

**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN**

**KHOA** **CÔNG NGHỆ THÔNG TIN VÀ KINH TẾ SỐ**

**------- \*\*\* -------**



**Môn: Đề án**

***Đề tài: Trang web bán đồ thương mại điện tử kết hợp chatbot giúp người dùng***

***dễ dàng trong việc chọn lựa sản phẩm***

*Họ và tên: Đoàn Trần Nghiệp*

*Lớp: 63B.Công nghệ thông tin*

*Mã sinh viên: 11214276*

*Giảng viên hướng dẫn: TS. Đặng Tuấn Anh*

***Hà Nội, ngày 20 tháng 11 năm 2024***

**Mục lục**

Contents

[**LỜI NÓI ĐẦU** 3](#_Toc183771848)

[**Chương 1: Tổng quan** 4](#_Toc183771849)

[I. Lý do chọn đề tài 4](#_Toc183771850)

[II. Một số khái niệm 4](#_Toc183771851)

[**Chương 2: Nội dung chính** 5](#_Toc183771852)

[**Phần 1: Các thư viện, ngôn ngữ, framework cần dùng cho bài toán** 5](#_Toc183771853)

[I. Các ngôn ngữ lập trình cần thiết 5](#_Toc183771854)

[II. Các thư viện cần cài đặt 5](#_Toc183771855)

[III. Các framework cần cài đặt 6](#_Toc183771856)

[**Phần 2: Xây dựng sản phẩm** 7](#_Toc183771857)

[I. Trang web bán hàng thương mại điện tử 7](#_Toc183771858)

[II. Mô hình Chatbot 8](#_Toc183771859)

[III. Triển khai chatbot 13](#_Toc183771860)

[IV. Xây dựng giao diện chatbot 15](#_Toc183771861)

[**Tài liệu tham khảo** 18](#_Toc183771862)

# **LỜI NÓI ĐẦU**

Trong thế kỷ 21, vai trò của trí tuệ nhân tạo đã trở nên ngày càng quan trọng và đa dạng hơn trong nhiều lĩnh vực của cuộc sống. Từ ứng dụng trong công nghiệp, y tế, tài chính, đến giáo dục và giải trí, trí tuệ nhân tạo đang đóng vai trò không thể phủ nhận trong việc tạo ra những thay đổi đáng kể và tích cực trong xã hội.

Và để hiện thực hóa ý nghĩa đó, em đã cố gắng để tạo ra 1 sản phẩm kết hợp giữa web bán hàng cùng với chatbot tự động trả lời câu hỏi của khách hàng nhờ vào trí tuệ nhân tạo. Do vẫn còn nhiều hạn chế nên chắc chắn sản phẩm của em sẽ có những sai sót nhất định, mong thầy xem xét và góp ý để giúp bài tiểu luận này hoàn thiện hơn ạ.

Em xin chân thành cảm ơn.

Đoàn Trần Nghiệp

# **Chương 1: Tổng quan**

1. Lý do chọn đề tài

Ở trong thời đại công nghệ hiện nay, con người không thể hoạt động 24/7 để có thể đáp ứng mọi nhu cầu của xã hội. Nhưng máy móc thì có thể, và vị trí tư vấn viên chắc chắn sẽ là điều em muốn nói tới ở đây. Nếu có đủ dữ liệu, chúng có thể dễ dàng trả lời mọi câu hỏi thường gặp của khách hàng. Có lẽ mọi website hiện nay đều đã có những chức năng tuyệt vời này, hay chúng ta vẫn gọi là “chatbot”. Vì để hợp với xu hướng thị trường, em cũng đã thử tự tạo cho mình 1 trang web kết hợp trí tuệ nhân tạo để tự tạo 1 con chatbot cho riêng mình.

1. Một số khái niệm
2. Chatbot

Chatbot là một phần mềm sử dụng trí tuệ nhân tạo (AI) hoặc các quy tắc lập trình để trò chuyện với con người qua các giao diện như tin nhắn văn bản, âm thanh, hoặc thậm chí video. Mục tiêu của chatbot là mô phỏng cuộc trò chuyện tự nhiên, cung cấp câu trả lời cho các câu hỏi, giải quyết vấn đề hoặc thực hiện các tác vụ cụ thể mà người dùng yêu cầu.

