TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

———————— * ————————

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI
Xây dựng mô hình sinh câu trả lời cho hệ thống Chatbot trong miền đóng

Sinh viên thực hiện : **Nguyễn Ngọc Linh**
Lớp CNTT2.03 – K58
Giáo viên hướng dẫn: PGS.TS **Lê Thanh Hương**

HÀ NỘI 06-2018

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Ngọc Linh

Điện thoại liên lạc: 0986 345 280 Email: ngoclinhnguyen.hust@gmail.com

Lớp: CNTT2.03 – K58 Hệ đào tạo: Chính quy

Đồ án tốt nghiệp được thực hiện tại: Trường đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 15/01/2018 đến 27/05/2018

2. Mục đích nội dung của ĐATN

Xây dựng mô hình sinh câu trả lời cho Chatbot trong miền đóng – lĩnh vực bán hàng thời trang sử dụng mô hình sinh chuỗi Sequence to sequence.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu về Chatbot, các hướng tiếp cận cho bài toán Chatbot
- Đề xuất mô hình xây dựng hệ thống Chatbot trong miền đóng
- Tìm hiểu Deep Learning và mô hình sinh chuỗi Seq2seq và attention Seq2seq
- Xây dựng và cài đặt mô hình
- Đánh giá mô hình thông qua dữ liệu test
- Kết luận, đưa ra ưu nhược điểm và đề xuất cải thiện mô hình

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Ngọc Linh* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS.TS. Lê Thanh Hương*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Nguyễn Ngọc Linh

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

Hà Nội, ngày tháng năm

Giáo viên hướng dẫn

PGS.TS. Lê Thanh Hương

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Đồ án tốt nghiệp giới thiệu và đưa ra các vấn đề của các hệ thống Chatbot miền đóng hiện nay, đề xuất phương pháp và mô hình xây dựng một hệ thống Chatbot trong lĩnh vực bán hàng thời trang.

Do thời gian không đủ để xây dựng một hệ thống Chatbot hoàn chỉnh, đồ án này chỉ xây dựng và đi sâu vào mô hình sinh câu trả lời cho Chatbot sử dụng mô hình sinh chuỗi Sequence to Sequence và Attention Sequence to Sequence. Mô hình được huấn luyện với 4 bộ tham số khác nhau và được đưa ra để so sánh bằng việc đánh giá các kết quả mà mỗi bộ tham số dự đoán được. Từ kết quả so sánh cho thấy việc áp dụng Attention Sequence to Sequence cho bài toán Chatbot không mang lại hiệu quả cao bằng việc sử dụng mô hình Sequence to Sequence cơ bản.

Bài báo cáo cũng đưa ra ưu nhược điểm và lý giải kết quả so sánh này. Một số đề xuất và hướng phát triển xây dựng hệ thống Chatbot cũng được đề cập ở phần Kết luận.

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2
TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3
MỤC LỤC 4
DANH MỤC BẢNG BIỂU 6
DANH MỤC HÌNH ẢNH 7
DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 8
LỜI CẢM ƠN 9
LỜI MỞ ĐẦU 10
PHẦN 1: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP 11
1.1. Giới thiệu bài toán 11
1.1.1. Giới thiệu Chatbot 11
1.1.2. Phân loại Chatbot 11
1.1.2.1. Phân loại theo dạng tương tác với con người 12
1.1.2.2. Phân loại theo miền ứng dụng 12
1.1.2.3. Phân loại theo hướng tiếp cận 12
1.1.3. Các phương pháp tiếp cận và kỹ thuật xây dựng Chatbot 13
1.1.3.1. Scripted Chatbot 13
1.1.3.2. AI Chatbot 13
1.1.4. Đặt vấn đề 14
1.1.5. Mục đích và phạm vi áp dụng 14
1.2. Định hướng giải pháp 14
1.3. Cơ sở lý thuyết áp dụng 16
1.3.1. Mạng nơ-ron nhân tạo (ANN – Artificial Neural Network) 16
1.3.2. Mạng nơ-ron hồi quy (RNN – Recurrent Neural Network) 17
1.3.3. Mạng LSTM (Long Short Term Memory Networks) 19
1.3.4. Word Embedding 23
1.3.5. Mô hình sinh chuỗi Sequence to Sequence 23
1.3.6. Attention Sequence to sequence 25
1.3.7. Hàm Softmax và loss 26

1.3.8. Một vài kỹ thuật tối ưu hóa hàm mất mát 26
PHẦN 2: CÁC KẾT QUẢ ĐẠT ĐƯỢC 29
2.1. Các thư viện, tài liệu tham khảo được sử dụng 29
2.1.1. Ngôn ngữ lập trình sử dụng 29
2.1.2. TensorFlow 29
2.1.3. Underthesea 29
2.1.4. Flask app 30
2.1.5. Một số thư viện khác 30
2.2. Tập dữ liệu 30
2.2.1. Thông tin dữ liệu/ Định dạng dữ liệu 30
2.2.2. Xử lý dữ liệu 32
2.3. Áp dụng seq2seq/ attention seq2seq vào bài toán Chatbot 33
2.3.1. Quá trình huấn luyện 33
2.3.2. Quá trình dự đoán, sinh ra câu trả lời 35
2.3.3. Sử dụng mô hình attention seq2seq 36
2.3.4. Áp dụng triển khai cài đặt 37
2.4. Cài đặt, đánh giá hiệu năng và kết quả của mô hình 40
2.4.1. Cài đặt và huấn luyện mô hình 40
2.4.2. Giao diện ứng dụng và một số hình ảnh demo 42
2.4.3. Đánh giá hiệu năng và kết quả 44
2.4.3.1. Kết quả 44
2.4.3.2. Nhận xét và giải thích 49
KẾT LUẬN 51
TÀI LIỆU THAM KHẢO 52
PHỤ LỤC A 54

DANH MỤC BẢNG BIỂU

Bảng 1: Danh mục từ viết tắt và thuật ngữ 8
Bảng 2: Bảng danh sách nhãn thực thể NER 32
Bảng 3: Danh sách bộ tham số được huấn luyện 40
Bảng 4: Thời gian huấn luyện và thời điểm dừng huấn luyện của các bộ tham số ..42
Bảng 5: Một vài kết quả đánh giá của bộ tham số 128_attention_3_layers 45
Bảng 6: Một vài kết quả đánh giá của bộ tham số 512_attention_3_layers 46
Bảng 7: Một vài kết quả đánh giá của bộ tham số 512_attention_1_layer 47
Bảng 8: Một vài kết quả đánh giá bộ tham số 512_no_attention_3_layers 48
Bảng 9: Đánh giá mô hình với các bộ tham số khác nhau 48

DANH MỤC HÌNH ẢNH

Hình 1.1: Đề xuất phương pháp xây dựng hệ thống Chatbot 15
Hình 1.2: Mạng RNN 17
Hình 1.3: Mô hình duỗi thẳng của mạng RNN 18
Hình 1.4: Vấn đề phụ thuộc xa của mạng RNN.....19
Hình 1.5: Kiến trúc mạng LSTM 19
Hình 1.6: Các kí hiệu trong kiến trúc mạng LSTM 20
Hình 1.7. Trạng thái tế bào của LSTM 20
Hình 1.8: Cấu trúc cổng trong LSTM 21
Hình 1.9: Tầng cổng quên của LSTM.....21
Hình 1.10: Tầng cổng vào.....22
Hình 1.11: Trạng thái tế bào thu được sau khi đi qua cổng quên và cổng vào.....22
Hình 1.12: Kết quả đầu ra và trạng thái tế bào của LSTM 23
Hình 1.13: Cấu trúc mô hình seq2seq 23
Hình 1.14: Minh họa mô hình seq2seq sử dụng 2 mạng neural LSTM cho thành phần encoder và decoder. 24
Hình 1.15: Attention visualization – ví dụ sự sắp xếp giữa các câu nguồn và câu đích [11] 25
Hình 2.1: Quá trình huấn luyện mô hình Seq2seq 33
Hình 2.2. Quá trình dự đoán của mô hình Seq2seq 35
Hình 2.3. Sử dụng mô hình attention seq2seq 36
Hình 2.4: Cơ chế tính toán attention 37
Hình 2.5: Hai hàm score tính trọng số attention phổ biến 37
Hình 3.4: Đồ thị eval và train của bộ tham số 128_attention_3_layers.....41
Hình 3.5: Đồ thị eval và train của bộ tham số 512_attention_3_layers.....41
Hình 3.6: Đồ thị eval và train của bộ tham số 512_attention_1_layer 42
Hình 3.7: Đồ thị eval và train của bộ tham số 512_no_attention_3_layers.....42
Hình 3.8. Giao diện demo 43
Hình 3.9. Giao diện demo 43
Hình 3.10. Giao diện demo 43
Hình 3.11. Giao diện demo 43

DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

STT	Từ viết tắt	Thuật ngữ	Giải thích
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	ML	Machine Learning	Học máy
3	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
4	DL	Deep Learning	Học sâu
5	ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
6	RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
7	LSTM	Long Short Term Memory	Mạng bộ nhớ thuật ngữ ngắn dài
8	GD, SGD	Gradient Descent, Stochastic Gradient Descent	
9	XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng
10	AIML	Artificial Intelligence Markup Language	Ngôn ngữ đánh dấu trí tuệ nhân tạo
11	NER	Named-Entity Recognition	Nhận dạng thực thể
12	Seq2seq	Sequence to Sequence	Mô hình sinh chuỗi sang chuỗi
13	CRF	Conditional Random Field	Mô hình xác suất trường điều kiện ngẫu nhiên
14	CNN	Convolution Neural Network	Mạng nơ-ron tích chập

Bảng 1: Danh mục từ viết tắt và thuật ngữ

LỜI CẢM ƠN

Năm năm học tại ngôi trường Bách Khoa thật nhiều cảm xúc, buồn vui, hạnh phúc, những giọt nước mắt và những nụ cười, niềm vui, lòng tự hào đầy kiêu hãnh vì được sống và học tập tại đây.

Em xin cảm ơn các thầy cô giáo giảng viên trường đại học Bách Khoa Hà Nội và đặc biệt là các thầy cô giáo của Viện Công Nghệ Thông Tin và Truyền Thông đã luôn nhiệt tình dạy dỗ và giúp đỡ chúng em, trao cho chúng em chìa khóa đến con đường tri thức trong ngành.

Em xin chân thành cảm ơn *PGS.TS. Lê Thanh Hương*, người đã luôn giúp đỡ, hướng dẫn và chỉ bảo em tận tình trong thời gian thực hiện đồ án tốt nghiệp này.

Con xin chân thành cảm ơn ông bà, bố mẹ đã sinh ra con, yêu thương con và luôn chăm chút, ở bên con cho con những điều tốt nhất để con có cơ hội trải qua những chặng đường tuyệt vời trong cuộc đời và Bách Khoa là điều tuyệt vời trong số đó.

Cuối cùng, xin chân thành cảm ơn các anh chị, bạn bè và người ấy, những người luôn sát cánh, luôn cổ vũ động viên và giúp đỡ tinh thần của em trong quá trình thực hiện đồ án này.

LỜI MỞ ĐẦU

Sự phát triển của trí tuệ nhân tạo (Artificial Intelligent) và học máy (Machine Learning) những năm gần đây đã nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống như hệ thống nhận dạng khuôn mặt của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon,…

Chatbot cũng là một ứng dụng trong số đó. Nó có thể đem lại lợi ích to lớn cho con người, đặc biệt là trong kinh doanh thương mại. Đồ án tốt nghiệp này xây dựng mô hình sinh câu trả lời cho hệ thống Chatbot trong miền đóng sử dụng mô hình sinh chuỗi. Bài báo cáo được chia làm ba phần như sau:

Phần 1. Đặt vấn đề và định hướng giải pháp: Chương này giới thiệu về Chatbot, các mô hình xây dựng Chatbot và đề xuất phương pháp xây dựng một hệ thống Chatbot trong miền đóng. Sau đó giới thiệu và tóm tắt cơ sở lý thuyết áp dụng trong đồ án.

Phần 2. Các kết quả đạt được: Chương này trình bày nội dung cài đặt, kết quả cài đặt và đánh giá mô hình sinh chuỗi Seq2seq.

Kết luận: Phần này đưa ra kết luận và hướng phát triển cho hệ thống sau này.

PHẦN 1: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

1.1. Giới thiệu bài toán

1.1.1. Giới thiệu Chatbot

Chatbot là một chương trình máy tính tương tác với người dùng bằng ngôn ngữ tự nhiên dưới dạng một giao diện đơn giản, âm thanh hoặc dưới dạng tin nhắn.

Tại F8, CEO Mark Zuckerberg đã chứng minh sức mạnh tiềm năng của chatbot bằng một chương trình 1-800-Flowers, một dịch vụ đặt hoa tại Mỹ. Người dùng không cần phải gọi điện mà vẫn đặt hoa nhanh chóng với Messenger, và các công việc khác như đặt taxi, cập nhật thị trường chứng khoán... cũng sẽ được thực hiện hoàn toàn trên ứng dụng nhắn tin của Facebook một cách dễ dàng và đơn giản. [1]

Bên cạnh đó lợi ích mà chatbot có thể đem lại rất lớn, đặc biệt cho các doanh nghiệp giúp giảm thiểu chi phí chăm sóc khách hàng bằng cách trả lời các câu hỏi hay các yêu cầu đơn giản, tiết kiệm chi phí thuê nhân công, gửi thông báo hay nhắn tin định kỳ cho khách hàng vào những ngày đặc biệt, xử lý được nhiều hơn và hiệu quả hơn trong cùng 1 khoảng thời gian và tăng trải nghiệm người dùng không có từ ngữ nào không đúng mực với khách hàng, xử lý được 80% nhu cầu của khách ngay lập tức, khách hàng sẽ luôn được phục vụ mà không cần chờ đợi… [2]

Ngoài ra, chatbot dưới dạng các trợ lý ảo còn giúp con người đơn giản hóa cuộc sống, chỉ một ứng dụng có thể thực hiện được nhiều chức năng của ứng dụng khác như mua sắm, đặt chỗ, book vé, thanh toán trực tuyến,... giúp con người tiết kiệm được thời gian và công sức.

Chatbot sẽ là phương tiện mới, hoạt động hiệu quả và nhanh chóng phổ biến, thay thế các ứng dụng trong tương lai gần. Đó chính là bước đột phá mà chatbot có thể đem lại cho cuộc sống.

Một số ứng dụng của Chatbot hiện nay:

- Trợ lý ảo như Siri của Apple, Tay của Microsoft, Cortana của Google
- Chatbot tư vấn quần áo – thời trang (H&M)
- Chatbot order pizza – thực phẩm (Dominos Pizza),
- …

1.1.2. Phân loại Chatbot

Có nhiều cách phân loại Chatbot như phân loại theo miền ứng dụng, phân loại theo hướng tiếp cận, và phân loại theo dạng tương tác với con người.

1.1.2.1. Phân loại theo dạng tương tác với con người

Chatbot 2 dạng tương tác với con người bao gồm:

- **Audiotory Chatbot**: Chatbot giao tiếp bằng giọng nói, âm thanh [2]
- **Textual Chatbot:** Chatbot giao tiếp thông qua tin nhắn, các cuộc hội thoại bằng văn bản. [2]

1.1.2.2. Phân loại theo miền ứng dụng

Có 2 miền ứng dụng chính của Chatbot bao gồm:

- **Miền đóng (Closed domain):** [3]

  Không gian đầu vào và đầu ra có thể trong miền đóng có giới hạn bởi vì hệ thống Chatbot miền đóng cố gắng đạt được những mục tiêu cụ thể như tư vấn khách chọn sản phẩm, chốt đơn và hỏi thông tin ship khi khách yêu cầu đặt mua,… Những hệ thống Chatbot này không cần nói về chính trị hay những gì ngoài khả năng của chúng, chúng chỉ cần hoàn thành tốt nhiệm vụ của mình càng hiệu quả càng tốt. Chắc chắn người dùng có thể thực hiện cuộc trò chuyện với bất kì điều gì họ muốn nhưng hệ thống thì không cần xử lý tất cả các trường hợp này.

- **Miền mở (Open domain):** [3]

  Trong miền mở, người dùng có thể nói chuyện về mọi thứ, không cần có mục tiêu hay ý định cụ thể nào. Số lượng chủ đề và các tri thức trên thế giới thì vô hạn, nhưng Chatbot cần được học lượng tri thức đủ nhiều để sinh ra một phản hồi hợp lý còn khá khó khăn.

