BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

XÂY DỰNG TRỢ LÝ ẢO CHO SINH VIÊN KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG, TRƯỜNG ĐẠI HỌC CẦN THƠ SỬ DỤNG MÔ HÌNH TRANSFORMER

Sinh viên: Ngụy Hữu Lộc
Mã số: B1706606
Khóa: K43

Cần Thơ, 06/2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

XÂY DỰNG TRỢ LÝ ẢO CHO SINH VIÊN KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG, TRƯỜNG ĐẠI HỌC CẦN THƠ SỬ DỤNG MÔ HÌNH TRANSFORMER

Người hướng dẫn
TS. Lâm Nhựt Khang

Sinh viên thực hiện
Ngụy Hữu Lộc
Mã số: B1706606
Khóa: K43

Cần Thơ, 06/2021

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Cần Thơ, ngày …tháng … năm …

Giáo viên hướng dẫn

TS. Lâm Nhựt Khang

LỜI CAM ĐOAN

Em xin cam đoan đề tài "Xây dựng trợ lý ảo cho sinh viên Khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ sử dụng mô hình Transformer" là công trình nghiên cứu độc lập dưới sự hướng dẫn của TS. Lâm Nhựt Khang. Không có bất kỳ một sự sao chép nào từ đề tài của người khác. Nội dung trong luận văn này là do bản thân em đã đúc kết ra trong suốt những tháng thực hiện đề tài luận văn. Các số liệu thống kê được trong luận văn đều do bản thân em đã thực hiện và hoàn toàn là trung thực, em xin chịu hoàn toàn trách nhiệm, kỷ luật của khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ nếu em có sự sai phạm nào.

Cần Thơ, ngày …..... tháng …… năm 2021

Sinh viên thực hiện

Ngụy Hữu Lộc

LỜI CẢM ƠN

Để hoàn thành đề tài luận văn "Xây dựng trợ lý ảo cho sinh viên Khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ", em xin chân thành cảm ơn TS. Lâm Nhựt Khang đã tận tình, giúp đỡ hướng dẫn, hỗ trợ cho em trong suốt những tháng thực hiện đề tài luận văn.

Em xin được cảm ơn đến Thầy, Cô ở khoa Công nghệ Thông tin và Truyền thông đã truyền đạt cho em những kiến thức bổ ích trong suốt những năm tháng học tập vừa qua, để em có thể thuận lợi thực hiện cũng như hoàn thành đề tài của mình.

Mặc dù đã cố gắng để hoàn thiện nhưng không sao tránh được nhưng sai sót, em rất mong nhận được sự thông cảm và những đóng góp quý báu của quý Thầy, Cô để luận văn của em được hoàn thiện nhất có thể.

Cuối lời, em xin được chân thành cảm ơn đến Thầy Cô, chúc Thầy Cô luôn có nhiều sức khỏe và thành công trong công việc và cuộc sống.

Cần Thơ, ngày …… tháng …… năm 2021

Sinh viên thực hiện

Ngụy Hữu Lộc

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BIỂU BẢNG

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ chuẩn
AI	Artificial Intelligence
ANN	Artificial Neural Network
BLEU	Bilingual Evaluation Understudy
CRF	Random Condition Fields
K	Key
KNN	K-Nearst-Neighbors
LSTM	Long Short-Term Memory
NLTK	Natural Language Toolkit
RNN	Recurrent Neural Network
Seq2seq	Sequence to Sequence
SVM	Support Vector Machine
V	Value

TÓM TẮT

Luận văn này đề xuất phương pháp xây dựng trợ lý ảo cho sinh viên Khoa Công nghệ Thông tin và Truyền thông (CICT), trường Đại học Cần Thơ bằng mô hình Transformer. Trợ lý ảo giúp trả lời các câu hỏi giới thiệu về CICT, bao gồm chương trình đào tạo và giảng viên của Khoa, quy chế học vụ, và sổ tay sinh viên. Các mô hình Transformer không chỉ được sử dụng để huấn luyện chatbot mà còn dùng để sửa các từ sai chính tả và chuyển đổi các từ không có dấu thành từ có dấu. Để cải thiện phản hồi từ chatbot, CRF được sử dụng để trích xuất các thực thể. Hệ thống chatbot được huấn luyện bằng cách sử dụng bộ dữ liệu CTUNLPBot bao gồm hơn 33.000 cặp câu hỏi-câu trả lời được xây dựng thủ công. Kết quả đánh giá trên một tập dữ liệu nhỏ được xây dựng bởi các sinh viên của CICT cho thấy mô hình đạt điểm BLEU-1: 0,292; BLEU-2: 0,251; BLEU-3: 0,223; và BLEU-4: 0,198.

ABSTRACT

This thesis proposes a method to construct a virtual assistant for students at the College of Information and Communication Technology (CICT) of Can Tho university using Transformer models. The chatbot system is required to answer introductory questions about CICT, including programs and staff, academic regulations, and study handbooks. The Transformer models are used not only to train the chatbot system but also to correct misspelled words and to transform words with no diacritics written above or below the vowels to correct spelling words. To improve responses from the chatbot, the CRF is used to extract name entities. The chatbot system is trained using the CTUNLPBot dataset consisting of more than 33,000 question-answer pairs constructed manually. Evaluation on a small dataset constructed by the students of CICT shows that the model achieves BLEU-1: 0,292; BLEU-2: 0,251; BLEU-3: 0,223; and BLEU-4: 0,198.

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN

Trợ lý ảo hay còn gọi là chatbot ngày càng xuất hiện nhiều hơn trong cuộc sống của con người, chúng dần đã trở thành một thành phần không thể thiếu trong bất kỳ hệ thống nào, ngày càng có vai trò góp phần thay đổi cuộc sống của con người hiện đại hơn. Trong luận văn này, "trợ lý ảo" hay "chatbot" sẽ được sử dụng thay thế cho nhau. Chương 1 sẽ trình bày tổng quan về chatbot, mục tiêu đề tài, đối tượng, phạm vi, phương pháp và nội dung nghiên cứu.

1.1. Tổng quan

Ngày nay cùng với sự phát triển của các nền tảng trí tuệ nhân tạo, các ứng dụng trí tuệ nhân tạo ngày cũng xuất hiện nhiều hơn trong cuộc sống. Nếu thời gian trước đây để có thể được hỗ trợ về một sản phẩm hay một vấn đề thì cần nhờ vào sự giúp đỡ của những nhân viên tư vấn, thì những năm gần đây, sự xuất hiện của nhân viên tư vấn người thật gần như đã không còn nhiều mà thay vào đó là nền tảng chatbot được đưa vào các website để có thể tự động trả lời các câu hỏi cũng như đưa ra các thông tin mà khách hàng mong muốn không khác gì sự xuất hiện của một nhân viên tư vấn là người thật. Các chabot trò chuyện trực tiếp này, được biết đến với nhiều tên gọi khác nhau như là chatbot đàm thoại, chat AI, trợ lý ảo AI, …Hay gọi chung cho tất cả là chatbot, một hệ thống phản hồi hay trả lời tự động cho các yêu cầu được đưa vào từ người dùng.

Nhận thấy được tính tất yếu về nhu cầu sử dụng cũng như là sự cần thiết của các trợ lý ảo trong việc tìm kiếm thông tin, nhất là đối với các sinh viên thuộc lĩnh vực công nghệ thông tin, những người có nhu cầu tìm kiếm và sử dụng thông tin hàng ngày. Luận văn này sẽ nghiên cứu xây dựng một trợ lý ảo cho sinh viên, cụ thể hơn là xây dựng trợ lý ảo cho sinh viên Khoa công Nghệ thông và Truyền thông, trường Đại học Cần Thơ sử dụng mô hình Transformer, cho phép tra cứu thông tin liên quan đến quy chế học vụ, các học phần, lịch thi học kỳ… .

1.2. Nghiên cứu liên quan

Trong nhiều năm qua, chatbot ngày càng phát triển. Điển hình cho sự phát triển mạnh mẽ của chatbot là các trợ lý ảo được tích hợp vào nhiều nền tảng như là Cortana[1] của Microsoft, Siri[2] của Apple, Google Assistant[3] của Google, hay Amazon Alexa[4] của Amazon,… Không còn chỉ đơn thuần là các chatbot sử dụng các tập lệnh được xây dựng sẵn để đưa ra câu trả lời về một chủ đề, một món hàng, hay một bộ phim nào đó, chatbot đã cho phép con người có thể tương tác bằng cách sử dụng nhiều hình thức như là chữ viết, hình ảnh và thậm chí là sử dụng ngôn ngữ tự nhiên để giao tiếp với chatbot.

[1] https://www.microsoft.com/en-us/cortana
[2] https://www.apple.com/siri/
[3] https://assistant.google.com/
[4] https://alexa.amazon.com

Dharwadkar và Deshpande [1] đã giới thiệu một chatbot về y khoa sử dụng mô hình cấu trúc AIML[5] kết hợp sử dụng các thuật toán phân loại KNN, SVM[6] và Native Bayes[7] để phân loại câu hỏi và đưa ra câu trả lời, với ba tập dữ liệu được thu thập từ nhiều thành phố khác nhau với độ lớn khác nhau và đa miền, dữ liệu được thu thập chủ yếu là bệnh tim, tác giả đã sử dụng 60% dữ liệu thu thập để huấn luyện và 40% còn lại làm tập kiểm tra, kết quả thực nghiệm với tỉ lệ số câu trả lời đúng trên tổng số câu hỏi được nhập vào, với các giải thuật phân lớp KNN, SVM, Native Bayes lần lượt là 88,67%; 94,67%; và 80,00%.

Bao và cộng sự [2] đã giới thiệu chatbot y tế trực tuyến dựa trên sơ đồ Knowledge Graph và Hierarchical Bi-Directional Attention, đây là một hệ thống có khả năng trợ giúp chăm sóc sức khỏe và trả lời trực tuyến các câu hỏi phức tạp. Tập dữ liệu được sử dụng lấy từ tập câu hỏi Quora[8] với hơn 400.000 cặp câu hỏi miền mở, sau đó các tác giả đã lọc ra tập dữ liệu con chứ dữ liệu y tế, rồi tạo ra bộ từ điển chứ từ khóa về bệnh và triệu chứng, với số từ khóa về bệnh và triệu chứng lần lượt là 668 và 2.367, bộ từ điển các tác giả đã thu thập gần 70.000 hồ sơ y tế từ lấy từ Quora, cuối cùng chọn ngẫu nhiên 10.000 cặp câu hỏi làm tập huấn luyện và kiểm tra với tỷ lệ 9.000 cặp cho huấn luyện và 1.000 cặp cho tập kiểm tra. Kết quả thực nghiệm cho thấy hệ thống nhận dạng được 81,2% số câu hỏi trên tổng số câu hỏi được nhập vào.

Serban và các cộng sự [3] đã giới thiệu hệ thống đối thoại sử dụng mô hình mạng Neural Generative Hierarchical, mô hình được huấn luyện dựa trên tập dữ liệu MovieTriples được mở rộng và phát triển từ tập dữ liệu Movie-DiC (tập dữ liệu thu thập được bao gồm 132.229 cuộc đối thoại với tổng số 764.146 lượt được trích xuất từ 753 bộ phim) bởi Banchs và cộng sự của ông [4], mô hình có sử dụng kết hợp với sự hỗ trợ của NLTK[9] để thực hiện mã hóa và nhận diện tập dữ liệu. Kết quả cho thấy mô hình mạng neural Generative Hierarchical đưa ra câu trả lời là chính xác và vượt trội hơn các mô hình sử dụng n-gram và mạng neural thông thường. Với kết quả đánh giá tỷ lệ lỗi (error rate) là 66%.

Kamphaug và các cộng sự [5] đã giới thiệu chatbot miền mở sử dụng kiến trúc GRU cho các cuộc trò chuyện theo hướng dữ liệu, kết hợp với mạng Bidirectional Recurrent Neural - BiRNN để nhận biết ý định, với tập dữ liệu 200.083 câu hỏi gồm nhiều nội dung, có 289 chủ đề, liên quan đến luật, sức khỏe và các vấn đề xã hội. Kết quả cho thấy mô hình BiRNN cho độ chính xác đạt 71,20% số câu trả lời đúng trên 10% dữ liệu dùng làm câu hỏi, được lấy từ tập huấn luyện dùng làm tập kiểm tra.

[5] https://en.wikipedia.org/wiki/AIML
[6] https://en.wikipedia.org/wiki/Support-vector_machine
[7] https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[8] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
[9] https://vi.wikipedia.org/wiki/Natural_Language_Toolkit

Chandra và Suyanto [6] đã giới thiệu một chatbot tuyển sinh đại học Indonesia sử dụng mô hình Seq2Seq với cơ chế Attention, tập dữ liệu được sử dụng là một tập hợp các cuộc hội thoại Whatsapp được thu thập từ Quản trị viên tuyển sinh trong một trường đại học, sau đó tăng số lượng câu hỏi bằng cách dùng từ đồng nghĩa với số lượng các cuộc hội thoại là 2.903 cuộc hội thoại, sau đó chia ra 2.506 cuộc hội thoại cho tập huấn luyện là 397 cuộc hội thoại cho tập kiểm tra. Kết quả đạt được khi sử dụng mô hình Seq2Seq không có kỹ thuật Attention và có sử dụng kỹ thuật Attention cho điểm BLEU lần lượt là 43,61 và 44,68.

Luận văn tốt nghiệp của Lê Nhật Nam [7] [8] , sử dụng kết hợp framework Rasa với phương pháp SVM cho phân loại ý định, cùng với CRF trích xuất thực thể và mô hình LSTM để huấn luyện đối thoại. Với tập dữ liệu bao gồm 441 câu hỏi thuộc 19 ý định (nhãn), 253 thực thể thuộc 6 lớp cùng với 133 cuộc trò chuyện với hơn 1.300 hành động phản hồi. Kết quả thực hiện với 8 cuộc trò chuyện và có đến 6 cuộc trò chuyện là đúng, độ chính xác 75% và 90 phản hồi là phù hợp với độ chính xác lên đến là 92.78%. Hay ở luận văn của Nguyễn Văn Vĩ [9] sử dụng mô hình SCST kết hợp với hai mô hình Seq2Seq và LSTM, xây dựng chatbot trên tập dữ liệu miền mở với hơn 1,5 triệu dòng hội thoại với điểm số BLEU trung bình đạt 0,07.

Mô hình Transformer lần đầu tiên được giới thiệu bởi Vaswani và cộng sự [10], mô hình có kiến trúc mạng neural được xây dựng dựa trên kỹ thuật Attention, kết hợp với hai cơ chế Encoder và Decoder. Mô hình cho kết quả dịch thuật là rất cao so với các mô hình khác trên cùng một tập dữ liệu. Mô hình cho ra điểm số BLEU là 28,4 khi thực hiện việc dịch từ tiếng Anh sang tiếng Đức và 41,8 điểm BLEU khi dịch từ tiếng Anh sang tiếng Pháp. Kết quả này cho thấy mô hình có một hiệu suất và độ chính xác là khá tốt. Do đó, chúng tôi nghiên cứu sử dụng mô hình Transformer vào bài toán xây dựng trợ lý ảo với mong muốn cải thiện khả năng phản hồi của chabot.

1.3. Mục tiêu đề tài

Luận văn này sẽ nghiên cứu xây dựng trợ lý ảo cho sinh viên Khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ dựa trên mô hình Transformer. Trợ lý ảo này có khả năng trợ giúp sinh viên trong việc tìm kiếm thông tin liên quan đến Khoa Công nghệ Thông tin và Truyền thông, quy chế học vụ, học phần, …

1.4. Đối tượng và phạm vi nghiên cứu

Đối tượng và phạm vi nghiên cứu của luận văn là mô hình Transformer trên tập dữ liệu tiếng Việt được xây dựng để hỗ trợ sinh viên khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ.

1.5. Phương pháp nghiên cứu

Để thực hiện và giải quyết vấn đề về xây dựng một trợ lý ảo có khả năng trả lời tự động trên phương diện ngôn ngữ là tiếng Việt, các phương pháp nghiên cứu được thực hiện như sau:

- Sử dụng Internet để tìm kiếm các tài liệu có liên quan, các kỹ thuật đã được sử dụng.
- Tìm hiểu các kỹ thuật đã thu thập được, nghiên cứu có thể áp dụng cho vấn đề được đặt ra không và lên ý tưởng giải quyết vấn đề.
- Tìm kiếm và tìm hiểu những công cụ, những thư viện hỗ trợ cho viện giải quyết vấn đề, thử nghiệm các kết quả tìm kiếm.

1.6. Nội dung nghiên cứu

- Tìm hiểu lý thuyết về mô hình Transformer
- Cách thức hoạt động mô hình Transformer
- Vận dụng mô hình Transformer để xây dựng hệ thống trả lời tự động
- Nghiên cứu phương pháp xây dựng tập dữ liệu phục vụ mục tiêu đề tài.
- Viết báo cáo, đánh giá và so sánh kết quả đạt được.

1.7. Bố cục

Nội dung của quyển luận văn bao gồm 05 chương:

- Chương 1- Giới thiệu tổng quan: trình bày tổng quan về chatbot, mục tiêu đề tài, đối tượng, phạm vi, phương pháp và nội dung nghiên cứu.
- Chương 2 - Cơ sở lý thuyết: phân loại chatbot, nghiên cứu về mạng neural và các phần mở rộng của mạng mạng neural, mô hình Transformer.
- Chương 3- Phương pháp thực hiện: trình bày phương pháp để xây dựng hệ thống trợ lý ảp.
- Chương 4- Thực nghiệm: Trình bày cách thức triển khai thực nghiệm, cách thức xử lý mô hình.
- Chương 5- Kết luận: tổng kết kết quả đạt được của nghiên cứu và đề xuất hướng phát triển cho đề tài.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này giới thiệu các khái niệm cơ bản về chatbot, cùng với các khái niệm mạng neural nhân tạo, cách thức hoạt động của mạng neural, các mô hình mở rộng của mạng neural (RNN, LSTM) và mô hình Transformer.

2.1. Chatbot

Chatbot là một chương trình máy tính, được xây dựng để tương tác với người dùng bằng cách sử dụng ngôn ngữ tự nhiên, âm thanh hoặc hình ảnh dưới dạng một giao diện đơn giản. Yêu cầu cho việc xây dựng chatbot là hiểu được các ngôn ngữ tự nhiên của người dùng, từ đó đưa ra được câu trả lời cho yêu cầu của người dùng. Cách thức để xây dựng chatbot thường được chia thành ba loại: phân theo hướng tiếp cận, phân theo độ dài đoạn hội thoại và theo miền.

