trên kỹ thuật Haar-cascade [13]. Độ nhiễu và độ rọi được giảm xuống bằng cách chuyển đổi hình ảnh đầu vào thành các hình ảnh đa cấp xám và áp dụng phép cân bằng mức xám nhằm giúp nâng cao chất lượng của hệ thống nhận diện khuôn mặt. Kỹ thuật Haar-cascade sử dụng cửa sổ trượt trên ảnh (từ trái sang phải, từ trên xuống dưới), trích rút các đặc trưng Haar-like (hình 4) trên cửa sổ đang xét dựa trên biểu đồ mức xám HOG, đưa vào mô hình phân lớp AdabBoost (Adaptive boosting) theo cơ chế phân tầng[14].

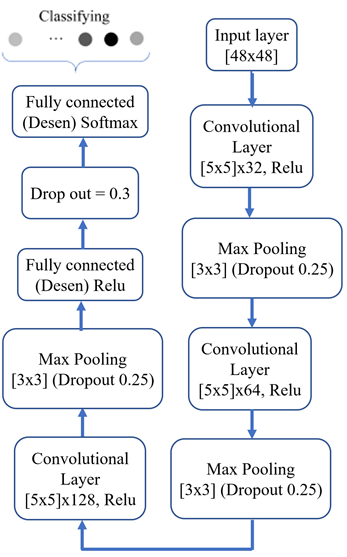


**Hình 4. Các đặc trưng Haar-like**

Ưu điểm của phương pháp này cho tốc độ nhanh trong tính toán nhờ việc rút trích đặc trưng dạng Haar-like so với các phương pháp rút trích đặc trưng khác. Ngoài ra, mô hình phân tầng AdabBoost (Adaptive boosting) đã loại bỏ hầu hết các vùng ứng viên không phải là khuôn mặt từ các tầng đầu tiên, chỉ xét rất ít vùng ứng viên ở các tầng cuối. So với công cụ phát hiện khuôn mặt trong thư viện OpenCV phổ biến, kỹ thuật phát hiện khuôn mặt Haar-cascade sử dụng thuật toán của Viola-Jones đã được cài đặt trong thư viện Dlib cho kết quả tốt hơn, ngay cả trong một số tình huống khó khăn và hạn chế của ảnh [15].

**3.2. Xây dựng mô hình**

Mô hình CNN được thiết kế gồm hai phần chức năng là trích chọn đặc trưng của biểu cảm khuôn mặt và phân lớp đối tượng dựa trên đặc trưng đã chọn. Mô hình CNN bao gồm nhiều lớp, số lớp nơ ron và độ lớn (số nơ ron) của mỗi lớp ảnh hưởng đến chất lượng cũng như độ phức tạp trong tính toán của mạng nơ ron. Các nghiên cứu thường điều chỉnh hai yếu tố này tuỳ theo bài toán ứng dụng để đạt được chất lượng mong muốn và đồng thời đảm bảo sự phức tạp tính toán chấp nhận được [16].



