**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

****

**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH TRƯỜNG HỌC MÁY TÍNH**

**Đề tài**

**XÂY DỰNG TRỢ LÝ ẢO**

**TƯ VẤN NGÀNH HỌC**

**Sinh viên thực hiện: Võ Ngọc Long**

**Mã số: B1812282**

**Khóa: 44**

**Cần Thơ, 07/2022BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

****

**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH TRƯỜNG HỌC MÁY TÍNH**

**Đề tài**

**XÂY DỰNG TRỢ LÝ ẢO**

**TƯ VẤN NGÀNH HỌC**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  **Ths. Phạm Xuân Hiền** | **Sinh viên thực hiện:** |
| **Võ Ngọc Long** |
|  | **Mã số: B1812282** |
| **Khóa: 44** |

**Cần Thơ, 07/2022**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Cần Thơ, ngày … tháng … năm 2022

Giáo viên hướng dẫn

**LỜI CẢM ƠN**

Lời đầu tiên em xin được phép bày tỏ lòng biết ơn chân thành và sâu sắc đến cô Phạm Xuân Hiền trường CNTT & TT đã tận tình hướng dẫn, chỉ bảo, góp ý trong suốt quá trình thực nghiên cứu và thực hiện đề tài luận văn.

Cùng với đó em xin gửi lời cảm ơn chân thành đến các Thầy Cô trường Đại học Cần Thơ, đặc biệt là cảm ơn các thầy cô Trường Công nghệ Thông tin và Truyền thông đã hướng dẫn, giảng dạy cho em những kiến thức làm nền tảng thực hiện luận văn. Trong suốt nhiều năm học tập và rèn luyện tại trường Đại học Cần Thơ, đặc biệt là Trường Công nghệ Thông tin và Truyền thông, đến nay em đã kết thúc khóa học và hoàn thành luận văn tốt nghiệp.

Lời cuối cùng, em xin gửi lời cảm ơn đến cha mẹ, gia đình là chỗ dựa tinh thần vững chắc, là nguồn động lực để em tiếp tục cố gắng trên con đường học tập của mình. Em cám ơn thầy cô đã luôn lo lắng, dạy dỗ, giúp đỡ em trên con đường học tập. Cám ơn người sử dụng bè, anh chị cùng trường đã giúp đỡ, chia sẻ khó khăn, cho em những góp ý chân thành.

Do không thể tránh được thiếu sót trong lúc thực hiện luận văn. Nhưng vẫn cố gắng hết sức để hoàn thành luận văn của mình một cách tốt nhất. Mong nhận được sự góp ý quý báu từ thầy cô và các người sử dụng để em có thêm kinh nghiệm thực hiện các nghiên cứu sau này.

Cần Thơ, ngày tháng 07 năm 2022

Người viết

Võ Ngọc Long

# MỤC LỤC

Contents

[MỤC LỤC i](#_Toc120110111)

[DANH MỤC HÌNH ẢNH iii](#_Toc120110112)

[DANH MỤC BIỂU BẢNG iv](#_Toc120110113)

[DANH MỤC TỪ VIẾT TẮT v](#_Toc120110114)

[TÓM TẮT vi](#_Toc120110115)

[ABSTRACT vii](#_Toc120110116)

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN 1](#_Toc120110117)

[1.1. Đặt vấn đề 1](#_Toc120110118)

[1.2. Nghiên cứu liên quan 2](#_Toc120110119)

[1.2.1. Website 2](#_Toc120110120)

[1.2.2. Chatbot 2](#_Toc120110121)

[1.2.3. Hệ thống phân loại bình luận 5](#_Toc120110122)

[1.3. Mục tiêu đề tài 5](#_Toc120110123)

[1.4. Đối tượng và phạm vi nghiên cứu 6](#_Toc120110124)

[1.5. Phương pháp nghiên cứu 6](#_Toc120110125)

[1.6. Nội dung nghiên cứu 6](#_Toc120110126)

[1.7. Bố cục 7](#_Toc120110127)

[CHƯƠNG 2. XÂY DỰNG HỆ THỐNG 8](#_Toc120110128)

[2.1. Mô tả chi tiết bài toán 8](#_Toc120110129)

[2.1.1. Vấn dề và giải pháp liên quan đến bài toán 9](#_Toc120110130)

[2.1.2. Ngôn ngữ lập trình Python 10](#_Toc120110135)

[2.1.3. Ngôn ngữ lập trình Javascript 10](#_Toc120110136)

[2.1.4. HTML và CSS 11](#_Toc120110137)

[2.1.5. Phương pháp học sâu (Deep Learning) 11](#_Toc120110138)

[2.1.6. Máy học véc-tơ hỗ trợ (Support Vector Machine) 17](#_Toc120110139)

[2.1.7. K láng giềng 19](#_Toc120110140)

[2.1.8. Thư viện Keras 20](#_Toc120110141)

[2.1.9. Framework Flask 21](#_Toc120110142)

[2.1.10. API 21](#_Toc120110143)

[2.1.11. Hệ quản trị cơ sở dữ liệu 22](#_Toc120110144)

[2.1.12. Word Tokenization 23](#_Toc120110145)

[2.2. Mô tả hệ thống trang web 23](#_Toc120110146)

[2.3. Yêu cầu chức năng 24](#_Toc120110147)

[2.3.1. Người dùng admin 24](#_Toc120110148)

[2.3.2. Người dùng 24](#_Toc120110149)

[CHƯƠNG 3. HỆ THỐNG HỖ TRỢ 25](#_Toc120110150)

[3.1. Chatbot 25](#_Toc120110151)

[3.2. Thiết kế hệ thống 25](#_Toc120110152)

[3.2.1. Tập dữ liệu 26](#_Toc120110153)

[3.2.2. Tiền xử lý dữ liệu 32](#_Toc120110154)

[3.2.3. Xây dựng dữ liệu huấn luyện 33](#_Toc120110155)

[3.2.4. Xây dựng mô hình huấn luyện 36](#_Toc120110156)

[3.2.5. Xử lý câu hỏi người dùng và đưa ra dự đoán 39](#_Toc120110157)

[3.2.6. Thu thập và thêm dữ liệu từ câu hỏi người dùng 40](#_Toc120110158)

[3.2.7. Các giải thuật dùng để huấn luyện mô hình và kiểm thử độ chính xác 40](#_Toc120110159)

[3.3. Xây dựng hệ thống 41](#_Toc120110160)

[3.3.1. Tạo mô hình dự đoán 41](#_Toc120110161)

[3.3.2. Xử lý câu hỏi người dùng và dự đoán câu trả lời 45](#_Toc120110162)

[3.3.3. Thu thập dữ liệu câu hỏi người dùng 47](#_Toc120110163)

[3.3.4. Thêm câu hỏi và tập huấn luyện 47](#_Toc120110164)

[CHƯƠNG 4. KIỂM THỬ VÀ ĐÁNH GIÁ 48](#_Toc120110165)

[4.1. Kiểm tra và đánh giá 48](#_Toc120110166)

[4.1.1. Độ chính xác của ba giải thuật máy học 48](#_Toc120110167)

[4.1.2. Độ chính xác của mô hình với giải thuật SGD 50](#_Toc120110168)

[TÀI LIỆU THAM KHẢO 51](#_Toc120110169)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BIỂU BẢNG

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Stt | Từ nguyên gốc | Từ viết tắt |
| 1 | Trợ lý ảo | Chatbot |
| 2 | Support Vector Machines | SVM |
| 3 | K Nearest Neighbor | kNN |
| 4 | Gausian Naïve Bayes | Bayes |
| 5 | Stochastic Gradient Descent | SGD |
| 6 |  | HTML |
| 7 |  | CSS |
| 8 | Application Programming Interface | API |
|  | Database Management System | DBMS |

# TÓM TẮT

Hằng năm, vào mỗi mùa tuyển sinh, phụ huynh và các người sử dụng học sinh sẽ có nhu cầu tham khảo về “Tuyển sinh tại các Trường Đại Học, Cao Đẳng”, với tâm thế là trường Đại học lớn tại khu vực Đồng Bằng Sông Cửu Long, Trường Đại học Cần Thơ – một điểm Trường nhận được rất nhiều sự quan tâm của phụ huynh và học sinh, những thắc mắc về liên quan đến ngành học và chương trình đào tạo. Do số lượng ngành học tại Đại học Cần Thơ rất lớn, đội ngũ tư vấn của trường thì không thể đáp ứng đáp ứng với số lượng câu hỏi lớn và không thể trả lời mọi lúc các câu hỏi từ phía học sinh và phụ huynh, nên nghiên cứu này sẽ tập tung ứng dụng Deep Learning để xây dựng hệ thống Chatbot phục vụ trên website. Chatbot sẽ được thử nghiệm để xử lý những thắc mắc liên quan đến khối ngành Công nghệ Thông tin của Trường Công nghệ Thông tin và Truyền Thông trực thuộc Đại học Cần Thơ. Số lượng các nhóm ngành về Công nghệ Thông tin tại Trường Công nghệ Thông Tin và Truyền thông là 8 ngành gồm có 6 ngành cũ và 2 ngành mới, điều này sẽ giúp nghiên cứu tập trung sửa lỗi và tạo lập một không gian dữ liệu nhanh chóng để có thể sử dụng những thuật toán như SVM, kNN, Bayes và SGD. Các thuật toán này dùng để huấn luyện những mô hình máy học, căn cứ vào kết quả huấn luyện để xây dựng hệ thống Chatbot có thể dùng để tư vấn tuyển sinh trong tương lai.

Đối với hệ thống sau khi được hoàn thiện, người sử dụng có thể thực hiện trao đổi với Chatbot theo hướng đặt câu hỏi và Chatbot sẽ trả lời. Tập dữ liệu được dùng để huấn luyện mô hình bao gồm 1043 câu hỏi và số câu trả lời là 101 câu trả lời. Số câu hỏi được thu thập thông qua trang tuyển sinh của Trường Đại học Cần Thơ, và thu thập thông qua thực hiện khảo sát từ nhiều phương tiện như: dùng hệ thống Google From, khảo sát trực tiếp từ các người sử dụng sinh viên,… các câu trả lời thì được quản trị viên trả lời dựa trên kinh nghiệm thực tiễn, thông tin được cung cấp từ giảng viên, thông tin tìm hiểu được từ nhiều phương tiện, những câu trả lời sẽ đáp ứng được yêu cầu của các người sử dụng học sinh và phụ huynh.

Sau khi thực hiện huấn luyện mô hình, kết quả dựa đoán chính xác của 4 giải thuật được sử dụng trong nghiên cứu không quá chênh lệch. Hai giải thuật SVM và SGD có độ chính xác xấp xỉ đạt hơn 90% với SGD độ chính xác xấp xỉ đạt 92% nên nghiên cứu quyết định sẽ xây dựng mô hình Chatbot dựa trên thuật toán SGD.

# ABSTRACT

# GIỚI THIỆU TỔNG QUAN

Chương 1 sẽ trình bày các phần gồm đặt vấn đề, nghiên cứu liên quan, website và ứng dụng, chatbot, máy học, mục tiêu đề tài, đối tượng và phạm vi nghiên cứu, phương pháp nghiên cứu, nội dung nghiên cứu, bố cục luận văn.

## Đặt vấn đề

Hằng năm vào mỗi mùa tuyển sinh, phụ huynh và học sinh có rất nhiều câu hỏi và thắc mắc về các ngành học cũng như những cơ hội nghề nghiệp sau khi ra trường, cách thức tuyển sinh,... Đa phần các trường đại học đều đăng tải những thông tin trên trên website cũng như đã có đội ngũ tư vấn để hỗ trợ thắc mắc cho phụ huynh và học sinh. Tuy nhiên, các văn bản, thông tin đôi khi sử dụng các thuật ngữ chuyên môn và ít thân thiện với người sử dụng dẫn đến việc phụ huynh, học sinh khó có thể hiểu hết được các thông tin cần thiết và quan trọng của người tư vấn.

Và vấn đề đặt ra là liệu người tư vấn có thể hỗ trợ những thắc mắc cho mọi người mọi lúc, điều đó đòi hỏi số lượng nhân viên tư vấn nhất định và tốn kém một chi phí khá lớn.

Ngày nay, trong thời đại mà trí tuệ nhân tạo và máy học đang phát triển ngày một rực rỡ, và cho ra đời rất nhiều thiết bị thông minh, không nằm ngoài xu thế đó- ứng dụng Chatbot với tích hợp trí tuệ nhân tạo cũng đang hiện hữu rất nhiều trong cuộc sống hiện đại này. Hiện tại Chatbot được ứng dụng trong nhiều lĩnh vực. Nếu người sử dụng muốn trò chuyện, gặp gỡ và trò chuyện về những thứ như nhà hàng, thời tiết và tin tức hàng ngày thì Chatbot Luka có thể dễ dàng đáp ứng được nhu cầu đó. Nhu cầu trong việc tìm một người theo dõi năng suất thể dục hằng ngày một cách thông minh, có tính tương tác thì Chatbot Lark là một huấn luyện viên thể dục mà người sử dụng có thể tìm đến. Tiếng anh không được tốt và muốn cải thiện tiếng anh nhanh chóng và thuận lợi thì Chatbot Andy chính là sự lựa chọn tốt nhất.

Sự ra đời Chatbot mang lại nhiều lợi ích cho rất nhiều lĩnh vực cũng như đáp ứng nhu cầu của mọi người, chúng không chỉ đáp ứng những gì mọi người cần mà vấn đề thời gian cũng được cải thiện. Đặc biệt khi phụ huynh và học sinh muốn biết những thông cần thiết cho việc tuyển sinh ở bất kì thời gian nào, và họ cần những câu trả lời chính xác với tốc độ nhanh. Để đáp ứng nhu cầu trên thì việc có một Chatbot trả lời tư vấn tuyển sinh là cần thiết.

