

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**NGUYỄN VĂN QUYỀN**

**XÂY DỰNG MÔ HÌNH BÁN HÀNG TỰ ĐỘNG  
TRÊN INTERNET**

**LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN**

**HÀ NỘI - 2016**

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**NGUYỄN VĂN QUYỀN**

**XÂY DỰNG MÔ HÌNH BÁN HÀNG TỰ ĐỘNG  
TRÊN INTERNET**

Ngành: Công nghệ thông tin

Chuyên ngành: Hệ thống thông tin

Mã số: 14025060

**LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN**

**NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. Nguyễn Văn Nam**

**HÀ NỘI – 2016**

## LỜI CAM ĐOAN

Tôi là Nguyễn Văn Quyền, học viên khóa K21, ngành Công nghệ thông tin, chuyên ngành Hệ Thống Thông Tin. Tôi xin cam đoan luận văn “*Xây dựng mô hình bán hàng tự động trên Internet*” là do tôi nghiên cứu, tìm hiểu và phát triển dưới sự hướng dẫn của TS. Nguyễn Văn Nam, không phải sao chép từ các tài liệu, công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

*Hà Nội, ngày      tháng      năm*

## MỤC LỤC

|   |    |
|---|----|
| TÓM TẮT NỘI DUNG .....  | 1  |
| 1. CHƯƠNG 1: GIỚI THIỆU CHUNG .....                             | 2  |
| 2. CHƯƠNG 2: TỔNG QUAN BÁN HÀNG TỰ ĐỘNG .....                   | 4  |
| 2.1 Các mô hình trả lời bán hàng tiêu biểu .....                | 4  |
| 2.2 Các vấn đề cần giải quyết và cải tiến.....                  | 11 |
| 3. CHƯƠNG 3: MẠNG NƠ-RON VÀ PHƯƠNG PHÁP SEQ2SEQ .....           | 14 |
| 3.1 Kiến thức tổng quan .....                                   | 14 |
| 3.2 Framework TensorFlow.....                                   | 18 |
| 3.3 Lý thuyết mạng nơ-ron.....                                  | 19 |
| 3.3.1 Mạng nơ-ron nhân tạo ANN.....                             | 19 |
| 3.3.2 Mạng nơ-ron tái phát RNN.....                             | 21 |
| 3.3.3 Mạng Long Short Term Memory LSTM .....                    | 24 |
| 3.4 Phương pháp học chuỗi Seq2Seq .....                         | 29 |
| 4. CHƯƠNG 4: GIẢI PHÁP VÀ XÂY DỰNG MÔ HÌNH ĐỀ XUẤT ISALES ..... | 36 |
| 4.1 Giải pháp đề xuất .....                                     | 36 |
| 4.2 Nguồn dữ liệu huấn luyện .....                              | 38 |
| 4.2.1 Facebook .....  | 39 |
| 4.2.2 GraphAPI .....  | 40 |
| 4.3 Xây dựng mô hình iSales .....                               | 42 |
| 4.3.1 Pha thu thập dữ liệu.....                                 | 42 |
| 4.3.2 Pha tiền xử lý dữ liệu.....                               | 44 |
| 4.3.3 Pha phân mảnh dữ liệu .....                               | 46 |
| 4.3.4 Pha huấn luyện dữ liệu.....                               | 48 |
| 4.3.5 Pha sinh câu trả lời .....                                | 51 |
| 5. CHƯƠNG 5: THỬ NGHIỆM, ĐÁNH GIÁ KẾT QUẢ .....                 | 54 |
| 5.1 Phát biểu usecase.....                                      | 54 |
| 5.2 Thử nghiệm iSales .....                                     | 55 |
| 5.3 Đánh giá kết quả.....                                       | 60 |
| 6. KẾT LUẬN.....  | 62 |
| TÀI LIỆU THAM KHẢO .....  | 63 |

## DANH MỤC KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

| Từ viết tắt | Từ chuẩn                    | Diễn giải  |
|-------------|-----------------------------|--|
| Chatbot     | Chatbot                     | Hệ thống trả lời tự động                             |
| Page        | Page                        | Khái niệm trang thông tin trên facebook              |
| Post        | Post                        | Khái niệm bài viết trên facebook                     |
| Comment     | Comment                     | Khái niệm bình luận trên facebook                    |
| NLP         | Natural Language Processing | Xử lý ngôn ngữ tự nhiên                              |
| ANN         | Artificial Neural Network   | Mạng nơ ron nhân tạo                                 |
| RNN         | Recurrent Neural Network    | Mạng nơ ron tái phát                                 |
| LSTM        | Long short-term memory      | Mạng cải tiến để giải quyết vấn đề phụ thuộc quá dài |
| Seq2Seq     | sequence to sequence        | Phương pháp học chuỗi liên tiếp trong DeepLearning   |

## DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ

|  |    |
|--|----|
| Hình 2.1: Luồng tương tác mô hình bán hàng sử dụng Messenger .....   | 5  |
| Hình 2.2: Mô hình bán hàng sử dụng Messenger.....                    | 6  |
| Hình 2.3: Từ điển sử dụng trong chatbot Skype .....                  | 6  |
| Hình 2.4: Mô hình bán hàng sử dụng chatbot Skype .....               | 7  |
| Hình 2.5: Mô hình bán hàng sử dụng chatbot Skype .....               | 8  |
| Hình 2.6: Mô hình bán hàng sử dụng uhChat .....                      | 10 |
| Hình 2.7: Mô hình bán hàng sử dụng Subiz.....                        | 11 |
| Hình 3.1: Các bước chung của mô hình tự động hiện tại.....           | 16 |
| Hình 3.2: Mô hình bán hàng tự động .....                             | 17 |
| Hình 3.3: Kiến trúc mạng nơ-ron nhân tạo ANN.....                    | 20 |
| Hình 3.4: Quá trình xử lý thông tin của nơ-ron j trong mạng ANN..... | 20 |
| Hình 3.5: Quá trình xử lý thông tin trong mạng RNN .....             | 23 |
| Hình 3.6: RNN phụ thuộc short-term. ....                             | 24 |
| Hình 3.7: RNN phụ thuộc long-term. ....                              | 25 |
| Hình 3.8: Các module lặp của mạng RNN chứa một layer. ....           | 26 |
| Hình 3.9: Các mô-đun lặp của mạng LSTM chứa bốn layer. ....          | 26 |
| Hình 3.10: Cell state của LSTM giống như một băng truyền. ....       | 27 |
| Hình 3.11: Cổng trạng thái LSTM. ....                                | 27 |
| Hình 3.12: LSTM focus f. ....  | 28 |
| Hình 3.13: LSTM focus i.....   | 28 |
| Hình 3.14: LSTM focus c. ....  | 29 |
| Hình 3.15: LSTM focus o.....   | 29 |
| Hình 3.16: Mô hình phát sinh văn bản .....                           | 30 |
| Hình 3.17: Quá trình huấn luyện và phát sinh văn bản .....           | 30 |
| Hình 3.18: Mô hình chuỗi liên tiếp Seq2Seq .....                     | 32 |
| Hình 3.19: Mô hình đối thoại seq2seq.....                            | 33 |
| Hình 3.20: Bộ mã hóa và giải mã seq2seq. ....                        | 33 |
| Hình 4.1: Các pha trong mô hình bán hàng tự động .....               | 36 |
| Hình 4.2: Sơ đồ quy trình của mô hình đề xuất.....                   | 37 |
| Hình 4.3: Mô hình giao tiếp sử dụng graphAPI Facebook .....          | 40 |
| Hình 4.4: Sử dụng graphAPI v2.6 để thu thập dữ liệu page .....       | 41 |
| Hình 4.5: Luồng nghiệp vụ pha thu thập dữ liệu.....                  | 43 |
| Hình 4.6: Mã nguồn sample pha thu thập dữ liệu .....                 | 43 |
| Hình 4.7: Kết quả sample pha thu thập dữ liệu .....                  | 44 |
| Hình 4.8: Luồng nghiệp vụ pha tiền xử lý dữ liệu .....               | 45 |
| Hình 4.9: Kết quả sample pha tiền xử lý dữ liệu.....                 | 45 |
| Hình 4.10: Luồng nghiệp vụ pha phân mảnh dữ liệu .....               | 47 |
| Hình 4.11: Kết quả sample pha phân mảnh dữ liệu .....                | 47 |

|   |    |
|---|----|
| Hình 4.12: Luồng nghiệp vụ pha huấn luyện dữ liệu .....   | 49 |
| Hình 4.13: Biểu đồ tuần tự pha huấn luyện dữ liệu.....    | 50 |
| Hình 4.14: Mã nguồn sample cho huấn luyện dữ liệu.....    | 51 |
| Hình 4.15: Luồng nghiệp vụ pha sinh câu trả lời .....     | 52 |
| Hình 4.16: Mã nguồn sample cho pha sinh câu trả lời ..... | 53 |
| Hình 5.1: Quy trình bán hàng tự động iSales .....         | 55 |

## **DANH MỤC BẢNG BIỂU**

|   |    |
|---|----|
| Bảng 2.1 Bảng phân loại mô hình bán hàng.....                   | 12 |
| Bảng 4.1 Các công cụ xây dựng mô hình iSales .....              | 38 |
| Bảng 4.2 Các phiên bản graphAPI Facebook.....                   | 40 |
| Bảng 4.3 Bảng nội dung làm sạch dữ liệu.....                    | 44 |
| Bảng 4.4 Danh sách các cấu phần xử lý trong pha huấn luyện..... | 50 |
| Bảng 5.1 Danh sách các page facebook thu thập dữ liệu .....     | 56 |
| Bảng 5.2 Bảng kết quả huấn luyện dữ liệu.....                   | 57 |
| Bảng 5.3 Danh sách câu hỏi thử nghiệm.....                      | 57 |
| Bảng 5.4 Thử nghiệm sinh câu trả lời trong trường hợp 1 .....   | 58 |
| Bảng 5.5 Thử nghiệm sinh câu trả lời trong trường hợp 2 .....   | 59 |
| Bảng 5.6 Thử nghiệm sinh câu trả lời trong trường hợp 3 .....   | 60 |



## TÓM TẮT NỘI DUNG

Trong những năm gần đây, thương mại điện tử bùng nổ mạnh, kèm theo đó là nhu cầu giải đáp khi mua hàng tăng cao. Điều này dẫn tới cần có một mô hình bán hàng, hỗ trợ người bán đưa ra các câu trả lời tự động, giảm thiểu công sức tư vấn cũng như tăng khả năng tương tác giữa người dùng và website thương mại điện tử.

Bên cạnh đó, trí tuệ nhân tạo ngày càng phát triển, dần đi vào các lĩnh vực ứng dụng đời sống hàng ngày. Với mục đích nghiên cứu một nhánh mới trong công nghệ học máy, tôi đã đề xuất và được phép nghiên cứu đề tài “***xây dựng mô hình bán hàng tự động trên Internet***”

Hiện tại, mô hình iSales được thiết kế dựa trên mạng nơ-ron, kết hợp phương pháp học chuỗi liên tiếp seq2seq, có khả năng hiểu Tiếng Việt, tự học từ các đoạn đối thoại được thu thập trên mạng hoặc cung cấp bởi người bán hàng và có thể sinh ra câu trả lời tự động. Mặc dù nghiên cứu hiện tại chưa thể đáp ứng cho sản phẩm thương mại nhưng iSales đã có một số kết quả nhất định, đặc biệt là ý nghĩa trong việc áp dụng phương pháp mới trong học máy.

## CHƯƠNG 1: GIỚI THIỆU CHUNG

Mạng xã hội ngày càng phát triển, và con người có thể bán hàng trực tiếp thông qua internet. Người bán hàng quảng cáo những sản phẩm của họ, người mua hàng cần nắm thông tin cũng như mức giá sản phẩm. Giữa hai bên cần có những cuộc đối thoại để tìm hiểu và đi đến kết luận trong phiên tư vấn. Bài toán được đặt ra ở đây là hiện nay, ngoài nhu cầu giới thiệu sản phẩm, những người bán hàng cần thêm mô hình tư vấn tự động cho website của mình. Vậy mô hình bán hàng tự động là gì? Tại sao lại cần mô hình như vậy? Những lợi ích và thuận tiện đạt được khi sử dụng mô hình này là gì?

Để giải đáp cho những câu hỏi ở trên, ta lấy ba ví dụ cụ thể của một website bán giày da, một website bán đồ thời trang online trên mạng, và một website bán hàng mỹ phẩm, cả ba website đã tích hợp ứng dụng chat. Ở ví dụ đầu tiên, cửa hàng bán giày da bình thường rất ít khách. Buổi sáng không có ai yêu cầu tư vấn về sản phẩm, buổi trưa nhân viên bán hàng đi ăn trưa. Đầu giờ chiều, khi quay lại, nhân viên bán hàng phát hiện có khách hàng hỏi về sản phẩm trong lúc mình ra ngoài. Do không có phản hồi tư vấn, khách hàng đó đã rời đi. Website bán giày da mất một khách hàng tiềm năng. Ngược lại, tại website bán đồ thời trang online, số lượng khách hàng hỏi về sản phẩm rất nhiều. Hai nhân viên bán hàng vừa hỗ trợ khách mua hàng tại shop, vừa tư vấn online qua ứng dụng chat. Công việc nhiều, nhân viên không thể đảm đương hết các trọng trách, những khách hàng online cảm thấy không thỏa mái vì không được tư vấn đã rời đi. Website bán đồ thời trang online mất đi lợi nhuận không nhỏ. Ở website cuối cùng, nữ nhân viên bán hàng mỹ phẩm liên tục phải trả lời những thắc mắc gần như giống nhau của các khách hàng về cùng một mẫu sản phẩm. Ứng dụng chat tích hợp sẵn trên website không cho phép đưa ra cùng một câu trả lời cho các câu hỏi tương tự như vậy. Những vấn đề nêu trên, chứng minh không phải lúc nào chúng ta cũng đủ thời gian và nguồn nhân lực để sẵn sàng kết nối, tư vấn với khách hàng. Do đó, cần có một mô hình trả lời bán hàng tự động. Tự động học dữ liệu từ những đoạn đối thoại mẫu, tự động sinh câu trả lời dựa trên câu hỏi đầu hỏi.

Microsoft đã đưa ra lời giải cho một nhánh nhỏ của bài toán, đó là xây dựng mô hình bán pizza tự động có tên là chatbot Skype. Mô hình xây dựng trên phương pháp trích xuất câu trả lời, cho phép người mua hàng đặt hàng pizza bằng cách trả lời các thông tin được hỏi từ chatbot. Tuy nhiên, lời giải trên chưa hỗ trợ Tiếng Việt, không có kết quả nếu người mua hàng không thực hiện đúng như yêu cầu chatbot. Những bất cập này làm cho việc vận hành và sử dụng hệ thống không

mang lại nhiều lợi ích thiết thực. Với mong muốn xây dựng mô hình có chất lượng tốt hơn chatbotSkype, đồng thời áp dụng hướng đi mới, sử dụng mạng nơ-ron kết hợp phương pháp học chuỗi liên tiếp seq2seq, tôi đã nghiên cứu và ***xây dựng mô hình bán hàng tự động trên Internet*** iSales hỗ trợ Tiếng Việt để phục vụ riêng cho nghiệp vụ thương mại điện tử.