Ở phân loại theo hướng tiếp cận, phương pháp xây dựng chatbot được chia làm 2 loại là Generative model (Mô hình sinh) và Retrieval-based model (Mô hình trích xuất). Xây dựng chatbot sử dụng mô hình sinh sẽ không định nghĩa trước câu trả lời; thay vào đó, các câu trả lời sẽ tạo ra một cách tự động, mô hình sinh khá thông minh, tuy nhiên nó yêu cầu nhiều thuật toán phức tạp. Mô hình trích xuất sử dụng một tập câu trả lời được định nghĩa sẵn, sử dụng các thuật toán tìm kiếm để chọn ra các câu trả lời. Các thuật toán tìm kiếm khá đơn giản như sử dụng các luật, hoặc phức tạp hơn thì sử dụng các thuật toán phân lớp của máy học. Tuy nhiên, hệ thống này, không thể sinh ra câu trả lời mới, chỉ có thể sử dụng các câu trả lời có sẵn.

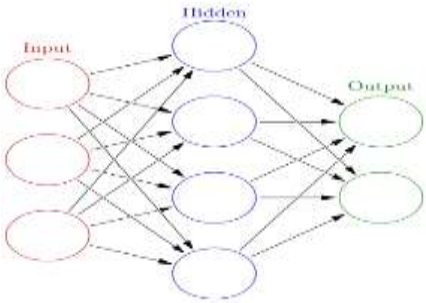
Với phân loại theo độ dài đoạn hội thoại, chatbot chia ra làm 2 loại là: Short-text Conversation (đoạn hội thoại ngắn) và Long-text Conversation (đoạn hội thoại dài). Với đoạn hội thoại ngắn, yêu cầu đặt ra là chatbot sẽ chỉ cần trả lời hay phản hồi một câu hỏi từ phía người dùng đưa ra. Còn với đoạn hội thoại dài, chatbot sẽ được xây dựng để có thể đưa ra các câu trả lời, các phản hồi qua nhiều lượt hỏi đáp qua lại giữa giữa người dùng và chatbot, yêu cầu đặt ra là chatbot cần giữ được thông tin của cuộc hội thoại đang thực hiện.

Phân loại theo miền thì cũng được chia làm 2 loại là Open domain (miền mở) và Closed Domain (miền đóng). Ở miền mở, người dùng có thể tạo ra cuộc hội thoại theo bất kỳ lĩnh vực nào, không cần định nghĩa trước mục đích hay ý định và chatbot có thể trả lời bất kỳ câu hỏi ở tất cả mọi chủ đề. Tuy nhiên, do có quá nhiều chủ đề nên dẫn đến chatbot có các câu trả lời là chưa được hợp lý. Ở miền đóng, giới hạn được những câu hỏi đầu vào và câu trả lời, hệ thống chỉ trả lời các câu hỏi trong một chủ đề đã được lập trình sẵn, đồng thời cũng sẽ tránh được các câu trả lời chưa được hợp lý.

2.2. Mạng neural nhân tạo

2.2.1. Kiến trúc mạng neural nhân tạo

Mạng neural nhân tạo (Artificial Neural Networks – ANN) được xây dựng dựa trên sự mô phỏng hoạt động của hệ thần kinh con người. ANN là một mạng phức tạp kết nối các đơn vị tính toán lại với nhau, trong đó mỗi đơn vị tính toán gọi là neural nhân nhân tạo, có thể có nhiều đầu vào nhưng chỉ có một đầu ra duy nhất được minh họa ở Hình 1.



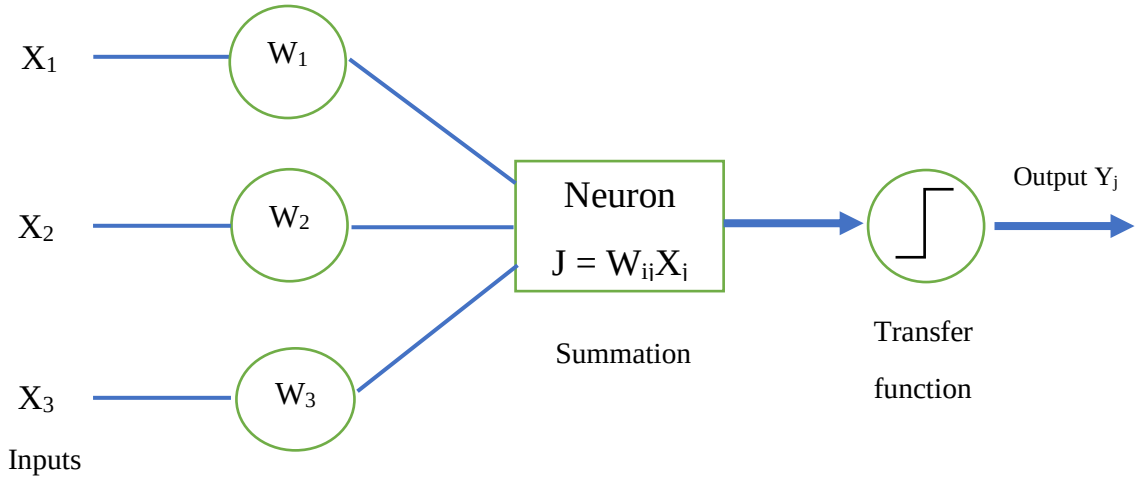
Hình 1 Mô hình mạng neural

(Nguồn: https://en.wikipedia.org/wiki/Artificial_neural_network)

Kiến trúc chung của một ANN gồm có 3 thành phần: Input layer, Hidden layer, Output layer. Trong đó, các các lớp ẩn (Hidden layer) sẽ nhận dữ liệu từ các neural ở lớp trước (layer) và chuyển đổi các input này cho các lớp xử lý phía sau. Trong một ANN có thể có nhiều lớp ẩn. Các PE (Processing Elements) của ANN gọi là neural, mỗi neural nhận các dữ liệu vào xử lý chúng và cho ra kết quả duy nhất. Kết quả xử lý của một neural có thể làm input cho các neural khác.

2.2.2. Quá trình xử lý thông tin của một mạng neural

Cũng giống như hệ thống mạng thần kinh của con người, mạng neural nhân tạo cũng nhận vào các thông tin và xử lý chúng. Quá trình xử lý của một ANN được biểu diễn như Hình 2.



Hình 2 Quá trình xử lý của mạng neural

Trong đó:

- Input: mỗi Input tương ứng với một thuộc tính (attribute) của dữ liệu vào (patterns).
- Output: kết quả tính toán của một ANN là một giải pháp cho một vấn đề.
- Connection weight (*W*) (trọng số liên kết): đây là thành phần rất quan trọng với một ANN, nó thể hiện độ quan trọng của dữ liệu đầu vào và đối với quá trình xử lý thông tin (quá trình chuyển đổi từ layer này sang layer khác). Quá trình học của ANN thực ra là quá trình điều chỉnh các trọng số (weight) của các input để có được kết quả mong muốn.
- Summation function (hàm tổng): Tính tổng các trọng số của tất cả các input được đưa vào mỗi neural (phần xử lý PE). Hàm tổng của một neural đối với *n* input được tính theo công thức sau:

$$Y = \sum_{i}^{n} X_i W_i \quad (1)$$

- Transfer function (hàm chuyển đổi): hàm tổng của một neural cho biết khả năng kích hoạt (activation) của neural đó, còn gọi là kích hoạt bên trong (internal activation). Các kết quả tính toán của một neural có thể được chuyển đến để làm input cho một neural khác hoặc không. Giả sử có 3 input X_1, X_2, X_3 tưởng ứng với 3 weight là W_1, W_2, W_3, mối quan hệ giữa internal activation và kết quả được thể hiện bằng hàm chuyển đổi như sau:

$$Y = X_1W_1 + X_2W_2 + X_3W_3 = \alpha \quad (\text{Với } \alpha \text{ là số thực bất kỳ}) \quad (2)$$

$$Y_T = \frac{1}{1 + e^{-\alpha}} = \beta \quad (\text{Với } \beta \text{ là số thực trong đoạn [0;1]}) \quad (3)$$

Việc lựa chọn hàm chuyển đổi có tác động rất lớn đến kết quả của ANN. Hàm chuyển đổi phi tuyến tính được sử dụng phổ biến là hàm *sigmoid*:

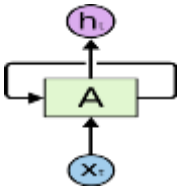
$$Y_T = \frac{1}{1 + e^{-Y}} \quad (4)$$

Trong đó Y_T là hàm chuyển đổi và Y là hàm tính tổng.

Kết quả của hàm *sigmoid* thuộc khoảng [0;1], nên còn gọi là hàm chuẩn hóa. Kết quả xử lý tại các neural đôi khi rất lớn, vì vậy *transfer* function được sử dụng để xử lý kết quả này trước khi chuyển đến lớp tiếp theo. Đôi khi thay vì sử dụng *transfer* function, người ta sử dụng giá trị ngưỡng (*threshold* value) để kiểm soát các kết quả của các neural tại một lớp nào đó trước khi chuyển các kết quả này đến các lớp tiếp theo. Nếu kết quả của một neural nào đó nhỏ hơn giá trị ngưỡng thì nó sẽ được chuyển đến lớp tiếp theo.

2.2.3. Recurrent Neural Network

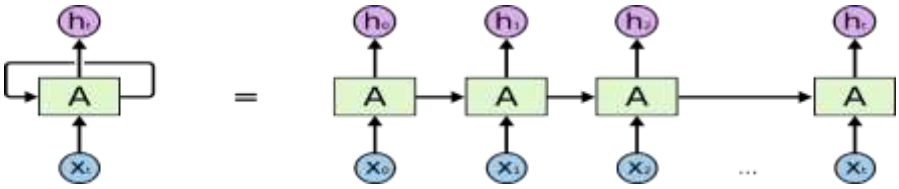
Trong mô hình ANN thông thường (Feed forward network), chỉ xem các input là dữ liệu độc lập, không có liên kết với nhau. Tuy nhiên, với ngôn ngữ tự nhiên, các từ trong một câu văn lại có liên kết với nhau quyết định nên ý nghĩa của câu văn đó. Chính vì vậy, việc áp dụng một ANN thông thường vào các bài toán xử lý ngôn ngữ tự nhiên thường không đạt được kết quả như mong muốn. Để có thể khắc phục được nhược điểm trên, mô hình mới được ra đời chính là Recurrent Neural Network– RNN. RNN coi dữ liệu đầu vào là một chuỗi (sequence) liên tục, nối tiếp nhau theo thứ tự thời gian. Ví dụ như một câu văn, được coi là một chuỗi các từ hoặc một chuỗi các ký tự. Tại thời điểm t, với input là x_t ta có kết quả là h_t. Tuy nhiên khác với ANN, h_t lại được sử dụng để làm input để tính kết quả cho thời thời điểm *t+1*. Điều này cho phép RNN có thể lưu trữ và truyền thông tin đến thời điểm tiếp theo. Mô hình hoạt động của RNN được mô tả như Hình 3.



Hình 3 Recurrent Neural Network

(Nguồn: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Để dễ hình dung hơn. Ta chuyển mô hình RNN thành dạng phẳng như Hình 4, với $x_0, x_1, x_2, \ldots x_t$ là dữ liệu đầu vào tại thời điểm (timestep) *t=0, t=1, t=2, ...t=n.*



Hình 4 Mở rộng của một RNN

(Nguồn: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

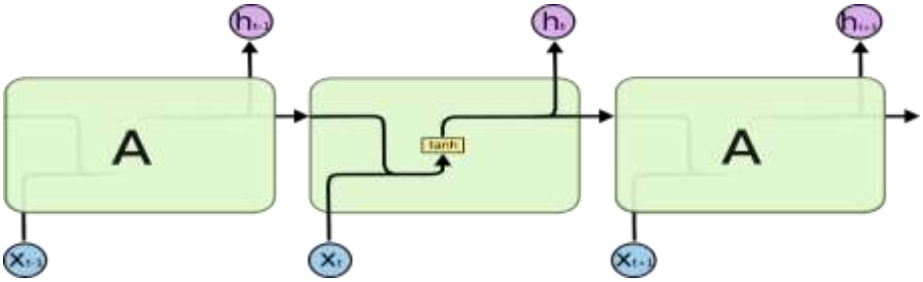
Trong ANN một kết quả cũng có thể trở thành input cho một lớp khác. Tuy nhiên điểm khác nhau giữa ANN và RNN là:

- ANN sử dụng giá trị *weight* khác nhau cho từng lớp. Ví dụ như kết quả của lớp thứ nhất chỉ là X_1 được nhân với W_1 , kết quả của lớp thứ hai X_2 chỉ nhân được với W_2.
- RNN sử dụng một mạng neural duy nhất (1 lớp) để tính giá trị kết quả cho từng timestep. Do đó các kết quả *X* sẽ trở thành output sẽ chỉ nhân với cùng một *W* duy nhất.

Nói một cách khác, RNN sử dụng một phép tính duy nhất cho từng phần tử của chuỗi dữ liệu đầu vào, các kết quả sẽ được dùng làm input cho các tính toán kế tiếp. Về mặt lý thuyết thì RNN có khả năng "nhớ" và kết nối các thông tin ở phía trước với thông tin hiện tại, tuy nhiên thực tế thì không thể; đây được gọi là vấn đề "phụ thuộc xa".

2.2.4. Long Short Term Memory network

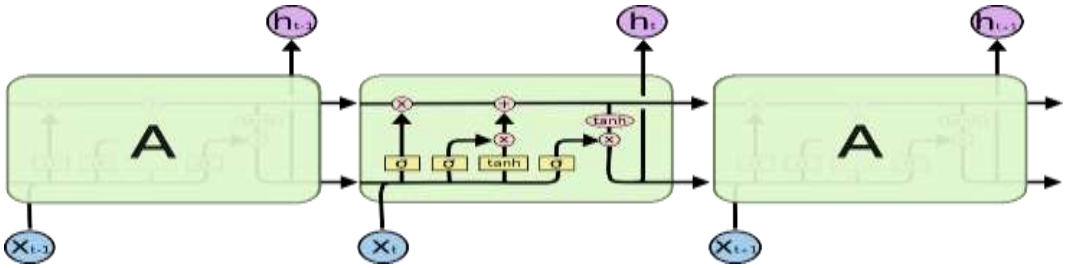
Long Short Term Memory network – LSTM [11], là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa nhờ khả năng ghi nhớ được thông tin. Với mọi mạng RNN đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng neural. Với mạng RNN chuẩn, các mô-đun này có cấu trúc rất đơn giản, thường là một tầng *tanh* như Hình 5.



Hình 5 Mô-đun RNN

(Nguồn: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun này có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng neural, LSTM có tới 4 tầng tương ứng với nhau một cách rất đặc biệt được minh họa ở Hình 6.



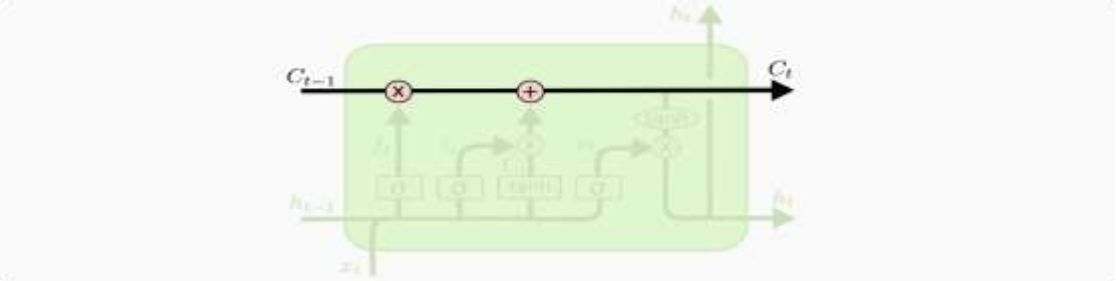
Hình 6 Mô-đun LSTM

(Nguồn: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Trong đó:

- Hình chữ nhật là các lớp ẩn của mạng neural.
- Hình tròn biểu diễn toán tử Pointwise.
- Đường kẻ gộp lại với nhau biểu thị phép nối các toán hạng, và.
- Đường rẽ nhánh biểu thị cho sự sao chép tự vị trí này sang vị trí khác.

Phần quan trọng nhất trong LSTM chính là các trạng thái tế bào (cell state). Trạng thái tế bào giống như băng truyền, nó chạy suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút, vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi, được minh họa như Hình 7.



Hình 7 Trạng thái tế bào

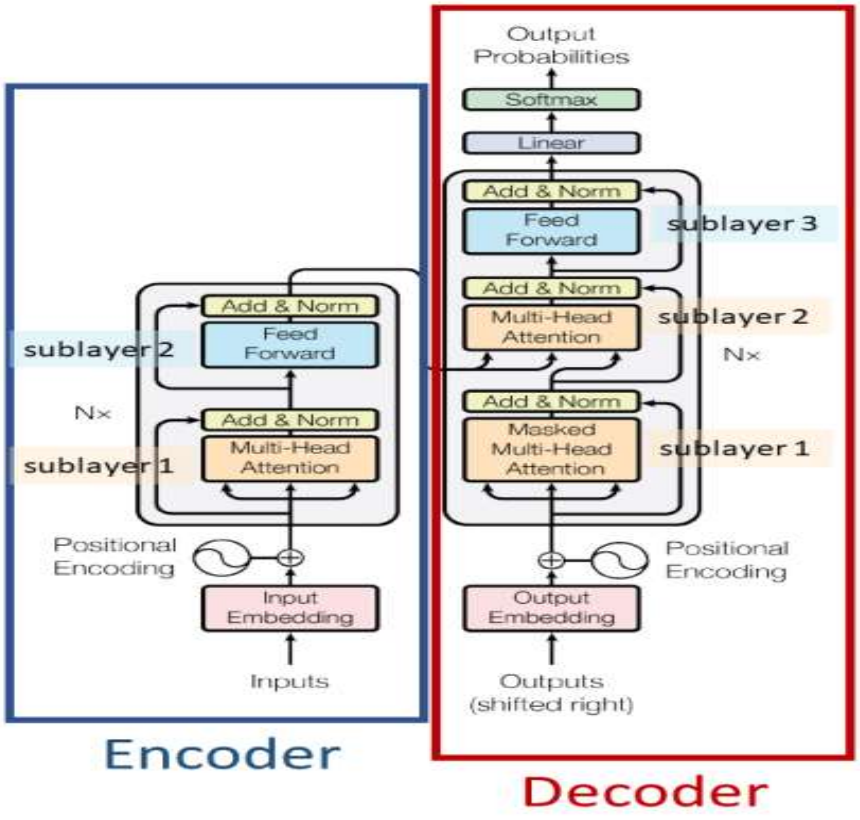
(Nguồn: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế bào, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate). Các cổng là nơi sàng lọc thông tin qua đó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân. Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0,1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩ cho tất cả thông tin đi qua nó. Một LSTM gồm có cả 3 cổng như vậy để duy trì và điều hành trạng thái tế bào.

