**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT – CÔNG NGHỆ**

**ĐỒ ÁN MÔN HỌC**

**ĐỒ ÁN CHUYÊN NGÀNH**

**Đề tài:** Xây dựng hệ thống Chatbot hỗ trợ tư vấn điện thoại

cho siêu thị Điện Máy Xanh

**GVHD :** DƯƠNG THỊ KIM CHI

**NHÓM SVTH :** Lương Thế Vinh 1724801040099

Nguyễn Quang Huy 1724801040128

Trần Văn Thái1724801040081

**LỚP :** D17HT02

*Bình Dương, ngày 25 tháng 11 năm 2020*

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT - CÔNG NGHỆ**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

Họ và tên giảng viên:  **Dương Thị Kim Chi**

Tên đề tài: **Xây dựng Hệ thống Chatbot hỗ trợ tư vấn điện thoại di động cho Siêu thị Điện Máy Xanh**

Nội dung nhận xét:

**Điểm:**

Bằng số:

Bằng chữ:

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN CHẤM**  *(Ký, ghi rõ họ tên)* |

**ThS. Dương Thị Kim Chi**

# LỜI NÓI ĐẦU

Hiện nay, khoa học công nghệ đang dần tiến lên một tầm cao mới. Mặc dù còn mới mẻ trong lĩnh vực khoa học công nghệ nhưng chatbot đang được nghiên cứu và phát triển với tốc độ chóng mặt bởi các trung tâm nghiên cứu, các trường đại học và học viện… rất nhiều các lĩnh vực được ứng dụng công nghệ mới này.

Chatbot là hình thức thô sơ của phầm mềm trí tuệ nhân tạo có thể trả lời các câu hỏi và xử lý tình huống, là công cụ có thể giao tiếp với con người thông qua một trí tuệ nhân tạo đã được lập trình sẵn.

Chatbot đang ngày càng được phổ biến rộng rãi bởi tính chính xác và nhanh chóng của một trí tuệ nhân tạo , mang lại không ít lợi ích cho người dùng mà còn giảm thiểu được kha khá chi phí , nhân công cho ngành kinh doanh .

Ngành kinh doanh nói riêng đang là môi trường nóng cho sự phát triển của Chatbot, nó đóng vai trò như một nhân viên tư vấn bán hàng , trả lời toàn bộ những câu hỏi được gửi về một cách nhanh chóng , không chỉ thế nó còn học hỏi các dữ liệu câu hỏi để tăng độ chính xác trong câu trả lời .

# MỤC LỤC

[LỜI NÓI ĐẦU i](#_Toc58584608)

[MỤC LỤC ii](#_Toc58584609)

[DANH MỤC HÌNH iv](#_Toc58584610)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc58584611)

[1.1 Đặt Vấn Đề 1](#_Toc58584612)

[1.2 Mục Tiêu : 2](#_Toc58584613)

[1.3 Nội Dung Nghiên Cứu 3](#_Toc58584614)

[1.4 Giới Hạn 3](#_Toc58584615)

[1.5 Môi trường lập trình 3](#_Toc58584616)

[1.6 Bố cục 4](#_Toc58584617)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc58584618)

[2.1. Khái niệm CHATBOT 5](#_Toc58584619)

[2.2. Lịch sử hình thành. 6](#_Toc58584620)

[2.3. Phân loại Chatbot: 15](#_Toc58584621)

[2.4. Cấu tạo Chatbot: 23](#_Toc58584622)

[2.5. Cách thức hoạt động: 23](#_Toc58584623)

[2.6. Một số ứng dụng của Chatbot: 25](#_Toc58584624)

[2.7. Xu hướng phát triển: 25](#_Toc58584625)

[2.8. Xây Dựng Chatbot tư vấn mua điện thoại di động cho Điện Máy Xanh 26](#_Toc58584626)

[CHƯƠNG 3: KỸ THUẬT ĐƯỢC SỬ DỤNG TRONG CHATBOT 29](#_Toc58584627)

[3.1 Một số kỹ thuật 29](#_Toc58584628)

[3.2 Kết luận chương 33](#_Toc58584629)

[CHƯƠNG 4: XÂY DỰNG VÀ CHẠY THỬ CHATBOT 33](#_Toc58584630)

[4.1. Kiến trúc ứng dụng 33](#_Toc58584631)

[4.2. Quá trình xây dựng 34](#_Toc58584632)

[4.3. Chạy thử 42](#_Toc58584633)

[4.4 Kết quả đạt được 46](#_Toc58584634)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 49](#_Toc58584635)

[1. Kết luận: 49](#_Toc58584636)

[2. Hướng phát triển: 50](#_Toc58584637)

[3. Hạn chế 50](#_Toc58584638)

[TÀI LIỆU THAM KHẢO 51](#_Toc58584639)

# DANH MỤC HÌNH

[Hình 1: Các phép thử 8](#_Toc58576370)

[Hình 2: ELIZA - 1966 9](#_Toc58576371)

[Hình 3: Chatbot Jabberwacky - 1988 11](#_Toc58576372)

[Hình 4: Dr. SBAITSO - 1992 11](#_Toc58576373)

[Hình 5: SmarterChild - 2001 13](#_Toc58576374)

[Hình 6: IBM Watson Chatbot 15](#_Toc58576375)

[Hình 7: Chatbot chăm sóc khách hàng 18](#_Toc58576376)

[Hình 8: Chat bot theo kịch bản 21](#_Toc58576377)

[Hình 10: : Chatbot nhận diện từ khóa 22](#_Toc58576378)

[Hình 11: Chatbot trò chuyện theo ngữ cảnh 23](#_Toc58576379)

[Hình 12: Đánh trọng số với TF 33](#_Toc58576380)

[Hình 13: Đánh trọng số với IDF 33](#_Toc58576381)

[Hình 14: TF-IDF 34](#_Toc58576382)

[Hình 15 : Kiến trúc của ứng dụng chatbot 36](#_Toc58576383)

[Hình 16: Trò chuyện cùng chatbot 49](#_Toc58576384)

[Hình 17: Chatbot không hiểu câu hỏi 50](#_Toc58576385)

[Hình 18: Quá trình training 51](#_Toc58576386)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1 Đặt Vấn Đề

Nhờ những thành tựu trong công nghệ, chatbot đã ra đời và trở nên phổ biến hơn bao giờ hết trong những năm gần đây. Đây là công cụ hữu ích được áp dụng trong hầu hết mọi lĩnh vực từ giáo dục , kinh doanh , y tế , giải trí , tự động hóa ,…

Kể từ khi chatbots được phát triển mạnh mẽ vào năm 2016, nó đã nhanh chóng trở thành một trong những xu hướng công nghệ được quan tâm nhất trong giai đoạn đó và cho đến nay. Trên thực tế cho thấy, sự gia tăng của chatbots đã dẫn đến dự đoán của Gartner (công ty nghiên cứu và tư vấn công nghệ thông tin hàng đầu thế giới, cung cấp cái nhìn sâu sắc liên quan đến công nghệ cần thiết cho khách hàng để đưa ra những quyết định đúng đắn mỗi ngày) rằng hơn 85% tương tác của khách hàng sẽ được quản lý mà không cần con người vào năm 2020. Chatbots phát triển dựa trên sự kết hợp của các kịch bản có trước và tự học trong quá trình tương tác. Ta sẽ tương tác với chatbots qua nền tảng tin nhắn. Với các câu hỏi được đặt ra, Chatbots sử dụng các hệ thống xử lý ngôn ngữ tự nhiên (Natural Language Processing) để phân tích dữ liệu sau đó chúng lựa chọn các thuật toán học máy để đưa ra các loại phản hồi khác nhau, chúng sẽ dự đoán và phản hồi chính xác nhất có thể. Chatbots sử dụng nhiều hệ thống quét các từ khoá bên trong đầu vào, sau đó bot khởi động một hành động, kéo một câu trả lời với các từ khóa phù hợp nhất và trả lời thông tin từ một cơ sở dữ liệu / API, hoặc bàn giao cho con người. Nếu tình huống đó chưa xảy ra (không có trong dữ liệu), Chatbot sẽ bỏ qua nhưng sẽ đồng thời tự học để áp dụng cho các cuộc trò chuyện về sau. Một trong các yếu tố làm nên sức mạnh của Chatbot là khả năng tự học hỏi. Càng được sử dụng, tương tác với người dùng nhiều, nền tảng Chatbot càng “thông minh”. Chatbot thông minh có khả năng tự học hỏi dựa trên các dữ liệu đưa vào mà không cần phải được lập trình cụ thể (đó được gọi là phương pháp máy học - Machine Learning). Chính điều này làm cho các nhà phát triển dễ dàng tạo các chương trình trò chuyện và tự động hoá các cuộc trò chuyện với người dùng.