Để mô tả kết quả nghiên cứu và phương án xây dựng, luận văn được chia thành các chương như sau:

- Chương 1: Giới thiệu chung.  
Đặt vấn đề bài toán, từ đó nêu ra ý tưởng xây dựng mô hình.
- Chương 2: Tổng quan bán hàng tự động.  
Giới thiệu về 4 mô hình bán hàng trên Internet hiện nay, gồm có Messenger, chatbot Skype, uhChat, suBiz. Phân loại, liệt kê ưu, nhược điểm của mỗi mô hình và đưa ra các vấn đề cần cải tiến.
- Chương 3: Mạng nơ-ron và phương pháp seq2seq.  
Đưa ra các lý thuyết nền tảng về học máy, mạng nơ-ron và phương pháp học chuỗi seq2seq giúp bổ sung kiến thức cơ bản khi xây dựng mô hình bán hàng tự động
- Chương 4: Giải pháp và xây dựng mô hình đề xuất iSales.  
Đề xuất giải pháp cho mô hình đề xuất iSales, mô tả các thành phần và phương án xây dựng chi tiết.
- Chương 5: Thử nghiệm, đánh giá kết quả.  
Thử nghiệm mô hình trong usecase thực tế, đánh giá kết quả và so sánh với mô hình chatbotSkype.
- Kết luận.

Đưa ra kết luận trong quá trình nghiên cứu và xây dựng “mô hình bán hàng tự động trên Internet”.

## CHƯƠNG 2: TỔNG QUAN BÁN HÀNG TỰ ĐỘNG

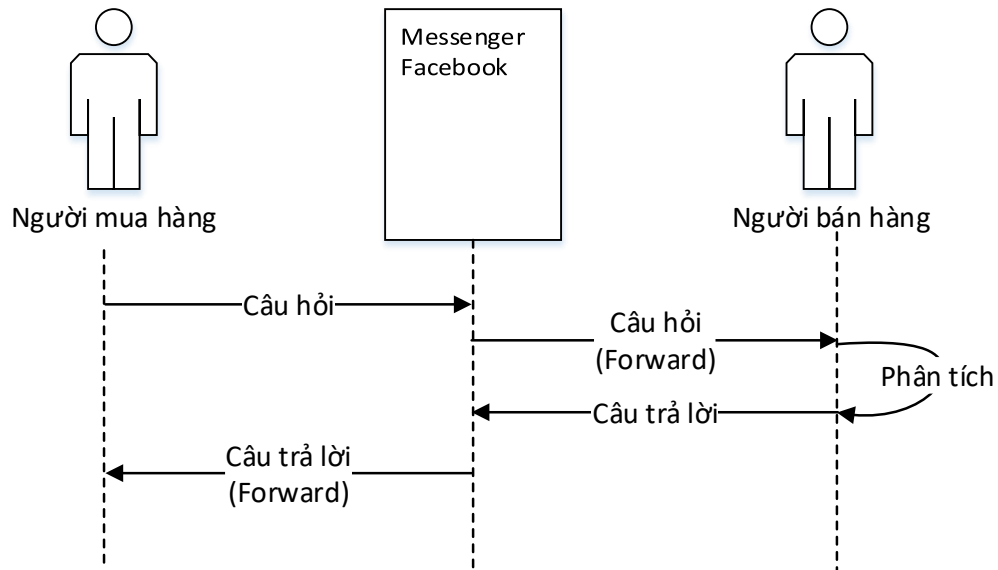
Chương này, luận văn giới thiệu một số khái niệm cơ bản. Mô hình bán hàng trên Internet là mô hình bán hàng của các doanh nghiệp, cá nhân, mà người mua hàng không cần phải đến cửa hàng cũng có thể nhận được tư vấn và mua được hàng đúng như ý muốn của mình. Các thành phần của mô hình gồm có: người mua hàng, ứng dụng hỗ trợ tư vấn, người bán hàng và những câu tư vấn. Trong giới hạn luận văn, việc xây dựng mô hình bán hàng tự động đồng nghĩa với việc xây dựng ứng dụng tư vấn có khả năng tự động trả lời bán hàng. Trong chương này giới thiệu thực trạng các mô hình trả lời bán hàng trên Internet của thế giới và Việt Nam. Phần đầu chương trình bày tổng quan bốn mô hình là Messenger, chatbotSkype, uhChat, suBiz. Từ những mô hình đó, tôi tiến hành phân nhóm, đánh giá và nêu lên các vấn đề cần cải tiến trong từng nhóm trong phần tiếp theo của chương.

### 2.1 Các mô hình trả lời bán hàng tiêu biểu

Trong quá trình nghiên cứu, tôi đã tham khảo một số mô hình trả lời bán hàng nổi tiếng trên thế giới cũng như ở Việt Nam. Tiêu biểu trong đó là bốn mô hình Messenger, chatbotSkype, uhChat và suBiz.

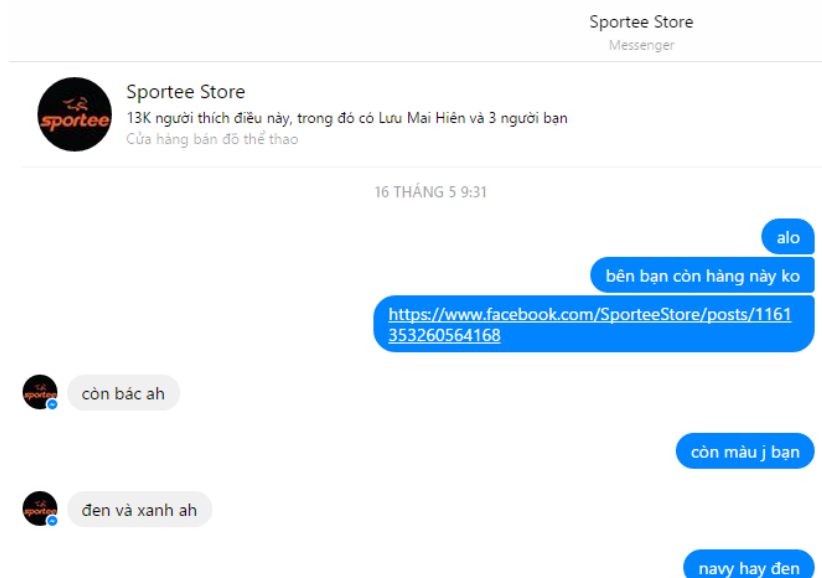
Mô hình đầu tiên là Facebook **Messenger** [1], là ứng dụng chat của Facebook, được phát hành phiên bản đầu tiên vào ngày 9 tháng 8 năm 2011 trên hệ điều hành iOS và Android. Đến ngày 11/10/2011 Messenger phát hành phiên bản sử dụng cho Blackberry OS. Tháng 12/2012, ứng dụng Facebook Messenger cho Android được đưa vào sử dụng ở vài nơi như Úc, Nam Á, Indonesia, Nam Phi, Venezuela... được tích hợp trên Facebook bằng cách tạo tên tài khoản và cung cấp số điện thoại. Thời gian sau đó, ứng dụng này liên tục được cải tiến và sử dụng rộng rãi trên các hệ điều hành. Với Facebook Messenger, người dùng có thể nhận, gửi tin nhắn nhanh chóng tới một cá nhân hay một nhóm tới bất kỳ địa chỉ liên hệ nào trong Facebook hoặc trên thiết bị điện thoại. Thêm vào đó, còn có thể xác định vị trí của mình, tạo kế hoạch với nhóm theo cách linh động nhất. Những ưu điểm nổi bật của Messenger có thể kể đến như ứng dụng nhẹ, dễ dàng cài đặt, sử dụng, hỗ trợ trên tất cả các trình duyệt và thiết bị di động thông minh, cho phép tìm kiếm lịch sử hội thoại, tự động lưu trữ và gửi thông báo tự động khi người nhận tin nhắn online, tích hợp tính năng gọi điện trực tiếp thông qua mạng Internet. Sự thông dụng của Facebook kèm với các ưu điểm trên đã khiến

Messenger trở thành một trong những ứng dụng chat được sử dụng nhiều nhất trên thế giới. Tuy nhiên, trên phương diện “mô hình trả lời bán hàng”, Messenger hoàn toàn cần sự can thiệp của tác nhân là người bán hàng trong giai đoạn tư vấn bán hàng, tức là chưa “tự động”. Ngoài trừ những hoạt động theo dõi và lưu trữ các cuộc đối thoại giữa hai bên, Facebook vẫn hoạt động theo mô hình truyền thống.



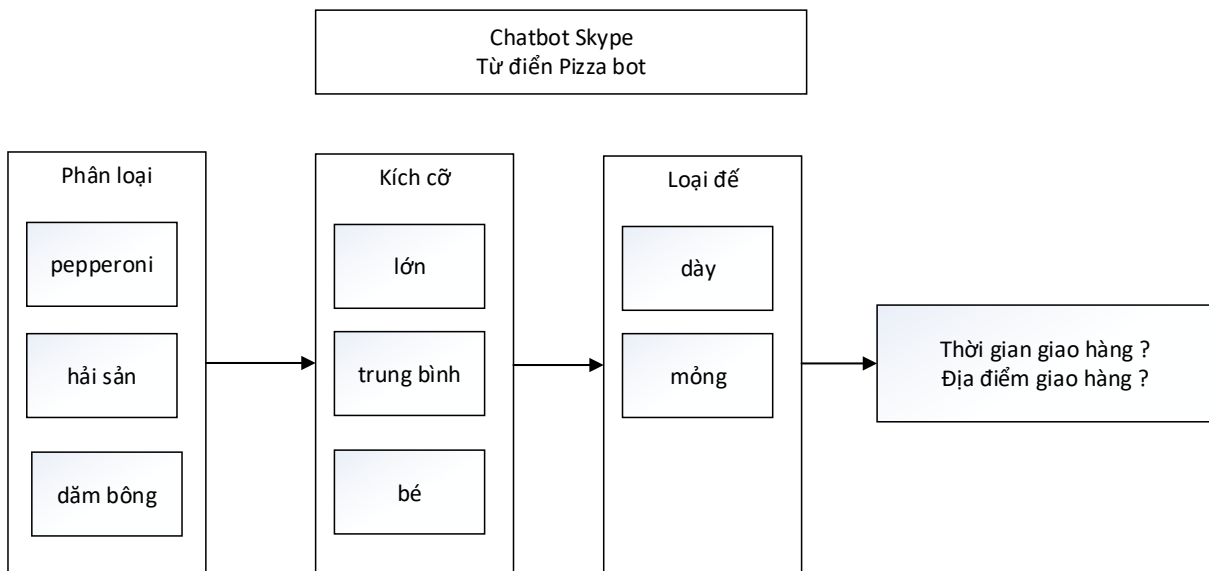
Hình 2.1: Luồng tương tác mô hình bán hàng sử dụng Messenger

Trong mô hình này, Messenger chỉ mang tính chất là một ứng dụng chat - forward những câu hỏi thoại giữa người bán hàng và người mua hàng.



Hình 2.2: Mô hình bán hàng sử dụng Messenger

Tại sự kiện Build 2016, được diễn ra tại San Francisco từ ngày 30/03/2016 đến 01/04/2016, Microsoft đã giới thiệu **chatbot Skype** [2], cho phép người mua hàng giao tiếp với một chatbot (*phần mềm tư vấn bán hàng tự động*) được lập trình sẵn. Sẽ không có tác nhân là người bán hàng tham gia vào cuộc hội thoại cho tới khi kết thúc phiên giao tiếp bán hàng. PizzaBot, demo của chatbot Skype, đây thực sự là “*mô hình trả lời bán hàng tự động*”, và chatbot này đã được huấn luyện để hiểu được ngôn ngữ con người. Ví dụ, khi người mua hàng gửi tin nhắn “*tôi muốn gọi một chiếc pizza pepperoni cỡ lớn*”, chatbot Skype sẽ bắt đầu phân tích câu nói này rồi dẫn dắt người mua hàng hoàn thiện quá trình mua hàng. Trên Messenger, khi người mua hàng gửi đi tin nhắn này, một nhân viên bán hàng của cửa hàng pizza sẽ tham gia vào cuộc hội thoại để giúp hoàn thiện đơn hàng. Nói cách khác, với Messenger thì ở phía cửa hàng chẳng có gì được tự động hóa. Quay lại với chatbot Skype, ở “*mô hình trả lời*” này sẽ có một bộ máy phân tích để tự động nhận diện các thông tin đã có trong tin nhắn được gửi đi từ khách hàng và yêu cầu họ cung cấp những thông tin liên quan còn thiếu. Để làm được điều này, Microsoft đã dựng lên một cuốn “*từ điển*” để chatbot Skype có thể tự động nhận biết các cụm từ có nghĩa trong câu nói.

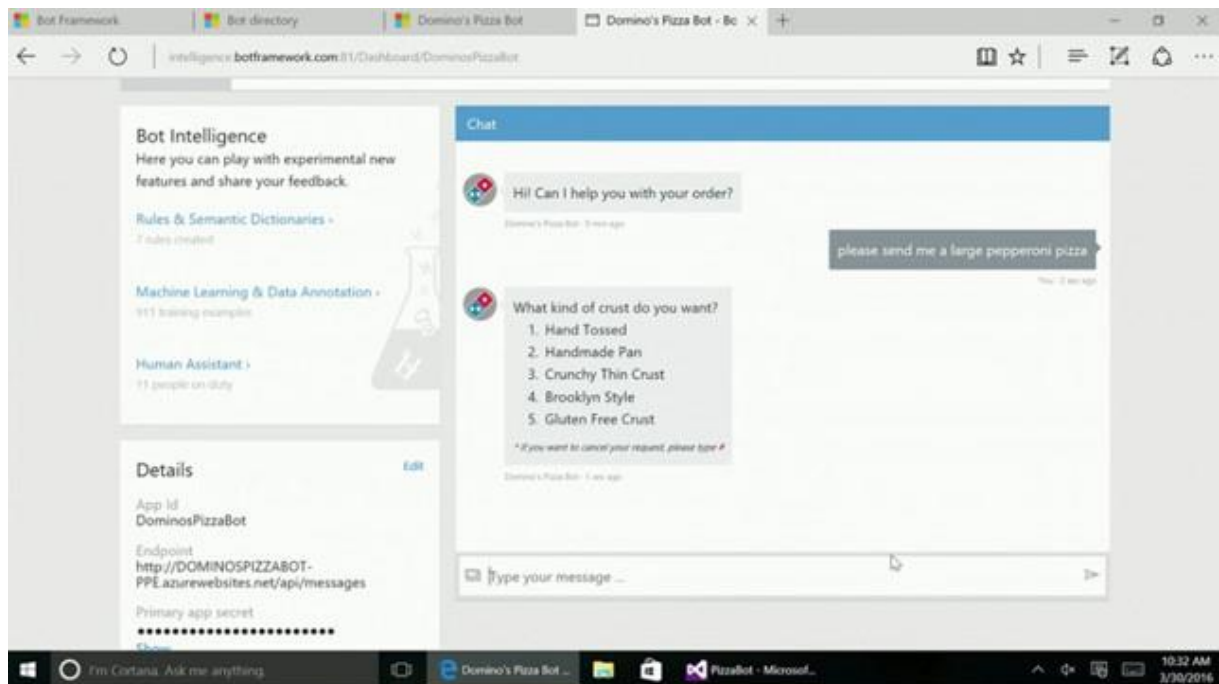


Hình 2.3: Từ điển sử dụng trong chatbot Skype

Từ điển này sẽ được phân loại thành các *nhóm dữ liệu*. Trong ví dụ ở trên, mô hình trả lời PizzaBot sẽ có 3 nhóm dữ liệu chính:

- Phân loại pizza: sẽ có các từ cho biết được “phân loại” của chiếc bánh mà khách hàng muốn đặt, có thể là “pepperoni”, “hải sản”, “dăm bông”...
- Kích cỡ pizza: gồm các từ cho biết được “kích cỡ” của chiếc bánh mà khách hàng muốn đặt, có thể là “lớn”, “trung bình”, “bé”....
- Loại đế pizza: gồm các từ cho biết được “loại đế” của chiếc bánh mà khách hàng muốn đặt, có thể là “dày”, “mỏng”...