1.1.2.3. Phân loại theo hướng tiếp cận

Hiện nay có hai hướng tiếp cận chính cho bài toán xây dựng Chatbot:

- **Chatbot kịch bản (Scripted Chatbot):** [4] Các Chatbot xây dựng và duy trì cuộc trò chuyện dựa trên quy tắc và hoạt động như một cây quyết định trong đó mỗi hành động của người dùng sẽ nhắc bot thực hiện hành động hoặc một phản hồi nào đó
- **Chatbot thông minh (AI Chatbot):** [4] Chatbot thông minh này được xây dựng trên các khả năng của ML và NLP. Chúng dựa trên khả năng

học tập và hấp thụ thông tin của con người giúp chúng hoạt động hiệu quả hơn và có thể xử lý nhanh hơn và linh hoạt hơn Chatbot kịch bản.

1.1.3. Các phương pháp tiếp cận và kỹ thuật xây dựng Chatbot

1.1.3.1. Scripted Chatbot

Các Chatbot kịch bản được xây dựng bằng mô hình dựa trên các tập luật (Rule-based Model). Một số kỹ thuật xây dựng Chatbot kịch bản bao gồm: [5]

- **Pattern Matching** (So khớp mẫu): Đây là kĩ thuật thông dụng và được sử dụng nhiều trong Chatbot. Các biến thể của các thuật toán so khớp mẫu tồn tại trên mọi hệ thống Chatbot hiện nay.

- **Parsing**: Textual Parsing – Phân tích cấu trúc cú pháp là một phương thức chuyển đổi văn bản gốc thành một tập hợp các từ (Phân tích từ vựng) với tính năng chủ yếu là xác định cấu trúc ngữ pháp của nó. Đầu tiên, cấu trúc từ vựng có thể được kiểm tra nếu nó tạo thành một biểu thức. Các trình cú pháp trước đó rất đơn giản, tìm từ khóa có thể nhận dạng theo thứ tự được phép. Với cách tiếp cận này, Chatbot với một tập hợp các mẫu có giới hạn có thể bao gồm nhiều câu đầu vào. Các trình cú pháp phức tạp được sử dụng trong các Chatbot sau này thực hiện phân tích ngữ pháp và hoàn chỉnh các câu tự nhiên.

- **A.I.M.L:** Để xây dựng một hệ thống Chatbot, cần có một ngôn ngữ tổng quát linh hoạt và dễ hiểu. AIML (Artificial Intelligence Markup Language) là một dẫn xuất của XML, một trong những phương pháp được sử dụng rộng rãi, đáp ứng các yêu cầu này. AIML đại diện cho tri thức được đưa vào Chatbot và dựa trên công nghệ phần mềm được phát triển cho A.L.I.C.E (Artificial Linguistic Internet Computer Entity). Nó có khả năng mô tả kiểu dữ liệu đối tượng (đối tượng AIML) và mô tả một phần các chương trình mà nó đang xử lý.

1.1.3.2. AI Chatbot

Để xây dựng một Chatbot thông minh có hai hướng tiếp cận: [6]

- **Mô hình trích xuất thông tin (Retrieval-based model)**

  Mô hình trích xuất thông tin sử dụng một kho lưu trữ các câu hỏi được xác định trước và sử dụng một số loại heuristic để chọn câu trả lời phù hợp dựa trên đầu vào và ngữ cảnh. Các heuristic này có thể đơn giản như một biểu

thức dựa trên rule-based expression match hay phức tạp như một tập hợp các bộ phân loại học máy. Các hệ thống này không tạo ra văn bản mới mà chúng chỉ chọn một câu trả lời từ một tập cố định cho trước.

- **Mô hình sinh (Generative model)**
  Mô hình sinh khá là thông minh. Chúng sinh ra câu trả lời, sinh từng từ dựa trên cầu đầu vào tuy nhiên các câu được sinh ra từ mô hình này dễ mắc phải các lỗi ngữ pháp. Các mô hình sinh thường dựa trên kỹ thuật dịch máy, nhưng thay vì dịch từ ngôn ngữ này sang ngôn ngữ khác, chúng sẽ dịch từ câu đầu vào sang câu phản hồi ở đầu ra.

1.1.4. Đặt vấn đề

Lợi ích mà Chatbot có thể đem lại cho cộng đồng đặc biệt là trong lĩnh vực kinh doanh là rất lớn. Vậy làm thế nào để có thể xây dựng một hệ thống Chatbot thông minh, giúp giải quyết, đáp ứng được nhu cầu đó trong các doanh nghiệp như hiện nay? Đó chính là lý do mà em chọn đề tài này với mong muốn có thể xây dựng một hệ thống Chatbot trợ giúp các doanh nghiệp lớn hay các cá nhân đang áp dụng mô hình kinh doanh online.

1.1.5. Mục đích và phạm vi áp dụng

- **Mục đích**: Xây dựng hệ thống Chatbot thông minh có khả năng tự động sinh câu trả lời cho những câu hỏi của khách hàng

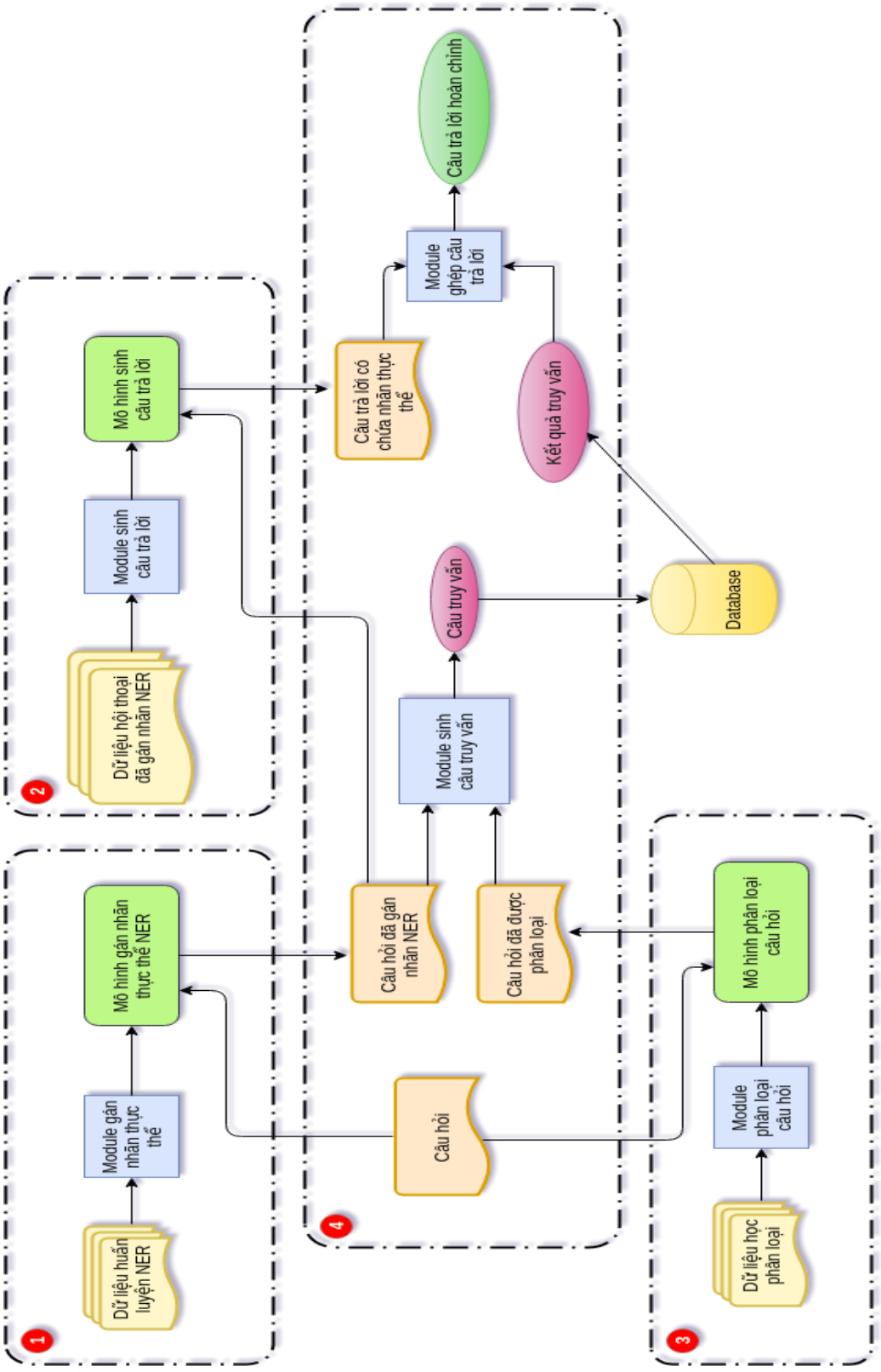
- **Phạm vi áp dụng**: Trước hết áp dụng trong lĩnh vực kinh doanh thời trang, cải tiến, tối ưu hóa khả năng của hệ thống sau đó mở rộng ra các lĩnh vực khác.

1.2. Định hướng giải pháp

Với một hệ thống Chatbot thông minh sử dụng mô hình sinh có thể sinh ra một câu trả lời chính xác. Nhưng để có thể áp dụng mô hình đó với nhiều cửa hàng khác nhau, nhiều lĩnh vực kinh doanh khác nhau thì điều đó lại là một vấn đề khó. Bộ dữ liệu cần đủ linh hoạt và đủ cho các lĩnh vực và thông thường những mặt hàng kinh doanh có một số yếu tố nhất định và cần linh hoạt.

Với một câu hỏi như "Giá cái áo này bao nhiêu?" thì Chatbot sẽ phải làm thế nào? Nếu chỉ sinh ra một câu trả lời đã được học được từ mô hình sinh "Giá áo này 35K nhé bạn" nhưng thực tế thì mỗi loại áo lại có giá khác nhau. Nếu chỉ sử dụng mô hình sinh thì chưa đủ tốt khi nó được sử dụng trong miền đóng. Do đó cần kết hợp câu trả lời này với cơ sở dữ liệu – những thông tin tĩnh chính và có khả năng thay đổi

mà khách hàng cần biết một cách chính xác vào những thời điểm nhất định. Do đó mô hình đề xuất cho hệ thống Chatbot được mô tả như hình 1.1 dưới đây:



Hình 1.1: Đề xuất phương pháp xây dựng hệ thống Chatbot

Hệ thống Chatbot bao gồm 4 phần hay 4 hệ thống con được phân cụm như hình 1.1.

Phần 1: Xây dựng mô hình gán nhãn thực thể NER cho các đối tượng tương ứng với từng lĩnh vực kinh doanh thông qua một module gán nhãn thực thể CRF hoặc CNN hay các mô hình học máy phân loại khác bằng việc huấn luyện mô hình thông qua dữ liệu huấn luyện đã có nhãn NER.

Phần 2: Sử dụng kết quả của phần 1 để dự đoán nhãn NER của tập dữ liệu hội thoại thu được dữ liệu hội thoại có nhãn NER. Sau đó sử dụng dữ liệu này để huấn luyện module sinh câu trả lời để học ra mô hình sinh câu trả lời cho hệ thống có chứa nhãn thực thể.

Phần 3: Sử dụng tập dữ liệu học phân loại câu hỏi để học mô hình phân loại câu hỏi cho từng câu hỏi.

Phần 4: Xây dựng module sinh câu truy vấn và module ghép sử dụng các mô hình của 3 phần trên kết hợp với Database để sinh ra một câu trả lời hoàn chỉnh.

Cụ thể, một câu hỏi đầu vào từ phía người dùng sẽ được đi qua mô hình gán nhãn thực thể và mô hình phân loại câu hỏi để gán nhãn và phân loại. Tiếp theo, câu hỏi đã có nhãn thực thể được sử dụng làm đầu vào của mô hình sinh câu trả lời trong phần 2 để sinh ra một câu trả lời có chứa nhãn thực thể. Sau đó, câu hỏi đã có nhãn thực thể và đã được phân loại được đưa qua module sinh câu truy vấn để thu được một câu truy vấn. Câu truy vấn này được sử dụng để truy vấn trong cơ sở dữ liệu để trả về kết quả truy vấn của câu hỏi ban đầu. Cuối cùng, module ghép câu trả lời sẽ kết hợp câu trả lời có chứa nhãn thực thể và kết quả truy vấn để thu được một câu trả lời hoàn chỉnh, đầy đủ thông tin và trả lời lại về phía người dùng.

Trong đồ án tốt nghiệp này, em thực thi phần 2, xây dựng mô hình sinh câu trả lời sử dụng mô hình sinh chuỗi Sequence to Sequence từ tập dữ liệu hội thoại đã gán nhãn thực thể từ phần 1.

1.3. Cơ sở lý thuyết áp dụng

1.3.1. Mạng nơ-ron nhân tạo (ANN – Artificial Neural Network)

Mạng nơ-ron nhân tạo mô phỏng các hệ thống nơ-ron sinh học (các bộ não con người). ANN là một cấu trúc (structure/network) được tạo nên bởi một số lượng các nơ-ron (artificial neurons) liên kết với nhau. Mỗi nơ-ron có một đặc tính vào/ra và thực hiện một tính toán cục bộ (một hàm cục bộ).

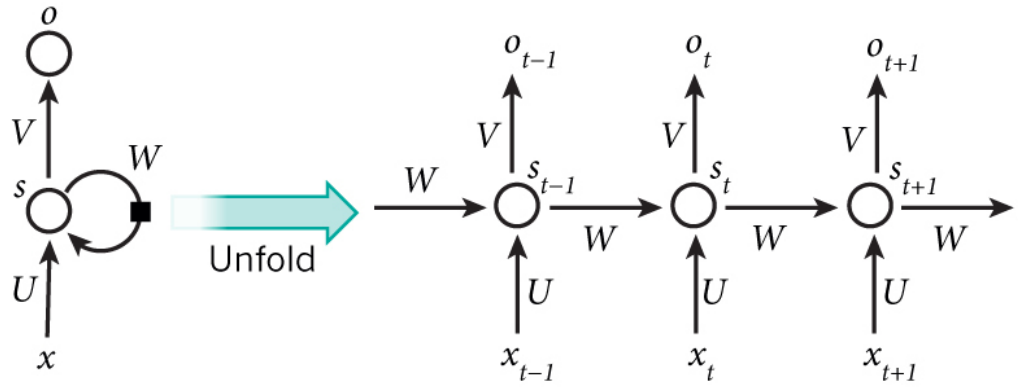
ANN có thể được xem như một cấu trúc xử lý thông tin một cách phân tán và song song ở mức cao. ANN có khả năng học (learn), nhớ lại (recall), và khái quát hóa (generalize) từ các dữ liệu học.

1.3.2. Mạng nơ-ron hồi quy (RNN – Recurrent Neural Network)

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào [7].

RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó.

Về cơ bản một mạng RNN có dạng như sau:



Hình 1.2: Mạng RNN

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự. Ví dụ ta có một câu gồm 4 chữ "I love my family", thì mạng nơ-ron được triển khai sẽ gồm 4 tầng nơ-ron tương ứng với mỗi chữ một tầng. Lúc đó việc tính toán bên trong RNN được thực hiện như sau: [7]

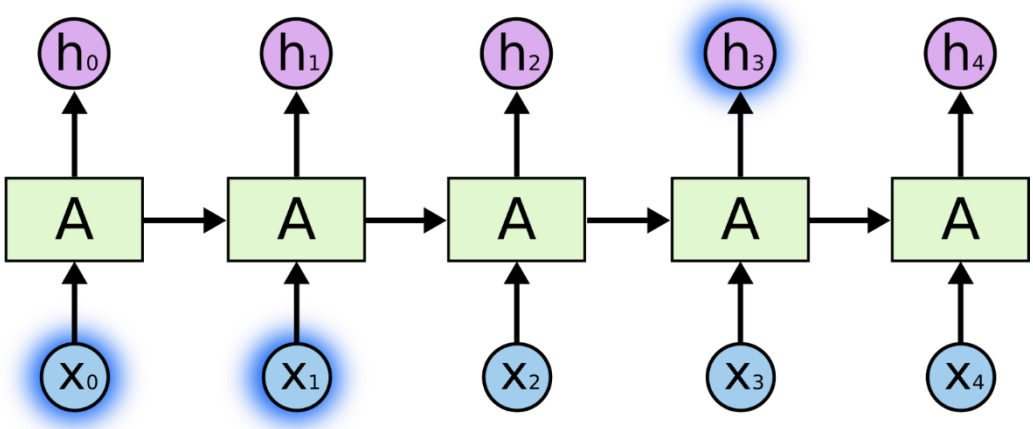
- $\mathbf{x_t}$ là đầu vào của bước t.
- $\mathbf{s_t}$ là trạng thái ẩn tại bước t. Nó chính là bộ nhớ của mạng. $\mathbf{s_t}$ được tính toán dựa trên trạng thái ẩn phía trước và đầu vào tại bước đó. $\mathbf{s_t} = f(U\mathbf{x_t} + W\mathbf{s_{t-1}})$. Hàm f thường là một hàm phi tuyến như tang hyperbolic (tanh) hay

ReLu. Để làm phép toán cho phần tử ẩn đầu tiên ta cần khởi tạo thêm s-1, thường được gán giá trị khởi tạo bằng 0.