## Nghiên cứu liên quan

### Website

Hiện nay có nhiều loại hình website cung cấp các thông tin tuyển sinh của các trường đại học như: Đại học Cần Thơ[[1]](#footnote-1), Đại học Bách Trường Đại học Quốc gia Thành phố Hồ Chí Minh[[2]](#footnote-2), ... Tuy nhiên là việc tư vấn cho phụ huynh và học sinh và tìm kiếm thông tin ngành học trên website này còn hạn chế là khi người sử dụng có những thắc mắc về các ngành nghề tuyển sinh, về chỉ tiêu, về tỉ lệ tốt nghiệp và nhiều yếu tố khác. Do chưa hoàn toàn hiểu rõ về trang web, khiến cho người sử dụng sẽ cần một “trợ lý ảo” để hướng dẫn những thao tác, và “trợ lý ảo” cũng có thể cung cấp thêm thông tin cần thiết. Nắm bắt được nhu cầu này, nghiên cứu “Xây dựng Trợ lý ảo tư vấn ngành học” sẽ thực hiện xây dựng một Chatbot với giao diện sử dụng thân thiện cho người sử dụng, giao diện được lấy cảm hứng từ những cửa sổ chat tương tự như những ứng dụng nổi tiếng: Messager, Zalo,…

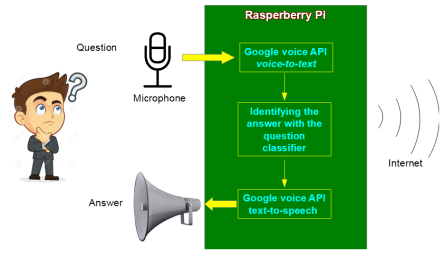
### Chatbot

Hiện nay chatbot đã trở nên rất phổ biến và ứng dụng trong khá nhiều lĩnh vực khác nhau. Trong đó có những tập đoàn lớn về công nghệ trên thế giới ra sức tạo những trợ lý ảo để người dùng có thể tương tác với các phần mềm, ứng dụng của họ.

Đỗ Viết Mạnh [1] xây dựng chatbot bán hàng sử dụng mô hình sinh trên nền tảng framework Rasa[[3]](#footnote-3) và Django[[4]](#footnote-4). Kết quả thử nghiệm của chatbot đạt độ chính xác là 80% dựa vào các câu hỏi đã được xây dựng trước đó theo kịch bản áp dụng cho hệ thống bán hàng.

Nghiên cứu của tác giả Đỗ Thanh Nghị đã xây dựng Chatbot trả lời tự động cho Sinh viên Công Nghệ Thông Tin [2]. Chatbot này được máy Raspberry Pi có thể trả lời tự động cho sinh viên ngành Công nghệ thông tin các câu hỏi liên quan đến môi trường học tập và phương pháp học tập bậc đại học, kĩ năng nghề nghiệp và xu hướng công nghệ trong tương lai.

Sinh viên thực hiện đặt câu hỏi cho Chatbot bằng cách nói bằng giọng nói Chatbot tiếp nhận xử lý và trả lời sinh viên bằng văn bản và phát ra câu trả lời bằng tiếng nói cho sinh viên nghê. Dữ liệu được thu thập và biên soạn gồm 986 câu hỏi và 213 câu trả lời từ nguồn dữ liệu tài liệu học tập.



Hình 1: Sơ đồ hệ thống Chatbot cho sinh viên Công nghệ thông tin

Nhóm nghiên cứu đã thực hiện huấn luyện mô hình bằng các phương pháp học máy và kết quả được ghi nhận và thống kê trong bảng sau:

|  |  |
| --- | --- |
| Phương pháp | Độ chính xác |
| Máy học Véc-tơ hỗ trợ | 76.77% |
| Mạng nơ-ron nhân tạo | 72.73% |
| Rừng ngẫu nhiên | 71.72% |
| k láng giềng | 65.66% |

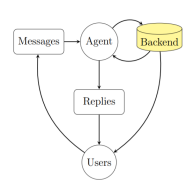
Bảng 1: Các phương pháp phân lớp và độ chính xác tương ứng

Windiatmoko [3] phát triển bởi các mô hình học sâu, chatbot được tích hợp vào Facebook[[5]](#footnote-5), được áp dụng bởi một mô hình trí tuệ nhân tạo nhằm tái tạo trí thông minh của con người với một số chương trình đào tạo cụ thể. Loại học sâu này dựa trên RNN[[6]](#footnote-6) có một số sơ đồ tiết kiệm bộ nhớ cụ thể cho mô hình học sâu, cụ thể là chatbot này sử dụng LSTM[[7]](#footnote-7) đã được tích hợp bởi khung Rasa. LSTM giúp tiết kiệm hiệu quả một số bộ nhớ cần thiết nhưng sẽ loại bỏ một số bộ nhớ không cần thiết.

Ngoài ra, Mamatha [4] tích hợp chatbot lên website thương mại điện tử giúp tư vấn khách hàng. Chatbot này sẽ hữu ích giúp bộ phận chăm sóc khách hàng phản ứng nhanh, có thể giải quyết nhiều câu hỏi từ khách hàng.

Trong lĩnh vực dịch vụ, nghiên cứu của giáo sư Louis Wehenkel có liên quan đến việc ứng dựng Chatbot trong việc hỗ trợ khách hành [5]. Nghiên cứu đặt trong những trường hợp thực tế về giải pháp sử dụng Chatbot cho công ty Gaming1. Mục tiêu nghiên cứu là thiết kế một giải thuật Chatbot đa ngôn ngữ có thể mở rộng và dễ dàng giao tiếp với các phần mềm hỗ trợ khác hàng hiện tại của công ty.

Quá trình xây dựng gồm việc xử lý dữ liệu được trích xuất thành một định dạng phù hợp cho các giải pháp máy học và triển khai trên nhiều nền tảng, có thể trả lời người dùng tròng thời gian thực. Và yêu cầu người dùng cung cấp thông tin nếu ý định họ không rõ ràng, đặc biệt hỗ trợ nhiều ngôn ngữ.



Hình 2: Mô hình Chatbot hỗ trợ khách hàng [2]

Dữ liệu được sử dụng trong quá trình nghiên cứu được trích xuất từ tập dữ liệu tin nhắn từ công ty có hệ thống hỗ trợ khách hàng. Không gian dữ liệu có khoảng 1.400.000 thông tin được công ty thư thập từ 2013. Khi dữ liệu được trích xuất, sẽ được phân tách theo ngôn ngữ bằng cách sử dụng các thẻ gắn liền với dữ liệu. Hiện có chín ngôn ngữ được hỗ trợ trong hệ thống Gaming1: tiếng Pháp, tiếng Hà Lan, tiếng Anh, tiếng Bồ Đào Nha, tiếng Tây Ban Nha, tiếng Romania, tiếng Serbia, tiếng Đức và tiếng Thổ Nhĩ kỳ.

Sau khi dữ liệu được xử lý sẽ được huấn luyện thông qua bốn mô hình gồm: LSTM đơn, GNU đơn, đầu vào GRU đảo ngược, GRU hai lớp. Kết quả thu lại như bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Network type | Best validation loss(lover is better) | Best validation accuracy (higher is better) | Total training time [min] |
| Single LSTM | 0.45 | 87.6% | 314 |
| Single GRU | 0.46 | 87.5% | 196 |
| Inverted GRU input | 0.49 | 88.2% | 198 |
| Two GRU layers | 0.52 | 86.4% | 352 |

Bảng 2: Kết quả nghiên cứu các mô hình cho Chatbot

Kết quả của nghiên cứu thể hiện rằng giải pháp đáp ứng yêu cầu và cho phép dễ dàng khai và phát triển nếu cần thực hiện các thay đổi. Nó có thể được hoạt động trên mọi hệ thống miễn là nó có thể lưu trữ môi trường Python.

### Hệ thống phân loại bình luận

## Mục tiêu đề tài

Với quá trình phát triển như hiện này, Trường Đại học Cần Thơ có số lượng tuyển sinh ngày càng nhiều dẫn đến việc số lượng các thắc mắc về những ngành học của trường nói chung, và trường Công Nghệ Thông Tin & Truyền thông nói riêng cũng tăng cao. Số lượng câu hỏi đến từ quý phụ huynh và học sinh càng lớn thì việc giải đáp tất cả những thắc mắc này một cách nhanh chóng – chính xác mà không có sự hỗ trợ từ công nghệ sẽ đặt ra một thách thức rất lớn với đội ngũ giảng viên, cán bộ, nhân viên,… của Đại học Cần Thơ.

Để có thể giải quyết vấn đề một cách nhanh chóng và có thể hỗ trợ giải đáp thắc mắc cho quý phụ huynh và các em học sinh cũng như đáp ứng được việc sử dụng các giải thuật về trí tuệ nhân tạo và máy học có thể hỗ trợ xây dựng một hệ thông trang web tích hợp chatbot. Chính điều này là cảm hứng cho Đề tài “Xây dựng trang web tích hợp trợ lý ảo tư vấn tuyển sinh trường Đại học Cần Thơ” nhằm tạo một trang web cung cấp các thông tin về Đại học Cần Thơ và các ngành, nhóm ngành tuyển sinh có tích hợp Chatbot có khả năng trả lời tự động những câu hỏi của quý phụ huynh và học sinh liên quan đến các quy đinh, thông tin tuyển của trường Công nghệ Thông tin và Truyền thông nói riêng và các nhóm ngành khác của Đại học Cần Thơ nói chung.

Hệ thống trang web tích hợp trợ lý ảo tư vấn tuyển sinh xây dựng với mục tiêu để đáp ứng được các chức năng sau:

* + Thiết kế giao diện dễ sử dụng, thân thiện.
  + Hỗ trợ người quản trị trong việc quản lý cơ sở dữ liệu.
  + Hỗ trợ chatbot tư vấn và hỗ trợ, giải đáp thắc mắc về các ngành, nhóm ngành Công nghệ Thông Tin.

## Đối tượng và phạm vi nghiên cứu

Chatbot được xây dựng nhằm mục đích hỗ trợ giải đáp các thắc mắc về thông tin tuyển sinh về các ngành của Đại học Cần Thơ. Đối tượng Chatbot hướng tới là những phụ huynh và các em học sinh có ý định thi hoặc xét tuyển vào các nhóm ngành này.

Ngoài ra đối tượng đề tài còn là ngôn ngữ lập trình Python[[8]](#footnote-8), thư viện Keras[[9]](#footnote-9), NLTK[[10]](#footnote-10) được sử dụng để lập trình hệ thống Chatbot, sử dụng HTML[[11]](#footnote-11), CSS[[12]](#footnote-12) và Javascript[[13]](#footnote-13) để thiết kế giao diện cho hệ thống

## Phương pháp nghiên cứu

“Xây dựng trang web tích hợp trợ lý ảo tư vấn tuyển sinh trường Đại học Cần Thơ” được lập trình chủ yếu bằng ngôn ngữ Python và đặc biệt là có sự hỗ trợ của một số thư viện như Keras, NLTK,… các phương pháp học máy như SVM, mạng nơ ron nhân tạo. Công cụ hỗ trợ cho việc lập trình là phần mềm Pycharm.

Bước đầu tiên trong quá trình xây dựng chương trình là tiền xử lý dữ liệu, phân chia các từ trong câu thành từ nhóm và xây dựng một tập dữ liệu. Chia tập dữ liệu thành các phần training và testing dựa trên dữ liệu có sẵn.

Xây dựng mạng nơ-ron bằng cách dùng Sequential API của Keras.

## Nội dung nghiên cứu

Các nội dung nghiên cứu của đề tài:

* + Nghiên cứu xây dựng front-end webiste bằng các ngôn ngữ: HTML, Bootstrap(CSS), Jquery, JavaScript.
  + Nghiên cứu xây dựng back-end website sử dụng.
  + Nghiên cứu hệ thống phân loại đánh giá sử dụng framework Python Flask.

## Bố cục

**Phần giới thiệu**

Giới thiệu tổng quát về đề tài.

**Phần nội dung**

**Chương 1:** Mô tả bài toán

**Chương 2:** Xây dựng hệ thống

**Chương 3:** Hệ thống hỗ trợ

**Phần kết luận**

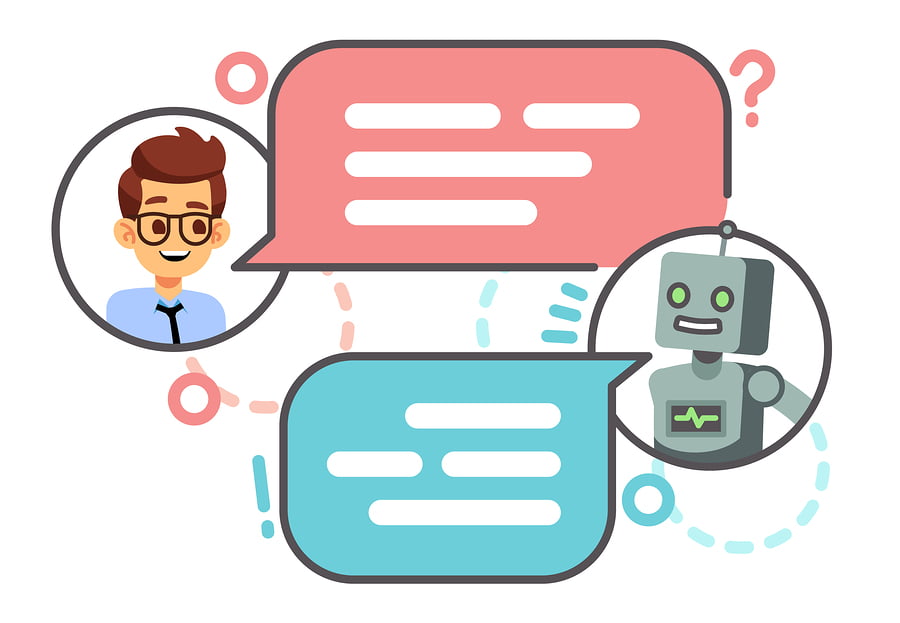
Trình bày kết quả đạt được và hướng phát triển hệ thống.