1. Trí tuệ nhân tạo

Trí tuệ nhân tạo là một lĩnh vực trong khoa học máy tính, nghiên cứu về việc tạo ra các hệ thống hoặc máy móc có khả năng thực hiện các nhiệm vụ mà thường đòi hỏi trí tuệ con người. Điều này bao gồm các hoạt động như học hỏi, nhận thức, suy luận, giải quyết vấn đề, và thậm chí là sáng tạo.

# **Chương 2: Nội dung chính**

## **Phần 1: Các thư viện, ngôn ngữ, framework cần dùng cho bài toán**

1. Các ngôn ngữ lập trình cần thiết
2. Python

Python là một ngôn ngữ lập trình cao cấp, dễ học và sử dụng, được phát triển bởi Guido van Rossum vào cuối những năm 1980 và phát hành lần đầu tiên vào năm 1991. Python nổi bật với cú pháp đơn giản, rõ ràng, và dễ đọc, giúp lập trình viên dễ dàng viết và duy trì mã nguồn.

1. Json

JSON là một định dạng dữ liệu nhẹ, dễ đọc và ghi, được sử dụng rộng rãi để trao đổi dữ liệu giữa các hệ thống và ứng dụng. JSON là một chuẩn mở và có thể được sử dụng với nhiều ngôn ngữ lập trình khác nhau, bao gồm JavaScript, Python, Ruby, Java, PHP, v.v.

1. Các thư viện cần cài đặt
2. NLTK

NLTK là một thư viện mã nguồn mở trong Python dành cho xử lý ngôn ngữ tự nhiên. NLTK cung cấp một bộ công cụ mạnh mẽ để xử lý, phân tích và hiểu các dữ liệu ngôn ngữ tự nhiên, phục vụ cho các tác vụ như phân tích cú pháp, phân tích ngữ nghĩa, nhận diện thực thể, phân loại văn bản, và nhiều nhiệm vụ khác liên quan đến ngôn ngữ.

1. TensorFlow

TensorFlow là một thư viện mã nguồn mở được phát triển bởi Google dành cho Machine Learning và Deep Learning. TensorFlow giúp xây dựng và triển khai các mô hình học máy, đặc biệt là các mô hình học sâu, một phương pháp quan trọng trong AI và phân tích dữ liệu. TensorFlow được thiết kế để có thể chạy trên nhiều nền tảng khác nhau như máy tính cá nhân, các máy chủ lớn, hoặc trên thiết bị di động

1. Các framework cần cài đặt
2. Flask

Flask là một framework web nhẹ cho Python, được sử dụng để phát triển các ứng dụng web và dịch vụ API. Flask được thiết kế với mục tiêu đơn giản hóa việc phát triển web mà không làm mất đi tính linh hoạt. Với Flask, bạn có thể tạo ra các ứng dụng từ đơn giản đến phức tạp, tùy thuộc vào nhu cầu của bạn.

Các đặc điểm nổi bật của Flask:

Nhẹ và đơn giản: Flask không đi kèm với quá nhiều tính năng mặc định, cho phép bạn chọn các thư viện và công cụ cần thiết cho dự án. Điều này giúp bạn có thể tùy chỉnh ứng dụng theo nhu cầu mà không bị ràng buộc bởi các tính năng không cần thiết.

Không cấu trúc cố định: Flask không yêu cầu bạn phải tuân theo một cấu trúc thư mục cụ thể, điều này mang lại sự linh hoạt tối đa. Bạn có thể tổ chức mã nguồn theo cách bạn muốn.

Hỗ trợ WSGI (Web Server Gateway Interface): Flask hoạt động theo chuẩn WSGI, giúp bạn dễ dàng triển khai ứng dụng trên các máy chủ web.

1. Werkzeug

Werkzeug là một thư viện Python mạnh mẽ và linh hoạt, chủ yếu được sử dụng như một bộ công cụ WSGI (Web Server Gateway Interface) để phát triển các ứng dụng web. Werkzeug là một thành phần quan trọng trong hệ sinh thái Flask, vì Flask sử dụng Werkzeug như là công cụ HTTP thấp cấp để xử lý các yêu cầu (request) và phản hồi (response).

Werkzeug không phải là một framework web, mà là một thư viện hỗ trợ các chức năng cơ bản để xây dựng ứng dụng web, giúp các nhà phát triển dễ dàng xử lý HTTP, tạo ra các ứng dụng có khả năng mở rộng cao.

Các tính năng nổi bật của Werkzeug:

Xử lý HTTP: Werkzeug cung cấp các công cụ để xử lý các yêu cầu và phản hồi HTTP, bao gồm việc đọc dữ liệu yêu cầu, xử lý URL, tạo các mã trạng thái HTTP và nhiều tính năng khác.

