

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------🙞🙜🕮🙞🙜------

****

**BÁO CÁO ĐỒ ÁN STP**

**Đề tài: “ Xây dựng OfficeBOT”**

**Sinh viên thực hiện :**

1. **Nguyễn Lan Anh – B16DCCN009**
2. **Hoàng Đức Hải – B16DCCN122**
3. **Hoàng Mậu Trung – B16DCCN370**

**Hà Nội, tháng 9 năm 2020**

Contents

[**LỜI NÓI ĐẦU** 2](#_Toc51488914)

[**CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN:** 5](#_Toc51488915)

[**1.** **Đặt vấn đề:** 5](#_Toc51488916)

[**2.** **Các chức năng của chatbot trong bài toán Office Bot:** 5](#_Toc51488917)

[**CHƯƠNG 2: TỔNG QUÁT VỀ CHATBOT** 6](#_Toc51488918)

[**1.** **Chatbot là gì và các loại chatbot?** 6](#_Toc51488919)

[**2.** **Cách thưc hoạt động của chatbot:** 8](#_Toc51488920)

[**3.** **Cách thức xây dựng chatbot** 9](#_Toc51488921)

[**a)** **Rasa:** 9](#_Toc51488922)

[**b)** **Google Dialogflow** 15](#_Toc51488923)

[**c)** **Sự khác biệt giữa khung chatbot Rasa và khung chatbot Dialogflow** 18](#_Toc51488924)

[**d)** **Lựa chọn cho đề tài** 19](#_Toc51488925)

[**CHƯƠNG 3: XÂY DỰNG DỮ LIỆU VÀ TIỀN XỬ LÝ DỮ LIỆU** 19](#_Toc51488926)

[**1.** **Xây dựng dữ liệu** 19](#_Toc51488927)

[**2.** **Tiền xử lý dữ liệu** 23](#_Toc51488928)

[**CHƯƠNG 4 XÂY DỰNG KHỐI XỬ LÝ LÕI** 24](#_Toc51488929)

[**2.** **Sử dụng machine learning truyền thống** 27](#_Toc51488930)

[**2.1** **Trích đặc trưng: CountVector TF-IDF <chưa viết>** 28](#_Toc51488931)

[**2.2** **Phân loại intent dựa trên các thuật toán machine truyền thống:** 28](#_Toc51488932)

[**a)** **Mutil Layer Perceptron (MLP):** 28](#_Toc51488933)

[**b)** **Decision Tree:** 29](#_Toc51488934)

[**c)** **Random Forest:** 32](#_Toc51488935)

[**d)** **Bảng so sánh kết quả: (Chưa có)** 34](#_Toc51488936)

[**3.** **Sử dụng deeplearning** 34](#_Toc51488937)

[**3.1** **Nhận diện tên thực thể: CRF** 34](#_Toc51488938)

[**3.2** **Nhận diện intent:** 34](#_Toc51488939)

[**3.3** **Đánh giá đưa ra nhận xét về hiệu năng và tốc độ.** 34](#_Toc51488940)

[**CHƯƠNG 5: TÍCH HỢP KHỐI XỬ LÝ LÕI VÀO ỨNG DỤNG CHAT** 34](#_Toc51488941)

# **LỜI NÓI ĐẦU**

Ngày nay với sự phát triển vượt bậc của công nghệ các nhà khoa học, nhà nghiên cứu không ngừng chinh phục giấc mơ tự động hóa, và thông minh hóa các sản phẩm công nghệ thông tin. Những nỗ lực đó khiến cho các chương trình càng ngày thông minh và giống con người. Trong đó xu hướng đang thu hút sự chú ý của phần lớn ngành công nghệ là chatbot. Sự pha trộn giữa phản ứng tức thì và kết nối nhất quán khiến chúng thành một thay đổi hấp dẫn cho các ứng dụng web.

Vậy chatbot là gì?

Chatbot xét một cách tổng thể chatbot là một chương trình máy tính được người lập trình thiết kế giúp trả lời tự động các câu hỏi của người dùng. Chúng giúp giao tiếp, tương tác/nói chuyện với con người một cách tự động 24/7. Trên thực thế chatbot nhanh hơn con người trong việc đưa ra câu trả lời. Điển hình một số chatbot nổi tiếng như: Google Asistant, Siri của Apple hay Bixby của SamSung.

Chatbot đang dần được ứng dụng sâu vào nhiều lĩnh vực như cuộc sống thường ngày, kinh doanh. Đối tượng chatbot cũng rất đa dạng của nhiều ngành nghề khác nhau. Trong đó nổi bật hơn hết là chatbot trong ngành kinh doanh được triển khai trong cách doanh nghiệp cho mục đích khác nhau như: quảng cáo, tuyển dụng, chăm sóc khách hàng…

Trong đề tài này nhóm tác giả sẽ khám phá vấn đề cho một chatbot trong doanh nghiệp trong lĩnh vực nhân sự. Một vấn đề mà bất cứ doanh nghiệp nào đều quan tâm và chatbot đang trở thành một HR đắc lực mà bất cứ doanh nghiệp nào cũng muốn sở hữu.

Nội dung của báo cáo này bao gồm 6 chương:

* Chương 1: Giới thiệu bài toán Office Bot: Nghiên cứu phân tích nhu cầu và các vấn đề trong trong doanh nghiệp khi xây dựng chatbot.
* Chương 2: Tổng quan về một chatbot: Các thành phần chatbot và cách thức tạo nên một chatbot
* Chương 3: Xây dựng bộ dữ liệu và tiền xử lý dữ liệu: Xây dựng bộ dữ liệu cho bài toán Office Bot cho vấn đề nhân sự
* Chương 4: Xây dựng khối xử lý lõi: Xây dựng khối xử lý lõi của chatbot dựa trên cách 2 cách tiếp cận:

4.1 Machine learning truyền thống.

4.2 Deep learning hiện đại.

* Chương 5: Tích hợp khối xử lý lõi vào các ứng dụng chat: Tích hợp các khối xử lý lõi vào các ứng dụng chat như Telegram, Facebook Mesenger
* Chương 6: Đánh giá, kiểm thử: Test và kiểm tra hoạt động của bot trong môi trường production.

# **CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN**

Trong chương này sẽ giới thiệu về bài toán Chatbot áp dụng cho doanh nghiệp cụ thể trong lĩnh vực chính sách nhân sự của một công ty. Các nội dung phân tích tổng quát các chức năng và tính hữu dụng trong việc ứng dụng tự động hóa cho bộ phận nhân sự.

1. **Đặt vấn đề:**

Chatbot là một chương trình tự động được xây dựng dựa trên các thuật toán của trí tuệ nhân tạo và ứng dụng cho nhiều ngành nghề và lĩnh vực từ kinh doanh, ẩm thực cho đến giáo dục và dịch vụ hỗ trợ đặt phòng, đặt vé, vận tải…

Ngày này sự đa dạng hóa ngành nghề, lĩnh vực dẫn tới nhu cầu việc làm ngày càng lớn trong khi đó bộ phận HR không thể đáp ứng hết nhu cầu của các ứng cử viên. Điều đó khiến cho các doanh nghiệp luôn mong muốn có một giải pháp giúp thay thế con người giải đáp thắc mắc về nhân tại bộ phận HR. Để đáp ứng nhu cầu đó thì một trong phương pháp hữu hiệu và giá cả phải chăng đó là sử dụng chatbot. Chatbot có thể đáp ứng nhu cầu liên tục nhanh chóng 24/7.

Chính điều đó một officebot nhân sự là trợ thủ đắc lực hỗ trợ thông tin đến những ứng cử viên giúp họ có thể năm bắt các thông tin tuyển dụng nhân sự trong công ty một cách nhanh chóng và hiệu quả.