Nắm bắt được tiềm lực phát triển mạnh mẽ của chatbot , chúng em quết định nghiên cứu và xây dựng một *“* ***ChatBot tư vấn mua bán điện thoại di động hỗ trợ Điện Máy Xanh****”* . Cũng như bao lĩnh vực khác , chat bot trong kinh doanh mua bán sẽ là trợ thủ đắc lực dành cho cửa hàng , dữ liệu của chatbot cũng sẽ được nâng cao liên tục vì nhờ tính tự học dữ liệu khi tiếp xúc với khách hàng , cũng như giải quyết được vấn đề cá nhân hóa của khách hàng , ngày càng tiên tiến sẽ đẩy mạnh hình ảnh và thương hiệu của cửa hàng tới người tiêu dùng trong mảng dịch vụ chăm sóc khách hàng .

## 1.2 Mục Tiêu :

Trong bài báo cáo này , mục tiêu chính là xây dựng được một Chatbot tự động tư vấn hỗ trợ trong việc bán điện thoại ở điện máy xanh , cụ thể hệ thống sẽ xây dựng trên nền tảng của Dialogflow và ngôn ngữ python . Nghiên cứu và tìm hiểu các kỹ thuật hỗ trợ trong việc xây dựng chatbot như là kỹ thuật tách từ , sửa lỗi chính tả , sử dụng xác suất để lọc dữ liệu , ….

Làm ra sản phẩm , train dữ liệu và bắt đầu test các chức năng của chatbot

## 1.3 Nội Dung Nghiên Cứu

* NỘI DUNG 1: Tìm hiểu về Chatbots, Tensorflow,
* NỘI DUNG 2: Nghiên cứu ngôn ngữ python trên giao diện Pycharm
* NỘI DUNG 3: Xây dựng hệ thống chatbot
* NỘI DUNG 4 Đánh giá kết quả thực hiện.
* NỘI DUNG 5: Viết báo cáo

## 1.4 Giới Hạn

Tìm hiểu về Chatbots và ứng dụng Chatbots vào bộ phận tư vấn bán hàng điện thoại di động

Chatbot trả lời các câu hỏi của khách hàng về các vấn đề liên quan tới công việc mua bán điện thoại di động

Hệ thống hoạt động thông qua việc tương tác giữa người với máy tính và gửi các yêu cầu xuống cho phần cứng thực hiện.

Kết quả thu được bao gồm việc trả lời tự động từ máy tính, dữ liệu được lưu lại và chatbot tự học hỏi

## 1.5 Môi trường lập trình

Sử dụng các phần mềm :

* Sublime Text
* Visual Studio Code
* Python 3.8.5
* Pycharm

## 1.6 Bố cục

Chương 1: Tổng Quan. Chương này trình bày đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, giới hạn và bố cục đồ án.

Chương 2: Cơ Sở Lý Thuyết.

Chương 3: Các kỹ thuật được sử dụng trong Chatbot

Chương 4: Xây dựng và Chạy demo Chatbot

Chương 5: Kết Quả, Nhận Xét và Đánh Giá. Các kết quả đạt được khi thực hiện chương trình, phân tích, nhận xét, đánh giá kết quả thu được.

Chương 6: Kết Luận và Hướng Phát Triển. Tóm tắt những kết quả đạt được, những hạn chế và nêu lên các hướng phát triển trong tương lai.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Khái niệm CHATBOT

CHATBOT bao gồm “Chat” và “Bot”

“Bot” là một phần mềm thực hiện các nhiệm vụ, công việc do con người yêu cầu một cách tự động. Hoặc “Bot” cũng có thể là một chương trình máy tính được lập trình để “giao tiếp” với người dùng thông qua kết nối Internet.

“Chat” nghĩa là trò chuyện, giao tiếp qua lại giữa hai cá thể.

Gộp chung hai khái niệm này lại chúng ta có được khái niệm tổng quát cho Chatbot như sau:

**Chatbot** là một hệ thống các Bot trong trạng thái trực tuyến trên các Website hoặc tại các nền tảng, giao diện chat khác của các trang mạng xã hội dùng để “tương tác tự động” - “ chat tự động” với người dùng.

**Chatbot** được hiểu thông thường như một “cái máy” có thể đối thoại một cách tự nhiên với con người. Ví dụ: Khi một người dùng bất kỳ hỏi một câu hỏi bất kỳ hoặc cho một câu lệnh bất kỳ, Bot sẽ tự động trả lời hoặc thực hiện các câu lệnh một cách phù hợp để đáp lại người dùng.

**Chatbot** tương tác với con người như một hệ thống trả lời tin nhắn nhanh chóng, tự động. Bằng cách xây dựng, giả lập mô hình tương tác cũng như các kịch bản có thể xảy ra như con người thông qua phương pháp Machine Learning, hệ thống Chatbot có thể “tự học”, “tự hiểu” các câu hỏi, nhu cầu của người dùng và thực hiện chúng.

**Chatbot** sử dụng Database - cơ sở dữ liệu, các câu đối thoại đã được “huấn luyện” để phản hồi lại cho người dùng tại bất kỳ thời điểm nào. Trong trường hợp Chatbot không hiểu câu hỏi, có thể do quá trình “huấn luyện” chưa kỹ lưỡng. Tuy nhiên khi gặp các trường hợp đó Chatbot sẽ chuyển thông tin đến người quản lý khi không hiểu thông tin từ người dùng.

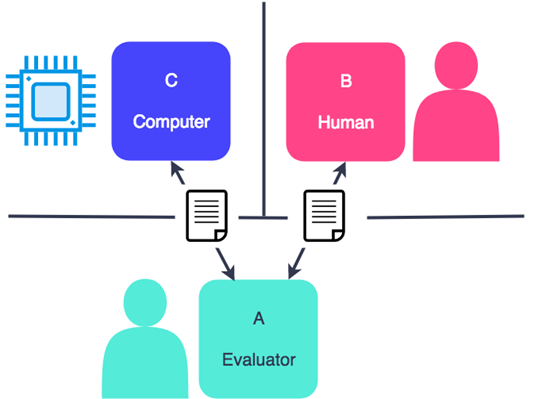
Hiện nay có rất nhiều các ứng dụng Chatbot nổi tiếng như Alexa, Siri, Cortana, Misuku, Poncho,... với các tiện ích hỗ trợ ở nhiều lĩnh vực khác nhau từ e-commerce đến du lịch,...

Trong cuộc sống hiện tại bạn có thể đã sử dụng qua Chatbot, ví dụ khi bạn nói “Hi Siri” để bắt đầu trò chuyện với Siri của Apple hay “OK google” khi sử dụng Google Home.

## 2.2. Lịch sử hình thành.

### 2.2.1. Các phép thử Turning - 1950:

Alan Turing viết báo cáo với tựa đề “Computing Machinery and Intelligence”, xây dựng các phép thử Turing. Về cơ bản, Các phép thử Turing là một loạt các phép thử dựa trên việc phân tích câu trả lời của một “máy tính”. Các phép thử Turing được thực hiện để xác định xem một chương trình máy tính có thể phân biệt được máy tính với con người trong một cuộc trò chuyện chỉ có văn bản thuần túy hay không?

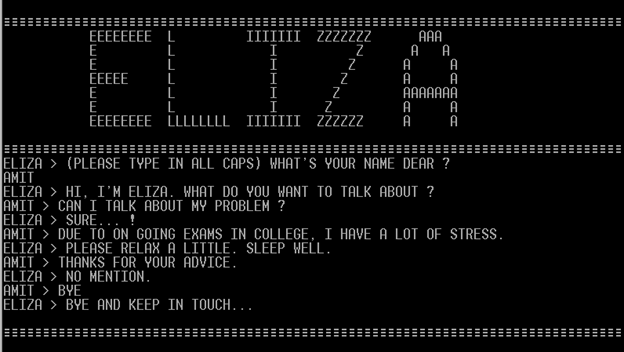
****

Hình 1: Các phép thử

Bằng cách gõ câu hỏi cho cả hai đối tượng thử nghiệm, người thẩm vấn sẽ cố gắng xác định đối tượng nào là máy tính và đối tượng nào là con người. Máy tính sẽ vượt qua Các phép thử Turing nếu người thẩm vấn không thể nói sự khác biệt giữa chủ thể con người và máy tính.

### 2.2.2. ELIZA - 1966:

Joseph Weizenbaum xuất bản chương trình ELIZA, được coi là một trong những chương trình Chatbots đầu tiên trên thế giới. ELIZA đã đạt được những thành tích đáng kể và được coi là thành tựu đỉnh cao về trí thông minh nhân tạo vào thời điểm đó. Bằng cách nhận ra các từ và cụm từ chính từ đầu vào (Input) của người dùng và đưa những câu trả lời tương ứng bằng cách sử dụng các tập lệnh viết sẵn.