**Hình 5. Kiến trúc mạng CNN**

Mỗi lớp nơ ron trong mô hình CNN lấy một mảng nhiều chiều gồm các số làm đầu vào và tạo ra một mảng số nhiều chiều khác ở đầu ra sau đó trở thành đầu vào của lớp tiếp theo. Khi phân loại hình ảnh khuôn mặt, đầu vào của lớp nơ ron đầu tiên là kích thước hình ảnh đầu vào. Kích thước đầu ra của lớp cuối cùng là tập hợp các khả năng của các lớp khác nhau được phân loại cho mỗi ảnh đầu vào. Chúng tôi sử dụng cả ba loại lớp nơ ron để xây dựng kiến trúc của CNN bao gồm: 3 lớp tích chập (convolutional layer), 3 lớp gộp chung hay còn gọi là lớp nơ-ron tổng hợp (Pooling) và 2 lớp nơ-ron kết nối đầy đủ để phân loại (gọi là Desen). Mỗi lớp tích chập được kết nối theo sau nó bởi một lớp nơ-ron tổng hợp, áp dụng cơ chế kích hoạt ReLu (Rectified Linear Unit), mặc định là max(x,0) sau mỗi lớp tích chập để đảm bảo đầu vào không âm cho lớp nơ ron kế tiếp. Theo nguyên tắc xếp chồng các lớp nơ ron và giảm không gian mẫu (downsampling) tại các kết quả đầu ra của chúng, CNN thực hiện trích xuất các đặc trưng ngày càng trừu tượng và phức tạp hơn, đồng thời là bất biến đối với các phép biến dạng và chuyển đổi[6]. Hơn nữa, để khắc phục hiện tượng quá khớp (overfit) trong huấn luyện mạng nơ ron, mô hình này sử dụng thêm kỹ thuật Dropout sau mỗi lớp Pooling. Kỹ thuật Dropout này được giới thiệu và sử dụng chủ yếu gần đây, nó thực hiện chọn ngẫu nhiên hàm kích hoạt với một lượng theo tỷ lệ (được đặt trước) của các nơ ron và đặt thành 0 (tức là đầu ra của nơ ron được chọn bằng 0) trong quá trình huấn luyện mạng, do đó mô hình sẽ trở nên ít nhạy cảm hơn với các trọng số cụ thể trong mạng. Giá trị tỷ lệ cho mỗi lớp Dropout trong mô hình được thiết lập theo phương pháp heuristic và dựa trên quá trình thử nghiệm[16]. Mạng tích chập CNN trong mô hình này được chia các khối như hình 5.

Ảnh đầu vào là ảnh xám nhị phân chứa khuôn mặt người có kích thước 48x48, ảnh qua xử lý trở thành ma trận 2 chiều được đưa vào mô hình CNN. Lớp chập đầu tiên có 32 cửa sổ bộ lọc kích thước 5x5. Bộ lọc được triển khai sẽ lần lượt di chuyển dọc với bước trượt chaỵ dọc trên ảnh để quét được toàn bộ ảnh, sau đó đến lớp Relu được đính kèm với lớp chập để phi tuyến tính tránh sự phụ thuộc và việc truyền tham số giữa các nơ ron. Lớp Pooling kết hợp với hàm kích hoạt kiểu Max pooling, cửa sổ xử lý có kích thước 2x2 được sử dụng và đầu ra bị loại bỏ (đặt về 0) ngẫu nhiên với xác suất 0.25 giúp giữ lại thông tin hữu ích và cắt giảm dữ liệu cần xử lý ở lớp tiếp theo. Lớp chập và Max pooling thứ hai tương tự và thu được ảnh đầu ra lần lượt qua mỗi lớp là 18x18 pixel và 9x9 pixel (chia đôi chiều cao và chiều rộng của ảnh đầu vào). Lớp chập cuối cùng có 128 bộ lọc kích thước 5x5 và được ma trận đầu ra sau xử lý là 6x6 pixel. Lớp kết nối đầy đủ (fully connected) kết hợp các tính năng lại với nhau để tạo ra một mô hình huấn luyện (trained model) và sử dụng hàm softmax để phân loại đầu ra.

**4. KẾT QUẢ MÔ PHỎNG VÀ THỰC NGHIỆM**

**4.1. Dữ liệu thử nghiệm**

Để kiểm tra mô hình được đề xuất ở trên, chúng tôi sử dụng bộ dữ liệu mẫu FERC-2013 được cung cấp bởi cuộc thi “Thách thức nhận dạng cảm xúc khuôn mặt người” trên Kaggle. Dữ liệu mẫu bao gồm 35887 hình ảnh màu xám có độ phân giải 48x48. Kaggle đã chia thành 28,709 hình ảnh dùng để huấn luyện, 3589 hình ảnh kiểm chứng và 3589 hình ảnh kiểm tra. Mỗi hình ảnh chứa một khuôn mặt người trong tự nhiên và được gắn nhãn từ 0 đến 6 tương ứng với 7 cảm xúc: Giận (angry), khó chịu (disgusted), sợ hãi (fearful), vui (happy), buồn (sad), bất ngờ (surprised) và bình thường (neutral)[5]. Một số hình ảnh của bộ dữ liệu FERC-2013 được thể hiện trong hình 6. Sự phân bố ảnh trên các cảm xúc trong tập dữ liệu được thống kê trong biểu đồ 1.

