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

TÓM TẮT LUẬN VĂN THẠC SĨ

(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **PGS.TS NGUYỄN TUẤN ĐĂNG**

Phản biện 1: .....

Phản biện 2: .....

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn tại Học viện Công nghệ Bưu chính Viễn Thông

Vào lúc: ..... giờ ..... ngày ..... tháng ..... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu Chính Viễn Thông.

MỞ ĐẦU

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.

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 phân tích câu hỏi có vai trò hết sức quan trọng, ảnh hưởng trực tiếp đến hoạt động của toàn bộ hệ thống. Bài toán đặt ra nhiều thách thức để phát hiện ra được câu trả lời phù hợp nhất, thông tin hữu ích nhất.

Luận văn gồm 5 chương chính với các nội dung sau:

Chương 1: Giới thiệu tổng quan về hệ thống trả lời tự động, các mô hình trả lời tự động và các cơ sở lý thuyết cần thiết khi nghiên cứu đề tài.

Chương 2: Trình bày về các công trình nghiên cứu trong và ngoài nước liên quan mật thiết tới đề tài

Chương 3: 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.

Chương 4: Trình bày chi tiết việc xây dựng bộ dữ liệu huấn luyện và quá trình cụ thể cài đặt mô hình cho thuật toán.

Chương 5: Kết luận nội dung đã được trong đề tài, nêu những khó khăn, hạn chế trong quá trình nghiên cứu đã gặp phải và đề xuất hướng phát triển tiếp theo.

Đề tài: ỨNG DỤNG MÁY HỌC TRONG TẠO SINH CÂU TRẢ LỜI CHO HỆ THỐNG HỎI ĐÁP

Tóm tắt luận văn

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Giới thiệu chung

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 vốn nhập nhằng, đa nghĩa, việc xác định được ngữ nghĩa của câu hỏi cũng như phát hiện ra câu trả lời là một thách thức không nhỏ. 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, thông tin hữu ích nhất.

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 lí ngôn ngữ tự nhiên (Natural Language Processing - NLP) – một hệ thống xử lí 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 lí 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 lí 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 lí câu hỏi, module quá trình xử lí 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 nhanh chóng và mạnh mẽ về vấn đề xử lí 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

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 lí 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 lí 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 nhặt ra một đáp án thích hợp nhất dự vào mẫu hỏi input và ngữ cảnh. Kỹ thuật heuristic sử dụng ở đây đơn giản có thể là sự so 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 kiểu này không sinh ra văn bản mới, chúng chỉ nhặt 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 định nghĩa trước. Chúng có khả năng tự sản sinh các đáp án từ đầu. Các mô hình sinh thường dựa vào các kỹ thuật Máy Dịch (Machine Translation), nhưng thay vì dịch từ ngôn ngữ này sang ngôn ngữ khác, thì nó có thể "dịch" từ một input sang một output.

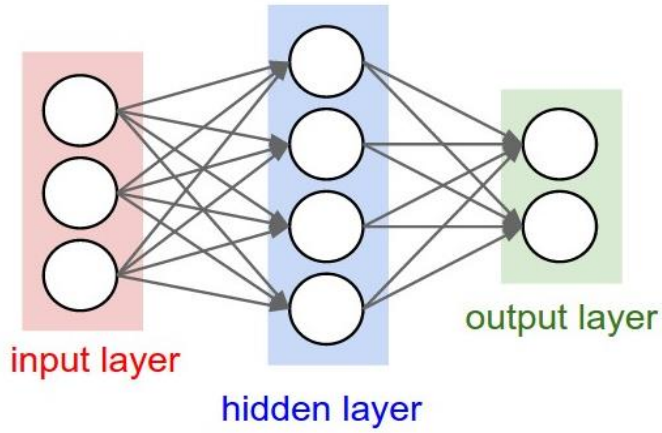
Nhờ vào kho dữ liệu với các bộ luật được thiết kế bằng tay, mô hình dựa trên trích chọn thông tin (retrieval-based) không mắc phải các lỗi về ngữ pháp. Tuy nhiên, chúng không thể xử lý được các trường hợp các mẫu chưa được quan sát, không có trong bộ luật. Vì những lý do đó, các mô hình này không thể 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 thể nhớ lại được các thực thể được nhắc đến trong mẫu hỏi và tạo ra cảm giác bạn đang nói chuyện với con người. Tuy nhiên, những mô hình này thì rất khó để huấn luyện, rất có thể bị mắc lỗi về ngữ pháp (đặc biệt trên các câu dài) và mô hình yêu cầu một lượng rất lớn dữ liệu để huấn luyện.

