**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

Khoa Công nghệ thông tin

Logo

Description automatically generated

**BÁO CÁO ĐỀ TÀI MÔN TRÍ TUỆ NHÂN TẠO**

NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG CHATBOT

GVHD: Ths. Nguyễn Thị Huệ

Nhóm thực hiện: Nhóm 2

Thành viên:

| Nguyễn Thành Dương | 2055010051 (Nhóm trưởng) |
| --- | --- |
| Dương Khánh Linh | 2055010153 |
| Phạm Trung Hiếu | 2055010093 |

Lớp 20CN3

Hà Nội, ngày 27 tháng 12 năm 2022

**MỤC LỤC**

[MỤC LỤC 2](#_heading=h.gjdgxs)x xxxxxx

[DANH MỤC HÌNH VẼ 4](#_heading=h.30j0zll)

[DANH MỤC CÁC TỪ VIẾT TẮT, TỪ TIẾNG ANH 5](#_heading=h.1fob9te)

[LỜI MỞ ĐẦU 6](#_heading=h.3znysh7)

[CHƯƠNG I: CƠ SỞ LÝ THUYẾT 7](#_heading=h.2et92p0)

[1. Tổng quan về trí tuệ nhân tạo 7](#_heading=h.tyjcwt)

[1.1. Các hướng nghiên cứu về trí tuệ nhân tạo 7](#_heading=h.3dy6vkm)

[1.2. Phân loại trí tuệ nhân tạo 8](#_heading=h.1t3h5sf)

[1.3. Mặt tích cực và hạn chế của trí tuệ nhân tạo 10](#_heading=h.17dp8vu)

[2. Ứng dụng của trí tuệ nhân tạo vào thực tế 10](#_heading=h.3rdcrjn)

[3. Lựa chọn ngôn ngữ và công nghệ cho đề tài 12](#_heading=h.26in1rg)

[3.1. Ngôn ngữ lập trình Python 12](#_heading=h.lnxbz9)

[3.2. Công nghệ và thuật toán được sử dụng trong đề tài 13](#_heading=h.35nkun2)

[CHƯƠNG II: CÔNG NGHỆ VÀ THUẬT TOÁN 15](#_heading=h.1ksv4uv)

[1. Neural Network (Mạng thần kinh nhân tạo) 15](#_heading=h.44sinio)

[1.1. Khái niệm 15](#_heading=h.2jxsxqh)

[1.2. Cách hoạt động 15](#_heading=h.3j2qqm3)

[1.3. Áp dụng vào đề tài 16](#_heading=h.4i7ojhp)

[2. Natural Language Processing (Xử lí ngôn ngữ tự nhiên) 17](#_heading=h.1ci93xb)

[2.1. Khái niệm 17](#_heading=h.3whwml4)

[2.2. Các bước xử lý 17](#_heading=h.2bn6wsx)

[2.3. Các bài toán và ứng dụng 17](#_heading=h.qsh70q)

[2.4. Áp dụng vào đề tài 18](#_heading=h.3as4poj)

[3. Sequential Search Algorithm (Thuật toán tìm kiếm tuần tự) 20](#_heading=h.147n2zr)

[3.1. Khái niệm 20](#_heading=h.3o7alnk)

[3.2. Mã giả 20](#_heading=h.23ckvvd)

[3.3. Áp dụng vào đề tài 20](#_heading=h.ihv636)

[CHƯƠNG III: SẢN PHẨM NGHIÊN CỨU VÀ KẾT LUẬN 22](#_heading=h.1hmsyys)

[1. Tổng quan 22](#_heading=h.41mghml)

[2. Sản phẩm nghiên cứu 22](#_heading=h.2grqrue)

[2.1. Cài đặt thư viện cần thiết 22](#_heading=h.vx1227)

[2.2. Huấn luyện cho Chatbot 23](#_heading=h.1v1yuxt)

[2.3. Chương trình Chatbot 24](#_heading=h.28h4qwu)

[2.4. Giao diện Chatbot 25](#_heading=h.37m2jsg)

[2.5. Thiết kế giao diện 26](#_heading=h.46r0co2)

[2.6. Các bước khởi chạy chương trình 26](#_heading=h.2lwamvv)

[3. Kết luận 28](#_heading=h.2zbgiuw)

[TÀI LIỆU THAM KHẢO 29](#_heading=h.1egqt2p)

**DANH MỤC HÌNH VẼ**

[Hình 2.1 1 Cấu trúc mạng nơ ron nhân tạo 15](#_heading=h.1y810tw)

[Hình 2.1 2 Thiết kế model Chatbot 16](#_heading=h.2xcytpi)

[Hình 2.2 1 Phân tích hình thái 18](#_heading=h.1pxezwc)

[Hình 2.2 2 Phân tích cú pháp 19](#_heading=h.49x2ik5)

[Hình 2.2 3 Phân tích và sự đoán ngữ nghĩa 19](#_heading=h.2p2csry)

[Hình 2.2 4 Áp dụng thuật toán tìm kiếm tuần tự 21](#_heading=h.32hioqz)

[Hình 3.2 1 Các thư viện cần cài đặt khi chạy chương trình 22](#_heading=h.3fwokq0)

[Hình 3.2 2 File kịch bản chứa tag, patterns, responses 23](#_heading=h.4f1mdlm)

[Hình 3.2 3 Code xử lí dữ liệu từ kịch bản 23](#_heading=h.2u6wntf)

[Hình 3.2 4 Xây dựng cấu trúc mạng nơ-ron nhân tạo đơn giản 24](#_heading=h.19c6y18)

[Hình 3.2 5 Tạo các đối tượng giảm sai sót và huấn luyện mô hình chatbot 24](#_heading=h.3tbugp1)

[Hình 3.2 6 Chạy chương trình bằng file app.py 25](#_heading=h.nmf14n)

[Hình 3.2 7 Giao diện ứng dụng Chatbot 25](#_heading=h.1mrcu09)

[Hình 3.2 8 Cài đặt mô đun punkt 26](#_heading=h.111kx3o)

[Hình 3.2 9 Kết quả cài đặt punkt thành công 27](#_heading=h.3l18frh)

[Hình 3.2 10 Kết quả sau khi huấn luyện mô hình Chatbot 27](#_heading=h.206ipza)

[Hình 3.2 11 Giao diện sau khi khởi chạy chương trình 27](#_heading=h.4k668n3)

**DANH MỤC CÁC TỪ VIẾT TẮT, TỪ TIẾNG ANH**

| **TỪ** | **Ý NGHĨA** |
| --- | --- |
| AI | (Artificial Intelligence) Trí tuệ nhân tạo |
| ANI | (Artificial Narrow Intelligence) Trí tuệ nhân tạo hẹp |
| AGI | (Artificial General Intelligence) Trí tuệ nhân tạo chung |
| ASI | (Artificial Super Intelligence) Siêu trí tuệ nhân tạo |
| LSTM | (Long Short-Term Memory) Bộ nhớ dài và ngắn hạn |
| GAN | (Generative Adversarial Network) Mạng lưới tự sinh |
| OOP | (Object Oriented Programing) Lập trình hướng đối tượng |
| NLP | (Natural Language Processing) Xử lí ngôn ngữ tự nhiên |
| NLTK | (Natural Language Toolkit) Thư viện xử lí ngôn ngữ tự nhiên |
| SNN | (Simulated Neural Network) Mạng nơ-ron mô phỏng |
| ANN | (Artificial Neural Network) Mạng nơ-ron nhân tạo |
| NN | (Neural Networks) Mạng nơ-ron |
| Neural Networks | Mạng nơ-ron |
| Reinforcement Learning | Học Tăng cường |
| Machine Learning | Máy học |
| File | Tệp tin |