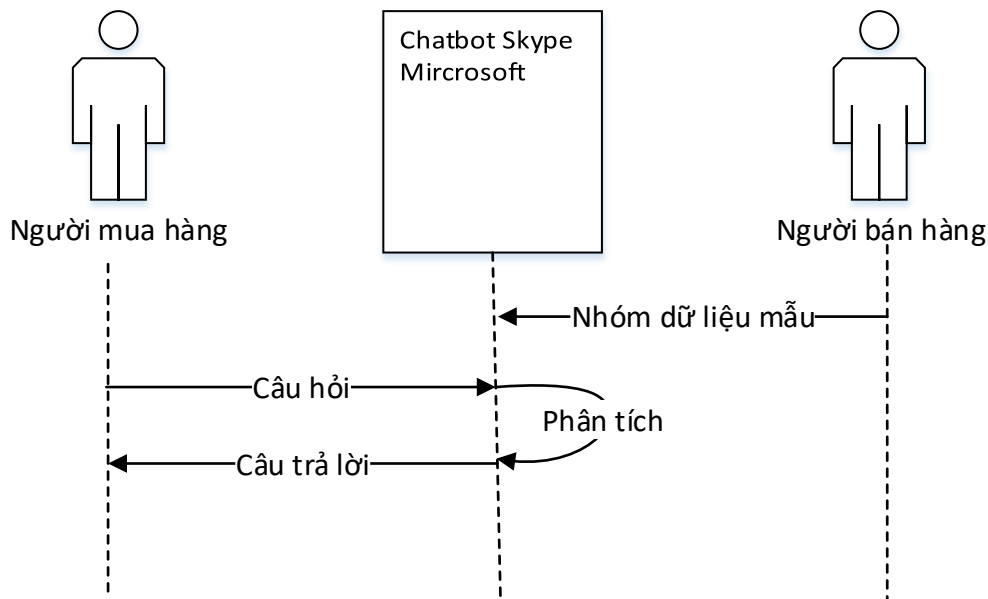
Trong quá trình “phân tích” để trả lời câu hỏi, mô hình sẽ tìm kiếm và khớp các thông tin được cung cấp vào các nhóm dữ liệu có sẵn. Nếu thông tin chưa đủ, mô hình tự động đưa ra các câu hỏi mẫu cho những thông tin còn thiếu để lấy thêm thông tin. Nếu đã đủ thông tin, mô hình đặt ra những câu hỏi chốt như thời gian, địa điểm giao hàng và kết thúc phiên tư vấn. Như câu nói ở trên, chatbot sẽ ghi nhận được 2 thông tin: loại bánh (pepperoni) và kích cỡ (lớn) rồi hỏi thông tin cuối cùng: “*bạn muốn đế dày hay mỏng?*” .



Hình 2.4: Mô hình bán hàng sử dụng chatbot Skype

Trường hợp người mua hàng đưa ra một từ mới, người bán hàng có thể nhanh chóng sắp xếp từ đó vào nhóm dữ liệu tương ứng. Khi từ điển này ngày càng tăng số từ, thì mô hình trả lời tự động ngày một thông minh hơn, và cuối cùng có thể hiểu được toàn bộ các câu lệnh đặt hàng theo chuẩn. Có thể hiểu được, trong mô hình phát triển bởi Microsoft, chatbot là thành phần thay thế người bán hàng trong pha giao tiếp. Dựa trên “nhóm dữ liệu mẫu” – tập hợp các thông tin

cần thiết và liên quan đến nhau, chatbot skype sẽ “phân tích” thay người bán hàng và có thể đưa ra thông tin trả lời.



*Hình 2.5: Mô hình bán hàng sử dụng chatbot Skype*

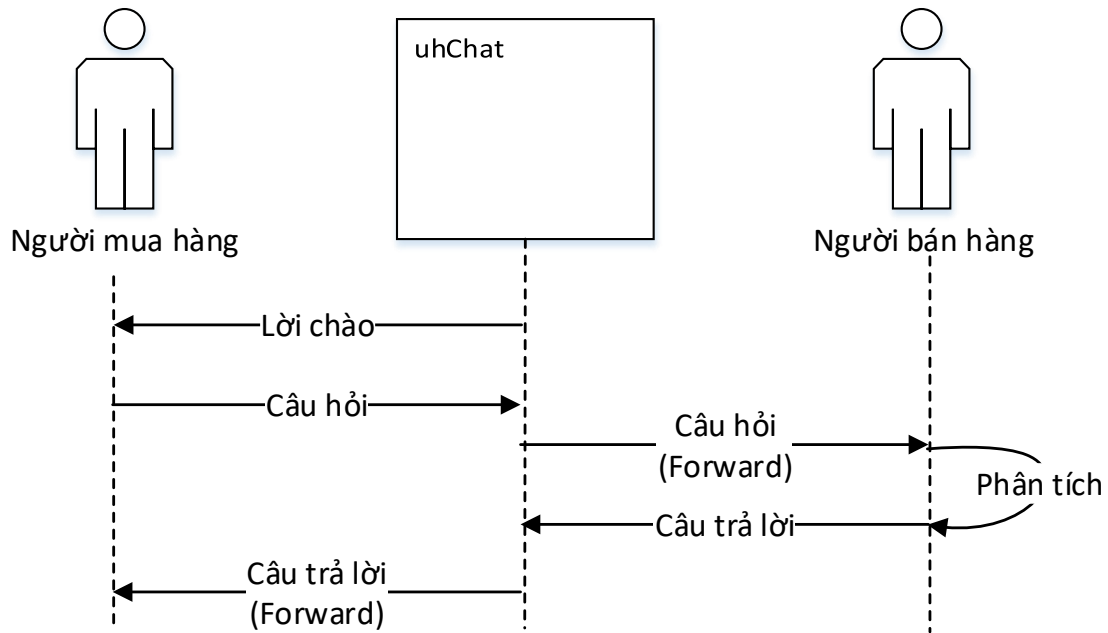
Mặc dù ở “mô hình bán hàng” Skype của Microsoft đã rất tiên bộ và “thông minh” hơn so với Messenger của Facebook, nhưng nó còn có hạn chế như cần nhập sẵn một số mẫu dữ liệu cố định để chatbot có thể truy xuất và trả lời thông qua những thông tin đó. Không trả lời được các câu hỏi mới nằm ngoài tập dữ liệu mới, cũng như giới hạn tư vấn bán hàng bị thu hẹp trong các miền nhỏ. Đặc biệt, chatbot này chưa hỗ trợ tiếng Việt. Những điều trên làm hạn chế khả năng giao tiếp của chatbot chỉ dừng lại ở một số câu chứ không thể có những đoạn hội thoại tùy biến giống như giao tiếp giữa hai con người.

Hội nhập với xu hướng chung của thế giới, từ 2010 đến nay thương mại điện tử ở Việt Nam bắt đầu phát triển mạnh mẽ và bùng nổ vào những năm 2012. Sự phổ cập Internet, kèm theo đó là sự phát triển của các website bán hàng làm cho nhu cầu mua bán trên mạng ngày một tăng cao. Nếu ở những năm đầu, website điện tử chỉ dừng lại ở nội dung bán hàng và cung cấp địa chỉ, cũng như số điện thoại để người bán và người mua có thể giao tiếp với nhau thông qua điện thoại, thì những năm gần đây, các ứng dụng chat được gắn vào website (hay có thể nói là những “mô hình trả lời bán hàng”) đã thay thế điều đó. Với lợi thế không mất phí, có thể sử dụng trên môi trường internet dù ở bất kỳ đâu, các mô hình này dần



chiếm được cảm tình của người sử dụng, giúp cho việc tương tác giữa đôi bên thuận lợi hơn.

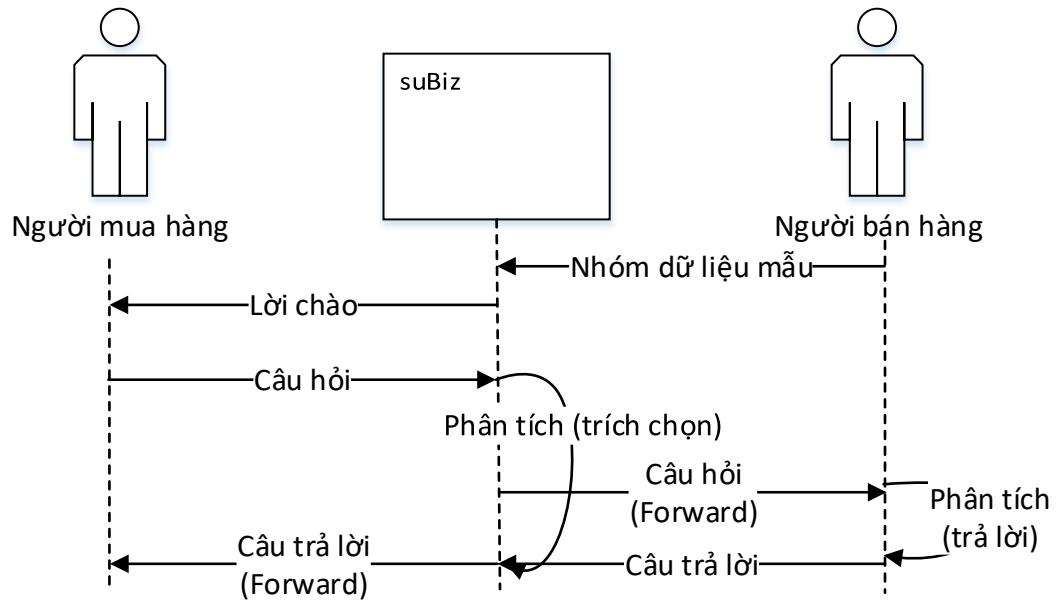
**uhChat**, một “mô hình trả lời bán hàng”, đã có một thống kê nội bộ tương đối rộng, và chứng minh được mô hình này giúp lưu lượng thông tin giao tiếp giữa khách hàng và admin của website (người bán hàng) tăng lên 6500%, tức là 65 lần. Điều này mang lại kết quả tốt đối với một trang web thương mại điện tử. Việc đăng ký và sử dụng mô hình này đơn giản, người bán hàng có thể đăng ký bằng email đang sử dụng và tạo một mật khẩu mới. Sau khi đăng ký thành công, người bán hàng sẽ nhận được một đoạn mã HTML, chỉ cần sao chép mã đó dán vào website thương mại điện tử của mình hoặc phần liên hệ cuối trang web. Mô hình trả lời này cho phép người mua hàng chỉ cần gõ nội dung và bấm gửi đến người bán hàng một cách nhanh chóng mà không cần phải đăng nhập tài khoản giống như Messenger. Việc này rút ngắn thời gian và làm cho người mua hàng cảm thấy thoải mái để chia sẻ thông tin cá nhân. Một trong các điểm mạnh của uhChat là khả năng chủ động giao tiếp với người mua hàng bằng những “lời chào” được người bán hàng định nghĩa. Mô hình cho phép thay vì phải ngồi trên máy tính và chờ khách vào để tư vấn, hộp chat tích hợp cho phép người bán hàng tùy chỉnh các câu nói tự động để giao tiếp với khách hàng trước, cho đến khi khách hàng trả lời thì một cuộc hội thoại mới thực sự bắt đầu. Việc này giúp tiết kiệm công sức của nhân viên bán hàng một cách tối đa. Thay vì hộp thoại xuất hiện ngay khi khách hàng truy cập website kèm theo tiếng thông báo, mô hình còn có tính năng tùy chỉnh thời gian xuất hiện hộp thoại, giúp tạo ấn tượng chuyên nghiệp ban đầu cho khách hàng. Khi cuộc hội thoại bắt đầu, các câu hỏi sẽ được chuyển đến người bán hàng (trong trường hợp online trực tuyến), hoặc được gửi tới email đăng ký của người bán hàng (trong trường hợp offline), người bán hàng có thể trả lời thông qua email khi có internet.



Hình 2.6: Mô hình bán hàng sử dụng *uhChat*

Rõ ràng, mô hình trả lời này vẫn chưa “tự động” thật sự, chỉ “tự động” ở mức độ kết nối với người dùng và đưa ra một số câu chào hỏi theo mẫu.

Tương tự như vậy, mô hình **subiz** cũng được phát triển theo luồng tương tác như trên. Nhưng điểm khác biệt, *subiz* cho phép người dùng đặt một “thư viện câu trả lời”, với những mẫu câu trả lời có sẵn. Thư viện này được xây dựng bởi người bán hàng. Với những câu hỏi có nội dung tương tự nhau, ứng dụng sẽ tìm kiếm và lựa chọn câu trả lời trong thư viện để trả lời. Nếu quá trình tìm kiếm không có kết quả, *subiz* sẽ đưa ra câu trả lời mặc định. Khi đó, người bán hàng cần tham gia vào cuộc hội thoại và tư vấn cho người mua hàng.



Hình 2.7: Mô hình bán hàng sử dụng Subiz

Tóm lại, các “mô hình trả lời bán hàng” hiện nay trên thế giới và Việt Nam, đa số vẫn chỉ dừng lại ở mức ứng dụng chat, là forwarder trong phiên giao tiếp bán hàng, vẫn yêu cầu sự có mặt và xử lý của người bán hàng. Chatbot Skype đã có sự tiến bộ hơn, có thể thay thế người bán hàng ở một mức độ nào đó, tuy nhiên cũng chỉ dừng lại ở những mẫu hội thoại ngắn và có nội dung đơn giản. Phần tiếp theo, tôi sẽ chia nhóm các mô hình đã tìm hiểu và phân tích điểm mạnh yếu của từng nhóm.

## 2.2 Các vấn đề cần giải quyết và cải tiến

Trong quá trình tìm hiểu các mô hình trả lời bán hàng trên, dựa trên các tính năng hỗ trợ cũng như hạn chế của từng mô hình, ta có thể tạm phân loại các mô hình làm 3 nhóm như bảng dưới.