Hỗ trợ WSGI: Werkzeug là một bộ công cụ WSGI, giúp kết nối các ứng dụng web Python với máy chủ web. Nó tương thích với các ứng dụng web theo chuẩn WSGI, vì vậy có thể hoạt động với nhiều máy chủ web như Gunicorn, uWSGI hoặc một máy chủ web đơn giản như Flask's built-in server.

Tính năng debugger: Werkzeug cung cấp một debugger mạnh mẽ để giúp phát hiện lỗi trong ứng dụng web, hiển thị thông tin chi tiết về lỗi và trạng thái của ứng dụng.

## **Phần 2: Xây dựng sản phẩm**

1. Trang web bán hàng thương mại điện tử

Đây là 1 nền tảng trực tuyến cho phép người dùng mua và bán sản phẩm hoặc dịch vụ qua Internet. Các trang web này cung cấp một trải nghiệm mua sắm trực tuyến đầy đủ, từ việc duyệt các sản phẩm đến thanh toán và giao hàng.

Cấu trúc đơn giản của một trang web thương mại điện tử:

Trang chủ: Đây là trang đầu tiên người dùng sẽ thấy khi truy cập vào website. Nó thường giới thiệu các sản phẩm nổi bật, khuyến mãi, hoặc các danh mục sản phẩm chính.

Danh mục sản phẩm: Các sản phẩm được phân loại theo từng nhóm hoặc thể loại (ví dụ: quần áo, điện tử, gia dụng, v.v.). Người dùng có thể dễ dàng tìm thấy sản phẩm mình cần thông qua các bộ lọc và tìm kiếm.

Trang sản phẩm: Mỗi sản phẩm có một trang riêng với thông tin chi tiết như tên, mô tả, giá, hình ảnh, đánh giá từ khách hàng, và nút "Thêm vào giỏ hàng".

Giỏ hàng: Đây là nơi người dùng lưu trữ các sản phẩm đã chọn để mua. Giỏ hàng cho phép người dùng xem lại sản phẩm, thay đổi số lượng, hoặc loại bỏ món hàng trước khi thanh toán.

Thanh toán: Quá trình người dùng điền thông tin để thanh toán cho các sản phẩm đã chọn. Điều này thường bao gồm lựa chọn phương thức thanh toán (thẻ tín dụng, PayPal, chuyển khoản, v.v.) và thông tin giao hàng.

Tài khoản người dùng: Người dùng có thể đăng ký hoặc đăng nhập vào tài khoản cá nhân để theo dõi đơn hàng, quản lý địa chỉ giao hàng, và lưu các sản phẩm yêu thích.

Thanh toán và giao hàng: Sau khi hoàn tất thanh toán, trang web sẽ xử lý đơn hàng và giao hàng cho người mua. Thông tin về trạng thái đơn hàng và các bản cập nhật sẽ được gửi qua email hoặc trong tài khoản người dùng.

Trang web bán đồ thương mại điện tử hiện nay có thể rất đơn giản hoặc phức tạp tùy thuộc vào quy mô và loại sản phẩm được bán, nhưng mục tiêu chung là tạo ra một trải nghiệm mua sắm trực tuyến dễ dàng, tiện lợi và an toàn cho người tiêu dùng.

1. Mô hình Chatbot
2. Mô hình Bag of Words

Bag of Words là một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên dùng để biểu diễn văn bản dưới dạng một tập hợp các từ. Ý tưởng chính của mô hình Bag of Words là chuyển đổi văn bản thành một tập hợp các từ, mà không quan tâm đến thứ tự của chúng, và chỉ tính tần suất xuất hiện của mỗi từ trong văn bản.

Nguyên lý của mô hình Bag of Words:

Không quan tâm đến thứ tự từ: Trong BoW, thứ tự của các từ trong văn bản không được tính đến. Điều này có nghĩa là chỉ cần biết mỗi từ xuất hiện trong văn bản hay không và số lần nó xuất hiện là đủ.

Biểu diễn dưới dạng vector: Mỗi văn bản sẽ được chuyển thành một vector số, trong đó mỗi phần tử của vector đại diện cho số lần một từ cụ thể xuất hiện trong văn bản. Các từ được biểu diễn bởi một "từ điển" toàn bộ các từ trong tập văn bản.