A collage of a person's face

Description automatically generated with medium confidence

**Hình 6. Bộ ảnh chứa khuôn mặt trực diện trong tập FERC-2013**

Biểu đồ biểu thị số lượng phân bố ảnh trong tệp

Biểu đồ 1: Số lượng ảnh phân bố trên các cảm xúc trong tập FERC-2013

Chart, bar chart, waterfall chart

Description automatically generated

Với 35887 ảnh sử dụng để huấn luyện cùng với kiểm thử trong tệp nhằm giúp người dùng có thể nghiên cứu và tạo ra một số mạng cơ bản để nhận diện cảm xúc.

**4.2 Kết quả thử nghiệm**

\* *Khởi tạo các thông số để huấn luyện mạng:*

- Tốc độ học: 0.001

- Dropout: 0.3

- Nơ-ron: 3072

- Batch size: 50

- Epock: 100

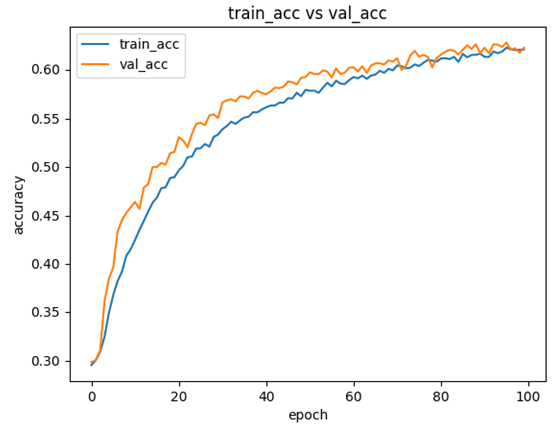
\* *Huấn luyện mạng:*

Môi trường phát triển hệ điều hành Ubuntu 18.04, Ram:8GB, HDD, i5 2.20GHz CPU, công cụ phát triển VSCode, ngôn ngữ phát triển python, cơ sở dữ liệu dataset FERC-2013.

Huấn luyện mạng: lấy 80% mẫu dữ liệu để huấn luyện (Giúp tránh hiện tượng quá khớp). Đầu tiên, phân loại và sắp xếp hình ảnh trong tập mẫu FERC-2013 cho mô hình. Sau đó chuyển đổi ảnh huấn luyện, ảnh xác nhận, ảnh kiểm tra thành 3 tập dữ liệu tương ứng train.npy, validation.npy và test.npy cho mô hình. Ba tập chứa dữ liệu này là tập hợp hình ảnh của 7 loại cảm xúc đã được chuyển đổi thành những ma trận nhị phân với kích thước 48x48 và có gắn nhãn. Tập tin train.npy để huấn luyện và validation.npy để hiệu chỉnh các trọng số trong quá trình huấn luyện. Khi kêt thúc quá trình huấn luyện thì các trọng số và dữ liệu của mạng sẽ được lưu vào 3 tập tin có đuôi là “.data-00000-of-00001, “.index” và “.meta” để được sử dụng khi đánh giá và tải vào chương trình ứng dụng.

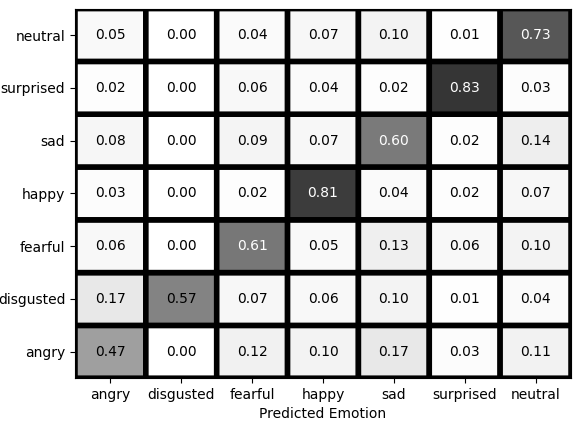
\**Kiểm tra mạng:*

Lấy 20% dữ liệu còn lại để kiểm tra mạng.