1. **Các chức năng của chatbot trong bài toán Office Bot:**

Chatbot được xem là một HR tự động với chi phí nhân công giá rẻ, nhưng vẫn đáp ứng tính hiệu quả và liên tục. Các chức năng mà chatbot sử dụng thay thế cho HR bao gồm:

* Trả lời câu hỏi ứng cử viên về chính sách công ty
* Quảng cáo về công ty
* Giải đáp phản hồi thắc mắc của nhân sự
* Thông báo về điều chỉnh chính sách nhân sự trong công ty đến các nhân viên.

# **CHƯƠNG 2: TỔNG QUÁT VỀ CHATBOT**

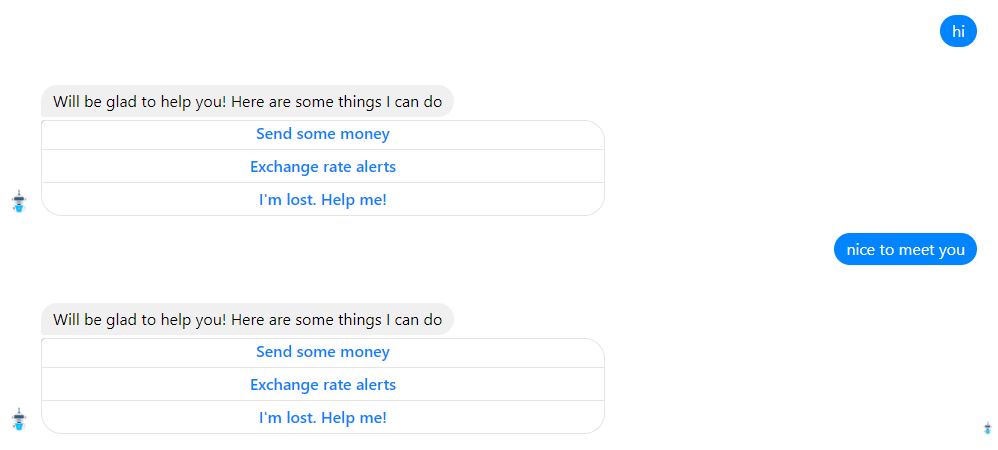
Trong chương này nhóm tác giả tập trung giới thiệu cách thức hoạt động của một chatbot. Nghiên cứu các cách thức xây dựng chatbot đang có sẵn hiện tại.

1. **Chatbot là gì và các loại chatbot?**

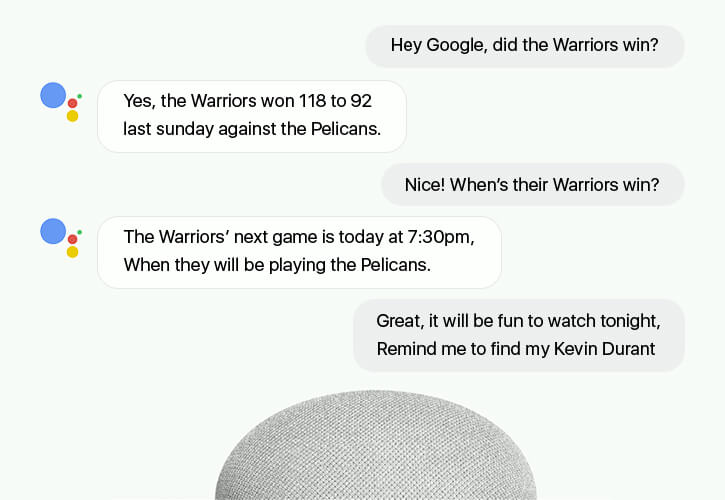
Chatbot là một cụm từ không còn xa lạ đối với nền công nghệ phát triển hiện nay trên thế giới. Về cơ bản, chatbot là một chương trình máy tính tiến hành cuộc trò chuyện thông qua nhắn tin nhắn, nó có thể tự động trả lời những câu hỏi hoặc tình huống. Phạm vi và sự phức tạp của chatbot được xác định bởi thuật toán của người tạo nên chúng. Chatbot thường được sử dụng truy cập thông qua các ứng dụng, trang web và các nền tảng nhắn tin của nhiều tổ chức như: Facebook, Google, Amazon, Line, Telegram.

Chatbot có 3 loại chính:

* Chatbot viết kịch bản (Scripted chatbot): Đây là chatbot có hành vi được xác định bởi các quy tắc. Tại mỗi bước của cuộc trò chuyện, người dùng cần chọn tùy chọn rõ ràng để xác định bước tiếp theo trong cuộc trò chuyện. Các cách tùy chọn được trình bày cho người dùng ở mỗi bước trong cuộc hội thoại, tức là liệu họ cần phản hồi bằng một văn bản, giọng nói hoặc cảm ứng thì sẽ phụ thuộc vào tính năng của nền tảng trò chuyện mà người dùng sử dụng và thiết kế của chatbot.



* Chatbot nhận dạng từ khóa: không như dạng chatbot theo kịch bản các chatbot dựa trên nhận dạng từ khóa có thể lắng nghe những gì mà người dùng gõ và trả lời một cách thích hợp. Những chatbot sử dụng các từ khóa tùy biến và AI để xác định làm thế nào để đưa ra câu trả lời thích hợp cho người dùng.
* Chatbot thông minh (Intelligent Chatbot): là loại chatbot được xây dựng với kĩ thuật, thuật toán trí thông minh nhân tạo AI. Trí thông minh nhân tạo (AI) cho phép chúng cải thiện tính linh hoạt về đầu vào người dùng có thể thu nhận. Chúng có thể thu nhận đầu vào dạng tự do dưới hình thức trình bày bằng văn bản hay giọng nói và cũng không có giới hạn hay ràng buộc đầu vào.



1. **Cách thưc hoạt động của chatbot:**

Chatbot trước hết nó là một ứng dụng bình thường, cũng có giao thức tầng ứng dụng (application layer), cơ sở dữ liệu (database) và giao diện ứng dụng lập trình API (Aplication Programing Interface). Giao diện người dùng UI (User Interface) của Chatbot là giao diện trò chuyện. Mặc dù Chatbot cung cấp các tính năng dễ dàng cho người dùng có thể sử dụng nhưng lại tăng độ phức tạp và tinh vi trong quá trình xây dựng, quản lý và huấn luyện.

Chatbot giao tiếp với người dùng trên một số quy trình sau:

* Translator: Thông tin/yêu cầu của người dùng (user) sẽ được dịch lại bằng ngôn ngữ lập trình. Máy có thể hiểu được các công việc cần thực hiện
* Processor: Công nghệ AI tiến hành xử lý yêu cầu người dùng
* Respondent: Máy tính nhận output từ từ AI và gửi trả kết quả người dùng kết quả tương ứng trên platform mesenger.