Mô hình đơn giản: BoW là một mô hình rất đơn giản nhưng lại khá mạnh mẽ trong nhiều tác vụ NLP như phân loại văn bản, phân tích cảm xúc, tìm kiếm tài liệu, v.v.

1. Mô hình mạng nơron

Để con chatbot có thể học được, em sẽ tạo 1 file json gồm những từ khóa cụ thể, để máy có thể học được 1 cách dễ dàng.

A screen shot of a computer

Description automatically generated

Cùng với đó, em sẽ phải cài đặt các thư viện đã được nhắc đến phía trên, và phải xử lý dữ liệu sao cho chatbot này chỉ nhận những các chữ cái mà không phải dấu câu.

A computer screen shot of a program code

Description automatically generated

Trong trường hợp này, mảng words sẽ chứa các từ đã được mã hóa từ từng ‘patterns’ chứa các câu thuộc về chủ đề, sau đó được sắp xếp lại theo đúng thứ tự, cùng lúc đó bỏ qua các dấu câu. Các từ trong mảng ’words’ đã được rút gọn thành từ gốc nhờ hàm lemmatize để giảm thiểu kích thước cho mảng và tăng tính nhất quán.

Mảng classes sẽ chứa các thuộc tính ‘tag’ của từng ‘intents’ trong cơ sở dữ liệu đã được tạo. Sau đó mảng classes cũng dược sắp xếp lại theo bảng chữ cái

Mảng document sẽ chứa danh sách từ đã được mã hóa từ ‘pattern’ của 1 chủ đề cộng với tên (‘tag’) của chủ đề đó.

Sau đó, 2 mảng words và classes sẽ được biến đổi thành 1 chuỗi dữ liệu byte, sau đó lần lượt được viết vào,lưu lại tại 2 file words.pkl và classes.pkl

A computer screen shot of a program code

Description automatically generated

Sau khi đã chuẩn bị xong, chúng ta sẽ tạo mảng training để chứa các dữ liệu dạy học cho bot, cùng với đó là mảng outputEmpty gồm số 0 và độ dài các “tag” cần phân loại. Sau đó, chúng ta sẽ duyệt qua từng từ trong document. Với từng phân tử, lưu trữ trong “bag” nếu tồn tại, còn không thì them số 0 vào.

outputRow sẽ là đầu ra của sản phẩm, đặt giá trị tại 1 vị trí tương ứng với class “tag” của lớp văn bản. Dữ liệu bao gồm “bag” và outputRow được đưa vào mảng training dưới dạng 1 vector huẩn luyện. Mảng trainX chứa dữ liệu đầu vào cho việc huẩn luyện “bag” còn mảng trainY sẽ chứa dữ liệu đầu ra.

Tiếp theo, chúng ta sẽ phải định nghĩa mạng nơron, bằng cách tạo 1 mô hình bằng hàm sequentinal.

A screen shot of a computer code

Description automatically generated

Em sẽ them lớp dense đầu tiên dài 128 đơn vị, chỉ định kích thước đầu vào của input. Hàm activation sẽ kích hoạt ReLU, giúp mô hình học được các quan hệ phi tuyến tính trong dữ liệu.

Tiếp theo ta sẽ them lớp dropout với tỷ lệ rớt là 50%. Dropout giúp giảm quá khớp bằng cách ngẫu nhiên “tắt” một phần đơn vị trong lớp trong quá trình huấn luyện, giúp mô hình không quá phụ thuộc vào bất kỳ đặc trưng nào.

Dense(len(trainY[0])): Đây là lớp đầu ra (output layer), số lượng đơn vị (neurons) trong lớp này được xác định bởi len(trainY[0]), nghĩa là số lớp đầu ra tương ứng với số lượng nhãn (labels) trong dữ liệu huấn luyện (y).

Cuối cùng ta sẽ khởi tạo và lưu trữ mô hình này.

A screen shot of a computer

Description automatically generated

Em sẽ tạo tối ưu hóa adam để có được hiệu suất huấn luyện tuyệt nhất, sau đó biên dịch mô hình với hàm mất mát là categorical\_crossentropy. Optimizer là adam đã được khai báo ở trên. Mô hình được đánh giá dựa trên độ chính xác khi huấn luyện mô hình.

Tiếp theo, ta huấn luyện mô hình bằng hàm fit, dữ liệu đầu vào và đầu ra được chuyển thành numpy trước khi truyền vào mô hình, số lần lặp (epoch) là 200, số mẫu sử dụng mỗi lần cập nhật trọng số là 5.