2.3. Transformer

2.3.1. Tổng quan

Transformer là một mô hình học sâu được đề xuất vào năm 2017 bởi Vaswani và các cộng sự [10]. Transformer cũng sử dụng hai phần Encoder và Decoder khá giống với RNN, tuy nhiên Transformer không yêu cầu dữ liệu đầu vào phải được đưa vào theo tuần tự, ví dụ như khi dữ liệu đầu vào là một câu, RNN phải thực hiện theo thứ tự từ đâu câu đến cuối câu; Transformer có thể thực hiện song song bằng việc đưa vào cả câu văn thay vì đưa từng từ theo thứ tự như RNN, với cơ chế này sẽ thay thế cho “recurrent” của RNN. Mô hình Transformer được trình bày ở Hình 8.

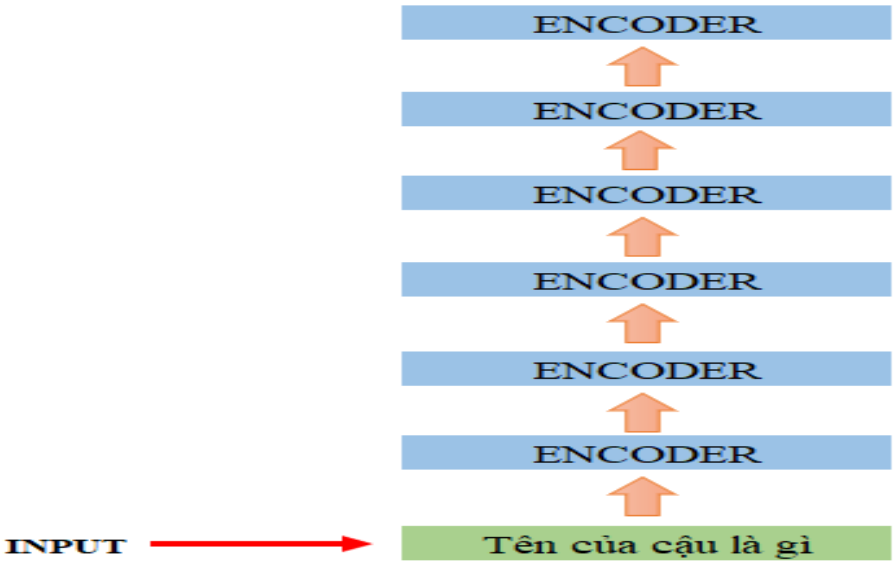


Hình 8 Mô hình Transformer

(Nguồn: http://jalammar.github.io/illustrated-transformer/)

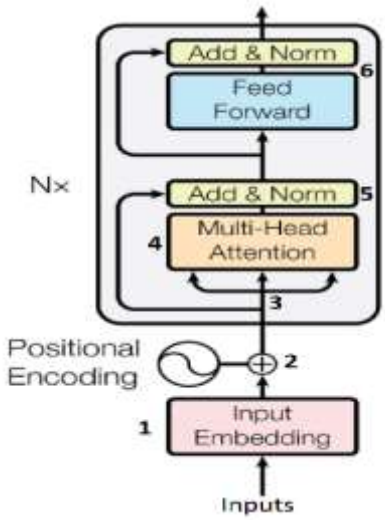
2.3.2. Encoder

Encoder là một khối gồm 6 Encoder xếp chồng lên nhau. Tất cả các Encoder đều giống nhau về cấu trúc (nhưng khác nhau về trọng số) (Hình 9).



Hình 9 Cấu tạo tổng quát của Encoder

Vai trò của Encoder là mã hóa câu đầu vào thành một tập vector có attention (sự "chú ý") giữa các từ trong câu hay nói cách khác kết quả của quá trình Encoder là một ma trận có kích thước S x D_{model}, với S là độ dài câu đầu vào (input), D_{model} là kích thước của phép nhúng có trọng số (Embedding weights) được sử dụng để huấn luyện mạng. Mô hình chi tiết bên trong của 1 khối Encoder được trình bày ở Hình 10.



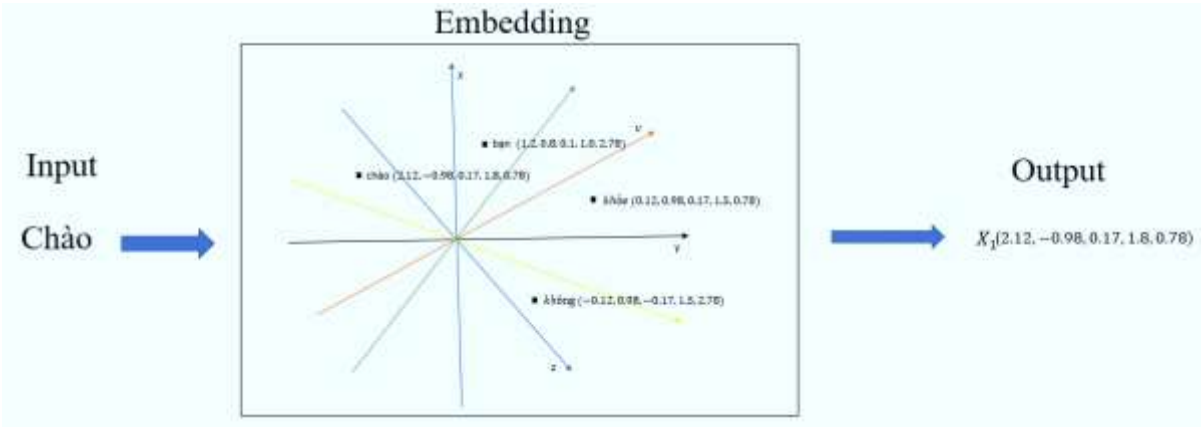
Hình 10 Cấu tạo chi tiết của 1 Encoder

(Nguồn: http://jalammar.github.io/illustrated-transformer/)

Phần tiếp theo sẽ trình bày từng khối bên trong Encoder và cách xây dựng ma trận attention.

2.3.2.1. Embedding

Nhiệm vụ của lớp Embedding là biểu diễn các từ trong câu đầu vào thành vector bằng cách sử dụng các phương pháp nhúng từ. Hình 11 minh họa vector hóa từ "Chào".



Hình 11 Ví dụ một từ được vector hóa

2.3.2.2. Positional Encoding

Các từ trong câu đầu vào sau khi đã được Embedding, sẽ được đi qua lớp Positional Encoding, lớp này giúp lưu giữ được thông tin vị trí các từ có trong câu đầu vào, thông tin vị trí các từ sẽ được lưu bằng cách cộng các PE vector vào Embedding vector, để tạo thành một vector có lưu thông tin vị trí các từ có trong câu.

Các PE vector này có số chiều bằng với số chiều của Embedding vector, với mỗi một vị trí có trong Embedding vector, thì giá trị tại vị trí tướng ứng ở PE vector sẽ được tính theo theo công thức sau:

$$PE_{(pos,2i)} = sin(\frac{pos}{1000^{\frac{2i}{d_{model}}}}) \quad (5)$$

$$PE_{(pos,2i+1)} = \cos(\frac{pos}{1000^{\frac{2i}{d_{model}}}}) \quad (6)$$

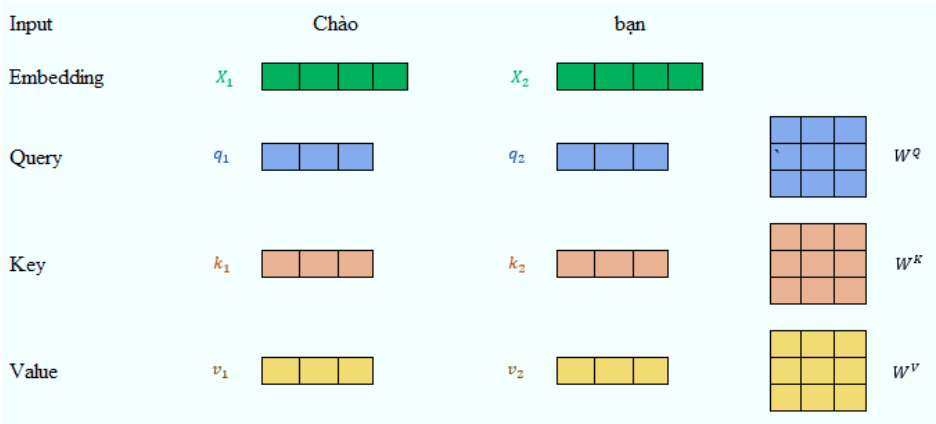
Trong đó:

- pos: là vị trí từ trong câu.
- PE: là giá trị phần tử thứ *i* trong Embedding vector có độ dài d_{model}.

2.3.2.3. Self-Attention

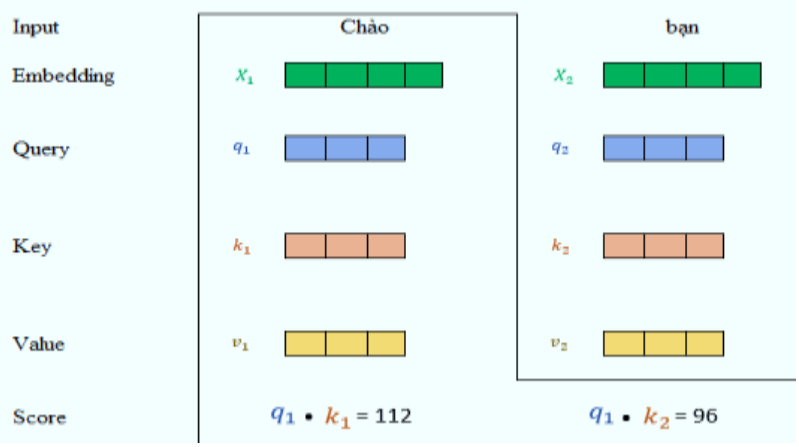
Self-Attention là kỹ thuật giúp Transformer "hiểu" được sự liên quan giữa các từ trong câu. Ví dụ như câu: "Tôi thích hoa vì nó thật đẹp". "Nó" ám chỉ điều gì? Có phải "hoa". Với con người thì thật đơn giản dễ hiểu nhưng với máy tính thì không. Chính vì vậy Self-Attention được sinh ra để giải quyết vấn đề này. Khi mô hình xử lý từ (các từ trong câu đầu vào), Self-Attention sẽ xem xét các vị trí khác trong câu đầu vào để tìm ra được sự liên kết và giúp cho mô hình mã hóa tốt hơn. Các bước thực hiện của Self- Attention như sau:

Bước 1: Tạo ra bộ 3 vector là vector *Q* (Query), vector *K* (Key) và vector *V* (Value), các vector này được tạo ra bằng cách nhân các vector đầu vào X_n (n là số thứ tự của từng từ) với ba ma trận W_Q, W_K, W_V (các ma trận này được sinh ra ngẫu nhiên và thay đổi trong quá trình đào tạo mô hình), ở ví dụ mình họa Hình 12, lần lượt nhân vector X_1với các ma trận W_Q, W_K, W_V sinh ra được vector q_1, k_1 và v_1 (tương ứng với từ "Chào"). Tương tự với từ "bạn", cũng sẽ có bộ ba vector q_2, k_2, v_2. Tùy theo độ dài câu mà sinh ra được số bộ vector tương ứng.



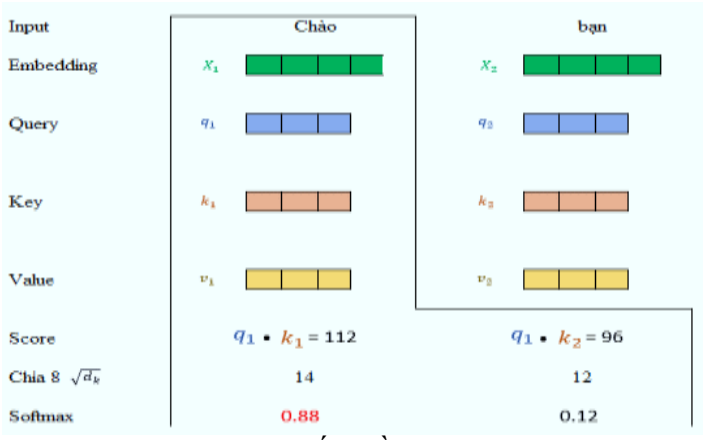
Hình 12 Các vector và ma trận trong Self-Attention

Bước 2: lấy vector q_1 vừa được tạo nhân lần lượt với từng vector k_1 và k_2 (Hình 13).



Hình 13 Nhân vector q1 với k1 và k2

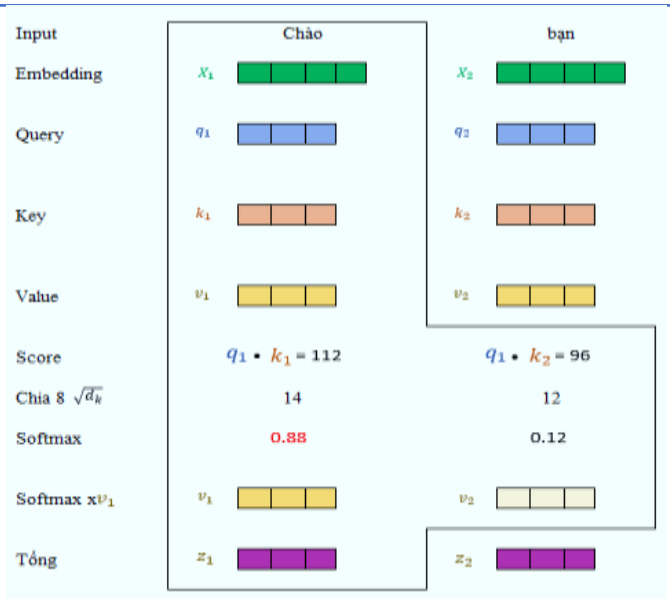
Bước 3: là chia cho căn bậc 2 số chiều của *vector* đầu vào. Điều này làm cho các gradient ổn định hơn, sau đó chuyển hết cho *softmax* (Hình 14). Điểm *softmax* này xác định mức độ mà mỗi từ sẽ được biểu diễn tại vị trí này. Rõ ràng là từ vị trí này, sẽ có điểm *softmax* cao nhất, nhưng đôi khi sẽ hữu ích khi xem một từ khác có liên quan đến từ hiện tại.



Hình 14 Chia score cho số chiều dài của k1 và k2

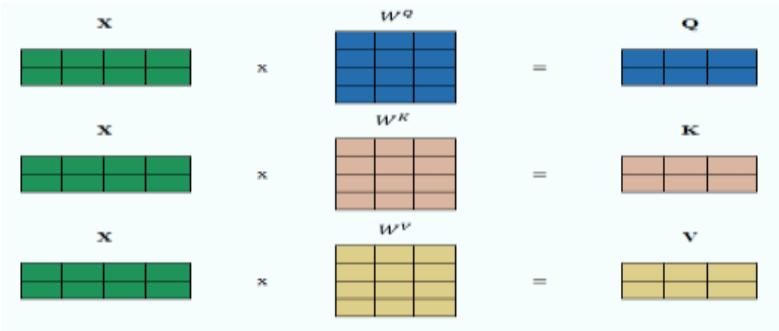
Bước 4: lấy các giá trị *softmax* vừa tính được nhân với từng vector v_1 và v_2. Ở bước này chủ yếu là để giữ nguyên các giá trị của các từ mà chúng ta muốn tập trung vào và loại bỏ các từ không liên quan.

Bước 5: cộng các giá trị có xác suất lớn ở bước 5 lại với nhau thu được vector z_1, rồi tiếp tục thực hiện tương tự với vector X_2 (tương ứng với từ "bạn"), cũng sẽ lấy q_2 nhân lần lượt với k_1 và k_2, qua các bước tính toán lại thu được một vector z_2 khác, quá trình sẽ được thực hiện cho đến khi kết thúc câu, minh họa Hình 15.



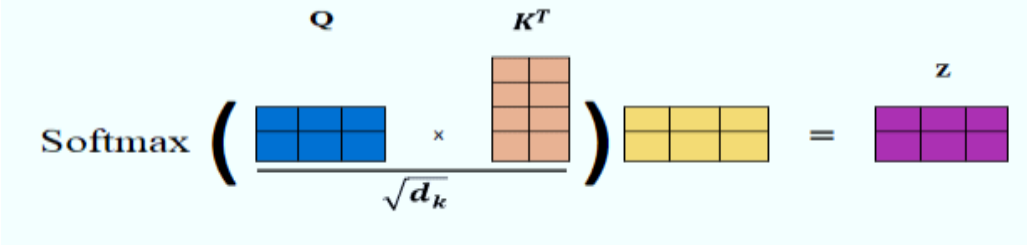
Hình 15 Các bước tổng quát tạo vector z

Tuy nhiên, trong thực tế khi thực hiện các bài toán, các phép tính lại được thực hiện dưới dạng các ma trận cho việc tính toán được nhanh hơn. Đầu tiên, tạo ma trận *X*, bằng cách ghép các vector là các từ trong câu đầu vào lại với nhau, với mỗi dòng của ma trận *X* tương ứng với một từ, sau đó lần lượt nhân ma trận *X* với từng ma trận W_Q, W_K, W_v để tạo ra các ma trận Q, K và V (giống với qua trình tạo ra các vector q_1, k_1, v_1) (Hình 16).



Hình 16 Tạo các ma trận Q, K, V từ ma trận X

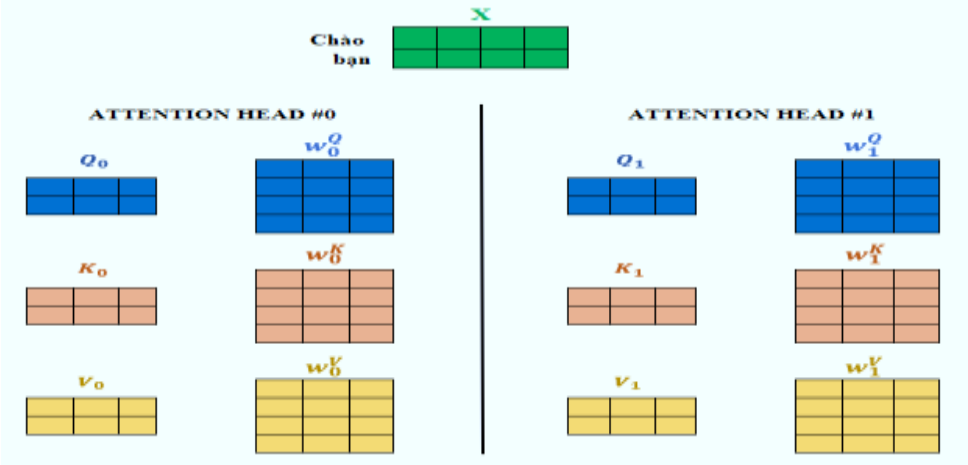
Vì đang thực hiện tính toán trên ma trận nên có thể thực hiện nhanh hơn, không phải thông qua nhiều bước như trên từng từ nữa, nên chỉ cần áp dụng thẳng *softmax* vào ta sẽ thu được ma trận attention Z, theo cách như sau (Hình 17).