*Bảng 2.1 Bảng phân loại mô hình bán hàng*

|  | Mô hình đơn giản     | Mô hình bán tự động              | Mô hình tự động   |
|--|----------------------|----------------------------------|-------------------|
| Ví dụ  | Messenger, uhChat... | suBiz...                         | Chatbot Skype...  |
| Mức độ phụ thuộc người bán hàng trong phiên tư vấn | Phục thuộc hoàn toàn | Chỉ cần hỗ trợ những câu hỏi mới | Không phụ thuộc   |
| Vai trò ứng dụng chat                              | Truyền tin           | Tự động với dữ liệu đã có        | Tự động hoàn toàn |
| Yêu cầu dữ liệu mẫu                                | Không                | Có                               | Có                |

Nhóm đầu tiên có thể gọi là nhóm “mô hình đơn giản”. Tiêu biểu cho mô hình này là Messenger của Facebook, uhChat.... Ở những mô hình này, vai trò quyết định đưa ra câu trả lời là người bán hàng. Người bán hàng sẽ phải tiếp nhận câu hỏi, phân tích và đưa ra câu trả lời phụ thuộc vào kinh nghiệm. Họ cần online, theo dõi màn hình, ai yêu cầu thì tư vấn. Ở đây, ứng dụng chat được tích hợp vào chỉ mang tính chất forwarder. Điểm mạnh của những mô hình này là dễ sử dụng, đơn giản trong việc tích hợp trên website thương mại đơn giản và thường hỗ trợ nhiều tính năng. Tuy nhiên, điểm yếu của các mô hình này là không tự động, ta sẽ không hướng đến nhóm này. Nhóm thứ 2 là nhóm “Mô hình bán tự động” gồm subiz và các mô hình tương tự. Với một số mẫu dữ liệu cài sẵn được nhập từ người bán hàng, và được lưu trữ vào cơ sở dữ liệu. Nếu có câu hỏi, ứng dụng chat sẽ phân tích và tìm kiếm trong tập mẫu, nếu có sẽ sử dụng làm câu trả lời, nếu không tìm thấy thì trả lại một câu mặc định nào đó. Khi đó, sự xuất hiện của người bán hàng là cần thiết để đưa ra câu trả lời hợp lý. Các thuật toán sử dụng cho mô hình này thường là thuật toán mapping. Ưu điểm lớn nhất là mô hình có thể trả lời những câu hỏi một cách hợp lý nếu tìm thấy câu trả lời trong tập mẫu. Tuy nhiên, mô hình này mang tính chất “học vẹt”. Nhóm thứ 3, là nhóm có trí thông minh hơn hẳn 2 nhóm trước. Nhóm “mô hình tự động” như chatbotSkype. Ở các mô hình tự động này, người bán hàng không cần tham gia phiên tư vấn trả lời, chatbot sẽ tự động phân tích và đưa ra câu trả lời. Điều bắt buộc ở các mô hình này là cần có sự xuất hiện của AI (Artificial intelligence), cho phép “hiểu” ngữ cảnh câu hỏi và đưa ra câu trả lời sau khi phân tích dữ liệu đầu vào và dữ liệu mẫu.

Trong 3 nhóm nêu trên, mô hình tự động là mô hình mà chúng ta sẽ nghiên cứu và hướng đến. Tuy nhiên, nếu dừng lại ở mức độ chatbot Skype thì còn có khá nhiều nhược điểm và vấn đề cần cải tiến:

Thứ nhất, vấn đề phát sinh những câu hỏi nằm ngoài “tập dữ liệu mẫu”, mô hình sẽ không trả lời, hoặc trả lời những câu mẫu được định nghĩa sẵn cho những trường hợp này.

Thứ hai, vấn đề xây dựng một mô hình có thể hiểu được ngôn ngữ tiếng Việt, mô hình chatbot Skype hoàn toàn không có khả năng này.

Thứ ba, vấn đề câu hỏi dài và phức tạp, các mô hình hiện tại đều gặp vấn đề này. Ở chatbot Skype đang dừng lại mô hình đưa ra gợi ý ngắn gọn cho người mua hàng lựa chọn, những đoạn đối thoại sinh ra thường ngắn.

Thứ tư, vấn đề tái sử dụng dữ liệu tư vấn: dữ liệu tư vấn giữa người mua hàng và người bán hàng được hầu hết các mô hình nêu trên lưu trữ lại. Tuy nhiên chỉ đáp ứng cho mục tiêu thống kê và báo cáo thì khá phí phạm, vì đây là tập dữ liệu lớn và chuẩn, có thể tái sử dụng trong các bài toán Machine Learning, Big Data.

Cuối cùng là vấn đề “tự động”: là vấn đề mấu chốt của luận văn, không chỉ dừng lại ở mức độ tự động trả lời với những câu hỏi – câu trả lời có sẵn trong cơ sở dữ liệu, mà mô hình cần tự động trong quá trình “phân tích” câu hỏi, quá trình sinh mới câu trả lời nằm ngoài dữ liệu có sẵn. Nói cách khác, mô hình có thể “suy nghĩ” và “tư vấn” như một con người, thay thế sự xuất hiện của người bán hàng trong pha tư vấn.

Rõ ràng, chúng ta cần thay đổi phương pháp sử dụng Machine Learning để cho mô hình tự học được một lượng lớn dữ liệu từ người dùng, tạo nên một bộ “tri thức” và ứng dụng trong quá trình phân tích và sinh câu trả lời, thì những vấn đề nêu trên có thể được giải quyết.

### CHƯƠNG 3: MẠNG NƠ-RON VÀ PHƯƠNG PHÁP SEQ2SEQ

Chương này giới thiệu cơ sở lý thuyết sử dụng cho luận văn, bao gồm: các kiến thức nền tảng về trí tuệ nhân tạo, học máy, mạng nơ-ron, phương pháp học chuỗi seq2seq... và lý do áp dụng những kiến thức này khi xây dựng mô hình đề xuất.

#### 3.1 Kiến thức tổng quan

Ở chương 2, các từ như “trí tuệ nhân tạo”, “học máy”, “kho dữ liệu”, “tự học”... được nhắc đến khá nhiều. Đó là những lý thuyết cần thiết khi xây dựng “mô hình bán hàng tự động”.

Trí tuệ nhân tạo (*artificial intelligence - AI*): là trí tuệ được biểu diễn bởi bất cứ một hệ thống nhân tạo nào. Thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo. Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc. Các ví dụ ứng dụng bao gồm các tác vụ điều khiển, lập kế hoạch và lập lịch (scheduling), khả năng trả lời các câu hỏi về chẩn đoán bệnh, trả lời khách hàng về các sản phẩm của một công ty, nhận dạng chữ viết tay, nhận dạng tiếng nói và khuôn mặt. Bởi vậy, trí thông minh nhân tạo đã trở thành một môn học, với mục đích chính là cung cấp lời giải cho các vấn đề của cuộc sống thực tế. Ngày nay, các hệ thống nhân tạo được dùng thường xuyên trong kinh tế, y dược, các ngành kỹ thuật và quân sự, cũng như trong các phần mềm máy tính thông dụng trong gia đình và trò chơi điện tử.

Học máy (*machine learning*): là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể “học” cách phân loại thư điện tử xem có phải thư rác (spam) hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ. Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán NP-khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được. Học máy có hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng

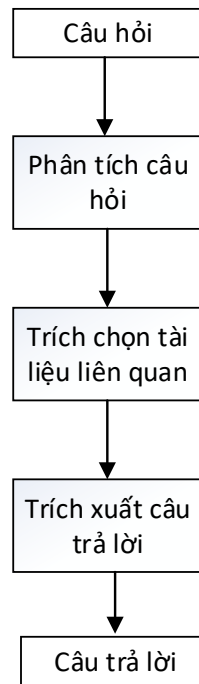
khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt (robot locomotion).

Kho dữ liệu (*data warehouse*): là một tập các dữ liệu có những đặc điểm sau: tập trung vào một chủ đề, tổng hợp từ nhiều nguồn dữ liệu khác nhau, từ nhiều thời gian, và không sửa đổi. Được dùng trong việc hỗ trợ ra quyết định trong công tác quản lý. Trong phạm trù của luận văn, kho dữ liệu được hiểu là tập dữ liệu được thu thập từ Internet, hoặc từ người bán hàng trong lĩnh vực thương mại điện tử. Kho dữ liệu sẽ là đầu vào cho quá trình tự học

Tự học (*auto learning*): là quá trình “học” tự động từ kho dữ liệu thu thập được thông qua một số phương pháp học máy nào đó. Kết quả của quá trình này phụ thuộc vào 2 yếu tố: nguồn dữ liệu đầu vào và thuật toán sử dụng để “học”.

Quay lại bài toán đặt ra ban đầu, yêu cầu tiên quyết để thực hiện mô hình là mô hình cần có sự tham gia của học máy, giúp cho ứng dụng tư vấn có thể tự học những mẫu giao tiếp thông qua kho dữ liệu người bán hàng.

Hiện tại, phương pháp xây dựng các hệ thống tự động trước đây, như chatbotSkype, thường theo quy trình 3 bước chung. Đầu tiên, phân tích câu hỏi nhằm tạo ra “*truy vấn*” cho bước trích chọn tài liệu liên quan và tìm ra *những thông tin hữu ích* cho bước trích xuất câu trả lời. Tiếp đến là trích chọn tài liệu liên quan, dựa trên câu truy vấn được tạo ra ở bước phân tích câu hỏi để tìm ra các tài liệu liên quan đến câu hỏi. Bước cuối cùng là trích xuất câu trả lời, phân tích câu trả lời từ bước trích chọn tài liệu liên quan và sử dụng các thông tin hữu ích từ bước phân tích câu hỏi để đưa ra câu trả lời phù hợp.

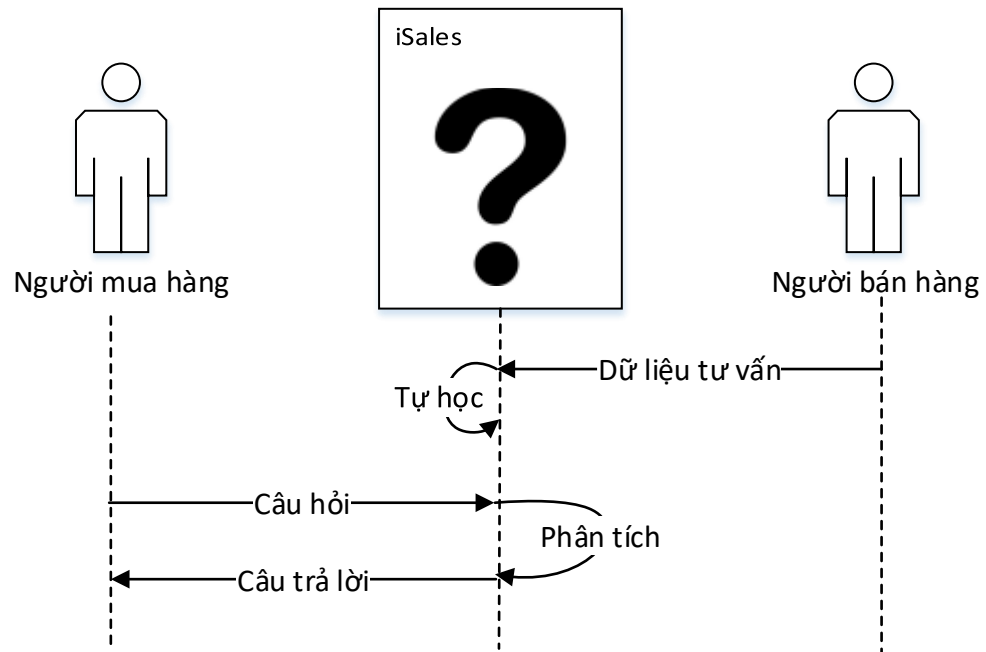


Hình 3.1: Các bước chung của mô hình tự động hiện tại

Những mô hình được xây dựng thông qua quy trình như trên đa phần tiếp cận đưa vào trích gọn thông tin (Retrieval-based). Các kỹ thuật thường sử dụng một kho đã định nghĩa trước các câu trả lời kết hợp với một vài phương pháp trích chọn Heuristic để nhặt ra một đáp án thích hợp nhất dựa vào mẫu hỏi input và ngữ cảnh. Kỹ thuật heuristic sử dụng ở đây đơn giản có thể là sự so khớp các biểu thức dựa vào luật (rule-based), hoặc phức tạp như việc kết hợp học máy để phân lớp các câu hỏi và đáp án trả về. Những hệ thống kiểu này *không sinh ra văn bản mới*, chúng chỉ nhặt một đáp án từ một tập dữ liệu cố định sẵn có. Kết quả như vậy sẽ không “thông minh”, và có hạn chế chung là không có khả năng tự động, đơn giản nhất nếu không trích chọn được tài liệu liên quan, mô hình sẽ trả về giá trị mặc định được cài đặt sẵn hoặc không có câu trả lời. Chatbot Skype của Microsoft được nhắc tới trong chương 1 là ví dụ điển hình.

Làm chủ được Machine Learning là một bài toán khó, khó hơn nữa là ứng dụng vào bài toán cụ thể, điều đó dẫn tới các mô hình bán hàng hiện tại đa phần chỉ dừng lại ở thao tác trực tiếp người dùng chứ không có một “bộ óc nhân tạo” xử lý. Vậy để xây dựng được mô hình đề xuất như trên, ta cần có ít nhất 3 pha:





Hình 3.2: Mô hình bán hàng tự động

- Pha thu thập dữ liệu: cần có một phương pháp thu thập dữ liệu từ các nguồn website trên mạng. Yêu cầu đầu tiên cho tập dữ liệu này là khối dữ liệu cần lớn, được thu thập từ nhiều nguồn và trong lĩnh vực bán hàng. Pha này có thể sử dụng những dữ liệu lưu trữ được trong quá trình tư vấn, giải quyết được vấn đề tái sử dụng dữ liệu, ngoài ra đây còn là nguồn dữ liệu “sạch”, “chuẩn” cho pha tự học của mô hình.
- Pha tự học: muốn cho mô hình có thể trả lời tự động, hay là có thể tư vấn mua hàng, thì mô hình cần phải có khả năng “tự học”. Cũng giống như một con người, quá trình học là một quá trình lâu dài và không thể có kết quả tốt trong thời gian ngắn. Việc xây dựng “tự học” cho mô hình bắt buộc phải sử dụng Machine Learning. Và cần phải có một giải pháp cụ thể để mô hình có thể hiểu được ngôn ngữ “tiếng Việt”.
- Pha phân tích và trả lời: “tự học” có thể đem lại cho mô hình các “tri thức”, nhưng vẫn chưa thể áp dụng ngay, vì một vấn đề đặt ra cho mô hình này là bài toán “*câu hỏi dài và phức tạp*”. Nếu chỉ dừng lại ở mức độ tự học và ứng dụng với những mẫu câu có sẵn, thì mô hình sẽ lặp lại vấn đề “*phát sinh câu hỏi nằm ngoài tập dữ liệu*”. Và để giải quyết triệt để, mô hình cần có thêm pha “*phân tích*”, sẽ cho phép phân tích dữ liệu đầu vào, và sản sinh ra các câu trả lời mới.