**Hình 7. Độ chính xác khi huấn luyện và kiểm thử**

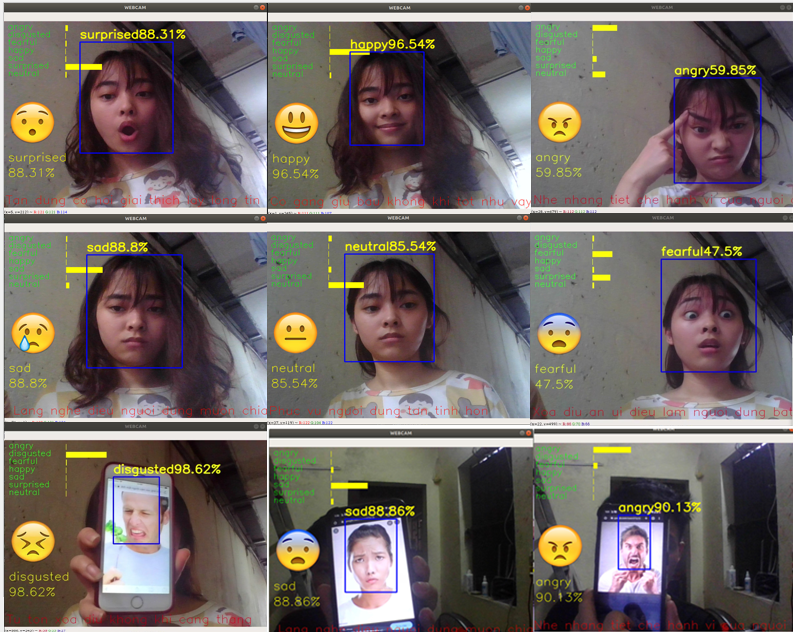
Theo biểu đồ hình 7 khi bắt đầu huấn luyện thì mức độ chính xác ở mức rất thấp và tăng dần. Mô hình ban đầu trong quá trình khi đang huấn luyện thì khoảng các mức chính xác khi huấn luyện và mức chính xác khi kiểm thử cách xa nhau, sau một quá trình huấn luyện khoảng 60 lần thì đồ thị đã gần nhau hơn, tránh được hiện tượng quá khớp. Cuối cùng, với độ chính xác xấp xỉ 67% chỉ hợp với yêu cầu xây dựng ứng dụng đơn giản. Ta thấy độ chính xác của huấn luyện và kiểm thử đã không tạo ra độ lệch quá lớn. Với thời gian huấn luyện là dưới 7 tiếng cũng đã tránh được hiện tượng chưa khớp (underfitting).



**Hình 8. Ma trận dự đoán của mô hình**

Hình 8 cho ta thấy ma trận dự đoán của mô hình được in ra sau khi nạp mô hình vào. Ma trận gồm đường chéo từ góc trên bên phải kéo dài đến góc dưới bên trái thể hiện được tỉ lệ phần trăm chính xác các cảm xúc. Cảm xúc bất ngờ, vui vẻ, và bình thường có tỉ lệ phần trăm chính xác cao nhất (trên 70%) do biểu cảm này nổi bật và có lượng dữ liệu để đào tạo lớn. Các cảm xúc như khó chịu, buồn, ghê sợ, tức giận có phần trăm chính xác chưa tốt (dưới 60%) do là dạng cảm xúc khó phân biệt. Theo ma trận dự đoán thì số lượng ảnh dữ liệu đầu vào để huấn luyện càng ít thì cho ra kết quả với chính xác thấp như số lượng ảnh “khó chịu”.