**LỜI MỞ ĐẦU**

Ngày nay, trí tuệ nhân tạo đang ngày càng phát triển mạnh mẽ. Các hãng lớn như Apple, Microsoft, Google đều đưa ra các công nghệ tương tác trực tiếp với người dùng là các trợ lí ảo như Siri, Cortana, Google Assistant,… Vì vậy, với sự giúp đỡ của giảng viên hướng dẫn, Ths. Nguyễn Thị Huệ, chúng em đã nghiên cứu và xây dựng chương trình Chatbot giao tiếp với người dùng được huấn luyện từ kịch bản cho trước bằng cách áp dụng cấu trúc mạng nơ-ron nhân tạo đơn giản, Chatbot đã có thể dự đoán và trả lời được một số câu hỏi thông dụng của người dùng. Do kiến thức và hiểu biết về trí tuệ nhân tạo còn hạn chế nên trong đề tài nghiên cứu vẫn có nhiều thiếu sót, chúng em kính mong thầy cô thông cảm và đưa ra thêm nhiều nhận xét và chỉnh sửa đề đề tài chúng em có thể hoàn thiện hơn nữa. Chúng em xin chân thành cảm ơn thầy cô.

Trong quá trình nghiên cứu đề tài, chúng em phân chia công việc của các thành viên trong nhóm như sau:

***Nguyễn Thành Dương (Nhóm trưởng)***: Tìm hiểu cách hoạt động của chương trình Chatbot, chương 3 (Sản phẩm nghiên cứu), tổng hợp các chương và trình bày báo cáo trên Word.

***Phạm Trung Hiếu***: Tìm hiểu chương 1 (Tổng quan lý thuyết), và chương 3 (Sản phẩm nghiên cứu).

***Dương Khánh Linh***: Tìm hiểu chương 2 (Công nghệ và thuật toán) và làm bài thuyết trình PowerPoint.

1. **CƠ SỞ LÝ THUYẾT**
2. **Tổng quan về trí tuệ nhân tạo**

Trí tuệ nhân tạo tiếng anh là Artificial Intelligence - viết tắt là AI. Đây là một ngành thuộc lĩnh vực khoa học máy tính. Được hình thành do con người lập trình trí tuệ nhân tạo với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

1. **Các hướng nghiên cứu về trí tuệ nhân tạo**

Mục đích của trí tuệ nhân tạo là xây dựng các thực thể thông minh. Tuy nhiên, do rất khó định nghĩa thế nào là thực thể thông minh nên cũng khó thống nhất định nghĩa trí tuệ nhân tạo. Theo một tài liệu được sử dụng rộng rãi trong giảng dạy trí tuệ nhân tạo hiện nay, các định nghĩa có thể nhóm thành bốn nhóm khác nhau, theo đó, trí tuệ nhân tạo là lĩnh vực nghiên cứu việc xây dựng các hệ thống máy tính có đặc điểm sau:

***Hành động như người***

Do con người được coi là động vật có trí tuệ, nên một cách rất tự nhiên là lấy con người làm thước đo khi đánh giá mức độ thông minh của máy tính.

Theo cách định nghĩa này, trí tuệ nhân tạo nhằm tạo ra các hệ thống có hành vi hay hành động tương tự con người, đặc biệt trong những hoạt động có liên quan tới trí tuệ.

***Suy nghĩ như người***

Theo nhóm định nghĩa này, hành động thông minh chỉ đạt được nếu được dẫn dắt bởi quá trình suy nghĩ tương tự quá trình suy nghĩ của con người.

Những nghiên cứu theo hướng này dựa trên việc nghiên cứu quá trình nhận thức và tư duy của con người, từ đây mô hình hóa và tạo ra những hệ thống có mô hình nhận thức, tư duy tương tự. Việc tìm hiểu quá trình nhận thức, tư duy của người có thể thực hiện theo một số phương pháp như: *Thực nghiệm về hành vi con người khi suy nghĩ hoặc giải quyết vấn đề;* *Chụp ảnh sóng não, đo tín hiệu điện từ hoặc các tín hiệu khác của não trong quá trình thực hiện các công việc khác nhau;* *Sử dụng các phương pháp nơ ron sinh học khác như kích thích não, giải phẫu não,* *v.v.*

***Suy nghĩ hợp lý***

Một cách tiếp cận tiêu biểu của suy nghĩ hợp lý là xây dựng những hệ thống có khả năng lập luận dựa trên việc sử dụng các hệ thống hình thức như logic. Tiền thân của cách tiếp cận này có gốc rễ từ triết học Hy lạp do Aristot khởi xướng. Cơ sở chủ yếu là sử dụng logic để biểu diễn bài toán và giải quyết bằng suy diễn logic. Một số hệ thống logic cho phép biểu diễn mọi loại đối tượng và quan hệ giữa các đối tượng đó. Sau khi đã biểu diễn dưới dạng logic, có thể xây dựng chương trình để giải quyết các bài toán về suy diễn và lập luận.

Khó khăn chủ yếu của cách tiếp cận này là việc mô tả hay biểu diện bài toán dưới dạng các cấu trúc logic để có thể giải quyết được. Trên thực tế, tri thức và thông tin về bài toán thường có yếu tố không đầy đủ, không chính xác. Ngoài ra, việc suy diễn logic đòi hỏi khối lượng tính toán lớn khi sử dụng trong thực tế và rất khó để triển khai cho các bài toán thực.

***Hành động hợp lý***

Một đặc điểm quan trọng của hành động hợp lý là hành động kiểu này có thể dựa trên việc suy nghĩ (suy luận) hợp lý hoặc không. Trong một số trường hợp, để quyết định hành động thế nào, cần dựa trên việc suy luận hợp lý. Tuy nhiên, trong nhiều tình huống, việc hành động theo phản xạ, chẳng hạn khi gặp nguy hiểm, không đòi hỏi suy diễn phức tạp, nhưng lại cho kết quả tốt hơn. Các hệ thống hành động hợp lý có thể sử dụng cả hai cách tiếp cận dựa trên suy diễn và dựa trên phản xạ để đạt được kết quả tốt.

Hệ thống có khả năng hành động hợp lý có thể bao gồm suy diễn hoặc không, có thể dựa trên cách suy nghĩ giống người hoặc không, có thể bao gồm cả các kỹ thuật dùng để vượt qua phép thử Turing. Do vậy, cách tiếp cận này được coi là tổng quát và bao gồm các cách tiếp cận khác. Hiện có nhiều ý kiến coi hệ thống trí tuệ nhân tạo là các hệ thống dạng này.