Trong thời gian nghiên cứu và ứng dụng Machine Learning vào mô hình hệ thống, tôi đã tiếp cận với TensorFlow, một nền tảng trí tuệ nhân tạo sử dụng kỹ thuật Deep Learning của Google. Sự kết hợp của TensorFlow trong mô hình đưa ra phương pháp mới cho những hệ thống bán hàng tự động. Thay vì việc bị động học những câu trả lời được cài đặt từ con người, hệ thống sẽ “tự học” cách trả lời sau quá trình huấn luyện.

### 3.2 Framework TensorFlow

**TensorFlow** [3], một nền tảng trí tuệ nhân tạo được Google phát hành theo giấy phép mã nguồn mở Apache 2.0 vào ngày 09/11/2015. Nền tảng này sử dụng kỹ thuật Machine Learning để học hỏi, nhận biết hình ảnh, giọng nói và các dấu vết dữ liệu. Hiện TensorFlow đang được sử dụng cho chức năng điều khiển giọng nói trong các ứng dụng Google, tìm hình ảnh trong Photos, và mới đây nhất là chức năng trả lời tự động Smart Reply của app email Inbox. Trước TensorFlow, Google cũng từng thiết lập một hệ thống Machine Learning thế hệ thứ 1 mang tên DistBelief. Tuy nhiên, DistBelief bị trói buộc nhiều vào hạ tầng kỹ thuật của Google, lại khá nặng nề và khó mở rộng. Trong khi đó, TensorFlow thì không còn bị ràng buộc gì về mặt hạ tầng nữa, và nó có khả năng chạy trên hầu hết mọi thứ, từ các app smartphone cho đến phần mềm trên server siêu mạnh. Nếu so với hệ thống DistBelief, TensorFlow là một khái niệm xa hơn bằng cách sử dụng công nghệ máy học chuyên sâu (Deep Learning) và mạng nơ-ron tái phát (Recurrent Neural Network) gồm nhiều lớp. Về cơ bản, TensorFlow chứa tất cả các lớp dữ liệu, gọi là nút (Node), để tìm hiểu và xác minh một vấn đề. Lớp đầu tiên sẽ yêu cầu hệ thống tìm kiếm một cái gì đó như xác định mẫu chung của vấn đề. Hệ thống sau đó di chuyển để tập hợp các dữ liệu tiếp theo, chẳng hạn như tìm kiếm và xác minh một phần của vấn đề đó. Việc di chuyển hệ thống từ nút tới nút được thực hiện để biên dịch đầy đủ thông tin được đưa ra để xác minh và khẳng định vấn đề đó. “Dòng chảy” tensor như vậy đã tạo nên cái tên TensorFlow. Cũng chính Google đã phải nhận định: *“TensorFlow nhanh hơn, thông minh hơn và linh hoạt hơn hệ thống cũ của chúng tôi (DistBelief), do đó nó có thể được điều chỉnh dễ dàng hơn với các sản phẩm mới và giúp quá trình nghiên cứu diễn ra thuận lợi hơn”*. [4]

Những đặc điểm nổi bật nhất của Tensor Flow có thể kể đến như khả năng sử dụng trên các platform khác nhau từ smartphone, pc tới distributed servers. Thư viện này có thể được chạy với CPU hoặc cùng với GPU, cùng với danh sách API rất dễ sử dụng, cho phép tính toán nhanh chóng trên mạng nơ-ron. Một số

các mã nguồn có sẵn trong thư viện Tensor Flow là nhận diện ảnh với mạng convolutional neural network, học ngữ nghĩa của từ với word2vec, mô hình hóa ngôn ngữ với recurrent neural network, máy phiên dịch và chatbot với seq2seq.

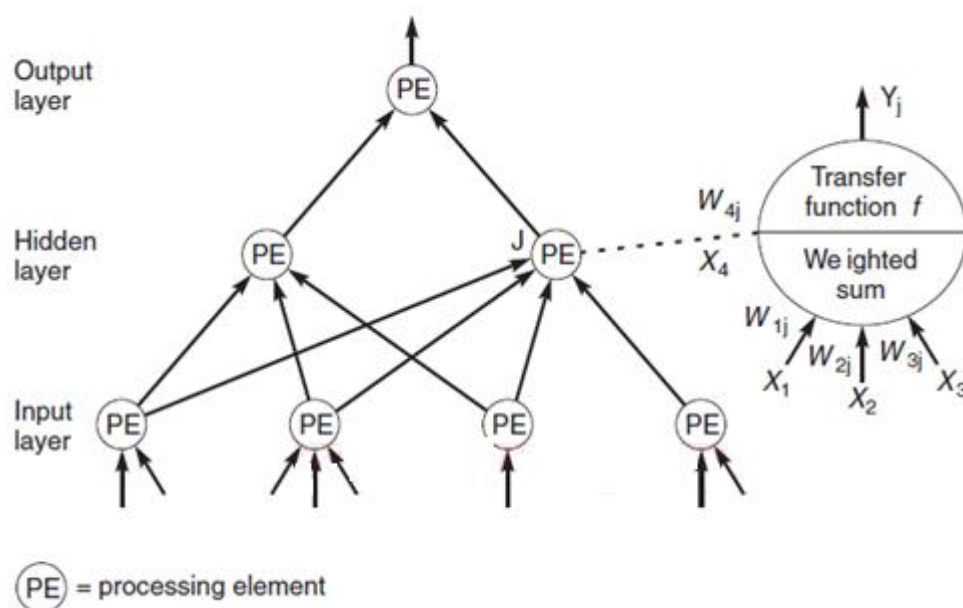
Ứng dụng của TensorFlow trong bài toán “*xây dựng mô hình bán hàng tự động trên Internet*” là sử dụng phương pháp học chuỗi liên tiếp seq2seq để huấn luyện dữ liệu thu thập từ người bán hàng thành một mô hình ngôn ngữ, từ đó sử dụng mô hình ngôn ngữ để “*sinh*” câu trả lời cho câu hỏi đầu vào.

### 3.3 Lý thuyết mạng nơ-ron

Để có thể hiểu về mạng phương pháp học chuỗi liên tiếp Seq2Seq, cần có những lý thuyết căn bản về mạng nơ-ron. Khái niệm này được bắt đầu vào cuối thập kỷ 1800 khi các nhà nghiên cứu cố gắng mô tả hoạt động của trí tuệ con người. Ý tưởng này bắt đầu được áp dụng cho các mô hình tính toán từ mạng Perceptron.

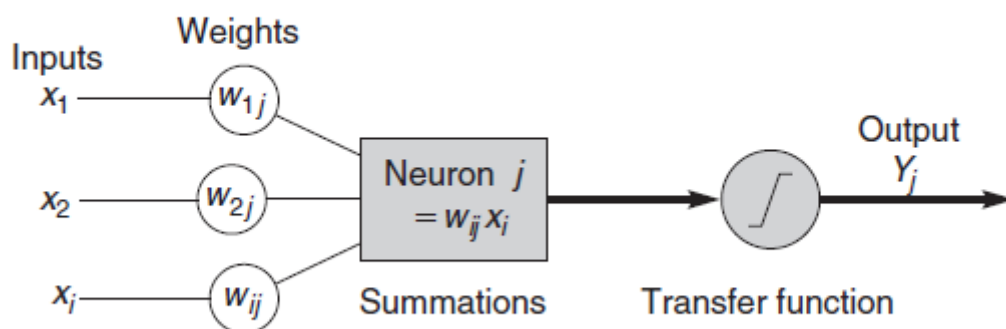
#### 3.3.1 Mạng nơ-ron nhân tạo ANN

Mạng nơ-ron nhân tạo (Artificial Neural Network – ANN) hay thường được gọi ngắn gọn là mạng nơ-ron là một mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các nơ-ron được gắn kết để xử lý thông tin. ANN được giới thiệu năm 1943 bởi nhà thần kinh học Warren McCulloch và nhà logic học Walter Pitts, nó hoạt động giống như bộ não của con người, được học bởi kinh nghiệm (thông qua việc huấn luyện), có khả năng lưu giữ các tri thức và sử dụng các tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data) [5]. Processing Elements (PE) của ANN gọi là nơ-ron, nhận các dữ liệu vào (inputs) xử lý chúng và cho ra một kết quả (output) duy nhất. Kết quả xử lý của một nơ-ron có thể làm input cho các nơ-ron khác.



Hình 3.3: Kiến trúc mạng nơ-ron nhân tạo ANN

Kiến trúc chung của một ANN gồm 3 thành phần đó là input layer, hidden layer và output layer. Trong đó, lớp ẩn (hidden layer) gồm các nơ-ron, nhận dữ liệu input từ các nơ-ron ở lớp trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Quá trình xử lý thông tin của một ANN như sau:



Hình 3.4: Quá trình xử lý thông tin của nơ-ron  $j$  trong mạng ANN

Trong đó, mỗi input tương ứng với 1 thuộc tính của dữ liệu. Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi input là một thuộc tính của khách hàng như thu nhập, nghề nghiệp, tuổi, số con... Output là một giải pháp cho một vấn đề, ví dụ như với bài toán xem xét chấp nhận cho khách hàng vay tiền hay không thì output là yes - cho vay hoặc no

- không cho vay. Trọng số liên kết (Connection Weights) là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng hay có thể hiểu là độ mạnh của dữ liệu đầu vào đối với quá trình xử lý thông tin, chuyển đổi dữ liệu từ layer này sang layer khác. Quá trình học (Learning Processing) của ANN thực ra là quá trình điều chỉnh các trọng số (Weight) của các input data để có được kết quả mong muốn. Hàm tổng (Summation Function) cho phép tính tổng trọng số của tất cả các input được đưa vào mỗi nơ-ron. Hàm tổng của một nơ-ron đối với  $n$  input được tính theo công thức sau:

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

Kết quả trên cho biết khả năng kích hoạt của nơ-ron đó. Các nơ-ron này có thể sinh ra một output hoặc không trong ANN, hay nói cách khác rằng có thể output của 1 nơ-ron có thể được chuyển đến layer tiếp trong mạng nơ-ron hoặc không là do ảnh hưởng bởi hàm chuyển đổi (Transfer Function). Việc lựa chọn Transfer Function có tác động lớn đến kết quả của ANN. Vì kết quả xử lý tại các nơ-ron là hàm tính tổng nên đôi khi rất lớn, nên transfer function được sử dụng để xử lý output này trước khi chuyển đến layer tiếp theo. Hàm chuyển đổi phi tuyến được sử dụng phổ biến trong ANN là sigmoid (logical activation) function.

$$Y_T = 1/(1 + e^{-Y})$$

Kết quả của Sigmoid Function thuộc khoảng  $[0, 1]$  nên còn gọi là hàm chuẩn hóa (Normalized Function). Đôi khi thay vì sử dụng hàm chuyển đổi, ta sử dụng giá trị ngưỡng (Threshold value) để kiểm soát các output của các nơ-ron tại một layer nào đó trước khi chuyển các output này đến các layer tiếp theo. Nếu output của một nơ-ron nào đó nhỏ hơn Threshold thì nó sẽ không được chuyển đến Layer tiếp theo. Ứng dụng thực tế của mạng nơ-ron thường được sử dụng trong các bài toán nhận dạng mẫu như nhận dạng chữ cái quang học (Optical character recognition), nhận dạng chữ viết tay, nhận dạng tiếng nói, nhận dạng khuôn mặt.

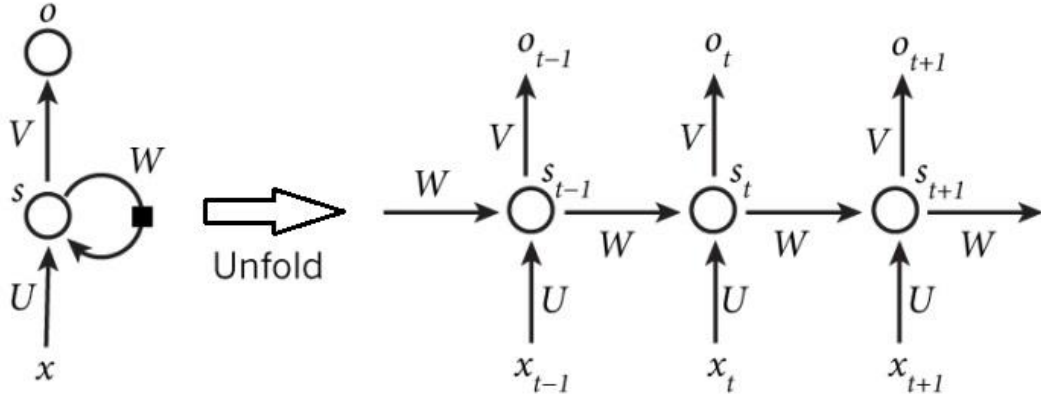
### 3.3.2 Mạng nơ-ron tái phát RNN

Mạng nơ-ron tái phát (RNN - Recurrent Neural Network) là một trong những mô hình Deep Learning được đánh giá có nhiều ưu điểm trong các tác vụ

xử lý ngôn ngữ tự nhiên (NLP). Ý tưởng của RNN đó là thiết kế một mạng nơ-ron sao cho có khả năng xử lý được thông tin dạng chuỗi như câu hỏi. Recurrent có nghĩa là thực hiện lặp lại cùng một tác vụ cho mỗi thành phần trong chuỗi. Trong đó, kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán của các thành phần ở những thời điểm trước đó. Nói cách khác, RNNs là một mô hình có trí nhớ, có khả năng nhớ được thông tin đã tính toán trước đó, không như các mạng nơ-ron truyền thống đó là thông tin đầu vào (input) hoàn toàn độc lập với thông tin đầu ra (output). Về lý thuyết, RNNs có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.

Các ứng dụng của RNN có khá nhiều trong các lĩnh vực như mô hình ngôn ngữ và phát sinh văn bản (Generating text). Mô hình ngôn ngữ cho ta biết xác suất của một câu trong một ngôn ngữ là bao nhiêu. Đây cũng là bài toán dự đoán xác suất từ tiếp theo của một câu cho trước là bao nhiêu. Từ bài toán này, ta có thể mở rộng thành bài toán phát sinh văn bản (generating text/generative model). Mô hình này cho phép ta phát sinh văn bản mới dựa vào tập dữ liệu huấn luyện. Ví dụ, khi huấn luyện mô hình này bằng các dữ liệu tư vấn bán hàng, ta có thể phát sinh được các câu trả lời cho các câu hỏi liên quan đến thương mại điện tử. Tùy theo loại dữ liệu huấn luyện, ta sẽ có nhiều loại ứng dụng khác nhau. Trong mô hình ngôn ngữ, input là chuỗi các từ (được mã hoá thành one-hot vector [13]), output là chuỗi các từ dự đoán được từ mô hình này. Một lĩnh vực khác của RNN là Dịch máy (Machine Translation). Bài toán dịch máy tương tự như mô hình ngôn ngữ. Trong đó, input là chuỗi các từ của ngôn ngữ nguồn (ví dụ tiếng Việt), output là chuỗi các từ của ngôn ngữ đích (ví dụ tiếng Anh). Điểm khác biệt ở đây đó là output chỉ có thể dự đoán được khi input đã hoàn toàn được phân tích. Điều này là do từ được dịch ra phải có đầy đủ thông tin của các từ trước đó. Hoặc RNN có thể áp dụng cho các bài toán phát sinh mô tả cho ảnh (Generating Image Descriptions). RNN kết hợp với Convolution Neural Networks có thể phát sinh ra được các đoạn mô tả cho ảnh. Mô hình này hoạt động bằng cách tạo ra những câu mô tả từ các features rút trích được trong bức ảnh.