****

Hình 2: **ELIZA - 1966**

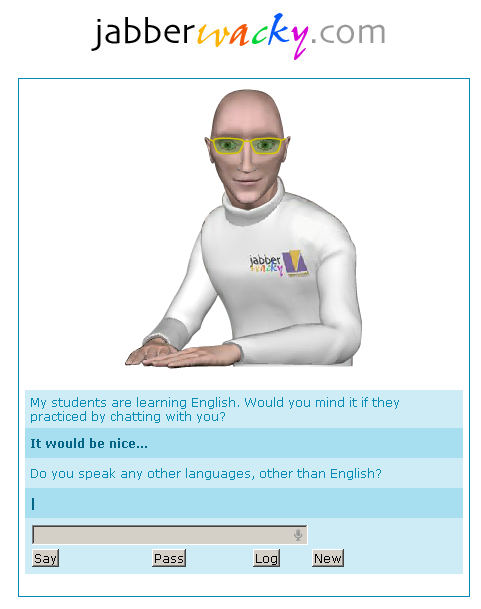
Một trong những kịch bản này có tên là DOCTOR, cho phép ELIZA đảm nhận vai trò như một nhà tâm lý học hay một bác sỹ tâm thần. Chatbot ELIZA có thể được cải thiện và nâng cấp từng bước bằng cách chỉnh sửa các kịch bản Chatbot của ELIZA. Khái niệm kịch bản Chatbots cũng được hình thành từ thời điểm này.

### 2.2.3. Chatbot Jabberwacky - 1988:

Được phát triển vào những năm 1980 và phát hành trực tuyến vào năm 1997, chatbot Jabberwacky được thiết kế để “Mô phỏng trò chuyện của con người tự nhiên theo cách thú vị và hài hước“. Mục đích ban đầu của dự án Chatbot Jabberwacky là tạo ra một trí tuệ nhân tạo có khả năng vượt qua Các phép thử Turing.

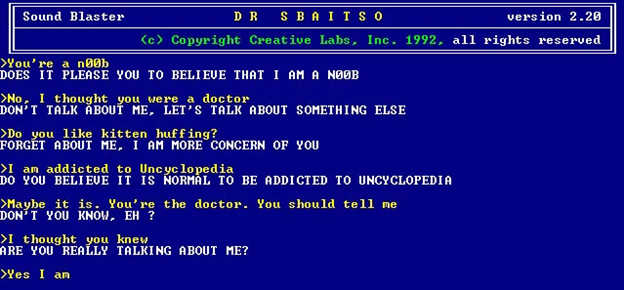
Nó được thiết kế để bắt chước tương tác của con người và thực hiện các cuộc hội thoại với người dùng. Mục đích cuối cùng của chương trình là chuyển từ một hệ thống dựa trên văn bản sang toàn bộ hoạt động bằng giọng nói. Tác giả của nó tin rằng nó có thể được kết hợp vào các vật thể xung quanh nhà như robot, các thiết bị thông minh,…

Trong khi tất cả các chatbot trước đó dựa trên cơ sở dữ liệu tĩnh để trả lời và trò chuyện, Jabberwacky thu thập cụm từ được sử dụng bởi những người tham gia trò chuyện với nó. Nó tự thêm những câu trả lời vào cơ sở dữ liệu và tự động phát triển nội dung của riêng mình. Trong năm 2008, Jabberwacky đã phát hành một phiên bản mới và đổi tên thành Cleverbot.



Hình 3: **Chatbot Jabberwacky - 1988**

### 2.2.4. Dr. SBAITSO - 1992:



Hình 4: **Dr. SBAITSO - 1992**

Được tạo ra bởi Creative Labs vào đầu những năm 1990, Dr SBAITSO là từ viết tắt của Sound Blaster Artificial Intelligent Text to Speech Operator).

**Dr. SBAITSO** “trò chuyện” với người dùng như thể nó là một nhà tâm lý học. Mặc dù hầu hết các câu trả lời của nó đều là “WHY DO YOU FEEL THAT WAY?” nghĩa là “Bạn cảm thấy như thế nào?”.

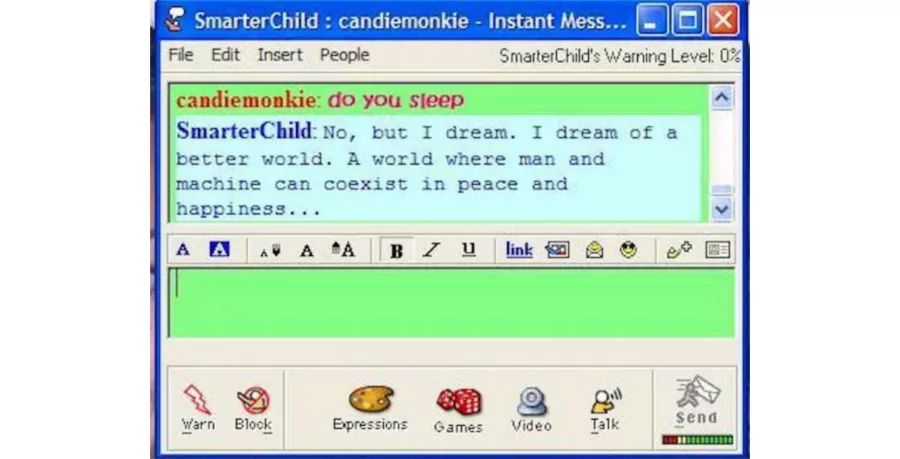
### 2.2.5. ALICE - 1995:

ALICE được xây dựng trên cùng một kỹ thuật được sử dụng để tạo nên ELIZA. ALICE ban đầu được sáng tạo bởi Richard Wallace, ra đời vào ngày 23 tháng 11 năm 1995. Chương trình được viết lại bằng ngôn ngữ Java vào năm 1998. ALICEBOT sử dụng một lược đồ XML có tên AIML (Artificial Intelligence Markup Language- Ngôn ngữ đánh dấu trí thông minh nhân tạo) để xác định các quy tắc trò chuyện heuristic. Tuy nhiên, nó lại không thể vượt qua Các phép thử Turing - 1950

### 2.2.6. SmarterChild - 2001:

SmarterChild là một Chatbot có sẵn trên mạng AOL Instant Messenger và Windows Live Messenger (trước đây là MSN Messenger). AOL Instant Messenger là một chương trình tin nhắn tức thời và hiện diện do AOL tạo ra, sử dụng giao thức nhắn tin tức thời OSCAR độc quyền và giao thức TOC để cho phép người dùng đăng ký giao tiếp trong thời gian thực.

SmarterChild đóng vai trò giới thiệu cho việc truy cập dữ liệu nhanh và cuộc trò chuyện được cá nhân hóa thú vị hơn. Hơn nữa, khi kết hợp với các nhà mạng, chúng trở thành một kênh tiếp thị hiệu quả và miễn phí. Chúng giúp người dùng giao tiếp nhanh chóng với hệ thống mạng bằng cách hiển thị các thông tin ngắn gọn với các lựa chọn trên bàn phím điện thoại.

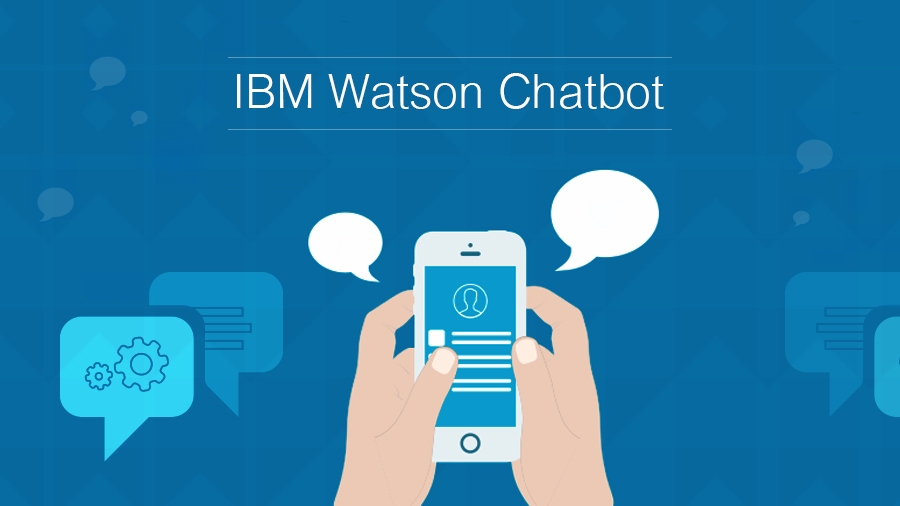


Hình 5: SmarterChild - 2001

### 2.2.7. IBM Watson - 2006

IBM Watson được tạo ra với mục tiêu vượt lên và chiến thắng các thí sinh tham dự cuộc thi Jeopardy! Với khả năng chạy hàng trăm thuật toán phân tích ngôn ngữ cùng một lúc, IBM Watson sở hữu một sự thông minh ngôn ngữ đáng ngạc nhiên. IBM thiết lập cho Watson có quyền truy cập vào cơ sở dữ liệu khổng lồ về thông tin. Watson có thể nhanh chóng truy cập 200 triệu trang dữ liệu, làm cho nó trở thành một máy trả lời câu hỏi lý tưởng (hoặc, trong trường hợp của Jeopardy, Watson trở thành máy tạo câu hỏi lý tưởng).