Các kỹ thuật học sâu Deep Learning có thể được sử dụng cho cả hai mô hình Retrieval-based hoặc Generative-based, nhưng các nhà nghiên cứu thường tập trung hướng vào mô hình Generative. Tuy nhiên, chúng ta vẫn đang ở giai đoạn đầu của việc tiếp cận với mô hình sinh và có kết quả khả quan. Song thời điểm hiện tại, các hệ thống thương mại vẫn phù hợp với các mô hình Retrieval-based.

1.4. Kiến trúc mạng nơ-ron nhân tạo

Lấy cảm hứng 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 Error! No text of specified style in document.**.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. Tuy nhiên trong thực tế người ta thường để chúng cùng dạng với nhau để 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, sử dụng chúng không đơn giản như vậy, một số các đặc tính và kinh nghiệm khi thiết kế một mạng nơ-ron ANN.

1.5. Hoạt động của mạng nơ-ron nhân tạo

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

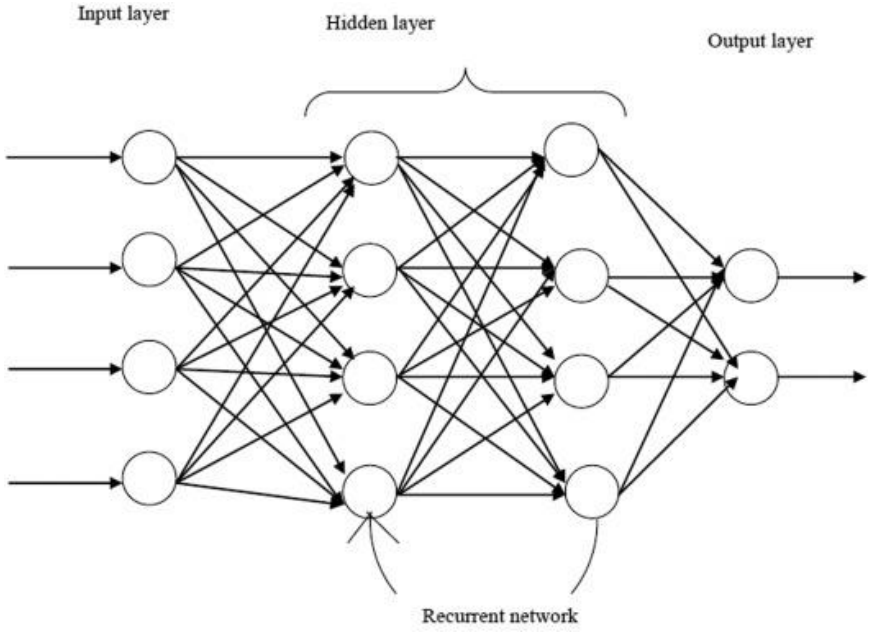
qua một hoặc nhiều đơn vị ẩn. Công việc của đơn vị ẩn là biến đầu vào thành thứ mà đơn vị đầu ra có thể sử dụng.

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.

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

1.6.1 Mạng nơ-ron RNN

Mạng nơ-ron RNN: Mạng nơ-ron tuần hoàn (RNN) là một loại mạng nơ-ron, 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 Error! No text of specified style in document.**.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 vectơ là $\{y_1, y_2, \ldots y_M\}$, chuỗi các trạng thái ẩn là $\{z_1, z_2, \ldots z_M\}$ và đơn vị đầu ra tương ứng là $\{v_1, v_2, \ldots v_M\}$.

Lớp hồi quy bao gồm hàm tái quy d, lấy vectơ 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:

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

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

$$v_x = soft\max\left(R \cdot z_x\right)$$

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 lí 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.

Hình Error! No text of specified style in document.**.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 cũng sử dụng đến thuật toán backpropagation (lan truyền ngược) nhưng có một chút tinh chỉnh. 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. Kĩ thuật này gọi là Backpropagation Through Time (BPPTT). Điểm hạn chế ở đây đó là hidden layer không có trí nhớ dài hạn. Vấn đề này còn gọi là vanishing/exploding gradient problem và như vậy, LSTM được sinh ra để giải quyết vấn đề này.