1. **Phân loại trí tuệ nhân tạo**

Vì trí tuệ nhân tạo là công nghệ mô phỏng trí tuệ của con người. Vậy nên, để phân loại trí tuệ nhân tạo chúng ta dựa trên mức độ thông minh, cụ thể là về sự linh hoạt cũng như độ hiệu quả mà nó làm được. Từ đó, dựa trên sự phát triển ta có thể phân loại trí tuệ nhân tạo thành 3 loại: *Artificial Narrow Intelligence*, *Artificial General Intelligence*, *Artificial Super Intelligence*, và dựa trên sự tương đồng của AI với trí tuệ con người, ta có thể chia làm 4 loại: *Reactive Machines*, *Limited Memory*, *Theory of Mind*, *Self-Awareness.*

***Artificial Narrow Intelligence (ANI - Trí tuệ nhân tạo hẹp)***

Artificial Narrow Intelligence (Weak AI) là một loại AI được lập trình để thực hiện một hoặc một số lượng giới hạn các công việc, thay vì sở hữu toàn bộ những khả năng nhận thức như não bộ con người. Vì chỉ tập trung vào một công việc nhất định, AI có thể thay thế, thậm chí vượt xa con người ở công việc đó. Loại AI này chỉ có thể phân tích dữ liệu theo những gì nó đã được “huấn luyện” chứ không thể thực hiện những công việc mà bạn chưa hề dạy cho nó.

***Artificial General Intelligence (AGI -Trí tuệ nhân tạo chung)***

Artificial General Intelligence (Strong AI) là một AI với trí thông minh sánh ngang với con người. Nghĩa là nó có ý thức tự nhận thức để giải quyết các vấn đề, học và thể hoàn toàn tự đưa ra các quyết định mà không cần phải được huấn luyện từ trước. So với ANI, AGI chắc chắn phức tạp hơn nhiều. Và vì vậy, cho đến bây giờ, dù công nghệ phát triển không ngừng, chúng ta vẫn chưa thể tạo ra cỗ máy nào thực sự là AGI.

***Artificial Super Intelligence (ASI - Siêu trí tuệ nhân tạo)***

Siêu trí tuệ nhân tạo là loại AI có khả năng vượt xa ANI và AGI, nó có khả năng tự nhận thức đủ để vượt qua khả năng của trí não và hành vi của con người. ASI có trí nhớ tốt hơn con người, xử lý và phân tích dữ liệu nhanh hơn con người, nhờ đó có khả năng đưa ra các quyết định tốt hơn con người. Ở thời điểm hiện tại, chúng ta chưa thể tạo ra Siêu trí tuệ AI, nó chỉ xuất hiện trong những bộ phim, những cuốn sách viễn tưởng với viễn cảnh cuộc sống con người bị đe dọa bởi những bộ máy siêu thông minh này.

***Reactive Machines (Trí tuệ nhân tạo phản ứng)***

Loại AI này là AI “cổ xưa” nhất, tuân theo những nguyên lý cơ bản nhất, và đồng thời có khả năng giới hạn nhất. Nó có khả năng phản ứng với các kích thích của môi trường xung quanh, một khả năng giống với trí não con người.

Tuy nhiên, đây cũng là chức năng duy nhất của AI phản ứng. Nó không có khả năng ghi nhớ, vì vậy không thể đưa ra các quyết định đưa ra dựa trên những gì nó đã quan sát được từ trước, hay “học” từ những trải nghiệm. Điều này đồng nghĩa với việc nó không thể cải thiện bản thân qua thời gian mà chỉ có thể phản ứng trong tích tắc.

***Limited Memory (Trí tuệ nhân tạo có trí nhớ ngắn hạn)***

AI với bộ nhớ giới hạn đã “thông minh” hơn so với AI phản ứng. Bên cạnh sở hữu khả năng phản ứng với cách kích thích của môi trường, loại trí tuệ nhân tạo này có thể lưu trữ dữ liệu và các dự đoán từ trước, từ đó phân tích dữ liệu. Nhờ phân tích, nó có thể tiếp thu kiến thức có ích, từ đó đưa ra các quyết định.

Limited Memory là những cỗ máy AI có khả năng học hỏi. Chính vì vậy, nó thường được sử dụng trong các mô hình Machine Learning - Học máy như Reinforcement Learning, Long Short-Term Memory (LSTM) và GAN.

***Theory of Mind (Lý thuyết về tâm trí)***

Theory of Mind phát triển hơn so với hai loại AI trên ở chỗ nó làm được những điều trên, hay áp dụng được lý thuyết về tâm trí. Tâm trí thì rất phức tạp, cảm xúc của chúng ta thay đổi không báo trước. Để “hiểu” được tâm trí, AI còn cần phải nắm được những khái niệm tâm lý học để hiểu một người, cách một người . Nó cần học cách cảm thông với con người, cách con người tự nhìn nhận bản thân.

***Self-Awareness (Tự nhận thức)***

AI tự nhận thức là bước phát triển cuối cùng của AI. Nếu AI Lý thuyết về tâm trí có thể hiểu được tâm trí của con người thì AI tự nhận thức sẽ phát triển cảm xúc của riêng nó và nhận dạng được những cảm xúc này. Nó sẽ có nhu cầu, niềm tin, cảm xúc, phát triển ý thức, hiểu về sự tồn tại của chính nó trên thế giới này.

1. **Mặt tích cực và hạn chế của trí tuệ nhân tạo**

***Tích cực***

Mạng lưới thần kinh nhân tạo và với khả năng học tập sâu đang phát triển nhanh chóng, AI xử lý được lượng lớn dữ liệu nhanh hơn nhiều và đưa ra dự đoán chính xác hơn khả năng của con người.

Khối lượng dữ liệu khổng lồ được tạo ra hàng ngày sẽ gây khó khăn cho các nhà nghiên cứu, AI sử dụng học máy để có thể lấy những dữ liệu đó và nhanh chóng biến nó thành thông tin có thể thực hiện được.

***Hạn chế***

Tốn kém rất nhiều tài nguyên khi xử lý một lượng lớn dữ liệu mà lập trình AI yêu cầu.

Khả năng giải thích sẽ một trở ngại trong việc sử dụng AI trong các lĩnh vực hoạt động theo các yêu cầu phải tuân thủ quy định nghiêm ngặt.

Ví dụ: Các tổ chức tài chính, khi quyết định từ chối cấp tín dụng được đưa ra bởi AI, khó có thể đưa ra những giải thích rõ ràng, các lý do không cấp tín dụng cho khách hàng.

1. **Ứng dụng của trí tuệ nhân tạo vào thực tế**

***Các chương trình trò chơi***

Xây dựng chương trình có khả năng chơi những trò chơi trí tuệ là lĩnh vực có nhiều thành tựu của trí tuệ nhân tạo. Với những trò chơi tương đối đơn giản như cờ ca rô hay cờ thỏ cáo, máy tính đã thắng người từ cách đây vài thập kỷ.

***Nhận dạng tiếng nói***

Nhận dạng tiếng nói là biến đổi từ âm thanh tiếng nói thành các văn bản. Hiện người dùng công cụ tìm kiếm Google có thể đọc vào câu truy vấn thay cho việc gõ từ khóa như trước. Các điện thoại di động thông minh cũng có khả năng nhận dạng giọng nói và trả lời các câu hỏi. Ví dụ điển hình là chương trình trợ giúp Siri trên điện thoại thông minh của Apple (sử dụng công nghệ nhận dạng tiếng nói của hãng Nuance) hay hệ thống Google Now.

***Thị giác máy tính***

Mặc dù nhiều ứng dụng của thị giác máy tính vẫn chưa đạt tới độ chính xác như người, nhưng trong một số bài toán, thị giác máy tính cho độ chính xác tương đương hoặc gần với khả năng của người. Tiêu biểu phải kể đến các hệ thống nhận dạng chữ in với độ chính xác gần như tuyệt đối, hệ thống nhận dạng tròng mắt, vân tay, mặt người. Những hệ thống dạng này được sử dụng rộng rãi trong sản xuất để kiểm tra sản phẩm, trong hệ thống camera an ninh. Ứng dụng nhận dạng mặt người trên Facebook được dùng để xác định những người quen xuất hiện trong ảnh và gán nhãn tên cho người đó.