# XÂY DỰNG HỆ THỐNG

Chương 2 sẽ trình bày các mô tả chi tiết bài toán, những kỹ thuật dùng trong quá trình xây dựng hệ thống bao gồm những ngôn ngữ lập trình, công cụ, cơ sở dữ liệu, framework, tool,…

## Mô tả chi tiết bài toán

Nhằm đáp ứng như cầu giải đáp các thắc mắc liên quan đến việc tuyển sinh trường Đại học Cần Thơ của quý phụ huynh và học sinh. Vì thế, chương trình phải được xây dựng với một cấu trúc đơn giản, dễ sử dụng và thân thiện với người dùng về mặt giao diện và tính năng của chương trình. Câu trả lời phải đáp ứng được các tiêu chí chính xác cao và phải có tốc độ xử lý cao, khi một yêu cầu được đặt ra vào bất kể thời gian nào thì hệ thống phải có câu trả lời cho mọi người.



Hình 3: Minh họa việc giao tiếp giữa người dùng và Chatbot

Để thuận tiện cho việc cung cấp thông tin cho quý phú huynh và các em học sinh trong quá trình tuyển sinh. Chương trình sẽ được xây dựng cho phép người sử dụng hỏi và nhận được câu trả lời từ hệ thống thông qua văn bản, và hệ thống chatbot sẽ được tích hợp trên một trang web có liên kết đường dẫn đến trang web tuyển sinh chính của Đại học Cần Thơ. Khi quý phụ huynh và học sinh đưa ra các câu hỏi liên quan đến những thông tin về ngành học, hệ thống sẽ tiếp nhận và đưa ra câu trả lời trực tiếp được hiển thị trong của sổ chat trên website.

### Vấn dề và giải pháp liên quan đến bài toán

2. 1. 1. * 1. **Xử lý văn bản tiếng việt**

Hệ thống trợ lý ảo tư vấn tuyển sinh dựa trên mô hình câu hỏi từ người dùng và câu trả lời từ Chatbot đều diễn ra dưới hình thức văn bản.

Khi người dùng đặt câu hỏi cho hệ thống thông qua văn bản, ngôn ngữ được sử dụng sẽ là Tiếng Việt. Tiếng Việt là một ngôn ngữ có dấu câu, đặc biệt trong quá trình đặt câu hỏi, người dùng có thể sẽ nhập vào những từ ngữ viết tắt theo thói quen của người dùng. Ví dụ như:

* “Cntt diem chuan bn?”
* “Ngành máy tính đào tạo nh j?”
* “Học ngành này có khó ko?”,…

Đa phần mọi người việc hiểu những câu viết tắt, những câu không dấu,… là hoàn toàn dễ dàng nhưng đối với máy tính thì sẽ không thể hiểu được vì máy tính sẽ không có dữ liệu hay bất kỳ căn cứ nào để hiểu được những câu từ không nằm trong bất kỳ văn bản chính thức nào. Nên trước khi vào xây dựng chương trình thì cần xử lý dữ liệu thô là các câu hỏi từ người dùng.

Không chỉ việc xử lý dữ liệu thô là các câu hỏi từ người dùng đặt ra trong hệ thống, việc xử lý dữ liệu chô mục đích huấn luyện mô hình cũng cực kỳ quan trọng. Tất cả câu hỏi sẽ được loại bỏ những thứ không cần thiết như:

* Dấu câu [“.”, “?”, “!”, “:”, “…”]

Phương pháp: xây dựng một hàm có tên là Punctuation(string), thực hiện tạo một mảng bao gồm các dấu câu, cho lặp chuỗi cần xử lý trong mảng, nếu tồn tại dấu câu thì sẽ được thay thế bằng rỗng thông qua hàm “replace()”



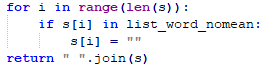


Hình 4: Xử lý dấu câu

* Các từ không quan trọng [“ạ”, “đi”, “em”, “thầy”, “ơi”, “e”, “cô”, “vậy”, “à”]

Phương pháp: xây dựng một hàm có tên là delete\_word(s) với s là một chuỗi, thực hiện tạo một mảng bao gồm các từ không quan trọng, lặp chuỗi “s” trong mảng, nếu có xuất hiện từ không quan trọng thì kết quả sẽ được thay thế bằng rỗng.

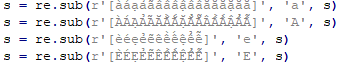




Hình 5: Xử lý các từ không quan trọng

* Thanh âm [“huyền”, “sắc”, “hỏi”, “ngã”, “nặng”]

Phương pháp: xây dựng một hàm xóa thanh âm, sử dụng thư việc “re” của ngôn ngữ lập trình Python và sử dụng các hàm “delete\_word(s)” và hàm “sub” có tác dụng thay thế các từ mang thanh âm thành từ không có thanh âm.



Hình 6: Xử lý thanh âm

### Ngôn ngữ lập trình Python

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1" \o "Guido van Rossum (trang không tồn tại)) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990" \o "1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o" \o "Trí tuệ nhân tạo).

### Ngôn ngữ lập trình Javascript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web.

Ví dụ: Khi sử dụng trong hệ thống Chatbot tư vấn tuyển sinh, những thao tác trao đổi với Chatbot giúp những đoạn văn bản được hiện lên chính là được xử lý bằng Javascript.

### HTML và CSS

* + - * 1. **HTML**

HTML là một ngôn ngữ đánh dấu được thiết kết ra để tạo nên các trang web, nghĩa là các mẩu thông tin được trình bày trên World Wide Web.

* + - * 1. **CSS**

CSS là định nghĩa về cách hiển thị của một tài liệu HTML. CSS đặc biệt hữu ích trong việc thiết kế web, giúp người thiết kế có thể dễ dàng triển khai các phong cách thiết kế lên bất kỳ trang nào của một website nhanh chóng và đồng bộ.

### Phương pháp học sâu (Deep Learning)

Học sâu là một nhánh máy học sử dụng nhiều lớp mạng Noron nhân tạo (Neurol network) để đưa ra một mô hình toán học dựa trên dữ liệu có sẵn. Học sâu hay Deep Learning[[14]](#footnote-14) thường được nhắc đến cùng với Dữ liệu lớn (Big Data[[15]](#footnote-15)) và Trí tuệ nhân tạo (AI[[16]](#footnote-16)) đang được ứng dụng ngày một rộng rãi hơn trong cuộc sống và xuất hiện hầu hết ở những lĩnh vực như: ý tế, dịch vụ, trong sản xuất, và được ứng dụng trong cả giảng dạy và rất nhiều lĩnh vực khác.



Hình 7: Mô hình mạng noron nhân tạo

Các kiến trúc mô hình học sâu khác nhau như: mạng noron sâu, mã mạng noron tích chập sâu, mạng niềm tin sâu và mạng noron tái phát đã được áp dụng cho các lĩnh vực như: thị giác máy tính, tự động nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên, nhận dạng âm thanh ngôn nữ và tin sinh học,… các mô hình học sâu này đã chứng minh là tạo ra được các kết quả rất tốt đối với nhiều nhiệm vụ khác nhau.

Sử dụng phương pháp mạng noron khi huấn liệu dựa trên tập dữ liệu, mạng noron sẽ khởi tạo các tập trọng số kí hiệu là weight. Những trọng số này sẽ được tối ưu trong suốt thời gian huấn luyện mô hình, và sẽ tạo ra các trọng số tối ưu hơn. Tổng quát quá trình xử lý một noron bao gồm các bước như sau:

Bước 1:

Tính tổng các trọng số đầu vào (input) với công thức như sau [4]:

Tổng các trọng số sẽ được tính theo công thức sau:

Bước 2:

Giá trị tính được đưa vào hàm kích hoạt để đưa ra kết quả cho đầu ra (output):

*Kí hiệu:*

* input: giá trị đầu vào
* output: giá trị đầu ra
* x1: giá trị đầu vào
* w1, w2, wn: trọng số đầu vào
* bias: độ lệch

Để tính giá trị đầu ra ta cần hai giá trị là trọng số và bias. Bias là độ lệch, là một giá trị không đổi, được thêm vào tổng các tích giữa các giá trị đầu vào và trọng số, chỉ số bias được sử dụng để chuyển đổi kết quả của chức năng kích hoạt về mặt tích cực hoặc tiêu cực.

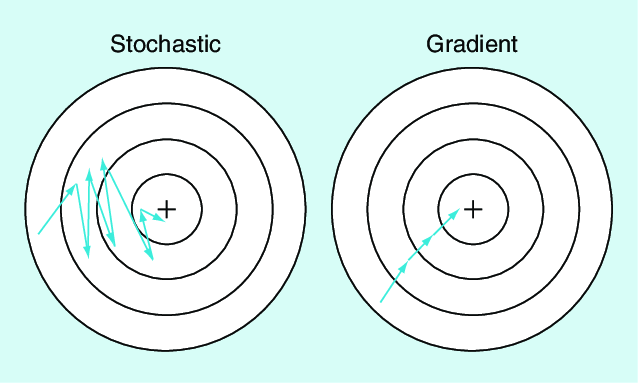
Ví dụ người dùng muốn mạng noron trả về 3 khi đầu vào vào là 0. Chúng ta có tổng trọng số đầu vào bằng 0, vậy câu hỏi đặt ra là làm thế nào để đảm bảo noron của mạng sẽ trả về giá trị bằng 3. Để giải quyết vấn đề này, chỉ cần đặt chỉ số bias là 3.

Nếu chúng ta thêm giá trị bias thì mạng noron nhân tạo chỉ thực hiện một công việc đơn giản là thực hiện phép nhân ma trận trên các đầu vào trọng số, việc này sẽ tạo nên tập dữ liệu có thể bị overfitting[[17]](#footnote-17).

#### Optimizer SGD (Stochastic Gradient Descent)

Trước khi nói đến SGD[[18]](#footnote-18), việc tìm hiểu về Optimizer cũng rất quan trọng. Đây là thuật toán tối ưu. Về cơ bản, thuật toán tối ưu là cơ sở để xây dựng mô hình Neural Network với mục đích “học” được các “pattern” của dữ liệu đầu vào, từ đó có thể tìm được một cặp trọng số (được gọi là weights) và bias phù hợp để tối ưu hóa mô hình. Phương pháp tìm weights và bias đó là lấy giá trị ngẫu nhiên weights và bias một số lần hữu hạn và hy vọng tại một bước nào đó ta có thể tìm được lời giải cho bài toán. Nếu ta thực hiện theo phương pháp như vậy, là một điều không khả dĩ và lãng phí tài nguyên nghiên cứu cũng như thời gian. Vấn đề đặt ra là cần tìm một giải pháp để cải thiện trọng số weights và bias theo từng bước, đó là lí do các thuật toán optimizer ra đời.

SGD là một biến thể GD (Gradient Descent) và GD còn nhiều hạn chế như phụ thuộc vào giá trị nghiệm khởi tạo ban đầu và tốc độ học của mô hình. Khi tốc độ học quá lớn sẽ khiến cho thuật toán không hội tụ. Thay vì mỗi vòng lặp (được gọi là epoch) chúng ta sẽ cập nhật giá trị trọng số một lần. Đối với GD thì trong mỗi vòng lặp có N điểm dữ liệu, chúng ta sẽ cập nhật trọng số N lần.



Hình 8: So sánh hội tụ giữa SGD và GD

Trong hinh chúng ta có thể thấy SGD có đường đi đi theo hình zíc-zắc, không mượt như GD. Vậy câu hỏi đặt ra “tại sao phải dùng SGD thay vì dùng GD?”. Bởi vì GD có hạn chế đối với cơ sở dữ liệu lớn, việc tính toán đạo hàm toàn bộ dữ liệu qua mỗi vòng lặp trở nên cồng kềnh. Bên cạnh đó, GD không phù hợp với online learning, khi dữ liệu cập nhật liên tục (ví dụ khi thêm câu hỏi) thì mỗi lần thêm dữ liệu ta phải tính toán lại đạo hàm trên toàn bộ dữ liệu, về mặt này SGD có thể giải quyết dễ dàng vấn đề đó. Ngoài ra, thuật toán SGD có một nhược điểm lớn là về tốc độ học, điểm dữ liệu ban đầu, vì vậy ta phải kết hợp SGD và một số thuật toán khác như Momentum….

#### Hàm kích hoạt (Activation Functions)

Hàm kích hoạt là những hàm mô phỏng tỷ lệ truyền xung qua axon của một neuron thần kinh. Trong một mạng nơ-ron nhân tạo, hàm kích hoạt đóng vai trò là thành phần phi tuyến tại output của các nơ-ron. Nếu không có các hàm kích hoạt, khả năng dự đoán của mạng nơ-ron sẽ bị giới hạn và giảm đi rất nhiều, sự kết hợp của các hàm kích hoạt giữa các tầng ẩn sẽ giúp cho mô hình học được các quan hệ phi tuyến phức tạp tiềm ẩn trong dữ liệu.

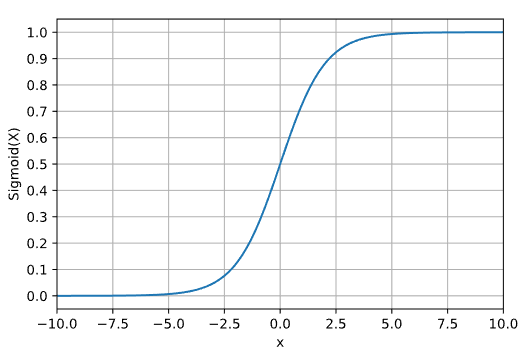
**Một số hàm kích hoạt phổ biến:**

*Hàm sigmoid*

Công thức:

Đạo hàm:

Đồ thị:

****

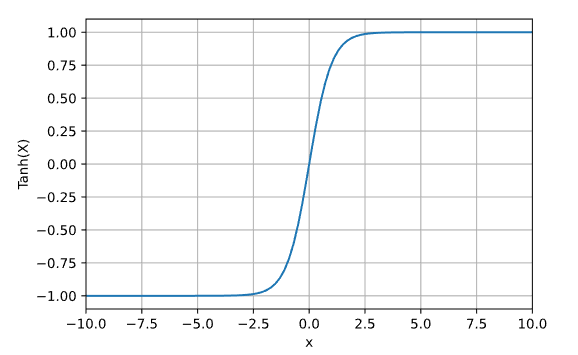
Hình 9: Đồ thị Hàm Sigmoid

Đầu vào của hàm Sigmoid là một số thực và chuyển thành một giá trị nằm trong khoảng (0; 1). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại nếu đầu vào là một số thực dương lớn sẽ cho đầu ra tiệm cận với 1.