Huấn luyện RNN tương tự như huấn luyện ANN truyền thống. Giá trị tại mỗi output không chỉ phụ thuộc vào kết quả tính toán của bước hiện tại mà còn phụ thuộc vào kết quả tính toán của các bước trước đó.



Hình 3.5: Quá trình xử lý thông tin trong mạng RNN

RNN có khả năng biểu diễn mối quan hệ phụ thuộc giữa các thành phần trong chuỗi. Ví dụ, nếu chuỗi đầu vào là một câu có 5 từ thì RNN này sẽ unfold (dàn ra) thành RNN có 5 layer, mỗi layer tương ứng với mỗi từ, chỉ số của các từ được đánh từ 0 tới 4. Trong hình vẽ ở trên,  $x_t$  là input (one-hot vector) tại thời điểm thứ  $t$ .  $S_t$  là hidden state [14] (memory) tại thời điểm thứ  $t$ , được tính dựa trên các hidden state trước đó kết hợp với input của thời điểm hiện tại với công thức:

$$S_t = \tanh(Ux_t + Ws_{t-1})$$

$S_{-1}$  là hidden state được khởi tạo là một vector không.  $O_t$  là output tại thời điểm thứ  $t$ , là một vector chứa xác suất của toàn bộ các từ trong từ điển.

$$O_t = \text{softmax}(Vs_t)$$

Không như ANN truyền thống, tại mỗi layer cần phải sử dụng một tham số khác, RNNs chỉ sử dụng một bộ parameters ( $U$ ,  $V$ ,  $W$ ) cho toàn bộ các bước. Trên lý thuyết, xây dựng một RNN có thể nhớ được thông tin của một chuỗi dài vô tận là có thể, nhưng thực tế xây dựng được như vậy là khó thực hiện vì điều kiện phần cứng và thuật toán chưa cho phép. Trong vài năm qua, các nhà nghiên cứu đã phát triển nhiều loại RNNs ngày càng tinh vi để giải quyết các mặt hạn chế của RNN.

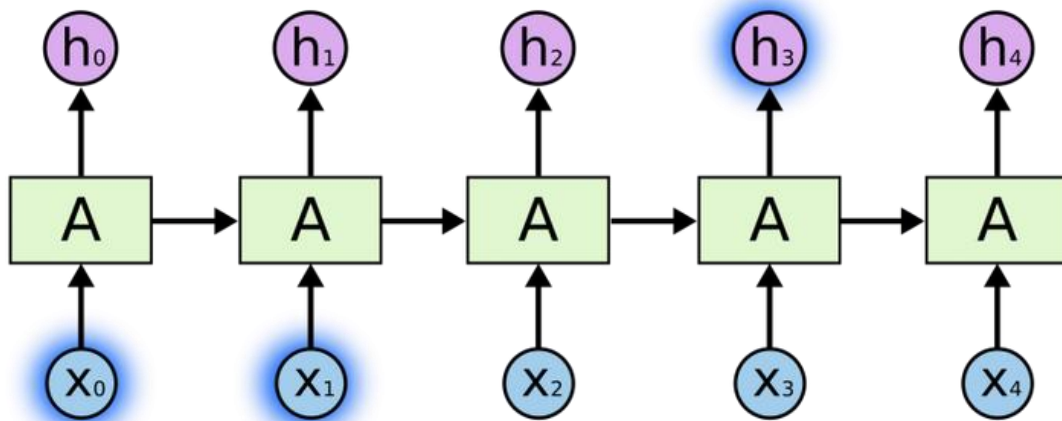
- Bidirectional RNN: dựa trên ý tưởng output tại thời điểm  $t$  không chỉ phụ thuộc vào các thành phần trước đó mà còn phụ thuộc vào các thành phần trong tương lai. Ví dụ, để dự đoán một từ bị thiếu (missing word) trong chuỗi, ta cần quan sát các từ bên trái và bên phải xung quanh từ đó. Mô

hình này chỉ gồm hai RNNs nạp chồng lên nhau. Trong đó, các hidden state được tính toán dựa trên cả hai thành phần bên trái và bên phải của mạng.

- Long short-term memory networks (LSTM): mô hình này có cấu trúc tương tự như RNNs nhưng có cách tính toán khác đối với các hidden layer. Memory trong LSTMs được gọi là cells (hạt nhân). Ta có thể xem đây là một hộp đen nhận thông tin đầu vào gồm hidden state trước đó và giá trị  $x_t$ . Bên trong các hạt nhân này, sẽ quyết định thông tin nào cần lưu lại và thông tin nào cần xóa đi, nhờ vậy mà mô hình này có thể lưu trữ được thông tin dài hạn.

### 3.3.3 Mạng Long Short Term Memory LSTM

Ý tưởng ban đầu của RNN là kết nối những thông tin trước đó nhằm hỗ trợ cho các xử lý hiện tại. Nhưng đôi khi, chỉ cần dựa vào một số thông tin gần nhất để thực hiện tác vụ hiện tại. Ví dụ, trong mô hình hóa ngôn ngữ, chúng ta cố gắng dự đoán từ tiếp theo dựa vào các từ trước đó. Nếu chúng ta dự đoán từ cuối cùng trong câu “đám\_mây bay trên bầu\_trời”, thì chúng ta không cần truy tìm quá nhiều từ trước đó, ta có thể đoán ngay từ tiếp theo sẽ là “bầu\_trời”. Trong trường hợp này, khoảng cách tới thông tin liên quan được rút ngắn lại, mạng RNN có thể học và sử dụng các thông tin quá khứ.

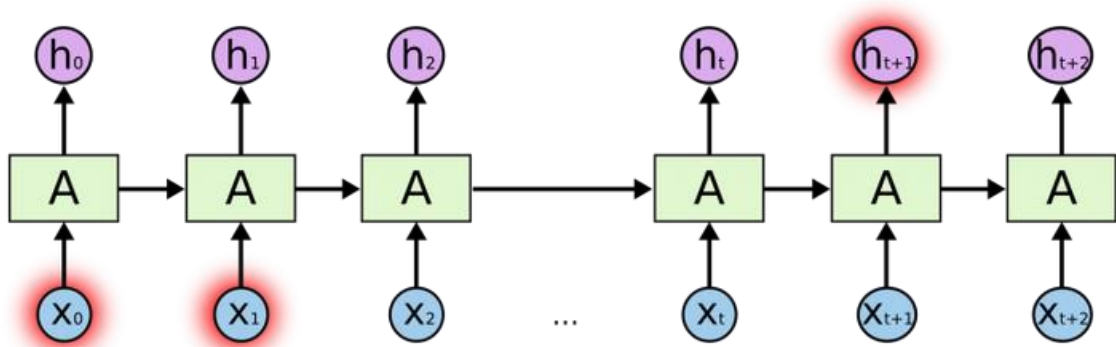


Hình 3.6: RNN phụ thuộc short-term.

Nhưng cũng có trường hợp chúng ta cần nhiều thông tin hơn, nghĩa là phụ thuộc vào ngữ cảnh. Ví dụ nhưng khi dự đoán từ cuối cùng trong đoạn văn bản “Tôi sinh ra và lớn lên ở Việt\_Nam ... Tôi có\_thể nói thuần\_thực Tiếng\_Việt.” Từ thông tin gần nhất cho thấy rằng từ tiếp theo là tên một ngôn ngữ, nhưng khi chúng ta muốn biết cụ thể ngôn ngữ nào, thì cần quay về quá khứ xa hơn, để tìm



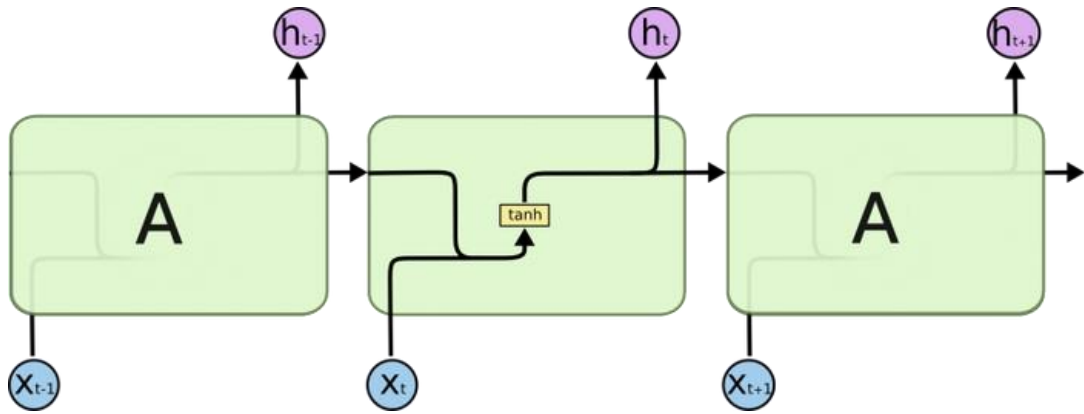
được ngữ cảnh Việt\_Nam. Và như vậy, RNN có thể phải tìm những thông tin có liên quan và số lượng các điểm đó trở nên rất lớn. Không được như mong đợi, RNN không thể học để kết nối các thông tin lại với nhau.



Hình 3.7: RNN phụ thuộc long-term.

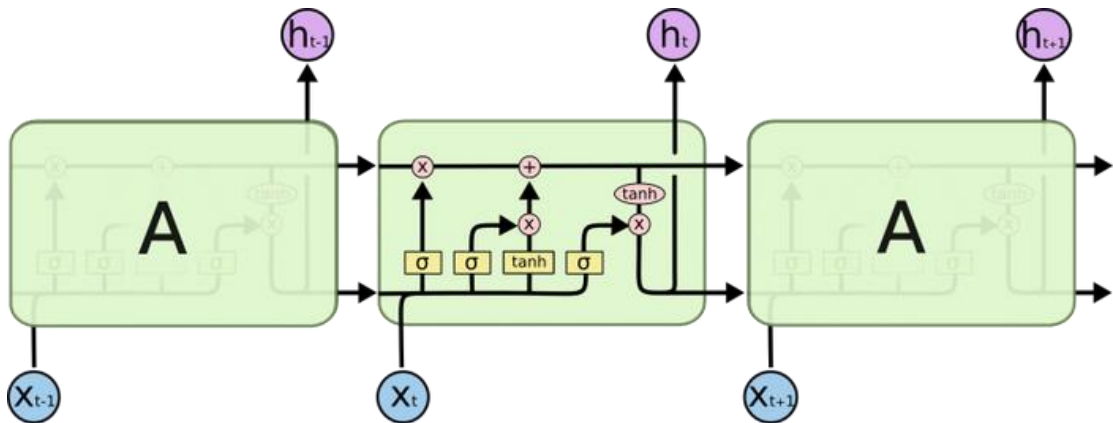
Hạn chế ở mạng RNN là hidden layer không có trí nhớ dài hạn, hay còn được nhắc tới với tên *vanishing/exploding gradient problem* [7]. Nếu chỉ dừng lại ở việc áp dụng phương án học chuỗi với RNN thì vấn đề độ dài câu đối thoại sẽ là một bài toán khó. Bằng việc cải tiến, bổ sung các module nhớ cho RNN, LSTM (Long Short Term Memory network) [8] – một là trường hợp đặc biệt của RNN được tích hợp sẵn trong phương pháp học chuỗi seq2seq, đã giải quyết được vấn đề này.

Long Short Term Memory là trường hợp đặc biệt của RNN, có khả năng học long-term dependencies. Mô hình này được giới thiệu bởi Hochreiter & Schmidhuber (1997) [8], và được cải tiến lại. Sau đó, mô hình này dần trở nên phổ biến nhờ vào các công trình nghiên cứu gần đây. Mô hình này có khả năng tương thích với nhiều bài toán nên được sử dụng rộng rãi ở các ngành liên quan. LSTM được thiết kế nhằm loại bỏ vấn đề phụ thuộc quá dài. Ta quan sát lại mô hình RNN bên dưới, các layer đều mắc nối với nhau. Trong RNN chuẩn, module repeating này có cấu trúc rất đơn giản chỉ gồm một lớp đơn giản tanh layer.



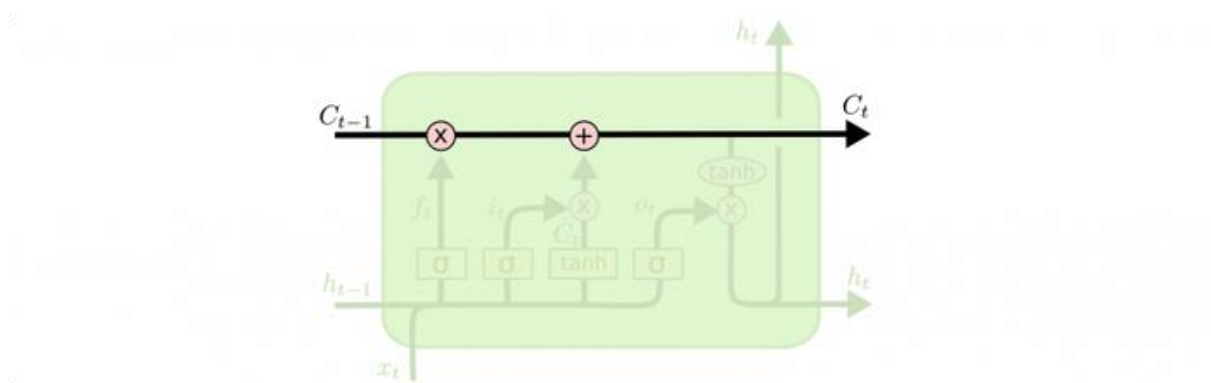
Hình 3.8: Các module lặp của mạng RNN chứa một layer.

LSTM cũng có cấu trúc mắt xích tương tự, nhưng các module lặp có cấu trúc khác hẳn. Thay vì chỉ có một layer neural network, thì LSTM có tới bốn layer, tương tác với nhau theo một cấu trúc cụ thể.



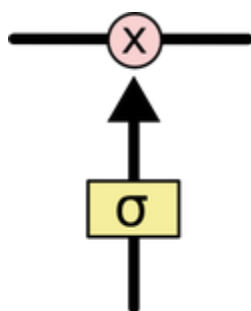
Hình 3.9: Các mô-đun lặp của mạng LSTM chứa bốn layer.