Rõ ràng, một hệ thống có thể nhanh chóng lấy thông tin dựa trên đầu vào đàm thoại cũng có thể cung cấp nền tảng cho việc tạo các trợ lý ảo mạnh mẽ. Hiện nay, IBM Watson phục vụ như là “bộ não” cho nhiều chatbots hoạt động trên nhiều ngành công nghiệp và lĩnh vực trên khắp thế giới.



Hình 6: IBM Watson Chatbot

### 2.2.8. Thời kỳ bùng nổ của các trợ lý ảo:

SIRI - 2010: Trợ lý dùng giọng nói và giao diện người dùng có ngôn ngữ tự nhiên để trả lời các câu hỏi, đưa ra các khuyến nghị và thực hiện hành động bằng cách chuyển các yêu cầu cho một bộ các dịch vụ Internet. Phần mềm này sẽ thích nghi với cách sử dụng ngôn ngữ, cách tìm kiếm và sở thích cá nhân của người dùng khi sử dụng. Kết quả trả lại được cá nhân hóa.

GOOGLE NOW - 2012: là một ứng dụng di động trợ giúp tìm kiếm cá nhân được tạo ra bởi Google và được hỗ trợ bởi chức năng nhận dạng giọng nói người dùng và phân biệt liệt kê ra kết quả cho họ trong đồ thị tri thức, được xây dựng trên nền tảng hệ điều hành Android 4.1 (Jelly bean).

ALEXA - 2015: hay còn gọi là Amazon Alexalà trợ lý ảo được phát triển bởi Amazon, đầu tiên nó được sử dụng cho Amazon Echo và loa thông minh Amazon Echo Dot được phát triển bởi Amazon Lab126. Nó có khả năng tương tác bằng giọng nói, chọn bài hát, lên danh sách cần làm, cài đặt báo thức, phát postcast, đọc sách, và cung cấp thông tin thời tiết, giao thông, thể thao và các thông tin hiện tại như tin tức. Alexa cũng có thể làm việc như một hệ thống điều khiển nhà tự động bằng cách điều khiển các thiết bị thông minh khác. Người sử dụng có thể nâng cấp khả năng của Alexa bằng cách cài đặt những "kỹ năng"

CORTANA - 2015: Là một trợ lý cá nhân được phát triển bởi Microsoft dành cho Windows 10 và các nền tảng các của Microsoft.

GOOGLE ASSISTANT - 2016: Trợ lý Google là một trợ lý cá nhân ảo được phát triển bởi Google cho thiết bị di động và nhà thông minh, được giới thiệu lần đầu tại hội nghị nhà phát triển của hãng vào tháng 5 năm 2016. Không giống như Google Now, Trợ lý Google có thể tham gia các cuộc trò chuyện hai chiều.

## 2.3. Phân loại Chatbot:

Chatbot có thể được phân ra thành 3 loại như sau:

* Phân loại chatbot theo dịch vụ mà chatbot hướng đến
* Phân loại chatbot theo nền tảng (platform)
* Phân loại chatbot dựa trên trải nghiệm người dùng.

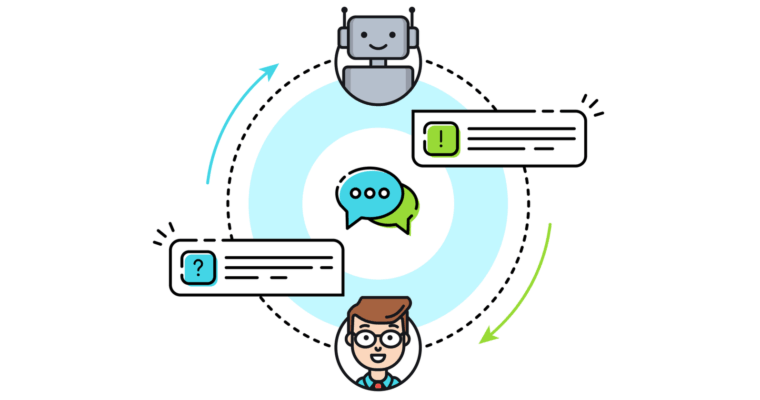
### 2.3.1. Phân loại Chatbot theo dịch vụ:

**a.. Chatbot bán hàng:**

Chatbot bán hàng có thể gắn lên bất kỳ page nào, sau quá trình thiết lập, bạn sẽ có ngay một công cụ bán hàng tự động. Ưu điểm nổi bật của loại chatbot này đó là:

* Không cần các kỹ năng công nghệ, tạo chatbot nhanh và dễ dàng
* Quản lý tập trung tin nhắn, comment của page. Phân loại dễ dàng các hội thoại, tin nhắn bằng các nhãn.
* Cập nhật thông tin khách hàng tự động, liên tục. Tạo đơn hàng tự động
* Hoạt động liên tục 24/7. Không bỏ sót comment, không bỏ sót khách hàng, không để khách hàng phải chờ lâu.

**b. Chatbot chăm sóc khách hàng:**



Hình 7: Chatbot chăm sóc khách hàng

Ở các trung tâm chăm sóc khách hàng lớn, loại chatbot này thường được sử dụng để trả lời những câu hỏi dễ dàng, theo kịch bản hoặc dữ liệu có sẵn để tiết kiệm chi phí thuê nhân sự. Ưu điểm của loại chatbot này dễ thấy như:

* Sử dụng ngôn ngữ tự nhiên, gần gũi. Hiểu được tiếng Việt
* Trả lời nhanh những câu hỏi đơn giản. Chuyển những câu hỏi khó trực tiếp đến nhân viên chăm sóc khách hàng là con người để trả lời.
* Tự động học tập những câu hỏi và những câu trả lời mới.
* Thống kê chi tiết về các loại câu hỏi, số người hỏi và tìm kiếm nội dung tương tự, …

Thường thì các doanh nghiệp sẽ áp dụng cả 2 loại chatbot này. Vừa để bán hàng, vừa để chăm sóc khách hàng, vừa giúp tiết kiệm thời gian, chi phí, nhân lực mà lại luôn luôn đáp ứng nhu cầu mọi lúc, mọi nơi. Như vậy, doanh số bán hàng cũng như trải nghiệm mua hàng và sự tin tưởng của khách hàng đối với doanh nghiệp, thương hiệu theo đó cũng tăng lên.

### 2.3.2. Phân loại Chatbot theo nền tảng:

Hiện nay, có rất nhiều ứng dụng chat, nhắn tin hỗ trợ bạn tạo các chatbot miễn phí. Những nền tảng trí thông minh nhân tạo được dùng để tạo chatbot miễn phí phổ biến nhất đó là:

**a. API.AI:**

Được hỗ trợ bởi Google và chạy trên Google Cloud Platform, cho phép bạn mở rộng tới hàng trăm triệu người dùng. Hỗ trợ text/voice messages và tương thích với hầu như tất cả các thiết bị smart phone, smart TV,… Hơn nữa, Chatbot hỗ trợ trên 20 ngôn ngữ và kết hợp với AI (trí thông minh nhân tạo và máy học) chính là những ưu điểm nổi bật của API.AI. Hỗ trợ xây dựng bởi nhiều ngôn ngữ lập trình: Android, iOS, Cordova, HTML, JavaScript, Node.js, .NET, Unity, Xamarin, C + +, Python, Ruby, PHP, Epson Moverio, Botkit và Java.

**b. WIT.AI:**

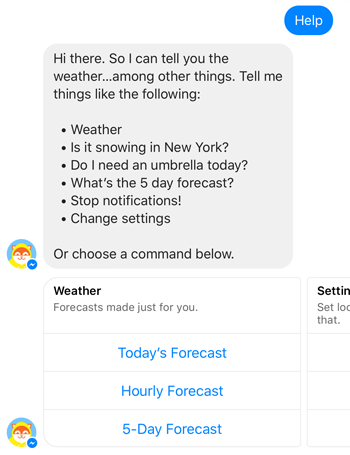
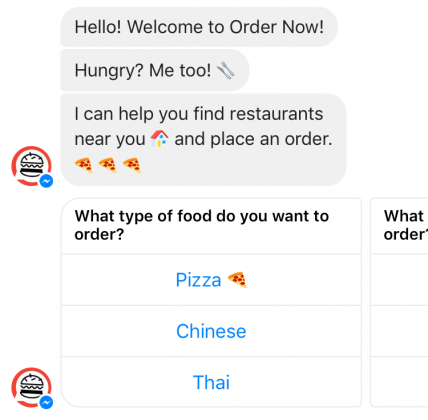
Giúp các nhà phát triển dễ dàng xây dựng các ứng dụng và thiết bị mà bạn có thể nói hoặc nhắn tin. Bạn có thể tạo các mẫu hoặc sử dụng các mẫu trong thư viện có sẵn được đóng góp bởi rất nhiều người dùng trên thế giới. Được Facebook xây dựng và ra mắt ở hội nghị F8 năm 2016, với tên gọi Messenger Platform. Chatfuel, Hara Funnel, Pandorabots và một số công cụ khác hỗ trợ xây dựng chatbot miễn phí và dễ dàng dựa trên nền tảng WIT.AI.