Hình 17 Tạo ma trận attention Z

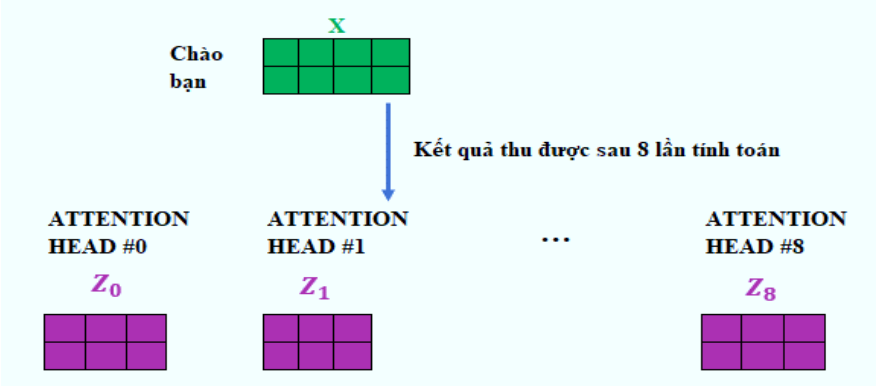
2.3.2.4. Multi-head Attention

Ở cơ chế Self-Attention các từ chỉ tập chung vào chính nó, nhưng điều đó lại là điều không mong muốn, điều mong muốn là sự liên kết giữa những từ ở các vị trí khác nhau trong câu. Chính vì vậy Multi-head Attention ra đời. Ý tưởng rất đơn giản là thay vì sử dụng một Attention (1 head) thì sử dụng nhiều Attention khác nhau (Multi-head) và biết đâu mỗi Attention chú ý đến một phần khác nhau trong câu. Với cơ chế Multi-head này sẽ cung cấp cho lớp Attention nhiều "không gian con biểu diễn", không chỉ có một mà có nhiều bộ ma trận trọng số *Q*, *K*, *V* (Transformer sử dụng 8 Attention head, vì vậy kết thúc với 8 bộ mã hóa / giải mã). Mỗi bộ này được khởi tạo ngẫu nhiên. Sau khi huấn luyện, mỗi tập hợp được sử dụng đối chiếu các Embedding (hoặc vector từ bộ mã hóa/ giải mã thấp hơn) và một không gian con khác (Hình 18).



Hình 18 Bộ các ma trận Q, K, V

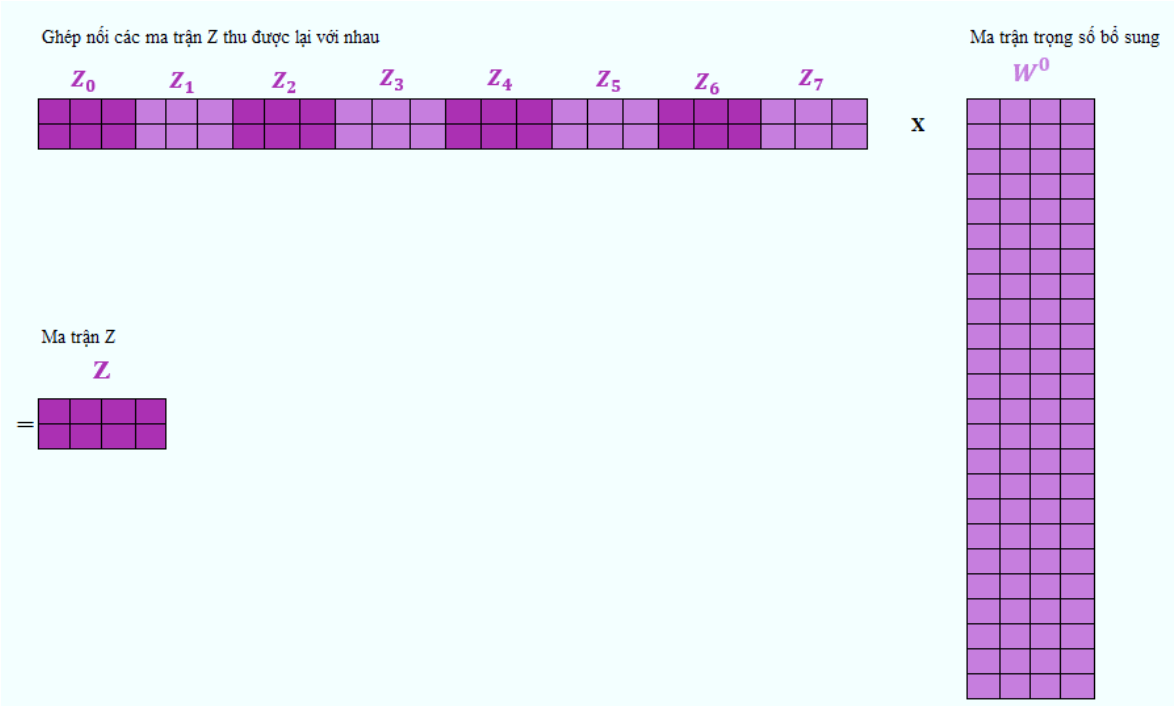
Nếu khi thực hiện các tính toán Self-Attention tương tự đã nêu ở trên, thì sau 8 lần tính khác nhau với các ma trận trọng lượng khác nhau sẽ thu 8 ma trận *Z* khác nhau với mỗi ma trận sẽ chú ý vào các phần khác nhau trong câu (Hình 19).



Hình 19 Các ma trận thu được

Tuy nhiên, lớp kế tiếp lại không mong nhận được 8 ma trận cùng một lúc mà chỉ cần 1 ma trận. Để giải quyết vấn đề này, chúng ta có thể thực hiện bằng cách nối

các ma trận lại với nhau và sau đó nhân chúng với một ma trận trọng số bổ sung W^0 (Hình 20).

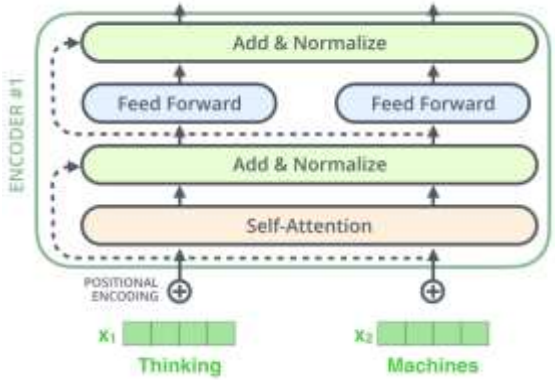


Hình 20 Tạo ma trận Z

Ma trận *Z* cuối cùng này, sẽ chứa thông tin với các liên kết giữa các từ ở trong một câu. Cũng chính là kết quả đã được đề nhắc đến ở đầu quá trình Encoder.

2.3.2.5. The Residuals

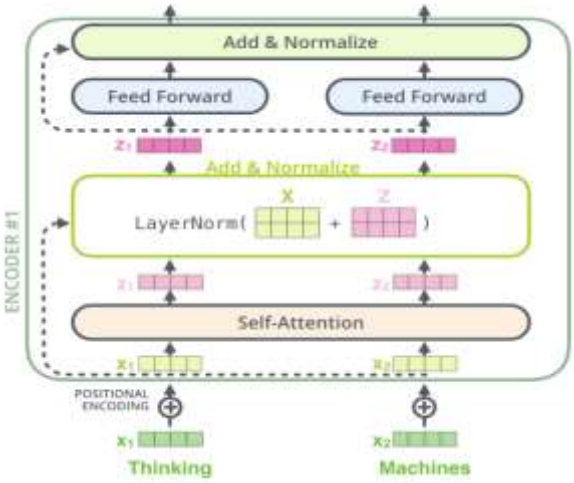
Giữa các khối trong Encoder và cả Decoder còn có một thành phần ở giữa, được liên kết lại với nhau như minh họa ở Hình 21, được gọi là các đường residuals.



Hình 21 Các đường residual

(Nguồn: http://jalammar.github.io/illustrated-transformer/)

Các đường nối này (minh họa Hình 22), có chức năng như sau: sau khi một từ đã đi qua lớp Position Encoding sẽ sinh ra vector x_1 và khi x_1đi qua lớp Self-Attention sẽ sinh ra vector z_1. Vector mới tạo ra z_1 sẽ được cộng với vector x_1 sau đó đưa vào một lớp Layer Normalization [12], lớp này có chức năng giúp cho mô hình huấn luyện được nhanh hơn và kết quả tốt hơn. Các đường residual này sẽ được lập đi lập lại qua mỗi khối.



Hình 22 Các lớp Layer Normalization

(Nguồn: http://jalammar.github.io/illustrated-transformer/)

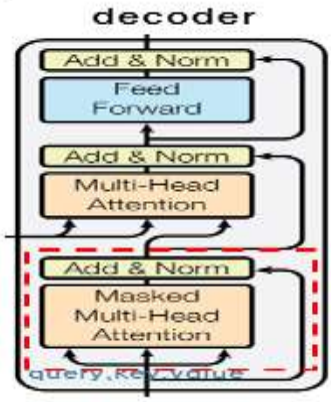
2.3.2.6. Feed Forward

Sau khi đã đi qua lớp Layer Normalization các vector *z* được đưa qua một mạng fully connected trước khi được đẩy qua quá trình Decoder. Vì các vector này không phụ thuộc vào nhau, nên có thể thực hiện tính toán một cách song song cho cả một câu.

2.3.3. Decoder

Các câu đầu vào ở quá trình Decoder (Hình 23) sẽ có phần đặt biệt hơn là được thêm vào các tiền tố và hậu tố kết thúc và bắt đầu câu, rồi mới đi vào các khối bên trong Decoder, cùng với đó sẽ có thêm cơ chế masking để giúp cho quá trình huấn luyện mô hình được nhanh hơn. Kết quả của quá trình cũng sẽ là một ma trận attention có kích thước T x D_{model}, với T là độ dài câu đã thêm các tiền tố và hậu tố, nhưng ma trận này không chứa thông tin liên kết các từ phía sau mà chỉ có thông tin của từ phía trước.

Ở Decoder, sẽ chỉ tập chung vào 2 khối là Masked Multi-Head Attention và Multi-Head Attention (Encoder-Decoder Attention) Vì chỉ có 2 khối này là có phần khác so với Encoder, còn lại là gần như tương tự.

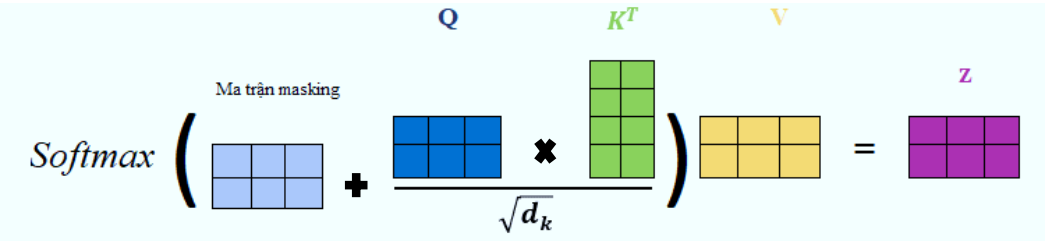


Hình 23 Decoder của Transformer
(Nguồn: https://pozalabs.github.io/transformer/)

2.3.3.1. Masked Multi-Head Attention

Giống với Multi-head Attention ở Encoder, quá trình này cũng cũng sẽ tạo ra ma trận attention *Z* với các từ có trong câu, nhưng nếu chỉ thực hiện mỗi quá trình tính toán các Self-Attention thì mô hình không thể nào thực hiện trình dự đoán từ kế được, để có thể thực hiện dự đoán, thì quá trình này của Decoder sẽ tiếp tục thực hiện thêm một cơ chế masking, cơ chế này dùng che đi các từ ở tương lai, tức là các từ ở phía sau một từ trong câu, ví dụ như câu "chào bạn" thì lúc này từ "bạn" sẽ bị che đi, chỉ còn lại từ "chào", cơ chế masking này như sau:

- Ở bước tính các Self-Attention, cũng sẽ tính tương tự như là ở Encoder, cũng sẽ tạo các ma trận *Q, K, V* từ kết quả ở bước Positional Encoding, và cũng sẽ tính trên các ma trận *Q, K, V* như ở Hình 17, tuy nhiên trước khi tính *softmax* sẽ cộng thêm ma trận masking vào, rồi mới tính *softmax*, và cuối cùng là nhân kết quả *softmax* với vector *V*. Công thức tổng được mình họa như Hình 24.



Hình 24 Công thức tạo ma trận có masking

- Ma trận masking này có kích thước bằng với ma trận *V* và có hình dạng như Hình 25.

0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	0	$-\infty$	$-\infty$	$-\infty$
0	0	0	$-\infty$	$-\infty$
0	0	0	0	$-\infty$
0	0	0	0	0

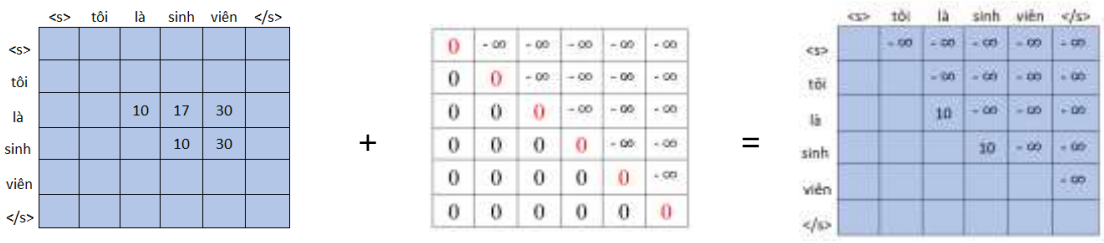
Hình 25 Ma trận masking

- Ví dụ như câu đầu vào "<s> tôi là sinh viên </s>", có ma trận kết quả sau khi lấy ma trận *Q* nhân với ma trận chuyển vị *K* và chia cho căn bậc hai số chiều của vector đầu vào như sau (Hình 26):

	<s>	tôi	là	sinh	viên	</s>
<s>						
tôi						
là			10	17	30	
sinh				10	30	
viên						
</s>						

Hình 26 Ví dụ ma trận kết quả của câu <s> tôi là sinh viên </s>

- Sau đó cộng với ma trận masking, để tạo thành ma trận có sự masking (Hình 27).



Hình 27 Kết quả quá trình tạo ma trận masking

Ở các vị trí là $-\infty$, hàm *softmax* sẽ trả về 0, tức là lúc này sự chú ý của câu tại đó là 0, nên sẽ được đánh dấu lại là masking (Hình 28).

	<s>	tôi	là	sinh	viên	</s>
<s>		masking	masking	masking	masking	masking
tôi			masking	masking	masking	masking
là			10	masking	masking	masking
sinh				10	masking	masking
viên						masking
</s>						

Hình 28 Kết quả masking khi có softmax

Sau khi đã thực hiện *softmax* xong thì kế tiếp là nhân kết quả đó với vector *V* để thu về ma trận attention *Z* và ma trận này chỉ chứa thông tin của từ phía một từ trong câu.

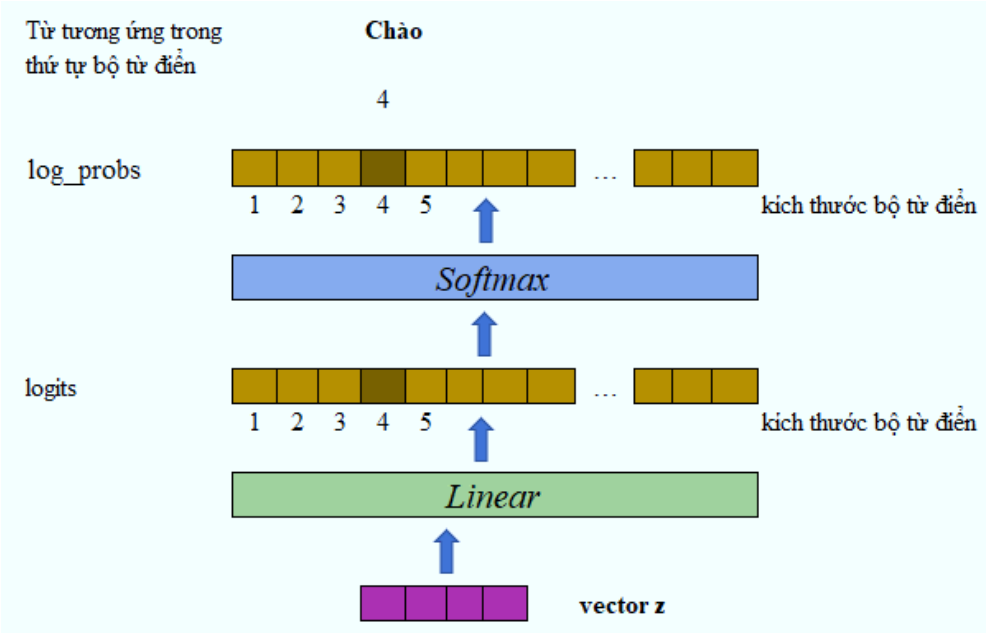
Thực hiện các tính toán tương tự, sau 8 lần tính toán với các ma trận trọng lượng khác nhau, sẽ thu được 8 ma trận *Z* khác nhau. Sau đó lại nối các ma trận *Z* này lại với nhau và thu về một ma trận attention *Z* duy nhất. Nhờ vào cơ chế masking này mà quá quá trình Decoder có thể đưa cả một câu vào để thực hiện huấn luyện mô hình, thay vì phải đưa tuần tự từng từ một.

2.3.3.2. Encoder-Decoder Attention

Tương ứng với Multi-Head Attention của khối Encoder thì thay vì được gọi cùng tên thì ở khối Decoder lại gọi là Encoder-Decoder Attention. Ma trận attention *Z* đã nhận được ở quá trình Masked Multi-Head Attention sẽ được nhân với duy nhất một ma trận W_Q để tạo thành tập vector *Q*. Ở quá trình Encoder sau khi cho ra tập các vector attention Z, cũng sẽ tiến hành nhân với tập các ma trận W_K, W_V để tạo thành 2 tập vector là *K* và *V*. Sau đó lại thực hiện các tính toán tương tự như lớp Multi-Head Attention ở quá trình Encoder. Quá trình được lặp lại 8 lần tương tự như ở Encoder. Kết quả đầu ra của quá trình này là một tập vector *Z* hay một ma trận *Z* có kích thước *T* x D_{model} đã được nhắc đến ở đầu quá trình Decoder.