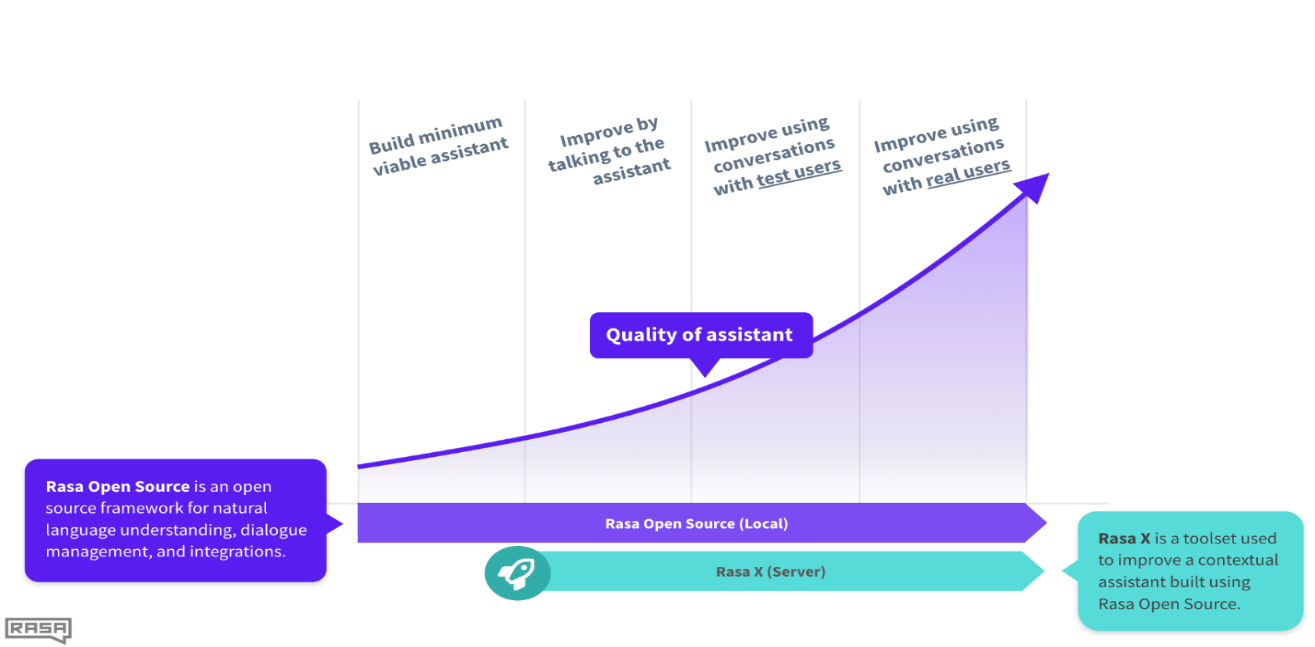


1. **Cách thức xây dựng chatbot**

Trong phần trước nhóm tác giả trình bày nguyên lý hoạt động của một chatbot, trong phần này sẽ giới thiệu về cách xây dựng một chatbot.

Ngày nay có nhiều platform, nhiều framework hỗ trợ cho việc xây dựng chatbot với tính ưu việt và hiệu quả cao như: Rasa, Google Dialogflow… Trong phần này nhóm tác giả sẽ giới thiệu cả hai nền tảng giúp xây dựng chatbot và lựa chọn một nền tảng cho bài toán đã trình bày bên trên.

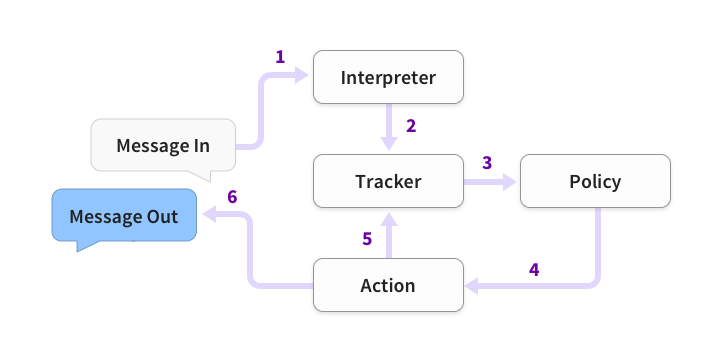
1. **Rasa:**



Rasa Open Source là một framework AI cho cuộc hội thoại nhằm giúp xây dựng trợ lý (chatbot) có thể hiểu được ngữ cảnh của cuộc hội thoại. Rasa Open Source bao gồm các phần:

* NLU: nhiệm vụ phần này giúp xác định người dùng muốn làm gì và nắm bắt ngữ cảnh chính.
* Core: chọn phản hồi hoặc dùng hành động tốt nhất tiếp theo dựa trên lịch sử của câu trò chuyện.
* Chanel và intergrations: giúp kết nối chatbot tới người sử dụng và các hệ thống back-end như Telegram hay Facebook Mesenger.

Qúa trình xử lý tin nhắn trong Rasa:



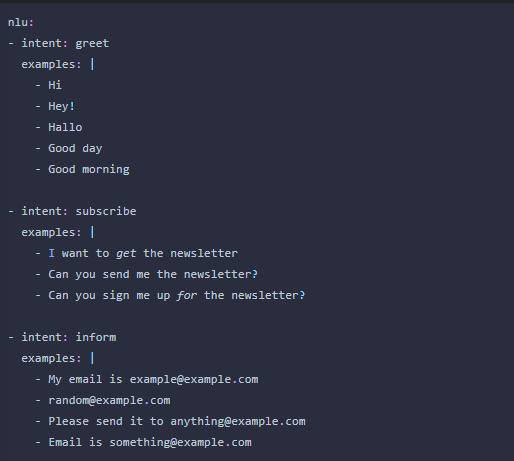
1. Thông báo được tiếp nhận và chuyển tới một Interpeter chuyển nó thành một từ điển bao gồm văn bản gốc, mục đích giúp nó tìm được các thực thể. Phần này do khối NLU đảm nhận.
2. Tracker là một đối tượng giúp theo dõi trạng thái cuộc hội thoại nó có nhiệm vụ phát hiện khi nào có tin nhắn mới đến.
3. Policy tiếp nhận trạng thái từ Tracker gửi đến.
4. Policy quyết định lựa chọn hành động nào sẽ được phản hồi cho người dùng
5. Action đó được Tracker ghi nhận
6. Action đó sẽ được phản hồi thành message out cho người dùng.

Xây dựng trợ lý với Rasa bao gồm loạt thao tác:

1. Dữ liệu NLU:

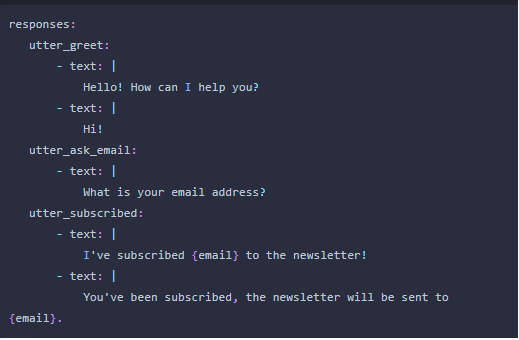
Dữ liệu NLP là phần quan trọng trong việc xây dựng chatbot nó là tất cả những điều khác nhau mà người có thể nói với chatbot.

Để chatbot có thể hiểu được người dùng muốn gì (thuật ngữ này được gọi intent) bất kể người dùng diễn đạt thông điệp của họ thế nào, chúng ta cần cung cấp các mẫu hay các câu mà người dùng có thể hỏi giúp cho chúng có thể trả lời. Chẳng hạn với ý định “lời chào” chứa các câu mẫu như: “xin chào”, “chào”, “chào buổi sáng”.



1. Phản hồi (Respond)

Bây giờ chatbot đã có thể hiểu được ý định của người dùng nó cần phải tiếp tục phản hồi cho người dùng. Chẳng hạn với ý định “chào” thì chatbot sẽ phản hồi cho người dùng “chào bạn, tôi có thể giúp gì cho bạn?”.

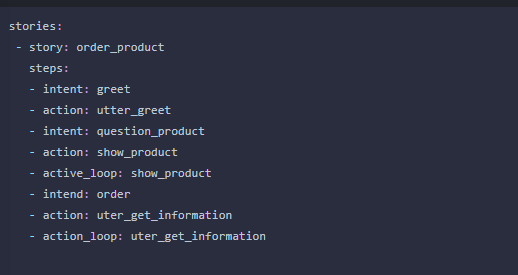


1. Câu chuyện (Stories)

Câu chuyện trong các hội thoại đó là ngữ cảnh của cuộc hội thoại, đó là sự tổng hợp các ngữ cảnh của các câu trong cuộc trò chuyện trong thời gian phía trước đó và giúp nó đưa ra mục đích cuối cùng của người dùng để phản hồi.