**c. IBM Waston:**

Được tạo ra như một hệ thống máy tính trả lời câu hỏi (QA) mà IBM xây dựng để áp dụng xử lý ngôn ngữ tự nhiên tiên tiến, truy xuất thông tin, trình bày tri thức, lý luận tự động và công nghệ học máy cho lĩnh vực trả lời câu hỏi mở. Khi được tạo ra, IBM nói rằng, “hơn 100 kỹ thuật khác nhau được sử dụng để phân tích ngôn ngữ tự nhiên, xác định nguồn, tìm và tạo giả thuyết, tìm và lấy bằng chứng, hợp nhất và xếp hạng giả thuyết.”

### 2.3.3. Phân loại chatbot dựa trên trải nghiệm người dùng:

**a. Chatbot theo kịch bản (dạng menu/button):**

Menu/Button là dạng chatbot cơ bản nhất. Trong hầu hết các trường hợp, các chatbots này là các hệ thống phân cấp cây quyết định được trình bày cho người dùng dưới dạng các nút (buttons). Tương tự như một menu với các lựa chọn hay như một chiếc điều khiển tivi, các chatbots này yêu cầu người dùng thực hiện một số lựa chọn để đào sâu hơn về phía câu trả lời cuối cùng. Sau đây là một số ví dụ về loại chatbot này:

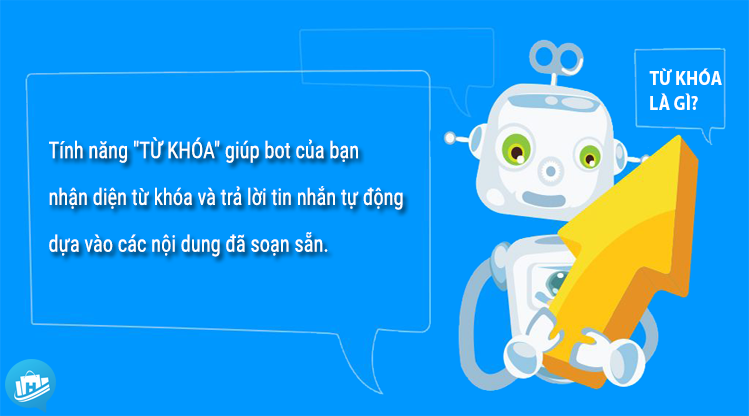
Hình 8: Chat bot theo kịch bản

**b. Chatbot nhận dạng từ khóa:**

Không giống như các chatbots dựa trên menu, các chatbots dựa trên nhận dạng từ khóa có thể lắng nghe những gì người dùng gõ và trả lời một cách thích hợp. Những chatbots 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 nhận dạng từ khóa vẫn có thể kết hợp với chatbot menu/button để đưa ra các lựa chọn cho khách hàng. Ở phần đầu video, bot dựa vào các từ khóa về địa điểm để tìm chặng bay, và thời gian bay. Sau đó hiển thị danh sách các chuyến bay để có thể lựa chọn.

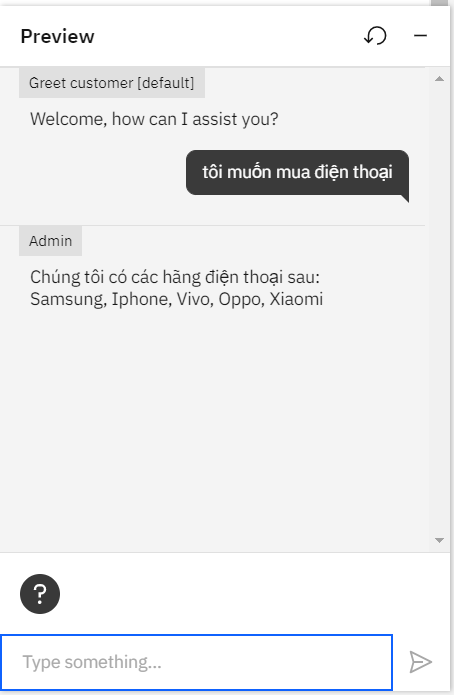
Loại chatbot nhận dạng từ khóa kết hợp menu/button đang rất phổ biến trong lĩnh vực dịch vụ. Những dịch vụ hay sử dụng loại chatbot này là dịch vụ ship đồ ăn, dịch vụ đặt vé máy bay, …



Hình 10: : Chatbot nhận diện từ khóa

**c. Chatbot trò chuyện theo ngữ cảnh:**

Là chatbot được xây dựng trên nền tảng công nghệ học máy (Machine Learning) và xử lý ngôn ngữ tự nhiên (NLP), mang đến cuộc hội thoại tự nhiên như người với khả năng đưa ra phản hồi tùy theo trường hợp, ngữ cảnh đang diễn ra. Đây là cấp độ tiên tiến nhất trong thang điểm đánh giá về khả năng đáp ứng trải nghiệm khách hàng. Bên cạnh đó, chatbot ngữ cảnh còn được biết đến với khả năng ghi nhớ sở thích và tương tác cá nhân hóa với khách hàng. Chatbot ngữ cảnh là giải pháp vượt trội, khắc phục được những điểm yếu của chatbot thế hệ cũ, đồng thời phát huy tính nhanh chóng và chính xác trong phục vụ khách hàng.



Hình 11: Chatbot trò chuyện theo ngữ cảnh

**Lợi ích của Chatbot ngữ cảnh**

**+**Trải nghiệm hộp thoại tự nhiên:

Mọi giao tiếp giữa người và người đều gắn liền với một ngữ cảnh, chủ đề nhất định. Tùy theo ngữ cảnh khác nhau, câu trả lời sẽ khác nhau. Điều này giúp Bot kiểm soát tốt các trường hợp tương tác tự nhiên của khách hàng. Chẳng hạn một khách hàng cần tư vấn cho vay có thể bắt đầu với câu hỏi “Tôi cần vay tiền mặt”, tiếp theo là những câu hỏi yêu cầu thông tin như “Lãi suất là bao nhiêu?”, “Thủ tục như thế nào?”,… Chatbot phát hiện từ khóa dường như không thể xử lý được những câu hỏi chung chung như vậy. Lãi suất có thể là lãi suất cho vay hoặc lãi suất tiết kiệm, thủ tục có thể là thủ tục mở thẻ hoặc thủ tục làm hồ sơ vay,… Không xác định được ngữ cảnh, những chatbot này không thể đưa ra phản hồi phù hợp. Được hỗ trợ bởi công nghệ NLP, chatbot ngữ cảnh sẽ xác định được chủ đề hội thoại là “tư vấn vay” thông qua câu nói “Tôi cần vay tiền mặt”, những câu hỏi tiếp theo đều được hiểu có liên quan đến chủ đề “tư vấn vay” giúp chatbot đưa ra phản hồi phù hợp cho đến khi kết thúc ngữ cảnh.

+Trải nghiệm khách hàng liền mạch:

Trong một số trường hợp, khách hàng muốn thay đổi thông tin đã cung cấp trước đó cho chatbot, họ mong muốn được tư vấn một phương án khác phù hợp với thông tin đó. Nếu không được phát triển bởi một hệ thống có NLP, chatbot từ khóa thông thường hầu như không xử lý được trường hợp này. Cuộc hội thoại có thể dừng lại vì chatbot không thể xử lý, khách hàng sẽ được yêu cầu cung cấp lại thông tin từ đầu. Điều này khiến họ cảm thấy bất tiện và họ sẽ rời đi nếu không được xử lý kịp thời. Một trải nghiệm không liền mạch là nguyên nhân khiến khách hàng thất vọng và có thể rời bỏ doanh nghiệp. Với khả năng ghi nhớ thông tin trong ngữ cảnh, chatbot ngữ cảnh hầu như giải quyết tốt vấn đề này.

## 2.4. Cấu tạo Chatbot:

Cấu tạo cơ bản của ChatBot gồm có ba phần bao gồm: cơ sở dữ liệu, lớp ứng dụng, quyền truy cập vào các API và giao diện đồ họa người dùng. Nhờ những thành phần cơ bản mà nó có thể hoạt động được.

* Cơ sở dữ liệu: Cơ sở dữ liệu lưu trữ các thông tin, dữ liệu và nội dung.