Trong đó, các ký hiệu sử dụng trong mạng LSTM gồm có: hình chữ nhật là các lớp ẩn của mạng nơ-ron, 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 từ vị trí này sang vị trí khác. Mô hình thiết kế của LSTM là một bảng mạch số, gồm các mạch logic và các phép toán logic trên đó. Thông tin, hay nói khác hơn là tần số của dòng điện di chuyển trong mạch sẽ được lưu trữ, lan truyền theo cách thiết kế bảng mạch. Mấu chốt của LSTM là cell state (trạng thái nhớ), đường kẻ ngang chạy dọc ở trên top diagram. Cell state giống như băng chuyền, chạy xuyên thẳng toàn bộ mắt xích, chỉ một vài tương tác nhỏ tuyến tính (minor linear interaction) được thực hiện. Điều này giúp cho thông tin ít bị thay đổi xuyên suốt quá trình lan truyền.



Hình 3.10: Cell state của LSTM giống như một băng truyền.

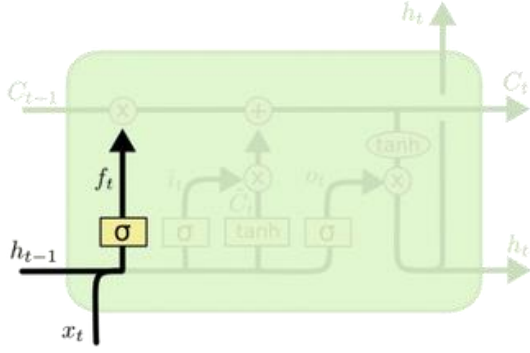
LSTM có khả năng thêm hoặc bớt thông tin vào cell state, được quy định một cách cẩn thận bởi các cấu trúc gọi là cổng (gate). Các cổng này là một cách (tùy chọn) để định nghĩa thông tin băng qua. Chúng được tạo bởi hàm sigmoid và một toán tử nhân pointwise.



Hình 3.11: Cổng trạng thái LSTM.

Hàm kích hoạt Sigmoid có giá trị từ 0 – 1, mô tả độ lớn thông tin được phép truyền qua tại mỗi lớp mạng. Nếu ta thu được zero điều này có nghĩa là “không cho bất kỳ cái gì đi qua”, ngược lại nếu thu được giá trị là một thì có nghĩa là “cho phép mọi thứ đi qua”. Một LSTM có ba cổng như vậy để bảo vệ và điều khiển cell state.

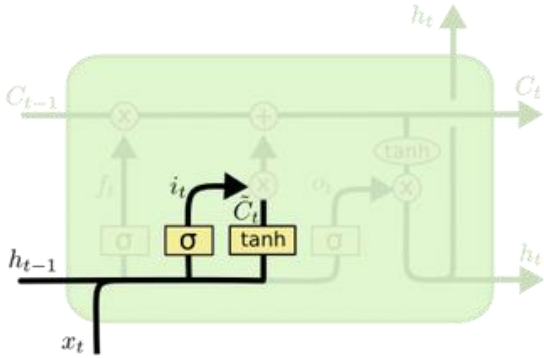
Quá trình hoạt động của LSTM được thông qua các bước cơ bản sau. Bước đầu tiên của mô hình LSTM là quyết định xem thông tin nào chúng ta cần loại bỏ khỏi cell state. Tiến trình này được thực hiện thông qua một sigmoid layer gọi là “forget gate layer” – cổng chặn. Đầu vào là  $h_{t-1}$  và  $x_t$ , đầu ra là một giá trị nằm trong khoảng  $[0, 1]$  cho cell state  $C_{t-1}$ . 1 tương đương với “giữ lại thông tin”, 0 tương đương với “loại bỏ thông tin”.



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

Hình 3.12: LSTM focus  $f$ .

Bước tiếp theo, cần quyết định thông tin nào cần được lưu lại tại cell state. Ta có hai phần là single sigmoid layer được gọi là “input gate layer” quyết định các giá trị chúng ta sẽ cập nhật. Tiếp theo, một tanh layer tạo ra một vector ứng viên mới  $\tilde{C}_t$  được thêm vào trong cell state.

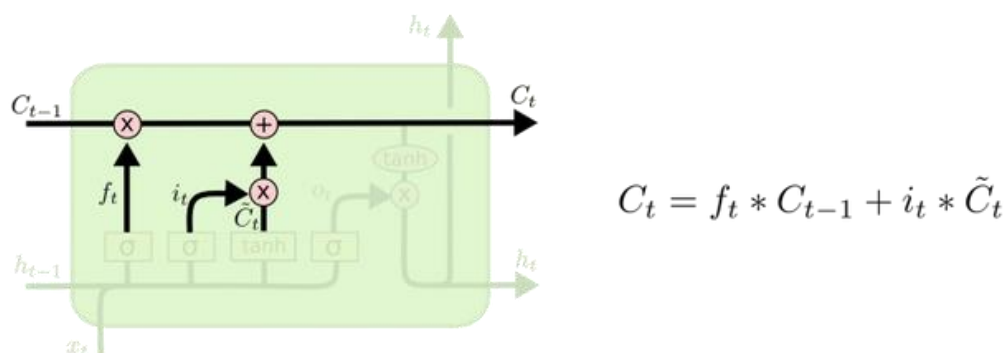


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

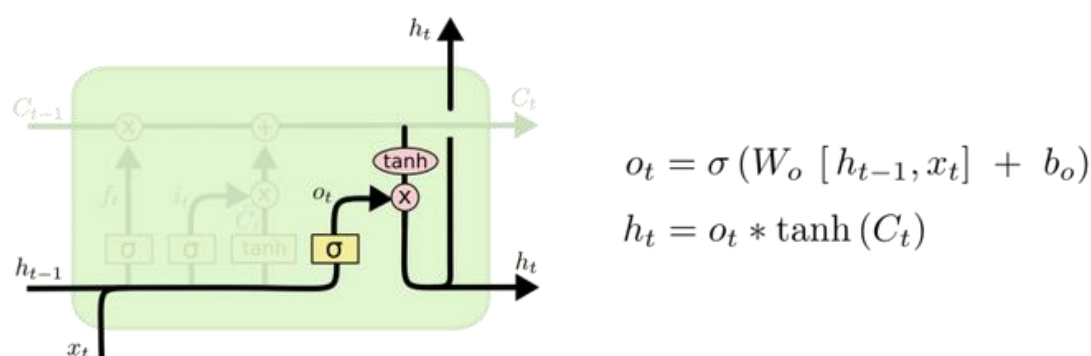
Hình 3.13: LSTM focus  $i$ .

Ở bước tiếp theo, sẽ kết hợp hai thành phần này lại để cập nhật vào cell state. Lúc cập nhật vào cell state cũ,  $C_{t-1}$ , vào cell state mới  $C_t$ . Ta sẽ đưa state cũ hàm  $f_t$ , để quên đi những gì trước đó. Sau đó, ta sẽ thêm  $i_t * \tilde{C}_t$ . Đây là giá trị ứng viên mới, co giãn (scale) số lượng giá trị mà ta muốn cập nhật cho mỗi state.



Hình 3.14: LSTM focus c.

Cuối cùng, cần quyết định xem thông tin output là gì. Output này cần dựa trên cell state, nhưng sẽ được lọc bớt thông tin. Đầu tiên, áp dụng single sigmoid layer để quyết định xem phần nào của cell state chúng ta dự định sẽ output. Sau đó, ta sẽ đẩy cell state qua tanh (đẩy giá trị vào khoảng -1 và 1) và nhân với một output sigmoid gate, để giữ lại những phần ta muốn output ra ngoài.



Hình 3.15: LSTM focus o.

Mô hình LSTM là một bước đột phá đạt được từ mô hình RNN. Nó giải quyết triệt để vấn đề không xử lý được câu hỏi dài mà những mô hình như chatbot Skype đang gặp phải.

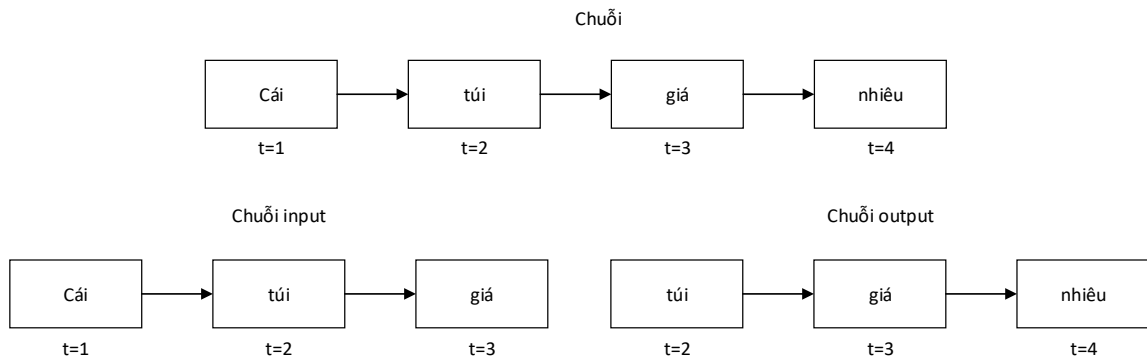
### 3.4 Phương pháp học chuỗi Seq2Seq

Ứng dụng của mô hình ngôn ngữ, cho phép đón nhận chuỗi các phân tử đầu vào, nhìn vào từng phân tử của chuỗi và cố gắng để dự đoán các phân tử tiếp theo của chuỗi văn bản. Có thể mô tả quá trình này bằng phương trình hàm số sau đây:

$$Y_t = f(Y_{t-1})$$

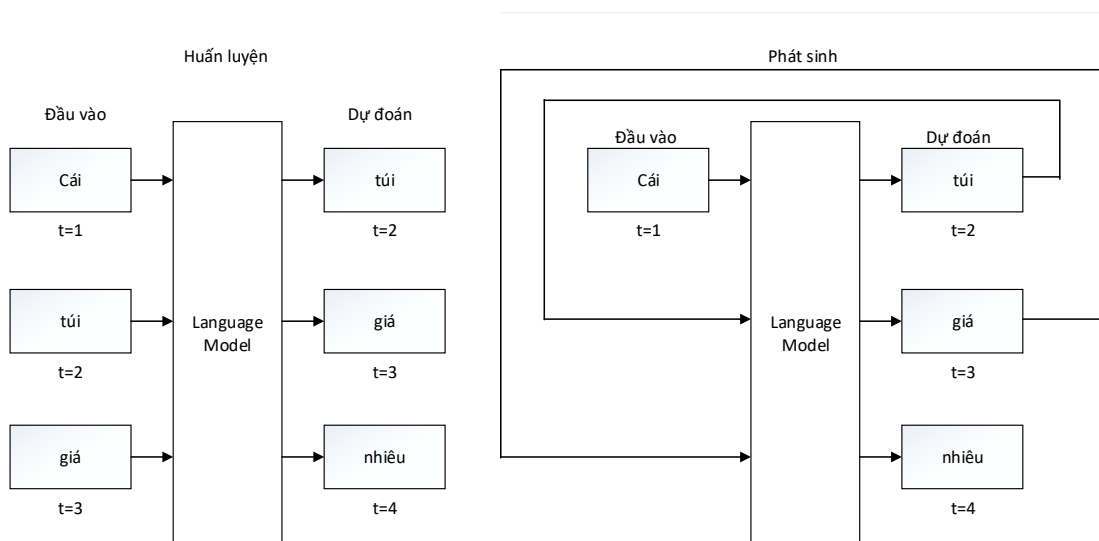
Trong đó,  $Y_t$  là phần tử chuỗi ở thời điểm  $t$ ,  $Y_{t-1}$  là phần tử chuỗi ở thời điểm trước đó và  $f$  là hàm ánh xạ các phần tử trước đó của chuỗi sang phần tử tiếp theo của chuỗi. Trong các ứng dụng hiện đại,  $f$  đại diện cho mạng nơ-ron mà có thể dự đoán được phần tử tiếp theo của một chuỗi, được cho trước bởi một phần tử hiện tại trong chuỗi đó. Mô hình ngôn ngữ có thể sinh sản, khi được huấn luyện thì chúng có thể được sử dụng để sinh ra các chuỗi thông tin bằng cách cho kết quả đầu ra ở bước trước trở lại làm đầu vào của mô hình.

Cho một chuỗi câu “Cái túi giá nhiều”. Một input là một lát cắt của chuỗi từ phần tử đầu tiên cho đến phần tử gần cuối. Chuỗi output là một lát cắt của chuỗi từ phần tử thứ 2 cho đến phần tử cuối cùng.



Hình 3.16: Mô hình phát sinh văn bản

Hình vẽ dưới đây là sơ đồ cho thấy việc huấn luyện và quá trình sinh sản của một mô hình ngôn ngữ

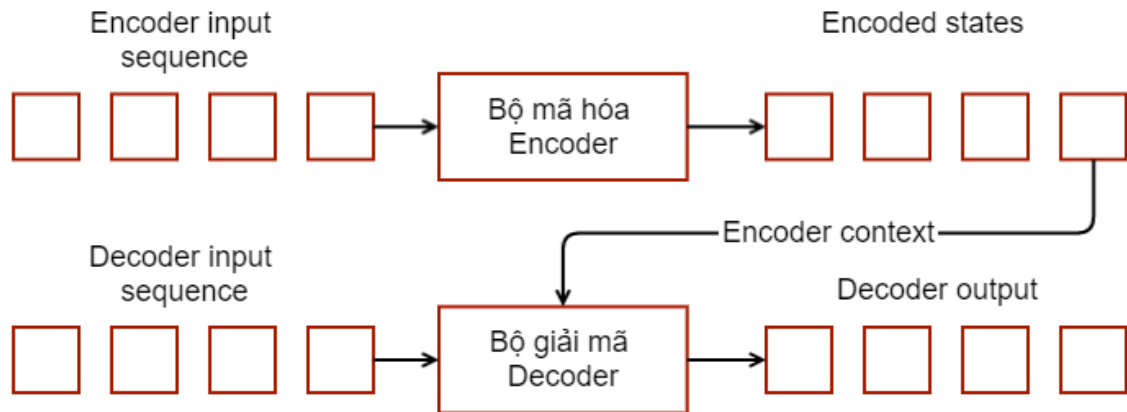


Hình 3.17: Quá trình huấn luyện và phát sinh văn bản

Trong quá trình huấn luyện, mô hình cố gắng dự đoán phần tử tiếp theo của chuỗi output được cho bởi phần tử hiện tại của chuỗi target. Trong quá trình phát sinh, mô hình sẽ lấy kết quả đã được sinh ra ở bước trước làm đầu vào cho lần dự báo tiếp theo. Không giống với các mô hình ngôn ngữ đơn giản là chỉ dự đoán xác suất cho từ tiếp theo khi được cho bởi từ hiện tại, LSTM được ứng dụng sẽ chụp lại toàn bộ bối cảnh của chuỗi đầu vào và dự đoán xác suất tạo ra các từ tiếp theo dựa trên các từ hiện tại, cũng như tất cả các từ trước. LSTM có thể được sử dụng như là mô hình ngôn ngữ cho việc sinh các phần tử của một chuỗi sau khi được huấn luyện bởi các chuỗi tương tự.