***Các thiết bị tự lái***

Các thiết bị tự lái bao gồm máy bay, ô tô, tầu thủy, thiết bị thám hiểm vũ trụ có thể tự di chuyển mà không có sự điều khiển của người (cả điều khiển trực tiếp và điều khiển từ xa). Hiện ô tô tự lái đang được một số hãng công nghệ và các tổ chức khác nghiên cứu và phát triển, trong đó có những dự án nổi tiếng như xe tự lái Tesla.

***Hệ chuyên gia***

Là các hệ thống làm việc dựa trên kinh nghiệm và tri thức của chuyên gia trong một lĩnh vực tương đối hẹp nào đó để đưa ra khuyến cáo, kết luận, chuẩn đoán một cách tự động. Các ví dụ gồm:

- MYCIN: hệ chuyên gian đầu tiên chẩn đoán bệnh về nhiễm trùng máu và cách điều trị với khả năng tương đương một bác sĩ giỏi trong lĩnh vực này.

- XCON của DEC: hỗ trợ chọn cấu hình máy tính tự động.

***Xử lý, hiểu ngôn ngữ tự nhiên***

Tiêu biểu là các hệ thống dịch tự động như hệ thống dịch của Google, các hệ thống tóm tắt nội dung văn bản tự động. Hệ thống dịch tự động của Google sử dụng các mô hình thống kê xây dựng từ các văn bản song ngữ và các văn bản đơn ngữ. Hệ thống này có khả năng dịch qua lại giữa vài chục ngôn ngữ.

***Lập kế hoạch, lập thời khóa biểu***

Kỹ thuật trí tuệ nhân tạo được sử dụng nhiều trong bài toán lập thời khóa biểu cho trường học, xí nghiệp, các bài toán lập kế hoạch khác. Một ví dụ lập kế hoạch thành công với quy mô lớn là kế hoạch đảm bảo hậu cần cho quân đội Mỹ trong chiến dịch Cơn bão sa mạc tại Iraq đã được thực hiện gần như hoàn toàn dựa trên kỹ thuật trí tuệ nhân tạo. Đây là một kế hoạch lớn, liên quan tới khoảng 50000 thiết bị vận tải và người tại cùng một thời điểm. Kế hoạch bao gồm điểm xuất phát, điểm tới, thời gian, phương tiện và người tham gia sao cho không mâu thuẫn và tối ưu theo các tiêu chí.

***Rô bốt***

Một số rô bốt được xây dựng sao cho có hình dạng tương tự con người và khả năng toàn diện như thị giác máy, giao tiếp bằng ngôn ngữ tự nhiên, khả năng lập luận nhất định, khả năng di chuyển và thực hiện các hành động như nhẩy múa. Các rô bốt này chủ yếu được tạo ra để chứng minh khả năng của kỹ thuật rô bốt thay vì hướng vào ứng dụng cụ thể. Trong số này có thể kể tới rô bốt Asimo, rô bốt Nao.

1. **Lựa chọn ngôn ngữ và công nghệ cho đề tài**
2. **Ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình thông dịch, hướng đối tượng, ngôn ngữ lập trình cấp cao được giải thích với ngữ nghĩa động. Python với triết lý thiết kế của nó rất thuận tiện cho việc đọc hiểu code, đơn giản và rõ ràng được thiết kế bởi Guido van Rossum.

Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Ngôn ngữ này có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Do đó nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Cấu trúc của nó còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, như nhận định của chính Guido van Rossum trong một bài phỏng vấn ông.

***Ngôn ngữ lập trình đơn giản, dễ học – dễ học***

Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

***Miễn phí, mã nguồn mở***

Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

***Khả năng di động linh hoạt***

Giả sử bạn giả sử bạn đã viết mã Python cho máy Windows của mình. Bây giờ, nếu bạn muốn chạy nó trên máy Mac, bạn không cần phải thay đổi nó như cũ. Nói cách khác, bạn có thể lấy một mã và chạy nó trên bất kỳ máy nào, không cần phải viết mã khác nhau cho các máy khác nhau. Điều này làm cho Python trở thành một ngôn ngữ di động. Tuy nhiên, bạn phải tránh mọi tính năng phụ thuộc hệ thống trong trường hợp này. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

***Khả năng mở rộng và có thể nhúng***

Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

***Ngôn ngữ thông dịch cấp cao***

Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,… Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Trong nội bộ, mã nguồn của nó được chuyển đổi thành một hình thức ngay lập tức được gọi là bytecode. Vì vậy, tất cả những gì bạn cần làm là chạy đoạn code Python của bạn mà không phải lo lắng về việc liên kết với các thư viện và những thứ khác.

***Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến***

Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code.

***Hướng đối tượng***

Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng. Python hỗ trợ cả lập trình hướng đối tượng, một trong những tính năng chính của nó. Nó cũng hỗ trợ nhiều kế thừa, không giống như Java.

1. **Công nghệ và thuật toán được sử dụng trong đề tài**

***Neural Network (Mạng thần kinh nhân tạo)***

Nơ-ron nhân tạo có các mô đun phần mềm, được gọi là nút. Mạng nơ-ron nhân tạo là chương trình phần mềm hoặc thuật toán mà về cơ bản, sử dụng hệ thống máy tính để giải quyết phép toán.

Mạng thần kinh nhân tạo có cấu tạo nhiều lớp nút, bao gồm lớp đầu vào, một hoặc nhiều lớp ẩn và lớp đầu ra. Mỗi nút, hoặc nơ-ron nhân tạo, kết nối với các nút khác có trọng số cùng ngưỡng liên quan. Chúng phối hợp, gửi tín hiệu đến nhau để cùng giải quyết vấn đề.

***Natural Language Processing (Xử lí ngôn ngữ tự nhiên)***

Natural Language Processing (NLP) là một nhánh của ngôn ngữ học, khoa học máy tính và trí tuệ nhân tạo liên quan đến sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người (Ví dụ: Tiếng Anh, Tiếng Việt,...), giọng nói hoặc văn bản.

Những thử thách trong lĩnh vực xử lý ngôn ngữ tự nhiên là: Nhận diện giọng nói, hiểu được ngôn ngữ tự nhiên của con người và mô phỏng lại ngôn ngữ tự nhiên.

***Sequential Search Algorithm (Thuật toán tìm kiếm tuần tự)***

Thuật toán tìm kiếm tuyến tính (Linear Search) hay còn gọi là thuật toán tìm kiếm tuần tự (Sequential Search) là một phương pháp tìm kiếm một phần tử cho trước trong một danh sách bằng cách duyệt lần lượt từng phần tử của danh sách đó cho đến lúc tìm thấy giá trị mong muốn hay đã duyệt qua toàn bộ danh sách.