Câu chuyện sẽ hiển thị luồng hội thoại trong đó trợ lý giúp người dùng hoàn thành mục tiêu một cách dễ dàng. Sau đó, bạn có thể thêm tình huống mà người dùng không muốn cung cấp thông tin, hoặc là chuyển chủ đề.

Chẳng hạn câu chuyện người dùng và chatbot trao đổi lời chào người dùng hỏi về mặt hàng X, sau đó người dùng đặt hàng, và kết thúc là thông tin giao hàng như số điện thoại, địa chỉ…



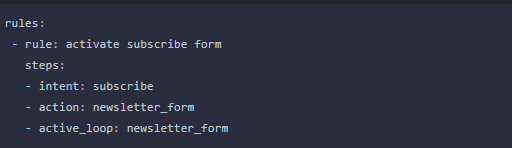
1. Mẫu (Forms)

Có nhiều tình huống mà trợ lý cần thu thập thông tin người dùng. Ví dụ người dùng yêu cầu đặt hàng chatbot sẽ hỏi người khách đó về yêu cầu các thông tin cho việc đặt hàng như: địa chỉ, số điện thoại, tên người nhận…



1. Luật (Rule)

Các quy tắc mô tả các phần của cuộc hội thoại luôn đi theo cùng một đường dẫn bất kể điều gì đã được nói trước đó. Mục đích là luôn hướng cuộc hội thoại đi theo một hướng và không quan tâm điều gì đã xảy ra xung quanh nó.



1. **Google Dialogflow**

Các thuật toán machine truyền thống yêu cầu đầu vào có cấu trúc và có thể đoán trước được hành động, điều này làm cho việc giao tiếp trong cuộc sống trở nên khó khăn đặc biệt là giao tiếp. Nếu như đầu vào của người dùng không có cấu trúc thì hệ thống khó phát hiện những việc gì cần làm. Chính vì lẽ đó mà cần xây dựng một chương trình giúp hiểu những ý định của người dùng và Dialogflow là một trong nền tảng hỗ trợ làm việc đó.

Tương tự như Rasa, Dialogflow là một nền tảng về ngôn ngữ tự nhiên (NLP) giúp phân tích, tích hợp giao diện trò chuyện vào ứng dụng di động, ứng dụng web.

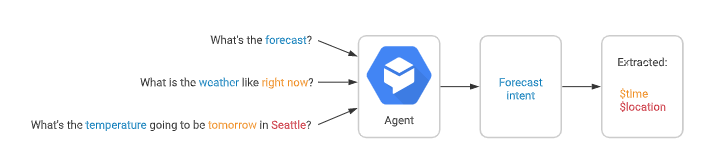
Các khái niệm cơ bản trong Dialogflow phần này sẽ có những khái niệm tương tự như Rasa nhưng được gọi với những thuật ngữ khác.

1. Agents

Agents Dialogflow là một tác nhân ảo xử lý cuộc trò chuyện với người dùng. Nó là một module giúp xử lý các vấn đề về ngôn ngữ tự nhiên để hiểu hiểu các sắc thái của ngôn ngữ con người. Dialogflow dịch văn bản hoặc âm thanh của người dùng trong trò chuyện sang dữ liệu có cấu túc mà các ứng dụng và dịch vụ có thể hiểu được. Agent Dialogflow như một bộ não trung tâm xử lý cuộc gọi của chatbot. Nó tương tự như phần NLU trong Rasa

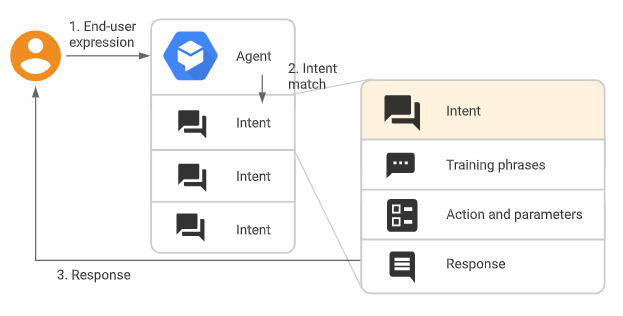
1. Intents (ý định)

Phân loại ý định là một trong thông tin quan trọng trong cuộc hội thoại với người dùng. Với mỗi agent chúng ta cần định nghĩa nhiều intent, ở đó chúng ta cần kết nối các intent để hoàn thành một cuộc trò chuyện. Khi một người dùng viết một cái gì đó Dialogflow so sánh đầu vào của người dùng với các ý định đã được định nghĩa sẵn. Vấn đề này được giải quyết bằng bài toán intent classification.



Intent cơ bản gồm những thành phần:

* Training phrase (Cụm từ đào tạo): Đây là cụm từ mà những người dùng có thể nói. Khi một đầu vào người dùng giống với cụm từ này này, thì Dialogflow sẽ khớp với mục đích. Khi đó chúng ta không cần phải xác định ngữ cảnh của tất cả câu.
* Actions (hành động) với mỗi ý định sẽ có một hành động, Dialogflow cung cấp hành động cho hệ thống hành động đó được sử dụng để kích hoạt các hành động xác định khác trong hệ thống.
* Parameters (Tham số): Khi một ý định được khớp trong thời gian chaỵ, Diaglogflow cung cấp các giá trị trích xuất từ biểu thức của người dùng dưới dạng tham số. Mỗi tham số có một số kiểu được gọi là kiểu thực thể cho biết chính xác cách dữ liệu được trích xuất.
* Responses: Có thể là text, âm thanh hay bất kì dạng mô tả dữ liệu mà hệ thống sẽ phản hồi lại cho người dùng. Đó chính là câu trả lời cho câu hỏi của người dùng.

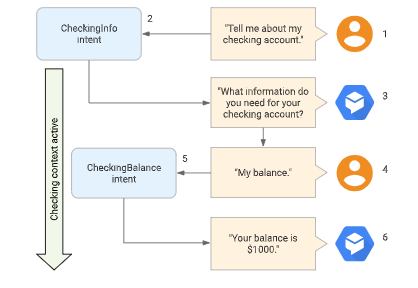


1. Entities

Dialogflow cung cấp các thực thể hệ thống được xác định trước có thể khớp với nhiều loại dữ liệu phổ biến. Ví dụ như ngày, địa chỉ email, tên công ty, tên tổ chức. Đây gắn liền với bài toán NER (Named Entity Recognition) nhận dạng tên thực thể.

1. Contexts

Bối cảnh của cuộc hội thoại tương tự như ngữ cảnh của ngôn ngữ trong tự nhiên. Nếu như một người nói “họ là công nhân”. Dialogflow xử lý đầu vào của người dùng giúp xác định chính xác ý định. Sử dụng ngữ cảnh bạn có thể kiểm soát luồng trò chuyện. Chẳng hạn như người dùng nói với bạn “họ đang đói” thì ngữ cảnh đây đang hướng đến việc “mua đồ ăn” hay “đề xuất đồ ăn”.



Sơ đồ bên trên cho thấy ví dụ ngữ cảnh cho một đại lý ngân hàng.

1. Người dùng yêu cầu thông tin về tài khoản
2. Dialogflow xác định được intent là CheckingInfo. Ý định này có một checking ngữ cảnh đầu ra do đó ngữ cảnh này được xác nhận. Ở đây ngữ cảnh xác lập là checking một cái gì đó.
3. Chatbot hỏi người dùng thông tin được checking
4. Người dùng trả lời bằng “balance”
5. Dialogflow tiếp tục so sánh đầu vào và khớp nó với các ý định tìm ra được ý định là CheckingBalance. Ý định này có checking ngữ cảnh đầu vào.
6. Hệ thống truy vấn trong cơ sở dữ liệu và đưa ra câu trả lời.
7. **Sự khác biệt giữa khung chatbot Rasa và khung chatbot Dialogflow**

Dialogflow là một sản phẩm nguồn đóng hoàn chỉnh với giao diện đồ họa web, API đầy đủ chức năng. Rasa (NLU + Core) là các thư viện python mở yêu cầu phát triển độ thấp hơn một chút.