- $\mathbf{o_t}$ chính là đầu ra tại bước t.

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Nếu mà RNN có thể làm được việc đó thì chúng sẽ cực kì hữu dụng, tuy nhiên liệu chúng có thể làm được không? Câu trả lời là còn tùy. [8]

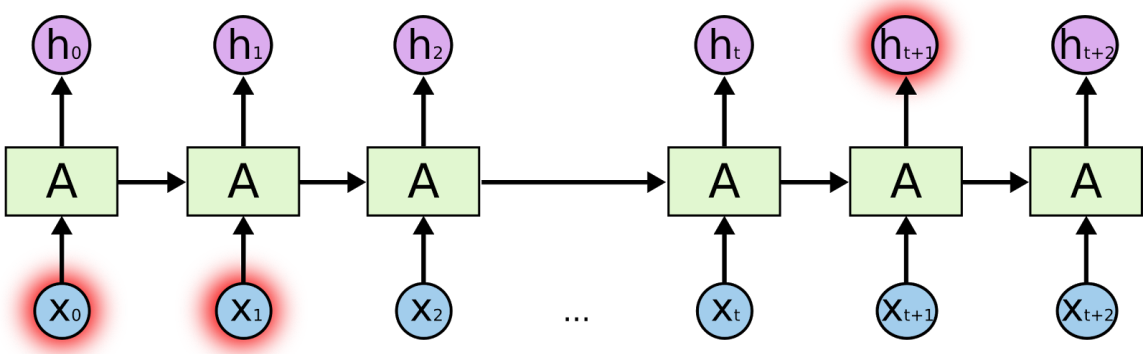
Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. Ví dụ, ta có câu: "các đám mây trên bầu trời" thì ta chỉ cần đọc tới "các đám mây trên bầu" là đủ biết được chữ tiếp theo là "trời" rồi. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.



Hình 1.3: Mô hình duỗi thẳng của mạng RNN

Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, dự đoán chữ cuối cùng trong đoạn: "I grew up in France… I speak fluently French.". Rõ ràng là các thông tin gần ("I speak fluently") chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì. Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh "I grew up in France" nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi.

Thật không may là với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa.



Hình 1.4: Vấn đề phụ thuộc xa của mạng RNN

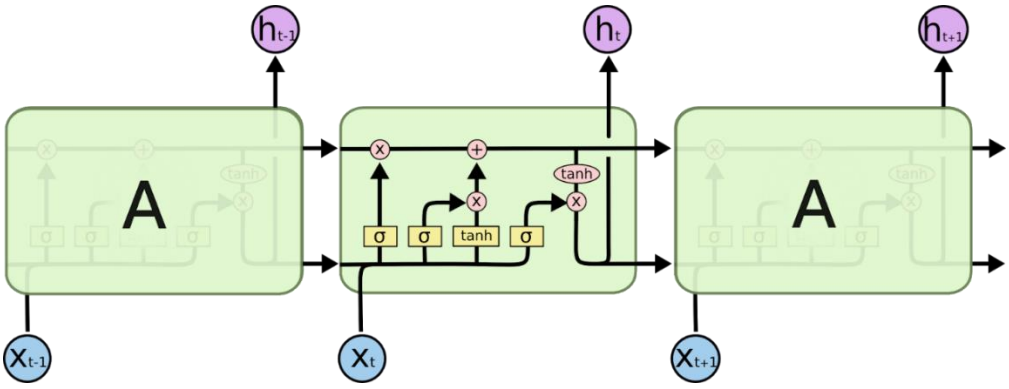
1.3.3. Mạng LSTM (Long Short Term Memory Networks)

Mạng Long Short Term Memory, thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay. [8]

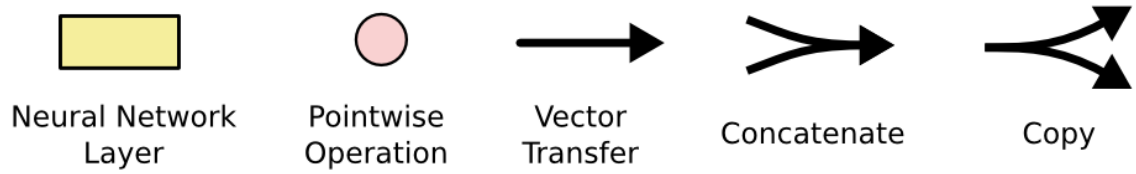
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-đun này có cấu trúc rất đơn giản, thường là một tầng tanh.

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



Hình 1.5: Kiến trúc mạng LSTM



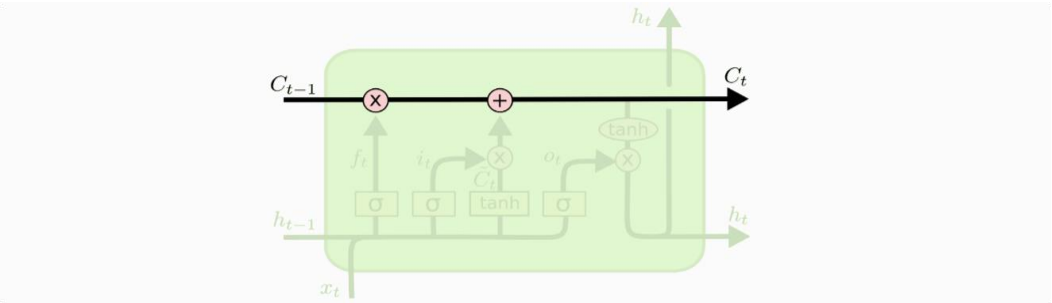
Hình 1.6: Các kí hiệu trong kiến trúc mạng LSTM

Trong sơ đồ trên ta có:

- Hình chữ nhật màu vàng biểu thị các tầng trong Mạng LSTM
- Hình tròn màu hồng thể hiện các phép toán được thực hiện
- Các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và di chuyển đến các nơi khác nhau
- Các đường hợp nhau kí hiệu cho việc kết hợp

Chìa khóa của LSTM là trạng thái tế bào (cell state) – chính là đường chạy thông ngang phía trên của sơ đồ hình vẽ.

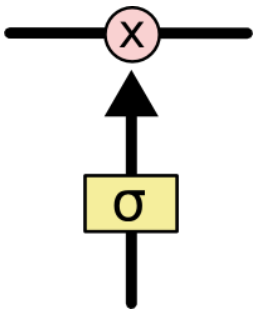
Trạng thái tế bào là một dạng giống như băng chuyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



Hình 1.7. Trạng thái tế bào của LSTM

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 đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



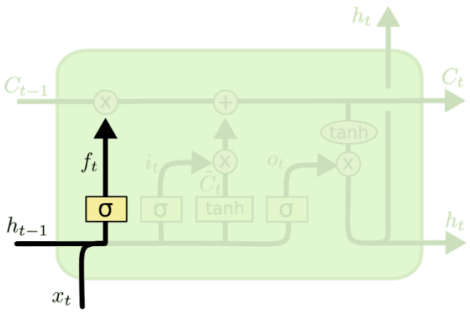
Hình 1.8: Cấu trúc cổng trong LSTM

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 nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Bên trong LSTM:

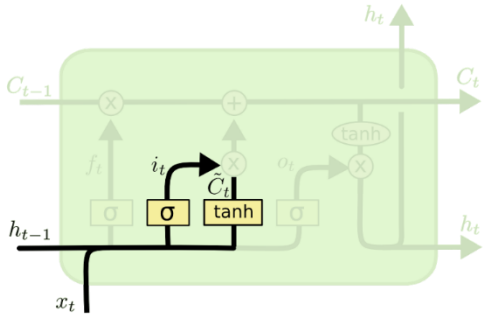
- Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là "tầng cổng quên" (forget gate layer). Nó sẽ lấy đầu vào là ht-1 và xt rồi đưa ra kết quả là một số trong khoảng [0, 1] cho mỗi số trong trạng thái tế bào Ct-1. Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hình 1.9: Tầng cổng quên của LSTM

- Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là "tầng cổng vào" (input gate layer) để quyết định giá trị nào ta sẽ cập nhật. Tiếp theo là một tầng tanh tạo ra một véc-tơ cho giá trị mới C_t nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhật cho trạng thái.

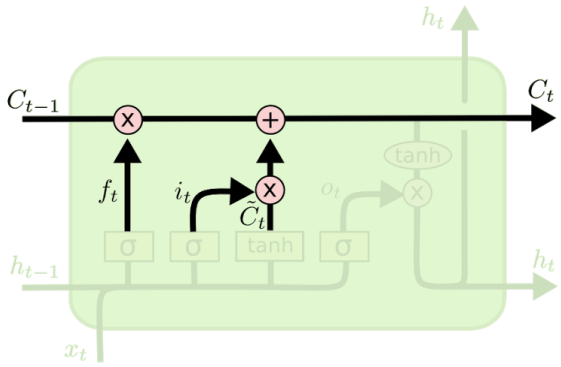


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 1.10: Tầng cổng vào

Giờ là lúc cập nhập trạng thái tế bào cũ $\mathbf{C_{t-1}}$ thành trạng thái mới $\mathbf{C_t}$. Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong.

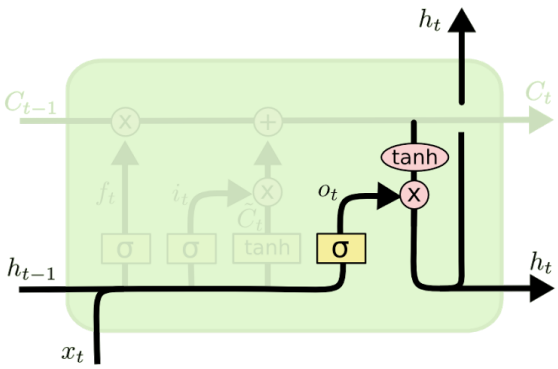
Ta sẽ nhân trạng thái cũ với $\mathbf{f_t}$ để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm $\mathbf{i_t * C_t}$. Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhật mỗi giá trị trạng thái ra sao.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Hình 1.11: Trạng thái tế bào thu được sau khi đi qua cổng quên và cổng vào

Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm tanh để co giá trị nó về khoảng [-1, 1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Hình 1.12: Kết quả đầu ra và trạng thái tế bào của LSTM

1.3.4. Word Embedding

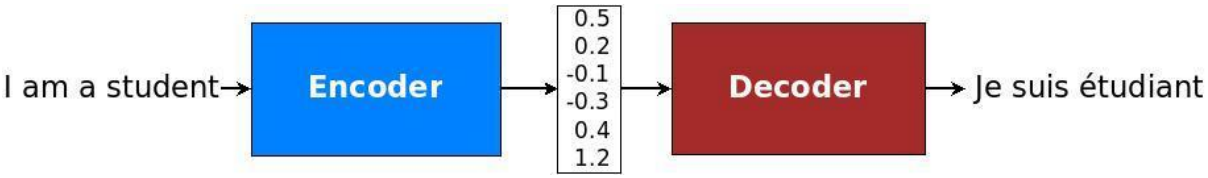
Word Embedding là kỹ thuật chuyển đổi dữ liệu dạng text sang dữ liệu dạng số và có thể có các cách biểu diễn khác nhau với cùng một văn bản.

Nhiều thuật toán học máy và hầu hết các kiến trúc Deep Learning không thể xử lý các xâu, chuỗi ký tự hay các dữ liệu văn bản thô mà chỉ có thể sử dụng số thực để làm đầu vào cho các bài toán này.

Word Embedding được sử dụng để ánh xạ các từ hay các cụm từ từ một bộ từ vựng thành một vector tương ứng của các số thực. Vector này được biểu diễn với không gian chiều ít hơn và được sử dụng để phân tích cú pháp ngữ nghĩa, để trích xuất văn bản và có khả năng hiểu ngôn ngữ tự nhiên. Vì thế Word Embedding chính là việc xây dựng được một vector đại diện với số chiều nhỏ đồng thời bảo toàn được ngữ nghĩa của từ theo ngữ cảnh. [9]

1.3.5. Mô hình sinh chuỗi Sequence to Sequence

Sequence to sequence (Seq2seq) là một mô hình học máy dùng để chuyển đổi các câu từ miền này sang một miền khác. Nó được ứng dụng thành công trong nhiều loại công việc khác nhau như dịch máy, nhận dạng giọng nói, tóm tắt văn bản,… Thông thường nó ứng dụng được bất cứ khi nào cần sinh văn bản. Mô hình seq2seq cơ bản gồm 2 mạng nơ-ron hồi quy: RNN Encoder và RNN Decoder. [10]



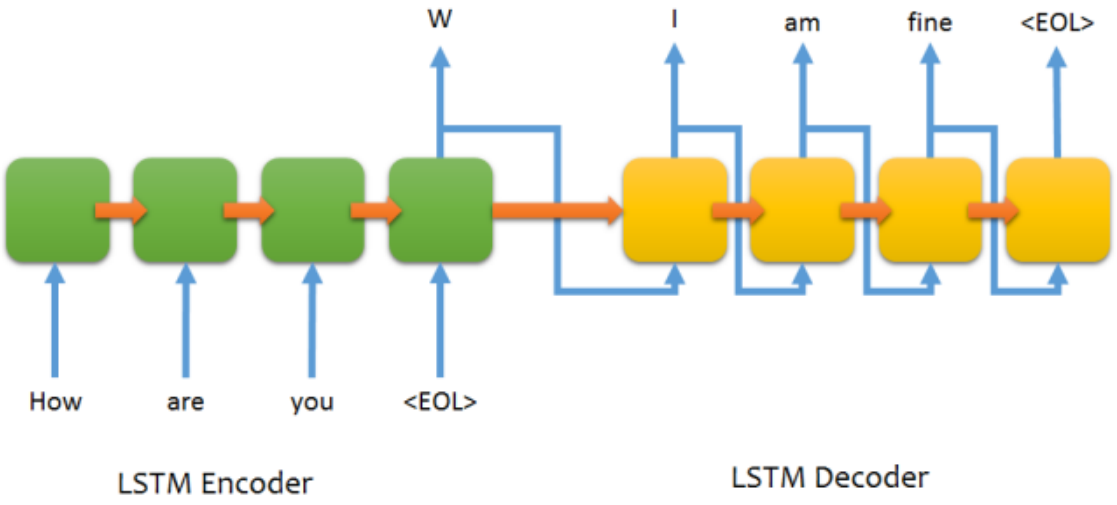
Hình 1.13: Cấu trúc mô hình seq2seq

Hình ảnh 1.13 minh họa mô hình Seq2seq áp dụng trong bài toán dịch máy từ tiếng Anh sang tiếng Pháp: một bộ Encoder chuyển đổi câu nguồn thành một vector

mang ngữ nghĩa sau đó sử dụng vector này để đi qua bộ Decoder để sinh ra một câu đã được dịch sang ngôn ngữ đích.

Encoder đưa một chuỗi (một câu) dưới dạng đầu vào và xử lý từng word tại mỗi timestep. Mục đích của nó à chuyển đổi một chuỗi các từ thành một vector đặc trưng mã hóa các thông tin quan trọng trong chuỗi đồng thời loại bỏ các thông tin không cần thiết. Mỗi trạng thái ẩn ảnh hưởng trực tiếp đến trạng thái ẩn tiếp theo, và trạng thái ẩn cuối cùng của Encoder được coi như là một sự tóm tắt ngữ nghĩa của chuỗi đầu vào. Trạng thái đó gọi là vector ngữ cảnh hay vector "suy nghĩ", bởi nó đại diện cho ý định, ngữ cảnh của chuỗi đầu vào.

Từ thông tin ngữ cảnh của Encoder, Decoder sinh ra một chuỗi khác, mỗi word tại mỗi timestep. Tại mỗi timestep, Decoder bị ảnh hưởng bởi vector ngữ cảnh và từ được sinh ra đằng trước nó cho đến khi gặp từ kết thúc trong câu.