Tìm kiếm tuyến tính là một giải thuật rất đơn giản khi hiện thực. Giải thuật này tỏ ra khá hiệu quả khi cần tìm kiếm trên một danh sách đủ nhỏ hoặc một danh sách chưa sắp thứ tự đơn giản. Trong trường hợp cần tìm kiếm nhiều lần, dữ liệu thường được xử lý một lần trước khi tìm kiếm: có thể được sắp xếp theo thứ tự, hoặc được xây dựng theo một cấu trúc dữ liệu đặc trưng cho giải thuật hiệu quả hơn,…

1. **CÔNG NGHỆ VÀ THUẬT TOÁN**
2. **Neural Network (Mạng thần kinh nhân tạo)**
3. **Khái niệm**

Mạng nơ-ron (NN - Neural Networks), còn được gọi là mạng nơ-ron nhân tạo (ANN - Artificial Neural Network) hoặc mạng nơ-ron mô phỏng (SNN - Simulated Neural Network), là một tập hợp con của học máy, trung tâm của thuật toán học sâu. Tên và cấu trúc của chúng được lấy cảm hứng từ bộ não con người, bắt chước cách tế bào thần kinh sinh học truyền tín hiệu cho nhau.

1. **Cách hoạt động**

Một mạng nơ ron nhân tạo cơ bản thường có cấu trúc 3 lớp như sau:

***Lớp đầu vào:*** Thông tin cần xử lý được đưa vào mạng nơ-ron nhân tạo qua lớp đầu vào. Nút đầu vào tiếp nhận, phân loại, phân tích dữ liệu và sau đó chuyển dữ liệu sang lớp tiếp theo.

***Lớp ẩn:*** Dữ liệu chuyển từ lớp đầu vào sang lớp ẩn, hoặc từ lớp ẩn này sang lớp ẩn khác. Mạng nơ-ron nhân tạo có thể có một hoặc nhiều lớp ẩn. Mỗi lớp ẩn phân tích dữ liệu đầu ra từ lớp trước, xử lý dữ liệu đó sâu hơn và chuyển dữ liệu sang lớp tiếp theo.

***Lớp đầu ra:*** Lớp đầu ra trả kết quả cuối cùng của tất cả dữ liệu được xử lý trước đó bởi mạng nơ-ron nhân tạo. Lớp này có thể có một hoặc nhiều nút.

Chart, diagram, bubble chart

Description automatically generated

*Hình 2.1 1 Cấu trúc mạng nơ ron nhân tạo*

Coi mỗi nút riêng lẻ là mô hình hồi quy tuyến tính của chính nó, bao gồm dữ liệu đầu vào, trọng số, độ lệch (hoặc ngưỡng) và đầu ra.

Khi dữ liệu ở lớp đầu vào được xác định, trọng số sẽ được gán. Các trọng số này giúp xác định tầm quan trọng của bất kỳ biến cụ thể nào, với những biến lớn hơn đóng góp đáng kể hơn vào đầu ra so với những đầu vào khác.

Tất cả đầu vào được nhân với trọng số tương ứng của chúng và tính tổng. Nếu đầu ra đó vượt quá một ngưỡng nhất định, nó sẽ “kích hoạt” nút, chuyển dữ liệu sang lớp tiếp theo trong mạng. Như vậy, đầu ra của một nút trở thành đầu vào của nút tiếp theo.

Hầu hết mạng nơ-ron nhân tạo đều là mạng nơ-ron truyền thẳng (feedforward), chảy theo một hướng từ đầu vào đến đầu ra. Tuy nhiên, chúng cũng có thể được đào tạo để truyền ngược; nghĩa là di chuyển theo hướng từ đầu ra quay trở lại đầu vào.

1. **Áp dụng vào đề tài**

Thiết kế mạng neural đơn giản cho model chatbot để thực hiện việc training:

* Model tạo bởi lớp đối tượng Sequential của thư viện Keras
* Trong mạng:
  + Lớp đầu vào: có số nút bằng độ dài của mảng train\_X[0] hay độ dài của mảng các từ xuất hiện trong kịch bản, các nút này sử dụng các hàm phi tuyến tính relu để tính toán và tạo ra các trường hợp sinh ra bởi đầu vào của nút đó.
  + Lớp ẩn: có 2 lớp, 1 lớp gồm 120 nút mạng, các nút này sử dụng các hàm phi tuyến tính relu, và một lớp gồm 60 nút mạng, các nút này sử dụng các hàm softmax để tính toán đầu ra cho lớp sau.
  + Lớp đầu ra: có số nút bằng số lượng ‘ý nghĩa’ mà kịch bản có (độ dài của mạng train\_y[0]).
* Các đối tượng Dense là tạo các lớp cho mạng nơ-ron, lớp Dropout dùng để giảm bớt các đường kết nối có thể bị thừa khi kết nối giữa 2 lớp

Text

Description automatically generated

*Hình 2.1 2 Thiết kế model Chatbot*

1. **Natural Language Processing (Xử lí ngôn ngữ tự nhiên)**
2. **Khái niệm**

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) 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.

1. **Các bước xử lý**

***Phân tích hình thái*** - Trong bước này từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh) và nhiều ngôn ngữ khác, các từ được phân tách với nhau bằng dấu cách. Tuy nhiên trong [tiếng Việt](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Vi%E1%BB%87t), dấu cách được dùng để phân tách các tiếng (âm tiết) chứ không phải từ. Cùng với các ngôn ngữ như [tiếng Trung](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Trung_Qu%E1%BB%91c), [tiếng Hàn](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Tri%E1%BB%81u_Ti%C3%AAn), [tiếng Nhật](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Nh%E1%BA%ADt), [phân tách từ](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_t%C3%A1ch_t%E1%BB%AB&action=edit&redlink=1) trong tiếng Việt là một công việc không hề đơn giản.

***Phân tích cú pháp*** - Dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.

***Phân tích ngữ nghĩa*** - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.

***Tích hợp văn bản*** - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau.

***Phân tích thực nghĩa*** - Cấu trúc thể hiện điều được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.

Tuy nhiên, ranh giới giữa 5 bước xử lý này cũng rất mong manh. Chúng có thể được tiến hành từng bước một, hoặc tiến hành cùng lúc - tùy thuộc vào giải thuật và ngữ cảnh cụ thể.

1. **Các bài toán và ứng dụng**

[***Nhận dạng chữ viết***](https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADn_d%E1%BA%A1ng_ch%E1%BB%AF_vi%E1%BA%BFt&action=edit&redlink=1)***:*** Chương trình nhận dạng chữ viết in có thể chuyển hàng ngàn đầu sách trong thư viện thành văn bản điện tử trong thời gian ngắn. 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***](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_ti%E1%BA%BFng_n%C3%B3i)***:*** Nhận dạng tiếng nói rồi chuyển chúng thành văn bản tương ứng. Đây cũng là bước đầu tiên cần phải thực hiện trong ước mơ thực hiện giao tiếp giữa con người với robot. Nhận dạng tiếng nói có khả năng trợ giúp người khiếm thị rất nhiều.

[***Tổng hợp tiếng nói***](https://vi.wikipedia.org/wiki/T%E1%BB%95ng_h%E1%BB%A3p_gi%E1%BB%8Dng_n%C3%B3i)***:*** Từ một văn bản tự động tổng hợp thành tiếng nói. Giống như nhận dạng tiếng nói, tổng hợp tiếng nói là sự trợ giúp tốt cho [người khiếm thị](https://vi.wikipedia.org/wiki/Ng%C6%B0%E1%BB%9Di_khi%E1%BA%BFm_th%E1%BB%8B), nhưng ngược lại nó là bước cuối cùng trong giao tiếp giữa [robot](https://vi.wikipedia.org/wiki/Robot) với người.