1.6.4. Các phiên bản mở rộng của RNN

Mạng nơ-ron tái phát hai chiều (Bidirectional recurrent neural networks - BRNN): 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. 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”.

Bộ nhớ ngắn hạn dài (LSTM): 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. 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. Ví dụ: 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.

1.7. Mô hình trả lời tự động

Bản thân mô hình seq2seq [10] nó 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. Mục tiêu của nó là chuyển đổi một chuỗi các phần tử vào một vectơ 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 vevtơ 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. Ở đây, tại mỗi bước, các bộ giải mã bị ảnh hưởng bởi bối cảnh và các phần tử được sinh ra trước đó.

1.8. Embedding và Keras Embedding Layer

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.

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 điễn 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

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. Các công trình liên quan trong và ngoài nước

2.2.1 Các nghiên cứu trong nước

- "Building Filters for Vietnamese Chatbot Responses"

- "A Vietnamese Question Answering System"

- "BERT+vnKG:Using Deep Learning and Knowledge Graph to Improve Vietnamese Question Answering System"

2.2.2 Các nghiên cứu ngoài nước

- "A Technical Question Answering System with Transfer Learning"

- "The Implementation of Question Answer System Using Deep Learning"

- "Evaluating the Performance of Recurrent Neural Network based Question Answering System with Easy and Complex bAbI QA Tasks"

- "Deep learning based question answering system in Bengali"

- "Code Mixed Question Answering Challenge using Deep Learning Methods"

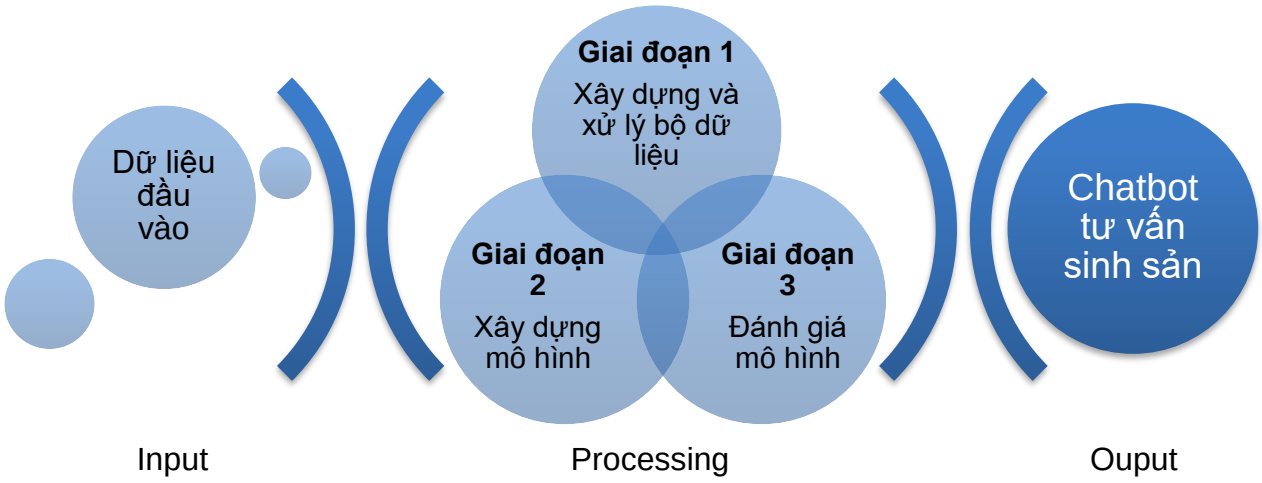
CHƯƠNG 3. ĐỀ XUẤT PHƯƠNG PHÁP XÂY DỰNG BỘ DỮ LIỆU VÀ MÔ HÌNH CHO BÀI TOÁN

3.1. Giới thiệu chương

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 Error! No text of specified style in document.**.4: Quy trình bài toán**

3.2.1. Input: dữ liệu 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

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.

4.2. Đề xuất phương pháp và thuật toán xử lý

4.2.1. Cơ sở lý thuyết của bộ dữ liệu

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:

- 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 sanh 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

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, input, response. Trong đó:

- 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 thẻ 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.

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 ***Tokenziner*** [25] đây là một nhánh con trong tập xử lý ngôn ngữ tự nhiên.