* Kết quả quả trình nhận dạng:

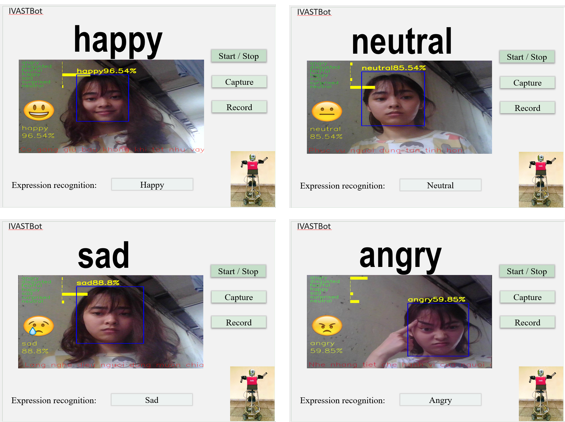


**Hình 9. Kết quả nhận dạng biểu cảm khuôn mặt**

Kết quả nhận dạng biểu cảm khuôn mặt cơ bản đã đáp ứng được thời gian thực, có khả năng nhận dạng được cảm xúc trên khuôn mặt của người thật hay ảnh mặt người đủ sáng, đủ rõ nét.

***4.4 Ứng dụng trên robot thông minh IVASTBot***

Để làm cho robot nhận biết được cảm xúc của con người, phương pháp trích xuất đặc trưng và thuật toán phân loại trong phần 3 nên được áp dụng. C++ nên được sử dụng để lập trình Kinect để chụp ảnh khuôn mặt theo thời gian thực. Các phương pháp trích xuất đặc trưng và phân lớp nên được thực hiện trong ROS. Một màn hình LCD để hiển thị giao diện hoạt động của hệ thống, giao diện hoạt động này hiển thị một số thông tin tương tác giữa con người và robot. Ví dụ hiển thị hình ảnh thời gian thực được chụp bới Kinect được trang bị trong robot và hiển thị kết quả nhận dạng và biểu hiện trên khuôn mặt.



**Hình 10. Nhận dạng khuôn mặt trên IVASTBot**

**5. KẾT LUẬN**

Một hệ thống ứng dụng trí tuệ nhân tạo nhận dạng cảm xúc người giao tiếp cho robot dạng người (FER-HRI) được đề xuất. Phương pháp nhận dạng cảm xúc khuôn mặt dựa trên mô hình mạng nơ-ron tích chập (CNN) kết hợp với đặc trưng truyền thống được áp dụng cho hệ thống FER-HRI. Robot thông minh IVASTBot có khả năng nhận biết cảm xúc trên khuôn mặt và thực hiện được tương tác đơn giản để thích ứng với người dùng. Kết quả thử nghiệm cho thấy người dùng có thể thực hiện được một số tương tác cảm xúc đơn giản với robot như: vui, buồn, tức giận… Robot có thể hiểu được cảm xúc của người và thể hiện cảm xúc bằng các biểu cảm đơn giản theo trạng thái cảm xúc của người dùng.

Hệ thống FER-HRI dựa trên nhận dạng cảm xúc khuôn mặt còn có nhiều ứng dụng tiềm năng, nhất là trong dịch vụ khách hàng, lái xe an toàn, dịch vụ tại nhà, chăm sóc sức khỏe... Hệ thống cung cấp các dịch vụ tốt hơn cho con người: chẳng hạn như khi khách hàng duyệt Web, hệ thống FER-HRI sẽ đánh giá mức độ hài lòng của khách hàng thông qua nét mặt, chuyển động của mắt; hệ thống có thể được ứng dụng trong lái xe an toàn bằng cách theo dõi trạng thái cảm xúc của người lái xe từ chuyển động của mắt, nét mặt và các thông số sinh lý như nhịp tim, hơi thở. Ngoài ra, hệ thống có thể được sử dụng để chăm sóc sức khỏe người cao tuổi và giao tiếp tình cảm họ. Trong nghiên cứu tiếp theo, chúng tôi sẽ phát triển hệ thống tương tác giữa người-robot đa phương thức dựa trên nét mặt, giọng nói, tư thế cơ thể, tín hiệu sinh lý và ứng dụng FER-HRI trong các lĩnh của cuộc sống.