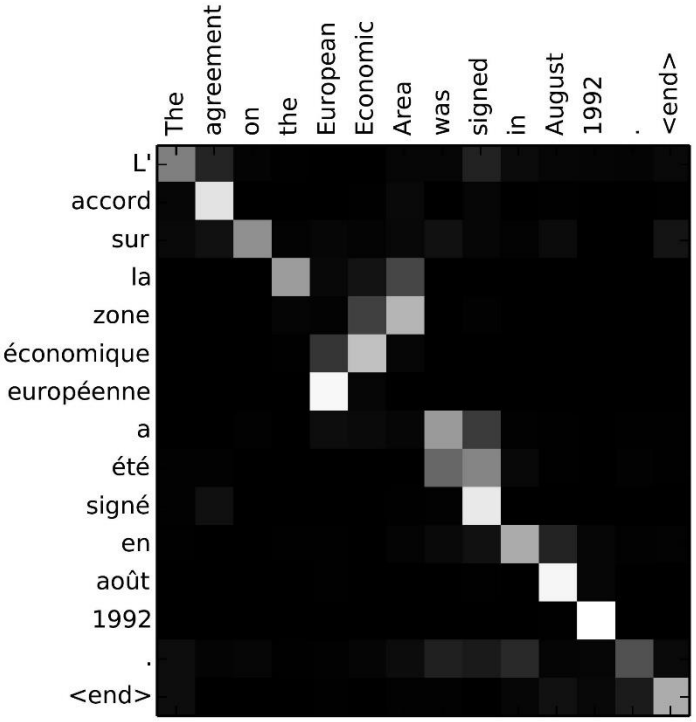
Hình 1.14: Minh họa mô hình seq2seq sử dụng 2 mạng neural LSTM cho thành phần encoder và decoder.

Mô hình seq2seq cơ bản có nhược điểm là yêu cầu RNN Decoder sử dụng toàn bộ thông tin mã hóa từ chuỗi đầu vào cho dù chuỗi đó dài hay ngắn. Thứ hai, RNN Encoder cần phải mã hóa chuỗi đầu vào thành một vector duy nhất và có độ dài cố định. Ràng buộc này không thực sự hiệu quả vì trong thức tế, việc sinh ra từ tại một bước timestep trong chuỗi đầu ra có khi phụ thuộc nhiều hơn vào một số những thành phần nhất định trong chuỗi đầu vào. Ví dụ, khi dịch một câu từ ngôn ngữ này sang ngôn ngữ khác, chúng ta thường quan tâm nhiều đến ngữ cảnh xung quanh của từ hiện tại so với các từ khác trong câu. Kỹ thuật Attention được đưa ra để giải quyết vấn đề đó. [6]

1.3.6. Attention Sequence to sequence

Một trong những hạn chế của seq2seq là toàn bộ thông tin trong chuỗi đầu vào được mã hóa thành một vector có độ dài cố định , gọi là ngữ cảnh. Khi độ dài của chuỗi lớn hơn, thì chúng bắt đầu mất một lượng thông tin đáng kể. Đó là lý do vì sao mô hình seq2seq cơ bản không hoạt động tốt trong việc giải mã các chuỗi có kích thước lớn và kỹ thuật Attention là kỹ thuật cho phép có thể học hiệu quả mô hình sinh khắc phục nhược điểm của seq2seq cơ bản. Hệ thống dịch máy của Google - Google Translate hiện đang áp dụng mô hình seq2seq với kỹ thuật attention và cho chất lượng vượt trội so với những phương pháp trước kia. [6]

Ý tưởng chính củ cơ chế Attention là thiết lập các kết nối trực tiếp giữa câu nguồn và câu đích bằng cách chú ý đến sự liên quan nội dụng câu nguồn như khi chúng ta dịch. Một ma trận giữa câu nguồn và câu đích thể hiện cơ chế attention được mô tả như hình 1.15. [11]



Hình 1.15: Attention visualization – ví dụ sự sắp xếp giữa các câu nguồn và câu đích [11]

Hình 1.15 biểu diễn ma trận trọng số và sự sắp xếp thứ tự của câu nguồn và câu đích trong bài toán dịch máy. Mỗi từ của câu đích được biểu diễn bằng vector trọng số - mức độ liên quan của nó đến từng từ hay cụm từ trong câu nguồn. Từ hình trên ta có thể thấy những vị trí ma trận có màu sáng nhất trong mỗi hàng biểu diễn cho sự tương đồng ngữ nghĩa và từ phù hợp tương ứng với từ hay cụm từ của câu nguồn với câu đích.

Phần mô tả Attention cụ thể được trình bày trong phần 2, mục 2.3.3.

1.3.7. Hàm Softmax và loss

Softmax Regression là một phương pháp được sử dụng rộng rãi như một phương pháp phân lớp khắc phục hạn chế về tổng các xác suất khi áp dụng kỹ thuật one-vs-rest với bài toán phân loại 2 lớp cho bài toán phân loại đa lớp.

Hàm softmax nhận đầu vào là một vector và đầu ra là một vector có cùng số chiều a: $\mathbf{R^n} \mapsto \mathbf{R^n}$

$$\boldsymbol{a_i} = \frac{\mathbf{exp}(z_i)}{\sum_j \mathbf{exp}(z_j)}, \text{ với } 0 \leq \mathbf{a_i} \leq 1 \text{ và } \sum_i a_i = 1$$

Hàm softmax này tính xác suất để một input x rơi vào một nhãn i. $\mathbf{a_i}$ càng lớn thì xác suất để một input x thuộc về nhãn i càng cao. Trong đó z được tính theo công thức $\mathbf{z_i = W_i x_i}$, với W là trọng số đối với phần tử $\mathbf{x_i}$ của input x. Lúc này có thể giả sử rằng:

$$\mathbf{P(y_k = i \mid x_k ; W) = a_i}$$

Trong đó P(y = i | x; W) được hiểu là xác suất để một điểm dữ liệu x rơi vào class thứ i nếu biết tham số mô hình và ma trận trọng số W. [12]

Sau khi dự đoán cần tiến hành so khớp để tối ưu lại tham số. Quá trình này sử dụng hàm mất mát là Cross entropy. Cross entropy giữa hai phân phối p và q rời rạc được viết dưới dạng:

$$H(p, q) = -\sum_{i=1}^{C} p_i \, log \, q_i$$

Hàm Cross entropy sẽ nhận giá trị nhỏ nhất khi p = q. Mặt khác, hàm này nhận giá trị rất cao (tức loss rất cao) khi p ở xa q. Về mặt tối ưu hàm cross entropy sẽ cho nghiệm gần với p hơn vì những nghiệm ở xa bị phạt rất nặng. Tính chất này khiến cho cross entropy được sử dụng rộng rãi khi tính khoảng cách giữa hai phân phối xác suất.

1.3.8. Một vài kỹ thuật tối ưu hóa hàm mất mát

Trong Machine Learning nói riêng và toán tối ưu nói chung, chúng ta thường xuyên phải đi tìm giá trị nhỏ nhất (hoặc lớn nhất) của một hàm số nào đó. Việc tìm điểm cực tiểu toàn cục của các hàm mất mát trong ML rất phức tạp, thậm chí là bất khả thi. Thay vào đó, người ta thường cố gắng tìm các điểm cực tiểu cục bộ và ở một mức độ nào đó, coi nó là nghiệm cần tìm của bài toán.

Các điểm cực tiểu cục bộ là nghiệm của phương trình đạo hàm bằng 0. Tuy nhiên trong hầu hết các trường hợp, việc giải phương trình đạo hàm bằng 0 là bất khả thi, có thể do sự phức tạp của dạng đạo hàm, hay các điểm dữ liệu có số chiều lớn hay có quá nhiều điểm dữ liệu. Do đó hướng tiếp cận phổ biến nhất là xuất phát từ một điểm được coi là gần với nghiệm của bài toán, sau đó dùng một phép lặp để

tiến dần đến điểm cần tìm, đến khi đạo hàm gần với 0. Gradient Descent (GD) và các biến thể của nó được sử dụng nhiều nhất. [13]

Gradient Descent cho hàm nhiều biến: Giả sử cần tìm điểm cực tiểu toàn cục cho hàm f(θ) trong đó θ là một vector các tham số cần tối ưu. Đạo hàm của số đó tại một điểm θ bất kỳ được kí hiệu là $\nabla_\theta f(\theta)$. Thuật toán GD bắt đầu dự đoán tại θ_0, sau đó ở vòng lặp thứ t, quy tắc cập nhật là

$$\boldsymbol{\theta_{t+1} = \theta_t - \eta \nabla_\theta f(\theta)}$$

Với η là learning rate – tốc độ học, dấu trừ thể hiện việc cập nhật trọng số phải đi ngược với đạo hàm, hướng về vị trí đạo hàm bằng 0. [13]

Stochastic Gradient Descent là một biến thể của GD. Trong thuật toán này, tại một thời điểm, ta chỉ tính đạo hàm của mất mát trên một điểm dữ liệu x_i rồi cập nhật θ dựa trên đạo hàm này. Việc này được thực hiện với từng điểm trên toàn bộ dữ liệu và sau đó lặp lại quá trình trên.

Mỗi lần duyệt một lượt qua tất cả các điểm trên toàn bộ dữ liệu được gọi là một epoch. Với GD thông thường, mỗi epoch ứng với một lần cập nhật θ, còn với SGD thì mỗi epoch ứng với N lần cập nhật θ với N là số điểm dữ liệu. Việc cập nhật từng điểm có thể làm giảm tốc độ thực hiện một epoch nhưng SGD chỉ yêu cầu một lượng epoch rất nhỏ, vì vậy SGD phù hợp với các bài toán có lượng dữ liệu lớn như Deep Learning và các bài toán yêu cầu mô hình thay đổi liên tục – online learning. Thứ tự lựa chọn điểm dữ liệu ảnh hưởng tới hiệu năng của SGD do sau mỗi epoch chúng ta cần xáo trộn thứ tự các dữ liệu để đảm bảo tính ngẫu nhiên. Quy tắc cập nhật của SGD là

$$\boldsymbol{\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta; \mathbf{x}_i, \mathbf{y}_i)}$$

Với $\mathbf{J(\theta, x_i, y_i)}$ là hàm mất mát với chỉ một cặp điểm dữ liệu (input, label) là $\mathbf{(x_i, y_i)}$. [14]

Adam Optimizer là thuật toán mở rộng của SGD được áp dụng rộng rãi trong các ứng dụng tring DL, thị giác máy tính và xử lý ngôn ngữ tự nhiên. Thuật toán Adam cập nhật các trọng số dựa trên việc lặp trong quá trình huấn luyện.

Ưu điểm của thuật toán Adam bao gồm tính toán hiệu quả, yêu cầu ít bộ nhớ, khá phù hợp với các vấn đề nhiều dữ liệu hoặc tham số, …

Adam khác với SGD ở chỗ SGD thì duy trì một tốc độ học để cập nhật tất cả các trọng số và tốc độ học này không thay đổi trong quá trình huấn luyện, còn tốc độ học trong Adam được duy trì cho mỗi trọng số mạng và được điều chỉnh riêng biệt trong quá trình học. Adam kết hợp các ưu điểm của hai thuật toán mở rộng khác của SGD cụ thể là Adaptive Gradient Algorithm (AdaGrad) và Root Mean Square Propagation (RMSProp).

AdaGrad duy trì tốc độ học trên mỗi tham số nhằm cải thiện hiệu năng của các bài toán với gradient thưa như xử lý ngôn ngữ tự nhiên và thị giác máy tính. Còn RMSProp thì duy trì tốc độ học trên mỗi tham số được điều chỉnh dựa trên mức trung bình của các độ lớn trọng số gần đây của gradient. Điều này giúp thuật toán hoạt động tốt trên các bài toán trực tuyến và không cố định.

Thay vì thích nghi với tốc độ học dựa trên thời điểm trung bình đầu tiên như trong RMSProp, Adam còn sử dụng trung bình của thời điểm thứ hai của gradient.

Adam là một thuật toán phổ biến trong lĩnh vực Deep Learning vì nó đạt được kết quả tốt một cách nhanh chóng.

Các tham số của Adam:

- alpha: còn được gọi là tốc độ học hoặc step size, tỉ lệ mà trọng số được cập nhật. (Ví dụ: 0.9)
- beta1: tỉ lệ phân rã theo cấp số nhân cho các ước tính thời điểm đầu tiên
- beta2: tỉ lệ phân rã theo cấp số mũ cho các ước tính thời điểm thứ 2. Ví dụ 0.999. giá trị này nên được đặt gần với 1 trong các bài toán có gradient thưa.
- epsilon: là một số rất nhỏ để ngăn chặn bất kì việc chia cho 0 trong quá trình thực thi.

Một số tham số mặc định tốt cho các bài toán ML alpha=0.001, beta1=0.9, beta2=0.999 và epsilon=1e-8.

PHẦN 2: CÁC KẾT QUẢ ĐẠT ĐƯỢC

2.1. Các thư viện, tài liệu tham khảo được sử dụng

2.1.1. Ngôn ngữ lập trình sử dụng

Trong phần cài đặt mô hình em sử dụng ngôn ngữ lập trình Python phiên bản 2.7 cho mô hình. Python là một ngôn ngữ lập trình hướng đối tượng cấp cao, mạnh mẽ, được xây dựng bởi Guido van Rosum vào năm 1990. Ưu điểm của Python:

- Là ngôn ngữ lập trình đơn giản, dễ học
- Cấu trúc động, sử dụng cơ chế cấp phát bộ nhớ động
- Miễn phí, mã nguồn mở
- Linh hoạt trên các nền tảng Windows, Linux, macOS
- Là ngôn ngữ thông dịch cấp cao
- Được tích hợp nhiều công cụ và có thư viện chuẩn phong phú, cộng đồng người lập trình Python lớn.

2.1.2. TensorFlow

TensorFlow là một thư viện phần mềm mã nguồn mở dành cho Học Máy trong nhiều loại hình tác vụ phân loại và hiểu ngôn ngữ tự nhiên. Nó đang được sử dụng cho cả nghiên cứu lẫn ứng dụng thực tế trong các sản phẩm thương mại của Google như nhận dạng giọng nói, Gmail, Google Photos và Google Search,…

TensorFlow nguyên thủy được phát triển bởi đội Google Brain cho mục đích nghiên cứu và sản xuất sau đó được phát hành theo giấy phép mã nguồn mở vào tháng 11 năm 2015. Với TensorFlow, Học Máy đang trở nên gần gũi và dễ dàng tiếp cận hơn cho các lập trình viên hiện nay với các đặc điểm:

- Hỗ trợ GPU
- Hỗ trợ huấn luyện phân tán
- Hiệu suất cao do core là C++
- Hỗ trợ các ngôn ngữ Python, C++…

2.1.3. Underthesea

Underthesea là một toolkit hỗ trợ việc nghiên cứu và phát triển xử lý ngôn ngữ tiếng Việt ra đời vào tháng 3 năm 2017.

Ưu điểm:

- Mã nguồn mở
- Dễ dàng cài đặt và sử dụng
- Document đầy đủ giúp tra cứu một cách nhanh chóng

2.1.4. Flask app

Flask là một micro web framework khá nổi tiếng được viết bằng Python. Armin Ronacher, người dẫn đầu nhóm Pocco – nhóm những người đam mê Python trên thế giới, đã phát triển nó. Flask dựa trên bộ công cụ Werkzeug WSGI và Jinja2. Cả hai đều là dự án của Pocco.

Điểm mạnh của Flask là sự nhỏ gọn, đơn giản, linh hoạt. Nó giúp cho việc tạo trang web một cách nhanh chóng và dễ dàng.

2.1.5. Một số thư viện khác

Đồ án này em có sử dụng một số thư viện khác trong Python như: numpy, pickle, re, os, sys,… trong quá trình cài đặt mô hình.

2.2. Tập dữ liệu

Tập dữ liệu kích thước 920KB bao gồm 5347 cặp đối thoại trong lĩnh vực thời trang đã được gán nhãn thực thể được cung cấp bởi công ty Rabiloo với kích thước bộ từ vựng là 1671. Tập dữ liệu được chia làm 3 phần theo tỉ lệ 70% train, 15% test, 15% validation.