2.3.3.3. Final Linear and Softmax Layer

Công việc của lớp Final Linear và Softmax layer chính là đưa các vector thu được ở cuối quá trình Decoder thành các từ có ý nghĩa. Quá trình này được minh họa ở Hình 29.



Hình 29 Cơ chế dự đoán từ kế tiếp

Lớp Linear là một mạng neural *fully connected* đơn giản, các vector thu được sẽ được chiếu lên một không gian vector lớn hơn có số chiều hay số ô đúng bằng độ dài của bộ từ điển (logits). Kế tiếp sử dụng hàm *softmax* để áp lên không gian tham chiếu, đưa những điểm trên không gian tham chiếu đó thành xác suất (log_probs). Tại điểm hay ô có xác suất cao nhất, mang tham chiếu với vị trí tương ứng trong bộ từ điển sẽ thu được từ cần dự đoán xuất hiện kế tiếp, tương ứng với từng vector có trong tập vector hay ma trận *Z* thu được. Kết quả cuối cùng sẽ là câu được dự đoán.

2.4. Thuật toán K-Nearest Neighbors

Thuật toán K-Nearest Neighbors hay kNN cho rằng các dữ liệu tương tự sẽ nằm gần nhau trong một không gian, từ đó việc của kNN chỉ là đi tìm các điểm *k* gần với dữ liệu cần kiểm tra nhất, việc tìm khoảng cách giữa hai điểm trong một không gian tùy thuộc vào trưởng hợp sử dụng mà có công thức tính khác nhau.

2.5. Phương pháp đánh giá

Phương pháp đánh giá BLEU cho các hệ thống dịch máy được được đề xuất bởi Papineni và cộng sự của ông [13], đây là thuật toán dùng để đánh giá chất lượng của các văn bản được dịch từ ngôn ngữ này sang ngôn ngữ khác và các vấn đề liên quan đến xử lý ngôn ngữ tự nhiên. BLEU có thể tham chiếu các chuỗi đầu ra do máy tạo ra với các bộ chuẩn tham chiếu. Cách tính điểm BLEU được tính theo công thức như sau:

$$BLEU = BP.e^{\sum_{n=1}^{4}\frac{1}{n}log_e P_n} \quad (7)$$

Trong đó:

- Chỉ số *BP* (Brevity penalty), bao gồm các tham số *c* là tổng số lượng các từ trong bản dịch cần đánh giá (candidate translation) từ hệ thống dịch máy, *r* là tổng số lượng các từ trong bản dịch tham khảo từ con người (reference translation). BP được tính như sau:

$$BP = \begin{cases} 1 & (\text{nếu } c > r) \\ e^{(1-\frac{r}{c})} & (\text{nếu } c \leq r) \end{cases} \quad (8)$$

- P_n là chỉ số modified n-gram precision, biểu diễn mức độ trùng khớp của văn bản cần đánh giá từ hệ thống dịch máy so với các bản dịch tham khảo từ con người. Được tính như sau:

$$P_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')} \quad (9)$$

- $Count_{clip}(n-gram)$ là số lượng các cụm có *n* từ liên tiếp (n-gram) trùng nhau giữa bản dịch cần đánh giá (candidate translation) và bản dịch tham khảo từ con người (reference translation).
- $Count(n-gram')$ là số lượng các cụm có *n* từ liên tiếp của bản dịch từ hệ thống dịch máy.

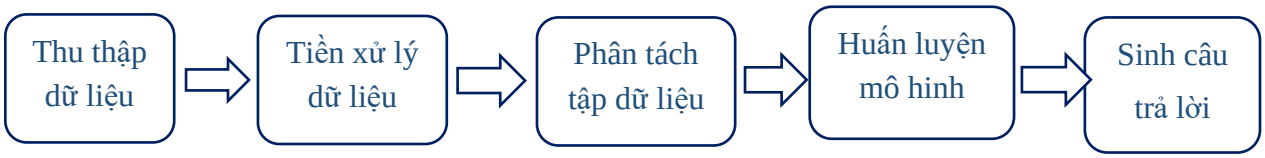
Giá trị của BLEU nằm trong khoảng từ 0 đến 1. Khi giá trị BLEU càng gần 1, chứng tỏ hệ thống dịch máy càng gần sát so với các bản dịch tham khảo. Với cách tính điểm như trên, luận văn này sẽ áp dụng tính cho các câu trả lời của chatbot, rồi từ đó đánh giá điểm cho chatbot.

CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN

Chương này trình bày phương pháp để xây dựng tập dữ liệu từ tập dữ liệu thô ban đầu, cách áp dụng mô hình Transformer để xây dựng một chatbot ở tiếng Việt.

3.1. Tổng quan các bước thực hiện

Các bước thực hiện được mô tả như Hình 30:



Hình 30 Các bước thực hiện

- Thu thập dữ liệu: tìm kiếm, xây dựng tập dữ liệu dùng cho huấn luyện.
- Tiền xử lý dữ liệu: từ tập dữ liệu đã thu được ở bước trên, tiến hành loại bỏ các ký tự đặc biệt có trong câu (!@#%&*,..), xóa bỏ khoảng trắng, loại bỏ các từ vô nghĩa (hmm, ag,..), các câu bị trùng lặp,… Sau đó lưu lại kết quả vào file text.
- Phân tách tập dữ liệu: loại bỏ những câu có độ dài vượt quá quy định, tạo bộ từ điển từ tập dữ liệu thu được ở bước trước.
- Huấn luyện mô hình: đọc các tập dữ liệu đã thu được ở bước phân tách tập dữ liệu, rồi tiến hành đưa vào mô hình Transformer để huấn luyện mô hình.
- Sinh câu trả lời: nhập dữ liệu đầu vào, tiến hành xử lý dữ liệu đó tương tự như bước tiền xử lý nhưng không phải lưu vào file text. Sau đó đưa vào mô hình để dự đoán câu trả lời, và in kết quả ra màn hình.

3.2. Tiến hành xây dựng mô hình

3.2.1. Thu thập dữ liệu

Yêu cầu quan trọng nhất khi xây dựng chatbot chính là cần có dữ liệu, vì khi có dữ liệu thì mới có cơ sở để mô hình chatbot hoạt động được. Chúng tôi đã xây dựng 2 tập dữ liệu:

- Tập dữ liệu CTUNLPBot là tập dữ liệu do chúng tôi xây dựng với hơn 33.000 bộ câu hỏi và câu trả lời được thu thập từ các nguồn: website Khoa Công nghệ thông tin và Truyền thông[10], Quy định công tác học vụ dành cho sinh viên trình độ đại học hệ chính quy của Hiệu trưởng trường Đại học Cần Thơ (Quyết định số 2093/QĐ-ĐHCT ngày 17/8/2020), chương trình đào tạo khóa 46 của sinh viên các ngành thuộc Khoa Công nghệ Thông tin và Truyền

[10] http://www.cit.ctu.edu.vn/

Thông của trường Đại học Cần Thơ; sổ tay sinh viên ngành MMT và TT[11], sổ tay sinh viên ngành KTPM[12].

- Tập dữ liệu thứ 2 với trên 400.000 cặp câu thoại của các nhân vật trong các bộ phim, được lấy từ website OpenSubtitle[13] 2020 sẽ được sử dụng cho chatbot miền mở. Tập dữ liệu này ở tiếng Anh, chúng tôi sẽ chuyển ngữ sang tiếng Việt, bằng cách sử dụng mô hình Transformer cho bài toán dịch máy từ tiếng Anh sang tiếng Việt của Phạm Bá Cường Quốc [14], với tập dữ liệu sử dụng 600.000 câu được lấy từ TED[14]. Tập dữ liệu thứ 2 được xây dựng ngoài mục đích giúp trợ lý ảo có thể giao tiếp với sinh viên linh hoạt, giúp sinh viên giải tỏa căng thẳng mà còn giúp chúng tôi đánh giá được tính khả thi của chatbot trên miền mở.

Tất cả dữ liệu đều được lưu dưới dạng là file text. Mỗi mô hình chatbot gồm hai file: một file là tập câu hỏi và file còn lại là tập câu trả lời với điều kiện ràng buộc là câu thứ *n* ở file này phải tương ứng với câu thứ *n* ở file kia, và các câu khác cũng phải như vậy, không có trường hợp ngoại lệ.

3.2.2. Tiền xử lý xử dữ liệu

Dữ liệu đã thu được ở dạng thô, chứa nhiều ký tự đặc biệt, cùng với nhiều từ viết tắt hoặc các từ không có ý nghĩa gì, …. Các bước làm sạch dữ liệu được thực hiện như sau:

- Chuyển hết tất cả các câu về dạng chữ thường không in hoa.
- Loại bỏ hết các ký tự đặc biệt có trong câu (@#$%^,..). Ví dụ: “hôm này sao rồi @@” thành “hôm nay sao rồi”
- Xóa bỏ các dấu bắt đầu câu và dấu phân tách câu (-,!?;,...). Ví dụ: “hôm qua, cậu có nhìn thấy cái mắt kính của tôi không? Enna. Tôi để trên bàn giờ lại không thấy đâu!” thành “hôm qua cậu có nhìn thấy cái mắt kính của tôi không enna tôi để trên bàn giờ lại không thấy đâu”
- Xóa hết các chú thích về âm thanh trong các bộ phim. Ví dụ: “hôm nay trời đẹp quá. enna (tiếng nhạc)” thành “hôm nay trời đẹp quá enna”
- Loại bỏ các câu tiếng Anh có trong các bộ phim.
- Loại bỏ hết các câu của tác giả sub như: “phim được thuyết minh bởi: abc”, “phim được dịch bởi: abc”, …
- Loại bỏ các câu từ không có nghĩa. Ví dụ: “Hmm”, “ak”, ….

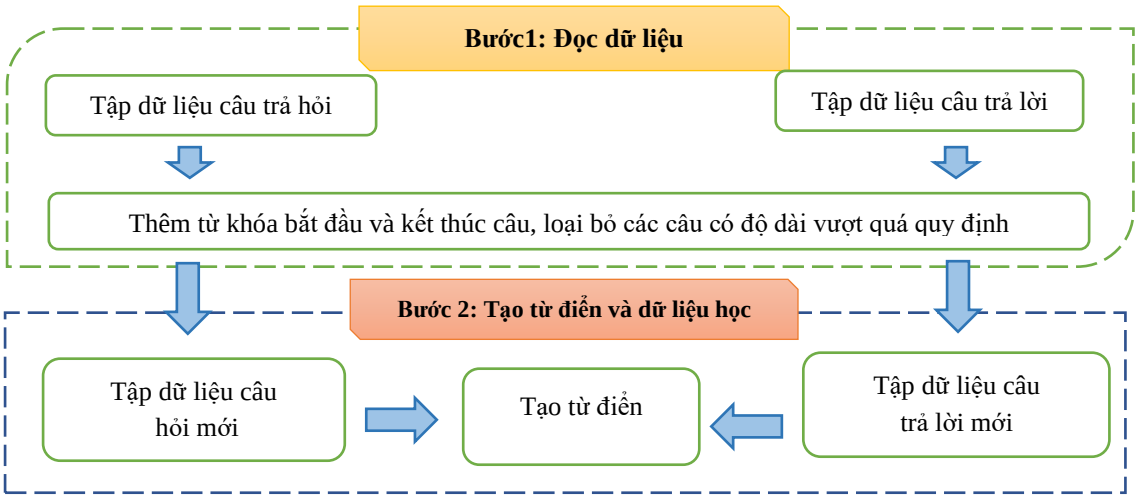
[11] http://www.cit.ctu.edu.vn/index.php/b-n-tin/tin-giao-v-khoa/648-s-tay-sinh-vien-k45-nganh-mmt-va-tt
[12] http://www.cit.ctu.edu.vn/index.php/b-n-tin/tin-giao-v-khoa/669-s-tay-sinh-vien-nganh-k-thu-t-ph-n-m-m
[13] https://www.opensubtitles.org/vi
[14] https://www.ted.com/

- Chuyển các từ viết tắt về câu chuẩn ở tập câu trả lời. Ví dụ "cntt" thành "công nghệ thông tin", "ktpm" thành kỹ thuật phần mềm", "ts" thành tiến sĩ", "ths" thành "thạc sĩ", …

Sau khi thực hiện bước này xong ta sẽ thu được các tập dữ liệu đã được xử lý hoàn toàn phù hợp với mô hình huấn luyện.

3.2.3. Phân tách dữ liệu

Quá trình phân tách tập dữ liệu gồm hai bước như hình 31:



Hình 31 Quy trình phân tách dữ liệu

Từ tập hai tập dữ liệu câu hỏi và câu trả lời tiến hành thêm các từ khóa bắt đầu (<s>) và kết thúc câu (</s>), sau quá trình thu thập dữ liệu, chúng tôi thống kê theo độ dài các câu có trong tập dữ liệu như Bảng 1.

	CTUNLPBot		**OpenSubtitle 2020**	
	Câu hỏi	Câu trả lời	Câu hỏi	Câu trả lời
Tổng số câu	33.906	33.906	419.712	419.712
Tổng số câu có độ dài dưới 20 từ	33.227	33.847	416.227	416.225
Tổng số câu có độ dài trong khoảng 20 đến 25 từ	679	29	3.405	3.457
Tổng số câu có độ dài trên 25 từ	0	30	30	30
Độ dài câu dài nhất	24	42	44	53
Độ dài câu ngắn nhất	4	4	4	3
Độ dài trung bình của một câu	12.54	8.1	9.64	9.59

Bảng 1 Bảng thống kế số lượng từ trong tập dữ liệu

Với tập dữ liệu CTUNLPBot, chúng tôi sẽ thực hiện xây dựng chatbot miền đóng trên bộ dữ liệu này, với độ dài (hay số từ tối đa) một câu ở bộ câu hỏi và câu trả lời sẽ là 25 từ trên một câu, vì với độ dài là 25 từ cho một câu chúng tôi sẽ có thể sử

dụng gần như tối đa dữ liệu mà chúng tôi thu thập được. Với tập dữ liệu OpenSubtitle 2020, chúng tôi sẽ thực nghiệm xây dựng chatbot miền mở với độ dài cho một câu ở bộ câu hỏi và câu trả lời sẽ là 20 từ cho một câu.

Loại bỏ hết các câu vượt quá độ dài đã thống nhất, kế tiếp tiến hành tạo bộ từ điển, bộ từ điển được tạo ra bằng cách sủ dụng hàm *split()* để tách từ theo khoảng trắng, cuối cùng là lưu lại bộ từ điển, bộ từ điển này có dạng là một từ và index của từ, cuối cùng là lưu bộ từ điển lại để sử dụng cho quá trình huấn luyện.

3.2.4. Huấn luyện mô hình

Bước này sẽ quyết định mô hình hoạt động có tốt hay không, ở bước này sẽ thực hiện các giai đoạn được mô tả như Hình 32.

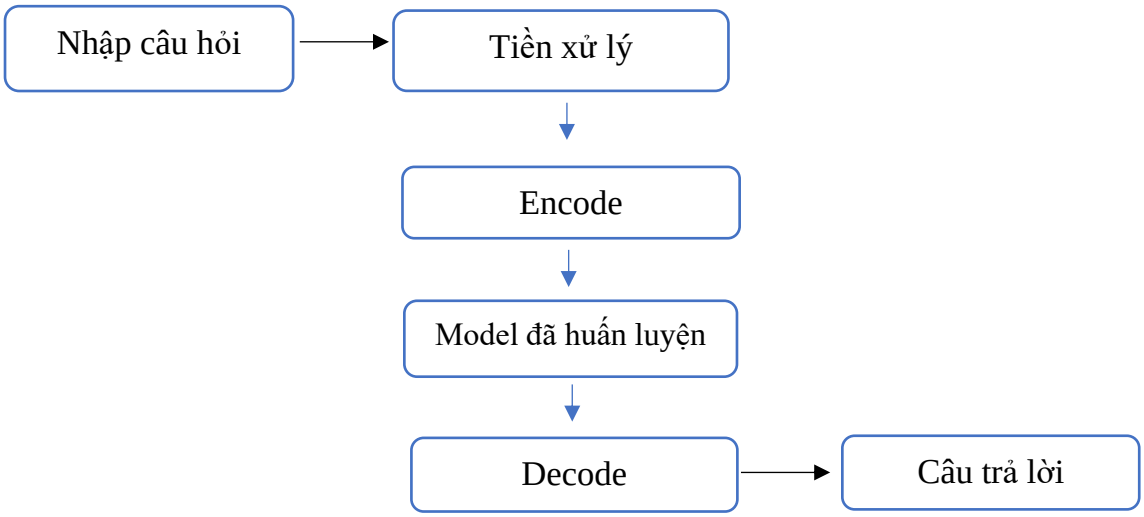
Ở bước 1: mô hình sử dụng lại tập dữ liệu, bộ từ điển đã thu được ở bước phân tách làm đầu vào cho mô hình huấn luyện.

Ở bước 2: quá trình đầu tiên sẽ đưa các câu đầu vào về dạng vector, ví dụ như câu "xin chào" khi đưa về vector có dạng [23, 13], quá trình này đơn giản là từ bộ từ điển tiến hành đối chiếu tìm ra vị trí từng từ trong câu đầu vào, rồi chuyển đổ thành số tương ứng với vị trí của từ đó trong bộ từ điển, ở ví dụ trên từ "xin" có số thứ tự là 23, còn từ "chào" có số thứ tự là 13, nên khi chuyển về vector có dạng [23,13], quá trình này cũng chính là giải đoạn Embedding cho câu đầu vào. Kế tiếp, câu đầu vào sao khi đã vector hóa sẽ tiến hành padding đưa câu đâu vào về cùng một độ dài, ví dụ cần đưa câu về cùng độ độ dài là 5 chẳng hạn, thì vector [23, 13] sẽ được chuyển đổ thành [23, 13, 0, 0, 0], quá trình này sẽ giúp các tính toán nhanh chống và đơn giản hơn, thay vì dữ liệu được đưa vào một cách không được đồng bộ. Ngoài ra, ở bước này cũng sẽ định nghĩa các thông số cho mô hình huấn luyện như: số head, số layer và kích thước D_{model}. Kế tiếp tiến hành huấn luyện mô hình dữ liệu sẽ đi qua lần lượt từng khối trong Encoder và Decoder, sau khi thực hiện các tính toán thì với mỗi một cặp câu hỏi và câu trả lời đầu vào của mô hình, sẽ thu được một ma trận *Z*, thực hiện tương tự hết tập dữ liệu, kết nối tất cả các ma trận *Z* thu được sẽ chính là model đã huấn luyện, quá trình huấn luyện này sẽ được thực hiện với một số lần quy định. Sau cùng thì tiến hành lưu model lại để thuận tiện hơn cho việc sử dụng sau này.