[***Dịch tự động***](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_t%E1%BB%B1_%C4%91%E1%BB%99ng)***(Machine Translate):*** Như tên gọi đây là chương trình dịch tự động từ ngôn ngữ này sang ngôn ngữ khác. Một ví dụ nổi tiếng là Google Translate của Google.

[***Tìm kiếm thông tin***](https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_th%C3%B4ng_tin)***(Information Retrieval):*** Đặt câu hỏi và chương trình tự tìm ra nội dung phù hợp nhất. Mới đây cộng đồng mạng đang xôn xao về trang [ChatGPT](https://vi.wikipedia.org/wiki/Wolfram_Alpha), được cho là có khả năng hiểu ngôn ngữ tự nhiên của con người và đưa ra câu trả lời chính xác. Lĩnh vực này hứa hẹn tạo ra bước nhảy trong cách thức tiếp nhận tri thức của cả cộng đồng.

[***Tóm tắt văn bản***](https://vi.wikipedia.org/w/index.php?title=T%C3%B3m_t%E1%BA%AFt_v%C4%83n_b%E1%BA%A3n&action=edit&redlink=1)***:*** Từ một văn bản dài tóm tắt thành một văn bản ngắn hơn theo mong muốn nhưng vẫn chứa những nội dung thiết yếu nhất.

[***Khai phá dữ liệu***](https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u)***(Data Mining) và***[***phát hiện tri thức***](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A1t_hi%E1%BB%87n_tri_th%E1%BB%A9c&action=edit&redlink=1)***:*** Từ rất nhiều tài liệu khác nhau phát hiện ra tri thức mới. Thực tế để làm được điều này rất khó, nó gần như là mô phỏng quá trình [học tập](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_t%E1%BA%ADp), khám phá khoa học của con người, đây là lĩnh vực đang trong giai đoạn đầu phát triển.

1. **Áp dụng vào đề tài**

***Đầu vào:*** Câu hỏi được lấy từ ứng dụng chatbott kiểu chuỗi kí tự

***Phân tích hình thái:*** Tách các từ khỏi câu, lọc các dấu câu không cần thiết bằng thư viện nltk.

Text

Description automatically generated

*Hình 2.2 1 Phân tích hình thái*

***Phân tích cú pháp:*** Ánh xạ mảng các từ sang một mảng nhị phân tương ứng với câu đó trong tập hợp các từ của kịch bản.

Text

Description automatically generated

*Hình 2.2 2 Phân tích cú pháp*

***Phân tích ngữ nghĩa, tích hợp văn bản, Dự đoán thực nghĩa:*** Xử lí dữ liệu số ở phần phân tích cú pháp và đối chiếu với dữ liệu đã được training, tích hợp và dự đoán ý nghĩa của dữ liệu đó.

Text

Description automatically generated

*Hình 2.2 3 Phân tích và sự đoán ngữ nghĩa*

***Kết quả:*** Trả về ý nghĩa, câu trả lời theo kiểu chuỗi kí tự

* ***Nhận dạng tiếng nói:*** Thư viện SpeechRecognition của Python để xử lí phần nhận diện giọng nói đầu vào cho chatbot. Chương trình sử dụng thuật toán nhận diện giọng nói của google để xử lí âm thanh đầu vào và chuyển đổi âm thanh đó sang chuỗi kí tự.
* ***Tổng hợp tiếng nói:*** Sau khi nhận được chuỗi kí tự đầu ra của chatbot (câu trả lời), chương trình sử dụng thư viên pyttsx3 để chuyển thể chuỗi đó sang dạng âm thanh và phát ra.

1. **Sequential Search Algorithm (Thuật toán tìm kiếm tuần tự)**
2. **Khái niệm**

Thuật toán tìm kiếm tuyến tính (Linear Search) hay còn gọi là thuật toán tìm kiếm tuần tự (Sequential Search) là một phương pháp tìm kiếm một phần tử cho trước trong một danh sách bằng cách duyệt lần lượt từng phần tử của danh sách đó cho đến lúc tìm thấy giá trị mong muốn hay đã duyệt qua toàn bộ danh sách.

Tìm kiếm tuyến tính là một giải thuật rất đơn giản khi hiện thực. Giải thuật này tỏ ra khá hiệu quả khi cần tìm kiếm trên một danh sách đủ nhỏ hoặc một danh sách chưa sắp thứ tự đơn giản. Trong trường hợp cần tìm kiếm nhiều lần, dữ liệu thường được xử lý một lần trước khi tìm kiếm: có thể được sắp xếp theo thứ tự, hoặc được xây dựng theo một cấu trúc dữ liệu đặc trưng cho giải thuật hiệu quả hơn,…

1. **Mã giả**

Đây là phiên bản hay gặp nhất của giải thuật này, kết quả trả về sẽ là vị trí của phần tử cần tìm hoặc một giá trị Δ thể hiện việc không tìm thấy phần tử trong danh sách đó.

1. For each item in the list:

1. if that item has the desired value,

1. stop the search and return the item's location.

2. Return 'Δ'

Nếu danh sách được lưu trữ dưới dạng [mảng](https://vi.wikipedia.org/wiki/M%E1%BA%A3ng), vị trí của phần tử cần tìm có thể là chỉ số của nó trong mảng, còn giá trị Δ có thể là chỉ số nằm trước phần tử đầu tiên (0 hoặc -1 tùy vào danh sách).

Nếu danh sách là một [danh sách liên kết](https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_li%C3%AAn_k%E1%BA%BFt), vị trí của phần tử được trả về có thể nằm dưới dạng [địa chỉ](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Ba_ch%E1%BB%89) của nó, còn giá trị Δ có thể là giá trị [null](https://vi.wikipedia.org/wiki/Null).

1. **Áp dụng vào đề tài**

***Giải thích:***

* Đầu vào: tag (ý nghĩa, loại câu có xác suất cao nhất), list (danh sách kịch bản), probability (Xác suất cao nhất).
* Đầu ra: Nếu xác suất trả lời đúng cao hơn 75%, chạy lần lượt qua danh sách list ,so sánh khóa ‘tag’ của phần tử đang xét với chuỗi ý nghĩa đầu vào. Nếu tìm được kết quả (2 giá trị khớp nhau) thì hàm thuật toán trả về câu trả lời ngẫu nhiên trong khóa ‘responses’, nếu không trả về chuỗi ‘Tôi không biết’.

***Mã giả trong đề tài:***

SequentialSearch(tag, list, probability):

if probability >= 75%:

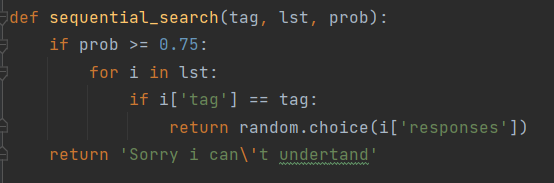
for i in list:

if i[‘tag’] = tag:

return ngẫu nhiên i[‘responses’]

return ‘Tôi không biết’

***Code:***



*Hình 2.2 4 Áp dụng thuật toán tìm kiếm tuần tự*

1. **SẢN PHẨM NGHIÊN CỨU VÀ KẾT LUẬN**
2. **Tổng quan**

Trong đề tài nghiên cứu này, chúng em sử dụng cấu trúc Neural Network đơn giản để huấn luyện 1 mô hình Chatbot có thể học theo kịch bản có sẵn và dự đoán ý nghĩa, trả lời câu hỏi được đặt ra theo lựa chọn có xác suất cao nhất.