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | Ramkumar Gandhinathan and Lentin Joseph, "ROS Robotics Project" *Packt Publishing Ltd,* 1181219, 2019. |
| [2] | K. Qian, J. Niu, and H. Yang, “Developing a gesture based remote human-robot interaction system using kinect,” *Int. J. Smart Home,* vol. 7, no. 4, pp. 203-208, Jul. 2013. |
| [3] | Z. K. Wang, K. Mulling, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, “Probabilistic movement modeling for intention ¨ inference in human-robot interaction,” *Int. J. Robot. Res.,* vol. 32, no. 7, pp. 841-858, Apr. 2013. |
| [4] | M. Awais and D. Henrich, “Human-robot interaction in an unknown human intention scenario,” in *Proc. 11th Int. Conf. Frontiers of Information Technology*, Washington, DC, USA, 2013, pp. 89-94. |
| [5] | L. Zhang, M. Jiang, D. Farid, and M. A. Hossain, “Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot,” *Exp. Syst. Appl.,* vol. 40, no. 13, pp. 5160-5168, Oct. 2013. |
| [6] | [Kam17] Patrik Kamencay, Miroslav Benco, Tomas Mizdos and Roman Radil, “A New Method for Face Recognition Using Convolutional Neural Network”, Digital Image Processing and Computer Graphics, Vol. 15, No. 4, pp.663-672, 2017. |
| [7] | A. Shima and F. Azar, “Convolutional Neural Networks for Facial Expression Recognition,” arXiv:1704.06756v1 [cs.CV], 2017. |
| [9] | K. Simonyan and A. Zisserman, “Very deep convolutional networks for large- scale image recognition,” *arXiv preprint arXiv*:pp. 1409-1556v6 [cs.CV], 2015. |
| [10] | Ngô Mạnh Tiến, Nguyễn Mạnh Cường, Hà Thị Kim Duyên, Bùi Quang Tuấn, Trần Bá Hiến, Nguyễn Minh Đông, Đỗ Quang Hiệp, "Xây dựng Hệ thống Bản đồ hóa SLAM và ứng dụng điều hướng cho Robot đa hướng sử dụng bánh xe Mecanum dựa trên hệ điều hành ROS," *Hội thảo quốc gia lần thứ XXIV: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Thái Nguyên,* 13-14/12/2021. |
| [11] | Nguyễn Thị Duyên, Ngô Mạnh Tiến, Hà Thị Kim Duyên, Bùi Quang Tuấn, Trần Bá Hiến, Nguyễn Minh Đông, Đỗ Quang Hiệp, "Xây dựng hệ điều hướng trên bản đồ, định vị SLAM cho Robot tự hành trong nhà kính nông nghiệp dựa trên hệ điều hành ROS," *Hội nghị - Triển lãm quốc tế lần thứ 4 về Điều khiển và Tự động hoá (VCCA),* 2021. |
| [12] | L. Xie, C. Scheifele, W. Xu, and K. A. Stol, "Heavy-duty omni-directional Mecanum-wheeled Robot for autonomous navigation: System development and simulation realization," *IEEE International Conference on Mechatronics (ICM),* pp. 256-261, 2015. |
| [13] | M. A. Abuzneid, A. Mahmood, “Enhance Human Face Recognition Using LBPH Descriptor, Multi-KNN, abd BPNN”, IEEE Access, Vol.6, pp.20642-20651, 2018. |
| [14] | Li Cuimei, Qi Zhiliang, Jia Nan and Wu Jianhua, “Human face detection algorithm via Haar cascade classifier combined with three additionl classifiers”, IEEE 13th International Conference on Electronic Measurement & Instruments, pp.43-487, 2017. |
| [15] | Ekberjan Derman and Albert Ali Salah, “Continuous Real-Time Vehicle Driver Authentication Using Convolutional Neural Network Based face Recognition”, 13th IEEE International Conference on Autimatic Face & Gesture Recogniton, 2018. |
| [16] | Dương Thăng Long, Bùi Thế Hùng, “Một phương pháp nhận dạng khuôn mặt dựa trên mạng nơ ron tích chập”, *Tạp chí Khoa học – Viện Đại học Mở Hà Nội ( 08/2019)1-20* |