Hình 32 Các bước huấn luyện mô hình

3.2.5. Sinh câu trả lời

Dựa trên kết quả đã được huấn luyện. Có thể tiến hành dự đoán câu trả lời từ câu hỏi nhập vào. Các bước được diễn ra như sau (Hình 33):



Hình 33 Mô hình sinh câu trả lời

Câu đầu vào sẽ được tiền xử lý loại bỏ đi hết các yếu tố không cần thiết, rồi tiến hành vector hóa dựa vào bộ từ điển đã được xây dựng, kế tiếp sẽ padding cho bằng với độ dài câu đã quy định, quá trình này chính là giai đoạn Encoder bên trên. Từ kết quả Encoder có được tiến hành đưa vào model đã huấn luyện. Kết quả đầu ra sẽ là một ma trận *Z'*, sau đó mỗi dòng trong *Z'* sẽ được đưa qua lớp Linear chiếu lên một không gian vector logits như đã nói ở phần trước, không gian vector này có số ô bằng với số từ của bộ từ điển – mỗi ô tương ứng với điểm của một từ duy nhất, sau cùng lớp *softmax* sẽ chuyển vector đó thành xác suất và đưa ra ô có giá trị lớn nhất, vị trí ô có xác suất lớn nhất này sẽ đối chiếu lên bộ từ điểm đưa ra từ tương ứng với vị trí đó, cũng chính là từ được dự đoán. Quá trình thực hiện đến khi hết tất cả các dòng trong *Z'*. Kết quả cuối cùng chính là câu trả lời của chatbot.

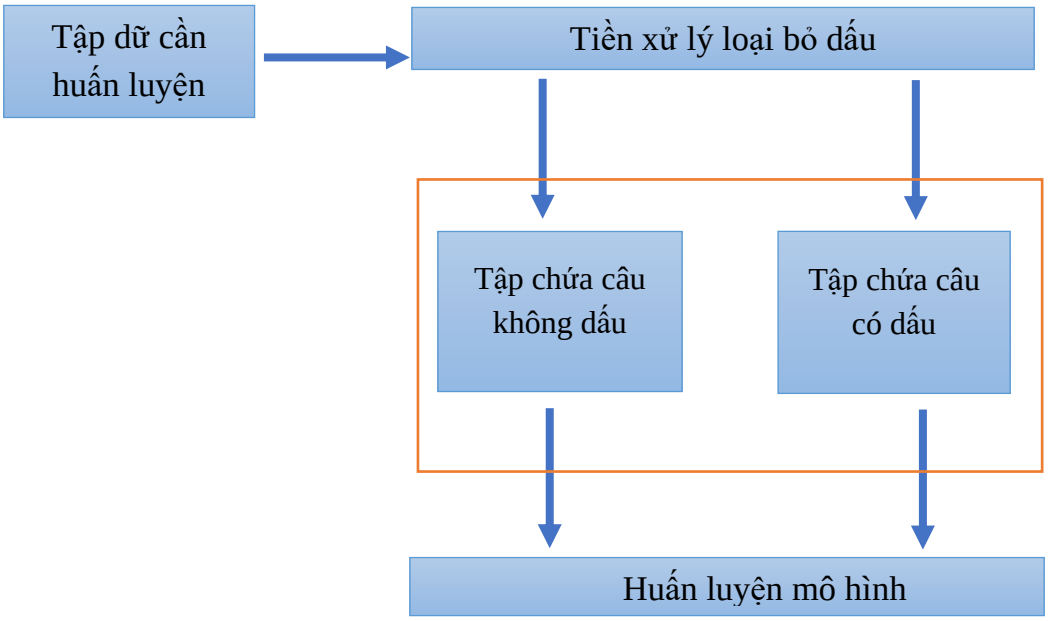
3.2.6. Cải thiện chatbot

Sau khi huấn luyện mô hình chatbot. Kết quả cho thấy mô hình hoạt động tốt với những câu đầu vào đúng chính tả và ngữ pháp. Tuy nhiên chatbot vẫn chưa cho kết quả tốt với các câu đầu vào chưa có dấu và các câu sai chính tả (có thể do người dùng nhập sai chính tả), nên cần cải thiện thêm bằng cách xây dựng thêm một mô hình để xử lý các câu đầu vào chưa đúng.

3.2.6.1. Chat không dấu

Đôi lúc khi để việc tìm kiếm thông tin được nhanh hơn, chúng ta vẫn thường có thói quen gõ một chuỗi câu không có dấu, để tìm kiếm thông tin mà chúng ta cần, hoặc do một cách vô ý nào đó chúng ta cũng gõ những câu không có dấu. Do đó, chúng tôi xây dựng thêm một mô hình chuyển câu không dấu thành có dấu. Chúng tôi nhận thấy mô hình chatbot được xây dựng dựa trên mô hình Transformer với các tác vụ cho phép xử lý ngôn ngữ tự nhiên, nên việc sử dụng mô hình Transformer để hỗ trợ việc chuyển câu không dấu thành có dấu, hỗ trợ cho việc chat được tốt hơn là

hoàn toàn khả thi. Các bước huấn luyện Transformer cho mô hình chuyển câu không dấu thành có dấu được thực hiện như Hình 33:



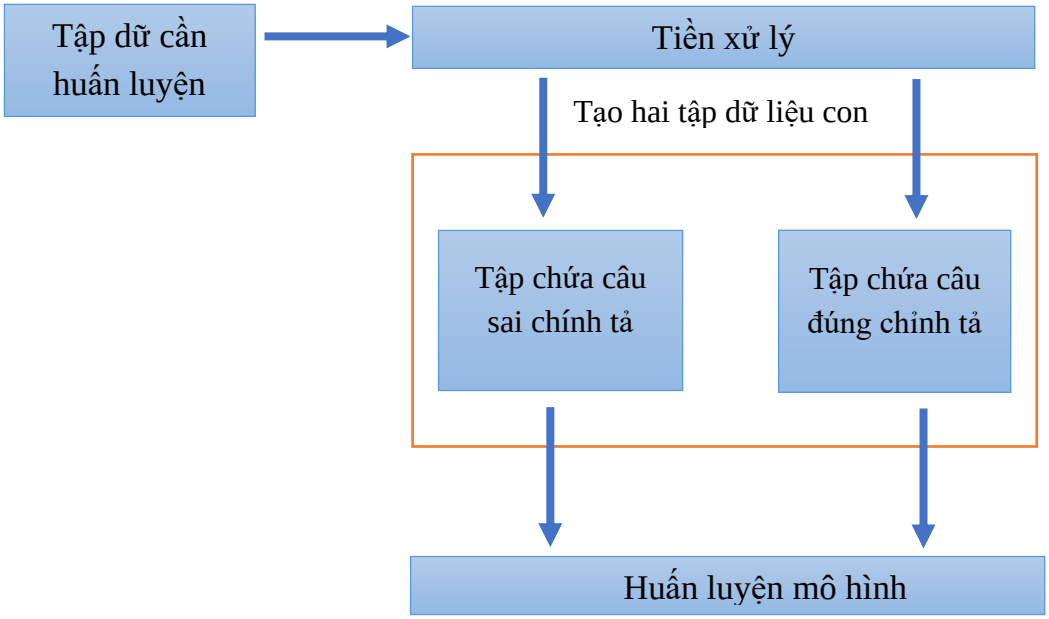
Hình 34 Mô hình chuyển câu không dấu thành có dấu

Tập dữ được sử dụng gồm có hai tập, một tập là tập câu hỏi của miền mở, tập còn lại là tập câu hỏi của miền đóng. Ở bước tiền xử lý, đầu tiên chúng tôi tạo ra thêm tập dữ liệu mới giống với tập câu hỏi của từng miền đã có, sau đó tiến hành loại bỏ dấu câu cho mỗi dòng ở tập dữ liệu vừa được tạo ra bằng cách sử dụng thư viện Unicode[15], cuối cùng là lưu kết quả loại bỏ dấu câu dưới dạng file text. Quá trình huấn luyện và sinh ra câu có dấu cũng tương tự như quá trình huấn luyện chatbot và sinh ra câu trả lời của chatbot.

3.2.6.2. Câu hỏi sai chính tả

Trong quá trình nhập câu hỏi, người dùng gõ sai lỗi chính tả là hoàn toàn có khả năng và điều này ảnh hưởng đến chất lượng của hệ thống chatbot. Chúng tôi giải quyết vấn đề này bằng cách tạo ra các từ nhiễu sau đó đưa vào mô hình huấn luyện, với khả năng xử lý ngôn ngữ tự nhiên nên chúng tôi tiếp tục sử dụng mô hình Transformer và thuật toán kNN để so sánh kết quả. Quy trình huấn luyện mô hình **c**huyểm từ sai chính tả về đúng chính tả sử dụng Transformer và thuật toán kNN được trình bài ở Hình 35.

[15] https://pypi.org/project/Unidecode/

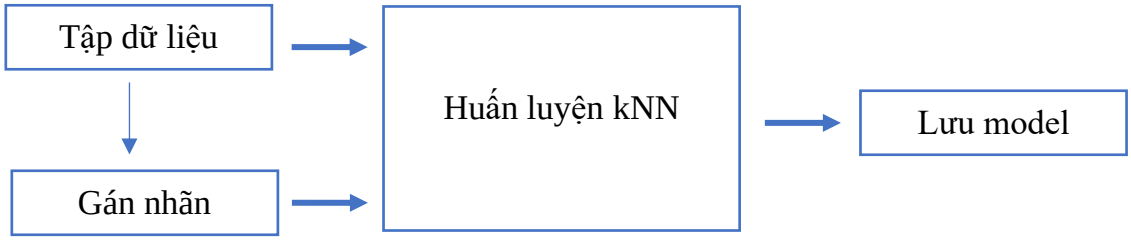


Hình 35 Quy trình huấn luyện mô hình chuyển từ sai chính tả về đúng chính tả

Chúng tôi sử dụng hai tập dữ liệu, tập thứ nhất là tập câu hỏi được lấy từ tập tập dữ liệu câu hỏi miền đóng, tập thứ hai gồm 200.000 câu hỏi được lấy từ tập dữ liệu câu hỏi của miền mở. Từ tập dữ liệu thứ nhất, tiến hành thực hiện tiền xử lý tương tự, sau đó mỗi câu trong tập dữ liệu sẽ được dùng để tạo ra một số câu có lỗi chính tả ở trong đó, sau đó phân tách ra thành hai tập con, một tập sẽ chỉ chưa toàn những câu sai lỗi chính tả, tập còn lại sẽ chứa câu đúng chính tả. Với tập dữ liệu thứ hai cũng sẽ thực hiện tương tự. Cuối cùng là đưa từng tập dữ liệu con tương ứng vào huấn luyện tương tự như huấn luyện chatbot và lưu model đã huấn luyện. Quá trình chuyển đổi câu từ có lỗi chính tả sang câu không có lỗi chính tả được thực hiện tương tự như quá trình sinh câu trả lời của chatbot. Với giải thuật kNN, các điểm *k* lần lượt là là các số lẻ trong khoảng từ 1 đến 17.

3.2.6.3. Chuyển đổi giữa hai miền của chatbot

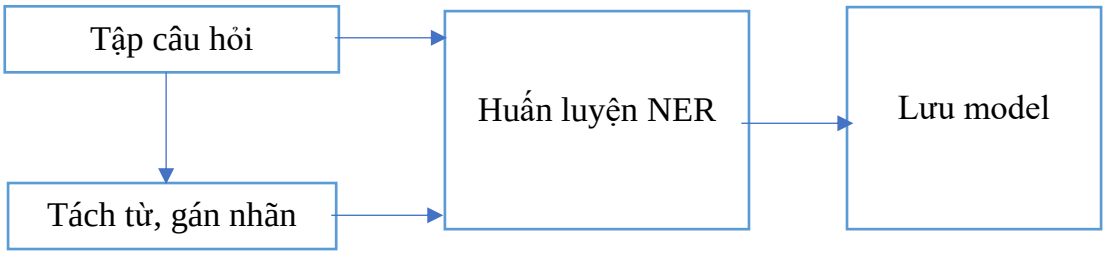
Giúp tăng khả năng phản hồi của chatbot hơn nữa, luận văn này còn xây dựng thêm cơ chế chuyển đổi giữa hai miền để chatbot có thể đáp ứng được nhiều phản hồi hơn cho người dùng. Chúng tôi đã sử dụng giải thuật kNN cho phân loại câu hỏi đưa vào mô hình, tác vụ phân loại này giúp chuyển đổi linh hoạt giữa hai miền đóng và mở ở chatbot. Mô hình này được mô tả như Hình 36. Với tập dữ liệu sẽ bao gồm hai tập câu hỏi của cả hai miền đóng và mở, chúng tôi sử dụng toàn bộ dữ liệu câu hỏi có ở cả hai miền để làm tập dữ liệu huấn luyện, sau đó tiến hành gán nhãn thủ công với 1 sẽ là câu thuộc miền đóng và 0 sẽ là các câu thuộc miền mở.



Hình 36 Mô hình sử dụng kNN để phân loại câu hỏi

3.2.6.4. Nhận diện thực thể trong câu

Bên cạnh việc chuyển câu không dấu, câu sai chính tả, chúng tôi tiến hành nhận diện thực thể hay dự đoán tên thực tể trong câu hỏi đầu vào. Chúng tôi sử dụng kết quả NER (Named Entity Recognition) của Nguyễn Chiên Thắng [15], với tập dữ liệu được sử dụng là tập câu hỏi của CTUNLPBot để huấn luyện mô hình (Hình 37).



Hình 37 Mô hình huấn luyện NER

Chúng tôi thực hiện tách từ với hàm *split*, tách các từ theo kiểu "*word by word*", sau đó tiền hành gán nhãn thủ công cho các từ sau khi đã tách, ví dụ xét một câu "mã học phần ct101 là gì" thì sẽ được gán nhãn [B-key, B-key, B-Key, B-mhp, O, O], trong đó "B-key, B-mhp" là các từ cần chú ý đến, còn "O" sẽ là các từ không cần chú ý đến để nhận diện thực thể. Trong NER có sử dụng thêm CRF (Conditional Random Field) để dự đoán tên cho thực thể, giai đoạn huấn luyện như sau: Câu câu đầu vào sẽ được mã hóa về dạng vector, nhãn của từng từ cũng sẽ được mã hóa về vector, bước kế tiếp là đưa cả câu và nhãn đã được vector hóa vào model và tiến hành huấn luyện với một số lần nhất định. cuối cùng là lưu model đã huấn luyện.

Khi nhập một câu, câu này sẽ được chuyển về dạng vector sau đó đưa vào model đã huấn luyện, sẽ cho ra một vector chứa thông tin hay chính là nhãn của từng từ có trong câu, chúng tôi chỉ quan tâm đến những từ có nhãn, các từ không nhãn sẽ bị bỏ qua, cuối cùng là sẽ thu được một câu chỉ chứa những từ có nhãn hay chính là câu rút trích chứa thông tin liên quan miền đóng.

CHƯƠNG 4. THỰC NGHIỆM

Chương này trình bày các đánh giá về độ chính xác của trợ lý ảo, nhận xét kết quả thực nghiệm của hai mô hình chatbot miền đóng, miền mở.

4.1. Kết quả thực nghiệm

Việc xây dựng mô hình được thực hiện trên môi trường Google Colab, với RAM là 12GB cùng với TPU do Google Colab cung cấp.

Chúng tôi thực nghiệm mô hình chatbot với từng giai đoạn đã được nói đến ở trên, với tập dữ liệu hơn 33.000 cặp câu hỏi và câu trả lời được xây dựng về khoa Công nghệ Thông tin và Truyền thông cho miền đóng, và hơn 400.000 cặp câu thoại được thu thập ở OpenSubTitle 2020 cho miền mở.

Thay vì chỉ sử dụng một độ dài (với số từ tối đa) trên một câu trong cặp câu hỏi và câu trả lời như quy định dựa vào kết quả ở Bảng 1, chúng tôi còn thực nghiệm thêm một độ dài khác để dùng làm kết quả đánh giá cho chatbot. Với miền đóng, bên cạnh xây dựng chatbot hỗ trợ độ dài là 25 từ cho một câu trong cặp câu hỏi và câu trả lời, chúng tôi thực nghiệm thêm với độ dài là 30 từ cho một câu trong cặp câu hỏi và câu trả lời. Tương tự với miền mở, chúng tôi thực nghiệm thêm với độ dài câu là 25 từ. Các tham số huấn luyện cho chatbot với miền đóng và miền mở được trình bày ở Bảng 2 và Bảng 3.