1. **Sản phẩm nghiên cứu**
2. **Cài đặt thư viện cần thiết**

Đầu tiên, trong đề tài chúng em sử dụng ngôn ngữ python phiên bản 3.10 do 3.11 chưa hỗ trợ các thư viện như *Tensorflow*,... Vậy nên để có thể thuận tiện chạy chương trình trên các thiết bị khác, người dùng cần cài đặt python các phiên bản từ 3.10 đến 3.7 là tốt nhất. Và kèm theo đó là các thư viện cần cài đặt để sử dụng cho chương trình, chúng em đã tổng hợp vào file ***requirements.txt***¸ như vậy thay vì cài đặt từng thư viện người dùng chỉ cần tải các thư viện được đọc từ file ***requirements.txt***.

Graphical user interface, text, application

Description automatically generated

*Hình 3.2 1 Các thư viện cần cài đặt khi chạy chương trình*

Để cài đặt các thư viện, người dùng cần mở terminal hoặc command prompt tại địa chỉ lưu file ***requirements.tx*** lên và gõ các lệnh như sau:

***Window:***

pip install -r requirements.txt

hoặc:

pip install -r requirements.txt --user

***MacOS:***

pip3 install -r requirements.txt

***Linux:***

sudo pip3 install -r requirements.txt

1. **Huấn luyện cho Chatbot**

Phần này là giai đoạn xử lí dữ liệu trong kịch bản được tạo sẵn và tạo, áp dụng cấu trúc mạng nơ ron nhân tạo (Neural Network) vào quá trình huấn luyện chatbot. Khi chạy file ***training.py*** thì quá trình này sẽ được thực hiện. Và đây là bước đầu tiên để thực hiện chương trình, vì bước này sẽ xuất ra một mô hình trí tuệ nhân tạo được sử dụng ở phần sau (Chương trình Chatbot).

***Kịch bản***

Kịch bản sẵn có của chúng em là 1 tập các câu hỏi theo ý nghĩa nhất định và các câu trả lời của câu hỏi đó. Với khóa ‘tag’ là các ý nghĩa của các câu hỏi, khóa ‘patterns’ là các câu hỏi có thể xảy ra, và khóa ‘responses’ chứa các câu trả lời khả thi dành cho câu hỏi đó.

A picture containing timeline

Description automatically generated

*Hình 3.2 2 File kịch bản chứa tag, patterns, responses*

***Xử lí dữ liệu từ kịch bản***

Bước xử lí dữ liệu kịch bản chúng em phân tách các câu từ có trong kịch bản ra và tập hợp, biểu diễn các từ đó dưới dạng số trong 1 mảng các từ của kịch bản, đồng thời, chúng em chia ra theo các ý nghĩa của kịch bản (ý nghĩa của các câu hỏi cũng được số hóa giúp cho máy tính có thể tính toán tốt hơn).

Text

Description automatically generated

*Hình 3.2 3 Code xử lí dữ liệu từ kịch bản*

***Tạo cấu trúc mạng nơ-ron nhân tạo***

Các lớp của mạng nơ-ron của chúng em được thiết kế dựa trên các hàm tính toán phi tuyến tính,... Gồm có: 1 lớp đầu vào có số nút bằng số từ xuất hiện trong kịch bản, 2 lớp ẩn một lớp 120 nút liên kết với một lớp 60 nút và 1 lớp đầu ra có số nút bằng số loại câu hỏi trong kịch bản.

Text

Description automatically generated

*Hình 3.2 4 Xây dựng cấu trúc mạng nơ-ron nhân tạo đơn giản*

***Giảm thiếu sai số trong công thức của các lớp, huấn luyện mô hình***

Ban đầu mô hình Chatbot sẽ lấy 1 giá trị ngẫu nhiên để xử lí và tìm ra công thức chính xác nhất đối với kịch bản. Để có thể tìm được giá trị tham số đúng của hàm số tính toán ta cần phải thay đổi các tham số ban đầu theo hàm sgd. Sau khi quá trình training kết thúc chúng ta sẽ xuất mô hình đó để nó làm cơ sở cho Chatbot.

Text

Description automatically generated

*Hình 3.2 5 Tạo các đối tượng giảm sai sót và huấn luyện mô hình chatbot*

1. **Chương trình Chatbot**

Phần này được viết trong file ***chatbot.py*** gồm các hàm xử lí câu từ đầu vào và cuối cùng là để có thể xuất ra được câu trả lời. Các hàm trên được áp dụng theo quá trình xử lí ngôn ngữ tự nhiên gồm có: *phân tích hình thái*, *phân tích cú pháp*, *phân tích ngữ nghĩa*, *tích hợp văn bản*, *phân tích thực nghĩa*.

Trong đó:

***Hàm clean\_up\_sentence***: Tách từ và xóa các dấu câu tạo thành 1 mảng các từ và trả về.

***Hàm bag\_of\_words***: Chuyển đổi các từ sang mảng số để Chatbot xử lí, dự đoán ngữ nghĩa của câu hỏi.

***Hàm predict\_class***: Là hàm tính toán và trả về thống kê xác suất của câu hỏi này đang có ý nghĩa là gì.

***Hàm sequential\_search***: Là hàm tìm kiếm câu trả lời trong file kịch bản được tạo sẵn nếu xác suất cao nhất tìm được lớn hơn hoặc bằng 75%.

***Hàm get\_response***: Đây là hàm được gọi khi khởi chạy chương trình, nó đảm nhận vai trò tổng hợp kết quả các hàm trên để xuất ra câu trả lời cuối cùng, chính xác nhất mà nó có thể tính toán được.

Sau cùng khi các dữ liệu đã được chuẩn bị xog chương trình sẽ báo ‘BOT is ready’ trên terminal.

Text

Description automatically generated

*Hình 3.2 6 Chạy chương trình bằng file app.py*

Các hàm trên được sử dụng khi chương trình ***app.py*** chạy và gọi hàm ***get\_response***.

1. **Giao diện Chatbot**

Graphical user interface, text, application

Description automatically generated

*Hình 3.2 7 Giao diện ứng dụng Chatbot*

***Giao diện:*** Chatbot được thiết kế dựa trên thư viên Tkinter, chia làm 3 thành phần Canvas: *Talk*, *Chat*, *Inputarea*. Phần *Talk* gồm nút mở tính năng nhận diện giọng nói để nói chuyện với Chatbot. Phần *Chat* hiển thị nội dung cuộc trò chuyện giữa người với bot. Phần *Inputarea* gồm 2 thành phần là nhập tin nhắn và gửi tin nhắn.

***Nhận diện giọng nói:*** Chúng em sử dụng thư viện *SpeechRecognition* của Python để nhận giọng nói và xử lí. Dữ liệu đầu vào sẽ được nghe bằng mic và sử dụng hàm ***recognize\_google*** để chuyển đổi dữ liệu đó sang dạng chuỗi kí tự để Chatbot có thể xử lí được.

***Tổng hợp giọng nói:*** Chúng em sử dụng thư viện *pyttsx3* để xử lí, tổng hợp chuỗi ký tự đầu ra (câu trả lời của chatbot) và chuyển đổi nó thành chuỗi âm thanh phát ra loa máy tính.

1. **Thiết kế giao diện**

Chúng em tạo một đối tượng cửa sổ chat bot để tạo giao diện ứng dụng.