### Tầng ứng dụng: Các giao thức của tầng ứng dụng thường được dùng để trao đổi dữ liệu giữa các chương trình chạy trên máy nguồn và máy đích. Tầng này đóng vai trò như cửa sổ dành cho hoạt động xử lý các trình ứng dụng, nó biểu diễn những dịch vụ hỗ trợ trực tiếp các ứng dụng người dùng, chẳng hạn như phần mềm chuyển tin, truy nhập cơ sở dữ liệu và email.v.v…

### Giao diện lập trình ứng dụng (API): là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng.

## 2.5. Cách thức hoạt động:

ChatBot là sự kết hợp của các kịch bản có trước và tự học trong quá trình tương tác. Ta sẽ tương tác với ChatBot qua nền tảng tin nhắn. Với các câu hỏi được đặt ra,ChatBot sử dụng các hệ thống xử lý ngôn ngữ tự nhiên (Natural Language Processing) để phân tích dữ liệu sau đó chúng lựa chọn các thuật toán học máy để tạo ra các loại phản ứng khác nhau, chúng sẽ dự đoán và phản hồi chính xác nhất có thể. ChatBot sử dụng nhiều hệ thống quét các từ khoá bên trong đầu vào, sau đó bot khởi động một hành động, kéo một câu trả lời với các từ khóa phù hợp nhất và trả lời thông tin từ một cơ sở dữ liệu / API, hoặc bàn giao cho con người. Nếu tình huống đó chưa xảy ra (không có trong dữ liệu), ChatBot sẽ bỏ qua nhưng sẽ đồng thời tự học để áp dụng cho các cuộc trò chuyện về sau. Một trong các yếu tố làm nên sức mạnh của ChatBot là khả năng tự học hỏi. Càng được sử dụng, tương tác với người dùng nhiều, nền tảng ChatBot càng “thông minh”. ChatBot thông minh có khả năng tự học hỏi dựa trên các dữ liệu đưa vào mà không cần phải được lập trình cụ thể (đó được gọi là phương pháp máy học-Machine Learning). Chính điều này làm cho các nhà phát triển dễ dàng tạo các chương trình trò chuyện và tự động hoá các cuộc trò chuyện với người dùng.

## 2.6. Một số ứng dụng của Chatbot:

* Kinh doanh dịch vụ ẩm thực, ăn uống: nhà hàng, quán ăn, quán café,…
* Kinh doanh thời trang: giày dép, quần áo, phụ kiện,…
* Kinh doanh lĩnh vực làm đẹp: thẩm mỹ viện, mỹ phẩm,…
* Lĩnh vực giáo dục – đào tạo: trung tâm ngoại ngữ, trung tâm dạy kỹ năng mềm,…
* Các dịch vụ hỗ trợ: đặt vé/đặt phòng online, vận chuyển,…
* Lĩnh vực y tế: Chẩn đoán bệnh, theo dõi sức khỏe,....

## 2.7. Xu hướng phát triển:

Với tốc độ phát triển của công nghệ hiện nay, đặc biệt là cuộc cách mạng công nghệ 4.0 đang diễn ra mạnh mẽ trên toàn cầu. Chatbot - dần được phát triển hoàn thiện hơn và ngày càng thông minh. Bởi lẽ nó được ứng dụng trí thông minh nhân tạo nhằm có được những phản hồi, những câu trả lời tốt nhất, phù hợp nhất thông qua việc học hỏi những câu trả lời từ khách hàng. Do đó những câu trả lời của Bot dần trở nên thực tế và gần gũi hơn với người dùng.

Chatbot đang được ứng dụng vào nhiều lĩnh vực và sắp tới sẽ được ứng dụng mạnh mẽ hơn vào ngành y tế, giúp các y bác sĩ có được những số liệu từ bệnh nhân mà không cần phải mất quá nhiều thời gian để dò hỏi từng người. Qua đó có thể khoanh vùng được các triệu chứng thường xảy ra ở những độ tuổi khác nhau về một loại bệnh,...

Như vậy, trong tương lai không xa, Chatbot sẽ trở nên phổ biến, biến mọi thứ trở nên đơn giản và thông minh hơn, hỗ trợ đắc lực cho con người trong nhiều lĩnh vực khác nhau của cuộc sống.

## 2.8. Xây Dựng Chatbot tư vấn mua điện thoại di động cho Điện Máy Xanh

### 2.8.1. Xu hướng dữ liệu:

Dữ liệu của Chatbot này được xây dựng dựa trên các câu hỏi thường gặp khi khách hàng đến Điện máy xanh.

Nhóm chúng em đã thực hiện khảo sát dựa trên các câu hỏi trên Website của Điện máy xanh và đưa ra được những xu hướng mà khách hàng thường hay nhắm đến khi đặt câu hỏi tại Website này:

* Chính sách trả góp.
* Hỏi về chính sách đổi trả
* Hỏi về cấu hình
* Hỏi về giá
* Hỏi về bảo hành.

### 2.8.2. Lập dữ liệu cho Chatbot:

Dữ liệu của Chatbot được lập ra dựa trên xu hướng các câu hỏi của khách hàng.

Cấu trúc file dữ liệu JSON được tổ chức theo dạng:

{"tag" :"",

"patterns":[],

"responses":[],

"context\_set": ""

}

**tag**: Đặt nhãn cho tình huống. Ví dụ: cần xây dựng dữ liệu cho điện thoại Samsung thì đặt **tag** : ”samsung”

Trong **tag** bao gồm **patterns -** mẫu điều kiện - hay được hiểu trong dữ liệu ở đây là yêu cầu của khách hàng, **responses -** phản hồi - Câu trả lời của Bot khi thõa điều kiện ở phần **patterns, context\_set** **-**  bối cảnh - có thể thêm bối cảnh vào hoặc không.

Ví dụ:

{"tag": "hang",

"patterns":["Hãng nào","Hãng","haxng","hangx","thương hiệu","hang","dien thoai hãng nào","hang dien thoai","hãng điện thoại"],

"responses":["Chúng tôi hiện đang mở bán các sản phẩm từ những hãng sau:::: Iphone, Oppo, Samsung, Vivo, Huawei, Vsmart ::::: Vui lòng chọn một trong các hãng ở đây !"],

"context\_set": ""

},

Nhóm chúng em nghĩ đến các trường hợp khách hàng nhập sai chính tả, không dấu và thêm vào **patterns** nhằm tăng xác suất trả lời đúng theo ý định của khách hàng.

### 2.8.3. Dữ liệu về chính sách trả góp và chính sách đổi trả:

Như đã nói ở phần **2.8.1. Xu hướng dữ liệu** khách hàng, lượng khách hàng quan tâm đến chính sách trả góp cũng như đổi trả khi ghé thăm Website Điện máy xanh là rất nhiều - chiếm phần lớn trong tổng số khách ghé thăm.

Nên dữ liệu về hai phần này sẽ được tổ chức kỹ lưỡng hơn các phần khác. Đem đến sự thuận tiện cho khách hàng, giúp họ có được câu trả lời nhanh nhất mà không phải đợi quá nhiều thời gian để có được câu trả lời mong muốn.

# [CHƯƠNG 3: KỸ THUẬT ĐƯỢC SỬ DỤNG TRONG CHATBOT](#_28h4qwu)

## 3.1 Một số kỹ thuật

### 3.1.1. Kỹ thuật xử lý chuỗi JSON

**a. Khái niệm:**

**JSON** hay **JavaScript Object Notation** là một kiểu dữ liệu mở trong JavaScript. Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị". Về cấu trúc, nó mô tả một vật thể bằng cách bọc những vật thể con trong vật thể lớn hơn trong dấu ngoặc nhọn ({ }). JSON là một kiểu dữ liệu trung gian, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình. Định dạng JSON sử dụng các cặp *key – value* để dữ liệu sử dụng. Nó hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng.

**b.. Sử dụng file JSON:**

Theo như nhóm em tìm hiểu được, dữ liệu trong JSON file sẽ được bốc ra tùy theo các *key* được gọi.

Nhiều *key* có thể có một *value* - xác suất trả về đúng *value* sẽ tăng lên nếu có số lượng *key*  lớn và ngược lại.

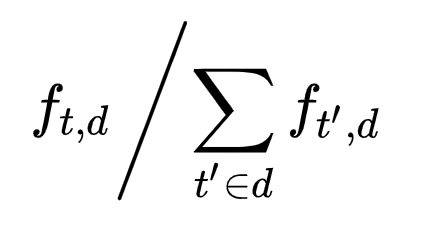
### 3.1.2. Kỹ thuật xử lý ngôn ngữ tự nhiên Bag - of - Words.

**a. Khái niệm**

Bag of Words là một thuật toán hỗ trợ xử lý ngôn ngữ tự nhiên và mục đích của BoW là phân loại text hay văn bản. Ý tưởng của BoW là phân tích và phân nhóm dựa theo "Bag of Words"(corpus). Với test data mới, tiến hành tìm ra số lần từng từ của test data xuất hiện trong "bag". Tuy nhiên BoW vẫn tồn tại khuyết điểm, nên TF-IDF là phương pháp khắc phục. Bạn có thể ứng dụng BoW + TF-IDF vào việc tìm kiếm, phân loại tài liệu, lọc mail spam xác định ý định của người dùng...