*Hàm Tanh*

Công thức:

Đồ thị:



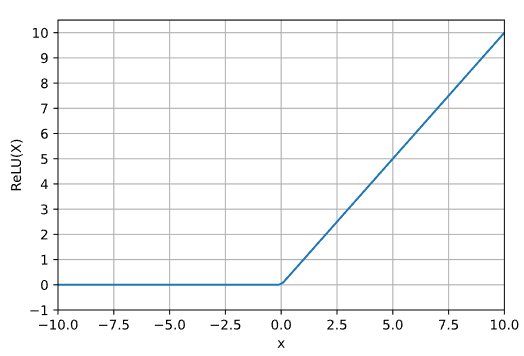
Hình 10: Đồ thị hàm Tanh

Đầu vào của hàm Tanh là một số thực và chuyển thành một giá trị nằm trong khoảng (-1; 1). Hàm Tanh bị bão hòa của hai đầu (gradient thay đổi rất nhỏ ở hai đầu).

*Hàm ReLU (Rectified Linear Unit)*

Công thức:

Đồ thị:



Hình 11: Đồ thị hàm ReLU

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng nơ-ron. ReLU đơn giản là lọc các giá trị nhỏ hơn 0. Một số ưu điểm khá vượt trội của nó so với Signmoid và Tanh là tốc độ hội tụ nhanh hơn hẳn, điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sighmoid và Tanh. Tính toán của ReLU nhanh hơn do công thức không phức tạp.

*Hàm Softmax*

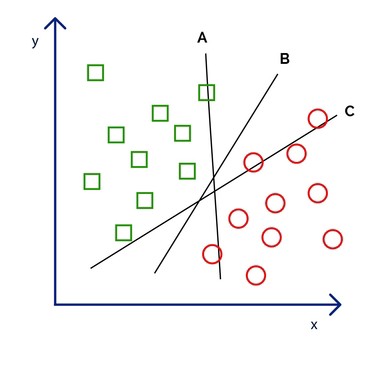
Công thức:

Hàm Softmax xuất ra một véc-tơ có giá trị tổng bằng 1 có thể hiểu là xác suất của các lớp. Softmax thường dùng ở đầu ra cho bài toán phân lớp, giả sử bài toán gồm nhiều lớp khi dùng Softmax kết quả trả về là xác suất của từng lớp.

### Máy học véc-tơ hỗ trợ (Support Vector Machine)

Máy học véc-tơ hỗ trợ là tập hợp các phương pháp học có giám sát liên quan đến nhau đề phân loại và phân tích hồi quy. SVM dạng chuẩn chấp nhận dữ liệu vào và phân loại dữ liệu và hai lớp khác nhau. Support Vectors được hiểu một các đơn giản là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới chia cắt các lớp tốt nhất.

Trong SVM điều quan trọng là cần xác định đúng đường Hyper-plane. Hyper-plane là một đường thẳng có tác dụng phân chia các lớp ra thành các đường riêng biệt.



Hình 12: Các đường phân chia SVM

Theo hình minh họa, tại đây có 3 đường hyper-plane (đường A, đường B và đường C). Theo như quy tắc số 1 về chọn lựa đường hyper-plane, chọn một hyper-plane phân chia hai lớp tốt nhấ, trong ví dụ này ta có:

* Đường A: Xuất hiện việc cắt ngang các phân tử tại hai hớp
* Đường C: Xuất hiện việc cắt ngang các phần tử tại hai lớp
* Đường B: Phân chia hai lớp tốt nhất, không cắt qua bất kỳ phần tử nào

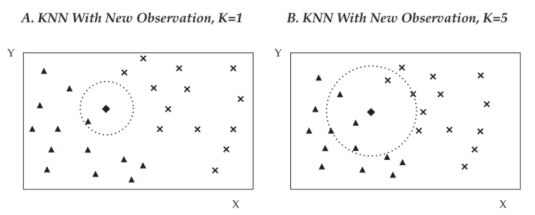
SVM là một thuật toán phân lớp hoạt động tốt với những tập dữ liệu có kích thước lớn, thường mang lại kết quả vượt trội so với các giải thuật học có giám sát khác. Một số ưu điểm của thuật toán SVM:

* Ưu điểm 1: Hoạt động hiệu quả với không gian cao chiều (high dismensional spaces): SVM là công cụ tính toán hiệu quả trong không gian cao chiều, trong đó đặc biệt áp dụng cho bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.
* Ưu điểm 2: Thuật toán tiêu tốn ít bộ nhớ vì chỉ sử dụng các điểm trong tập hỗ trợ để dự báo trong hàm quyết định.
* Ưu điểm 3: Có thể tạo ra nhiều hàm quyết định từ những hàm kernel khác nhau. Khả năng áp dụng kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó cho hiệu xuất phân loại tốt hơn.

SVM là một phương pháp hiệu quả cho bài toán phân lớp dữ liệu. Nó là công cụ đắc lực cho các bài toán phân loại văn bản, phân tích quan điểm, SVM cũng từng là mô hình cực kỳ phổ biến trong phân loại ảnh trước khi CNN[[19]](#footnote-19) và Deep Learning bùng nổ. Một yếu tố khác làm nên hiệu quả của SVM đó là thuật toán này sử dụng Kernel function khiến cho các phương pháp chuyển không gian trở nên linh hoạt hơn.

### K láng giềng

Thuật toán K láng giềng [[20]](#footnote-20)là một kỹ thuật học có giám sát, dùng để phân loại quan sát mới bằng tìm kiếm tương đồng giữa quan sát mới với dữ liệu sẵn có.



Hình 13: kNN với k=1 và k=5

Chúng ta cần xác định rõ hình thoi trong cả hai ví dụ đang cần được phân loại và hình dấu nhân hoặc hình tam giác. Hình thoi sẽ được phân loại vào cùng loại với điểm dữ liệu gần nhất:

* Xét k = 1: Hình thoi tại thời điểm k=1 sẽ được phân loại vào cùng loại với hình tam giác. Vì đây có một điểm dữ liệu hình tam giác nằm gần hình thoi nhất về phía bên trái.
* Xét k = 5: Tại thời điểm này, thuật toán kNN sẽ xét 5 điểm dữ liệu gần với hình thoi nhất. Trong đó số lượng hình tam giác gần hình thoi là 3 hình, còn số lượng hình dấu nhân là 2 hình. Dựa theo quy tắc, hình thơi sẽ được phân loại và cùng loại với hình có số điểm dữ liệu lớn hơn trên tổng số 5 điểm dữ liệu gần nhất với hình thoi. Trong trường hợp này, hình thoi được phân loại và hình tam giác.

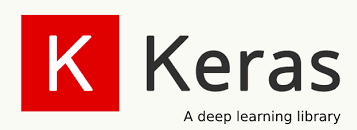
KNN là một mô hình đơn giản, trực quan nhưng vẫn có hiệu quả rất cao vì kNN không cần tham số, mô hình không đưa ra bất kỳ giả định nào về việc phân phối dữ liệu, và còn có thể được sử dụng trực tiếp để phân loại đa lớp.

Thuật toán kNN có rất nhiều ứng dụng trong ngành đầu tư, bao gồm: các dự án phá sản, dự đoán đánh giá cổ phiếu, phân bố xếp hạng tín dụng trái phiếu doanh nghiệp.

Khi sử dụng kNN cần phải xác định thước đo để tính khoảng cách giữa các đối tượng cần phân lớp và các đối tượng còn lại trong cơ sở dữ liệu. Vì đây là lựa chọn mang tính chủ quan, nếu chojnc một thước đo không phù hợp thì mô hình sẽ không hiệu quả.

### Thư viện Keras

Keras[[21]](#footnote-21) là một thư viện được phát triển vào năm 2015 bởi Francois Chollet, một kĩ sư nghiên cứu Deep Learning. Keras là một Open Source cho Nerual Network được viết bởi ngôn ngữ Python. Và nó được xem là một API bậc cao có thể sử dụng chung với các thư viện Deep Learning nổi tiếng như Tensorflow[[22]](#footnote-22) (được phát triển bởi Google), CNTK[[23]](#footnote-23) (được phát triển bởi Microsoft)...



Hình 14: Thư viện Keras

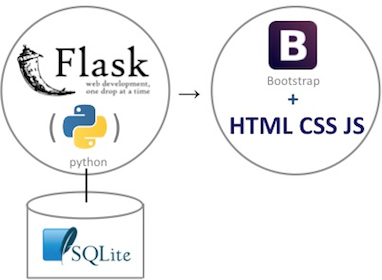
Keras là API cấp cao của TensorFlow 2: một giao diện dễ tiếp cận, hiệu quả cao để giải quyết các vấn đề về máy học, tập trung vào học sâu hiện đại. Nó cung cấp các khối xây dựng và trừu tượng cần thiết để phát triển và vận chuyển các giải pháp máy học với tốc độ lặp lại cao.

Keras cấp quyền cho các kỹ sư và nhà nghiên cứu để tận dụng tối đa khả năng mở rộng và khả năng đa nền tảng của TensorFlow 2: có thể chạy Keras trên TPU hoặc trên các cụm GPU lớn và người sử dụng có thể xuất các mô hình Keras của mình để chạy trong trình duyệt hoặc trên thiết bị di động thiết bị.

### Framework Flask

Flask[[24]](#footnote-24) là một web Framework được phát triển bằng ngôn ngữ lập trình Python, không yêu cầu tool hay thư viện cụ thể nào. Flask có hỗ trợ các thành phần tiện ích mở rộng cho ứng dụng như tích hợp: cơ sở dữ liệu, xác thực biểu mẫu, xử lý upload, các công nghệ xác thực, Template, Email, RESTful,…..phụ thuộc vào nhu cầu sử dụng mà người dùng có thể chọn sử dụng tiện ích nào. Người dùng có thể tập trung xây dựng một Web Application ngay từ khi bắt đầu dự án, trong một khoảng thời gian ngắn, theo tiến độ phát triển của dự án mà có thể phát triển ứng dụng theo bất kỳ yêu cầu nào đặt ra.

Để thuận tiện cho người dùng, Flask cung cấp rất nhiều tài liệu hướng dẫn, từ việc cài đặt, thực thi và triển khai, trong đó có cả hướng dẫn nhanh và hướng dẫn chi tiết. Người dùng có thể dễ dàng tìm kiếm những tài liệu này để tham khảo, học tập về ứng dụng web và thực hành cài đặt Flask Framework miễn phí trên internet.



Hình 15: Mô hình ứng dụng web với Flask Framework

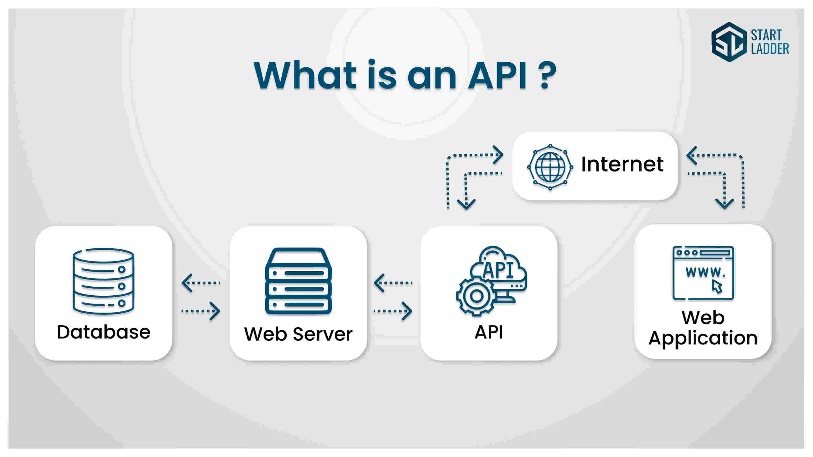
### API

API[[25]](#footnote-25) là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. API là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

Web API là hệ thống API được sử dụng trong hệ thống website. Đa số các website đều được ứng dụng đến web API cho phép người dùng kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu.

Ví dụ: Người thiết kế chức năng login vào Google, Facebook, Twitter, Github,… điều này có nghĩa là đang gọi đến API của máy chủ các dịch vụ tương ứng, hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API. Nhóm nghiên cứu đã có một đề tài sử dụng API để lấy dữ liệu từ một ứng dụng du lịch là Booking[[26]](#footnote-26), việc lấy dữ liệu thông qua API sẽ thuận lợi hơn việc cào dữ liệu trực tiếp từ trang web, dữ liệu dược lấy về sẽ được lưu trữ dưới dạng tập tin .json.

Phương thức Web API cho phép các ứng dụng khác nhau có thể giao tiếp dữ liệu qua lại. Dự liệu được web API trả lại thường ở dạng JSON hoặc XML thông qua các giao thức HTTP hoặc HTTPS.



Hình 16: Mô hình web API

### Hệ quản trị cơ sở dữ liệu

Trong nghiên cứu này, tập dữ liệu được lưu trữ dưới dạng tập tin .json và những câu hỏi được thu thập sẽ được lưu trữ và sử dụng bằng hệ cơ sở dữ liệu SQLite[[27]](#footnote-27).

SQLite là một hệ quản trị cơ sở dữ liệu (DBMS) quan hê, tương tự như MySQL, Oracle, PostgreSQL,…Đặc điểm nổi bật nhất của SQLite so với các DBMS khác là gọn, nhẹ, đơn giản, đặc biệt là không cần mô hình client – server, không cần cài đặt, cấu hình hay khởi động nên sẽ không cần user, password hay quyền bên trong SQLite Database, dữ liệu sẽ được lưu ở một file duy nhất.

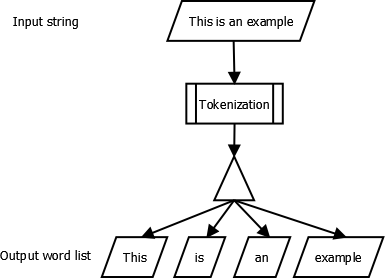


Hình 17: Hệ quản trị cơ sở dữ liệu SQLite

SQLite không yêu cầu một quy trình hoặc hệ thống máy chủ riêng biệt để hoạt động và không cần cầu hình nghĩa là không cần thiết lập hoặc quản trị, SQLite rất nhỏ và trọng lượng nhẹ, dưới 400KiB được cấu hình đầy đủ. SQLite hỗ trợ hầu hết các tính năng ngôn ngữ truy vấn được tìm thấy trong tiêu chuẩn SQL92.

### Word Tokenization

Word Tokenization – Mã hóa từ là quá trình chia nhỏ một mẫu văn bản lớn thành các từ. Đây là một yêu cầu trong các nhiệm vụ xử lý ngôn ngữ tự nhiên, nơi mỗi từ cần được nắm bắt và phân tích sâu hơn như phân loại và đếm chúng cho một cảm xúc cụ thể,… bộ Công cụ Ngôn ngữ Tự nhiên là một thư viện được sử dụng để đạt được điều này. Cài đặt NLTK trong Python trước khi sử dụng để mã hóa từ.



Hình 18: Mã hóa từ trong Python

## Mô tả hệ thống trang web

Trang web được thiết kế theo hình thức hỏi - đáp, tương tự với những ứng dụng như messerger, zalo, whatapp,… tổng thể của trang web là một ô cửa sổ chat, tập trung vào thể hiện sự trao đổi giữa người dùng và Chatbot. Người dùng có thể đặt câu hỏi về ngành, nhóm ngành Công nghệ Thông tin của Trường Công nghệ Thông tin và Truyền thông, Chatbot sẽ căn cứ vào dữ liệu để trả lời cho câu hỏi của người dùng.



Hình 19: Giao diện ứng dụng website

## Yêu cầu chức năng

### Người dùng admin

Người dùng admin có những chức năng sau:

* Quyền quản trị
* Thêm, xóa, sửa câu hỏi và câu trả lời.
* Xem lịch sử chat của người dùng và chatbot
* Cập nhật, thêm tính năng cho hệ thống

### Người dùng

Người dùng có những chức năng sau:

* Hỏi đáp, trao đổi thông tin với chatbot
* Liên hệ với admin thông qua địa chỉ mail, số điện thoại,…

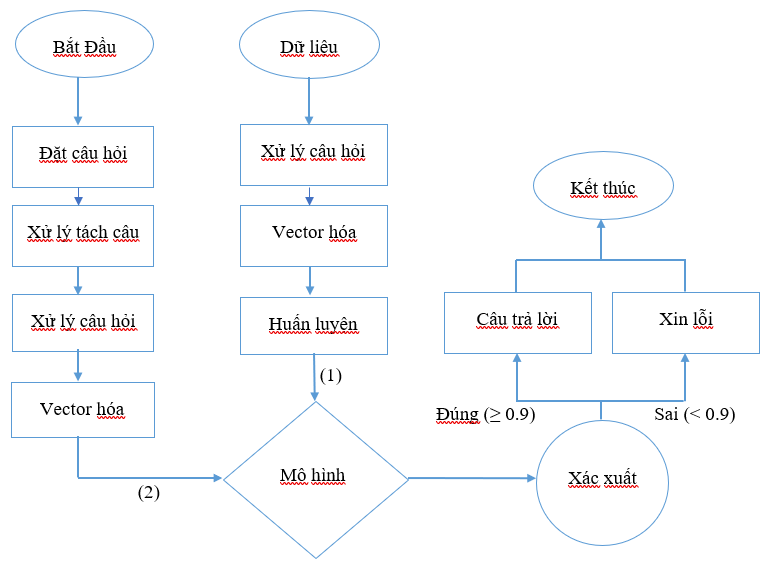
# HỆ THỐNG HỖ TRỢ

Chương 3 sẽ giới thiệu về tổng quan về hệ thống chatbot với framework RASA, phương pháp xử lý dữ liệu và huấn luyện mô hình trên ba phương pháp học máy là SVM, K láng giềng, Bayes Thơ ngây, SGD

## Chatbot

Chatbot[[28]](#footnote-28) là một ứng dụng phần mềm dùng để quản lý tạo cuộc hội thoại trực tuyến bằng văn bản hoặc văn bản chuyển thành giọng nói, thay vì cung cấp các cuộc hội thoại trực tiếp với người dùng có thật.

## Thiết kế hệ thống



Hình 20: Lưu đồ hoạt đông của hệ thống

Quá trình xử lý của hệ thống sẽ chia ra làm 2 giai đoạn:

Giai đoạn (1): Xử lý tập dữ liệu, dùng để tạo mô hình dự đoán. Bao gồm các quá trình:

Xử lý câu hỏi ⭢ Vector hóa ⭢ Huấn luyện

Tập dữ liệu sau khi được thu thập, sẽ trải qua quá trình xử lý câu hỏi, sau đó sẽ vector hóa dữ liệu đã được xử lý, tạo ra mạng vector. Sau đó đưa mảng vector vừa được tạo ra vào mô hình mạng noron 2 lớp để huấn luyện tạo ra Mô hình.

Giai đoạn (2): Xử lý tập câu hỏi người dùng đưa ra để dự đoán. Bao gồm các quá trình:

Tách câu hỏi ⭢ xử lý câu hỏi ⭢ vector hóa ⭢ Mô hình dự báo

Quá trình bắt đầu từ việc nhận được câu hỏi của ngưởi sử dùng, câu hỏi sẽ được tách thành các câu hỏi nhỏ, và được xử lý câu hỏi, sau đó sẽ vector hóa câu hỏi thành mảng vector. Mảng vector sau khi xử lý sẽ đựa đưa vào mô hình dựa đoán. Kết quả sẽ là xác xuất dự đoán. Tại đây, hệ thống sẽ chia ra hai trường hợp:

Trường hợp 1: Kết quả dự báo có xác xuất lớn hơn hoặc bằng 0.9 (≥ 90%): Kết quả trả về sẽ là câu trả lời cho câu hỏi của người dùng.

Trường hợp 2: Kết quả dự báo có xác xuất bé hơn 0.9 (< 90%): Kết quả trả về sẽ là câu “Xin lỗi, hệ thống không thể hồi đáp”.

### Tập dữ liệu

Để thực hiện huấn luyện mô hình chatbot, tôi thực hiện thu thập dữ liệu dùng để huấn luyện mô hình từ trang Fanpage ‘Tư vấn tuyển sinh chính quy[[29]](#footnote-29)” của trường Đại Học Cần Thơ, và dữ liệu cũng được thu thập thông qua việc khảo sát trên không gian mạng thông qua sự hỗ trợ của Google Form. Tại các buổi tuyển sinh trực tiếp, trực tuyến của trường Đại Học Cần Thơ để tư vấn cho các người sử dụng học sinh, các bậc phụ huynh về thông tin tuyển sinh của trường. Quý phụ huynh và các người sử dụng học sinh đã có những câu hỏi, thắc mắc của mình về những ngành học và chương trình đạo tạo của Trường Đại học Cần Thơ nói chung và trường Công nghệ Thông tin và Truyền thông nói riêng.

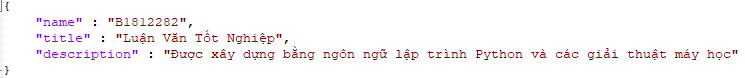
Sau khi thu thập dữ liệu về các câu hỏi của các người sử dụng học sinh và các bậc phụ huynh, những câu hỏi sẽ được thực hiện phân loại. Thủ thuật gom nhóm những câu hỏi có cùng câu trả lời sẽ tạo thành một nhóm và những câu hỏi thu thập vẫn chưa được xử lý.

Tổng quan về tập dữ liệu như sau:

* Số lượng câu hỏi được thu thập để tạo thành tập dữ liệu: 1043 câu hỏi.
* Số lượng nhãn hiện tại bao gồm: 25 nhãn.
* Tập dữ liệu được lưu trữ với cấu trúc là một file dạng .json

Trước khi mô tả rõ hơn về tập dữ liệu, cần tìm hiểu trước về JSON và cấu trúc của một file JSON.

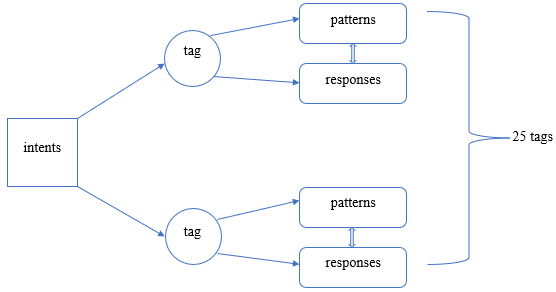
JSON là viết tắt của JavaScript Object Notation, là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được .json là một tiêu chuẩn mở để trao đổi dữ liệu trên web. Cấu trúc của một file .json là theo dạng key – value để dữ liệu sử dung. Hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng. Ví dụ một tập tin infor.json với nội dung dưới đây sử dụng format kiểu JSON để lưu trữ thông tin:



Hình 21: Nội dung tập tin infor.json

Tập dữ liệu dùng trong nghiên cứu có tên là data.json, có 1043 câu hỏi được chia vào 25 nhãn. Cấu trúc của tập dữ liệu được mô tả như sau:

* intents: dùng để gọi tập dữ liệu trong quá trình lập trình
* tag: đóng vai trò là nhãn
* patterns: dùng để lưu trữ tất cả câu hỏi thuộc về nhãn hiện hành
* responses: dùng để lưu trữ tất cả câu trả lời ứng với những câu hỏi nằm trong nhãn hiện hành.



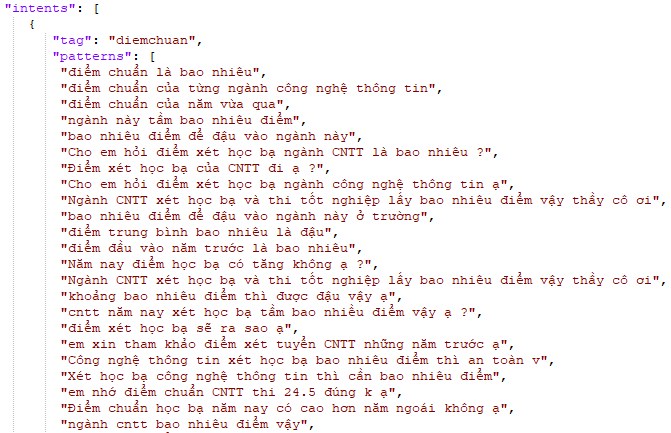
Hình 22: Cấu trúc tập tin data.json

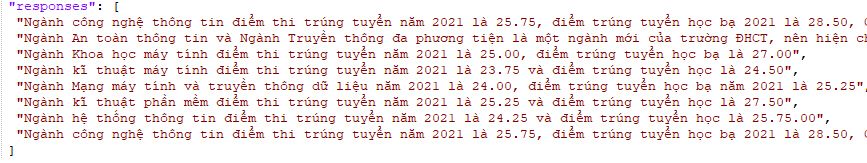
Với 25 nhãn trong tập dữ liệu bao gồm các nhãn như: diemchuan, phuongthuc, chitieu, tilevieclam, tiletonghiep,… Tùy theo nhu cầu của người sử dụng mà số lượng câu hỏi nằm trong trường “patterns” của các nhãn có sự phân hóa rõ ràng. Ví dụ tại nhãn “diemchuan” có số lượng câu hỏi là 176 câu hỏi, nhãn “phuongthuc” có số lượng câu hỏi là 174 câu hỏi, nhưng nhãn “kytucxa” có số lượng câu hỏi khá hạn chế là 11 câu hỏi. Điều này phản ánh rõ mức độ quan tâm của người sử dụng vào những yếu tố như: điểm chuẩn, phương thức xét tuyển, chỉ tiêu,… trong quá trình tuyển sinh.

Đối với mức độ dày đặc của câu hỏi thì phần thiết kế câu trả lời nằm trong trường “patterns” cũng sẽ dựa trên số lượng câu hỏi hiện hành. Ví dụ với nhãn “diemchuan” có 176 câu hỏi – 9 câu trả lời, nhãn “phuongthuc” có 174 câu hỏi – 9 câu trả lời, nhãn “kytucxa” có 11 câu hỏi – 2 câu trả lời.

Trong từng nhãn – trường “tag” bên trong tập dữ liệu, số lượng câu trả lời có nội dung tương tự nhau sẽ được gói vào một nhóm và đưa vào trường “patterns”, số lượng câu hỏi không cố định và không có giới hạn cụ thể, càng nhiều câu hỏi thì dữ liệu dùng cho huấn luyện sẽ càng phong phú hơn. Những câu trả lời tương ứng với các câu hỏi bên trong trường “patterns” sẽ được người quản trị trả lời dựa trên những thông tin thu thập được trong quá trình thu thập dữ liệu, tất cả câu trả lời sẽ được chọn lọc và xử lý những lỗi chính tả, sau đó sắp xếp câu trả lời theo 1 thứ tự nhất định ứng với thứ tự các nhóm câu hỏi hướng tới cùng một câu trả lời để dễ dàng rà xoát và xử lý các lỗi phát sinh trong quá trình thử nghiệm, thực hiện đưa tất cả câu hỏi và trường “responses” để thực hiện huấn luyện mô hình.

Minh họa cho tập dữ liệu với nhãn “diemchuan”

****

****

Hình 23: Cấu trúc tập dữ liệu đối với nhãn diemchuan

|  |  |
| --- | --- |
| Tên trường | Chức năng |
| “tag” | Nhãn của một tập hợp câu hỏi |
| “patterns” | Chứa tập hợp câu hỏi có chung nhãn |
| “responses” | Chứa câu trả lời cho một nhãn |

Bảng 3: Chức năng các trường trong tập dữ liệu huấn luyện

Trong một nhãn gồm nhiều câu hỏi, mỗi một câu hỏi lại có một câu trả lời riêng biệt, ví dụ khi người sử dung hỏi về “điểm chuẩn ngành trường học máy tính” và “điểm chuẩn ngành kỹ thuật phần mềm”, hai câu hỏi này hiện đang xếp cùng nhãn đó là nhãn “diemchuan” nhưng người sử dụng lại cần hai câu trả lời khác nhau nên trong trường “responses” được dùng để lưu trữ câu trả lời của các nhóm ngành khác nhau và được sắp xếp theo một thứ tự cố định là:

|  |  |
| --- | --- |
| STT | Tên ngành |
| 0 | Công nghệ thông tin |
| 0 | Công nghệ thông tin chất lượng cao |
| 1 | An toàn thông tin |
| 2 | Truyền thông đa phương tiện |
| 3 | Trường học máy tính |
| 4 | Kỹ thuật máy tính |
| 5 | Mạng máy tính và Truyền thông dữ liệu |
| 6 | Kỹ thuật phần mềm |
| 7 | Hệ thống thông tin |

Bảng 4: Số thứ tự các ngành

Thứ tự câu trả lời của các ngành được sắp xếp tương tự theo danh sách các ngành về “Máy tính và công nghệ thông tin” trong trang web tuyển sinh Đại học Cần Thơ, có một sự thay đổi về vị trí của ba ngành: Công nghệ thông tin, Công nghệ thông tin chất lượng cao và Công nghệ thông tin học tại khu Hòa An, ba ngành này sẽ mang số thứ tự là 0, và có cùng 1 câu trả lời chung về ngành Công nghê thông tin.



Hình 24: Danh sách các ngành Máy tính và Công nghệ thông tin

(chương trình đại trà)

Ví dụ về nhãn “diemchuan” chúng ta có các câu hỏi liên quan đến điểm chuẩn như “điểm chuẩn là bao nhiêu”, “điểm chuẩn ngành trường học máy tính là bao nhiêu”. Với những câu hỏi liên quan đến điểm chuẩn ta sẽ thiết kế câu trả lời liên quan đến điểm chuẩn là: “Ngành Trường học máy tính điểm trúng tuyển 2021 là 25.00, điểm thi trúng tuyển học bạ năm 2021 là 27.00”, thực hiện tương với các câu hỏi và câu trả lời đối với các ngành còn lại trong nhãn “diemchuan”.

Các nhãn khác như “phuongthuc”, “chitieu”, “vitrivieclam”,… cũng thực hiện sắp xếp và thiết kế các câu hỏi trong trường “patterns” và câu trả lời trong trường “responses” tương tự như nhãn “diemchuan”.

### Tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu được thực hiện qua các công đoạn. Đối với tập câu hỏi sau khi thu thập sẽ được chuyển hóa tất cả thành chữ thường, xử lý thành những câu hỏi không dấu, các từ viết tắt và những chữ đặc biệt sẽ được xử lý về đúng nghĩa của nó, thực hiện lược bỏ những từ ngữ không cần thiết trong quá trình huấn luyện mô hình.

#### Loại bỏ các từ kéo dài

Các câu hỏi có thể xen lẫn giữa chữ hoa và chữ thường nên cần đưa hết về dạng chữ thường. Vì câu hỏi có thể có những ký tự cố tình viết kéo thì xử lý loại bỏ: Ví dụ như “được khôngggg” ? “cảm ơnnnnn”…



Hình 25: Loại bỏ các ký tực kéo dài

#### Xử lý dấu câu

Dữ liệu câu hỏi sau khi thu thập và phân loại sẽ được loại bỏ dấu câu trong Tiếng Việt như [“.”, “?”, “!”, “:”, “…”], các thanh âm của câu hỏi cũng cần được loại bỏ là: [“huyền”, “sắc”, “hỏi”, “ngã”, “nặng”]. Việc bỏ dấu câu giúp tiết kiệm thời gian cũng như không gian lưu trữ, và lược bỏ thanh âm là cần thiết vì khi ta không lượt bỏ thanh âm dẫn đến từ có dấu và từ không dấu sẽ khác nhau ví dụ như “điểm” và “diem” sẽ khác nhau trong quá trình huấn luyện làm ảnh hưởng rất lớn đến độ chính xác của bài toán.

#### Loại bỏ từ vô nghĩa và từ viết tắt

Ngoài việc loại bỏ các thanh âm, các từ viết tắt đặc biệt như “cntt, khmt, attt, ktpm, httt” được chuyển về đúng ngữ pháp và loại bỏ các từ không quan trọng trong quá trình huấn luyện như “ạ, vậy ạ, sao, em, ơi”.

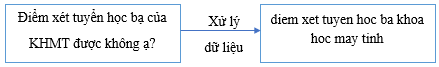
Danh mục các từ viết tắt cần xử lý.

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Từ chuyển đổi |
| 1 | cntt | cong nghe thong tin |
| 2 | attt | an toan thong tin |
| 3 | ttdpt | truyen thong da phuong tien |
| 4 | khmt | trường hoc may tinh |
| 5 | ktmt | ky thuat may tin |
| 6 | mmt | mang may tinh |
| 7 | ktpm | ky thuat phan mem |
| 8 | httt | he thong thong tin |
| 9 | clc | chat luong cao |
| 10 | ttdl | truyen thong du lieu |
| 11 | bn | bao nhieu |
| 12 | ko | khong |
| 13 | ntn | nhu the nao |
| 14 | ktx | ky tuc xa |
| 15 | nhiu | nhieu |
| 16 | hc | hoc |
| 17 | dc | duoc |
| 18 | nghanh | nganh |

Bảng 5: Danh mục các từ viết tắt

Sau khi tiền xử lý dữ liệu kết quả sẽ đưa về một bộ câu hỏi đáp ứng các tiêu chí đề ra ban đầu:

* Chuyển hóa tất cả thành chữ in thường
* Câu hỏi không mang dấu câu
* Xử lý các từ viết tắt, những ký tự đặc biệt
* Loại bỏ các từ ngữ không cần thiết



Hình 26: Chuyển đổi dữ liệu

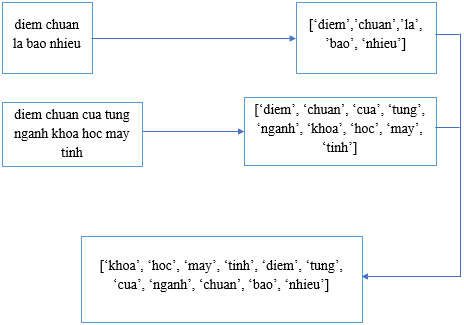
### Xây dựng dữ liệu huấn luyện

Sau khi thực hiện tiền xử lý dữ liệu, tất cả các câu hỏi trong tập dữ liệu sẽ được tách thành các từ thêm vào biến “words”. Tất cả các từ sẽ được sắp xếp và đảm bảo các từ trong đó là tồn tại duy nhất để tạo thành một vector tập hợp tất cả các từ trong câu hỏi.

#### Xử lý tách câu thành các mảng từ

Quá trình xử lý tách câu thành các mảng từ sử dụng hàm của thư viện NLTK là word\_tokenize[[30]](#footnote-30) có dùng để phân tách các câu thành những mảng từ.

Mô hình xử lý tách những câu hỏi thành mảng các từ:

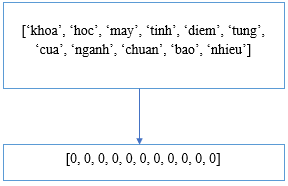


Hình 27: Xử lý câu hỏi tạo tập hợp từ

#### Tạo vector

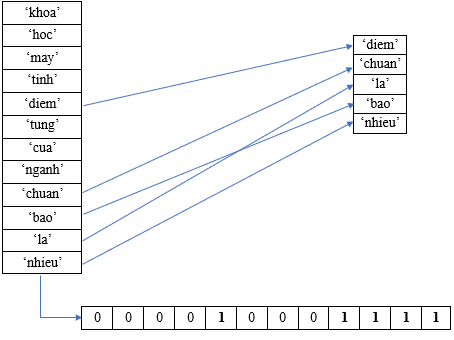
Sau khi tạo được tập hợp những mảng bao gồm tất cả các từ, tiếp tục xử lý khởi tạo một mảng với độ dài bằng với độ dài mảng tập hợp các từ trong tập dữ liệu với giá trị trong mảng khởi tạo là 0.

Khởi tạo mảng:



Hình 28: Tạo vector ứng với độ dài tập hợp mảng từ

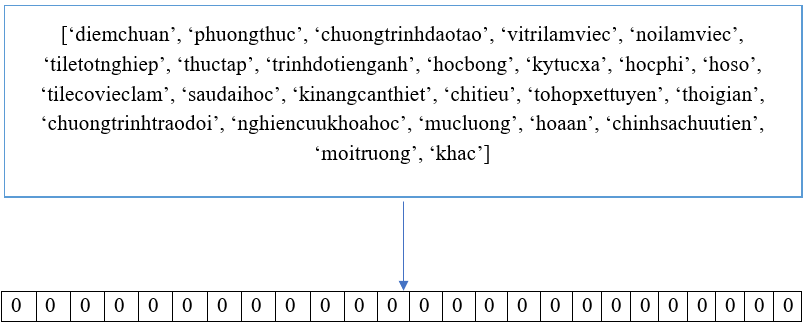
Sau khi tạo thành vector là mảng tất cả các từ trong tập dữ liệu, xét từng câu hỏi trong câu trong tập dữ liệu, so sánh với các từ trong vector với các từ trong câu hỏi nếu từ đó có trong câu hỏi sẽ được gán giá trị là 1, ngược lại nếu không tồn tại sẽ được gán giá trị là 0.



Hình 29: Chuyển đổi câu hỏi thành vector

Mỗi câu hỏi sẽ được chuyển đổi để tạo thành một vector, thực hiện quá trình chuyển đổi toàn bộ câu hỏi trong tập dữ liệu thành vector để thực hiện huấn luyện mô hình.

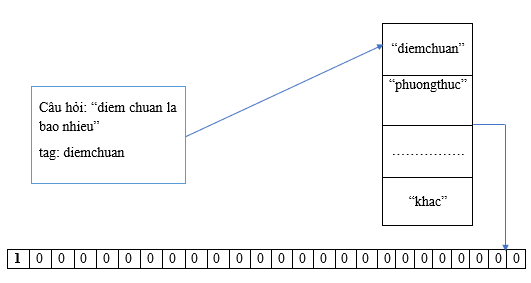
Cách tạo ra một vector là thực hiện việc so sánh các từ trong câu hỏi nằm bên trong tập dữ liệu sẽ được gán nhãn. Nhãn của tập dữ liệu huấn luyện là tập hợp các trường “tag” của tập dữ liệu. Thực hiện khởi tạo một mảng có độ dành bằng với số lượng nhãn trông tập dữ liệu có độ lớn bằng 0.



Hình 30: Tạo vector ứng với số lượng nhãn

Nếu câu hỏi có nhãn tương ứng với mảng trên vị trí nhãn sẽ là 1, ví dụ câu hỏi “điểm chuẩn là bao nhiêu” có “tag” là “diemchuan” nhãn của câu hỏi đó sẽ là:

[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]



Hình 31: Câu hỏi được chuyển hóa thành vector

### Xây dựng mô hình huấn luyện

Sau khi hoàn thành việc xây dựng vector câu hỏi và mảng các nhãn của câu hỏi tưởng ứng, và đưa tất qua mô hình huấn luyện.

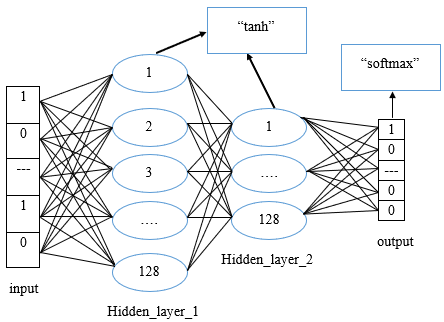
#### Xây dựng mô hình chatbot

Khi làm việc với dữ liệu văn bản, chúng ta cần thực hiện tiền xử lý trên dữ liệu trước khi chúng ta tạo ra một mô hình Machine-Learning hoặc Deep-Learning, Tokennizing là bước cơ bản nhất có thể làm trên dạng dữ liệu văn bản, đây là quá trình chia toàn bộ văn bản thành các phần nhỏ như các từ. Sau khi thực hiện chia nhỏ các từ, chúng ta sẽ bổ sung từng từ và xóa các từ trung lặp khỏi danh sách.

Tiếp theo, chúng ta cần tạo một tập training bao gồm các input và output:

* Input: là các mẫu (câu hỏi)
* Output: là các câu trả lời tương ứng

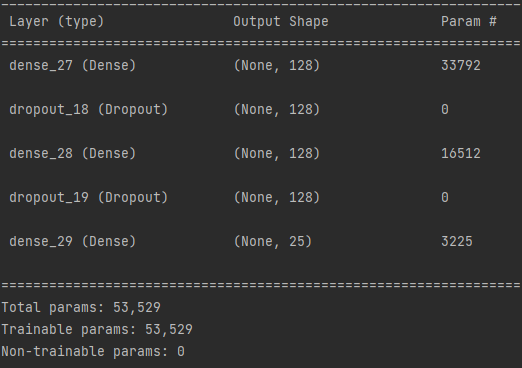
Xây dựng “optimizer[[31]](#footnote-31)” cho mô hình huấn luyện là SGD với các chỉ số bao gồm tốc độ học là 0.01, với chỉ số “decay[[32]](#footnote-32)” bằng 1e-6 giúp cập nhật lại tốc độ học qua từng vòng lặp (epochs). Và với với “Mementum[[33]](#footnote-33)” là 0.9 có tác dụng thúc đẩy quá trình vượt qua những điểm ngăn cản quá trình tiến đến mực tiêu nhưng khi tiến đến mục tiêu nó lại mất nhiều thời gian để dừng lại, để khắc phục điều đó với “Nesterov[[34]](#footnote-34)” là “true” giúp ta tiến tới quá trình hội tụ được nhanh hơn.



Hình 32: Mô hình mạng noron của mô hình Chatbot

Mạng nơ-ron xây dựng cho mô hình gồm đầu vào (Input) mảng véc-tơ từ với độ dài là 263 được xây dựng từ trước, tiếp theo 2 lớp ẩn (Hidden\_layer\_1) bao gồm lớp đầu tiên gồm 128 nút và lớp thứ hai số lượng nút là 128, trong hai lớp ẩn được xây dựng thêm hàm kích hoạt “tanh”. Cuối cùng là đầu ra (Output) là mảng giá trị vector nhãn độ dài là 25 tương ứng số lượng nhãn trong tập dữ liệu và có thêm hàm kích hoạt “softmax”.

Mạng nơ-ron sau khi thêm các lớp và các hàm kích hoạt cần thiết, mô hình sẽ được huấn luyện trên tập dữ liệu gồm 1043 câu hỏi. Với số vòng lặp (epochs) là 100, và số lượng batch\_size là 4. Sau khi thực hiện huấn luyện xong, mô hình được lưu với tên là “Chatbot\_model.h5”



Hình 33: Mô hình Chatbot\_model.h5

#### Đánh giá mô hình tối ưu hóa với giải thuật SGD

Sau khi xây dựng được mô hình, chúng ta lần lượt thực hiện đánh giá lại các hàm kích hoạt đã sử dụng là: relu, signod và tanh. Mô hình vẫn được huấn luyện trên tập dữ liệu gồm 1043 câu hỏi, số lượng vòng lặp (epochs) là 100, số lượng batch\_size là 4. Kết quả trả về lần lượt là:

* Đối với hàm “tanh” xấp xỉ đạt: 0.9494736790657043
* Đối với hàm “signod” xấp xỉ đạt: 0.8863157629966736
* Đối với hàm “relu” xấp xỉ đạt: 0.945263147354125

Trong mô hình, hàm kích hoạt “tanh” cho ra độ chính xác cao nhất, độ chính xác lên đến 94,94%.

### Xử lý câu hỏi người dùng và đưa ra dự đoán

Giao diện của website được xây dựng bằng ngôn ngữ HTML, được căn chỉnh bằng CSS và tạo các tính năng bằng ngôn ngữ lập trình Javascript và sự hỗ trợ của Framework Flask. Ứng dụng website được xây dựng đơn giản, thân thiện tối ưu hóa khả năng chat của người sử dụng, ứng dụng được chạy local trên máy với địa chỉ là: <http://127.0.0.1:5000/>.

Ứng dụng website có phương thức trao đổi theo hình thức Người hỏi – Người trả lời. Phương thức này thông qua việc sử dụng văn bản để truyền tải thông điệp theo yêu cẩu của người dùng

Người sử dụng sẽ đóng vai trò là người hỏi, sử dụng phương thức nhập câu hỏi thông qua văn bản, có thể xuất hiện các từ viết tắt, viết có dấu hoặc không dấu, viết in hoa trong câu hỏi.

Người trả lời là Chatbot sau khi nhận được câu hỏi của người dùng, sẽ thực hiện các bước bên trong để xử lý câu hỏi của người dùng như các xử lý tập dữ liệu bằng việc chuyển câu hỏi về dạng không dấu, xử lý các từ viết tắt, tách từ,… với các câu hỏi có nhiều ý sẽ được tách thành các ý riêng biệt để trả lời. Sau khi thực hiện xong quá trình xử lý câu hỏi, ứng dụng sẽ thực hiện tạo mảng véc-tơ đưa vào mạng noron để đưa ra dự đoán câu trả lời.

Kết quả dựa đoán sẽ đưa ra kết quả là xác xuất câu hỏi đó thuộc nhãn nào bao nhiêu phần trăm. Nếu nhãn hệ thống dự đoán có kết quả lớn hơn 90% thì hệ thống sẽ tìm kiếm bên trong trường “responses” câu trả lời phù hợp với yêu cầu, ngược lại với xác xuất nhỏ hơn 90% thì hệ thống sẽ trả lời người dùng là: “Xin lỗi, hiện tại câu hỏi này tôi không thể trả lời bạn.”

Ví dụ với câu hỏi “Điểm chuẩn khoa học máy tính là bao nhiêu?”

### Thu thập và thêm dữ liệu từ câu hỏi người dùng

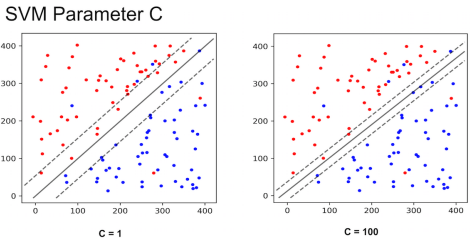
#### Thu thập dữ liệu

#### Thêm dữ liệu

### Các giải thuật dùng để huấn luyện mô hình và kiểm thử độ chính xác

#### Giải thuật SVM

Dữ liệu training sẽ được điều chỉnh nhãn để phù hợp với giải thuật. Đối với SVM nguyên cứu sử dụng thư viện Sklearn trong Python để tạo mô hình SVM. Mô hình sử dụng Kernel là Linear. Với Linear được sử dụng khi dữ liệu có thể phân tách tuyển tính, tức là nó có thể được phân tách bằng một Dòng duy nhất. Nó là một trong những nhân phổ biến nhất được sử dụng. Nó chủ yếu được sử dụng khi có một số lượng lớn tính năng trong một Tập dữ liệu cụ thể. Một trong những ví dụ có rất nhiều tính năng là Phân loại văn bản, vì mỗi bảng chữ cái là một tính năng mới.



Hình 34: Giải thuật SVM với tham số C=1 và C=100

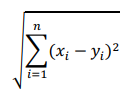
Tham số C cho biết mức độ tối ưu hóa SVM mà người sử dụng muốn để tránh phân loại sai từng ví dụ đào tạo. Đối với các giá trị lớn của C, việc tối ưu hóa sẽ chọn một siêu phẳng có lợi nhuận nhỏ hơn nếu siêu phẳng đó thực hiện tốt hơn việc nhận được tất cả các điểm rèn luyện được phân loại chính xác. Ngược lại, một giá trị rất nhỏ của C sẽ khiến trình tối ưu hóa tìm kiếm siêu phẳng phân tách có lề lớn hơn, ngay cả khi siêu phẳng đó phân loại sai nhiều điểm hơn. Đối với các giá trị rất nhỏ của C, người sử dụng sẽ nhận được các ví dụ bị phân loại sai, thường xuyên ngay cả khi dữ liệu đào tạo của người sử dụng có thể phân tách tuyến tính.

Trong nghiên cứu này mô hình SVM được dùng với kernel là Linear và C là 0.05.

#### Giải thuật K – láng giềng

Nghiên cứu sử dụng K – láng giềng để làm phép so sánh về độ chính xác với mô hình nghiên cứu, giải thuật sử dụng n\_neighbors = 5 và được xây dựng bằng cách sử dụng thư viện Sklearn trong Python để tạo mô hình.

Trong quá trình xử lý, giải thuật K – láng giềng sẽ dự đoán khoảng các giữa 2 vector bằng cách tính doán dựa trên công thức:



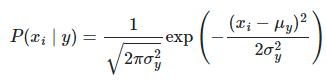
Hình 35: Công thức khoảng cách K – láng giềng

Để đưa ra dự đoán cho câu hỏi, giải thuật sẽ tính khoảng cách câu hỏi đó với toàn bộ câu hỏi trong tập dữ liệu. Nếu khoảng cách câu hỏi nào nhỏ nhất khi xét với câu hỏi cần dự đoán sẽ lấy câu trả lời của câu hỏi đó ra làm câu trả lời cho câu hỏi cần dự đoán.

#### Giải thuật Naives Bayes

Dữ liệu training sẽ được điều chỉnh nhãn để phù hợp với giải thuật. Đối với Gausian Naïve Bayes nguyên cứu sử dụng thư viện Sklearn trong Python để tạo mô hình Gausian Naïve Bayes

GaussianNB triển khai thuật toán Gaussian Naive Bayes để phân loại. Khả năng các tính năng được giả định là Gaussian:



Hình 36: Công thức cho giải thuật

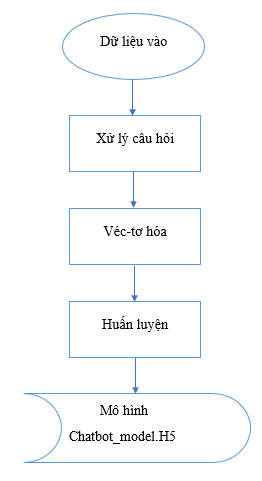
Các tham số và được ước tính bằng cách sử dụng khả năng tối đa.

## Xây dựng hệ thống

Cài đặt giải thuật được chia làm hai giai đoạn: Tạo mô hình dự đoán và Dự đoán câu trả lời.

### Tạo mô hình dự đoán

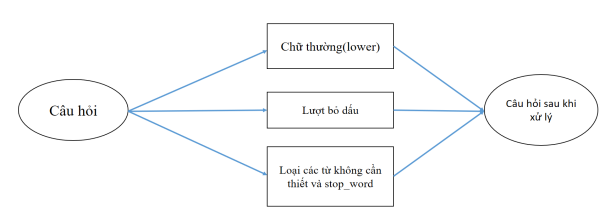
Dữ liệu được lấy từ tập dữ liệu 1043 câu hỏi được thu thập từ nhiều nguồn khác nhau. Trải qua quá trình xử lý câu hỏi cũng như là véc-tơ hóa câu hỏi đó. Tạo thành một dãy những con số 0, 1 ứng với từng câu hỏi. Sau khi trải qua quá trình đó dữ liệu bao gồm 1043 câu hỏi sẽ được đem đi huấn luyện để tạo hành mô hình Chatbot\_model.H5, tất cả quả trình được minh hoạ như sau:



Hình 37: Tạo mô hình dự đoán

#### Xử lý dữ liệu câu hỏi

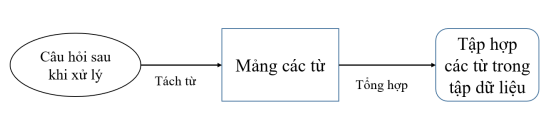
Câu hỏi sẽ được chuyển thành chữ thường, lượt bỏ dấu và loại các từ không cần thiết trong câu và stop\_word. Quá trình xử dụng các hàm trong python như lower(), replace(), sub().

****

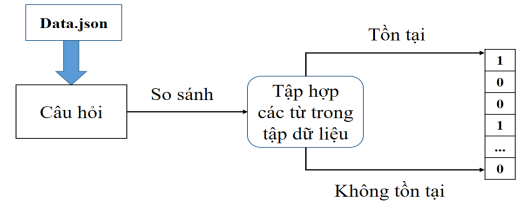
Hình 38: Xử lý câu hỏi

#### Véc-tơ hóa câu hỏi

Sau khi câu hỏi được xử lý từ bước 1, thực hiện tách các từ trong câu hỏi để tạo thành 1 mảng từ. Tổng hợp nhiều mảng từ để tạo thành 1 tập hợp các từ trong tập dữ liệu. Quá trình đó như sau:

****

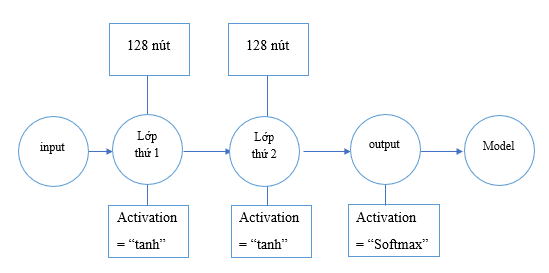
Hình 39: Tạo tập hợp từ



Hình 40: Véc-tơ hóa câu hỏi

Tất cả câu hỏi trong tập dữ liệu sẽ được so sánh với tập hợp các từ trong tập dữ liệu (263 từ). Quá trình véc-tơ sử dụng hàm word\_tokenzie() giúp phân tách các từ trong câu thành mảng các từ. Sau đó chuyển hóa thành véc-tơ mảng số để đưa vào mô hình huấn luyện.

#### Xây dựng mô hình huấn luyện:

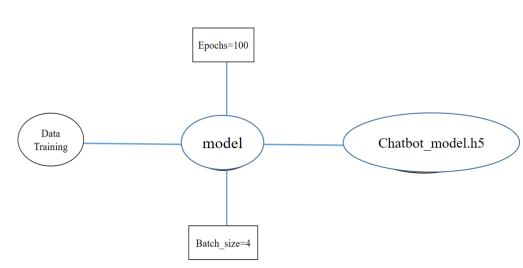
****

Hình 41: Mô hình mạng noron

Mô hình được xây dựng thông qua thư viện Keras. Đầu tiên là xây dựng hàm tối ưu hóa SGD với các thông số thích hơp. Tiếp theo xây dựng từng lớp ẩn cho mạng nơ-ron.

#### Huấn luyện mô hình

Dữ liệu với 1043 câu hỏi sau khi được véc-tơ hóa sẽ được huấn luyện với số vòng lặp là 100 và được chia batch\_size là 4. Và kết quả của quá trình là được mô hình “Chatbot\_model.H5”.

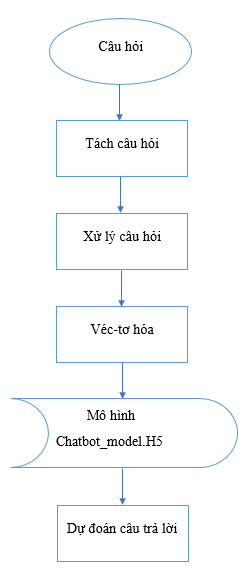
****

Hình 42: Quá trình huấn luyện mô hình

### Xử lý câu hỏi người dùng và dự đoán câu trả lời

#### Dự đoán câu trả lời

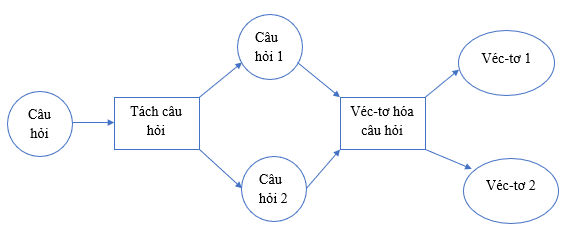
Quá trình dự đoán câu trả lời của người dùng sẽ được minh họa bằng mô hình như sau:



Hình 43: Quá trình dự đoán câu trả lời của người dùng

#### Xử lý câu hỏi của người dùng:

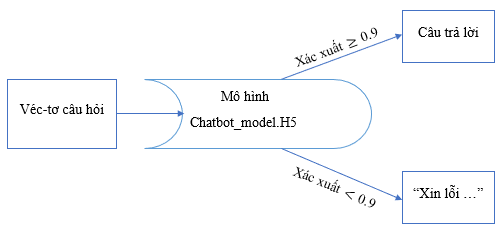
Quá trình xử lý câu hỏi của người dùng được mô hình hóa như sau:



Hình 44: Quá trình xử lý câu hỏi người dùng

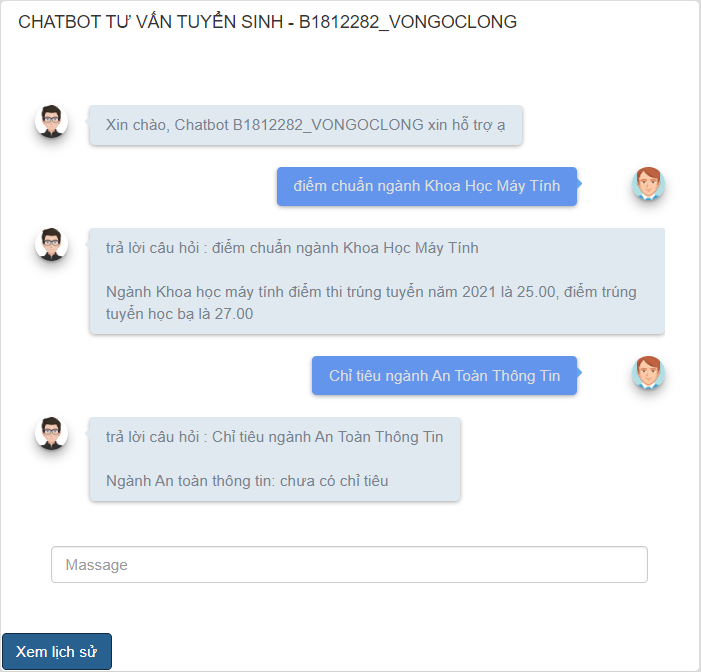
Câu hỏi của người dùng khi vào hệ thống sẽ được tách thành từng ý (nếu có), sau đó sẽ trải qua quá trình tiền xử lý dữ liệu và được véc-tơ hóa thành mảng một chiều. Nếu trong câu hỏi có các từ liên quan đến nhóm ngành Công nghệ Thông Tin thuộc Trường Công nghệ Thông tin và Truyền thông bao gồm các ngành: Công nghệ Thông Tin, Công nghệ Thông Tin Chất Lượng Cao, Công nghệ Thông Tin học tại Khu Hòa An (nhóm ngành Công nghệ Thông tin), An toàn thông tin, Truyền thông Đa phương tiện, Khoa học máy tính, Kỹ thuật máy tính, Mạng máy tính và Truyền Thông dữ liệu, Kỹ thuật phần mềm và Hệ thống thông tin sẽ được đánh dấu để trả lời người dùng đúng ngành nghề mà người dùng mong muốn.

#### Đưa ra dự đoán câu trả lời:



Hình 45: Quá trình đưa ra dự đoán câu trả lời

Khi véc-tơ hóa câu hỏi người dùng, mô hình “Chatbot\_model.h5” sẽ ra dự đoán dựa trên véc-tơ hóa câu hỏi người dùng nếu xác suất nhãn dự đoán lớn hoặc bằng 0.9 thì câu trả lời sẽ được đưa ra cho người dùng, ngược lại sẽ được đưa ra thông báo “Xin lỗi, Hiện tại câu hỏi này tôi không thể trả lời bạn”.



Hình 46: Chatbot trả lời câu hỏi của người dùng

### Thu thập dữ liệu câu hỏi người dùng

### Thêm câu hỏi và tập huấn luyện

# KIỂM THỬ VÀ ĐÁNH GIÁ

## Kiểm tra và đánh giá

### Độ chính xác của ba giải thuật máy học

Với tập dữ liệu với số lượng câu hỏi là 1043, nghiên cứu thực hiện đánh giá độ chính xác bằng cách xử dụng hai nghi thức Hold-out nhằm tính toán chỉ số accuracy\_score của bai giải thuật: SVM, kNN, Gausian Naïve Bayes. Sau đó thực hiện so sánh các giá trị.

Đối với nghi thức Hold-out số lần lặp là 10

|  |  |  |  |
| --- | --- | --- | --- |
|  | SVM | kNN | Gausian Naïve Bayes |
| 1 | 0.9167 | 0.8994 | 0.8793 |
| 2 | 0.8937 | 0.8448 | 0.8736 |
| 3 | 0.9195 | 0.8075 | 0.9052 |
| 4 | 0.9339 | 0.8793 | 0.8994 |
| 5 | 0.8879 | 0.8707 | 0.9052 |
| 6 | 0.9080 | 0.8851 | 0.8851 |
| 7 | 0.9138 | 0.8592 | 0.9023 |
| 8 | 0.8994 | 0.8707 | 0.9023 |
| 9 | 0.8764 | 0.8563 | 0.9109 |
| 10 | 0.8534 | 0.8477 | 0.9167 |

Bảng 7: Độ chính xác của ba giải thuật với nghi thức Hold-out

Hình 47: Biểu đồ độ chính xác của 3 giải thuật với nghi thức Hold-out

Độ chính xác trung bình của ba giải thuật SVM, kNN, Gausian Naïve Bayes sau 10 lần lặp lần lượt xấp xỉ đạt: 90.02%, 89.79%, 86.20%. Giải thuật SVM là giải thuật có độ chính xác tốt nhất sau 10 lần lặp ở nghi thức Hold-out.

### Độ chính xác của mô hình với giải thuật SGD

Độ chính xác của mô hình Chatbot\_model.H5 với giải thuật SGD sử dụng tập dữ liệu bao gồm 1043 câu hỏi được tính toán bằng nghi thức Hold-out với số lần lặp là 10 lần, tập train là 2/3 và tập test là 1/3.

So sánh độ chính xác của mô hình với giải thuật SVM, giải thuật có độ chính xác cao nhất khi sử dụng nghi thức Hold-out để tính so với 2 giải thuật còn lại.

|  |  |  |
| --- | --- | --- |
|  | SVM | SGD |
| 1 | 0.9167 | 0.9339 |
| 2 | 0.8937 | 0.9310 |
| 3 | 0.9195 | 0.9109 |
| 4 | 0.9339 | 0.9296 |
| 5 | 0.8879 | 0.9239 |
| 6 | 0.9080 | 0.9339 |
| 7 | 0.9138 | 0.9282 |
| 8 | 0.8994 | 0.9167 |
| 9 | 0.8764 | 0.9230 |
| 10 | 0.8534 | 0.9181 |

Bảng 8: Độ chính xác của hai giải thuật với nghi thức Hold-out

Độ chính xác trung bình của mô hình với giải thuật SGD sau 10 lần lặp xấp xỉ: 0.9247.

Cùng với số lần lặp tương tự như thế, độ chính xác của 3 giải thuật còn lại là:

* Giải thuật SVM xấp xỉ đạt: 90.02% xếp thứ 2
* Giải thuật Gausian Naïve Bayes xấp xỉ đạt: 89.79% xếp thứ 3
* Giải thuật kNN xấp xỉ đạt: 86.20% xếp thứ 4

Hình 48: Biểu đồ độ chính xác của các giải thuật với nghi thức Hold-out

# TÀI LIỆU THAM KHẢO

[1] Do Thanh Nghi, Hoang Tung, Chatbot cho sinh vien Cong Nghe Thong Tin, Can Tho University, 2019.

[2] P. L. Wehenkel, Design and implementation of a chatbot in the context of customer support, 2017-2018.

[3] T. M. Luan, "VIBLO," 2018. [Online]. Available: https://viblo.asia/p/phan-1-deep-learning-cho-chatbot-gioi-thieu-gDVK2QEr5Lj.

[4] "AI curious," 2019. [Online]. Available: https://aicurious.io/posts/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks/#3-relu.

[5] T. T. Truc, "VIBLO," 2020. [Online]. Available: https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8.

[6] M. Pham, "Vietnambiz," 2020. [Online]. Available: https://vietnambiz.vn/thuat-toan-k-lang-gieng-gan-nhat-k-nearest-neighbor-knn-la-gi-2020022911113334.htm. [7] Funda, "Machine Learning co ban," 2017. [Online]. Available: <https://machinelearningcoban.com/2017/04/09/smv/>.

[8] "Keras," [Online]. Available: <https://keras.io/about/>.

[9] N. V. Hoang, "VIBLO," 2020. [Online]. Available: https://viblo.asia/p/gioi-thieu-google-speech-to-text-api-dich-mot-audio-file-sang-text-tren-nen-tang-nodejs-jvElayBDlkw.

[10] "TOPDeV," [Online]. Available: <https://topdev.vn/blog/json-la-gi/>.

[11] "FREETUTS," [Online]. Available: https://freetuts.net/gioi-thieu-sqlite-sqlite-la-gi-1719.html.

[12] "Tutorialspoint,"[Online].Available: <https://www.tutorialspoint.com/python_data_science/python_word_tokenization.htm>.

[14] "DataFlair," 2020. [Online]. Available: https://data-flair.training/blogs/pythonchatbot-project/. [15] "ScititLearn,"[Online].Available:https://scikitlearn.org/stable/modules/generated/sklearn.svm.SVC.html

1. https://tuyensinh.ctu.edu.vn/ [↑](#footnote-ref-1)
2. http://tuyensinh.hcmut.edu.vn/admission/ [↑](#footnote-ref-2)
3. https://rasa.com/ [↑](#footnote-ref-3)
4. https://www.djangoproject.com/ [↑](#footnote-ref-4)
5. https://www.facebook.com/ [↑](#footnote-ref-5)
6. https://en.wikipedia.org/wiki/Recurrent\_neural\_network [↑](#footnote-ref-6)
7. https://en.wikipedia.org/wiki/Long\_short-term\_memory [↑](#footnote-ref-7)
8. https://vi.wikipedia.org/wiki/Python [↑](#footnote-ref-8)
9. https://en.wikipedia.org/wiki/Keras [↑](#footnote-ref-9)
10. https://en.wikipedia.org/wiki/Natural\_Language\_Toolkit [↑](#footnote-ref-10)
11. https://vi.wikipedia.org/wiki/HTML [↑](#footnote-ref-11)
12. https://vi.wikipedia.org/wiki/CSS [↑](#footnote-ref-12)
13. https://vi.wikipedia.org/wiki/JavaScript [↑](#footnote-ref-13)
14. https://en.wikipedia.org/wiki/Deep\_learning [↑](#footnote-ref-14)
15. https://topdev.vn/blog/big-data/ [↑](#footnote-ref-15)
16. https://en.wikipedia.org/wiki/Artificial\_intelligence [↑](#footnote-ref-16)
17. https://en.wikipedia.org/wiki/Overfitting [↑](#footnote-ref-17)
18. https://machinelearningcoban.com/2017/01/16/gradientdescent2/ [↑](#footnote-ref-18)
19. https://en.wikipedia.org/wiki/Convolutional\_neural\_network [↑](#footnote-ref-19)
20. https://en.wikipedia.org/wiki/K-nearest\_neighbors\_algorithm [↑](#footnote-ref-20)
21. <https://keras.io/about/> [↑](#footnote-ref-21)
22. https://www.tensorflow.org/federated [↑](#footnote-ref-22)
23. https://docs.microsoft.com/en-us/cognitive-toolkit/ [↑](#footnote-ref-23)
24. https://flask.palletsprojects.com/en/2.2.x/ [↑](#footnote-ref-24)
25. https://aws.amazon.com/vi/what-is/api/ [↑](#footnote-ref-25)
26. https://github.com/h3x4n1um/CTU-Data\_Mining [↑](#footnote-ref-26)
27. https://vi.wikipedia.org/wiki/SQLite [↑](#footnote-ref-27)
28. https://vi.wikipedia.org/wiki/Chatbot [↑](#footnote-ref-28)
29. https://tuyensinh.ctu.edu.vn/ [↑](#footnote-ref-29)
30. https://product.vinbigdata.org/cac-ky-thuat-tach-tu-trong-xu-ly-ngon-ngu-tu-nhien/ [↑](#footnote-ref-30)
31. https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8 [↑](#footnote-ref-31)
32. https://viblo.asia/p/demon-momentum-decay-cho-mo-hinh-nn-aWj53jm8l6m [↑](#footnote-ref-32)
33. https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8 [↑](#footnote-ref-33)
34. https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8 [↑](#footnote-ref-34)