Dialogflow

* Công cụ hoàn chỉnh cho chatbot. Hoàn thành hầu hết các nhiệm vụ cần cho chatbot.
* Có thể xử lý phân loại ý định, thực thể. Nó sử dụng những gì là bối cảnh để đối thoại.
* API mạnh mẽ cho phép xác định thực thể/ý định quá API với giao diện trên web.
* Dữ liệu lưu trữ trên đám mây và mọi tương tác API đều thực hiện trên đám mây

Rasa

* Để có thể có mức độ tương tự như Dialogflow, cần sử dụng cả Rasa NLU và Rasa Core. Rasa NLU xử lý các ý định/thực thể, Rasa Core xử lý vấn đề đối thoại.
* Rasa không cung cấp GUI nguồn mở hoàn chỉnh nên các tác vụ tương tác với NLU trong file Json hoặc markdown.

Rasa core yêu cầu phát triển trực tiếp bằng Python.

* Không cung cấp dịch vụ lưu trữ

1. **Lựa chọn cho đề tài**

Nếu như nhìn qua thì Dialogflow là một công cụ tuyệt vời cho xây dựng chatbot với hầu hết các thành phần. Nhưng mọi thứ triển khai của nó qua cloud vấn đề lớn nhất là độ trễ đường truyền trong khi giao tiếp. Hầu hết trong doanh nghiệp đều sử dụng một mạng nội bộ giúp bảo mật thông tin và giảm độ trễ khi giao tiếp các thành phần. Nhưng khi sử dụng dịch vụ từ Dialogflow thì lại gặp khó khăn. Bên cạnh đó Rasa cung cấp 2 phần tách biệt là NLU và Core giúp nhiệm vụ riêng, khi đó nhà nghiên cứu dễ dàng phát triển cũng như bảo trì cho hệ thống và có thể chỉ sử dụng một hoặc cả hai thành phần. Chính những rào cản đó nhóm tác giả lựa chọn Rasa như một phần trong việc xây dựng chatbot cho doanh nghiệp.

# **CHƯƠNG 3: XÂY DỰNG DỮ LIỆU VÀ TIỀN XỬ LÝ DỮ LIỆU**

Trong bước này thì nhóm tác giả trình bày quá trình xây dựng dữ liệu và cách xử lý dữ liệu trước khi được xử dụng huấn luyện các mô hình.

1. **Xây dựng dữ liệu**

Trong các bài toán dựa trên các kĩ thuật trí tuệ nhân tạo thì ắt hẳn vấn đề lớn nhất đó là dữ liệu. Trong đó chatbot cũng không phải là trường hợp ngoại lệ. Hầu hết chúng ta thấy có thể tìm kiếm dữ liệu trên các tập dữ liệu được public nhưng trong bài toán đặc thù là Office Bot các câu hỏi trong mỗi doanh nghiệp đều có cách trả lời mỗi câu hỏi khác nhau thay đổi theo thời gian. Chính lẽ đó để xây dựng một Office cần có một tập dữ liệu được xây dựng để phù hợp với bài toán và phù hợp với mô hình cũng như mục đích của doanh nghiệp.

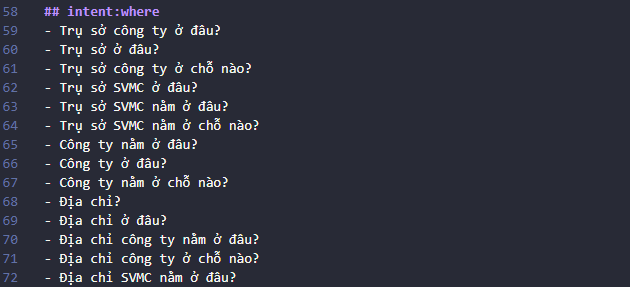
Trong khuôn khổ đề tài này nhóm tác giả tập trung các câu hỏi vào nhóm đối tượng là người dùng muốn hỏi về vấn đề nhân sự cũng như việc làm tại công ty SVMC (SamSung Việt Nam Mobile Center). Các câu hỏi tập trung vấn đề thường nhật trong công ty mà bất kể nhân viên nào cũng muốn hỏi rõ trước khi bắt đầu câu việc tại công ty như: lương, thưởng, chế độ thai sản, bảo hiểm lao động…

Tập dữ liệu được xây dựng thành 2 phần: theo cấu trúc của NLU data phù hợp cho việc training, huấn luyên mô hình sử dụng Rasa và theo cấu trúc thông thường nhằm nghiên cứu các phương pháp handcraft để xử lý.

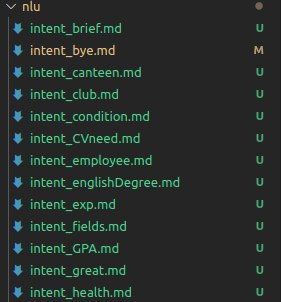
Quy trình xây dựng dữ liệu NLU data:

1. Xây dựng data cho Intent theo định dạng markdown format.

Tập dữ liệu được xây dựng với 32 intent mỗi intent cần được xác định rõ các câu hỏi người dùng có thể hỏi tương ứng với intent đó. Mỗi intent đều lưu thành một file riêng dưới dạng: intent\_name\_intent.md.



Các intent được lưu trong thư mục data/nlu



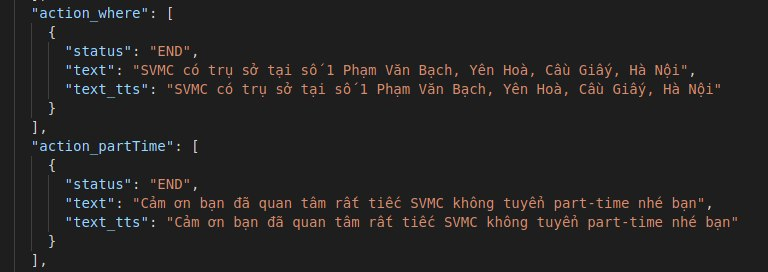
1. Xây dựng các actions cho các câu trả lời.

Các action được định nghĩa bằng một class, trong đó sẽ định nghĩa action nào được khớp với intent đã được xây dựng trong phần 1. Các action lần lượt được định nghĩa trong file actions.py với mã được viết bằng python.