2.2.1. Thông tin dữ liệu/ Định dạng dữ liệu

Dữ liệu dạng thô là các file **txt** chứa các đoạn hội thoại hỏi-đáp giữa người bán và người mua như dưới đây:

```
1  :  <B-REFE> <I-REFE> còn không bạn , cho mình xin giá nhé.

0  :  tớ còn đủ <B-SIZE> <B-SIZE> nàng ơi , <B-TYPE> <B-PRIC> nhé.
```

Trong đó:

- Câu đầu tiên được đánh số 1 là câu mà người mua hỏi người bán
- Câu sau đánh số 0 là câu mà người bán trả lời lại người mua
- <B-REFE>, <I-REFE>, <B-SIZE>, <B-TYPE>, <B-PRIC> là các nhãn thực thể. Thông tin về các nhãn được mô tả cụ thể trong bảng dưới đây:

STT	Ký hiệu	Tên nhãn	Ví dụ minh họa	Ghi chú
1	COLO	Màu sắc	Đỏ, Xanh, Hồng, ...	

2	MATE	Chất liệu sản phẩm	Len, Lụa, ...	
3	SIZE	Kích thước	S, M, L, XL, ...	
4	PRIC	Giá tiền	100K, 100.000, ...	
5	GEND	Giới tính	Nam, nữ	
6	ORIG	Xuất xứ	VNXK, Nhật, Quảng Châu, ...	
7	TRADE	Thương hiệu	Canifa, Gucci, H&M, ...	
8	TYPE	Loại	Áo, quần, quần bò, áo dài, áo khoác, ...	
9	LOC	Vị trí	Hai Bà Trưng, Hà Nội, ...	
10	SAOF	Giảm giá	50%, 80%, ...	
11	CURU	Đơn vị tiền tệ	VNĐ, ...	
12	SHME	Phương thức vận chuyển	COD, Chuyển phát nhanh, ...	
13	TIME	Thời gian	4h, 4h30, 4 giờ, 4 giờ 30, ...	
14	REFE	Tham chiếu	Áo này, quần này, áo kia, quần kia, ...	Sử dụng nhãn này để tham chiếu đến các đối tượng mà

				người dùng đã nói trước đó
15	WEIG	Cân nặng	50kg, 75kg, ...	
16	HEIG	Chiều cao	1m6, 1m83, ...	
17	DIGIT	Số	123, 5, …	
18	CODE	Từ chứa cả số và chữ cái mà không phải các nhãn ở trên	HM123, …	

Bảng 2: Bảng danh sách nhãn thực thể NER

2.2.2. Xử lý dữ liệu

Quá trình tiền xử lý dữ liệu thô đã gán nhãn NER được thực hiện như sau:

- Bước 1: Dữ liệu dạng thô được loại bỏ các kí tự đặc biệt, các dấu khoảng trắng thừa
- Bước 2: Dữ liệu đã được xử lý ở bước 1 được chuyển đổi nhãn NER từ dạng <B-XXXX> thành XXXX và loại bỏ các nhãn <I-XXXX>
- Bước 3: Tiến hành tách từ tiếng Việt sử dụng Word_tokenizer của underthesea

Quá trình xây dựng bộ từ vựng và chuẩn hóa dữ liệu học

- Bước 1: Đọc từng file dữ liệu hội thoại và phân chia thành hai tập, tập các câu hỏi và tập các câu trả lời tương ứng sử dụng nhãn 0 và nhãn 1 trước mỗi câu.
- Bước 2: Tiến hành tiền xử lý dữ liệu tập các câu hỏi và tập các câu trả lời và lọc bỏ những cặp hỏi-đáp có câu hỏi hoặc câu trả lời có độ dài quá 50 từ.
- Bước 3: Xây dựng bộ từ vựng từ tập dữ liệu từ dữ liệu đã xử lý ở bước 2 và lưu lại thành file ***vocabulary.pkl***. Bộ từ vựng này cần có thêm 2 từ
  "UNK" - sử dụng để đánh dấu những từ mới không có trong bộ từ vựng
  "_" – kí tự kết thúc câu
- Bước 4: Xây dựng 2 tập ***index2word*** và ***word2index*** nhằm chuyển đổi dữ liệu dạng text sang dạng số và làm đầu vào cho quá trình huấn luyện và dự đoán và được lưu lại trong file ***metadata.pkl.***
- Bước 5: Từ các bộ dữ liệu xây dựng được từ bước 4, chuyển đổi các câu hỏi và câu trả lời sang dạng mảng với mỗi phần tử là số thứ tự trong bộ từ vựng vocabulary.pkl. Sau đó mỗi câu trả lời và các câu hỏi được padding 0 với độ dài 50 và lưu lại thành 2 file **npy** tương ứng với tập câu hỏi và tập câu trả lời.

Với câu hỏi "TYPE giá bao nhiêu ạ?" được tiền xử lý thu được "TYPE giá bao_nhiêu ạ", giả sử chuyển đổi câu này sử dụng word2index thu được mảng có giá trị [3, 15, 83, 916]. Sau đó được padding với 0 thu được một mảng có kích thước 50 với 4 phần tử đầu là [3, 15, 83, 916] và 46 phần tử tiếp theo có giá trị 0. Mô hình Seq2seq không xử lý được các câu có độ dài thay đổi do đó cần cố định độ dài câu hỏi và câu trả lời sử dụng kỹ thuật padding.

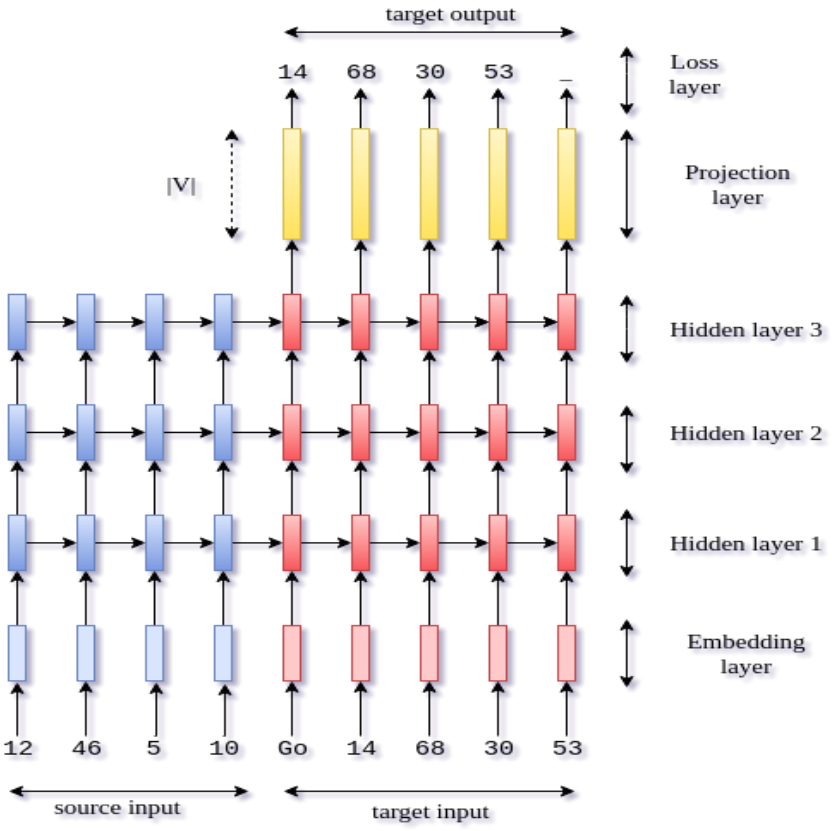
Quá trình hậu xử lý dữ liệu

Loại bỏ các từ lặp có vị trí gần nhau trong câu trả lời sinh ra bởi chatbot

2.3. Áp dụng seq2seq/ attention seq2seq vào bài toán Chatbot

2.3.1. Quá trình huấn luyện

Mô hình sử dụng 2 mạng LSTM cho Encoder và Decoder. Tại mỗi bước huấn luyện, mô hình lấy ngẫu nhiên batch_size=32 câu trong tập câu hỏi và câu trả lời của tập train sử dụng để huấn luyện. Mỗi lần huấn luyện, dữ liệu được đi qua một tầng embedding, 3 tầng ẩn (hoặc 1 tầng ẩn), một tầng projection và một tầng loss.



Hình 2.1: Quá trình huấn luyện mô hình Seq2seq

Đầu tiên, mỗi từ trong dữ liệu câu nguồn và câu đích được đi qua tầng embedding để chuyển đổi thành các vector embedding mang ý nghĩa ngữ cảnh của mỗi từ.

Sau đó các vector embedding này lần lượt được đưa vào LSTM tại tầng ẩn đầu tiên. LSTM tổng hợp các thông tin đi qua nó tại mỗi từ t, sử dụng cổng quên để xác định thông tin nào được giữ lại, thông tin nào cần loại bỏ, sau đó trả về 2 giá trị, một là output tại từ t $\mathbf{o_t}$ sẽ được sử dụng làm đầu vào cho tầng tiếp theo, và một là trạng thái ẩn được sử dụng làm đầu vào ở từ thứ t+1 để tiếp tục tổng hợp thông tin.

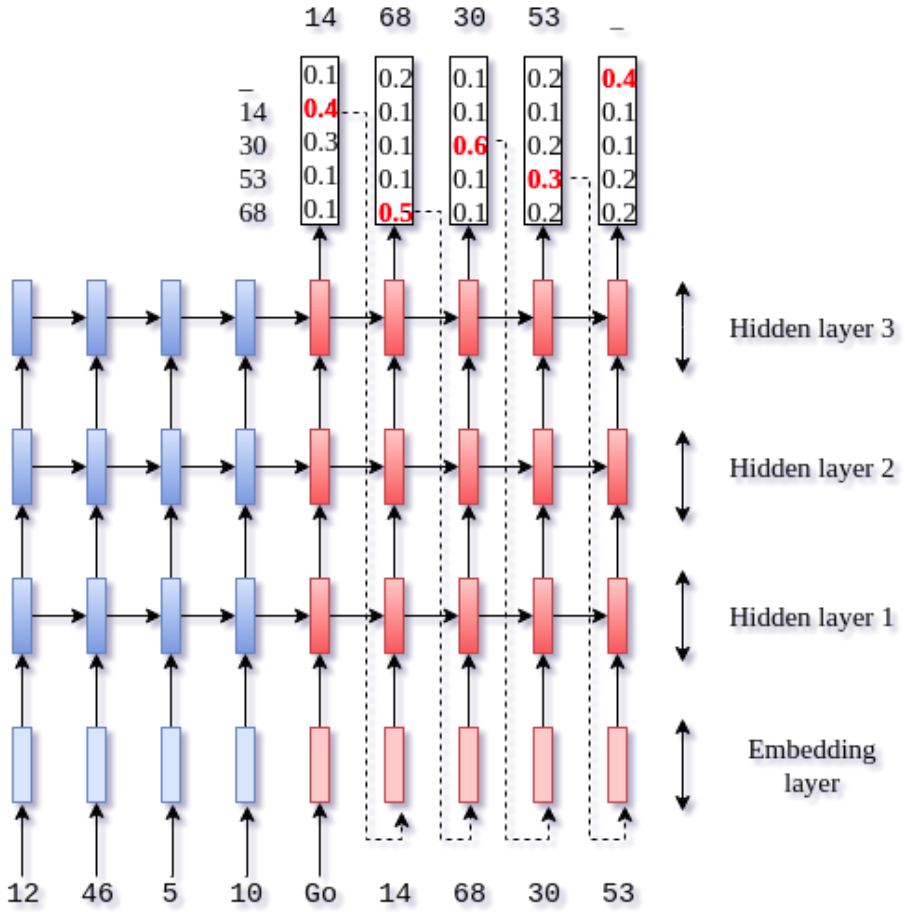
Khi gặp biểu tượng "**Go"** - start of decoder – thì vector cuối cùng đi qua LSTM Encoder trả về một vector mang ý nghĩa chuỗi câu nguồn của đầu vào và được đưa vào LSTM Decoder kết hợp cùng với các vector embedding của chuỗi dữ liệu câu đích để huấn luyện mô hình, tổng hợp thông tin của câu đích.

Các tầng ẩn tiếp theo tương tự như tầng ẩn đầu tiên nhưng đầu vào tại mỗi timestep **t** không phải là vector embedding mà là vector output tại timestep **t** của tầng ẩn trước nó.

Kết thúc các tầng ẩn, các output tại tầng ẩn cuối cùng của bộ Decoder sẽ được đưa vào projection layer, một ma trận dày đặc, để biến đổi các 3D-vector trạng thái ẩn thành các logit vector **|V|** chiều. Với **|V|** là kích thước bộ từ vựng của dữ liệu, mỗi phần tử trong vector đại diện cho một từ trong bộ từ vựng.

Sau đó các vector logit này được đưa vào loss layer. Tại đây, các vector sẽ đi qua hàm softmax để dự đoán các từ tại output, sau đó tính toán lỗi sử dụng cross entropy và gradient descent để cực tiểu hóa lỗi. Cuối cùng, sau khi đã tính toán tối ưu tại loss layer, thông tin sẽ được lan truyền ngược để cập nhật lại bộ trọng số trước khi thực hiện lần huấn luyện tiếp theo.

2.3.2. Quá trình dự đoán, sinh ra câu trả lời



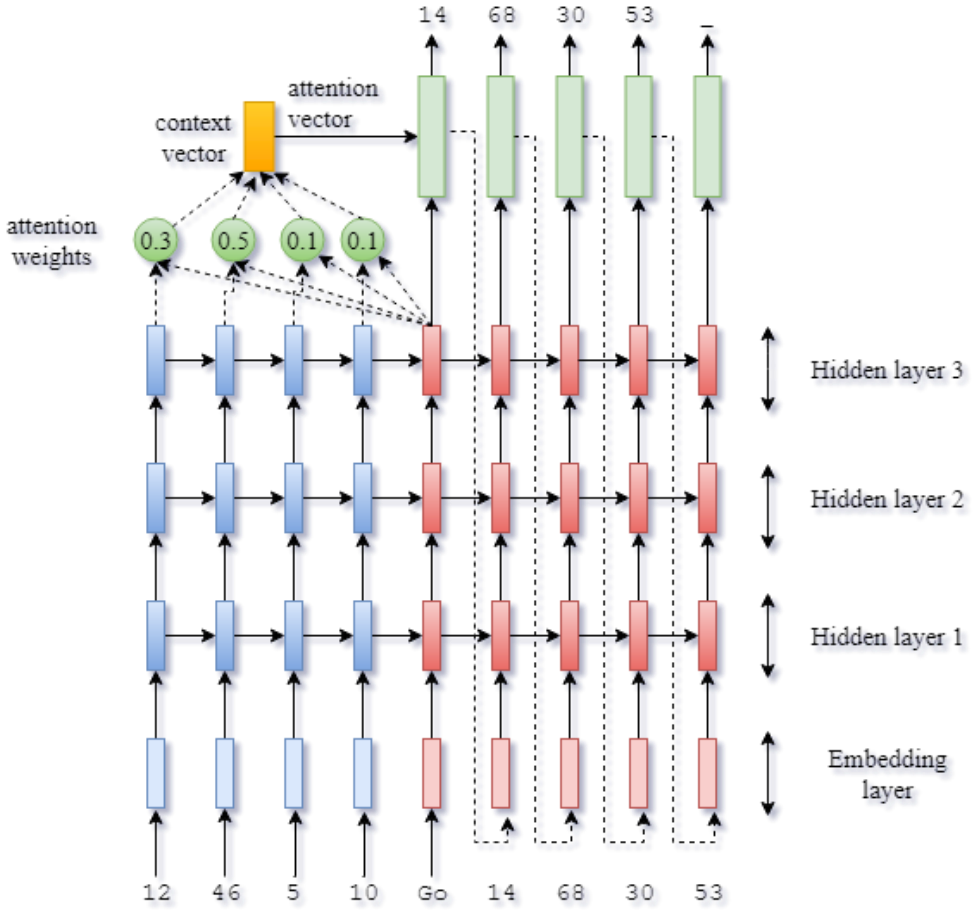
Hình 2.2. Quá trình dự đoán của mô hình Seq2seq

Đầu tiên, mô hình sẽ khôi phục lại session thông qua thông tin checkpoint được lưu lại trong quá trình huấn luyện và sử dụng session này để đưa ra câu trả lời.

Tại các tầng, LSTM Encoder thực hiện giống như quá trình huấn luyện.