Tokenziner cho phép ta vectơ 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*** [26] 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*** [27] 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
```

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 reccurent với cổng ***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***.

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
```

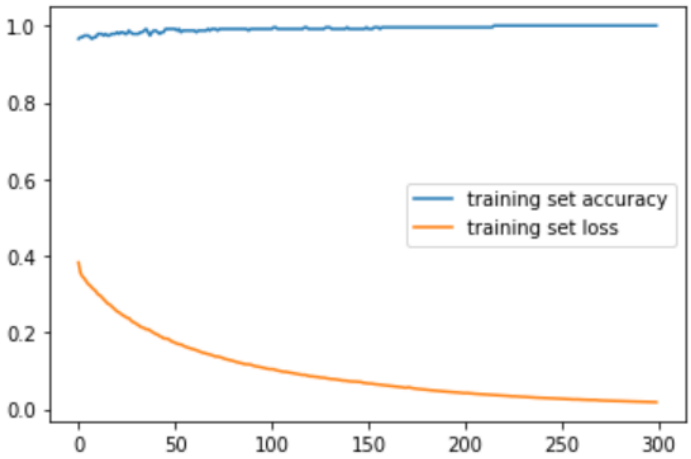
Hình Error! No text of specified style in document.**.5: Huấn luyện cho mô hình**

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

Đánh giá độ chính xác của mô hình trên bộ dữ liệu đào tạo.

```
<matplotlib.legend.Legend at 0x7f4226c4ef90>
```



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 vào Chatbot để tiến hành kiểm thử.

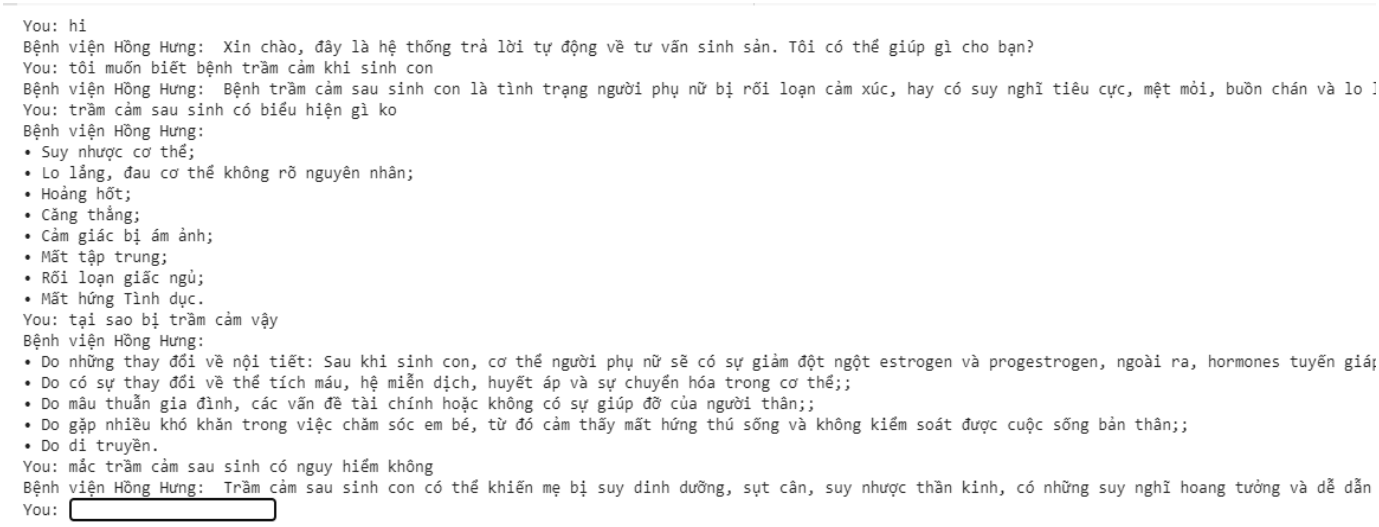
- Tư vấn bị động thai

```
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
```

- 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 cuống rốn.
• 36 đến 40 tuần: mỗi tuần khám 1 lần. Kiểm tra tử cung và tư vấn các dấu hiệu sắp sinh.
You: bổ sung dĩnh 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 cám ơn
Bệnh viện Hồng Hưng:  Tạm biệt
```

- Tư vấn về vấn đề 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. Nhận xét

Từ kết quả chạy mô hình và thực nghiệm nhận 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

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. Đề xuất phương pháp và thuật toán xử lý

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.