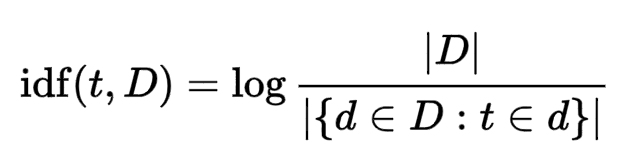
**b. Cách Bag - of - Words hoạt động**

* **Hình thành vector**
* **Đánh lại trọng số với TF-IDF:**
* Đây là một phương pháp thống kê **TF-IDF.** Giá trị TF-IDF của một từ là một con số thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.
* TF(Term Frequency) là tần số xuất hiện của 1 từ trong 1 văn bản có cách tính như sau



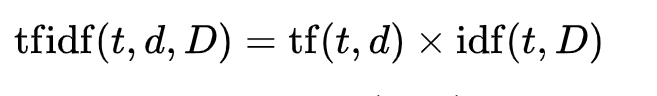
Hình 12: Đánh trọng số với TF

* **f(t,d)** - số lần xuất hiện từ t trong văn bản d.
* mẫu số là tổng số từ trong văn bản d
* Tiếp theo là IDF (Inverse Document Frequency): Tần số nghịch của 1 từ trong tập văn bản (corpus).
* Mục đích của việc tính IDF là giảm giá trị của các từ thường xuyên xuất hiện như "is", "the"... Do các từ này không mang nhiều ý nghĩa trong việc phân loại văn bản.



Hình 13: Đánh trọng số với IDF

* |D|: tổng số văn bản trong tập D \* mẫu số là số văn bản có chứa từ t . Nếu từ đó không xuất hiện ở bất cứ 1 văn bản nào trong tập thì mẫu số sẽ bằng 0 => phép chia cho không không hợp lệ, vì thế với trường hợp này thường cộng thêm 1 vào mẫu số.
* Và cuối cùng TF-IDF bằng:



Hình 14: TF-IDF

* Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

**c. Áp dụng vào ứng dụng:**

def bag\_of\_words(s, words):

bag = [0 for \_ in range(len(words))]

s\_words = nltk.word\_tokenize(s)

s\_words = [stemmer.stem(word.lower()) for word in s\_words]

for se in s\_words:

for i, w in enumerate(words):

if w == se:

bag[i] = 1

return numpy.array(bag)

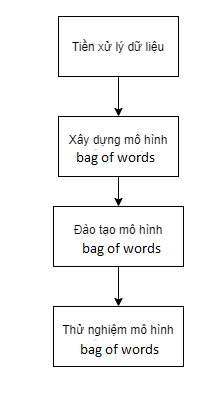
## 3.2 Kết luận chương

Chương này đã trình bày và giới thiệu những kiến thức tổng quan nhất về một hệ thống Chatbot, phân tích các ưu nhược điểm của mô hình Chatbot bán hàng hiện nay từ đó định hướng mô hình Chatbot đồ án nghiên cứu và xây dựng, đưa ra giới thiệu về chi tiết các thành phần cấu trúc và những vấn đề khi gặp phải khi xây dựng hệ thống Chatbot.

# CHƯƠNG 4: XÂY DỰNG VÀ CHẠY THỬ CHATBOT

## 4.1. Kiến trúc ứng dụng

Để tạo ra ứng dụng chatbot, chúng tôi có sử dụng xác suất, bốc dữ liệu từ data và lọc xác suất, lọc túi từ.Dưới đây là kiến trúc của ứng dụng chatbot :



Hình 15 : **Kiến trúc của ứng dụng chatbot**

Để triển khai thì nhóm đã sử dụng bộ khung chính là : Python3, Tensorflow, cùng với bộ tập dữ liệu từ file JSON.

## 4.2. Quá trình xây dựng

### 4.2.1. Xây dựng dữ liệu

Sử dụng mô hình túi từ mà tạo ra các dữ liệu cho chatbot.

Tạo một danh sách tất cả các cuộc hội thoại

Lấy câu hỏi và câu trả lời riêng biệt

Làm sạch câu hỏi và câu trả lời

Lọc ra các câu hỏi, câu trả lời quá ngắn hoặc quá dài

Mapping câu trả lời

Trích xuất câu trả lời

Đưa mỗi túi từ dưới dạng đối tượng JSON và quy cho biến [Javascript](https://vi.wikipedia.org/wiki/JavaScript) tương ứng:

BoW1 = {"Phú":1,"thích":2,"xem":2,"phim":2,"Đạt":1,"cũng":1};

BoW2 = {"Bích":1,"cũng":1,"thích":1,"xem":1,"các":1,"trận":1,"bóng":1,"đá":1};

### 4.2.2. Xây dựng mô hình BoW

Giai đoạn này là xây dựng mô hình Bag of Words để triển khai chatbot.

* **Chuẩn bị dữ liệu :**

Chuẩn bị một bộ dữ liệu training đủ để giúp hệ thống tư vấn một cách đầy đủ.

Định nghĩa cấu trúc dữ liệu theo file JSON.

{"intents": [

{"tag": "chaohoi",

"patterns": ["Xin chào", "Chào", "Hello", "Hi"],

"responses": ["Chào bạn", "Chào đằng ấy :)", "Xin chào, tôi giúp gì được cho bạn?"],

"context\_set": ""

},

{"tag": "tambiet",

"patterns": ["bye","hẹn gặp lại", "Goodbye"],

"responses": ["Hẹn gặp bạn sau", "Bye bye :))","Tạm biệt !!","..."],

"context\_set": ""

},

{"tag": "ten",

"patterns": ["Tên", "Bạn là ai?", "Ai","bạn là ai"],

"responses": ["Tên tôi là Admin nè :)", "Gọi tôi là anh Admin nha","bạn muốn tư vấn gì từ điện máy xanh thì cứ nói tôi nha"],

"context\_set": ""

},

{"tag": "hang",

"patterns":["Hãng nào","mình muốn mua điện thoại","Tôi muốn mua điện thoại","hãng","mua điện thoại","toi muon mua dien thoai","minh muon mua dien thoai","toi muốn ua đinệ thoại"],

"responses":[" Dưới đây là một số hãng bạn có thể lựa chọn :::: Iphone, Oppo, Samsung, Vivo, Xiaomi"],

"context\_set": ""

}

* **Training model**

Sử dụng TF learn - High level API của [Tensorflow](https://www.tensorflow.org/) và import các thư viện cần thiết

import nltk

nltk.download('punkt')

from nltk.stem.lancaster import LancasterStemmer

stemmer = LancasterStemmer()

import numpy

import tflearn

import tensorflow

from tensorflow.contrib.framework.python.ops import add\_arg\_scope as contrib\_add\_arg\_scope

import random

import json

import pickle

import os

from os import path

* Loading data và thực hiện trainning.

Mở file dữ liệu dưới định dạng UTF-8

with open("BoCauHoi.json", encoding="utf8") as file:

    data = json.load(file)

try:

    with open("data.pickle", "rb") as f:

        words, labels, training, output = pickle.load(f)

except:

    words = []

    labels = []

    docs\_x = []

    docs\_y = []

    for intent in data["intents"]:

        for pattern in intent["patterns"]:

            wrds = nltk.word\_tokenize(pattern)

            words.extend(wrds)

            docs\_x.append(wrds)

            docs\_y.append(intent["tag"])

        if intent["tag"] not in labels:

            labels.append(intent["tag"])

    words = [stemmer.stem(w.lower()) for w in words if w != "?"]

    words = sorted(list(set(words)))

    labels = sorted(labels)

    training = []

    output = []

    out\_empty = [0 for \_ in range(len(labels))]

    for x, doc in enumerate(docs\_x):

        bag = []

        wrds = [stemmer.stem(w) for w in doc]

        for w in words:

            if w in wrds:

                bag.append(1)

            else:

                bag.append(0)

        output\_row = out\_empty[:]

        output\_row[labels.index(docs\_y[x])] = 1

        training.append(bag)

        output.append(output\_row)

    training  = numpy.array(training)

    output = numpy.array(output)

    with open("data.pickle", "wb") as f:

        pickle.dump((words, labels, training, output), f)

tensorflow.reset\_default\_graph()

net = tflearn.input\_data(shape=[None, len(training[0])])

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, len(output[0]), activation="softmax")

net = tflearn.regression(net)

model = tflearn.DNN(net)

if os.path.exists("model.tflearn.meta"):

    model.load("model.tflearn")

else:

    model.fit(training, output, n\_epoch=1000, batch\_size=8, show\_metric=True)

    model.save("model.tflearn")

### 4.2.3. Đào tạo mô hình BoW

* Sử dụng mô hình Bag - of - Words để lọc xác suất và lấy dữ liệu phù hợp

def bag\_of\_words(s, words):

    bag = [0 for \_ in range(len(words))]

    s\_words = nltk.word\_tokenize(s)

    s\_words = [stemmer.stem(word.lower()) for word in s\_words]

    for se in s\_words:

        for i, w in enumerate(words):

            if w == se:

                bag[i] = 1

    return numpy.array(bag)

### 4.2.4. Thử nghiệm mô hình BoW

Giai đoạn này sẽ thiết lập cho chatbot hoạt động dựa trên bộ tập dữ liệu (dataset) đã huấn luyện trước đó.