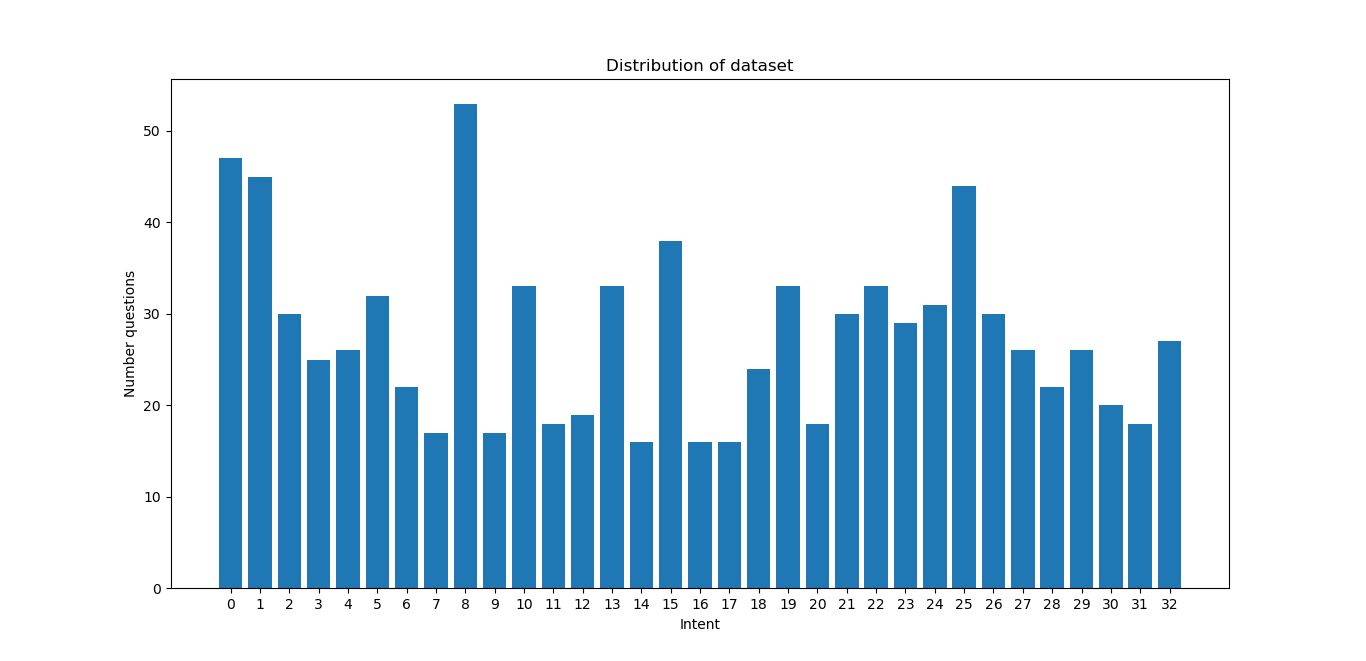
1. Xây dựng các câu phản hồi (câu trả lời)

Các câu trả lời được viết trong file config/config\_template\_action.json. Đây sẽ là lời nhắn của bot với những intent của người dùng.



Thống kê dữ liệu thu thập được:

Dữ liệu của 33 intents mỗi intent gồm một số câu hỏi được phân bổ như sơ đồ phân bố dữ liệu bên dưới, tổng số câu hỏi cho 33 itent là 914 questions.



1. **Tiền xử lý dữ liệu**

Thao tác tiền xử lý dữ liệu là một trong thao tác quan trọng giúp định hình, chuẩn hóa dữ liệu. Nó có liên quan mật thiết với độ chính xác của mô hình trong đề tài nhóm thực hiện các thao tác đơn giản nhằm chuẩn hóa dữ liệu và tăng tính đa dạng của dữ liệu.

Các thao tác đó bao gồm:

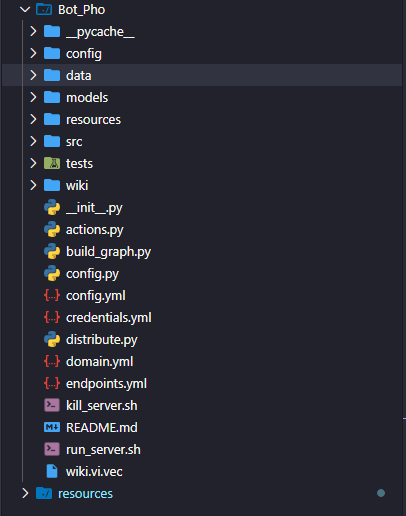
* Chuẩn hóa dấu câu, các câu có dấu câu được thêm vào và bỏ đi.
* Chuẩn hóa chữ đều dạng chữ viết thường
* Bổ sung thêm các từ viết tắt cho data, các từ thường sai chính tả.

# **CHƯƠNG 4 XÂY DỰNG KHỐI XỬ LÝ LÕI**

Trong chương này sẽ trình bày cách thức xây dựng khối lõi xử lý của chatbot bao gồm nhiều cách khác nhau đó là:

1. **Sử dụng Rasa Framwork:**

Cấu trúc project Rasa:



Trong đó:

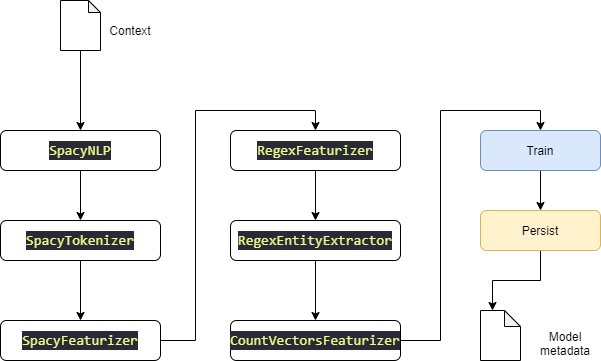
* Cofig chứa các file cofig cho phản hồi mà bot sẽ phản hồi lại cho người dùng được viết dưới dạng các file json.
* Data: thư mục chưa các file .md là dataset dùng để train khối NLU của Rasa
* Models: thư mục lưu lại model sau khi huấn luyện
* Resources: Các file bổ trợ cho việc sinh dữ liệu, và cofig ánh xạ giữa intent với action
* Src: chứa các file script python phục vụ cho việc custom các bước trong quá training như Word2Vect, mô hình classification.
* Wiki: chứa file được crawl trên wikimedia dùng để trích đặc trưng word2vect
* Actions.py cụ thể hóa các hành động sẽ được bot thực hiện
* Cofig.py các tham số cấu hình cho trainning là các path, các file được định nghĩa sẵn
* Cofig.yml file cấu hình cho khối NLU bao gồm việc trích đặc trưng, tham số cho training learning rate, số lượng epochs.
* Domail.yml định nghĩa các intent và các action đã được thiết kế
* Các file còn lại là các file bổ trợ để mở rộng Rasa project và để hướng dẫn chạy, huấn luyện mô hình.

Cofig cấu hình cho Rasa Project:

|  |  |  |
| --- | --- | --- |
| STT | Tham số | Gía trị |
| 1 | epochs | 100 |
| 2 | Batch\_size | [32, 64] |
| 3 | Entity\_recognition | False |
| 4 | Learning rate | 1e-4 |
| 5 | Mô hình ngôn ngữ | Spacy cho vi |
| 6 | Random\_seed | 666 |
| 7 | Trình tối ưu | Adam |
| 8 | Ngôn ngữ | vi |

Các tham số còn lại đều sử dụng giá trị default trong Rasa.

Turning NLU model theo đường ống được minh họa hình vẽ bên dưới:



Các thao tác đó bao gồm:

1/ SpacyNLP: khởi tạo mô hình ngôn ngữ được đào tạo trước bởi Spacy. Vậy mô hình ngôn ngữ là gì?

Mô hình ngôn ngữ là xác xuất phân bố trên tập văn bản. Nói một cách đơn giản, mô hình ngôn ngữ cho biết xác xuất cho một câu (cụm từ) thuộc một ngôn ngữ là bao nhiêu. Ví dụ khi áp dụng mô hình ngôn ngữ cho tiếng Việt.

P [“hôm qua là thứ năm] = 0.001.

P [“hôm qua là hôm qua”] = 0.

2/ SpacyTokenizer có tác dụng phân đoạn văn bản thành các từ và các dấu câu. Ngoài ra chúng ta có thể sử dụng WhiteSpaceTokenizer như một cách phân đoạn văn bản bằng dấu cách.

3/ SpacyFeaturizer biểu diễn câu văn bản thành vector sử dụng trình tạo vector của spacy. Đầu ra của quá trình này giúp chúng ta nhận được ma trận đặc điểm về trình tự có kích thước (number-of-tokens x featur-dimension) và một ma trận biểu diễn tính năng của câu kích thước (1x feature-dimensions).