Còn Decoder, tại tầng ẩn đầu tiên, Decoder sử dụng embedding vector của biểu tượng **Go** và kết hợp vector ngữ nghĩa học được bởi Encoder trong tầng đó để sinh ra một trạng thái ẩn và một output. Output này tiếp tục được sử dụng làm đầu vào cùng vector ngữ nghĩa tại tầng ẩn tiếp theo và sinh ra tương tự. Tại tầng ẩn cuối cùng, output từ biểu tượng **Go** ban đầu này và vector ngữ cảnh của Encoder được đưa vào projection layer để chuyển thành vector V chiều, sau đó đi qua softmax để dự đoán từ đầu tiên của câu trả lời với xác suất cao nhất. Từ mới được sinh ra này được coi như đầu vào của câu đích trong quá trình huấn luyện, tiếp tục được đi qua tầng embedding, các tầng ẩn, để dự đoán từ tiếp theo. Quá trình này lặp lại cho đến khi gặp **"_"** - end of sentence thì dừng và trả về kết quả đã dự đoán được.

2.3.3. Sử dụng mô hình attention seq2seq



Hình 2.3. Sử dụng mô hình attention seq2seq

Việc tính toán trọng số attention được xảy ra tại mỗi timestep Decoder bao gồm các giai đoạn sau: [10]

- Trạng thái ẩn mục tiêu hiện tại được so sánh với tất cả các trạng thái nguồn để thu được trọng số attention, được mô tả trong hình 2.3.

- Dựa trên các trọng số attention, chúng ta tính toán một vector ngữ cảnh bằng cách lấy trung bình trọng số của các trạng thái nguồn.

- Sau đó kết hợp vector ngữ cảnh với trạng thái ẩn mục tiêu hiện tại để thu được vector attention.

- Cuối cùng, vector attention này được sử dụng như đầu vào của timestep tiếp theo.

Ba bước đầu được tóm tắt bằng các biểu thức trong hình 2.4 dưới đây:

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \quad \text{[Attention weights]} \quad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \quad \text{[Context vector]} \quad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad \text{[Attention vector]} \quad (3)$$

Hình 2.4: Cơ chế tính toán attention

Hàm **score** được sử dụng để so sánh trạng thái ẩn mục tiêu $\mathbf{h_t}$ với mỗi trạng thái nguồn $\bar{\mathbf{h}}_\mathbf{s}$ và kết quả được chuẩn hóa để sinh ra các trọng số attention. Có nhiều cách chọn hàm **score** khác nhau, hàm phổ biến nhất được cho ở hình 2.5. Một khi đã được tính toán, vector attention $\mathbf{a_t}$ được sử dụng để lấy softmax logit và loss. Điều này tương tự với trạng thái ẩn mục tiêu tại tầng trên cùng trong mô hình seq2seq cơ bản. Hàm **f** cũng có thể được sử dụng công thức khác.

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{W} \bar{\boldsymbol{h}}_s & \text{[Luong's multiplicative style]} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_1} \boldsymbol{h}_t + \boldsymbol{W_2} \bar{\boldsymbol{h}}_s\right) & \text{[Bahdanau's additive style]} \end{cases} \quad (4)$$

Hình 2.5: Hai hàm score tính trọng số attention phổ biến

2.3.4. Áp dụng triển khai cài đặt

Đầu tiên em xây dựng một Class Seq2seq sử dụng để xây dựng graph, huấn luyện, đánh giá, bên cạnh đó lưu trữ và khôi phục mô hình đã huấn luyện được với các tham số khởi tạo:

- **xseq_len, yseq_len**: Độ dài của câu hỏi x và câu trả lời y. Ở đây các câu hỏi và câu trả lời được cố định độ dài là 50.
- **xvocab_size, yvocab_size**: Kích thước bộ từ vựng cho câu nguồn và câu đích. Với bài toán dịch máy, dịch từ ngôn ngữ này sang ngôn ngữ khác thì cần sử dụng 2 bộ từ vựng khác nhau còn với bài toán Chatbot này sử dụng chung một bộ từ vựng cho cả Encoder và Decoder.
- **emb_dim**: Số lượng units/nodes trong mỗi tầng ẩn và là số chiều của vector embedding trong phần cài đặt này.
- **num_layers**: Số tầng ẩn của mô hình.
- **ckpt_path**: Đường dẫn lưu mô hình đã huấn luyện được.
- **lr=0.0001**: tốc độ học được sử dụng để cực tiểu hóa hàm mất mát.
- **epochs=100000**: Số lượng bước lặp tối đa trong quá trình huấn luyện

Để xây dựng graph, cần có một lượng các trình giữ chỗ (placeholders) để đưa các dữ liệu vào bao gồm: encoder inputs, labels và decoder inputs. Labels có giá trị bằng chuỗi output thực tế của câu trả lời. Decoder inputs bắt đầu với một kí hiệu **Go** và các

phần tử tiếp theo chính là labels. Với mỗi câu đầu vào, cần tạo một danh sách các placeholders kiểu dữ liệu tf.int64 với chiều emb_dim x batch_size.

```
#  encoder inputs : list of indices of length xseq_len

self.enc_ip = [ tf.placeholder(shape=[None,],
                dtype=tf.int64,
                name='ei_{}'.format(t)) for t in range(xseq_len) ]
#  labels that represent the real outputs

self.labels = [ tf.placeholder(shape=[None,],
                dtype=tf.int64,
                name='ei_{}'.format(t)) for t in range(yseq_len) ]
#  decoder inputs : 'GO' + [ y1, y2, ... y_t-1 ]

self.dec_ip = [ tf.zeros_like(self.enc_ip[0], dtype=tf.int64, name='GO') ]
    + self.labels[:-1]
```

Tiếp theo, khởi tạo LSTM cell, phần quan trọng nhất của graph. Khởi tạo một placeholder cho keep_prob được sử dụng để điều khiển dropout – một kỹ thuật đơn giản tránh overfitting. Với một tầng với đầy đủ kết nối giữa các unit làm cho các nơ-ron phát triển đồng phụ thuộc lẫn nhau trong quá trình huấn luyện, làm hạn chế khả năng của nơ-ron dẫn đến overfitting trong việc huấn luyện. Do đó Dropout giúp giải quyết vấn đề này, tại mỗi giai đoạn huấn luyện, các node bị loại bỏ khỏi mạng với xác suất 1-p hoặc được giữ lại với xác suất p. Khi một nút bị loại bỏ thì node này và các kết nối của nó tới node khác sẽ không được xét đến trong một quá trình học tại thời điểm đó.

Sau đó định nghĩa một LSTM cell cơ bản và đặt nó vào trong Dropout Wrapper và xây dựng một stacked_lstm sử dụng MultiRNNCell. Sử dụng biến type_model để phân biệt các quá trình train, test và evaluation và biến using_attention để chỉ ra sử dụng mô hình có attention hay không.

```
# Basic LSTM cell wrapped in Dropout Wrapper

self.keep_prob = tf.placeholder(tf.float32)
# type
self.type_model = tf.placeholder(tf.int32)
# using attention
self.using_attention = tf.placeholder(tf. int32)
# define the basic cell
```

```
basic_cell = tf.nn.rnn_cell.DropoutWrapper(
        tf.nn.rnn_cell.BasicLSTMCell(emb_dim, state_is_tuple=True),
        output_keep_prob=self.keep_prob)
# stack cells together : n layered model


stacked_lstm = tf.nn.rnn_cell.MultiRNNCell([basic_cell]*num_layers, state_is_tu
ple=True)
```

Có hai mô hình tương ứng seq2seq và attention seq2seq được sử dụng trong đồ án này. Nếu using_attention !=1, hàm **embedding_rnn_seq2seq** được dùng để tạo model. Đầu tiên mô hình này nhúng encoder_input bằng một vector embedding mới có số chiều [kích thước bộ từ vựng x batch_size]. Sau đó nó thực thi một LSTM để mã hóa encoder_inputs nhúng này thành một vetor trạng thái. Tiếp theo, nó nhúng vector decoder_inputs bằng một vector khác có chiều tương tự và thực thi LSTM decoder, khởi tạo với trạng thái encoder cuối cùng trên decoder_inputs nhúng. Mô hình này có các tham số: [15]

- **encoder_inputs**: Một danh sách batch_size tensor 1D int32
- **decoder_inputs**: Một danh sách batch_size tensor 1D int32
- **cell**: chính là stacked_lstm
- **num_encoder_symbols**: kích thước bộ từ vựng encoder.
- **num_decoder_symbols**: kích thước bộ từ vựng decoder.
- **embedding_size**: Số chiều của vector embedding
- **output_projection**: là None hoặc một cặp (W, B), với W là bộ trọng số projection và B là bias của bộ trọng số đó. Nếu tồn tại output_projection và feed_previous=True thì mỗi output feed previous được trả về bằng việc nhân với bộ trọng số W và cộng thêm B.
- **feed_previous**: biến Boolean hoặc Boolean Tensor; Nếu True, chỉ trạng thái đầu tiên của decoder_inputs sẽ được sử dụng và tất cả các decoder_inputs khác được lấy từ outputs trước đó của nó. Còn nếu False, decoder_inputs được sử dụng như trường hợp Decoder cơ bản.

Hàm này trả về (outputs, state) với outputs là một danh sách cùng độ dài các tensor 2D như decoder_inputs và state là state của mỗi cell Decoder tại mỗi timestep. Còn nếu using_attention=1, ta dùng hàm **embedding_attention_seq2seq**. Đầu tiên mô hình này cũng khởi tạo một vector nhúng cho encoder_inputs, sau đó thực thi một LSTM mã hóa encoder_inputs nhúng này thành một vector trạng thái. Nó giữ vector trạng thái này tại mỗi bước để sử dụng cho việc chú ý sau này. Tiếp theo nó nhúng decoder_inputs giống như trên và thực thi attention decoder, khởi tạo với trạng thái encoder cuối cùng, trong decoder_inputs nhúng và chú ý đến các encoder outputs. Hàm này có các tham số tương tự như hàm **embedding _seq2seq** và có thêm 2 tham số num_heads biểu diễn số lượng attention heads được đọc từ attention_states và

initial_state_attention khởi tạo trạng thái attention bằng 0 nếu biến = False và khởi tạo từ trạng thái khởi tạo và trạng thái attention nếu biến = True. [16]

Tiếp theo, sử dụng một hàm bậc cao sequence_loss để tính độ mất mát. Hàm sequence_loss với đầu vào là decoder_ouputs và labels để tính độ mất mát sử dụng hàm softmax thưa với cross entropy. Sau đó xây dựng một tác vụ để tối thiểu hóa độ mất mát sử dụng Adam Optimizer với tốc độ học **lr=0.0001**.

```
loss_weights = [ tf.ones_like(label, dtype=tf.float32)
        for label in self.labels ]
self.loss = tf.nn.seq2seq.sequence_loss(self.decode_outputs,
        self.labels, loss_weights, yvocab_size)
self.train_op = tf.train.AdamOptimizer(learning_rate=lr).minimize(self.loss)
```

2.4. Cài đặt, đánh giá hiệu năng và kết quả của mô hình

2.4.1. Cài đặt và huấn luyện mô hình

2.4.1.1. Thông số của mô hình

Trong bài đồ án này, em huấn luyện mô hình với các thông số khác nhau như sau:

- Lượng number_unit trong mỗi tầng ẩn là 128 hoặc 512
- Mô hình được huấn luyện: Seq2seq cơ bản hoặc attention seq2seq
- Số tầng ẩn: 1 hoặc 3 tầng

Từ đó em huấn luyện mô hình thành 4 hệ thống con tương ứng với 4 bộ tham số có sự khác biệt như bảng dưới đây:

STT	Tên tóm tắt bộ tham số	Number units	Mô hình áp dụng	Số tầng ẩn
1	128_attention_3_layers	128	Attention Seq2seq	3
2	512_attention_3_layers	512	Attention Seq2seq	3
3	512_attention_1_layer	512	Attention Seq2seq	1
4	512_no_attention_3_layers	512	Seq2seq	3

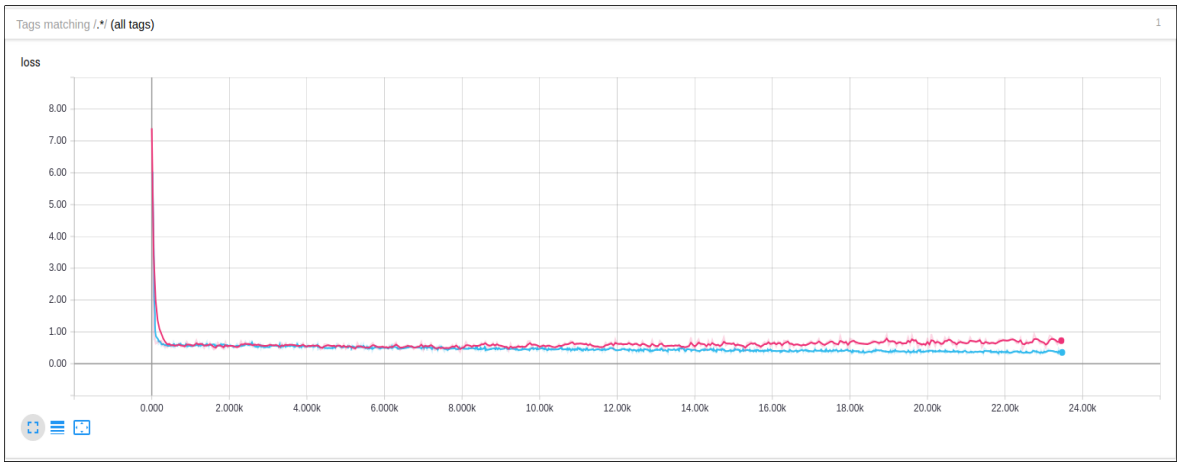
Bảng 3: Danh sách bộ tham số được huấn luyện

2.4.1.2. Quá trình huấn luyện

Tất cả các quá trình huấn luyện được thực hiện trên máy Intel(R) Core(TM) i7-3720 QM CPU @2.60GHz Ram 24GB.

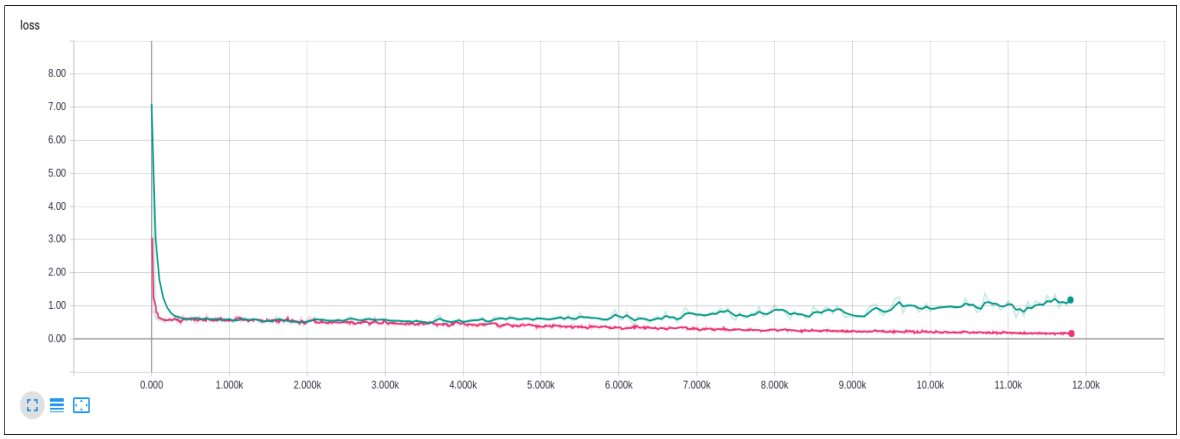
Trong quá trình huấn luyện kết hợp xây dựng đồ thị hàm mất mát của quá trình eval và train của mỗi bộ tham số. Từ đó tìm ra giá trị bước lặp thứ n, tại giá trị đó có loss eval là thấp nhất mà train là thấp vừa phải để tránh overfitting. Sử dụng bước lặp thứ n đó để dự đoán kết quả của bộ học.