Cuối cùng, lưu mô hình vào file chatbot\_model.h5. Model.fit trả về một đối tượng hist để người dùng lưu trữ để có thể sử dụng lại sau.

1. Triển khai chatbot
2. Import tất cả thư viện cùng thuộc tính để xây dựng chatbot

A screen shot of a computer program

Description automatically generated

Lemmatizer: Khởi tạo đối tượng lemmatizer từ NLTK để chuẩn hóa từ ngữ (lemmatization). Điều này giúp biến các từ như "run", "running", "runs" về cùng một gốc "run".

pickle.load: Sử dụng pickle.load() để nạp từ điển hoặc danh sách các từ đã được lưu trữ trong tệp words.pkl. Đây có thể là danh sách các từ đã được xử lý từ dữ liệu huấn luyện, giúp chatbot nhận diện các từ khoá.

self.classes: Tải các lớp (class labels) đã được huấn luyện và lưu trữ trong tệp classes.pkl. Các lớp này tương ứng với các nhãn phân loại mà chatbot có thể phản hồi.

self.model: Nạp mô hình học máy đã huấn luyện từ tệp chatbot\_model.h5 sử dụng Keras. Đây là mô hình mạng nơ-ron được sử dụng để phân loại các câu hỏi của người dùng thành các lớp hoặc chủ đề cụ thể.

self.stateSpace: Đây là một danh sách các trạng thái có thể có của chatbot trong quá trình giao tiếp. Các trạng thái có thể bao gồm "default" (trạng thái mặc định) và "typing coeff" (có thể là trạng thái khi chatbot đang xử lý một phép toán hay tính toán nào đó).

1. Xử lý các vấn đề còn lại

Đầu tiên sẽ là phải xử lý câu được đưa vào

A computer code on a black background

Description automatically generated

Câu hỏi sẽ được chia thành các từ và được chuẩn hóa thành dạng gốc qua hàm lemmatizer như đã nói ở trên, cuối cùng là trả về danh sách các từ đã qua xử lý.

Tiếp theo sẽ là chuyển đổi câu đưa vào thành vector

A screen shot of a computer code

Description automatically generated

Chúng ta sẽ dung hàm clean\_up\_sentence để xử lý câu. Em sẽ tạo 1 danh sách có số phần tử bằng số từ của câu đưa vào với các phần tử đều bằng 0. Sau đó duyệt từng từ xem có thuộc tập từ vựng không, nếu có thì gán giá trị bằng 1. Cuối cùng, trả lại vector túi từ đã được chuyển thành numPy có giá trị 1 tại mỗi vị trí mà từ đó tồn tại trong câu đưa vào và 0 cho phần còn lại.

Vì dung trí tuệ nhân tạo nên việc phải dự đoán từ vựng cũng rất quan trọng.

A computer screen with text on it

Description automatically generated

Đầu tiên sẽ dung biến bow để chuyển câu thành vector đặc trưng, sau đó dự đoán bằng mô hình học máy và lọc các lớp có xác suất lớn hơn ngưỡng (0.25). Chúng ta sẽ sắp xếp theo xác suất giảm dần và trả về intent xác suất.

Bước cuối cùng sẽ là trả về kết quả.

A screen shot of a computer program

Description automatically generated

Em sẽ in ra danh sách các intent và lấy tag intent đầu tiên. Em cũng sẽ duyệt qua các intent trong intents\_json để chọn phản hồi ngẫu nhiên từ các câu trả lời này. Cuối cùng sẽ là trả về kết quả tìm được.

1. Xây dựng giao diện chatbot

Em sẽ xây dựng 1 file app.py, 1 file chat.html và 1 style.css

A computer screen shot of text

Description automatically generated

A black screen with text on it

Description automatically generated

**A screen shot of a computer

Description automatically generated**

# **Tài liệu tham khảo**

1. [Welcome to Flask — Flask Documentation (3.1.x) (palletsprojects.com)](https://flask.palletsprojects.com/en/stable/)
2. [NLTK :: Natural Language Toolkit](https://www.nltk.org/)
3. [Keras: Deep Learning for humans](https://keras.io/)
4. [6.2. Feature extraction — scikit-learn 1.5.2 documentation](https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction)
5. [Tutorials  |  TensorFlow Core](https://www.tensorflow.org/tutorials)
6. Wikipedia
7. ChatGPT