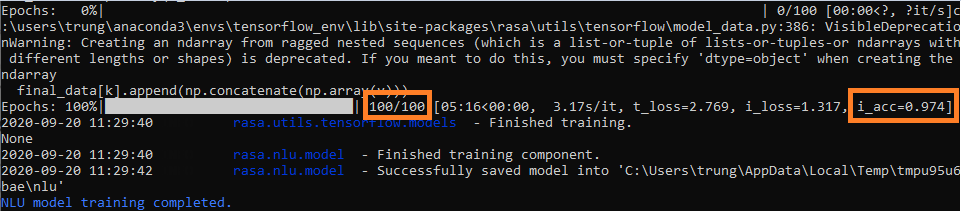
4/ Regex Feature: tạo biểu diễn vector thông điệp người dùng. Tạo ra các feature để trích xuất thực thể và phân loại các intent.

5/ RegexEntinyExtractor: mô hình nhận diện thực theo phương pháp CRF. (CRF sẽ được trình bày trong phần sau trong bài toán nhận diện tên thực thể NER)

6/ CountVectorFeaturizer: đếm số lần xuất hiện của mỗi từ trong câu mà có xuất hiện trong tập từ điển đã loại bỏ các stop word. Phương thức này được triển khai trong sklearn. Ngoài ra nó có thể được cấu hình để đếm các kí tự khi tùy chỉnh tham số analyzer.

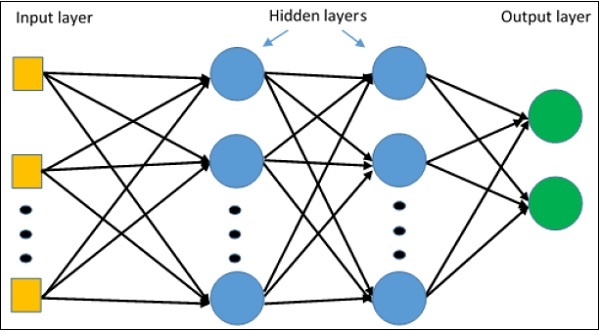
7/ Train: tham số được thiết lập là DIETClassifier là một kiến trúc đa tác vụ để phân loại ý định và nhận diện thực thể. Kiến trúc này nhận diện thực thể theo phương pháp CRF, bộ phận loại được triển khai là một mạng no-ron truyền thẳng (MLP multi layer classification) kiến trúc của nó sẽ được trình bày ở phần sau. Hidden\_layers\_sizes có thể điều chỉnh được (mặc định là 300). Tổng số lượng tham số của mô hình là 2.821.172

Training và kết quả:



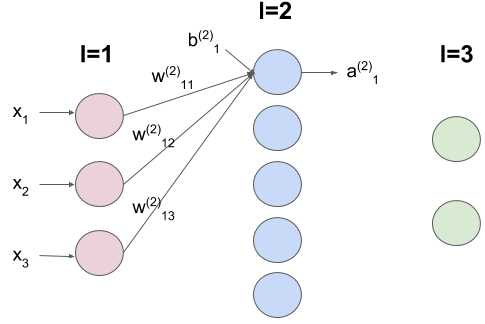
Kết quả huấn luyện của mô hình đạt 97.4% trên tập dữ liệu đào tạo. <Tập test đang được xây dựng và kiểm tra trong>

1. **Sử dụng machine learning truyền thống**
   1. **Trích đặc trưng: CountVector TF-IDF <chưa viết>**
   2. **Phân loại intent dựa trên các thuật toán machine truyền thống:**
2. **Mutil Layer Perceptron (MLP):**



Gọi là Multi-layer Perceptron (perceptron nhiều lớp) bởi vì nó là tập hợp của các perceptron chia làm nhiều nhóm, mỗi nhóm tương ứng với một layer. Trong hình trên ta có một ANN với 3 lớp: Input layer (lớp đầu vào), Output layer (lớp đầu ra) và Hidden layer (lớp ẩn). Thông thường khi giải quyết một bài toán ta chỉ quan tâm đến input và output của một model, do vậy trong MLP nói riêng và ANN nói chung ngoài lớp Input và Output ra thì các lớp neuron ở giữa được gọi chung là Hidden (ẩn không phải là không nhìn thấy mà đơn giản là không quan tâm đến).

Tư tưởng của MLP có thể được mô tả ngắn gọn như sau: Nó là sự mô tả sự lan truyền thông tin của các nơ-ron (các hình tròn trong hình). Sự truyển thông tin như các tầng khác nhau đến tầng cao hơn. Tầng cao hơn nhận được thông tin từ tầng thấp hơn và lan truyền đến tầng khác.



Trong hình vẽ trên mô tả nguyên lý hoạt động của một mạng nơ-ron đa tầng. Đầu vào là vector có 3 phần tử: [x1, x2, x3]. Thông tin đầu vào được truyền đến nơ-ron khác tại tần thứ 2 (I=2). Tại đó thông tin nhận được theo một phương trình tuyến tính:



Và cuối cùng trước khi đưa ra đầu ra nó được đưa ra qua một hàm activation (tương tự như thuật toán 1) hàm này giúp đưa ra xác suất mà đầu vào mỗi nhóm phân loại.

* Tại sao lại có sự lan truyền này?

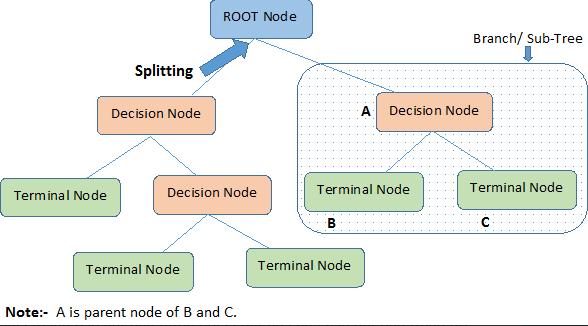
Điều này đúng đắn và được chứng minh cho việc học ánh xạ từ đầu vào thành đầu ra của tập dữ liệu. Trong quá trình huấn luyện thì các trọng số w sẽ được thay đổi và điều chỉnh sao cho việc ánh xạ đầu vào đạt hiệu quả nhất với việc đưa ra đầu ra. Điều này giống việc bạn thực hiện tìm hệ số a, b trong phương trình y = ax + b khi bạn biết x và y.

Thuật toán điều chỉnh trọng số w đó là Backpropagation.

1. **Decision Tree:**

Cây quyết định là một loại thuật toán học tập có giám sát được sử dụng nhiều trong bài toán phân loại.

Có 2 loại cây quyết định phụ thuộc vào mục tiếp áp dụng nó: mục tiêu phân loại với mục tiêu liêu tục ở biến đầu ra.



Tư tưởng của thuật toán cây quyết định đó là xây dựng một cây với các nhánh. Một cây quyết định là một đồ thị của các quyết định và các hậu quả có thể của nó. Cây quyết định được sử dụng để xây dựng một kế hoạch nhằm đạt được mục tiêu mong muốn. Các cây quyết định được dùng để hỗ trợ quá trình ra quyết định. Cây quyết định là một dạng đặc biệt của cấu trúc cây.

Xét bài toán: Có 30 học sinh gồm 3 biến đặc trưng cơ bản:

x: giới tính (Nam/Nữ) y: lớp (IX, X) z: chiều cao (5-6 ft). Trong đó có 15/30 chơi mobile ledgend bang bang trong thời gian giải trí.

Mục đích chúng ta cần phân loại nhóm sinh viên ra 2 loại chơi game MLBB trong giờ giải trí ra khi biết các đặc trưng đầu vào x, y, z với từng học sinh.