Bảng tham số huấn luyện chatbot với miền đóng		
Số từ tối đa cho một câu	25 từ	30 từ
Số epoch	200	200
Dữ liệu học thực tế	33.876 cặp câu	33.876 cặp câu
Số từ bộ từ điển	2.742 từ	2.742 từ
Thời gian huấn luyện	8 phút 6 giây	10 phút 17 giây

Bảng 2 Tham số huấn luyện chatbot với miền đóng

Bảng tham số huấn luyện chatbot với miền mở		
Số từ tối đa cho một câu	20 từ	25 từ
Số epoch	200	200
Dữ liệu học thực tế	412.875 cặp câu	412.875 cặp câu
Số từ bộ từ điển	13.126 từ	13.126 từ
Thời gian huấn luyện	2 giờ 35 phút	2 giờ 45 phút

Bảng 3 Tham số huấn luyện chatbot với miền mở

Và dưới đây là các kết quả thực nghiệm thu được với chatbot miền đóng, trong đó "YOU" là người dùng nhập và "CITBOT" trợ lý ảo.

```
YOU: chào bạn
CITBOT: Hi, xin chào bạn

YOU: bạn tên gì
CITBOT: Tên của tôi là CitBot

YOU: bạn có thể giúp gì được cho tôi
CITBOT: Tôi có thể trả lời các câu hỏi về Khoa, chương trình học cho bạn

YOU: trưởng khoa là ai
CITBOT: Trưởng khoa thầy Nguyễn Hữu Hòa

YOU: mã học phần ltcb a
CITBOT: CT101

YOU: học phần tiên quyết ct101
CITBOT: Không có HPTQ

YOU: bot có biết cô Khang không
CITBOT: Mình biết nhe, là cô Lâm Nhựt Khang ở bộ môn CNTT

YOU: cô Khang có khó không
CITBOT: Theo mình là cô không khó nhe

YOU: số tín chỉ tốt nghiệp khóa 42
CITBOT: Khóa 42 cần 155 tín chỉ để tốt nghiệp

YOU: điều kiện lvtn cntt
CITBOT: Số tín chỉ tích lũy phải ≥ 120 Tín chỉ

YOU: học kỳ 2 2020 2021 ct178 04 được ai giảng dạy
CITBOT: Người giảng dạy là GV Nguyễn Hoàng Việt

YOU: lịch thi học kỳ 2 2020 2021 ct182 nhóm 03
CITBOT: Lịch thi là 12/05/2021

YOU: ngày thi và phòng thi ct054h nhóm m01 học kỳ 2 2020 2021
CITBOT: Lịch thi là 25/05/2021, phòng thi là P12

YOU: hình thức thi nhập môn công nghệ phần mềm m01 học kỳ 2 2020 2021
CITBOT: Hình thức thi là TN+TL
```

Hình 38 Ví dụ hội thoại trên miền đóng với độ dài câu thoại tối đa là 25 từ

```
YOU: chào bot
CITBOT: Hello, chào bạn

YOU: cho tôi biết thông tin về bạn
CITBOT: Tôi là chatbot CitBot, được tạo ra bởi Hữu Lộc

YOU: người ta phát minh ra bạn để làm gì
CITBOT: Tôi có thể trả lời các câu hỏi về Khoa, CT học cho bạn

YOU: tốt nghiệp
CITBOT: Tích lũy đủ số TC quy định , ĐTBCTL từ 2.0 trở lên theo thang điểm 4

YOU: lịch thi học kỳ 2 2020 2021 tt. thcb nhóm chiều 5 phòng 14
CITBOT: Lịch thi là 01/06/2021

YOU: lịch thi học kỳ 2 2020 2021 thcb nhóm a08
CITBOT: Thi chung nhóm TH

YOU: học kỳ 2 2020 2021 thcb nhóm 20 được ai giảng dạy
CITBOT: Người giảng dạy là GV Nguyễn Đức Khoa

YOU: cô diễm dạy môn gì
CITBOT: Ý của bạn là cô Lê Thị Diễm hay là cô Phạm Thị Ngọc Diễm vậy ạ

YOU: Cô Lê Thị Diễm dạy môn gì
CITBOT: cô Lê Thị Diễm dạy môn TT. THCB, LTCB A
```

Hình 39 Ví dụ hội thoại trên miền đóng với độ dài câu thoại tối đa là 30 từ

Ở kết quả thực nghiệm chatbot vẫn có những câu hỏi chưa hoàn toàn nhận được câu trả lời chính xác như ý định của người hỏi, tuy nhiên tỷ lệ câu trả lời sai là rất thấp. Kết quả chatbot miền mở, với "BOT" là người dùng nhập và "KEY" là hệ thống trả lời tự động.

```
============================================
================ Bắt đầu chat ==============
============== Nhấn q để thoát =============
============================================

YOU: chào buổi sáng
KEY: chào buổi sáng sarah

YOU: mình trao đổi nhé
KEY: cậu muốn mua cho mình cái này

YOU: cậu đi mua chút gì để anh em mình nhâm nhi đi
KEY: tớ sẽ đi tắm rửa

YOU: cậu đi không
KEY: đi đi đi đi

YOU: hôm nay là chủ nhật tớ không chạy bộ vào chủ nhật
KEY: cậu đã làm gì

YOU: hôm qua chúng ta uống nhiều rượu quá đấy
KEY: tớ không nghĩ là cậu ta sẽ uống rượu

YOU: cái nào cậu thích bạn hiền cái nào cũng được
KEY: yeah đó là một câu chuyện thú vị

YOU: buổi sáng tôi tập thái cực quyền
KEY: tôi đã làm việc ở đây
```

Hình 40 Ví dụ hội thoại trên miền mở với độ dài câu thoại tối đa là 20 từ

```
================================================
================== Bắt đầu chat ================
================ Nhấn q để thoát ===============
================================================

YOU: chào buổi sáng
KEY: chào buổi sáng thưa ông

YOU: cậu có muốn đi chơi không nào
KEY: tôi có thể làm được

YOU: chúng ta cùng đi ăn sáng với nhau chứ
KEY: tôi không biết anh đang ở đâu

YOU: cậu là một phóng viên à
KEY: tôi là bác sĩ tâm lý

YOU: anh nghĩ nặng bao nhiêu
KEY: anh không thể ngăn được nó

YOU: cô trẻ quá
KEY: tôi không biết nữa

YOU: tôi cần một lọ hóa chất diệt cỏ
KEY: có phải đó là chỗ của mình

YOU: cậu đang nói chuyện với ai thế
KEY: tôi không biết gì cả

YOU: lee kang uk đâu rồi
KEY: anh ấy đang ở trên sân khấu
```

Hình 41 Ví dụ hội thoại trên miền mở với độ dài câu thoại tối đa là 25 từ

Với kết quả huấn luyện trên, tỷ lệ câu trả lời chính xác từ chatbot còn khá là thấp. Nhưng có thể thấy được chatbot đã có đưa ra được các câu trả lời khác nhau cho từng câu hỏi, chất lượng của câu trả lời sẽ phụ thuộc rất nhiều vào tập dữ liệu học. Dưới đây là ví dụ minh họa khi nhập câu có chứa các thực thể sai với chatbot miền đóng Hình 42 và miền mở Hình 43.

```
YOU: chào bạn
CITBOT: Hi, xin chào bạn

YOU: banj tên gì
CITBOT: Tên của tôi là CitBot

YOU: ban co the giup gi cho toi
CITBOT: Tôi có thể trả lời các câu hỏi về Khoa, chương trình học cho bạn

YOU: trưởng khoa là ai
CITBOT: Trưởng khoa thầy Nguyễn Hữu Hòa

YOU: phó trưởng khoa
CITBOT: Phó TK thầy Ngô Bá Hùng, thầy Huỳnh Xuân Hiệp và thầy Phạm Nguyên Khang

YOU: bot biết thầy Phi không
CITBOT: Mình biết nhe là Thầy Phạm Thế Phi ở bộ môn CNTT

YOU: bot biết cô Khang khoong
CITBOT: Mình biết nhe, là cô Lâm Nhựt Khang ở bộ môn CNTT

YOU: cô Khang có khos không bot
CITBOT: Theo mình là cô không khó nhe

YOU: mas hoc phan vi tich phan a1
CITBOT: Là học phần bắt buộc

YOU: ma hoc phan ltcb a
CITBOT: CT101
```

Hình 42 Ví dụ thực nghiệm chatbot miền đóng có nhập câu có thực thể sai

```
YOU: chào buổi sangs
CITBOT: chào buổi sáng thưa ông

YOU: đi du lịch với tooi không
CITBOT: tôi sẽ đi cùng cô

YOU: thời tiết hôm nay đẹp chưs
CITBOT: tôi đã nói là tôi đã có thể giúp cô

YOU: chung ta di choi di
CITBOT: anh không thể làm được

YOU: cậu tên j
CITBOT: tôi là người duy nhất có thể tin tưởng

YOU: nhà cậu ở đâu thees
CITBOT: tôi sẽ gọi lại sau

YOU: tôi qua nhà cậu ăn tooiss nhes
CITBOT: tôi không biết tôi đang làm gì
```

Hình 43 Ví dụ thực nghiệm chatbot miền mở khi nhập câu có thực thể sai

Bên cạnh đó còn sử dụng kNN cho phân loại câu hỏi, kết quả thực nghiệm khi sử dụng kNN vào phân loại câu hỏi cho chatbot (Hình 44).

```
============================================
================ Bắt đầu chat ==============
================ Nhấn q để thoát ===========
============================================

YOU: chào bot
BOT Open: chào anh bạn nhỏ của tôi à

YOU: anh văn chuyên ngành 1
BOT CNTT: Mã học phần XH031 số tín chỉ 4

YOU: cậu có muốn đi chơi không
BOT Open: tôi sẽ đi chơi với anh

YOU: bộ môn cong nghệ thông tin
BOT CNTT: Bộ môn Công nghệ thông tin thuộc khoa Công nghệ thông tin và Truyền thông, thành lập ngày 16/09/2013

YOU: đi chơi khoong
BOT Open: chơi một trò chơi đi

YOU: đi du lịch với tôi không
BOT Open: tôi không muốn đi đâu
```

Hình 44 Ví dụ thực nghiệm chatbot sử dụng kNN phân loại câu hỏi

Ngoài việc chỉ xây dựng các vấn đề liên quan về khoa Công nghệ Thông tin và Truyền thông trường Đại học Cần Thơ, cũng như là nhằm nâng cao khả năng cho chatbot, chúng tôi có đưa thêm vào tập dữ liệu các dữ liệu quan đến lịch thi từng học phần trong học kỳ 2 năm học 2020-2021, vì đây là tập dữ liệu nội bộ nên sẽ không đi thêm vào chi tiết. Dưới đây là kết quả khi chúng tôi thực nghiệm chatbot (Hình 45).

```
YOU: chào bạn
CITBOT: Hi, xin chào bạn
YOU: ngày thi và phòng thi lập trình java nhóm 01 học kỳ 2 2020 2021
CITBOT: Lịch thi là 26/04/2021, phòng thi là P11/DI
YOU: ai giảng dạy toán rời rạc 08 học kỳ 2 2020 2021
CITBOT: Người giảng dạy là GV Võ Trí Thức
YOU: cô Khang dạy môn gì
CITBOT: cô Khang dạy môn NL HĐH, MTT, Anh văn chuyên môn công nghệ thông tin 2
YOU: lịch thi học kỳ 2 2020 2021 nl khoa học máy tính nhóm 07
CITBOT: Lịch thi là 01/06/2021
YOU: phòng thi tt. thcb nhóm chiều 2 phòng 05 học kỳ 2 2020 2021
CITBOT: Phòng thi là P13
YOU: email của cô Khang
CITBOT: Email là lnkhang@cit.ctu.edu.vn
YOU: cảm ơn bot
CITBOT: Hi, không có chi
```

Hình 45 Ví dụ thực nghiệm chatbot miền đóng trên dữ liệu nội bộ

4.2. Đánh giá độ chính xác

Việc đánh giá độ chính xác cho một chatbot thường không đơn giản. Trước tiên, chúng tôi xây dựng tập test cho mỗi miền. Mỗi tập test gồm 100 cặp câu hỏi – trả lời do sinh viên Khoa Công nghệ Thông tin và Truyền thông xây dựng thủ công. Kế tiếp việc đánh giá chatbot được thực hiện theo 2 cách sau:

- Sử dụng thang điểm BLEU để đánh giá độ chính xác của chatbot trên tập test. Các trọng số trên được tham khảo từ bài blog "A Gentle Introduction to Calculating the BLEU Score for Text in Python" của tác giả Jason Brownlee[16] như sau: BLEU-1 với weight=(1; 0; 0; 0); BLEU-2 với weight=(0,50; 0,50; 0; 0); BLEU-3 với weight=(0,33; 0,33; 0,33; 0) và BLEU-4 với weight=(0,25; 0,25; 0,25; 0,25). Kết quả đánh giá được trình bày ở Bảng 4.
- Đánh giá dựa trên số lượng câu trả lời đúng trên tổng số câu hỏi được nhập vào. Cách đánh giá này chỉ áp dụng cho miền đóng, vì chatbot miền mở có câu trả lời là tự do không có một quy chuẩn cụ thể, nên sẽ không áp dụng cách đánh giá này. Kết quả đánh giá được trình bày ở Bảng 5, với tỉ lệ số câu trả lời đúng được tính theo phần trăm (%).

Đánh giá BLEU	**Số từ tối đa cho mỗi câu trong cặp câu hỏi và câu trả lời**			
	Miền đóng		**Miền mở**	
	25 từ	**30 từ**	**20 từ**	**25 từ**
BLEU-1	0,292	0,297	0,18	0,18
BLEU-2	0,251	0,256	0,13	0,12
BLEU-3	0,223	0,226	0,10	0,10
BLEU-4	0,198	0,200	0,08	0,08

Bảng 4 Kết quả đánh giá BLEU cho chatbot

Sau khi thực nghiệm mô hình chatbot miền đóng, kết quả thực nghiệm khi đặt câu hỏi cho chatbot có tỷ lệ chính xác cao hơn miền mở một phần do quản lý được bộ dữ liệu để đưa vào huấn luyện chatbot. Quan sát thực nghiệm cho thấy chatbot có thể sinh ra các câu trả lời với mức độ phù hợp ngữ cảnh, kết quả câu trả lời phụ thuộc vào tập dữ liệu ban đầu được xây dựng để huấn luyện.

So với kết quả sử dụng mô hình chatbot miền mở sử dụng Seq2Seq và LSTM ở luận văn của Nguyễn Văn Vĩ [9], có điểm số BLEU 0,07, thì mô hình chatbot miền mở sử dụng Transformer cho kết quả là 0,07, với kết quả mặc dù cả hai đều cho cùng điểm số BLEU, nhưng với mô hình của chúng tôi thời gian huấn luyện mô hình chưa đến 3 giờ cũng đã đạt được kết quả tương tự, so với thời gian huấn luyện lên đến 72 giờ ở luận văn của Nguyễn Văn Vĩ, cho thấy được tốc độ học của mô hình Transformer là rất vượt trội.

[16] https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

Kết quả đánh giá theo số lượng câu trả lời đúng		
Số từ tối đa một câu	**25 từ**	**30 từ**
Tỉ lệ số câu trả lời đúng	49%	50%

Bảng 5 Kết quả đánh giá theo số lượng câu trả lời đúng

Từ kết quả trên, khi so sánh với mô hình chatbot trên miền đóng sử dụng framework RASA kết hợp SVM và CRF để trích xuất ở luận văn của Lê Nhật Nam [7] với độ chính xác đạt được là 92,7% (tỉ lệ số câu trả lời đúng trên tổng số câu nhập vào), chatbot chúng tôi chỉ đạt được tỉ lệ là 49% và 50%, cả hai đều thấp hơn so với với framework RASA. Tỉ lệ câu trả lời đúng của chatbot khá thấp do chatbot chỉ đưa ra câu trả lời không có chủ ngữ vị ngữ, mà chỉ đưa ra câu trả lời rút gọn từ ý chính của câu hỏi, dẫn đến khi so sánh với tập test số câu trả lời hoàn toàn trùng khớp khá thấp. Bên cạnh đó, kết quả cho thấy câu có số từ tối đa là 30 từ đạt kết quả có phần tốt hơn câu có độ dài tối đa là 25 từ. Điều này cho thấy khả năng tương thích của Transformer cho các chuỗi đầu vào tương đối dài là khá tốt.

Để tăng khả năng phản hồi của chatbot, luận văn còn áp dụng mô hình Transformer cho việc chuyển câu không dấu khi người dùng nhập vào và được đánh giá bằng điểm BLEU. Bảng 6 trình bày tham số khi xây dựng mô hình và Bảng 7 là kết quả điểm BLEU cho tác vụ chuyển đổi câu không dấu thành câu có dấu.

Tham số	**Miền đóng**	**Miền mở**
Tập dữ liệu	33.907 cặp câu	416.277 cặp câu
Độ dài tối đa 1 câu	25 từ	20 từ
Số từ trong bộ từ điển	1.510 từ	12.621 từ
Số epoch huấn luyện	200	200
Thời gian huấn luyện	8 phút	2 giờ 29 phút
Tập kiểm tra	100 câu	100 câu

Bảng 6 Tham số sử dụng khi xây dựng mô hình chuyển câu không dấu thành câu có dấu

Điểm số BLEU	Số từ tối đa trên một câu trong cặp câu hỏi và câu trả lời	
	Miền đóng (25 từ)	Miền mở (20 từ)
BLEU-1	0,92	0,98
BLEU-2	0,88	0,98
BLEU-3	0,84	0,96
BLEU-4	0,79	0,94

Bảng 7 Kết quả điểm số BLEU cho tác vụ chuyển câu không dấu thành câu có dấu

Ví dụ minh họa khi chuyển câu không dấu thành câu có dấu cho miền đóng được trình bày ở Hình 46 và miền mở được trình bày ở Hình 47, trong đó "YOU" là chuỗi được được nhập, "KEY" là chuỗi chuyển đổi.

```
YOU: ban ten gi
KEY: bạn tên gì

YOU: truong khoa cong nghe thong tin
KEY: trường khoa công nghệ thông tin

YOU: bo mon ky thuat phan mem
KEY: bộ môn kỹ thuật phần mềm

YOU: lich thi ct101 01 hoc ky 2 nam hoc 2020 2021
KEY: lịch thi ct101 01 học kỳ 2 2020 2021 học kỳ

YOU: ma hoc phan vi tich phan a1
KEY: mã học phần vi tích phân a1
```

Hình 46 Ví dụ minh họa chuyển câu không dấu thành câu có dấu ở miền đóng

```
YOU: chao ban
KEY: chào bạn

YOU: ban ten gi
KEY: bạn tên gì

YOU: nha ban o dau vay
KEY: nhà bạn ở đâu vậy

YOU: di du lich voi toi khong nao
KEY: đi du lịch với tôi không nào

YOU: thoi tiet hom nay dep qua nhi
KEY: thời tiết hôm nay đẹp quá nhỉ

YOU: co cong mai sat co ngay nen kim
KEY: cô công mài sắt cô ngày nên kim

YOU: sao cau khong he noi voi toi cau dinh lam the nay
KEY: sao cậu không hề nói với tôi cậu định làm thế này

YOU: that toi da mong cau co the mac no de du le hom nay
KEY: thật tôi đã mong cậu có thể mặc nó để dự lễ hôm nay

YOU: duoc thoi neu no khong gia tri gi thi toi cung chang nen giu no nua
KEY: được thôi nếu nó không giá trị gì thì tôi cũng chẳng nên giữ nó nữa

YOU: phai toi di tuan khap 6 quan nhung quyet dinh dung l
KEY: phải tôi đi tuần khắp 6 quận nhưng quyết định đúng l
```

Hình 47 Ví dụ minh họa chuyển câu không dấu thành câu có dấu ở miền mở

Bảng 8 trình bày các tham số dùng chung cho việc chuyển câu sai chính tả về đúng chính tả áp dụng cho hai phương pháp sử dụng Transformer và giải thuật kNN.

Tham số	Miền đóng	Miền mở
Tập dữ liệu	339.061 cặp câu	596.367 cặp câu
Độ dài tối đa 1 câu	25 từ	20 từ
Số từ trong bộ từ điển	14.705 từ	22.596 từ
Số epoch huấn luyện	200	200
Thời gian huấn luyện Transformer	2 giờ 7 phút	3 giờ 20 phút
Tập kiểm tra	100 câu	1.000 câu

Bảng 8 Tham số dùng để chuyển câu sai chính tả về đúng chính tả

Bảng 9 trình bày kết quả điểm số BLEU cho tác vụ chuyển câu sai chính tả về đúng chính tả với Transformer.