- 128_attention_3_layers



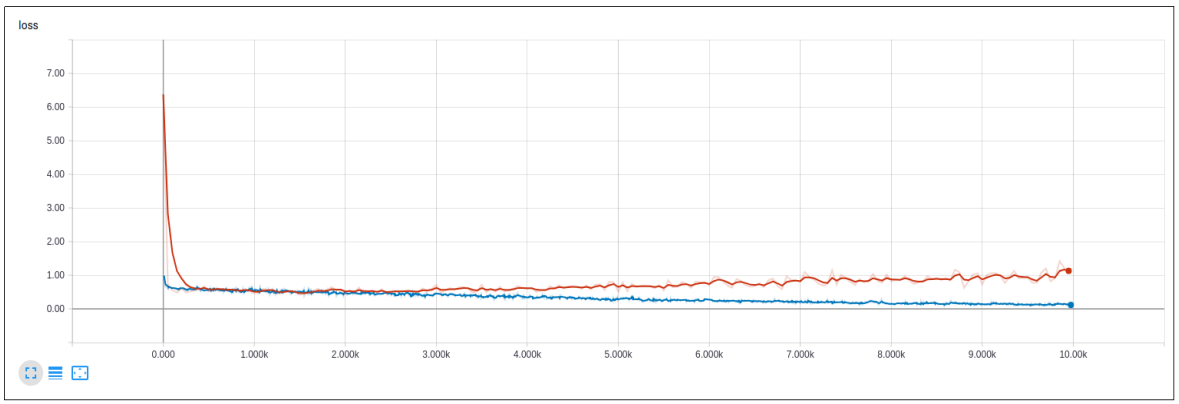
Hình 2.6: Đồ thị eval và train của bộ tham số 128_attention_3_layers

- 512_attention_3_layers



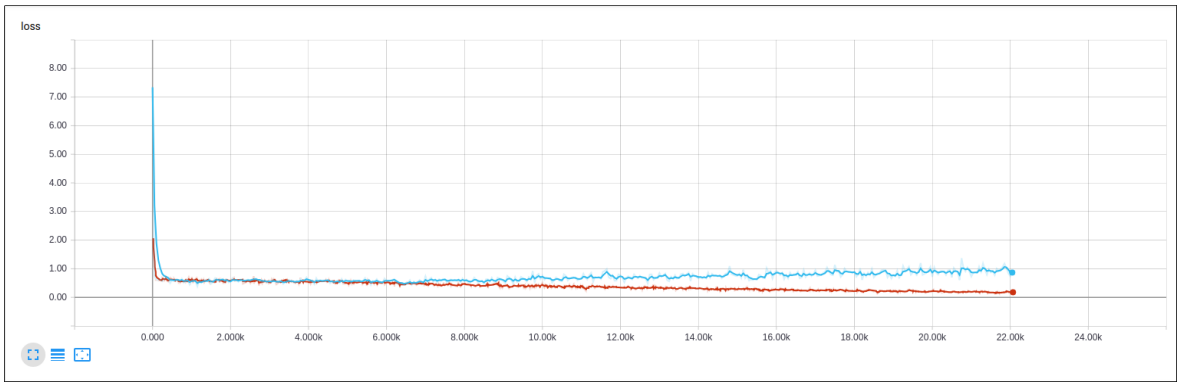
Hình 2.7: Đồ thị eval và train của bộ tham số 512_attention_3_layers

- 512_attention_1_layer



Hình 2.8: Đồ thị eval và train của bộ tham số 512_attention_1_layer

- 512_no_attention_3_layers



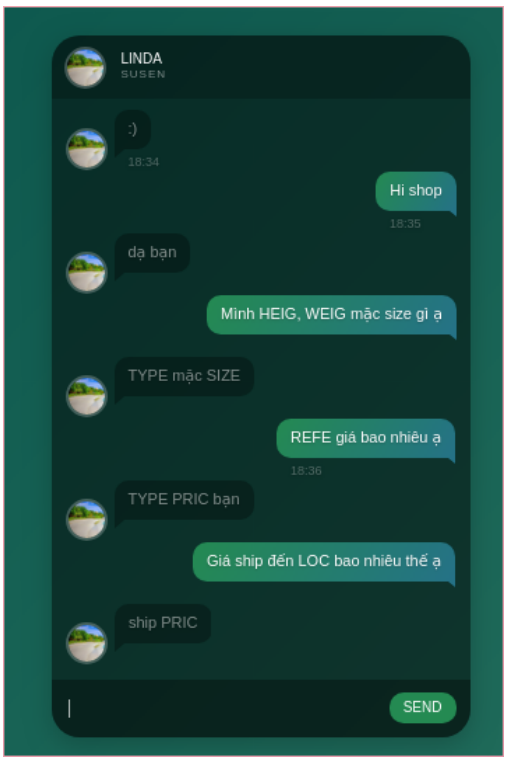
Hình 2.9: Đồ thị eval và train của bộ tham số 512_no_attention_3_layers

Trong các đồ thị trên ta thấy đều có một thời điểm mà sau thời điểm này giá trị loss của train giảm nhưng loss của eval lại tăng dẫn đến overfitting. Bảng 3 dưới đây thể hiện điều kiện dừng train tại bước lặp có loss eval và train hợp lý để tránh overfit và thời gian train mỗi bước lặp của 4 bộ tham số.

STT	Tên tóm tắt bộ tham số	Thời gian train (s) / Bước lặp (batch_size=32)	Bước lặp được chọn để dự đoán
1	128_attention_3_layers	1.677980s	14.000
2	512_attention_3_layers	7.220729s	4.000
3	512_attention_1_layer	4.696341s	4.500
4	512_no_attention_3_layers	3.931894s	10.000

Bảng 4: Thời gian huấn luyện và thời điểm dừng huấn luyện của các bộ tham số

2.4.2. Giao diện ứng dụng và một số hình ảnh demo



Hình 2.10. Giao diện demo

Hình 2.11. Giao diện demo

Hình 2.12. Giao diện demo

Hình 2.13. Giao diện demo

Trên đây là một vài cuộc hội thoại của mô hình với con người của bộ tham số thứ 2 - ***512_attention_3_layers***.

2.4.3. Đánh giá hiệu năng và kết quả

2.4.3.1. Kết quả

Xét tập dữ liệu kiểm thử khoảng 200 câu hỏi đã được gán nhãn NER được lấy từ tập dữ liệu ban đầu và đưa vào mỗi bộ tham số huấn luyện của mô hình để đánh giá và so sánh.

Để có thể đánh giá một cách khách quan kết quả của mô hình thông qua bộ dữ liệu kiểm thử này, yêu cầu 4 chuyên gia đánh giá trên 4 bộ tham số. Mỗi chuyên gia sẽ đánh giá 4 bộ tham số, mỗi bộ tham số 200 câu hỏi và câu trả lời tương ứng. Với mỗi câu hỏi và câu trả lời được gán nhãn, các chuyên gia cần xem xét và liên tưởng đến trường hợp xảy ra thật trong đời sống, nếu câu trả lời phù hợp với câu hỏi, chuyên gia sẽ cho điểm đánh giá là 1, ngược lại cho điểm 0. Một số kết quả dự đoán và đánh giá của các chuyên gia với 4 bộ tham số được thể hiện trong các bảng 5,6,7,8 dưới đây.

Question	**Answer**	**Người đánh giá 1**	**Người đánh giá 2**	**Người đánh giá 3**	**Người đánh giá 4**
cho mình hỏi giá đôi giày	dạ đúng DIGIT bạn ạ	1	1	0	0
bạn cho mình xem ảnh rõ trong TYPE được không bạn	dạ xem up mình bạn ạ	0	0	0	0
còn TYPE size SIZE không shop	TYPE mình size SIZE ạ	1	1	1	0
size SIZE còn không bạn	TYPE mình DIGIT SIZE ạ	1	0	0	1
REFE bao_nhiêu bạn	TYPE PRIC bạn	1	0	0	1
còn COLO không bạn bao_nhiêu	mình còn ạ	1	1	1	1
TYPE bao_nhiêu ạ	TYPE PRIC bạn	1	1	1	1
shop ơi khi_nào REFE về shop nhắn_tin mình nhé	dạ bạn ạ	1	1	1	1
bạn ơi TYPE mình hỏi bao_giờ về nhỉ	dạ mình về ạ	0	0	0	0
cho mình hỏi giá_áo với TYPE bao_nhiêu ạ	TYPE PRIC bạn	1	1	1	1
cho mình hỏi DIGIT mẫu TYPE REFE còn không ạ	mình còn DIGIT mình bạn ạ	1	0	1	0
TYPE bao_giờ có hàng mới về ạ	dạ mình không về nữa ạ	1	0	1	1

jump này còn không ạ	mình còn ạ	1	1	1	1
REFE bao_nhiêu vậy	TYPE PRIC bạn	1	1	1	1
REFE thì còn không nhỉ	mình còn ạ	1	1	1	1
còn ở cơ_sở nào vậy nhỉ để mình qua xem	DIGIT cơ_sở còn hết bạn ạ	0	0	1	1
mình qua shop xem rồi	dạ bạn	0	1	1	1
ok TYPE còn đủ màu đúng không bạn nhỉ	dạ mình còn hết thôi ạ	0	0	0	0
cho mình hỏi DIGIT REFE còn ở cơ_sở nào nhỉ	dạ cơ_sở đều ạ	0	0	0	0
cho mình xin giá REFE TYPE COLO này với	TYPE PRIC bạn	1	1	1	1

Bảng 5: Một vài kết quả đánh giá của bộ tham số 128_attention_3_layers

Question	**Answer**	**Người đánh giá 1**	**Người đánh giá 2**	**Người đánh giá 3**	**Người đánh giá 4**
cho mình hỏi giá đôi giày	REFE mình bán ạ	0	0	0	0
bạn cho mình xem ảnh rõ trong TYPE được không bạn	bạn rõ ảnh sao nhé	0	0	0	0
còn TYPE size SIZE không shop	REFE mình hết mất rồi ạ	1	1	1	1
size SIZE còn không bạn	REFE mình có DIGIT SIZE form	1	1	1	0
REFE bao_nhiêu bạn	TYPE PRIC bạn	1	1	1	1
còn COLO không bạn bao_nhiêu	mình còn hết rồi ạ	0	1	0	0
TYPE bao_nhiêu a	TYPE PRIC bạn nhé	1	1	1	1
shop ơi khi_nào REFE về shop nhắn_tin mình nhé	dạ ok ạ	1	1	1	1
bạn ơi TYPE mình hỏi bao_giờ về nhỉ	REFE chắc mình về ạ	0	0	0	0
cho mình hỏi giá_áo với TYPE bao_nhiêu ạ	TYPE PRIC bạn	1	1	1	1
cho mình hỏi DIGIT mẫu TYPE REFE còn không ạ	TYPE còn kẻ COLO ở mình bạn ạ	0	0	0	0
TYPE bao_giờ có hàng mới về ạ	REFE mình không về nữa ạ	1	1	1	1
jump này còn không ạ	mình còn hết ạ	0	0	0	1

REFE bao_nhiêu vậy	TYPE PRIC bạn	1	1	1	1
REFE thì còn không nhỉ	mình còn hết ạ	0	0	0	1
còn ở cơ_sở nào vậy nhỉ để mình qua xem	DIGIT còn DIGIT cơ_sở bạn	0	0	0	0
mình qua shop xem rồi	ok bạn ạ	1	1	1	1
ok TYPE còn đủ màu đúng không bạn nhỉ	vâng mình còn COLO ạ	0	0	1	1
cho mình hỏi DIGIT REFE còn ở cơ_sở nào nhỉ	DIGIT cơ_sở đều còn bạn	1	1	1	1
cho mình xin giá REFE TYPE COLO này với	TYPE giá PRIC bạn nhé	1	1	1	1

Bảng 6: Một vài kết quả đánh giá của bộ tham số 512_attention_3_layers

Question	**Answer**	**Người đánh giá 1**	**Người đánh giá 2**	**Người đánh giá 3**	**Người đánh giá 4**
cho mình hỏi giá đôi giày	dạ mình không bán ạ	1	1	1	1
bạn cho mình xem ảnh rõ trong TYPE được không bạn	TYPE xem sale bạn ạ	0	0	0	0
còn TYPE size SIZE không shop	REFE mình hết SIZE ạ	1	1	1	1
size SIZE còn không bạn	mình còn SIZE ạ	1	1	1	1
REFE bao_nhiêu bạn	TYPE PRIC bạn	1	1	1	1
còn COLO không bạn bao_nhiêu	mình còn ạ	1	1	1	1
TYPE bao_nhiêu a	TYPE PRIC bạn	1	1	1	1
shop ơi khi_nào REFE về shop nhắn_tin mình nhé	ok bạn	1	1	1	1
bạn ơi TYPE mình hỏi bao_giờ về nhỉ	REFE chắc không lại_nữa ạ	1	1	1	1
cho mình hỏi giá_áo với TYPE bao_nhiêu ạ	TYPE PRIC bạn nhé	1	1	1	1
cho mình hỏi DIGIT mẫu TYPE REFE còn không ạ	TYPE còn mình CODE mình còn bạn	0	0	0	0
TYPE bao_giờ có hàng mới về ạ	REFE mình không về ạ	1	1	1	1
jump này còn không ạ	dạ mình hết ạ	1	1	1	1
REFE bao_nhiêu vậy	TYPE PRIC bạn	1	1	1	1

REFE thì còn không nhỉ	REFE hết ạ	1	1	1	1
còn ở cơ_sở nào vậy nhỉ để mình qua xem	DIGIT cơ_sở đều bạn	0	0	0	0
mình qua shop xem rồi	ok bạn ạ	1	1	1	1
ok TYPE còn đủ màu đúng không bạn nhỉ	vâng mình còn màu rồi ạ	0	0	0	0
cho mình hỏi DIGIT REFE còn ở cơ_sở nào nhỉ	DIGIT cơ_sở đều bạn	0	0	0	0
cho mình xin giá REFE TYPE COLO này với	TYPE PRIC bạn	1	1	1	1

Bảng 7: Một vài kết quả đánh giá của bộ tham số 512_attention_1_layer

Question	**Answer**	**Người đánh giá 1**	**Người đánh giá 2**	**Người đánh giá 3**	**Người đánh giá 4**
cho mình hỏi giá đôi giày	giày mình không bán ạ	1	1	1	1
bạn cho mình xem ảnh rõ trong TYPE được không bạn	ảnh rõ TYPE là sao ạ CODE	1	1	1	1
còn TYPE size SIZE không shop	dạ REFE mình hết hàng rồi ạ	1	1	1	1
size SIZE còn không bạn	mình còn DIGIT thêm	0	0	0	0
REFE bao_nhiêu bạn	TYPE PRIC bạn nhé	1	1	1	1
còn COLO không bạn bao_nhiêu	TYPE mình hết hàng rồi ạ	1	1	1	1
TYPE bao_nhiêu a	TYPE giá PRIC ạ	1	1	1	1
shop ơi khi_nào REFE về shop nhắn_tin mình nhé	dạ ok bạn ạ	1	1	1	1
bạn ơi TYPE mình hỏi bao_giờ về nhỉ	REFE bên mình không về lại_nữa ạ	1	1	1	1
cho mình hỏi giá_áo với TYPE bao_nhiêu ạ	TYPE PRIC CODE bạn nha	1	1	1	1
cho mình hỏi DIGIT mẫu TYPE REFE còn không ạ	TYPE COLO thì mình hết ạ	0	0	0	1
TYPE bao_giờ có hàng mới về ạ	REFE mình không về nữa ạ	1	1	1	1
jump này còn không ạ	dạ mình hết ạ	1	1	1	1
REFE bao_nhiêu vậy	TYPE PRIC bạn	1	1	1	1

REFE thì còn không nhỉ	dạ REFE mình hết rồi	1	1	1	1
còn ở cơ_sở nào vậy nhỉ để mình qua xem	DIGIT cơ_sở đều còn màu COLO bạn nha	1	1	1	1
mình qua shop xem rồi	cám ơn bạn nha	1	1	1	1
ok TYPE còn đủ màu đúng không bạn nhỉ	dạ mình còn đủ màu ạ	1	1	1	1
cho mình hỏi DIGIT REFE còn ở cơ_sở nào nhỉ	DIGIT cơ_sở đều còn bạn nhé	1	1	1	1
cho mình xin giá REFE TYPE COLO này với	TYPE PRIC CODE bạn nhé	1	1	1	1

Bảng 8: Một vài kết quả đánh giá bộ tham số 512_no_attention_3_layers

Bảng 9 dưới đây tóm tắt tổng đánh giá của bộ dữ liệu kiểm thử với từng bộ tham số. Chi tiết một phần kết quả dự đoán của bộ tham số có tỉ lệ đánh giá cao nhất được thể hiện trong Phụ lục A.