***Hàm \_set\_up\_win\_app:*** dùng để thiết lập và cái đặt các thành phần trong ứng dụng như: mic, nút gửi, khung chat,…

***Hàm run:*** dùng để chạy chương trình, mở cửa sổ lên màn hình máy tính.

***Hàm \_sendmsg:*** được gọi khi ấn Enter, hoặc nút gửi tin.

***Hàm \_getmsg:*** nhận tham số msg(là tin nhắn của người dùng) và in nó ra khung chat, và xóa chữ trong khung nhập tin.

***Hàm \_printresponses:*** nhận vào tin nhắn của người dùng và thực hiện lấy câu trả lời trả về từ Chatbot.

***Hàm \_onmic:*** Bật tính năng nhận diện giọng nói của chương trình để nhận tin nhắn âm thanh.

1. **Các bước khởi chạy chương trình**

Đầu tiên, chúng ta chạy file ***training.py*** để xử lí dữ liệu kịch bản trước và huấn luyện mô hình cho Chatbot rồi xuất file chứa mô hình đó. Ở bước này, có một hoạt động một lần là cài đặt ***mô-đun PUNKT*** của thư viện ***nltk***.

Text

Description automatically generated

*Hình 3.2 8 Cài đặt mô đun punkt*

Trong *NLTK*, *PUNKT* là một mô hình có thể đào tạo không giám sát, có nghĩa là nó có thể được đào tạo trên dữ liệu chưa được gắn nhãn (Dữ liệu chưa được gắn thẻ với thông tin xác định đặc điểm, thuộc tính hoặc danh mục của nó được gọi là dữ liệu chưa được gắn nhãn).

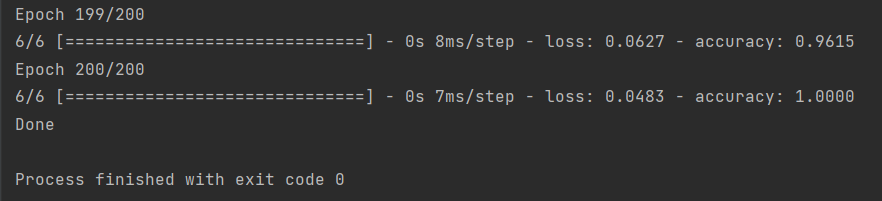
Sau lần chạy đầu tiên, cài đặt mô-đun trên máy, chúng ta có thể đóng hoặc xóa dòng code đó lại để các lần chạy sau không phải thực hiện thêm và rút ngắn thời gian training.

Text

Description automatically generated

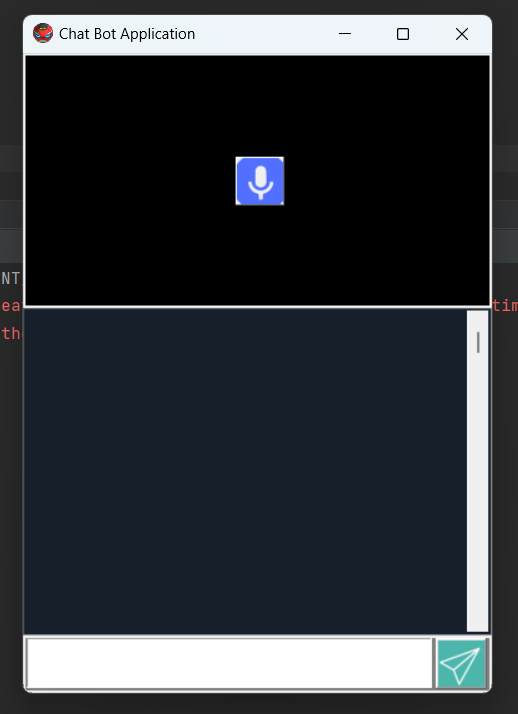
*Hình 3.2 9 Kết quả cài đặt punkt thành công*

Huấn luyện mô hình thành công:



*Hình 3.2 10 Kết quả sau khi huấn luyện mô hình Chatbot*

Sau khi huấn luyện mô hình, chúng ta xuất được mô hình đó ra 1 file để sử dụng cho phần xử lí của chatbot trong file ***chatbot.py***. File ***chatbot.****py* không cần khởi chạy mà sẽ được gọi khi chạy giao diện của Chatbot. Để chạy giao diện Chatbot, chúng ta import hàm ***get\_response*** của file ***chatbot.py*** vào file ***app.py*** và chạy file. Kết quả chúng ta sẽ được như sau:



*Hình 3.2 11 Giao diện sau khi khởi chạy chương trình*

1. **Kết luận**

Trong quá trình tìm hiểu và xây dựng chương trình Chatbot trên, nhóm chúng em đã đạt được những kết quả như sau:

***Kết quả đạt được:***

* Thiết kế được mạng Neural Network đơn giản để huấn luyện chatbot tự học từ kịch bản cho trước.
* Chatbot sau khi huấn luyện có thể dự đoán, và sử dụng thuật toán tìm kiếm tuần tự để tìm kiếm câu trả lời có xác suất cao nhất.
* Xuất ra được model cho chương trình.
* Xây dựng được giao diện chatbot có thể tương tác qua tin nhắn hoặc giọng nói.

***Hạn chế:***

* Chúng em chưa thể xử lí được những lỗi training, tính toán xác suất của model
* Tính năng nhận diện giọng nói chưa hoạt động tốt

***Kết luận:***

Do kiến thức về trí tuệ nhân tạo, Neural Network còn hạn chế nên chắc chắn còn nhiều thiếu sót trong quá trình thực hiện đề tài. Chúng em mong nhận được nhiều ý kiến, sự chỉ bảo từ thầy cô để có thể phát triển đề tài tốt hơn.

Chúng em xin chân thành cảm ơn thầy cô!

**TÀI LIỆU THAM KHẢO**

Intelligent AI Chatbot in Python – NeuralNine (năm 2020)

<https://www.youtube.com/watch?v=1lwddP0KUEg>

Create A Chatbot GUI Application With Tkinter - Patrick Loeber (năm 2021)

<https://www.youtube.com/watch?v=RNEcewpVZUQ>

Mạng nơ ron nhân tạo là gì? Tại sao chúng ta cần mạng nơ ron nhân tạo? – Elcom (năm 2022)

<https://www.elcom.com.vn/mang-no-ron-nhan-tao-la-gi-tai-sao-chung-ta-can-mang-no-ron-nhan-tao-1669018888>

Thuật Toán Tìm Kiếm Tuyến Tính (Tìm Kiếm Tuần Tự) - Vdragon(dnmtechs - năm 2019)

<https://dnmtechs.com/thuat-toan-tim-kiem-tuyen-tinh-tim-kiem-tuan-tu/?fbclid=IwAR37bct7KPMNOlLOK1GFBH8G-qGS5pKGQeszjAU6WGnCzi0PNE3_XKTkL0o>

Xử lí ngôn ngữ tự nhiên là gì? - funix.edu.vn (năm 2021)

<https://funix.edu.vn/chia-se-kien-thuc/xu-ly-ngon-ngu-tu-nhien-la-gi/?fbclid=IwAR2_G8FzopT0WVxCcvsmopAjNwbgjHo_A5EG93-aenpVffABUI-261QyW9Q>

Các loại trí tuệ nhân tạo – Khánh Chi (cohost)

<https://www.cohost.vn/blog-posts/cac-loai-tri-tue-nhan-tao>

Giáo trình nhập môn trí tuệ nhân tạo – Từ Minh Phương (năm 2014)