Điểm số BLEU	Số từ tối đa trên một câu trong cặp câu hỏi và câu trả lời	
	Miền đóng (25 từ)	Miền mở (20 từ)
BLEU1	0,98	0,99
BLEU2	0,98	0,99
BLEU3	0,97	0,98
BLEU4	0,95	0,96

Bảng 9 Kết quả điểm số BLEU cho tác vụ chuyển câu sai chính tả về đúng chính tả

Ví dụ minh họa chuyển đổi câu sai chính tả với Transformer cho miền đóng và miền mở được trình bày lần lượt ở Hình 47 và Hình 48.

```
YOU: chào banj
KEY: chào bạn

YOU: trưởng khoa khoa công ngheej thông tin
KEY: trưởng khoa công nghệ thông tin

YOU: tin học uwngss dụng
KEY: tin học ứng dụng

YOU: bộ môn cntt
KEY: bộ môn Công nghệ thông tin

YOU: khoa công nghệ thông tjn
KEY: khoa công nghệ thông tin
```

Hình 48 Ví dụ minh họa chuyển câu sai chính tả ở miền đóng

```
YOU: tôy khỏe còn bạn thì sao
KEY: tôi khỏe còn bạn thì sao

YOU: cậu c biết nấu cơm không
KEY: cậu có biết nấu cơm không

YOU: cậu chẳng lam được gì ngếu không có chìa khoá đâu
KEY: cậu chẳng làm được gì nếu không có chìa khoá đâu

YOU: nếu do là sự thật cậu đã không nói ra
KEY: nếu đó là sự thật cậu đã không nói ra

YOU: chờ cái gif
KEY: chờ cái gì

YOU: tôi trẳng trả giá đâu khoả năng của tôi là 400 đô
KEY: tôi chẳng trả giá đâu khả năng của tôi là 400 đô

YOU: đi chơi với tooi không
KEY: đi chơi với tôi không

YOU: đi ăn sáng không naof
KEY: đi ăn sáng không muốn

YOU: cậu có thích chơi ddass banh không
KEY: cậu có thích chơi sâm banh không

YOU: đi đá banh với tôi khongo
KEY: đi đá banh với tôi teo
```

Hình 49 Ví dụ minh họa chuyển câu sai chính tả ở miền mở

Bảng 10, trình bày kết quả khi thực hiện tính độ chính xác với giải thuật phân lớp kNN chuyển câu sai chính tả về đúng chính tả, khi sử dụng tập kiểm tra (100 câu hỏi có chứa lỗi sai chính tả), đơn vị tính phần trăm (%).

k	**1**	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**
Miến đóng	72,00	69,74	69,84	69,74	69,54	69,54	69,54	69,54	69,35
Miến mở	73,86	65,18	64,20	64,59	64,59	64,69	64,69	64,59	64,59

Bảng 10 Kết quả huấn luyện kNN chuyển câu sai chính tả về đúng chính tả

Từ kết quả ở Bảng 10, chúng tôi lựa chọn điểm $k = 1$ để sử dụng cho việc chuyển câu sai chính tả về đúng chính tả, ví dụ minh họa được trình bày ở Hình 50 và 51.

```
nhap: điều kiện thực tập thực tế hệ thống thông tyn
key:  điều kiện thực tập thực tế hệ thống thông tin
nhap: trưởng khoa là ai
key:  trưởng khoa là ai
nhap: điều kiiện niên luận thud
key:  mã hpsh niên luận thud
nhap: ct100 hojc phần
key:  ct100 học phần
nhap: xếp loại giỏi khi nafo
key:  xếp loại giỏi khi nào
nhap: học phần song hành quản lý dự án phần meefm
key:  học phần song hành quản lý dự án phần mềm
nhap: học phần nien luận cơ sở ngành ktpm có fải là ghuyên ngành ngành cntt không
key:  học phần niên luận cơ sở ngành ktpm có phải là chuyên ngành ngành cntt không
nhap: hông học ct453 có học ct101 được không
key:  không học ct463 có học ct173 được không
nhap: lịch thy học kỳ 2 2020 2021 ct109 nhóm 06
key:  lịch thi học kỳ 2 2020 2021 ct109 nhóm 06
nhap: khi nào bị cho thoi học
key:  khi nào bị cho thôi học
nhap: học phần nền tảng phần mềm nhúng và iot thuộc
key:  học phần nền tảng phần mềm nhúng và iot thuộc
nhap: tiết 5 bắt đầu luc mấy gờ
key:  tiết 5 bắt đầu lúc mấy giờ
```

Hình 50 Ví dụ minh họa chatbot sửa lỗi chính tả với kNN cho miền đóng

```
nhap: xin chaof
key:  xin lỗi
nhap: sao cậu không hề nói với toi cậuu định làm thế này
key:  sao cậu không hề nói với tôi cậu định làm thế này
nhap: khôn sao đâu tên tôi là elyzabeth
key:  không sao đâu tên tôi là elizabeth
nhap: anh khoe không
key:  anh khoe không
nhap: thế thì anh hãy nosi xem anh đang suy tính gì vậy
key:  thế thì anh hãy nói xem anh đang suy tính gì vậy
nhap: chúng ta cùng nhau đi chơi nhes
key:  chuyện này diễn ra bao lâu rồi
nhap: cậu có muốn ddi xem phim không
key:  không thì đã tưởng anh tán tôi
nhap: thôi nào đi ăn trưa naof người anh em
key:  hay là chúng ta liều bước lên đó đi
nhap: cậu nghĩ sao veef bữa tối
key:  tôi sẽ gọi thêm trợ giúp
```

Hình 51 Ví dụ minh họa chatbot sửa lỗi chính tả với kNN cho miền mở

Khi thực nghiệm với nhóm câu hỏi ở miền mở để thực hiện chuyển câu sai chính tả về đúng chính tả, số câu được chuyển đúng còn khá thấp, tuy vậy kết quả vẫn có thể là ở mức chấp nhận được, vì lượng dữ liệu câu hỏi cho miền mở còn khá hạn chế nên dẫn đến kết quả là chưa tốt.

Bên cạnh việc tách biệt huấn luyện riêng chuyển câu không dấu thành có dấu và câu sai chính tả về đúng chính tả; chúng tôi thực nghiệm thêm việc huấn luyện kết hợp cả hai hình thức có cả câu không dấu và câu sai chính tả sử dụng mô hình Transformer. Các số liệu dùng để huấn luyện được trình bày ở Bảng 11, tập dữ liệu sử dụng là kết hợp cả hai tập chuyển câu không dấu thành có dấu, và câu sai chính tả về đúng chính tả tương ứng với mỗi miền, kết quả đánh giá BLEU được trình bày ở Bảng 12.

Tham số	**Miền đóng**	**Miền mở**
Tập dữ liệu	339.061 cặp câu	795.156 cặp câu
Độ dài tối đa 1 câu	25 từ	20 từ
Số từ trong bộ từ điển	13.654 từ	23.011 từ
Số epoch huấn luyện	200	200
Thời gian huấn luyện Transformer	2 giờ 13 phút	6 giờ 6 phút 52 giây
Tập kiểm tra	100 câu	100 câu

Bảng 11 Bảng thống kê số liệu cho chuyển câu không dấu và câu sai chính tả

Điểm số BLEU	**Số từ tối đa trên một câu trong cặp câu hỏi và câu trả lời**	
	Miền đóng (25 từ)	**Miền mở (20 từ)**
BLEU-1	0,97	0,97
BLEU-2	0,96	0,96
BLEU-3	0,94	0,94
BLEU-4	0,92	0,91

Bảng 12 Kết quả điểm sô BLEU cho tác vụ chuyển câu không dấu thành câu có dấu và tác vụ sửa câu sai chính tả

Ví dụ minh họa chatbot được huấn luyện kết hợp chuyển câu không dấu thành câu có dấu và chuyển câu sai chính tả thành đúng chính tả ở miền đóng được trình bày ở Hình 52, và ở miền mở được trình bày ở Hình 53.

```
YOU: chao ban
KEY: chào bạn

YOU: ban co the giup gi cho toi
KEY: bạn có thể giúp gì cho tôi

YOU: trương khoa la ai
KEY: trưởng khoa là ai

YOU: bộ moon thud
KEY: bộ môn thud

YOU: ma học phần ltcb a
KEY: mã học phần ltcb a

YOU: học phaanff tiên quyết là j
KEY: học bổng quyết là gì

YOU: kế hoạch hocj tập
KEY: kế hoạch học tập

YOU: diều kiện tốt nghiệp là gi
KEY: điều kiện tốt nghiệp là gì

YOU: ma hoc phan dien toan dam may
KEY: mã học phần điện toán đám mây
```

Hình 52 Ví dụ minh họa chatbot được huấn luyện chuyển câu không dấu thành câu có dấu và chuyển câu sai chính tả thành câu đúng chính tả ở miền đóng

```
YOU: bạn nghĩ sao veeff chuyến du lịch xa
KEY: bạn nghĩ sao về chuyến du lịch xa

YOU: hay là cùng nhau đi ăn cơm did
KEY: hay là cùng nhau đi ăn cơm nữa

YOU: hôm nay trời có nhiêu xao quá
KEY: hôm nay trời có nhiều sao quá

YOU: thời tiết hoom nay thế nào
KEY: thời tiết hôm nay thế nào

YOU: cau nghi sao ve viec song o thanh pho
KEY: cậu nghĩ sao về việc sống ở thành phố

YOU: hay la chung ta chuyển nhà đi
KEY: hay là chúng ta chuyển nhà đi

YOU: tôi nghix anh cần được nghỉ ngơi
KEY: tôi nghĩ anh cần được nghỉ ngơi
```

Hình 53 Ví dụ thực nghiệm chatbot được huấn luyện chuyển câu không dấu thành câu có dấu và chuyển câu sai chính tả thành câu đúng chính tả ở miền mở

Bên cạnh việc sử dụng kNN cho việc chuyển câu sai chính tả và đúng chỉnh tả, thì ở luận văn này còn vận dụng kNN cho việc phân loại câu hỏi được nhập vào, từ đó chuyển đổi linh hoạt giữa hai miền của chatbot, kết quả sau khi huấn luyện với tập dữ liệu 903.720 câu hỏi được gán nhãn 0 và 1, với 0 là câu hỏi cho miền mở và 1 là câu hỏi miền đóng cùng với 1.000 câu dùng làm tập kiểm tra, kết quả được trình bày ở bảng bên dưới (Bảng 13).

k	1	3	5	7	9	11	13	15	17
Độ chính xác	98,71	98,22	98,00	97,91	97,91	97,91	97,91	97,74	97,74

Bảng 13 Kết quả sử dụng kNN cho tác vụ phân loại câu hỏi theo miền

Kết quả sử dụng kNN cho phân loại câu hỏi theo miền đóng hay miền mở đạt kết quả khá cao. Tuy nhiên khi thực nghiệm thực tế thì kết quả không đạt được như yêu cầu, nhưng vẫn có thể chấp nhận được cho việc phân loại các câu hỏi được đưa vào. Kết quả sử dụng kNN cho phân loại câu hỏi như Hình 54, kết quả 1 cho phân loại miền đóng và 0 cho miền mở:

```
nhap: trưởng khoa là ai
key:  1
nhap: bộ môn cntt
key:  1
nhap: học phần vi tích phân a1
key:  0
nhap: đi chơi với tôi không
key:  0
nhap: học phần là gì
key:  1
nhap: ct101
key:  1
nhap: nguyên lý hệ điều hành
key:  0
```

Hình 54 Ví dụ phân loại câu hỏi theo miền với kNN

Hình 55 trình bày ví dụ minh họa thực nghiệm nhận diện thực thể NER. Kết quả chưa được tốt, quá trình gán nhãn vẫn còn nhiều thiếu sót, nhưng nhìn chung đã rút trích được và nhận diện được những thực thể chính có trong câu.

```
YOU: khi nào tôi có thể rút học phần vậy
NER: thể rút học phần

YOU: tôi cần biết thông tin mã học phần nguyên lý máy học
NER: thông tin mã học phần nguyên lý máy học

YOU: chào bot
NER: chào bot

YOU: bạn có thể giúp gì được cho tôi
NER: bạn thể

YOU: cho tôi biết về trưởng khoa là ai
NER: trưởng khoa

YOU: giải thích giúp tôi học phần điều kiện
NER: giải học phần điều
```

Hình 55 Ví dụ thực nghiệm rút trích câu hỏi với NER

Kết quả điểm số BLEU cho trợ lý ảo sử dụng tất cả các phương pháp cải thiện chất lượng chatbot được trình bày ở Bảng 14. Kết quả thực nghiệm cho thấy

Độ nhiễu của các câu hỏi kiểm tra	BLEU-1	BLEU-2	BLEU-3	BLEU-4
100% không dấu	0.281	0.247	0.211	0.190
50% không dấu	0.294	0.253	0.223	0.198
10% không dấu	0.292	0.251	0.222	0.198
50% sai chính tả	0,287	0,248	0,220	0,197
10% sai chính tả	0.292	0.251	0.222	0.198
100% không dấu và 50% sai chính tả	0,279	0,237	0,209	0,190
50% không dấu và 50% sai chính tả	0.280	0.243	0.212	0.190
30% không dấu và 30% sai chính tả	0.300	0.256	0.225	0.200
10% không dấu và 10% sai chính tả	0.292	0.251	0.222	0.198

Bảng 14 Kết quả điểm sô BLEU cho chatbot có kết hợp chuyển câu không dấu thành câu có dấu và sửa câu sai chính tả

Với kết quả thu được, khi so sánh với kết quả chấm điểm BLEU ở Bảng 4, các trường hợp cho ra kết quả khá là gần nhau, điều này cho thấy chatbot khi có tác vụ chuyển câu không dấu thành có dấu và sửa câu sai chính tả hoạt động tương đối tốt. Vì có hạn chế về mặt dữ liệu học dẫn đến kết quả chưa tốt, cần cải thiện thêm bằng cách tăng dữ liệu học cho chatbot hơn nữa.

CHƯƠNG 5. KẾT LUẬN

Chương này trình bày tổng kết kết quả đạt được của nghiên cứu và đề xuất hướng phát triển cho đề tài.

5.1. Kết quả đạt được và những hạn chế

Trong luận văn này, chúng tôi đã xây được một chatbot miền đóng, hay chính xác hơn là một trợ lý ảo dành cho sinh viên khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ, dựa trên mô hình Transformer. Hệ thống có khả năng đưa ra các câu trả lời tốt, có khả năng hiểu và trả lời được các câu hỏi người dùng nhập sai chính tả hoặc gõ không dấu. Chúng tôi còn thực nghiệm xây dựng chatbot miền mở với những khả năng tương tự như chatbot miền đóng. Ngoài ra, bên cạnh việc xây dựng hai chatbot ở hai miền, chúng tôi đã tạo nên liên kết cho chatbot ở cả hai miền bằng cách tạo phân loại và nhận diện câu đầu vào và từ đó chọn đúng miền chatbot để đưa ra câu trả lời là phù hợp nhất có thể. Tuy đạt được kết quả khá tốt, nhưng dữ liệu được xây dựng cho chatbot miền đóng còn khá hạn chế, các câu hỏi và câu trả lời vẫn chưa bao quát hết các vấn đề liên quan đến miền đóng. Mặc dù còn hạn chế về mặt dữ liệu, nhưng nhìn tổng thể chatbot mà chúng tôi xây dựng đã hoàn thành mục tiêu đề tài đặt ra ban đầu.

5.2. Hướng phát triển

Xây dựng hệ thống trả lời tự động dựa trên mô hình Transformer cho tiếng Việt có kết quả là khá tốt, tuy vẫn còn một số hạn chế về câu trả lời được đưa ra, nhưng vẫn có thể cải thiện để bot có thể trả lời tốt hơn. Chúng tôi sẽ mở rộng và cải thiện hơn nữa tập dữ liệu huấn luyện cho chatbot, điều chỉnh các tham số đầu vào cho mô hình Tranformer để thu được nhiều kết quả tốt hơn. Kỳ vọng tương lai là chatbot chúng tôi xây dựng sẽ có thể được tích hợp vào website của khoa Công nghệ Thông tin và Truyền thông hoặc Facebook của Khoa để có thể hỗ trợ nhiều hơn cho các bạn sinh viên.

TÀI LIỆU THAM KHẢO

[1] Rashmi Dharwadka, Neeta A. Deshpande, "A Medical ChatBot," *Int J Comp Trends Technol*, vol. 60.1, 2018.

[2] Qiming Bao, Lin Ni, Jiamou Liu, "HHH: an online medical chatbot system based on knowledge graph and hierarchical bi-directional attention," *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1-10, 2020.

[3] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, Joelle Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[4] BANCHS, Rafael E, "Movie-DiC: A movie dialogue corpus," *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 203-207, 2012.

[5] Asmund Kamphaug , Ole-Christoffer Granmo, Morten Goodwin and Vladimir I. Zadorozhny , "Towards open domain chatbots—a gru architecture for data driven conversations," *International Conference on Internet Science* , pp. 213-222, 2017.

[6] Yogi Wisesa Chandra, Suyanto Suyanto, "Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model," *Procedia Computer Science*, vol. 157, pp. 367-374, 2019.

[7] L. N. Nam, "Xây dựng Chatbot giới thiệu Khoa Công nghệ Thông tin và Truyền Thông," 2019.

[8] N. N. L. J. K. Khang Nhut Lam, "Building a Chatbot on a Closed Domain using RASA," pp. 144-148, 2020.

[9] N. V. Vĩ, "Xây dựng hệ thống trả lời tự động bằng phương pháp học tăng cường và tự phê bình," 2019.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, "Attention Is All You Need," 2017.

[11] Hochreiter, Sepp, Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, pp. 1735 - 1780, 1997.

[12] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[13] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," *Proceedings of the*

40th annual meeting of the Association for Computational Linguistics, pp. 311-318, 2002.

[14] P. B. C. Quốc, "Transformer Dịch máy giữa ngôn ngữ Việt Anh," [Online]. Available: https://github.com/pbcquoc/transformer.

[15] N. C. Thắng, "Named Entity Recognition – Nhận diện thực thể trong câu khi xử lý ngôn ngữ tự nhiên," 16 08 2020. [Online]. Available: https://www.miai.vn/2020/08/16/named-entity-recognition-nhan-dien-thuc-the-trong-cau-khi-xu-ly-ngon-ngu-tu-nhien/.

[16] J. Alammar, "The illustrated transformer. GitHub Blog," 2018. [Online]. Available: http://jalammar. github. io/illustrated-transformer.

[17] D. V. Tú, "Xây dựng mô hình nhận diện giọng nói tiếng Việt," 2019.

[18] C. Olah, "Understanding lstm networks," 2015.