STT	Tên tóm tắt bộ tham số	Số điểm đánh giá/Số câu của Người đánh giá thứ 1	Số điểm đánh giá/Số câu của Người đánh giá thứ 2	Số điểm đánh giá/Số câu của Người đánh giá thứ 3	Số điểm đánh giá/Số câu của Người đánh giá thứ 4	Số điểm đánh giá trung bình
1	128_attention_3_layers	139/200	134/200	130/200	145/200	***137/200***
2	512_attention_3_layers	154/200	143/200	138/200	149/200	***146/200***
3	512_attention_1_layer	125/200	124/200	128/200	130/200	***127/200***
4	512_no_attention_3_layers	153/200	152/200	152/200	158/200	***154/200***

Bảng 9: Đánh giá mô hình với các bộ tham số khác nhau

Từ bảng 9 ta thấy bộ tham số ***512_no_attention_3_layers*** có chất lượng vượt trội hơn hẳn so các bộ tham số còn lại và các câu trả lời của bộ tham số này có độ linh hoạt và đa dạng mẫu trả lời hơn so với các bộ còn lại. Bộ tham số ***512_attention_3_layers*** có kết quả đánh giá tốt hơn bộ tham số ***512_attention_1_layer*** và tốt hơn bộ ***128_attention_3_layers***.

2.4.3.2. Nhận xét và giải thích

a. Nhận xét

Ưu điểm của mô hình sinh được huấn luyện bởi 4 bộ tham số thì đều trả lời được đa dạng các câu hỏi và các câu trả lời này đều gần với ngôn ngữ tự nhiên.

Nhược điểm chung của các bộ tham số:

- Kết quả của mô hình sinh chuỗi trả lời còn chưa tốt. Với một câu hỏi đầu vào "Xin chào shop", mô hình sinh câu trả lời "Vâng ạ ạ ạ", có khá nhiều từ bị lặp trong câu trả lời do đó cần hậu xử lý câu trả lời bằng cách loại bỏ các từ bị trùng lặp.
- Hệ thống chưa trả lời chính xác những câu hỏi đồng nghĩa do tập dữ liệu huấn luyện chưa đủ nhiều và còn sai chính tả. Ví dụ với 2 câu hỏi: "Giá TYPE thế _nào?" và "Giá TYPE bao_nhiêu ạ?" thì mô hình của cả 4 bộ tham số hầu như chỉ trả lời tốt ở câu hỏi thứ hai. Do đó cần có bộ dữ liệu nhiều và đa dạng các câu hỏi và câu trả lời hơn.
- Số lượng nhãn NER còn chưa đủ, cần có thêm nhãn thông tin INVENTORY – số lượng hàng còn trong kho. Như với trường hợp khách hàng hỏi còn hàng hay hết hàng thì hệ thống cần truy vấn cơ sở dữ liệu để đưa ra câu trả lời phù hợp.

Các bộ tham số sử dụng attention

Nhược điểm:

- Vai vế, cách xưng hô giữa chatbot với người dùng còn chưa được tự nhiên. Ví dụ khi người dùng xưng "chị", nói chuyện với hệ thống là "có gì nhắn_tin chị nhé" nhưng hệ thống lại trả lời người dùng là "dạ em ạ."
- Câu trả lời thường có vẻ có ý đúng nhưng lại không hoàn chỉnh từ ngữ. Nếu câu trả lời cho câu hỏi dưới đây là "REFE chắc không về lại nữa ạ" thì việc đánh giá của các chuyên gia có độ chính xác cao hơn.

bạn ơi TYPE mình hỏi bao_giờ về nhỉ	REFE chắc không lại_nữa ạ

Bộ tham số 512_no_attention_3_layers

Ưu điểm:

- Trả lời mượt mà và gần với ngôn ngữ tự nhiên nhất trong số các bộ tham số còn lại
- Cách xưng hô vai vế trong cuộc trò chuyện chuẩn xác
- Cấu trúc ngữ pháp câu trả lời như một câu hội thoại trong tự nhiên
- Câu trả lời phong phú, đa dạng với các câu hỏi cùng ý nghĩa.

Ví dụ như "TYPE này bao nhiêu shop?", "TYPE này bao nhiêu ạ" thì bộ tham số này sẽ sinh ra 2 câu trả lời khác nhau thay vì một câu trả lời "TYPE PRIC ạ" của 3 bộ tham số sử dụng attention.

b. Giải thích

Mô hình seq2seq kết hợp attention là một mô hình được giới thiệu trong các bài toán Dịch Máy, mỗi từ trong câu nguồn được học một trọng số và có ý nghĩa tương ứng một từ hoặc cụm từ trong câu đích. Với câu nguồn "Tôi đi học" và câu "I go to school" là câu đích với bản dịch tiếng Anh tương ứng thì từ "Tôi" – "I", cụm từ "đi học" – "go to school" có cùng ý nghĩa. Mặt khác, với bài toán Dịch Máy, cấu trúc câu giữa câu nguồn và câu đích là ánh xạ 1:1, câu nguồn là câu chủ động thì câu đích cũng sẽ là câu chủ động, và giữa các câu nguồn và câu đích có một thứ tự sắp xếp nhất định tương ứng với ngữ pháp của loại hình ngôn ngữ đó. Do đó, việc áp dụng Attention cho bài toán Dịch Máy là phù hợp và hợp lý, mang lại hiệu quả cao.

Còn với bài toán Chatbot, giữa câu hỏi và câu trả lời không có sự liên kết ánh xạ 1:1. Một câu hỏi có thể có nhiều câu trả lời khác nhau và không có một cách sắp xếp hay một quy tắc nào cố định cho câu trả lời bằng ngôn ngữ tự nhiên trong khi với Dịch Máy một câu nguồn chỉ tương ứng với 1 câu đích. Với một câu "REFE COLO bao nhiêu tiền?" mà khách hàng hỏi shop, có thể trả lời "Mình hết mất rồi ạ" hoặc "Bạn hỏi REFE nào ạ" hay "Giá PRIC bạn nhé",… Do đó việc áp dụng Attention vào bài toán Chatbot không hợp lý nên kết quả mang lại không cao bằng sử dụng mô hình Seq2seq cơ bản.

Ngoài ra mô hình ***512_attention_3_layers*** đạt được mức đánh giá cao hơn ***128_attention_3_layers*** làm cho lượng thông tin lưu lại nhiều hơn và thu được kết quả tốt hơn. ***512_attention_3_layers*** được học qua nhiều tầng hơn giúp thông tin được sàng lọc kĩ hơn, chất lượng câu trả lời cũng tốt hơn so với ***512_attention_1_layer***.

KẾT LUẬN

Đồ án tốt nghiệp của em đã nêu ra được vấn đề, đề xuất phương pháp và xây dựng một hệ thống sinh câu trả lời sử dụng mô hình sinh chuỗi. Mô hình đã huấn luyện được trong đồ án tốt nghiệp còn chưa được tốt nguyên nhân do:

- Lượng dữ liệu chưa đủ nhiều, chưa có sự đa dạng các loại câu hỏi và câu trả lời
- Dữ liệu còn các thành phần sai chính tả và viết tắt mà chưa được xử lý triệt để.
- Bộ dữ liệu đã gán nhãn NER độ chính xác chưa tối đa
- Ước lượng thời điểm hàm mất mát có giá trị eval cao nhất mà train thấp nhất chưa chuẩn xác làm giảm khả năng dự đoán của mỗi bộ tham số của mô hình.
- Chưa có quản lý hội thoại, mô hình mới chỉ trả lời được dựa trên câu hỏi hiện tại chứ không xâu chuỗi các đoạn hội thoại trước đó.

Hướng phát triển tiếp theo sau này bao gồm:

- Xây dựng phần quản lý hội thoại, định hướng hội thoại với mục tiêu cụ thể trong ngành kinh doanh thời trang như: Tư vấn để khách hàng mua hàng, khi Khách hàng đặt hàng thì sẽ tạo một bản ghi Order trong cơ sở dữ liệu tích hợp vào hệ thống giúp người bán hàng dễ dàng quản lý, …
- Xây dựng các thành phần khác trong mô hình đề xuất để có một hệ thống Chatbot hoàn chỉnh.
- Mở rộng ra các lĩnh vực khác như Chatbot tư vấn chăm sóc sức khỏe, bán hàng thực phẩm chức năng, …

TÀI LIỆU THAM KHẢO

[1] J. Bang, "Chatbot là gì, tại sao những ông lớn công nghệ như Microsoft, Facebook lại xem nó như ưu tiên hàng đầu và ném hàng tỷ USD vào đó?," 5 Jan 2017. [Online]. Available: http://bit.ly/2KY7QGa. [Accessed May 2018].

[2] M. H. Quang, "Tất cả những gì bạn cần biết về chatbot," [Online]. Available: https://viblo.asia/p/tat-ca-nhung-gi-ban-can-biet-ve-chatbot-Az45bnNg5xY.

[3] D. Britz, 6 April 2016. [Online]. Available: http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/. [Accessed May 2018].

[4] Rajarshi Sengupta and Shankar Lakshman, "Conversational Chatbots – Let's chat," Deloitte Analysis, [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/in/Documents/strategy/in-strategy-innovation-conversational-chatbots-lets-chat-final-report-noexp.pdf. [Accessed May 2018].

[5] Niranjan Dandekar, Suyog Ghodey, "Implementation of a Chatbot using Natural," in *9th International Conference on Recent Development in Engineering Science, Humanities and Management, Mahratta Chamber of Commerce, Industries and Agriculture*, Pune (India), 2017.

[6] S. Ram, "Chatbots with Seq2Seq," 28 June 2016. [Online]. Available: http://suriyadeepan.github.io/2016-06-28-easy-seq2seq/. [Accessed May 2018].

[7] D. M. Hai, "[RNN] RNN là gì?," 19 October 2017. [Online]. Available: https://dominhhai.github.io/vi/2017/10/what-is-rnn/. [Accessed April 2018].

[8] D. M. Hai, "[RNN] LSTM là gì?," 20 October 2017. [Online]. Available: https://dominhhai.github.io/vi/2017/10/what-is-lstm/. [Accessed April 2018].

[9] J. Collis, "What is word embedding in deep learning?," 20 April 2017. [Online]. Available: https://www.quora.com/What-is-word-embedding-in-deep-learning. [Accessed May 2018].

[10] E. B. R. Z. Thang Luong, "Neural Machine Translation (seq2seq) Tutorial," [Online]. Available: https://www.tensorflow.org/tutorials/seq2seq. [Accessed April 2018].

[11] K. C. a. Y. B. Dzmitry Bahdanau, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[12] V. H. Tiep, "Softmax Regression," 17 February 2017. [Online]. Available: https://machinelearningcoban.com/2017/02/17/softmax/. [Accessed May 2018].

[13] V. H. Tiep, "Gradient Descent (phần 1/2)," 12 January 2017. [Online]. Available: https://machinelearningcoban.com/2017/01/12/gradientdescent/. [Accessed May 2018].

[14] V. H. Tiep, "Gradient Descent (phần 2/2)," 16 January 2017. [Online]. Available: https://machinelearningcoban.com/2017/01/16/gradientdescent2/#-stochastic-gradient-descent. [Accessed May 2018].

[15] [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/contrib/legacy_seq2seq/embedding_rnn_seq2seq.

[16] [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/contrib/legacy_seq2seq/embedding_attention_seq2seq.

[17] S. Ram, "Practical seq2seq," 31 December 2016. [Online]. Available: http://suriyadeepan.github.io/2016-12-31-practical-seq2seq/. [Accessed May 2018].

[18] P. Q. N. Minh, "Kỹ thuật Attention trong mô hình Sequence-to-Sequence và ứng dụng trong xử lý ngôn ngữ tự nhiên," 10 October 2017. [Online]. Available: https://tech.fpt.com.vn/ky-thuat-attention-trong-mo-hinh-sequence-sequence-va-ung-dung-trong-xu-ly-ngon-ngu-tu-nhien/. [Accessed May 2018].

[19] F. Ottaviani, "Daily UI #013: Direct Messaging," [Online]. Available: https://codepen.io/supah/pen/jqOBqp. [Accessed May 2018].

PHỤ LỤC A

Một số kết quả dự đoán của mô hình có bộ trọng số được đánh giá cao nhất 512_no_attention trong mục 2.4.3.1.

Question	**Answer**
TYPE jogger bnhiêu ạ	TYPE TRAD giá PRIC bạn nha
REFE có những màu gì ạ	mình có DIGIT màu như ảnh ạ
TYPE có size không ạ	TYPE có size SIZE
REFE còn size SIZE không a	dạ mình còn ạ
giầy này bao_nhiêu ạ	TYPE PRIC bạn nha
cho em giá REFE nữa ạ	TYPE TRAD giá PRIC bạn nha
TYPE có size không ạ	TYPE có size SIZE
đây là TYPE hay ạ bao_nhiêu ạ	TYPE giá PRIC ạ
có size SIZE không ạ	TYPE mình có size SIZE CODE mình còn thôi ạ
shop ơi REFE còn không ạ đây nữa ạ	REFE bên mình hết rồi ạ
cho em giá ạ	TYPE PRIC bạn nha
REFE bao_nhiêu bạn	TYPE PRIC bạn nhé
REFE còn không bạn	mình còn ạ
REFE còn không a	mình còn nhé ạ
còn REFE không shop	mình còn size SIZE
shop còn REFE không ạ	mình hết ạ
dạ vâng em cảm_ơn khi nào có lại vậy ạ	dạ vâng bạn ạ CODE sẽ báo bạn nhé
vâng em cảm_ơn ạ	ok bạn
REFE bao_nhiêu vậy ạ	TYPE mình hết nhé
dạ vâng shop có ảnh TYPE màu COLO không ạ	TYPE COLO SIZE bạn nhỉ
màu khác còn hàng của REFE ạ	mình còn màu COLO bạn ạ
dạ shop còn REFE không ạ	mình hết ạ
cho em hỏi TYPE bên mình có sale không ạ	TYPE DIGIT mình còn ạ
REFE bên mình còn size không ạ	dạ mình hết ạ
REFE bao_nhiêu vậy shop	TYPE PRIC bạn nhé
TYPE này còn không shop	mình còn TYPE COLO ở ạ

TYPE COLO bao_nhiêu ạ shop	TYPE giá PRIC bạn nha
có ship không shop	dạ mình không có ạ
ok shop còn chân_váy kẻ không ạ	mình còn TYPE SIZE CODE PRIC ạ
vậy kẻ màu COLO ý ạ đây ạ cho mình giá REFE ạ	TYPE mình có size SIZE bạn nhé
REFE còn không bạn ơi	mình còn ạ
TYPE da bao_nhiêu hả bạn	TYPE giá PRIC ạ
còn TYPE không shop ơi	mình hết mất rồi ạ
còn DIGIT TYPE với loe COLO không ạ	mình còn nha ạ
shop TIME ship em được không ạ	bạn ở đâu ạ CODE
shop bán giày này không ạ	giày mình không bán giày
shop còn REFE không ạ	mình hết ạ
shop còn chân_váy này không ạ	tớ còn size SIZE bạn nha
shop còn TYPE dạ này sale không ạ	mình hết mất rồi ạ
ok ạ bao_nhiêu vậy shop	TYPE giá PRIC bạn nhé
shop ơi REFE WEIG HEIG mặc vừa không ạ	bạn mặc dài a
có nhận ship không ạ	mình nhận ship ạ
shop có bán giày không	dạ mình không bán giày ạ
TYPE bao_nhiêu vậy bạn	TYPE PRIC nhé
REFE còn không bạn	mình còn ạ
TYPE còn không bạn	mình còn DIGIT size SIZE bạn
thế bạn giữ cho mình	dạ vâng ạ
REFE còn không bạn	mình còn ạ
REFE bao_nhiêu vậy bạn	TYPE PRIC
TYPE còn size SIZE không bạn	TYPE còn DIGIT size SIZE form suông nha
TYPE bao_nhiêu vậy bạn	TYPE PRIC nhé
bạn ới shop bạn có REFE không	dạ mình có ạ
TYPE bao_nhiêu vậy bạn	TYPE PRIC nhé
TYPE có size không bạn	TYPE có size SIZE bạn nhé
không phải REFE đâu đúng không bạn	vâng DIGIT mẫu khác_nhau ạ
bạn ơi TYPE dạ TRAD COLO có về size SIZE không	không ấy ạ

TYPE có mũ không shop chất gió đúng không	dạ đúng rồi ạ
có mũ không bạn	có mũ ạ
REFE còn size SIZE không shop	mình còn size SIZE
bên cơ_sở nào đang còn vậy bạn	DIGIT cơ_sở đều còn bạn nhé
TYPE bao_nhiêu vậy shop	TYPE PRIC sale SAOF bạn
cho mình hỏi TYPE COLO và bao_nhiêu tiền vậy ạ	TYPE PRIC CODE bạn nha
có size SIZE không ạ	TYPE mình có size SIZE CODE mình còn thôi ạ
có còn nhận SHME nữa không ạ	mình nhận ship LOC thôi ạ