Các bước thực hiện:

* Tải bộ dữ liệu đã đào tạo ở trước và chạy phiên
* Chuyển đổi các câu hỏi sang số nguyên mã hóa
* Bắt đầu cuộc trò chuyện

## 4.3. Chạy thử

- Import một số thư viện :

import nltk

nltk.download('punkt')

from nltk.stem.lancaster import LancasterStemmer

stemmer = LancasterStemmer()

import numpy

import tflearn

import tensorflow

from tensorflow.contrib.framework.python.ops import add\_arg\_scope as contrib\_add\_arg\_scope

import random

import json

import pickle

import os

from os import path

* Lấy dữ liệu và training.

print(input)

with open("BoCauHoi.json", encoding="utf8") as file:

    data = json.load(file)

try:

    with open("data.pickle", "rb") as f:

        words, labels, training, output = pickle.load(f)

except:

    words = []

    labels = []

    docs\_x = []

    docs\_y = []

    for intent in data["intents"]:

        for pattern in intent["patterns"]:

            wrds = nltk.word\_tokenize(pattern)

            words.extend(wrds)

            docs\_x.append(wrds)

            docs\_y.append(intent["tag"])

        if intent["tag"] not in labels:

            labels.append(intent["tag"])

    words = [stemmer.stem(w.lower()) for w in words if w != "?"]

    words = sorted(list(set(words)))

    labels = sorted(labels)

    training = []

    output = []

    out\_empty = [0 for \_ in range(len(labels))]

    for x, doc in enumerate(docs\_x):

        bag = []

        wrds = [stemmer.stem(w) for w in doc]

        for w in words:

            if w in wrds:

                bag.append(1)

            else:

                bag.append(0)

        output\_row = out\_empty[:]

        output\_row[labels.index(docs\_y[x])] = 1

        training.append(bag)

        output.append(output\_row)

    training  = numpy.array(training)

    output = numpy.array(output)

    with open("data.pickle", "wb") as f:

        pickle.dump((words, labels, training, output), f)

tensorflow.reset\_default\_graph()

net = tflearn.input\_data(shape=[None, len(training[0])])

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, len(output[0]), activation="softmax")

net = tflearn.regression(net)

model = tflearn.DNN(net)

"""try:

    model.load("model.tflearn")

except:

    model.fit(training, output, n\_epoch=1000, batch\_size=8, show\_metric=True)

    model.save("model.tflearn")"""

if os.path.exists("model.tflearn.meta"):

    model.load("model.tflearn")

else:

    model.fit(training, output, n\_epoch=1000, batch\_size=8, show\_metric=True)

    model.save("model.tflearn")

* Sử dụng mô hình BoW.

    def bag\_of\_words(s, words):

        bag = [0 for \_ in range(len(words))]

        s\_words = nltk.word\_tokenize(s)

        s\_words = [stemmer.stem(word.lower()) for word in s\_words]

        for se in s\_words:

            for i, w in enumerate(words):

                if w == se:

                    bag[i] = 1

        return numpy.array(bag)

-Code trò chuyện:

def chat():

    print("Cung tro chuyen voi Dien May Xanh nao :)))")

    while True:

        inp = input("Bạn: ")

        if inp.lower() == "quit":

            break

        results = model.predict([bag\_of\_words(inp, words)])[0]

        results\_index = numpy.argmax(results)

        tag = labels[results\_index]

        if results[results\_index] > 0.7:

            for tg in data["intents"]:

                if tg['tag'] == tag:

                    responses  = tg['responses']

            print("Admin dep trai: ", random.choice(responses))

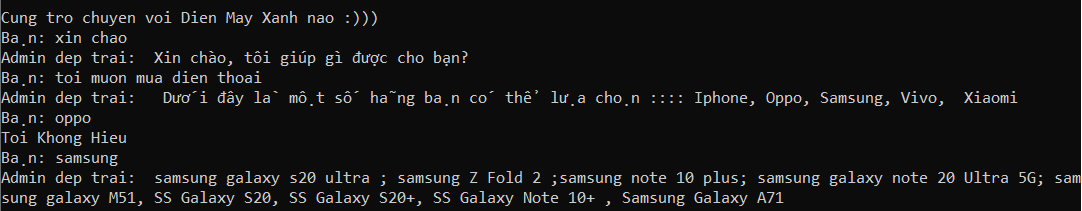
        else:

            print("Toi Khong Hieu")

chat()

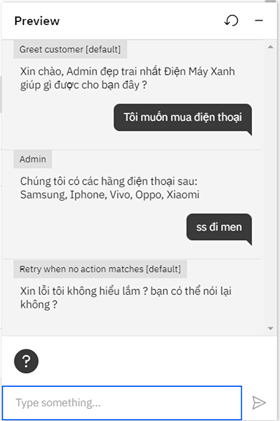
## 4.4 Kết quả đạt được

Dưới đây là hình ảnh kết quả đạt được của dự án:



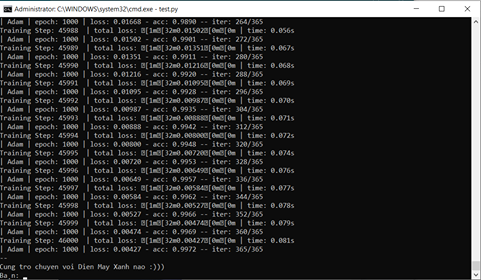
Hình 16: Trò chuyện cùng chatbot

### 4.4.1. Khi hệ thống không hiểu câu hỏi :



Hình 17: Chatbot không hiểu câu hỏi

### 4.1.2. Quá trình training



Hình 18: Quá trình training

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận:

Với sự tiện lợi và tối ưu của Deep learning cùng với xử lý ngôn ngữ tự nhiên chúng tôi đã tạo ra một ứng dụng chatbot có khả năng trò chuyện tương tác được với nhau trong một số ngữ cảnh nhất định nhưng không dựa trên một kịch bản có sẵn nào.

Tuy nhiên, khả năng tương tác với chatbot ở mức độ kém vẫn có nhưng câu trả lời sai với ngữ cảnh hoặc sai với ý người hỏi (người sử dụng). Ngoài ra khả năng huấn luyện chatbot diễn ra khá lâu. Và sự bất tiện lớn nhất là chatbot hiện tại chỉ có khả năng tương tác bằng tiếng Anh. Bên cạnh đó giao diện web còn khá đơn giản, cần nâng cấp thêm giao diện web.

Sau khi tìm hiểu và thực hiện đề tài : “Xây dựng hệ thống chatbot tư vấn mua bán điện thoại di động hỗ trợ Điện Máy Xanh” nhóm đã thực hiện được đề tài theo yêu cầu cơ bản ban đầu.

Trong quá trình thực hiện đề tài và tìm hiểu lý thuyết về chatbot, ứng dụng chatbot trong việc tư vấn mua-bán hàng, nhóm đã thu được những kết quả như sau:

* Nắm được những kiến thức cơ bản về chatbot
* Thiết lập được cuộc hội thoại tương tác giữa người và máy
* Cách sử dụng Tensorflow
* Tìm hiểu và xử lý dữ liệu của file json vào python.
* Xây dựng cụ thể các file casual intents và business intents

Về kết quả, chương trình đáp ứng được các yêu cầu đặt ra, thực hiện trò chuyện tự động với chatbot, tư vấn

## 2. Hướng phát triển:

Cải tiến thuật toán, kết hợp nhiều mô hình khác để đem lại độ chính xác cũng như thời gian đào tạo.

Tích hợp Chatbot vào môi trường Web

Đưa Chatbot ứng dụng vào Website thương mại điện tử - Tăng giá trị cho Website.

## 3. Hạn chế

Do hạn chế về mặt kiến thức và kĩ năng nên trong quá trình làm việc nhóm vẫn còn nhiều hạ chế:

* Hiện tại nhóm vẫn chưa tích hợp Chatbot này vào môi trường Web được.
* Chưa lưu được các câu trả lời của khách hàng.

# TÀI LIỆU THAM KHẢO

Sách

[1] John Seymour, Joseph O’ Connor, *NLP căn bản*, tái bản 2018.

[2] Bùi Việt Hà, *Python cơ bản*, ĐH Quốc Gia Hà Nội, 2020.

Website

[3] Phạm Đình Khánh, *Khoa học dữ liệu,*<https://phamdinhkhanh.github.io/home>

[4] *Website tin tức dành cho lập trình* <https://viblo.asia/o/viblo>, truy cập ngày 07/09/2020