Để giải bài toán này bằng cây quyết định ta phải thử lần lượt các thuộc tính:

Xét thuộc tính x = [Nam, Nu]

Với thuộc tính này chia tập dữ liệu thành 2 nhóm:

+ Nhóm 1: Toàn sinh viên Nam (20 sinh viên, 13 người chơi game)

+ Nhóm 2: Toàn sinh viên Nữ (10 sinh viên, 2 người chơi game)

Tương tự cho 2 loại thuộc tính như mỗi thuộc tính ta đc 2 nhóm

Thuộc tính y:

+ Nhóm 1: Học sinh lớp IX (14 người, 6 người chơi game)

+ Nhóm 2: Học sinh lớp X (16 người, 9 người chơi game)

Thuộc tính z:

+ Nhóm 1: Học sinh cao trên >= 5.5ft (18 người, 10 người chơi game)

+ Nhóm 2: Học sinh thấp hơn < 5.5ft (12 người, 5 người chơi game)

Vậy thì lựa chọn thuộc tính nào để phân loại? Tiêu chí nào đánh giá thuộc tính nào phân loại tốt hơn.

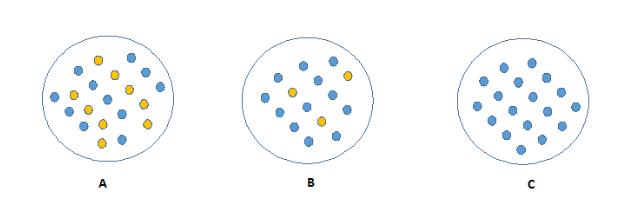
**Gini (chỉ số Gini):**

* Áp dụng cho phân chia cây dạng nhị phân
* Một tập bất kì khi ta chia 2 tập con từ tập gốc thì xác suất mỗi tập được chia ra là như nhau và xác suất khi tập đó đồng nhất (chỉ có 1 loại) là 1
* Chỉ số Gini càng cao thì chứng tỏ chỉ số đồng nhất càng cao tức mỗi tập đó gần giống nhau nhiều nhất

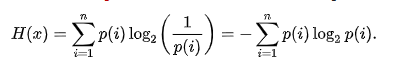
Cách tính chỉ số Gini:

Tính toán Gini cho các nút phụ, sử dụng tổng công thức bình phương xác suất thành công và thất bại (p ^ 2 + q ^ 2). Tính toán Gini cho phân chia bằng cách sử dụng điểm Gini có trọng số của từng nút của phân chia đó

**Entropy:**



* Theo trực giác quan sát được hình sẽ ít tạp chất hơn 2 hình A và B. Do tập C có nhiều các points tương đồng. Từ đó ta suy ra được C cần ít thông tin hơn để mô tả nó.
* Lý thuyết thông tin là một biện pháp để xác định mức độ vô tổ chức này trong một hệ thống được gọi là Entropy. Nếu mẫu hoàn toàn đồng nhất, thì entropy bằng 0 và nếu mẫu được chia đều (50% - 50%) thì nó có entropy là 1. Công thức tính entropy là:



1. **Random Forest:**

Random Forest là một phương pháp học máy đa năng có khả năng thực hiện cả các nhiệm vụ hồi quy và phân loại. Nó cũng thực hiện các phương pháp giảm kích thước, xử lý các giá trị còn thiếu, giá trị ngoại lệ và các bước thiết yếu khác của khám phá dữ liệu và thực hiện công việc khá tốt. Nó là một loại phương pháp học tập đồng bộ, trong đó một nhóm các mô hình yếu kết hợp với nhau để tạo thành một mô hình mạnh mẽ.

* Nó hoạt động ra sao?

Ý tưởng: Đối với thuật toán cây quyết định chúng ta chỉ dùng một cây duy nhất cho tác vụ phân loại hoặc hồi quy, trong rừng ngẫu nhiên chúng ta trồng nhiều cây một cách ngẫu nhiên. Mỗi cây đưa ra một phân loại và nói rằng cây đã đưa ra 1 phiếu bầu cho lớp đó và tổng số phiếu bầu lớp nào cao nhất là đầu ra dự đoán.

Vậy trồng cây thế nào?

1. Gỉa sử có n mẫu dữ liệu cho huấn luyện, ta lấy ngẫu nhiên tập con trong số n mẫu huấn luyện đã cho và có thay thế mẫu con trong quá trình huấn luyện.
2. Gỉa sử có M giá trị đầu vào, ta sử dụng m giá trị (m < M) để xây dựng cây như cây quyết định
3. Mỗi cây được trồng kích thước lớn nhất và không cắt tỉa.
4. Dự đoán là tổng hợp dự đoán của tất cả các cây.

Ưu điểm của rừng ngẫu nhiên?

* Thuật toán này giải quyết được cả vấn đề hồi quy và phân loại.
* Một trong những lợi ích của Rừng ngẫu nhiên là, sức mạnh xử lý tập dữ liệu lớn với tính chiều cao hơn. Nó có thể xử lý hàng ngàn biến đầu vào và xác định hầu hết các biến quan trọng để nó được coi là một trong những phương pháp giảm kích thước.
* Nó có một phương pháp hiệu quả để ước tính dữ liệu bị thiếu và duy trì độ chính xác khi thiếu một tỷ lệ lớn dữ liệu.
* Nó có các phương thức để cân bằng các lỗi trong các tập dữ liệu trong đó các lớp bị mất cân bằng. Các khả năng của ở trên có thể được mở rộng thành dữ liệu chưa được gán nhãn, dẫn đến phân cụm không được giám sát, xem dữ liệu và phát hiện ngoại lệ.
* Random Forest bao gồm lấy mẫu dữ liệu đầu vào với sự thay thế được gọi là lấy mẫu bootstrap. Ở đây một phần ba dữ liệu không được sử dụng cho đào tạo và có thể được sử dụng để kiểm tra. Chúng được gọi là mẫu ra khỏi túi. Lỗi ước tính trên các mẫu trong túi này được gọi là lỗi túi. Nghiên cứu ước tính lỗi bằng Out of bag, đưa ra bằng chứng cho thấy ước tính hết túi là chính xác như sử dụng một bộ kiểm tra có cùng kích thước với tập huấn luyện. Do đó, việc sử dụng ước tính lỗi xuất túi sẽ loại bỏ sự cần thiết của một bộ kiểm tra dành riêng.

Nhược điểm: Rất tốt cho nhiệm vụ phân loại, không tốt cho nhiệm vụ hồi quy.

1. **Bảng so sánh kết quả: (Chưa có)**
2. Cây quyết định
3. SVM
4. MLP
5. Randomforest

Đánh giá và nhận xét và đưa ra tốc độ dự đoán của các mô hình cho việc phát hiện intent.

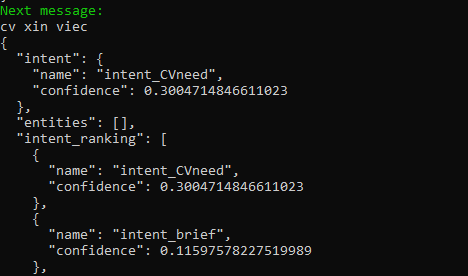
1. **Sử dụng deeplearning**
   1. **Nhận diện tên thực thể: CRF**
   2. **Nhận diện intent:**

* CNN, LSTM, BiLSTM các mạng lai.
  1. **Đánh giá đưa ra nhận xét về hiệu năng và tốc độ.**

# **CHƯƠNG 5: TÍCH HỢP KHỐI XỬ LÝ LÕI VÀO ỨNG DỤNG CHAT**

Nhóm mới code và đưa vào chạy trên web, chưa tích hợp vào các ứng dụng

Test sử dụng rasa shell:



Một vài kết quả demo trên web tại localhost:

