

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

---



**Nguyễn Minh Trí**

**ỨNG DỤNG MÁY HỌC TRONG TẠO SINH  
CÂU TRẢ LỜI CHO HỆ THỐNG HỎI - ĐÁP**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**  
**(Theo định hướng ứng dụng)**

**TP. HỒ CHÍ MINH - 2022**

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

-----



**Nguyễn Minh Trí**

**ỨNG DỤNG MÁY HỌC TRONG TẠO SINH  
CÂU TRẢ LỜI CHO HỆ THỐNG HỎI - ĐÁP**

**Chuyên ngành: Hệ thống thông tin**

**Mã số: 8.48.01.04**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

**(Theo định hướng ứng dụng)**

**NGƯỜI HƯỚNG DẪN KHOA HỌC:**

**PGS.TS NGUYỄN TUẤN ĐĂNG**

**TP. HỒ CHÍ MINH - 2022**

## LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: “*Ứng dụng máy học trong tạo sinh câu trả lời cho hệ thống hỏi đáp*” là công trình nghiên cứu của chính tôi.

Những kết quả nghiên cứu được trình bày trong luận văn là công trình của riêng của tôi dưới sự hướng dẫn của **PGS.TS Nguyễn Tuấn Đăng**.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được công bố trong công trình nghiên cứu nào khác.

Không có bất cứ thông tin nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

Tây Ninh, ngày 25 tháng 01 năm 2022

**Học viên thực hiện luận văn**

**Nguyễn Minh Trí**

## LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện luận văn, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới:

Tôi xin chân thành cảm ơn Ban Giám hiệu, quý thầy cô Khoa Đào tạo sau đại học của Học viện Công nghệ Bru chính Viễn thông đã tạo điều kiện thuận lợi giúp tôi hoàn thành luận văn này.

Tôi xin chân thành cảm ơn Thầy **PGS.TS Nguyễn Tuấn Đăng**, đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành luận văn.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn.

Mặc dù tôi đã rất cố gắng và hết sức nỗ lực nhưng không thể tránh khỏi những thiếu sót do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế, Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để có thể hoàn thiện hơn về kiến thức của bản thân.

Xin chân thành cảm ơn!

Tây Ninh, ngày 25 tháng 01 năm 2022

**Học viên thực hiện luận văn**

**Nguyễn Minh Trí**

## DANH MỤC HÌNH VẼ

Hình 1.1. Mạng nơ-ron nhân tạo.....	6
Hình 1.2. Kiến trúc của mạng RNN.....	9
Hình 1.3. Ứng dụng RNN trong phát sinh mô tả cho ảnh.....	11
Hình 1.4. Cấu trúc chung của RNN một chiều thông thường được hiển thị (a) với một đường trễ và (b) mở ra theo thời gian trong hai bước thời gian.....	12
Hình 1.5. Cấu trúc chung của mạng nơ-ron lặp lại hai chiều (BRNN) được hiển thị theo thời gian trong ba bước thời gian.....	13
Hình 1.6. Mô-đun lặp lại trong mạng RNN tiêu chuẩn chứa một lớp duy nhất. ....	13
Hình 1.7. Mô-đun lặp lại trong một LSTM chứa bốn lớp tương tác. ....	14
Hình 1.8. Công trong mô hình LSTM.....	14
Hình 1.9. LSTM và GRU.....	15
Hình 3.1. Quy trình bài toán.....	28
Hình 4.1. Mô tả bộ dữ liệu.....	32
Hình 4.2. Kết quả xử lý dữ liệu.....	33
Hình 4.3. Huấn luyện cho mô hình.....	34
Hình 4.4. Biểu đồ miêu tả độ chính xác của mô hình.....	35
Hình 4.5. Chatbot tư vấn về vấn đề động thai.....	35
Hình 4.6. Chatbot tư vấn về vấn đề khám thai.....	36
Hình 4.7. Chatbot tư vấn về bệnh trầm cảm sau sinh.....	36

**DANH MỤC TỪ VIẾT TẮT**

AI	Artificial Intelligence
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
BRNN	Bidirectional recurrent neural networks
LSTM	Long Short Term Memory
GPUs	Gated recurrent units

## MỤC LỤC

LỜI CAM ĐOAN .....	i
LỜI CẢM ƠN .....	ii
DANH MỤC HÌNH VẼ.....	iii
DANH MỤC TỪ VIẾT TẮT.....	iv
MỤC LỤC.....	v
I. MỞ ĐẦU .....	1
1. Lý do chọn đề tài.....	1
2. Tổng quan về vấn đề nghiên cứu .....	1
3. Mục đích nghiên cứu.....	2
4. Đối tượng và phạm vi nghiên cứu.....	2
5. Phương pháp nghiên cứu.....	2
II. PHẦN NỘI DUNG.....	3
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT .....	3
1.1 Giới thiệu chương 1.....	3
1.2 Hệ thống trả lời tự động .....	3
1.3 Phân loại các mô hình trả lời tự động .....	4
1.4 Kiến trúc mạng nơ-ron nhân tạo .....	6
1.5 Hoạt động của mạng nơ-ron nhân tạo [13] .....	7
1.6 Mạng nơ-ron RNN (Recurrent Neural Network) và ứng dụng.....	8
1.7 Mô hình trả lời tự động .....	15
1.8 Embedding và Keras Embedding Layer [18], [19].....	16
CHƯƠNG 2. CÁC CÔNG TRÌNH LIÊN QUAN.....	17
2.1 Giới thiệu chương 2.....	17

2.2 Tình hình nghiên cứu trong và ngoài nước .....	17
CHƯƠNG 3. ĐỀ XUẤT PHƯƠNG PHÁP XÂY DỰNG BỘ DỮ LIỆU VÀ MÔ HÌNH CHO BÀI TOÁN.....	28
3.1 Giới thiệu chương 3.....	28
3.2 Đề xuất phương pháp và thuật toán xử lý .....	28
CHƯƠNG 4. CÀI ĐẶT VÀ THỰC NGHIỆM .....	30
4.1 Giới thiệu chương 4.....	30
4.2 Bộ dữ liệu của bài toán.....	30
4.3 Xây dựng mô hình.....	33
4.4 Đánh giá mô hình và áp dụng vào chatbot.....	34
4.5 Nhận xét .....	37
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	39
5.1 Kết quả nghiên cứu của đề tài .....	39
5.2 Hạn chế của đề tài .....	39
5.3 Hướng phát triển của đề tài .....	39
TÀI LIỆU THAM KHẢO.....	40
PHỤ LỤC.....	46



# I. MỞ ĐẦU

## 1. Lý do chọn đề tài

Các hệ thống trả lời câu hỏi (Question-Answering System - QAS) là những hệ thống có thể tự phân tích câu hỏi và tự đưa ra câu trả lời. Các hệ thống QAS được ứng dụng trong kinh doanh và thương mại điện tử có thể hỗ trợ khách hàng mua sản phẩm và giúp doanh nghiệp tăng doanh thu. Ví dụ, khi mua sắm trên mạng, người dùng có thể truy cập vào trang web của các doanh nghiệp và đặt câu hỏi để hiểu rõ hơn về sản phẩm. Yêu cầu của người mua hàng sẽ được các chatbot trên các website phân tích và đưa ra những câu trả lời với thông tin có ích cho người mua hàng. Các chatbot là những hệ thống trả lời tự động, có thể giúp cải thiện doanh thu bán hàng đáng kể và là thành phần không thể thiếu trong các website bán hàng ngày nay.

## 2. Tổng quan về vấn đề nghiên cứu

Trong những năm gần đây, khái niệm về trợ lý ảo, hay chatbot, hay hệ thống trả lời tự động được các công ty lớn như Microsoft (Cortana), Google (Google Assistant), Facebook (M), Apple (Siri), Samsung (Viv) đã giới thiệu và đưa vào khai thác. Gần đây nhất Microsoft đã tạo ra Microsoft Chat Framework cho phép các nhà phát triển tạo ra các chatbot trên nền tảng Web và Skype, hay Facebook cũng phát hành F8 SDK cho phép nhà phát triển tích hợp vào Messenger.

Hệ thống trả lời tự động có thể trợ giúp cho con người trong rất nhiều lĩnh vực: y tế, giáo dục, thương mại điện tử ... Với sự ra đời của framework sequence-to-sequence gần đây, nhiều hệ thống huấn luyện đã sử dụng các mạng nơron để sinh ra các câu trả lời mới khi đưa vào mạng một câu hỏi hoặc một thông điệp. Đây là một hướng tiếp cận mới rất có triển vọng trong việc xây dựng một hệ thống trả lời tự động.

Trong đề tài này, luận văn sẽ nghiên cứu các mô hình máy học để xây dựng mô hình hệ thống trả lời tự động có chức năng tạo sinh câu trả lời bằng tiếng Việt cho người dùng.

### **3. Mục đích nghiên cứu**

Đề tài luận văn nhằm mục tiêu nghiên cứu sử dụng các mô hình máy học và học sâu để xây dựng một hệ thống trả lời tự động (chatbot) có chức năng tạo sinh câu trả lời tiếng Việt trong một lĩnh vực ứng dụng cụ thể.

Phân luồng câu hỏi (phân tích câu hỏi) là pha đầu tiên trong kiến trúc chung của một hệ thống hỏi đáp, có nhiệm vụ tìm ra các thông tin cần thiết làm đầu vào cho quá trình xử lý của các pha sau (trích chọn tài liệu, trích xuất câu trả lời, ...). Vì vậy việc phân tích câu hỏi rất quan trọng, nó ảnh hưởng trực tiếp đến hoạt động của toàn bộ hệ thống. Nếu phân tích câu hỏi không tốt thì sẽ không thể tìm ra được câu trả lời. Chính vì thế đề tài “Hệ thống tự động phân luồng câu hỏi và giải đáp yêu cầu trực tuyến” được tác giả chọn và nghiên cứu.

### **4. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu: Mô hình hệ thống trả lời tự động (chatbot), các mô hình máy học và học sâu.

Phạm vi nghiên cứu: nghiên cứu hệ thống tư vấn hỏi đáp các vấn đề về sức khỏe sinh sản cho người bệnh tại bệnh viện Hồng Hưng tỉnh Tây Ninh

### **5. Phương pháp nghiên cứu**

Nghiên cứu các mô hình hệ thống trả lời tự động (chatbot) dựa trên máy học và học sâu.

Mô hình có khả năng ghi nhớ và trả lời chính xác câu hỏi đã được nhìn thấy trong thời gian đào tạo (gọi là Generator)

Mô hình có khả năng trả lời các câu hỏi mới tại thời điểm kiểm tra và chọn một câu trả lời từ tập hợp các câu trả lời trong quá trình đào tạo (gọi là Retriever-Generator)

Mô hình có khả năng trả lời các câu hỏi không có trong tập dữ liệu đào tạo (gọi là Retriever-Reader)

## II. PHẦN NỘI DUNG

### CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

#### 1.1 Giới thiệu chương 1

Bài toán xây dựng hệ thống hỏi đáp là một bài toán khó thuộc lĩnh vực xử lý ngôn ngữ tự nhiên. Chúng ta biết rằng ngôn ngữ tự nhiên có nhiều nghĩa, việc xác định được ngữ nghĩa của câu hỏi cũng như để phát hiện ra câu trả lời là một thách thức khá lớn. Không những vậy, giữa câu hỏi và câu trả lời còn tồn tại các quan hệ “ngầm” hay phụ thuộc vào ngữ cảnh. Bài toán đặt ra nhiều thách thức để phát hiện ra được câu trả lời phù hợp nhất với câu hỏi. Chương này sẽ giới thiệu tổng quan về hệ thống trả lời tự động, tìm hiểu các nghiên cứu ở trong và ngoài nước để thấy được tình hình nghiên cứu và các phương pháp tiếp cận của các nghiên cứu trước đây.

#### 1.2 Hệ thống trả lời tự động

Hệ thống trả lời tự động (QA) [1] là một phạm vi của ngành khoa học máy tính trong các lĩnh vực truy xuất thông tin và xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) – một hệ thống xử lý và trả lời các câu hỏi do con người đặt ra dưới dạng ngôn ngữ tự nhiên. QA thường được vận hành bởi một chương trình máy tính, xây dựng các câu trả lời bằng cách truy vấn đến một cơ sở dữ liệu có cấu trúc chứa các thông tin hoặc kiến thức liên quan, thường là dựa trên kiến thức. ELIZA – một trong những hệ thống trả lời tự động đầu tiên được phát triển vào năm 1964 có sự thành công vượt trội khi được công nhận là một ứng dụng hữu ích trong lĩnh vực y tế. ELIZA được xem là một bác sĩ trong lĩnh vực y tế, nó có nhiệm vụ là tương tác với người dùng qua một giao diện tin nhắn, trả lời các câu hỏi và phản hồi đến hộp thoại tin nhắn của người dùng theo cách “bắt chước” liệu pháp tâm lý của khách hàng trung tâm giữa khách hàng (người dùng) và bác sĩ của họ (chương trình máy tính chạy ứng dụng của một bác sĩ).

QA [1], [2] được thiết kế để tìm ra các câu trả lời cho phạm vi các câu hỏi trong một tập tài liệu hoặc tạo ra câu trả lời từ một nguồn dữ liệu [3]. Hệ thống cho phép người dùng hỏi các câu hỏi bằng ngôn ngữ tự nhiên (Natural Language - NL),

sau đó sẽ trích xuất các câu trả lời liên quan, phản hồi lại câu hỏi của người dùng một cách chính xác, gần như theo ngôn ngữ tự nhiên và tức thời thay vì gửi các tập tài liệu liên quan như các loại công cụ tìm kiếm [4], [5], [6]. QA [1] ngày càng thu hút được nhiều các nhà khoa học nghiên cứu và phát triển vì nhiều người dùng mong muốn hệ thống có thể trả lời câu hỏi một cách nhanh chóng và chính xác nhất có thể. Đồng thời, việc phát triển và mở rộng hệ thống QA sẽ giúp cho quá trình xử lý các tác vụ trong hệ thống trở nên tốt và hiệu quả hơn. Hệ thống bao gồm ba module cơ bản: module quá trình xử lý câu hỏi, module quá trình xử lý tài liệu và module hình thành các công thức và trích xuất câu trả lời.

Để hệ thống QA ngày càng được cải tiến và phát triển về độ chính xác, một số hướng tiếp cận với trí tuệ nhận tạo (Artificial Intelligence – AI) và thuật toán được áp dụng trong mô hình học có giám sát và không giám sát [7]. Bên cạnh đó, hệ thống QA vẫn còn gặp nhiều thử thách trong quá trình NLP [8]. Tuy nhiên trong những năm gần đây, lĩnh vực NLP được phát triển mạnh mẽ và nhanh chóng và mạnh mẽ về vấn đề xử lý thuật ngữ máy tính và AI [9] nhằm cải thiện độ chính xác của các câu trả lời và thể hiện ngôn ngữ một cách tự nhiên nhất có thể.

### **1.3 Phân loại các mô hình trả lời tự động**

#### ***1.3.1 Phân loại theo miền ứng dụng [5]***

**Miền mở (Open Domain):** Hệ thống trả lời tự động trên miền mở có nhiệm vụ xác định các câu trả lời cho các kiểu câu hỏi mang ngôn ngữ tự nhiên từ kho tài liệu khổng lồ. Hệ thống QA miền mở điển hình sẽ bắt đầu với việc truy xuất thông tin để chọn ra một tập hợp con các tài liệu từ kho tài liệu, sau đó được xử lý bởi một bộ đọc máy để chọn các khoảng câu trả lời [10], [11]. Ngoài ra, hệ thống trả lời câu hỏi miền mở có khả năng giải quyết đa dạng các loại câu hỏi và chỉ có thể dựa trên các bản thể học chung (ontology) và các kiến thức trên thế giới. Mặt khác, các hệ thống này thường có sẵn nhiều dữ liệu hơn để trích xuất câu trả lời phù hợp [5].

**Miền đóng (Close Domain):** Hệ thống trả lời câu hỏi miền đóng sẽ xử lý các câu hỏi theo một miền cụ thể [12] và đây có thể được xem là một nhiệm vụ dễ dàng hơn vì quá trình xử lý ngôn ngữ tự nhiên (NLP) có thể khai thác các kiến thức về một

miền cụ thể, có nội dung tin tưởng và thường được chính thức hóa trong các bản thể học. Trong một số trường hợp, hệ thống QA miền đóng sẽ chỉ đáp ứng được một số các câu hỏi hạn chế, ví dụ như câu hỏi yêu cầu về thông tin mô tả thay vì thủ tục [5].

### ***1.3.2 Phân loại theo hướng tiếp cận***

Tiếp cận dựa vào trích chọn thông tin (Retrieval-based): Các kỹ thuật thường sử dụng một kho đã định nghĩa trước các câu trả lời kết hợp với một vài phương pháp trích chọn Heuristic để nhả ra một đáp án thích hợp nhất dựa vào mẫu hỏi input và ngữ cảnh. Kỹ thuật heuristic sử dụng đơn giản có thể là sự so sánh trùng khớp các biểu thức dựa vào luật (rule-based), hoặc phức tạp như việc kết hợp học máy (Machine Learning) để phân lớp các câu hỏi và đáp án trả về. Những hệ thống này không sinh ra văn bản mới, chúng chỉ sử dụng một đáp án từ một tập dữ liệu cố định sẵn có.

Tiếp cận dựa vào mô hình sinh (Generative-based): Mô hình này không dựa trên tập trả lời đã được định nghĩa trước đó. Chúng có khả năng tự xây dựng các đáp án từ đầu. Các mô hình sinh sẽ dựa vào các kỹ thuật Máy Dịch (Machine Translation), nhưng không dịch từ ngôn ngữ này sang ngôn ngữ khác, mà nó có thể “dịch” từ một input sang một output.

Nhờ vào kho dữ liệu với các bộ luật được thiết kế bằng tay, mô hình dựa trên trích chọn thông tin (retrieval-based) không mắc phải các lỗi về ngữ pháp. Tuy nhiên, chúng không thể xử lý được các trường hợp các mẫu chưa được quan sát, không có trong bộ luật. Vì những lý do đó, các mô hình này không thể nhớ được các thông tin ngữ cảnh trước đó như “tên người” được đề cập trong đoạn hội thoại.

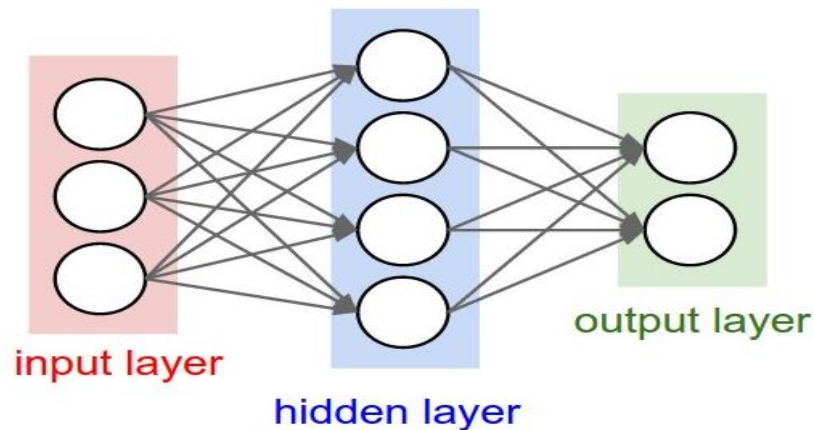
Mô hình sinh thì “thông minh hơn”. Chúng có khả năng nhớ lại được các thực thể được nhắc đến trong mẫu hỏi và bạn sẽ có cảm giác đang nói chuyện với con người. Tuy nhiên, những mô hình này thì rất khó để huấn luyện, dễ bị mắc lỗi về ngữ pháp (đặc biệt đối với các câu dài) và mô hình yêu cầu một lượng rất lớn dữ liệu để huấn luyện.

Các kỹ thuật học sâu Deep Learning được sử dụng cho cả hai mô hình Retrieval-based hoặc Generative-based, nhưng các nhà nghiên cứu thường tập trung hướng về mô hình Generative. Hiện tại, chúng ta vẫn đang ở giai

đoạn đầu của việc tiếp cận với mô hình sinh và có kết quả khả quan. Và các hệ thống thương mại vẫn phù hợp với các mô hình Retrieval-based.

#### 1.4 Kiến trúc mạng nơ-ron nhân tạo

Với sự ảnh hưởng từ mạng nơ-ron sinh học, mạng nơ-ron nhân tạo (Artificial Neural Network – ANN) được hình thành từ các tầng nơ-ron nhân tạo. Mạng NN gồm 3 kiểu tầng chính là **tầng vào** (*input layer*) biểu diễn cho đầu vào, **tầng ra** (*output layer*) biểu diễn cho kết quả đầu ra và **tầng ẩn** (*hidden layer*) thể hiện cho các bước suy luận trung gian. Mỗi nơ-ron sẽ nhận tất cả đầu vào từ các nơ-ron ở tầng trước đó và sử dụng một **hàm kích hoạt dạng** (*activation function*) phi tuyến như *sigmoid*, *ReLU*, *tanh* để tính toán đầu ra.



**Hình 1.1: Mạng nơ-ron nhân tạo**

Trong ANN, mỗi nút mạng là một sigmoid nơ-ron nhưng hàm kích hoạt của chúng có thể khác nhau. Trong thực tế người ta thường để chúng cùng dạng với nhau nhằm mục đích tính toán cho thuận lợi.

Lợi thế lớn nhất của các mạng ANN là khả năng được sử dụng như một cơ chế xấp xỉ hàm tùy ý mà “học” được từ các dữ liệu quan sát. Tuy nhiên, việc sử dụng mạng ANN lại không đơn giản, một số các đặc tính và kinh nghiệm khi thiết kế một mạng nơ-ron ANN cần lưu ý một số vấn đề như sau:

- Chọn mô hình: Điều này phụ thuộc vào cách trình bày dữ liệu và các ứng dụng. Mô hình quá phức tạp có xu hướng dẫn đến những thách thức trong quá trình học.

- Cấu trúc và sự liên kết giữa các nơ-ron.
- Thuật toán học: mỗi mạng ANN có hai vấn đề cần học
  - Học tham số của mô hình (parameter learning): là thay đổi trọng số của các liên kết giữa các nơ-ron trong một mạng.
  - Học cấu trúc (structure learning): là việc điều chỉnh cấu trúc mạng bằng việc thay đổi số lớp ẩn, số nơ-ron mỗi lớp và cách liên kết giữa chúng.
  - Học tham số và học cấu trúc có thể được thực hiện đồng thời hoặc tách biệt.

Nếu các mô hình, hàm chi phí và thuật toán học được lựa chọn một cách thích hợp, thì mạng ANN có thể sẽ cho kết quả vô cùng mạnh mẽ và hiệu quả.

### **1.5 Hoạt động của mạng nơ-ron nhân tạo [13]**

Mạng nơ-ron nhân tạo sử dụng các lớp xử lý toán học khác nhau để hiểu thông tin mà nó được cung cấp. Thông thường, một mạng nơ-ron nhân tạo có từ hàng chục đến hàng triệu nơ-ron nhân tạo - được gọi là các đơn vị - được sắp xếp thành một loạt các lớp. Lớp đầu vào nhận các dạng thông tin khác nhau từ thế giới bên ngoài. Đây là dữ liệu mà mạng nhắm đến để xử lý hoặc tìm hiểu. Từ đơn vị đầu vào, dữ liệu đi qua một hoặc nhiều đơn vị ẩn. Công việc của đơn vị ẩn là biến đầu vào thành thứ mà đơn vị đầu ra có thể sử dụng.

Phần lớn các mạng nơ-ron được kết nối đầy đủ từ lớp này sang lớp khác. Các kết nối này có trọng số; Con số càng cao thì một đơn vị này càng có ảnh hưởng lớn đến đơn vị khác, tương tự như bộ não con người. Khi dữ liệu đi qua từng đơn vị, mạng sẽ tìm hiểu thêm về dữ liệu. Ở phía bên kia của mạng là các đơn vị đầu ra và đây là nơi mạng phản hồi dữ liệu mà nó được cung cấp và xử lý.

Các nhà khoa học thần kinh nhận thức đã học được rất nhiều điều về bộ não con người kể từ khi các nhà khoa học máy tính lần đầu tiên thử nghiệm mạng nơ-ron nhân tạo ban đầu. Một trong những điều họ học được là các phần khác nhau của não chịu trách nhiệm xử lý các khía cạnh khác nhau của thông tin và các phần này được sắp xếp theo thứ bậc. Vì vậy, thông tin đầu vào đi vào não và mỗi cấp độ tế bào thần

kinh cung cấp cái nhìn sâu sắc và sau đó thông tin được chuyển sang cấp độ tiếp theo, cao cấp hơn. Đó chính xác là cơ chế mà ANN đang cố gắng tái tạo.

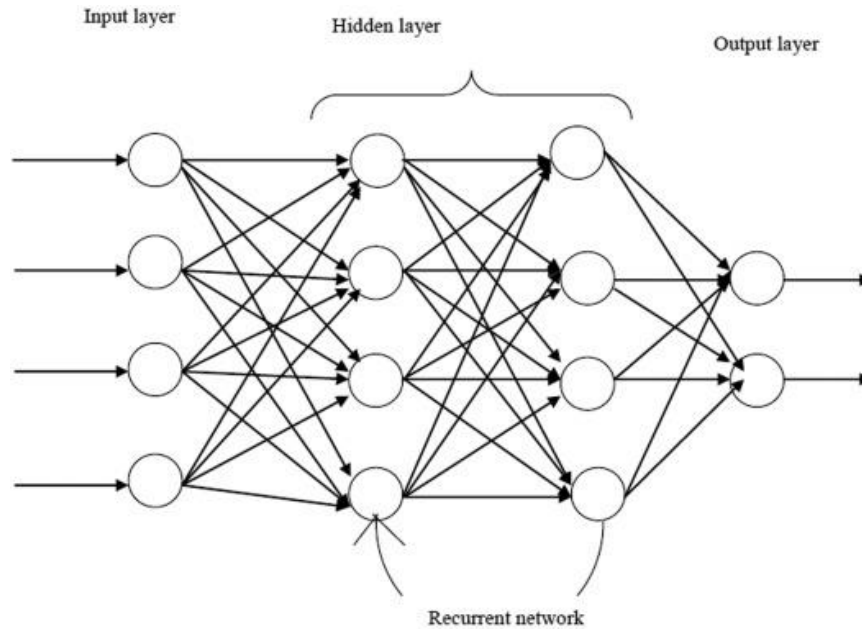
Để ANN có thể học được, họ cần phải có một lượng lớn thông tin được gọi là tập huấn luyện. Khi bạn đang cố gắng dạy ANN cách phân biệt mèo với chó, tập huấn luyện sẽ cung cấp hàng nghìn hình ảnh được gắn thẻ là chó để mạng bắt đầu tìm hiểu. Sau khi được đào tạo với lượng dữ liệu đáng kể, nó sẽ cố gắng phân loại dữ liệu trong tương lai dựa trên những gì nó cho rằng nó đang nhìn thấy (hoặc nghe thấy, tùy thuộc vào tập dữ liệu) trong các đơn vị khác nhau. Trong thời gian đào tạo, đầu ra của máy được so sánh với mô tả do con người cung cấp về những gì cần quan sát. Nếu chúng giống nhau, máy đã được xác nhận. Nếu không chính xác, nó sẽ sử dụng phương pháp truyền ngược để điều chỉnh việc học của mình - quay lại các lớp để điều chỉnh phương trình toán học. Được gọi là học sâu, đây là điều làm cho một mạng trở nên thông minh.

## **1.6 Mạng nơ-ron RNN (Recurrent Neural Network) và ứng dụng**

### ***1.6.1 Mạng nơ-ron RNN***

Mạng nơ-ron tuần hoàn (RNN) là một loại neural network, được sử dụng rộng rãi để thực hiện quá trình phân tích trình tự vì RNN được thiết kế để trích xuất thông tin ngữ cảnh bằng cách xác định sự phụ thuộc giữa các tem thời gian khác nhau. RNN bao gồm nhiều lớp lặp lại liên tiếp và các lớp này được lập mô hình tuần tự để ánh xạ trình tự với các trình tự khác. RNN có một khả năng mạnh mẽ để thu thập dữ liệu theo ngữ cảnh từ chuỗi. Tuy nhiên, các dấu hiệu ngữ cảnh trong cấu trúc mạng là ổn định và được sử dụng hiệu quả để đạt được quá trình phân loại dữ liệu. RNN có thể vận hành các chuỗi với độ dài tùy ý.





**Hình 1.2: Kiến trúc của mạng RNN**

RNN là phần mở rộng của neural network cấp tiếp với sự hiện diện của các vòng lặp trong các lớp ẩn. RNN lấy đầu vào là chuỗi các mẫu và xác định mối quan hệ thời gian giữa các mẫu. Bộ nhớ ngắn hạn dài (LSTM) giải quyết các vấn đề phân loại bằng cách thêm các tham số mạng với nút ẩn và giải phóng trạng thái dựa trên các giá trị đầu vào. RNN đạt được hiệu suất tốt hơn LSTM bằng cách kích hoạt các trạng thái dựa trên các sự kiện mạng. Nút RNN thông thường bao gồm một thiên vị và trọng số duy nhất. RNN được đánh giá bằng cách sử dụng đơn vị định kỳ định kỳ và LSTM. Cấu hình mạng một đối một được hình thành bằng cách sử dụng các tham số mạng, trong đó bước thời gian của mỗi dữ liệu đầu vào tạo ra kết quả đầu ra với bước thời gian cụ thể. Nút RNN thông thường bao gồm một thiên vị và trọng số duy nhất, trong khi LSTM bao gồm bốn thiên vị hoặc trọng số như được chỉ định bên dưới:

- Lớp cổng quên
- Lớp cổng đầu vào
- Lớp cổng đầu ra
- Lớp cổng trạng thái

Đầu vào và cổng quên kiểm soát trạng thái ẩn trước đó và trạng thái đầu vào hiện tại góp phần vào trạng thái ô. Tuy nhiên, đầu vào, đầu ra và kích hoạt cổng quên

được chia tỷ lệ bằng cách sử dụng hàm sigmoid và đầu ra của trạng thái ẩn được lọc bằng cách sử dụng hàm hyperbol. Việc tối ưu hóa các tham số mạng bằng cách sử dụng gradient ngẫu nhiên được thực hiện dựa trên chuỗi dữ liệu đầu vào. Tuy nhiên, các siêu tham số lần lượt là cấu trúc của mạng (kích thước và các lớp), độ dài chuỗi, kích thước lô, động lượng và tốc độ học. Các siêu tham số được thiết lập thông qua tìm kiếm ngẫu nhiên hoặc thủ công.

Đầu vào của RNN là chuỗi các vector là  $\{y_1, y_2, \dots, y_M\}$ , chuỗi các trạng thái ẩn là  $\{z_1, z_2, \dots, z_M\}$  và đơn vị đầu ra tương ứng là  $\{v_1, v_2, \dots, v_M\}$ .

Lớp hồi quy bao gồm hàm tái quy  $d$ , lấy vector đầu vào  $y_x$  và đơn vị ẩn của trạng thái trước  $z_x$  làm đầu vào và tạo ra trạng thái ẩn dưới dạng:

$$z_x = d(y_x, z_{x-1}) = \tanh(P \cdot y_x + Q \cdot z_{x-1})$$

Hơn nữa, các đơn vị đầu ra được tính như sau:

$$v_x = \text{soft max}(R \cdot z_x)$$

Ở đây,  $P$ ,  $Q$  và  $R$  đại diện cho ma trận trọng số và hàm kích hoạt tanh biểu thị hàm tiếp tuyến hyperbol. RNN sử dụng chức năng rất phức tạp để tìm hiểu và kiểm soát luồng thông tin trong lớp lặp lại để nắm bắt các phụ thuộc dài hạn.

### 1.6.2 Các ứng dụng của RNN

- Phát sinh mô tả cho ảnh (Generating Image Descriptions)
- RNN kết hợp với Convolution Neural Networks có thể phát sinh ra được các đoạn mô tả cho ảnh. Mô hình này hoạt động bằng cách tạo ra những câu mô
- Tả từ các đặc trưng rút trích được trong bức ảnh.
- Dự đoán chuỗi thời gian (Time Series Prediction): Bất kỳ vấn đề chuỗi thời gian nào, như dự đoán giá cổ phiếu trong một tháng cụ thể, đều có thể được giải quyết bằng cách sử dụng RNN.
- Xử lý ngôn ngữ tự nhiên (Natural Language Processing): Lấy một chuỗi các từ làm đầu vào, RNN sẽ tiến hành dự đoán khả năng xuất hiện của từ tiếp theo. Đây có thể được coi là một trong những cách tiếp cận hữu ích nhất để phiên dịch các loại ngôn ngữ vì câu có nhiều khả năng nhất sẽ là câu đúng. Trong

phương pháp này, xác suất đầu ra của một “time-step” cụ thể sẽ được sử dụng để làm mẫu để xác định các từ trong lần lặp tiếp theo.



“A Dog catching a ball in mid air”

**Hình 1.3: Ứng dụng RNN trong phát sinh mô tả cho ảnh**

### ***1.6.3 Huấn luyện mạng***

Huấn luyện RNN tương tự như huấn luyện Neural Network truyền thống. Chúng ta sử dụng đến thuật toán backpropagation (lan truyền ngược) nhưng có một vài điều chỉnh cho tinh gọn và phù hợp. Gradient tại mỗi output không chỉ phụ thuộc vào kết quả tính toán của bước hiện tại mà còn phụ thuộc vào kết quả tính toán của các bước trước đó.

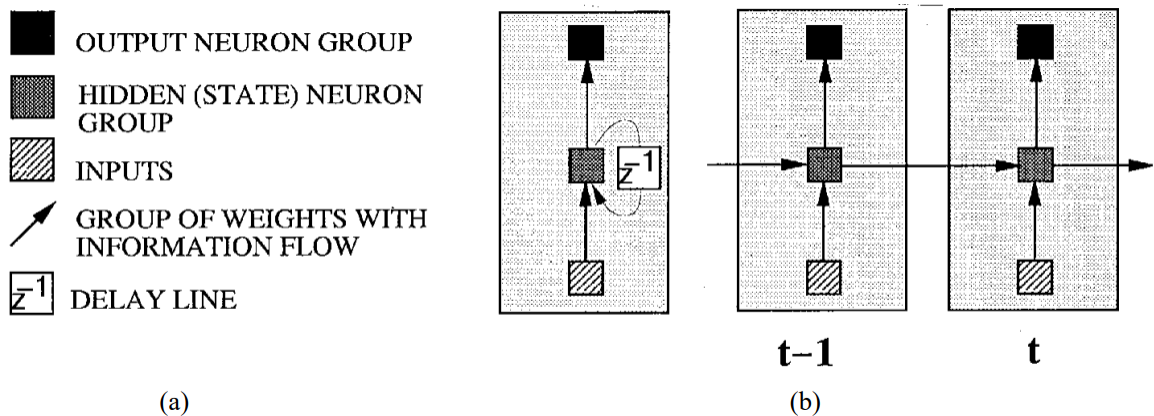
Ví dụ, để tính gradient tại thời điểm  $t = 4$ , ta cần backpropagation 3 bước trước đó và cộng dồn các gradient này lại với nhau. Kỹ thuật này gọi là Backpropagation Through Time (BPPTT). Ở đây, điểm hạn chế là hidden layer không có trí nhớ dài hạn. Vấn đề này gọi là vanishing/exploding gradient problem và LSTM được sinh ra để giải quyết vấn đề này.

### ***1.6.4 Các phiên bản mở rộng của RNN [14]***

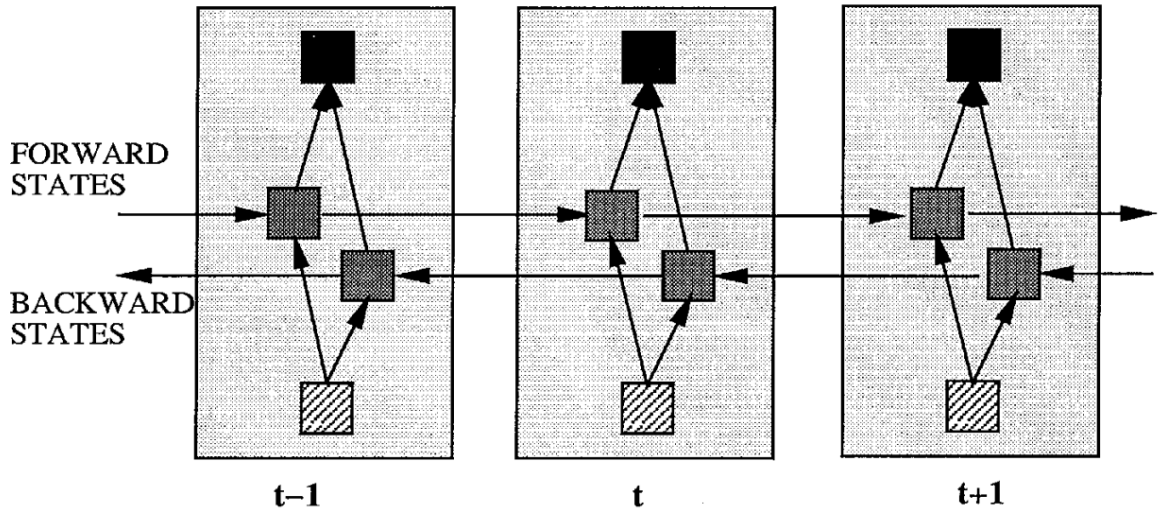
***Mạng nơ-ron tái phát hai chiều (Bidirectional recurrent neural networks - BRNN) [15]***

Đây là một kiến trúc mạng biến thể của RNN. Trong khi các RNN một chiều chỉ có thể được lấy từ các đầu vào trước đó để đưa ra dự đoán về trạng thái hiện tại, các RNN hai chiều lấy dữ liệu trong tương lai để cải thiện độ chính xác của nó. Ý tưởng là chia các tế bào thần kinh trạng thái của RNN thông thường thành một phần

chịu trách nhiệm về hướng thời gian dương (trạng thái thuận) và tách biệt cho hướng thời gian âm (trạng thái lùi). Đầu ra từ trạng thái chuyển tiếp không được kết nối với đầu vào của trạng thái lùi và ngược lại. Điều này dẫn đến cấu trúc chung có thể được nhìn thấy trong Hình 1.5, nơi nó được mở ra qua ba bước thời gian. Không thể hiển thị cấu trúc BRNN trong một hình tương tự như Hình 1.4 với đường trễ vì độ trễ sẽ phải có giá trị dương và âm theo thời gian. Lưu ý rằng không có trạng thái lùi, cấu trúc này đơn giản hóa thành RNN chuyển tiếp một chiều thông thường, như thể hiện trong Hình 1.4. Nếu các trạng thái tiến được loại bỏ, RNN thông thường với trục thời gian đảo ngược sẽ dẫn đến kết quả. Với cả hai hướng thời gian được xử lý trong cùng một mạng, thông tin đầu vào trong quá khứ và tương lai của khung thời gian được đánh giá hiện tại có thể được sử dụng trực tiếp để giảm thiểu hàm mục tiêu mà không cần có sự chậm trễ để bao gồm thông tin trong tương lai, như đối với một hướng thông thường RNN đã thảo luận ở trên. Ví dụ về cụm từ “feeling under the weather”, mô hình có thể dự đoán tốt hơn rằng từ thứ hai trong cụm từ đó là “under” nếu nó biết rằng từ cuối cùng trong chuỗi là “weather”.



**Hình 1.4:** Cấu trúc chung của RNN một chiều thông thường được hiển thị (a) với một đường trễ và (b) mở ra theo thời gian trong hai bước thời gian

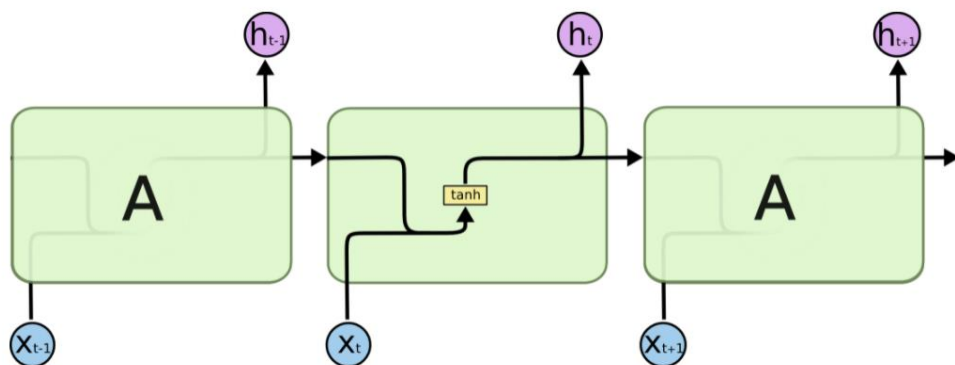


**Hình 1.5: Cấu trúc chung của mạng nơ-ron lặp lại hai chiều (BRNN) được hiển thị theo thời gian trong ba bước thời gian**

### ***Bộ nhớ ngắn hạn dài (LSTM) [16] [17]***

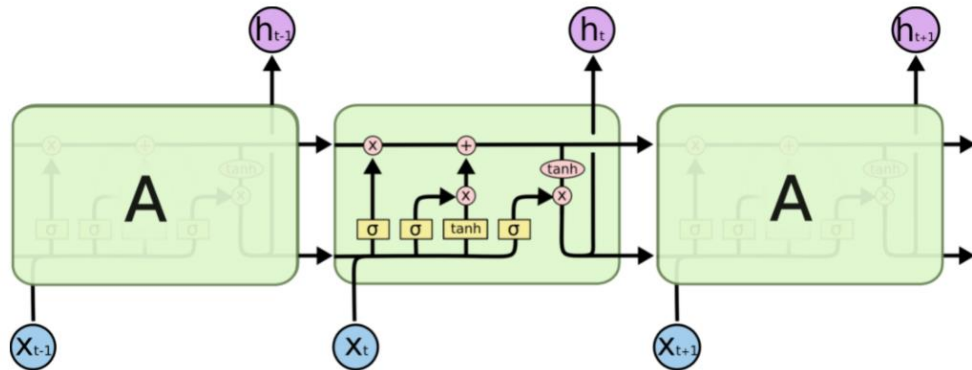
Đây là một kiến trúc RNN phổ biến, được giới thiệu bởi Sepp Hochreiter và Juergen Schmidhuber như một giải pháp cho vấn đề biến mất gradient. LSTM được thiết kế rõ ràng để tránh vấn đề phụ thuộc lâu dài. Ghi nhớ thông tin trong thời gian dài thực tế là hành vi mặc định của LSTM. Có nghĩa là, nếu trạng thái trước đó đang ảnh hưởng đến dự đoán hiện tại không phải là trong quá khứ gần đây, thì mô hình RNN có thể không thể dự đoán chính xác trạng thái hiện tại.

Tất cả các mạng nơ-ron tuần hoàn đều có dạng một chuỗi các mô-đun lặp lại của mạng nơ-ron. Trong các RNN tiêu chuẩn, mô-đun lặp này sẽ có cấu trúc rất đơn giản, chẳng hạn như một lớp tanh.



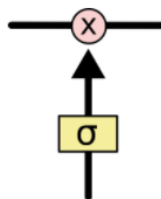
**Hình 1.6: Mô-đun lặp lại trong mạng RNN tiêu chuẩn chứa một lớp duy nhất.**

Các LSTM cũng có cấu trúc giống như chuỗi này, nhưng mô-đun lặp lại có cấu trúc khác. Thay vì có một lớp mạng nơ-ron duy nhất, có bốn lớp, tương tác theo một cách rất đặc biệt.



**Hình 1.7: Mô-đun lặp lại trong một LSTM chứa bốn lớp tương tác.**

Ý tưởng cốt lõi đằng sau LSTM chính là trạng thái ô, là đường ngang chạy qua đầu sơ đồ. Trạng thái tế bào giống như một băng chuyền. Nó chạy thẳng xuống toàn bộ chuỗi, chỉ với một số tương tác tuyến tính nhỏ. Rất dễ dàng để thông tin trôi theo dòng không thay đổi. LSTM có khả năng loại bỏ hoặc thêm thông tin vào trạng thái tế bào, được điều chỉnh cẩn thận bởi các cấu trúc được gọi là cổng.



**Hình 1.8: Cổng trong mô hình LSTM**

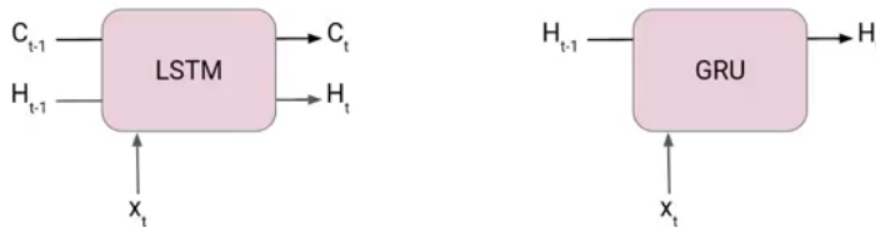
Cổng là một cách tùy chọn để thông qua thông tin. Chúng được cấu tạo từ một lớp mạng thần kinh sigmoid và một phép toán nhân theo chiều kim loại. Lớp sigmoid xuất ra các số từ 0 đến 1, mô tả mức độ thông qua của từng thành phần. Giá trị bằng 0 có nghĩa là "không có gì thông qua", trong khi giá trị bằng một có nghĩa là "để mọi thứ thông qua!". Một LSTM có ba trong số các cổng này, để bảo vệ và kiểm soát trạng thái tế bào.

Lấy một ví dụ để dễ dàng hiểu được LSTM: giả sử muốn dự đoán các từ in nghiêng sau đây, "Alice bị dị ứng với các loại hạt. Cô ấy không thể ăn *bơ đậu phộng*." Bối cảnh của dị ứng hạt có thể giúp chúng ta biết trước rằng thực phẩm không thể

ăn được có chứa các loại hạt. Tuy nhiên, nếu bối cảnh đó là một vài câu trước đó, thì RNN sẽ khó hoặc thậm chí không thể kết nối thông tin. Để khắc phục điều này, các LSTM có "ô" trong các lớp ẩn của mạng nơ-ron, có ba cổng - một cổng input, một cổng output và một cổng forget. Các cổng này kiểm soát luồng thông tin cần thiết để dự đoán đầu ra trong mạng. Ví dụ: nếu đại từ giới tính, chẳng hạn như "Cô ấy", được lặp lại nhiều lần trong các câu trước, bạn có thể loại trừ đại từ đó khỏi trạng thái ô.

### *Gated recurrent units (GRUs)*

Biến thể RNN này tương tự như LSTM vì nó cũng hoạt động để giải quyết vấn đề bộ nhớ ngắn hạn của các mô hình RNN. Thay vì sử dụng thông tin điều chỉnh "trạng thái ô" (cell state), nó sử dụng các trạng thái ẩn và thay vì ba cổng, nó có hai - một cổng đặt lại và một cổng cập nhật. Tương tự như các cổng trong LSTM, các cổng đặt lại và cập nhật kiểm soát lượng và thông tin nào cần giữ lại.



**Hình 1.9: LSTM và GRU**

## 1.7 Mô hình trả lời tự động

Bản thân mô hình seq2seq [10] bao gồm hai mạng RNN: Một cho bộ mã hóa, và một cho bộ giải mã. Bộ mã hóa nhận một chuỗi (câu) đầu vào và xử lý một phần tử (từ trong câu) tại mỗi bước. Với mục tiêu là chuyển đổi một chuỗi các phần tử vào một vector đặc trưng có kích thước cố định mà nó chỉ mã hóa thông tin quan trọng trong chuỗi và bỏ qua các thông tin không cần thiết. Có thể hình dung luồng dữ liệu trong bộ mã hóa dọc theo trục thời gian, giống như dòng chảy thông tin cục bộ từ một phần tử kết thúc của chuỗi sang chuỗi khác.

Mỗi trạng thái ẩn ảnh hưởng đến trạng thái ẩn tiếp theo và trạng thái ẩn cuối cùng được xem như tích lũy tóm tắt về chuỗi. Trạng thái này được gọi là bối cảnh hay vector suy diễn, vì nó đại diện cho ý định của chuỗi. Từ bối cảnh đó, các bộ giải mã tạo ra một chuỗi, một phần tử (word) tại một thời điểm.

## 1.8 Embedding và Keras Embedding Layer [18], [19]

Embedding là một kỹ thuật đưa một vector có số chiều lớn, thường ở dạng thưa, về một vector có số chiều nhỏ, thường ở dạng dày đặc. Phương pháp này đặc biệt hữu ích với những đặc trưng hạng mục có số phần tử lớn ở đó phương pháp chủ yếu để biểu diễn mỗi giá trị thường là một vector dạng one-hot. Một cách lý tưởng, các giá trị có ý nghĩa tương tự nhau nằm gần nhau trong không gian embedding.

Ví dụ nổi bật nhất là biểu diễn các từ trong một bộ từ điển dưới dạng số. Khi từ điển có hàng triệu từ, biểu diễn các từ dưới dạng one-hot vector dẫn tới số chiều vô cùng lớn. Hơn nữa, các từ này sẽ có khoảng cách đều nhau tới mọi từ khác (căn bậc hai của 2), dẫn đến việc thiếu thông tin giá trị cho việc huấn luyện mô hình machine learning. Chẳng hạn, một cách biểu diễn tốt các từ tiếng Việt cần mô tả tốt sự liên quan giữa cặp từ (vua, hoàng hậu) và (chồng, vợ) vì chúng có ý nghĩa gần nhau.

Keras cung cấp một Embedding layer để sử dụng cho mạng nơ-ron trên tập dữ liệu văn bản. Đầu vào yêu cầu là số nguyên được mã hóa, sao cho mỗi từ được biểu diễn bằng một số nguyên duy nhất. Bước chuẩn bị này có thể được thực hiện bằng cách sử dụng Tokenizer API có sẵn trong Keras.

Embedding layer được khởi tạo với trọng số (weight) ngẫu nhiên và sẽ tìm hiểu cách nhúng cho tất cả các từ trong tập dữ liệu training. Các thông số cơ bản để khởi tạo embedding layer như sau:

- **input\_dim**: kích thước của từ điển trong dữ liệu đầu vào, nếu dữ liệu đầu vào có giá trị là  $n$  thì kích thước là  $n+1$  từ.
- **output\_dim**: độ dài của vec-tơ tương ứng cho mỗi từ.
- **input\_length**: Độ dài của chuỗi đầu vào



## CHƯƠNG 2. CÁC CÔNG TRÌNH LIÊN QUAN

### 2.1 Giới thiệu chương 2

Chương này trình bày về các công trình nghiên cứu trong và ngoài nước liên quan mật thiết tới đề tài. Những công trình này là nền tảng cho nghiên cứu và cũng là cơ sở để giúp luận văn xác định được hướng phát triển cho đề tài.

### 2.2 Tình hình nghiên cứu trong và ngoài nước

#### 2.2.1 Các nghiên cứu trong nước

*“Building Filters for Vietnamese Chatbot Responses”* [20]

Chatbot tự động đang là xu hướng phát triển trên thế giới bởi tính tiện lợi, hiệu quả, khả năng ứng dụng trong nhiều lĩnh vực như Thương mại điện tử, chăm sóc khách hàng. Thông qua Chatbot, chúng tôi có thể cắt giảm nhân lực tại nơi mà Chatbot có thể thay thế, tăng sự tiện lợi cho cả khách hàng và công ty. Một số hệ thống chatbot tốt trên thế giới chủ yếu sử dụng tiếng Anh. Tuy nhiên, với tiếng Việt, đầu ra của các mô hình này thường không tốt do cấu trúc và ngữ pháp khác nhau. Một số đầu ra (tức là phản hồi của chatbot) không chính xác hoặc thậm chí gây hiểu lầm cho người dùng. Điều này có thể gây ra một số vấn đề nghiêm trọng trong một số ứng dụng pháp lý hoặc tài chính. Trong bài báo này, chúng tôi đã đề xuất một cách tiếp cận để đánh giá lại kết quả đầu ra của chatbot nói chung để giảm các phản hồi không chính xác. Cụ thể, chúng tôi lần đầu tiên triển khai các phương pháp tiếp cận nổi tiếng bằng chatbot tiếng Anh sang chatbot tiếng Việt. Sau đó, chúng tôi xây dựng bộ lọc để đánh giá phản hồi của chatbot dựa trên đặc điểm của ngôn ngữ tiếng Việt. Nếu phản hồi không phù hợp, hệ thống sẽ không trả lại cho người dùng. Trong trường hợp này, hệ thống có thể yêu cầu thêm thông tin để tìm phản hồi tốt hơn hoặc ít nhất là trả lại thông báo thay vì phản hồi không chính xác. Chúng tôi cũng đã phát triển một ứng dụng dựa trên web có tích hợp chatbot để đánh giá cách tiếp cận của chúng tôi trong kịch bản cuộc sống thực.

***“A Vietnamese Question Answering System”*** [21]

Hệ thống trả lời câu hỏi nhằm tạo ra câu trả lời chính xác cho câu hỏi của người dùng thay vì danh sách các tài liệu liên quan như được sử dụng bởi các công cụ tìm kiếm hiện tại. Trong bài báo này, chúng tôi đề xuất một hệ thống trả lời câu hỏi tiếng Việt dựa trên bản thể học cho phép người dùng diễn đạt câu hỏi của họ bằng ngôn ngữ tự nhiên. Theo hiểu biết của chúng tôi, đây là nỗ lực đầu tiên cho phép người dùng truy vấn cơ sở tri thức bản thể học bằng ngôn ngữ tự nhiên Việt Nam. Các thử nghiệm của hệ thống của chúng tôi trên bản thể học tổ chức cho thấy những kết quả đầy hứa hẹn.

***“BERT+vnKG:Using Deep Learning and Knowledge Graph to Improve Vietnam-ese Question Answering System”*** [22]

Hệ thống trả lời câu hỏi (QA) dựa trên xử lý ngôn ngữ tự nhiên và học sâu là một lĩnh vực nổi bật và đang được nghiên cứu rộng rãi. Mô hình Bộ nhớ Ngắn hạn Dài (LSTM) là một loạt các Mạng thần kinh tái tạo (RNN) được sử dụng phổ biến trong hệ thống dịch máy và trả lời câu hỏi. Tuy nhiên, mô hình đó chắc chắn vẫn có những khả năng hạn chế, vì vậy một mô hình mới có tên là Biểu diễn mã hóa hai chiều từ máy biến áp (BERT) đã xuất hiện để giải quyết những hạn chế này. BERT có nhiều tính năng tiên tiến hơn LSTM và cho thấy kết quả hiện đại trong nhiều nhiệm vụ, đặc biệt là trong hệ thống trả lời câu hỏi đa ngôn ngữ trong vài năm qua. Tuy nhiên, chúng tôi đã thử áp dụng mô hình BERT đa ngôn ngữ cho hệ thống QA của Việt Nam và nhận thấy rằng mô hình BERT chắc chắn vẫn có hạn chế về thời gian và độ chính xác để trả lời câu trả lời bằng tiếng Việt. Mục đích của nghiên cứu này là đề xuất một phương pháp giải quyết hạn chế trên của BERT đa ngôn ngữ và áp dụng cho hệ thống trả lời câu hỏi về du lịch ở Việt Nam. Phương pháp của chúng tôi kết hợp BERT và đồ thị kiến thức để nâng cao độ chính xác và nhanh chóng tìm ra câu trả lời. Chúng tôi đã thử nghiệm dữ liệu QA thủ công của mình về du lịch Việt Nam trên ba mô hình như LSTM, BERT tinh chỉnh đa ngôn ngữ cho QA (BERT cho QA) và BERT + vnKG. Kết quả là mô hình của chúng tôi vượt trội hơn hai mô hình trước đó về độ chính xác và thời gian. Nghiên cứu này cũng có thể được áp dụng cho các lĩnh vực khác như tài chính, thương mại điện tử, v.v...

**“A Neural Network based Vietnamese Chatbot” [23]**

Ngày nay, chatbot đang là một chủ đề nóng, các chatbot được xây dựng từ các mô hình chung đang đạt được nhiều thành công. Mục đích của bài viết này là xây dựng một chatbot tiếng Việt dựa trên mô hình seq2seq kết hợp cơ chế chú ý. Chúng tôi đã xây dựng mô hình và thử nghiệm trên khung học sâu Pytorch sử dụng GPU. Mô hình được đào tạo từ đầu đến cuối không theo quy tắc thủ công. Mô hình được xây dựng từ một tập dữ liệu nhỏ và có thể tạo phản hồi cho người dùng. Tuy nhiên, các phản hồi đã tạo vẫn cần được cải thiện để có được một cuộc trò chuyện có ý nghĩa. Chúng tôi sử dụng mô hình rất chặt chẽ dựa trên mô hình Dịch máy thần kinh của Pytorch của Sean Robertson, 2017. Đó là mô hình RNN với cơ chế chú ý cho phép bộ giải mã truy cập trực tiếp hơn vào đầu ra trạng thái ẩn của bộ mã hóa. Mạng Seq2seq là một mô hình bao gồm hai mạng nơ-ron tuần hoàn (RNN). Đầu vào là một câu hoặc một câu hỏi được nhập bởi con người, bộ mã hóa của mạng là một mạng nơ-ron tuần hoàn xuất ra một vector duy nhất và một trạng thái ẩn. Và bộ giải mã đọc vector được tạo bởi bộ mã hóa và xuất ra phản hồi.

**“NEU-chatbot: Chatbot for admission of National Economics University”**

[24]

Trong nghiên cứu này, chúng tôi đã sử dụng nền tảng Rasa để phát triển NEU-chatbot. Rasa Core và Rasa NLU là hai thành phần chính của nền tảng Rasa. Rasa Core được sử dụng để xử lý luồng hội thoại, lời nói và hành động, trong khi Rasa NLU được sử dụng để hiểu, phân loại ý định và trích xuất các thực thể của đầu vào văn bản. Nói cách khác, NLU đưa ra ý định; lõi Rasa thực hiện hành động tương ứng với nó và bot sẽ trả lời bằng hành động đó. Phần này trình bày cách chúng tôi xây dựng một chatbot Rasa cho các thắc mắc của Trường Đại học Kinh tế Quốc dân. Tất cả các bước liên tiếp như Interpreter, Tracker, Policy, Action, ngoại trừ bước đầu tiên “Message In” đều do Rasa NLU thực hiện, đều do Rasa Core xử lý. Sau khi nhận được thông báo đầu vào và sau đó được chuyển tiếp đến trình thông dịch để trích xuất ý định, thực thể và thông tin cần thiết, Rasa NLU và trình theo dõi sẽ theo dõi, phát hiện và duy trì trạng thái của ngữ cảnh hội thoại thông qua các thông báo tin nhắn đã

nhận. Sau đó, đầu ra của Trình theo dõi (trạng thái ngữ cảnh) chuyển đến trình quản lý chính sách và chính sách sẽ xác định hành động tiếp theo. Cụ thể, chatbot đã trả lời 90,29% câu hỏi một cách thích hợp vì nhiều câu hỏi không liên quan đã được đặt ra. Tuy nhiên, nếu chỉ tính các câu hỏi liên quan đến việc nhập học NEU, thì triển vọng trả lời thích hợp sẽ là tương đối 95,79%. Chúng tôi cũng đã thực hiện một cuộc khảo sát trên Facebook về mức độ hài lòng của người dùng. Kết quả cho thấy gần 98,61% trong số 1000 khách hàng đưa ra câu hỏi hài lòng với câu trả lời của chatbot này. Ở khía cạnh khác, chatbot đã mang lại một số lợi ích, chẳng hạn như số lượng chuyên gia tư vấn tuyển sinh có thể giảm 80%, nhưng chất lượng dịch vụ vượt trội vẫn được đảm bảo.

**“Understanding what the users say in chatbots: A case study for the Vietnamese language” [25]**

Bài báo này trình bày một nghiên cứu về việc hiểu những gì người dùng nói trong hệ thống chatbot: tình huống mà người dùng nhập lời nói của người dùng vào chương trình sẽ hy vọng (1) phát hiện ý định và (2) nhận ra ngữ cảnh tương ứng được ngụ ý bởi lời nói. Điều này giúp bot hiểu rõ hơn những gì người dùng đang nói và thực hiện nhiều hành động hơn. Vì vậy, chúng tôi đề xuất một khung mô hình hóa nhiệm vụ đầu tiên là bài toán phân loại và nhiệm vụ thứ hai là bài toán ghi nhãn trình tự hai lớp. Khung công tác khám phá các mạng thần kinh sâu để tự động tìm hiểu các tính năng hữu ích ở cả cấp độ ký tự và từ. Chúng tôi áp dụng khuôn khổ này để xây dựng một chatbot trong miền thương mại điện tử Việt Nam nhằm giúp các thương hiệu bán lẻ giao tiếp tốt hơn với khách hàng của họ. Kết quả thử nghiệm trên bốn bộ dữ liệu mới được xây dựng chứng minh rằng mạng nơ-ron sâu có thể hoạt động tốt hơn các phương pháp học máy thông thường mạnh mẽ. Khi phát hiện ý định, chúng tôi đạt được độ đo F tốt nhất là 82,32%. Khi trích xuất các ngữ cảnh, phương pháp đề xuất mang lại các biện pháp F đầy hứa hẹn, dao động từ 78% đến 91% tùy thuộc vào các loại ngữ cảnh cụ thể.

## **2.2.2 Các nghiên cứu ngoài nước**

**“A Technical Question Answering System with Transfer Learning” [8]**

Trong những năm gần đây, nhu cầu về các trang web trả lời câu hỏi kỹ thuật cộng đồng đã tăng lên đáng kể. Tuy nhiên, các chuyên gia về con người thường rất tốn kém để đưa ra những câu trả lời hữu ích và kịp thời trên các diễn đàn đó. Nhóm nghiên cứu gồm Wenhao Yu và các cộng sự đã phát triển TransTQA, là một hệ thống mới cung cấp các câu trả lời tự động bằng cách truy xuất các câu trả lời thích hợp dựa trên các câu hỏi tương tự được trả lời đúng trong quá khứ. TransTQA được xây dựng dựa trên mạng ALBERT tiếng xiêm, cho phép nó phản hồi nhanh chóng và chính xác. Hơn nữa, TransTQA áp dụng các tiêu chuẩn chiến lược học tập chuyên giao sâu để cải thiện khả năng hỗ trợ nhiều lĩnh vực kỹ thuật.

***“The Implementation of Question Answer System Using Deep Learning”***

[26]

Hệ thống trả lời câu hỏi được gọi là hệ thống tiên tiến có thể được sử dụng để cung cấp câu trả lời cho các câu hỏi mà người dùng yêu cầu. Vấn đề điển hình trong xử lý ngôn ngữ tự nhiên là trả lời câu hỏi tự động. Việc trả lời câu hỏi nhằm mục đích thiết kế các hệ thống có thể tự động trả lời một câu hỏi, giống như cách con người có thể tìm ra câu trả lời cho các câu hỏi. Dịch vụ trả lời câu hỏi của cộng đồng (CQA) đang trở nên phổ biến trong vài năm qua. Nó cho phép các thành viên của cộng đồng đăng bài cũng như trả lời các câu hỏi. Nó giúp người dùng nhận được thông tin từ một tập hợp các câu hỏi toàn diện được trả lời tốt. Trong hệ thống được đề xuất, một mô hình dựa trên học tập sâu được sử dụng để trả lời tự động các câu hỏi của người dùng. Đầu tiên, các câu hỏi từ tập dữ liệu được nhúng. Mạng nơ-ron sâu được đào tạo để tìm ra sự giống nhau giữa các câu hỏi. Câu trả lời tốt nhất cho mỗi câu hỏi được tìm thấy là câu trả lời có điểm tương đồng cao nhất. Mục đích của hệ thống được đề xuất là thiết kế một mô hình giúp nhận được câu trả lời của một câu hỏi một cách tự động. Hệ thống được đề xuất sử dụng thuật toán phân cụm phân cấp để phân nhóm các câu hỏi.

***“Evaluating the Performance of Recurrent Neural Network based Question Answering System with Easy and Complex bAbI QA Tasks”*** [27]

Hệ thống trả lời câu hỏi (QA) là một lĩnh vực xử lý ngôn ngữ tự nhiên, cho phép người dùng đặt câu hỏi bằng cách sử dụng câu ngôn ngữ tự nhiên và trả lại câu

trả lời ngắn gọn cho người dùng thay vì danh sách tài liệu. Công việc này dự định sử dụng các thuật toán Học sâu dựa trên Mạng thần kinh tái tạo (RNN) để giải quyết vấn đề Hệ thống trả lời câu hỏi. Việc sử dụng mạng nơ-ron tuần hoàn cho phép chúng tôi mở rộng và áp dụng mô hình này cho nhiều nhiệm vụ trả lời câu hỏi khác nhau. Trong công việc này, Hệ thống trả lời câu hỏi đơn giản dựa trên RNN được triển khai và hiệu suất của nó được đánh giá với các tác vụ trả lời câu hỏi đơn giản và phức tạp bằng cách sử dụng tập dữ liệu bAbI. Việc thực hiện đào tạo và kiểm tra với các thước đo phù hợp được nghiên cứu và quan sát thấy sự khác biệt về hiệu suất trong hai nhiệm vụ trả lời câu hỏi.

***“Deep learning based question answering system in Bengali”*** [28]

Những tiến bộ gần đây trong lĩnh vực xử lý ngôn ngữ tự nhiên đã cải thiện hiệu suất hiện đại trong nhiều nhiệm vụ bao gồm trả lời câu hỏi cho các ngôn ngữ như tiếng Anh. Tiếng Bengali được xếp hạng thứ bảy và được khoảng 300 triệu người trên thế giới sử dụng. Nhưng do thiếu dữ liệu và nghiên cứu tích cực về QA nên tiến độ tương tự đã không đạt được đối với tiếng Bengali. Không giống như tiếng Anh, không có bộ dữ liệu QA quy mô lớn chuẩn nào được thu thập cho tiếng Bengali, không có mô hình ngôn ngữ được đào tạo trước nào có thể được sửa đổi để trả lời câu hỏi bằng tiếng Bengali và cũng không có điểm cơ bản nào của con người cho QA. Trong công trình này, chúng tôi sử dụng các mô hình máy biến áp hiện đại để đào tạo hệ thống QA trên một tập dữ liệu đọc hiểu tổng hợp được dịch từ một trong những tập dữ liệu điểm chuẩn phổ biến nhất bằng tiếng Anh có tên là SQuAD 2.0. Chúng tôi thu thập tập dữ liệu QA có chú thích của con người nhỏ hơn từ Wikipedia tiếng Bengali với các chủ đề phổ biến từ văn hóa Bangladesh để đánh giá các mô hình của chúng tôi. Cuối cùng, chúng tôi so sánh các mô hình của chúng tôi với trẻ em của con người để thiết lập điểm chuẩn bằng cách sử dụng các thử nghiệm khảo sát.

***“Code Mixed Question Answering Challenge using Deep Learning Methods”*** [29]

Trong phương tiện truyền thông xã hội, các câu hỏi bằng ngôn ngữ trộn mã (kết hợp hai ngôn ngữ riêng biệt) đang trở thành phương thức biểu đạt và giao tiếp được ưa chuộng. Trong twitter, mọi người có thể sử dụng ngôn ngữ hỗn hợp, nơi các

cá nhân thường sử dụng tiếng mẹ đẻ của họ cùng với tiếng Anh. Nhưng hệ thống trả lời câu hỏi không hỗ trợ ngôn ngữ hỗn hợp. Họ chỉ hoạt động trên một ngôn ngữ duy nhất như tiếng Anh, tiếng Pháp và tiếng Đức. Bài báo này trình bày một khung trả lời câu hỏi trực tuyến cho các ngôn ngữ hỗn hợp. Hệ thống này phân tích truy vấn ngôn ngữ hỗn hợp của người dùng và đưa ra câu trả lời, đồng thời trình bày một liên quan giúp hỗ trợ giải mã tiếng Anh pha trộn với 3 ngôn ngữ Ấn Độ khác nhau: Hindi, Telugu và Tamil. Các từ hỗn hợp mã sẽ được dịch sang tiếng Anh để giảm độ phức tạp của dữ liệu. Sau đó, quá trình tạo lồng được thực hiện trên chúng. Các thuật toán học sâu như Mạng thần kinh tái tạo (RNN), Mạng chú ý phân cấp (HAN) được sử dụng để phân loại câu hỏi. Ma trận nhầm lẫn được sử dụng làm thước đo đánh giá RNN, HAN. Hệ thống này được hỗ trợ trên web và trong tương lai, nó có thể được sử dụng để thu thập thêm dữ liệu câu hỏi và câu trả lời hỗn hợp mã để cải thiện hơn nữa.

***“A building regulation question answering system: A deep learning methodology”*** [30]

Các quy định đóng một vai trò quan trọng trong việc đảm bảo chất lượng xây dựng của một tòa nhà và giảm thiểu các tác động xấu đến môi trường của nó. Các kỹ sư và những người tương tự cần truy xuất thông tin quy định để đảm bảo tòa nhà tuân thủ các tiêu chuẩn quy định. Mặc dù có sẵn các công cụ tìm kiếm và cơ sở dữ liệu kỹ thuật số có thể được sử dụng để lưu trữ các quy định, ví dụ: các kỹ sư không thể truy xuất thông tin cho các nhu cầu cụ thể của miền một cách kịp thời. Do đó, người dùng thường phải đối phó với gánh nặng duyệt và lọc thông tin, đây có thể là một quá trình tốn nhiều thời gian. Nghiên cứu này phát triển một phương pháp luận end-to-end mạnh mẽ để nâng cao hiệu quả và hiệu quả của việc truy xuất các truy vấn liên quan đến các quy định xây dựng. Phương pháp đã phát triển tích hợp truy xuất thông tin với mô hình học sâu về Xử lý ngôn ngữ tự nhiên (NLP) để cung cấp câu trả lời chính xác và nhanh chóng cho các câu hỏi của người dùng từ tập hợp các quy định xây dựng. Phương pháp luận được đánh giá và một hệ thống nguyên mẫu để truy xuất các truy vấn được phát triển. Do đó, đóng góp của bài báo sẽ tăng gấp đôi khi nó phát triển một: (1) phương pháp kết hợp NLP và học sâu để có thể giải quyết các thách thức

được nêu ra về các quy định của tòa nhà; và (2) chatbot của hệ thống trả lời câu hỏi, mà chúng tôi gọi là QAS4CQAR. Phương pháp được đề xuất của chúng tôi có khả năng đại diện và học tập tính năng mạnh mẽ và do đó có thể được áp dụng để xây dựng các quy định ở các khu vực pháp lý khác.

***“Knowledge Base Question Answering Based on Deep Learning Models”***

[31]

Bài báo này tập trung vào nhiệm vụ trả lời câu hỏi dựa trên kiến thức (KBQA). KBQA nhằm mục đích -khớp các câu hỏi với ngữ nghĩa có cấu trúc trong cơ sở kiến thức. Trong bài báo này, chúng tôi đề xuất một phương pháp hai giai đoạn. Thứ nhất, chúng tôi đề xuất mô hình trích xuất thực thể chủ đề (TEEM) để trích xuất thực thể chủ đề trong các câu hỏi, mô hình này không dựa vào các tính năng hoặc công cụ ngôn ngữ được tạo thủ công. Chúng tôi trích xuất các thực thể chủ đề trong các câu hỏi với TEEM và sau đó tìm kiếm bộ ba kiến thức có liên quan đến các thực thể chủ đề từ cơ sở kiến thức như bộ ba kiến thức ứng viên. Sau đó, chúng tôi áp dụng Mô hình ngữ nghĩa có cấu trúc sâu dựa trên mạng nơ-ron tích tụ và bộ nhớ ngắn hạn dài hạn hai chiều để khớp các câu hỏi và vị từ trong bộ ba kiến thức ứng viên. Để có được tập dữ liệu đào tạo tốt hơn, chúng tôi sử dụng cách tiếp cận lặp lại để truy xuất bộ ba kiến thức từ cơ sở kiến thức. Kết quả đánh giá cho thấy hệ thống của chúng tôi đạt được số đo Average F1 là 79,57% trên tập dữ liệu thử nghiệm.

***“Deep learning-based question answering system for intelligent humanoid robot”*** [32]

Sự phát triển của Robot hình người thông minh tập trung vào hệ thống trả lời câu hỏi có thể tương tác với con người là rất hạn chế. Trong nghiên cứu này, chúng tôi muốn đề xuất một Robot hình người thông minh có khả năng tự học để chấp nhận và đưa ra phản hồi từ mọi người dựa trên nền tảng kiến thức về Học sâu và Dữ liệu lớn. Loại robot này có thể được sử dụng rộng rãi trong các khách sạn, trường đại học và các dịch vụ công cộng. Robot hình người nên xem xét kiểu câu hỏi và kết luận câu trả lời thông qua cuộc trò chuyện giữa người máy và người dùng. Trong kịch bản của chúng tôi, rô bốt sẽ phát hiện khuôn mặt của người dùng và chấp nhận lệnh từ người dùng để thực hiện một hành động.



Câu hỏi từ người dùng sẽ được xử lý bằng cách sử dụng học sâu và kết quả sẽ được so sánh với cơ sở kiến thức trên hệ thống. Chúng tôi đã đề xuất phương pháp Học tập sâu của mình, dựa trên bộ mã hóa Mạng thần kinh tái tạo (RNN), bộ mã hóa Mạng thần kinh Convolution (CNN), với Luồng chú ý hai chiều (BiDAF).

Đánh giá của chúng tôi chỉ ra rằng việc sử dụng bộ mã hóa dựa trên RNN với BiDAF cho điểm cao hơn so với bộ mã hóa CNN với BiDAF. Dựa trên thử nghiệm của chúng tôi, mô hình của chúng tôi nhận được 82,43% điểm F1 và bộ mã hóa dựa trên RNN sẽ cho điểm EM / F1 cao hơn so với sử dụng bộ mã hóa CNN.

**“MedFuseNet: An attention-based multimodal deep learning model for visual question answering in the medical domain” [33]**

Hình ảnh y tế rất khó để hiểu đối với một người không có chuyên môn. Sự khan hiếm nhân viên y tế trên toàn cầu thường phải đối mặt với vấn đề mệt mỏi về thể chất và tinh thần do số lượng ca bệnh cao, gây ra sai sót cho con người trong quá trình chẩn đoán. Trong những tình huống như vậy, có thêm ý kiến có thể hữu ích trong việc nâng cao lòng tin của người ra quyết định. Do đó, điều quan trọng là phải có một hệ thống trả lời câu hỏi trực quan đáng tin cậy (VQA) để cung cấp 'ý kiến hứ hai' về các trường hợp y tế. Tuy nhiên, hầu hết các hệ thống VQA hoạt động ngày nay đều phục vụ cho các vấn đề trong thế giới thực và không được thiết kế riêng để xử lý hình ảnh y tế. Hơn nữa, hệ thống VQA cho hình ảnh y tế cần phải xem xét một lượng hạn chế dữ liệu đào tạo có sẵn trong miền này. Trong bài báo này, chúng tôi phát triển MedFuseNet, một mô hình học sâu đa phương thức dựa trên sự chú ý, cho VQA trên các hình ảnh y tế có tính đến các thách thức liên quan. MedFuseNet của chúng tôi nhằm mục đích tối đa hóa việc học với độ phức tạp tối thiểu bằng cách chia câu hỏi vấn đề thành các nhiệm vụ đơn giản hơn và dự đoán câu trả lời. Chúng tôi giải quyết hai loại dự đoán câu trả lời — phân loại và tạo. Chúng tôi đã tiến hành một loạt các phân tích định lượng và định tính để đánh giá hiệu suất của MedFuseNet. Các thử nghiệm của chúng tôi chứng minh rằng MedFuseNet hoạt động tốt hơn các phương pháp VQA hiện đại và việc hình dung các chú ý được thu thập cho thấy khả năng không thể nghe được của các kết quả dự đoán của mô hình của chúng tôi.

***“Deep learning of grammatically-interpretable representations through question-answering” [34]***

Chúng tôi giới thiệu một kiến trúc trong đó các biểu diễn bên trong - được học bằng cách tối ưu hóa end-to-end trong mạng nơ-ron sâu thực hiện nhiệm vụ trả lời câu hỏi dạng văn bản — có thể được giải thích bằng cách sử dụng các khái niệm cơ bản từ lý thuyết ngôn ngữ. Khả năng diễn giải này có chi phí chỉ giảm một vài điểm phần trăm về độ chính xác so với mô hình ban đầu dựa trên mô hình mới (BiDAF [1]). Biểu diễn bên trong được hiểu là một thông báo Tensor Product Repre: đối với mỗi từ đầu vào, mô hình sẽ chọn một biểu tượng để mã hóa từ đó và một vai trò trong đó đặt biểu tượng và liên kết hai từ đó lại với nhau. Việc lựa chọn thông qua sự chú ý nhẹ nhàng. Diễn giải tổng thể được xây dựng từ các diễn giải của các ký hiệu, như được mô hình được đào tạo tuyển dụng và diễn giải về các vai trò được sử dụng bởi mô hình. Chúng tôi nhận thấy sự ủng hộ cho giả thuyết ban đầu của chúng tôi rằng các biểu tượng có thể được hiểu là nghĩa của từ từ vựng-ngữ nghĩa, trong khi các vai trò có thể được hiểu là các vai trò (hoặc danh mục) ngữ pháp gần đúng như chủ đề, wh-từ, xác định, v.v. Thông qua cực kỳ chi tiết, tốt -phân tích chuyên sâu, chúng tôi tìm thấy sự tương ứng cụ thể giữa các vai trò đã học và các phần của lời nói như được chỉ định bởi trình phân tích cú pháp tiêu chuẩn [2], và tìm thấy một số điểm khác biệt có lợi cho mô hình. Theo nghĩa này, mô hình học các khía cạnh quan trọng của ngữ pháp, sau khi chỉ tiếp xúc với văn bản, câu hỏi và câu trả lời không có chú thích ngôn ngữ: không có kiến thức ngôn ngữ nào trước đó được cung cấp cho mô hình.

***RobotVQA — A Scene-Graph- and Deep-Learning-based Visual Question Answering System for Robot Manipulation” [35]***

Nhận thức về thị giác của rô-bốt là một thách thức đối với việc điều khiển rô-bốt thành công trong môi trường ồn ào, lộn xộn và năng động. Trong khi một số hệ thống nhận thức không cung cấp ngữ nghĩa đầy đủ về hiện trường, những hệ thống khác lại không đưa ra được các mô hình học tập và dữ liệu đào tạo thích hợp. Một vấn đề chính khác gặp phải trong một số hệ thống nhận thức của robot là chúng không có khả năng phản ứng kịp thời với các chương trình điều khiển robot mà thời gian thực của nó là rất quan trọng. Kiến trúc bao gồm một trình tạo tập dữ liệu đào tạo và

một trình mô tả cảnh có thể học được, được đặt tên là RobotVQA dành cho Trả lời câu hỏi bằng hình ảnh robot. Kiến trúc này tận dụng sức mạnh của học sâu để dự đoán và đào tạo các thế giới ảo như ảnh thực tế. RobotVQA nhận làm đầu vào hình ảnh RGB hoặc RGBD của cảnh robot, phát hiện tất cả các đối tượng có liên quan trong đó, sau đó mô tả trong thời gian thực từng đối tượng về danh mục, màu sắc, chất liệu, hình dạng, khả năng hoạt động, tạo dáng 6D và mặt nạ phân đoạn. Hơn nữa, RobotVQA tính toán các quan hệ không gian định tính giữa các đối tượng đó. Chúng tôi đề cập đến mô tả cảnh như vậy trong bài báo này dưới dạng đồ thị cảnh hoặc đồ thị ngữ nghĩa của cảnh. Trong RobotVQA, dự đoán và đào tạo diễn ra theo một cách thống nhất. Cuối cùng, chúng tôi chứng minh cách RobotVQA phù hợp với các hệ thống điều khiển robot diễn giải nhận thức như một quá trình trả lời câu hỏi.

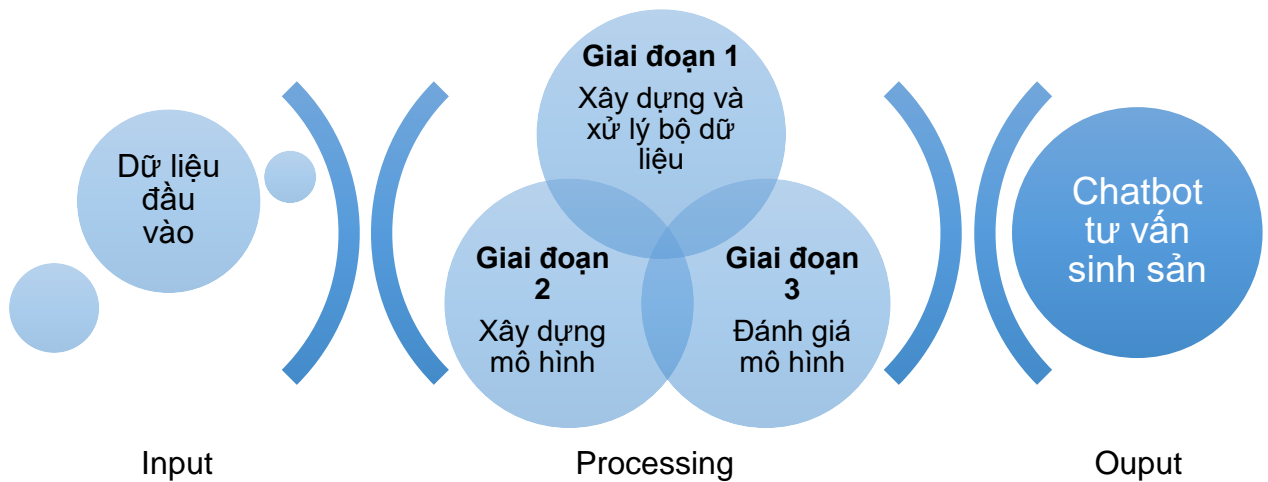
## CHƯƠNG 3. ĐỀ XUẤT PHƯƠNG PHÁP XÂY DỰNG BỘ DỮ LIỆU VÀ MÔ HÌNH CHO BÀI TOÁN

### 3.1 Giới thiệu chương 3

Chương này tác giả tập trung giới thiệu về cách xây dựng nên bộ dữ liệu đầu vào để làm dữ liệu training cho mô hình từ chuỗi văn bản. Bên cạnh đó nêu lên đề xuất của mình về phương pháp thực hiện xây dựng mô hình của bài toán bằng cách áp dụng các thư viện Keras, Tensorflow của Machine Learning. Cuối cùng đánh giá kết quả và thử nghiệm thực tế.

### 3.2 Đề xuất phương pháp và thuật toán xử lý

Quy trình của phương pháp được đề xuất như sau:



Hình 3.1: Quy trình bài toán

#### 3.2.1 Input: dữ liệu tư vấn sinh sản.

Để huấn luyện cho mô hình, luận văn sẽ sử dụng dữ liệu về khía cạnh tư vấn sinh sản. Đầu tiên, tiến hành nghiên cứu xu hướng hỏi đáp của chị em phụ nữ về việc tư vấn trước, trong và sau thời kỳ mang thai. Các định nghĩa, lý thuyết về những căn bệnh hay triệu chứng có thể gặp phải xuyên suốt khoảng thời gian này. Thông tin cần thiết của bộ dữ liệu chính là các khái niệm, nguyên nhân, triệu chứng, dấu hiệu, các yếu tố nguy cơ, các phương pháp xử lý, phòng ngừa một số vấn đề về sinh sản được

cung cấp từ bệnh viện Hồng Hưng. Nội dung này sẽ được làm rõ trong chương 4 của luận văn.

### **3.2.2 Processing**

Bao gồm 3 giai đoạn:

- **Giai đoạn 1:** Từ văn bản thô đầu vào, xây dựng bộ dữ liệu trả lời tự động với cấu trúc gồm 3 phần: tags (chứa nhãn của cuộc hội thoại), input (chứa câu hỏi mà người dùng có thể đặt ra), responses (chứa những phản hồi từ hệ thống). Ba phần này sẽ tạo thành một nhóm câu hỏi – đáp, bộ dữ liệu sẽ gồm nhiều nhóm thế này kết hợp lại với nhau. Sau khi xây dựng thành công bộ dữ liệu sẽ tiến hành xử lý bộ dữ liệu bằng cách làm sạch và mã hóa chúng với các phương thức được cung cấp bởi Keras của Tensorflow trước khi đưa vào mô hình làm dữ liệu đào tạo.
- **Giai đoạn 2:** Xây dựng mô hình dự đoán cho bài toán. Mô hình LTSM được giới thiệu ở chương 2 sẽ được sử dụng để giúp học những câu hỏi từ đầu vào ở giai đoạn 1 và đưa ra dự đoán chính xác. Kèm theo LTSM, luận văn cũng sử dụng thêm lớp Embedding của Keras để xử lý chuỗi văn bản và lớp Flatten để làm phẳng đầu ra của dữ liệu sau khi đi qua lớp LTSM.
- **Giai đoạn 3:** Đánh giá độ chính xác của mô hình. Luận văn sử dụng chỉ số đánh giá là độ chính xác (accuracy) để tiến hành đánh giá mô hình. Độ chính xác càng cao cho thấy mô hình càng chính xác.

### **3.2.3 Output**

Từ mô hình đã xây dựng, áp dụng vào chatbot và tiến hành kiểm thử để quan sát kết quả dự đoán. Mô hình hoạt động hiệu quả khi đưa ra câu trả lời đúng hoặc gần đúng với câu hỏi được nhập từ người dùng.

## CHƯƠNG 4. CÀI ĐẶT VÀ THỰC NGHIỆM

### 4.1 Giới thiệu chương 4

Chương này sẽ trình bày chi tiết việc xây dựng bộ dữ liệu huấn luyện và quá trình cụ thể cài đặt mô hình cho thuật toán. Mô hình cài đặt và thực nghiệm của luận văn được thực hiện bằng ngôn ngữ Python trên Google Colaboratory [36], chi tiết của phần code này sẽ được liệt kê ở phụ lục của luận văn.

### 4.2 Bộ dữ liệu của bài toán

#### 4.2.1 Cơ sở lý thuyết của bộ dữ liệu

Với mục tiêu xây dựng hệ thống trả lời tự động, luận văn tập trung vào khía cạnh tư vấn sinh sản cho phụ nữ sắp mang thai, đang trong thời gian mang thai và những thai phụ đang trải qua thời kỳ hậu sản. Từ những nội dung đã tìm hiểu về chủ đề tư vấn sinh sản, luận văn tiếp tục nghiên cứu thị hiếu của người dùng, các thắc mắc thường hay gặp phải để tiến hành xây dựng bộ dữ liệu. Vì chủ đề này, luận văn tập trung nghiên cứu các câu hỏi xoay quanh việc tư vấn sinh sản cụ thể với các thông tin nghiên cứu được từ bệnh viện Hồng Hưng cụ thể như sau:

- Các thông tin về khoa Sản của bệnh viện, dịch vụ, kỹ thuật mà khoa cung cấp.
- Khám thai: tập trung nghiên cứu về các mốc giai đoạn quan trọng cần đi khám thai
- Dinh dưỡng thai kỳ: nghiên cứu nhu cầu dinh dưỡng của thai phụ trong thời gian mang thai, thực phẩm nên tránh khi mang thai.
- Ốm nghén: thời gian thai phụ gặp tình trạng ốm nghén, biểu hiện, lời khuyên.
- Chi phí các gói sinh tại bệnh viện
- Động thai (dọa sảy thai): nghiên cứu các khái niệm, nguyên nhân, dấu hiệu, cách xử lý, lưu ý, tư thế nằm, món ăn cần thiết, biện pháp chăm sóc và phòng ngừa.
- Sảy thai: nghiên cứu các khái niệm, triệu chứng, chuẩn đoán và điều trị đồng thời cũng tìm hiểu cách phòng ngừa và làm giảm nguy cơ sảy thai cho lần mang thai sau.

- Mang thai ngoài tử cung: nghiên cứu các khái niệm, nguyên nhân, triệu chứng, dấu hiệu, các yếu tố nguy cơ, phương pháp điều trị.
- Nạo phá thai: những điều cần lưu ý, các phương pháp phá thai, hậu quả.
- Hậu sản: các bệnh hậu sản thường gặp
- Các trường hợp đặc biệt khi sinh con như: sinh non, sinh già tháng, sinh bọc, sinh mổ. Nghiên cứu tập trung vào nguyên nhân, nguy hiểm, biến chứng và các câu hỏi liên quan.
- Chăm sóc mẹ sau sinh: chứng bệnh trầm cảm sau sinh với những nguyên nhân, dấu hiệu và nguy hiểm căn bệnh này mang lại.
- Và nhiều nội dung khác...

#### **4.2.2 Xây dựng bộ dữ liệu**

Từ những thông tin thu được từ quá trình tìm tòi nghiên cứu lý thuyết về chủ đề tư vấn sinh sản. Tiếp theo, luận văn tiến hành xây dựng bộ dữ liệu đào tạo. Bộ dữ liệu được xây dựng thành file json chứa các đoạn đối thoại. Mỗi đoạn đối thoại sẽ gồm 3 thành phần:

- Tag: nhãn được sử dụng để phân loại các đầu vào và ánh xạ chúng tới một loại phản hồi cụ thể.
- Input: là những thông điệp mà người dùng sẽ gửi đến bot (chứa nội dung cần tư vấn)
- Response: sau khi ánh xạ đầu vào cho một thể thích hợp, ta có thể chọn một trong các phản hồi để trả lại cho người dùng.

Với cấu trúc trên luận văn đã xây dựng được 103 đề tài, 232 input từ người dùng, và 103 response. Ta có thể dễ dàng hình dung bộ dữ liệu xây dựng thông qua hình ảnh dưới đây. Mỗi input từ người dùng đều sẽ được gắn với nhãn tương ứng của nó để phân loại.



The screenshot shows a Jupyter Notebook interface with a variable named 'data'. The variable contains a list of dictionaries, each representing a row of data. The columns are 'inputs' and 'tags'. The first five rows are related to greetings, and the last five rows are related to placental abruption symptoms and treatments.

	inputs	tags
0	xin chào	Greeting
1	chào	Greeting
2	bạn có thể giúp tôi không ?	Greeting
3	tôi cần giúp đỡ	Greeting
4	tôi cần tư vấn	Greeting
...	...	...
227	biến chứng của rau bong non	DangerOfAblatioPlacentae
228	triệu chứng của nhau bong non	SignsOfAblatioPlacentae
229	dấu hiệu của rau bong non	SignsOfAblatioPlacentae
230	điều trị rau bong non	TreatmentAblatioPlacentae
231	điều trị nhau bong non như thế nào	TreatmentAblatioPlacentae

232 rows × 2 columns

**Hình 4.1: Mô tả bộ dữ liệu**

### 4.2.3 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là một trong những bước quan trọng nhất khi giải quyết bất kỳ bài toán nào trong lĩnh vực học máy. Để mô hình có thể đưa ra kết quả có độ chính xác cao thì bộ dữ liệu luôn cần được xử lý, làm sạch và biến đổi trước khi trở thành dữ liệu huấn luyện cho mô hình học máy. Đối với luận văn này, ta quan tâm đến input của người dùng nhập vào và bộ dữ liệu đã được xây dựng trước đó. Để tránh việc dữ liệu input của người dùng không đạt chuẩn, ta tiến hành làm sạch bằng 2 bước sau đây.

1. Loại bỏ các dấu câu, các ký hiệu đặc biệt khỏi input

```
! "# $% & '() * +, -./:;<=>?@[\\[_`{|}~
```

2. Chuyển các giá trị thành chữ thường (lowercase)

Tiếp theo, ta tiến hành xử lý bộ dữ liệu tư vấn sinh sản. Ở đây ta sẽ áp dụng thuật toán *Tokenizer* [37] đây là một nhánh con trong tập xử lý ngôn ngữ tự nhiên. Tokenizer cho phép ta vector hóa một kho ngữ liệu văn bản, bằng cách biến mỗi văn bản thành một chuỗi các số nguyên (mỗi số nguyên là chỉ mục của một mã thông báo



trong từ điển) hoặc thành một vectơ trong đó hệ số cho mỗi mã thông báo có thể là nhị phân, dựa trên số từ, dựa trên tf-idf ...

Với bộ dữ liệu này, luận văn quyết định chỉ tối đa 5000 từ được giữ lại, dựa trên tần suất của từ. Chỉ *num\_words-1* từ phổ biến nhất được giữ lại. Sau đó tiến hành cập nhật từ vựng dựa trên danh sách các input. Tiếp theo là vec-tơ hóa từng input của danh sách các input thành *chuỗi các số nguyên (sequences)*. Tiếp đó với chuỗi các số nguyên với độ dài ngắn khác nhau do ảnh hưởng bởi số lượng từ của mỗi câu sẽ được chuyển về cùng một độ dài với hàm *pad\_sequences* [38] của Keras để tạo nên sự nhất quán về dữ liệu. Bước cuối cùng ta tiến hành mã hóa các nhãn bằng cách sử dụng *LabelEncoder* [39] của thư viện scikitlearn, ở đây chính là các *tags*.

```
[14] input_shape = x_train.shape[1]
      print(input_shape)
```

```
14
```

```
[15] #define vocabulary
      vocabulary = len(tokenizer.word_index)
      print("number of unique words : ",vocabulary)
      output_length = le.classes_.shape[0]
      print("output length: ",output_length)
```

```
number of unique words : 275
output length: 103
```

**Hình 4.2: Kết quả xử lý dữ liệu**

Sau các bước xử lý dữ liệu, ta nhận được số chiều của bộ dữ liệu là 14, với 275 từ độc nhất, và đầu ra là 103.

### 4.3 Xây dựng mô hình

Mô hình với đầu vào – lớp *Input* với số chiều được tính toán ở bước trên, sẽ bao gồm thêm một lớp *Embedding* để tạo véc tơ nhúng cho mỗi từ trong câu, đầu ra của lớp này là đầu vào của lớp recurrent với công *LSTM*. Sau đó, đầu ra này tiếp tục trở thành đầu vào của lớp *Flatten* với mục làm phẳng đầu ra của lớp LSTM: chuyển

đổi mảng nhiều chiều thành một chiều. Và cuối cùng một lớp *Dense* được sử dụng làm đầu ra cho mô hình với hàm kích hoạt là *softmax*.

Xây dựng mô hình thành công, ta tiến hành biên dịch mô hình với các thiết lập về thông số như: hàm mất mát là *sparse\_categorical\_crossentropy*, sử dụng thuật toán *adam* để tối ưu hóa mô hình kèm theo chỉ số độ chính xác *accuracy* để quan sát.

Sau khi thiết lập, chạy huấn luyện cho mô hình với epochs = 300. Ta được kết quả như sau:

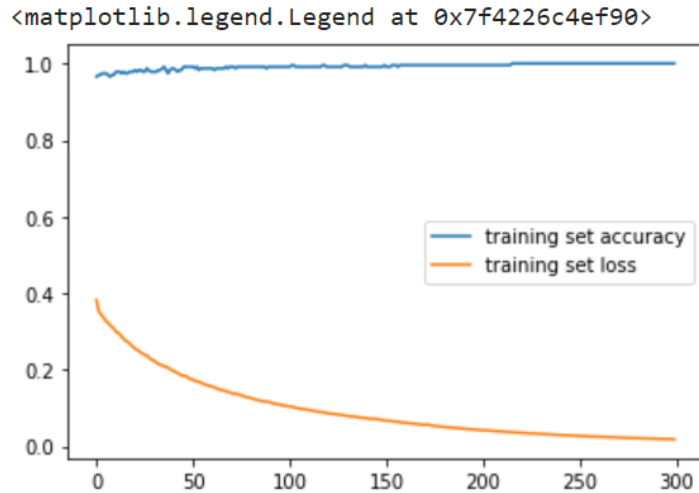
```
Epoch 290/300
8/8 [=====] - 0s 7ms/step - loss: 0.0195 - accuracy: 1.0000
Epoch 291/300
8/8 [=====] - 0s 7ms/step - loss: 0.0193 - accuracy: 1.0000
Epoch 292/300
8/8 [=====] - 0s 7ms/step - loss: 0.0192 - accuracy: 1.0000
Epoch 293/300
8/8 [=====] - 0s 8ms/step - loss: 0.0190 - accuracy: 1.0000
Epoch 294/300
8/8 [=====] - 0s 7ms/step - loss: 0.0192 - accuracy: 1.0000
Epoch 295/300
8/8 [=====] - 0s 7ms/step - loss: 0.0186 - accuracy: 1.0000
Epoch 296/300
8/8 [=====] - 0s 7ms/step - loss: 0.0189 - accuracy: 1.0000
Epoch 297/300
8/8 [=====] - 0s 7ms/step - loss: 0.0186 - accuracy: 1.0000
Epoch 298/300
8/8 [=====] - 0s 11ms/step - loss: 0.0184 - accuracy: 1.0000
Epoch 299/300
8/8 [=====] - 0s 7ms/step - loss: 0.0183 - accuracy: 1.0000
Epoch 300/300
8/8 [=====] - 0s 7ms/step - loss: 0.0180 - accuracy: 1.0000
```

### Hình 4.3: Huấn luyện cho mô hình

Sau khi chạy huấn luyện cho mô hình, ta quan sát thấy độ chính xác rất cao đạt tới 100% và độ mất mát nhỏ giảm dần chỉ đạt 1.8%. Đây là một kết quả rất khả quan thể hiện mô hình có tỉ lệ trả lời đúng rất cao và ít trường hợp trả lời sai. Tuy nhiên để xác định chính xác liệu mô hình có hoạt động hiệu quả hay không, ta sẽ tiến hành xây dựng chatbot đơn giản áp dụng dự đoán từ mô hình để đưa ra kết quả trả lời một cách tự động với đầu vào ngẫu nhiên của người dùng về vấn đề tư vấn sinh sản.

## 4.4 Đánh giá mô hình và áp dụng vào chatbot

Đầu tiên ta in ra kết quả huấn luyện mô hình để có đánh giá ban đầu về mô hình. Ở đây luận văn đánh giá mô hình dựa trên độ chính xác, ta tiến hành vẽ biểu đồ để quan sát kết quả huấn luyện của mô hình.



**Hình 4.4: Biểu đồ miêu tả độ chính xác của mô hình**

Từ biểu đồ ta quan sát thấy độ chính xác ban đầu của mô hình rất cao, gần với ngưỡng 100% và có độ mất mát giảm dần từ 40% về 0%. Để đảm bảo giá trị này là hợp lý ta sẽ ứng dụng vào chatbot để đưa ra phản hồi từ hệ thống.

Áp dụng mô hình xây dựng vào chatbot để tư vấn cho 3 vấn đề: bị động thai, khám thai và bệnh trầm cảm sau sinh.

#### - Tư vấn bị động thai

You: xin chào, tôi cần tư vấn  
 Bệnh viện Hồng Hưng: Xin chào, đây là hệ thống trả lời tự động về tư vấn sinh sản. Tôi có thể giúp gì cho bạn?  
 You: tôi muốn biết nguyên do bị động thai  
 Bệnh viện Hồng Hưng:  
 • Trong 3 tháng đầu tiên: Xảy ra sự bất thường về nhiễm sắc thể và gen ở thai nhi hay bào thai; Người mẹ lớn tuổi; Thai nhi bất thường; Người mẹ bị ti  
 • 3 tháng tiếp theo: Người mẹ mắc chứng cao huyết áp; Lượng đường trong máu không được kiểm soát; Thai phụ mắc bệnh ban đỏ, bệnh thận; Những người có  
 You: nên làm gì khi bị động thai  
 Bệnh viện Hồng Hưng: Ngay khi nhận thấy cơ thể có dấu hiệu động thai, thai phụ cần được di chuyển đến phòng ngủ và nằm nghỉ ngơi, tránh di chuyển và  
 You: món ăn tốt cho cơ thể khi bị động thai là gì  
 Bệnh viện Hồng Hưng: Cháo cá chép giúp bổ sung dinh dưỡng. Cháo bí ngô giúp bổ máu. Cháo gà gạo nếp giúp an thai.  
 You: xin cảm ơn  
 Bệnh viện Hồng Hưng: Tạm biệt

**Hình 4.5: Chatbot tư vấn về vấn đề động thai**

#### - Tư vấn về việc khám thai

☞ You: mình cần giúp đỡ ạ  
 Bệnh viện Hồng Hưng: Xin chào, đây là hệ thống trả lời tự động về tư vấn sinh sản. Tôi có thể giúp gì cho bạn?  
 You: khi nào nên đi khám thai ạ  
 Bệnh viện Hồng Hưng:  
 • 11 đến 14 tuần: Kiểm tra các dị tật ở thai nhi.  
 • 16 đến 20 tuần: Kiểm tra sự phát triển của thai nhi và các dị tật bẩm sinh bằng các xét nghiệm chính xác hơn.  
 • 20 đến 24 tuần: kiểm tra hình thái thai nhi, tầm soát các bất thường và kiểm tra vị trí bám của nhau thai, lượng nước ối.  
 • 24 đến 28 tuần: Kiểm tra sự bất đồng nhóm máu, sự thay đổi bất thường trên cơ thể mẹ có khả năng ảnh hưởng đến thai nhi.  
 • 29 đến 36 tuần: mỗi 2 tuần khám 1 lần. Kiểm tra ngôi thai, sự phát triển của thai, và tiêm phòng uốn ván.  
 • 36 đến 40 tuần: mỗi tuần khám 1 lần. Kiểm tra tử cung và tư vấn các dấu hiệu sắp sinh.  
 You: bổ sung dinh dưỡng trong 3 tháng đầu mang thai sao ạ  
 Bệnh viện Hồng Hưng:  
 • Năng lượng: Trung bình, bà mẹ mang thai trong 3 tháng đầu cần cung cấp khoảng 2300 - 2400 kcal/ngày;  
 • Axit folic: Mẹ bầu có thể bổ sung axit folic qua các loại thực phẩm hoặc có thể dùng thêm viên uống bổ sung axit folic theo hướng dẫn của bác sĩ;  
 • Protein: Thai phụ cần khoảng 85 - 90g protein/ngày, cao hơn bình thường 10-15g/ngày;  
 • Sắt: cần cung cấp 36 - 40mg sắt mỗi ngày.  
 • Vitamin A: cần được cung cấp đủ 600mcg vitamin A/ngày.  
 • Canxi và vitamin D, Vitamin C.  
 • Các nguyên tố vi lượng: Magie, selen, i-ốt, kẽm, vitamin nhóm B, DHA/EPA, ...  
 You: xin cảm ơn  
 Bệnh viện Hồng Hưng: Tạm biệt

### Hình 4.6: Chatbot tư vấn về vấn đề khám thai

#### - Tư vấn biểu hiện trầm cảm sau sinh.

You: hi  
 Bệnh viện Hồng Hưng: Xin chào, đây là hệ thống trả lời tự động về tư vấn sinh sản. Tôi có thể giúp gì cho bạn?  
 You: tôi muốn biết bệnh trầm cảm khi sinh con  
 Bệnh viện Hồng Hưng: Bệnh trầm cảm sau sinh con là tình trạng người phụ nữ bị rối loạn cảm xúc, hay có suy nghĩ tiêu cực, mệt mỏi, buồn chán và lo lắng.  
 You: trầm cảm sau sinh có biểu hiện gì ko  
 Bệnh viện Hồng Hưng:  
 • Suy nhược cơ thể;  
 • Lo lắng, đau cơ thể không rõ nguyên nhân;  
 • Hoảng hốt;  
 • Căng thẳng;  
 • Cảm giác bị ám ảnh;  
 • Mất tập trung;  
 • Rối loạn giấc ngủ;  
 • Mất hứng thú sinh hoạt.  
 You: tại sao bị trầm cảm vậy  
 Bệnh viện Hồng Hưng:  
 • Do những thay đổi về nội tiết: Sau khi sinh con, cơ thể người phụ nữ sẽ có sự giảm đột ngột estrogen và progesterone, ngoài ra, hormones tuyến giáp;  
 • Do có sự thay đổi về thể tích máu, hệ miễn dịch, huyết áp và sự chuyển hóa trong cơ thể;  
 • Do mâu thuẫn gia đình, các vấn đề tài chính hoặc không có sự giúp đỡ của người thân;  
 • Do gặp nhiều khó khăn trong việc chăm sóc em bé, từ đó cảm thấy mất hứng thú sống và không kiểm soát được cuộc sống bản thân;  
 • Do di truyền.  
 You: mắc trầm cảm sau sinh có nguy hiểm không  
 Bệnh viện Hồng Hưng: Trầm cảm sau sinh con có thể khiến mẹ bị suy dinh dưỡng, sụt cân, suy nhược thần kinh, có những suy nghĩ hoang tưởng và dễ dẫn  
 You:

### Hình 4.7: Chatbot tư vấn về bệnh trầm cảm sau sinh

Sau khi chạy thử chatbot với 3 trường hợp nêu trên, với các từ khóa hợp lý, mô hình đã đưa ra dự đoán và trả lời chính xác câu hỏi từ phía người dùng. Nhưng vẫn không loại trừ khả năng mô hình có thể đưa ra dự đoán sai với những từ khóa chưa có trong bộ dữ liệu. Tuy nhiên, với kết quả này, nhận thấy mô hình đã có hoạt động hiệu quả và chính xác, có thể ứng dụng vào thực tế.

#### 4.5 Đánh giá thông qua trải nghiệm người dùng và chuyên gia

Đánh giá mô hình thông qua **chỉ số hiệu suất người dùng (performance metrics)**. Danh mục này tập trung vào các chỉ số chính liên quan đến việc sử dụng và tương tác, chẳng hạn như:

- Tổng số người dùng đang hoạt động;
- Số lượng người dùng mới sử dụng bot;
- Tổng số tương tác;
- Người dùng trực tiếp hoặc hữu cơ;
- Số lượng cuộc trò chuyện trung bình hàng ngày được xử lý bởi bot;
- Phiên trung bình hoặc thời gian xử lý trung bình; và
- Số lượng chủ đề.

Đánh giá mô hình thông qua **chỉ số trải nghiệm khách hàng (Customer experience metrics)**. Danh mục này tập trung vào trải nghiệm tổng thể của khách hàng, bao gồm khả năng tự phục vụ, mức độ hài lòng của khách hàng và tỷ lệ duy trì. Một số chỉ số này bao gồm:

- Trải nghiệm của khách hàng với bot - các doanh nghiệp có thể đo lường điều này bằng điểm số của người quảng cáo rờng, mức độ hài lòng của khách hàng hoặc điểm nỗ lực của khách hàng;
- Người dùng lần đầu tiên và tỷ lệ duy trì;
- Tỷ lệ giải quyết tự phục vụ hoặc tỷ lệ hoàn thành mục tiêu;
- Leo thang giữa chatbot với con người - đây được coi là những tương tác không hiệu quả cần được theo dõi để cải thiện liên tục; và
- Điểm phân tích tình cảm.

Đánh giá mô hình thông qua hiệu suất tài chính / bán hàng / tăng trưởng (). Danh mục này tập trung vào doanh số và chi phí, bao gồm:

- Tỷ lệ chuyển đổi - chatbots chuyển đổi một cuộc trò chuyện thành một cuộc mua bán;
- Tổng doanh số bán chatbot trực tiếp - khách hàng sử dụng chatbot để mua;

- Tổng chi phí sở hữu chatbot - chi phí vận hành và bảo trì;
- Tiết kiệm chi phí chatbot - các doanh nghiệp có thể đo lường điều này bằng phần trăm gia tăng chuyển đổi, giảm leo thang.

Các chỉ số đánh giá chatbot này có thể giúp các trung tâm liên hệ đo lường hiệu suất tổng thể của chatbot trong các lĩnh vực tư vấn sức khỏe sinh sản để đánh giá và liên quan đến kết quả kinh doanh của bệnh viện Hồng Hưng.

#### **4.6 Nhận xét**

Từ kết quả chạy mô hình và thực nghiệm phía trên cho thấy mô hình xây dựng có độ chính xác cao. Có thể đưa ra những tư vấn chính xác với câu hỏi từ phía người dùng. Bên cạnh đó, mô hình cũng đã học được nhiều trường hợp đã được đưa vào trong bộ dữ liệu đào tạo. Dù vậy, nhận xét thấy bộ dữ liệu sử dụng huấn luyện vẫn còn tương đối nhỏ và cần phải nghiên cứu, bổ sung nhiều hơn, tìm hiểu những thông tin chính xác hơn nữa để mang lại câu trả lời tốt nhất đáp ứng được nhu cầu của người dùng.

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết quả nghiên cứu của đề tài

Việc xây dựng bài toán ứng dụng máy học trong tạo sinh câu trả lời cho hệ thống hỏi đáp đã và đang được nhiều nhà nghiên cứu đặc biệt quan tâm trong thời gian gần đây. Mục tiêu của các công trình nghiên cứu nhằm tìm ra phương pháp xây dựng mô hình chính xác và hiệu quả. Cùng với việc phát triển mạnh mẽ của trí tuệ nhân tạo, tác giả đã tập trung nghiên cứu ứng dụng và phát triển các thuật toán trí tuệ nhân tạo vào việc xây dựng Chatbot tư vấn sinh sản cho bệnh viện Hồng Hưng. Qua quá trình học tập, nghiên cứu thực hiện luận văn, tác giả đã đạt được kết quả như sau:

Xây dựng được bộ dữ liệu tư vấn sinh sản và mô hình trả lời tự động bằng các kỹ thuật hỗ trợ bởi Tensorflow. Với độ chính xác của mô hình đạt tới 100% với độ mất mát nhỏ 1.8%, nhận thấy có thể áp dụng mô hình vào sử dụng thực tế, tự động hóa công tác tư vấn cho người dùng. Tuy nhiên, bộ dữ liệu còn tương đối nhỏ cần cải thiện và bổ sung nhiều hơn.

### 5.2 Hạn chế của đề tài

Trong quá trình thực hiện bài luận văn cũng không tránh khỏi thiếu sót:

- Bộ dữ liệu tương đối nhỏ, cần bổ sung nhiều dữ liệu hơn.
- Cách xử lý ngôn ngữ tiếng Việt còn nhiều thiếu sót.

### 5.3 Hướng phát triển của đề tài

Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu:

- Đưa mô hình đề xuất vào ứng dụng thực tế, xây dựng cơ sở dữ liệu to lớn và chính xác hơn nữa.
- Tìm thêm các cách xử lý tối ưu dữ liệu, xử lý ngôn ngữ tiếng Việt, tối ưu hóa mô hình, hiệu chỉnh độ chính xác của mô hình, giảm bộ mất mát hơn nữa.

## TÀI LIỆU THAM KHẢO

- [1] Meera Udani, Avinash Shrivastava, Vaibhav Shukla, “Question Answering System Based on Artificial Intelligence for Restricted Domain,” *International Journal of Engineering Research & Technology*, 2013.
- [2] Tiansi Dong, Ulrich Furbach, Ingo Glockner, Björn Pelzer, “A Natural Language Question Answering System as a Participant in Human Q&A Portals.,” trong *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [3] Tait Larson, Johnson (Heng) Gong, Josh Daniel, “Providing a Simple Question Answering System By Mapping Questions to Questions,” Technical report, Department of Computer Science, Stanford University, 2006.
- [4] A.Bouziane, D.Bouchiha, N.Doumib, M.Malki, “Question Answering Systems: Survey and Trends,” trong *The International Conference on Advanced Wireless, Information, and Communication*, 2015.
- [5] R.Mervin, “An Overview of Question Answering System,” trong *Proceedings of National Conference on Recent Trends In Web Technologies*, Chennai, 2013.
- [6] A.Bansal, Z.Eberhart, L.Wuy, C.McMillan, “A Neural Question Answering System for Basic Questions about Subroutines,” trong *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2021.
- [7] S.Sarkar, V.Rao, Baala Mithra SM, S.VRK.Rao, “NLP Algorithm Based Question and Answering System,” trong *2015 Seventh International Conference on Computational Intelligence, Modelling and Simulation*, Chennai, 2015.



- [8] W.Yu, L.Wu, Y.Deng, R.Mahindru, Q.Zeng, S.Guven, M.Jiang, "A Technical Question Answering System with Transfer Learning," trong Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020.
- [9] T.Zhao, K.Lee, "Talk to Papers: Bringing Neural Question Answering to Academic Search," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 30-36, July 2020.
- [10] Sina J. Semnani, Manish Pandey, "Revisiting the Open-Domain Question Answering Pipeline," Sep 2020.
- [11] R. Anantha, S. Vakulenko, Z. Tu, Sh. Longpre, S. Pulman, S. Chappidi, "Open-Domain Question Answering Goes Conversational via Question Rewriting," trong Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Amsterdam, 2021.
- [12] Sweta P. Lende, M. M. Raghuvanshi, "CLOSED DOMAIN QUESTION ANSWERING SYSTEM USING NLP TECHNIQUES," INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, pp. 632-639, 2016.
- [13] B. Marr, "Forbes," 24 September 2018. [Trực tuyến]. Available: <https://www.forbes.com/sites/bernardmarr/2018/09/24/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/?sh=282405fd1245>.
- [14] I. C. Education, "IBM," 14 September 2020. [Trực tuyến]. Available: <https://www.ibm.com/cloud/learn/recurrent-neural-networks>.
- [15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, 1999.
- [16] Hochreiter, Sepp & Schmidhuber, Jürgen, "Long Short-term Memory," Neural computation, 1997.

- [17] C. Olah, "Understanding LSTM Networks," [Trực tuyến]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [18] T. Vu, "Embedding," [Online]. Available: [https://machinelearningcoban.com/tabml\\_book/ch\\_embedding/embedding.html](https://machinelearningcoban.com/tabml_book/ch_embedding/embedding.html). [Accessed 2021].
- [19] Keras, "Embedding layer," [Online]. Available: [https://keras.io/api/layers/core\\_layers/embedding/](https://keras.io/api/layers/core_layers/embedding/).
- [20] N.T.Le, T.B.Le, S.T.Truong, "Building Filters for Vietnamese Chatbot Responses," trong 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh, 2020.
- [21] D.Q.Nguyen, D.Q.Nguyen, S.B.Pham, "A Vietnamese Question Answering System," trong The 2009 International Conference on Knowledge and Systems Engineering, KSE 2009, Ha Noi, 2009.
- [22] Truong H. V Phan, Phuc Do, "BERT+vnKG:Using Deep Learning and Knowledge Graph to Improve Vietnamese Question Answering System," International Journal of Advanced Computer Science and Applications 11(7), tập 11, pp. 480-487, 2020.
- [23] T. Nguyen and M. Shcherbakov, "A Neural Network based Vietnamese Chatbot," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), 2018.
- [24] Trung Thanh Nguyen, Anh Duc Le, Ha Thanh Hoang, Tuan Nguyen, "NEU-chatbot: Chatbot for admission of National Economics University," Computers and Education: Artificial Intelligence.
- [25] Oanh Thi Tran, Tho Chi Luong, "Understanding what the users say in chatbots: A case study for the Vietnamese language," Engineering Applications of Artificial Intelligence, 2020.

- [26] Vaishali Fulmal, K.P.Moholkar, S.H.Patil, "The Implementation of Question Answer System Using Deep Learning," Turkish Journal of Computer and Mathematics Education, tập 12, pp. 176-182, 2021.
- [27] R. Poonguzhali, Dr. K. Lakshmi, "Evaluating the Performance of Recurrent Neural Network based Question Answering System with Easy and Complex bAbI QA Tasks," International Journal of Advanced Science and Technology, tập 29, pp. 1389-1402, 2020.
- [28] T. T. Mayeesha , A. M. Sarwar, R. M. Rahman, "Deep learning based question answering system in Bengali," JOURNAL OF INFORMATION AND TELECOMMUNICATION, 2020.
- [29] Thara S, Sampath E, Venkata Sitarami Reddy B, Vidhya Sai Bhagavan M, Phanindra Reddy M, "Code Mixed Question Answering Challenge using Deep Learning Methods," trong Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020), 2020.
- [30] Botao Zhong, Wanlei He, Ziwei Huang, Peter E.D. Love, Junqing Tang, Hanbin Luo, "A building regulation question answering system: A deep learning methodology," Advanced Engineering Informatics, 2020.
- [31] Xie Z., Zeng Z., Zhou G., He T, "Knowledge Base Question Answering Based on Deep Learning Models. In: Lin CY., Xue N., Zhao D., Huang X., Feng Y.," Natural Language Understanding and Intelligent Applications, 2016.
- [32] Budiharto, W., Andreas, V. & Gunawan, A.A.S, "Deep learning-based question answering system for intelligent humanoid robot," J Big Data 7, 2020.
- [33] Sharma, D., Purushotham, S. & Reddy, C.K. , "MedFuseNet: An attention-based multimodal deep learning model for visual question answering in the medical domain," Sci Rep 11, 2021.

- [34] Palangi H, Smolensky P, He X, Deng L, Redmond WA, "Deep learning of grammatically-interpretable representations through question-answering," arXiv preprint arXiv:1705.08432. 2017 May.
- [35] F. Kenghagho Kenfack, F. Ahmed Siddiky, F. Balint-Benczedi and M. Beetz, "RobotVQA — A Scene-Graph- and Deep-Learning-based Visual Question Answering System for Robot Manipulation," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.
- [36] "Google Colaboratory," [Online]. Available: <https://colab.research.google.com/>.
- [37] TensorFlow, "tf.keras.preprocessing.text.Tokenizer," [Online]. Available: [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/text/Tokenizer](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer). [Accessed 02 September 2021].
- [38] TensorFlow, "tf.keras.preprocessing.sequence.pad\_sequences," [Online]. Available: [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/sequence/pad\\_sequences](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/sequence/pad_sequences). [Accessed 05 November 2021].
- [39] scikit-learn, "sklearn.preprocessing.LabelEncoder," [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>.



Hệ thống hỗ trợ nâng cao chất lượng tài liệu

## KẾT QUẢ KIỂM TRA TRÙNG LẬP TÀI LIỆU

### THÔNG TIN TÀI LIỆU

Tác giả	Nguyễn Minh Trí
Tên tài liệu	LuanVan_NguyenMinhTri
Thời gian kiểm tra	09-12-2021, 05:32:07
Thời gian tạo báo cáo	09-12-2021, 05:33:25

### KẾT QUẢ KIỂM TRA TRÙNG LẬP



(\*) Kết quả trùng lặp phụ thuộc vào dữ liệu hệ thống tại thời điểm kiểm tra

Người hướng dẫn

Học viên

**PGS.TS Nguyễn Tuấn Đăng**

**Nguyễn Minh Trí**

## **BẢNG CAM ĐOAN**

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn/ luận án qua phần mềm <http://doit.uet.vnu.edu.vn/> một cách trung thực và đạt kết quả mức độ tương đồng 11% trên toàn bộ nội dung luận văn/ luận án. Bản kiểm tra luận văn/ luận án qua phần mềm là bản cứng luận văn/luận án đã nộp bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỉ luật theo qui định hiện hành của Học viện.

Tây Ninh, ngày 25 tháng 01 năm 2022

Học viên cao học

**Nguyễn Minh Trí**

## PHỤ LỤC

### Chatbot for Fertility Counseling

```
[1] # import google colab to read the google drive directory => get the folder store the source code & data
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
[2] import os
os.chdir("/content/drive/My Drive/LuanVan") #
os.listdir()
```

['Untitled0.ipynb', 'chatbot.ipynb', 'content-vn.json']

```
[3] #importing the libraries
import tensorflow as tf
import numpy as np
import pandas as pd
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.layers import Input, Embedding, LSTM, Dense, GlobalMaxPooling1D, Flatten
from tensorflow.keras.models import Model
import matplotlib.pyplot as plt
import json # to read the json data file
import nltk
```

```
▶ # Enable Eager Execution and Print Versions
if tf.__version__ < "2.0.0":
    tf.enable_eager_execution()
    print("Eager execution enabled.")
else:
    print("Eager execution enabled by default.")

print("TensorFlow " + tf.__version__)
```

```
[6] # import the dataset
with open('content-vn.json') as content:
    data1 = json.load(content)
```

▶ data1

```
{
  'tag': 'WhatIsGestationalDiabetes'},
{'input': ['yếu tố nguy cơ của đái tháo đường thai kỳ'],
 'responses': ['• Thừa cân, béo phì. • Tiền sử gia đình: Có người bị đái tháo đường, đặc biệt'],
 'tag': 'GestationalDiabetesRiskFactor'},
{'input': ['tác hại của đái tháo đường thai kỳ'],
 'responses': ['Bệnh tiểu đường khi mang thai làm tăng nguy cơ thai to, khó sinh do kẹt vai'],
 'tag': 'DangerOfGestationalDiabetes'},
{'input': ['điều trị đái tháo đường thai kỳ'],
 'responses': ['Khi mắc đái tháo đường thai kỳ, thai phụ cần liên hệ bệnh viện để có những'],
 'tag': 'TreatmentGestationalDiabetes'},
{'input': ['cách phát hiện mắc đái tháo đường thai kỳ'],
 'responses': ['Sàng lọc đái tháo đường thai kỳ'],
 'tag': 'ScreeningGestationalDiabetes'},
{'input': ['phòng ngừa đái tháo đường thai kỳ'],
 'responses': ['phòng tránh đái tháo đường thai kỳ'],
 'tag': 'PreventGestationalDiabetes'},
{'input': ['nhau tiền đạo là gì'],
 'responses': ['• Bình thường, bánh nhau bám ở vùng đáy hoặc thân tử cung. Trong trường hợp'],
 'tag': 'WhatIsPlacentaPrevia'},
{'input': ['dấu hiệu nghi ngờ nhau tiền đạo', 'dấu hiệu mắc nhau tiền đạo'],
 'responses': ['• Trong 3 tháng cuối thai kỳ, đột ngột thai phụ bị ra huyết đỏ tươi, có thể'],
 'tag': 'SignsOfPlacentaPrevia'},
{'input': ['phát hiện nhau tiền đạo',
 'làm thế nào để phát hiện nhau tiền đạo'],
 'responses': ['• Hiện nay phương pháp an toàn và được dùng nhiều nhất tại Việt Nam là siêu'],
 'tag': 'DetectPlacentaPrevia'},
{'input': ['tại sao phụ nữ mang thai bị nhau tiền đạo'],
 'responses': ['Nguyên nhân chính xác vẫn chưa được xác định rõ ràng. Tuy nhiên, người ta t'],
 'tag': 'WhyGetDetectPlacentaPrevia'},
{'input': ['ảnh hưởng của nhau tiền đạo trên mẹ và thai'],
 'responses': ['• Ảnh hưởng trên mẹ: Ra huyết âm đạo gây thiếu máu. Nếu ra huyết nhiều có k'],
 'tag': 'DangerOfPlacentaPrevia'},
{'input': ['cần phải làm gì khi được chẩn đoán nhau thai tiền đạo'],
 'responses': ['• Nếu có ra huyết âm đạo: Cần phải vào bệnh viện có khoa sản gần nhất để đư'],
 'tag': 'WhatToDoWhenDiagnosedPlacentaPrevia'},
{'input': ['Có phải mổ lấy thai cho tất cả các trường hợp nhau tiền đạo?'],
 'responses': ['• Không phải mổ lấy thai cho tất cả các trường hợp nhau tiền đạo. Chỉ cần m'],
 'tag': 'CaesareanForAllCasePlacentaPrevia'},
```



```
[8] #getting all the data to lists
tags = []
inputs = []
responses={}
for intent in data1['intents']:
    responses[intent['tag']]=intent['responses']
    for lines in intent['input']:
        inputs.append(lines)
        tags.append(intent['tag'])
```

```
▶ #converting to dataframe
data = pd.DataFrame({"inputs":inputs,
                    "tags":tags})
```

```
[10] data
```

	inputs	tags
0	xin chào	Greeting
1	chào	Greeting
2	bạn có thể giúp tôi không ?	Greeting
3	tôi cần giúp đỡ	Greeting
4	tôi cần tư vấn	Greeting
...	...	...
227	biến chứng của rau bong non	DangerOfAblatioPlacentae
228	triệu chứng của nhau bong non	SignsOfAblatioPlacentae
229	dấu hiệu của rau bong non	SignsOfAblatioPlacentae
230	điều trị rau bong non	TreatmentAblatioPlacentae
231	điều trị nhau bong non như thế nào	TreatmentAblatioPlacentae

232 rows × 2 columns

```
[11] data = data.sample(frac=1)
```

## Pre-Processing

```
[12] #removing punctuations
import string
data['inputs'] = data['inputs']
      .apply(lambda wrd:[ltrs.lower() for ltrs in wrd if ltrs not in string.punctuation])
data['inputs'] = data['inputs'].apply(lambda wrd: ''.join(wrd))
```

```
▶ # tokenizer is an algorithm that is responsible for separating words and phrases
  # in the text and is a sub-branch of NLP

  # tokenize the data
from tensorflow.keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(data['inputs'])
train = tokenizer.texts_to_sequences(data['inputs'])

# apply padding
from tensorflow.keras.preprocessing.sequence import pad_sequences
x_train = pad_sequences(train)

# encoding the outputs
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y_train = le.fit_transform(data['tags'])
```

```
[14] input_shape = x_train.shape[1]
      print(input_shape)
```

```
[15] #define vocabulary
      vocabulary = len(tokenizer.word_index)
      print("number of unique words : ",vocabulary)
      output_length = le.classes_.shape[0]
      print("output length: ",output_length)
```

## Neural Network

```
[16] #creating the model
      i = Input(shape=(input_shape,))
      x = Embedding(vocabulary+1,10)(i)
      x = LSTM(10,return_sequences=True)(x)
      x = Flatten()(x)
      x = Dense(output_length,activation="softmax")(x)

      model = Model(i,x)
```

```
[18] #compiling the model
      model.compile(loss="sparse_categorical_crossentropy",optimizer='adam'
                    ,metrics=['accuracy'])

#training the model
train = model.fit(x_train,y_train,epochs=300)
```

## Model Analysis

The model got a perfect accuracy of 100%.

```
[19] #plotting model accuracy
plt.plot(train.history['accuracy']
         ,label='training set accuracy')
plt.plot(train.history['loss']
         ,label='training set loss')
plt.legend()
```

```
[27] #chatting
import random

while True:
    texts_p = []
    prediction_input = input('You: ')

    # removing punctuation and converting to lowercase
    prediction_input = [letters.lower() for letters in prediction_input if letters not in string.punctuation]
    prediction_input = ''.join(prediction_input)

    # apply word segmentation
    from underthesea import word_tokenize
    word_tokenize(prediction_input, format="text")

    texts_p.append(prediction_input)

    #tokenizing and padding
    prediction_input = tokenizer.texts_to_sequences(texts_p)
    prediction_input = np.array(prediction_input).reshape(-1)
    prediction_input = pad_sequences([prediction_input],input_shape)

    #getting output from model
    output = model.predict(prediction_input)
    output = output.argmax()

    #finding the right tag and predicting
    response_tag = le.inverse_transform([output])[0]
    print("Bệnh viện Hồng Hưng: ",random.choice(responses[response_tag]).replace("•", "\n•"))
    if response_tag == "Goodbye":
        break
```