Quay trở lại bài toán của luận văn, rõ ràng kết quả nêu trên vẫn còn thiếu các thành phần cần thiết cho việc xây dựng các mô hình đề xuất, bởi vì kết quả chỉ có thể thao tác trên một chuỗi đơn, trong khi quá trình tư vấn bán hàng, luôn có câu hỏi và câu trả lời. Phương pháp học chuỗi liên tiếp seq2seq được ứng dụng trong bài toán, cho phép thao tác trên cả hai câu. Thao tác này nhằm mục đích cho mạng LSTM thông minh hơn trong việc “sinh” từ vì có thể hiểu được ngữ cảnh qua chuỗi đầu vào. Giải pháp để thực hiện là thêm vào bộ mã hóa *Encoder* và bộ giải mã *Decoder*. Trong bước mã hóa encode, một mô hình chuyển đổi một chuỗi đầu vào thành một đại diện cố định. Trong bước giải mã decode, một mô hình ngôn ngữ được huấn luyện trên cả hai chuỗi output được sinh ra và chuỗi đại diện cố định (từ bộ mã hóa encoder). Khi bộ mã hóa nhìn thấy cả hai thông tin chuỗi đầu và đã được mã hóa và chuỗi được dịch ra, nó có thể dự đoán thông minh hơn về các từ tương lai dựa và các từ hiện tại. Ví dụ, trong một mô hình ngôn ngữ chuẩn, chúng ta có thể thấy một từ “**đi**” trong tiếng Việt và không chắc từ tiếp theo là về sự dịch chuyển bằng hai chi dưới (ví dụ: *tôi đi rất nhanh nhưng vẫn không đuổi kịp anh ấy*) hay chỉ một người nào đó đã *chết* (ví dụ: *Anh ấy ra đi mà không kịp nói lời trăn trối*). Tuy nhiên, nếu chúng ta đã đi qua một bối cảnh của bộ mã hóa, thì bộ giải mã nhận ra rằng các chuỗi đầu vào đang đề cập đến việc di chuyển của con người, chứ không phải sự việc một người đã chết. Với bối cảnh đó, bộ giải mã có thể lựa chọn từ kế tiếp thích hợp và cung cấp một bản dịch chính xác. Như vậy, ta đã có thể hiểu được cơ bản về mô hình chuỗi sang chuỗi, chúng ta có thể xem xét làm thế nào để xây dựng được một mô hình dịch sử dụng mạng nơ-ron: Với bộ mã hóa, sẽ sử dụng một mạng RNN. Mạng này sẽ xử lý chuỗi đầu vào, sau đó chuyển nó thành chuỗi đầu ra của nó vào bộ giải mã decoder như một biến ngữ cảnh. Với bộ giải mã, cũng sử dụng một mạng RNN. Nhiệm vụ của nó là xem kết quả được dịch, và sau đó cố gắng dự đoán từ tiếp theo trong chuỗi được

giải mã khi đã biết được từ hiện tại trong chuỗi đã được dịch. Sau khi huấn luyện, mô hình có thể dịch bằng việc mã hóa câu mà chúng ta muốn dịch và sau đó chạy mạng ở chế độ sinh văn bản. Mô hình chuỗi sang chuỗi được mô phỏng như hình dưới đây:

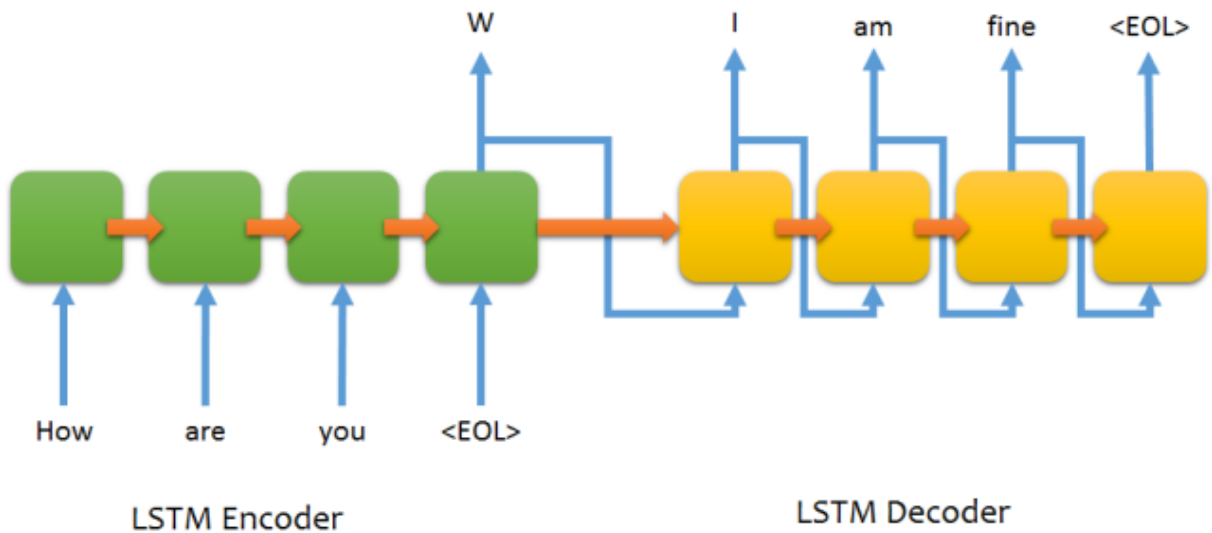


Hình 3.18: Mô hình chuỗi liên tiếp Seq2Seq

Mô hình chuỗi sang chuỗi, bộ mã hóa sinh ra một chuỗi các trạng thái. Bộ giải mã là một mô hình ngôn ngữ với một tham số bổ sung cho các trạng thái cuối cùng của bộ mã hóa. Như vậy, chúng ta đã thấy được một mô hình ngôn ngữ đơn giản cho phép chúng ta mô hình hóa các chuỗi đơn giản bằng việc dự đoán tiếp theo trong một chuỗi khi cho một từ trước đó trong chuỗi. Thêm nữa là chúng ta đã thấy quá trình xây dựng một mô hình phức tạp có phân tách các bước như mã hóa một chuỗi đầu vào thành một bối cảnh, và sinh một chuỗi đầu ra bằng việc sử dụng một mạng nơ-ron tách biệt.

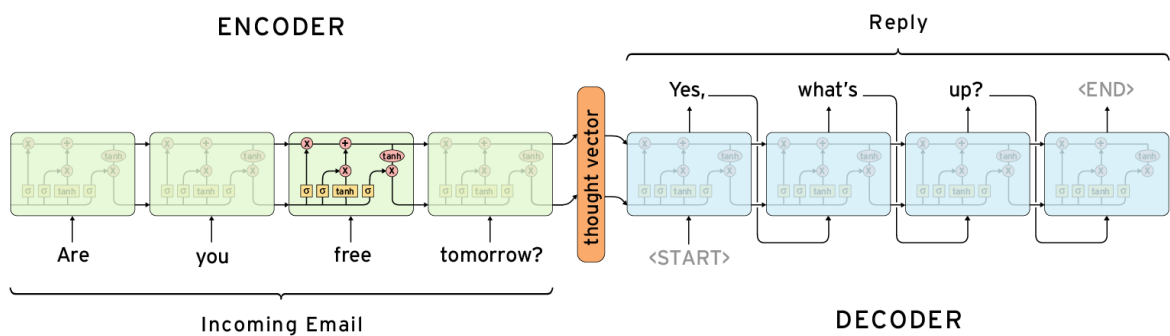
Bản thân mô hình seq2seq [6] bao gồm hai mạng RNN: Một cho bộ mã hóa, và một cho bộ giải mã. Bộ mã hóa nhận một chuỗi (câu) đầu vào và xử lý một phần tử (từ trong câu) tại mỗi bước. Mục tiêu của nó là chuyển đổi một chuỗi các phần tử vào một vector đặc trưng có kích thước cố định mà nó chỉ mã hóa thông tin quan trọng trong chuỗi và bỏ qua các thông tin không cần thiết. Có thể hình dung luồng dữ liệu trong bộ mã hóa dọc theo trục thời gian, giống như dòng chảy thông tin cục bộ từ một phần tử kết thúc của chuỗi sang chuỗi khác.





Hình 3.19: Mô hình đối thoại seq2seq.

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



Hình 3.20: Bộ mã hóa và giải mã seq2seq.

Có một vài thách thức trong việc sử dụng mô hình này. Một trong những vấn đề đáng ngại nhất là các mô hình không thể xử lý được các chuỗi dài. Bởi vì hầu như tất cả các ứng dụng chuỗi sang chuỗi, bao gồm cả độ dài các chuỗi. Vấn đề tiếp theo là kích thước từ vựng. Bộ giải mã phải chạy hàm softmax hơn trên một tập rất lớn các từ vựng (khoảng 20,000 từ) cho mỗi một từ xuất ra. Điều này

sẽ làm chậm quá trình huấn luyện, cho dù phần cứng có thể đáp ứng được khả năng xử lý. Đại diện của một từ là rất quan trọng. Làm thế nào để có thể biểu diễn được các từ trong chuỗi? Sử dụng one-hot vector có nghĩa là chúng ta phải đối mặt với các vector thừa thớt lớn, do kích thước vốn từ vựng lớn mà không có ý nghĩa về mặt ngữ nghĩa của từ được mã hóa bên trong các vector one-hot. Sau đây là cách giải quyết một số vấn đề mà chúng ta sẽ gặp phải.

### **PADDING – Tạo độ dài cố định**

Trước khi huấn luyện, chúng ta cần chuyển đổi độ dài của các phần tử trong chuỗi thành các chuỗi có độ dài cố định, bằng việc thêm vào các phần tử đệm PADDING. Các phần tử đệm đặc biệt mà chúng ta sẽ sử dụng:

- **EOS**: Kết thúc câu (End of sentence)
- **PAD**: Phần đệm bù (Filler)
- **GO**: Bắt đầu giải mã (Start decoding)
- **UNK**: Unknown; từ không biết, không có trong từ điển từ vựng

Xem xét một cặp ví dụ HỎI – ĐÁP sau đây:

*Q: How are you?*

*A: I am fine.*

Giả sử chúng ta muốn xử lý các đoạn hội thoại có độ dài 10, kết quả cặp Q/A trên sẽ được chuyển đổi thành như sau:

*Q: [PAD, PAD, PAD, PAD, PAD, PAD, “?”, “you”, “are”, “How”]*

*A: [GO, “I”, “am”, “fine”, “.”, EOS, PAD, PAD, PAD, PAD]*

### **BUCKETING – Tránh lu mờ thông tin**

Bộ đệm đã giải quyết được vấn đề độ dài của các chuỗi, nhưng hãy xem xét một trường hợp các câu có độ dài lớn. Nếu câu dài nhất trong tập dữ liệu có độ dài là 100, chúng ta cần mã hóa tất cả các chuỗi còn lại bằng độ dài 100, để không mất thông tin của bất kỳ từ nào. Như vậy, chuyện gì xảy ra với chuỗi từ “How are you?”. Sẽ có 97 phần tử đệm PAD được sử dụng khi mã hóa một chuỗi câu. Điều này sẽ làm lu mờ thông tin thực tế trong câu.

Bucketing giải quyết vấn đề này bằng việc đặt các câu vào các xô buckets có kích thước khác nhau. Ví ta có một danh sách các xô buckets: [(5, 10), (10, 15), (20, 25), (40, 50)]. Nếu độ dài của mẫu hỏi là 4 như ví dụ trên sẽ được đặt vào xô (5, 10). Mẫu hỏi sẽ được đếm với độ dài 5 và đáp án được đếm với độ dài 10. Trong lúc chạy mô hình (huấn luyện hoặc dự đoán), chúng ta sẽ sử dụng một mô hình khác cho mỗi bucket, tương ứng với các độ dài của mẫu hỏi và câu trả lời. Tất cả những mô hình này chia sẻ các tham số giống nhau và do đó hoạt động chính xác theo cùng một cách.

Nếu chúng ta sử dụng xô (5, 10), thì các câu sẽ được mã hóa thành:

*Q: [PAD, “?”, “you”, “are”, “How”]*

*A: [GO, “I”, “am”, “fine”, “.”, EOS, PAD, PAD, PAD, PAD]*

### **Word Embedding – Mật độ dày đặc**

Word Embedding là một kỹ thuật cho việc học mật độ dày đặc thông tin đại diện của từ trong một không gian vector có số chiều nhỏ hơn. Mỗi một từ có thể được xem như là một điểm trong không gian này, được đại diện bởi một vector có độ dài cố định. Word Embedding thường được thực hiện trong lớp đầu tiên của mạng: Trong đó lớp embedding sẽ ánh xạ một từ (chỉ số index của từ trong từ điển từ vựng) từ từ điển sang một vector dày đặc với kích thước đã cho. Trong mô hình seq2seq, trọng số của lớp embedding được huấn luyện giống nhau với các tham số khác của mô hình.

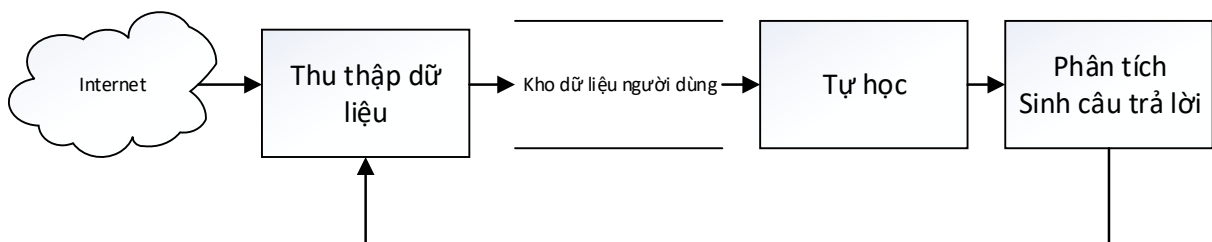
Qua chương 3, ta đã có cái nhìn tổng quan các khái niệm nền tảng, các công cụ cần thiết phục vụ cho xây dựng mô hình bán hàng tự động. Dựa trên những kiến thức trên, chương 4 sẽ đưa ra giải pháp và phương án thực hiện.

## CHƯƠNG 4: GIẢI PHÁP VÀ XÂY DỰNG MÔ HÌNH ĐỀ XUẤT ISALES

Ở chương này, luận văn đi vào phân tích các nghiệp vụ cần thiết cần có của mô hình bán hàng tự động trên Internet, kết hợp với các kiến thức được đề cập trong chương 3 để đưa ra mô hình đề xuất iSales. Phần đầu tiên, xin được trình bày giải pháp tổng thể. Phần xây dựng mô hình sẽ được đề cập ở phần thứ sau đó.

### 4.1 Giải pháp đề xuất

Mục 3.1, chương 3 cho ta biết được cần xây dựng phương án mới cho bài toán trả lời bán hàng tự động. Từ đó, có thể phác thảo về các pha chính cho một mô hình bán hàng tự động, bao gồm 3 pha. Pha đầu tiên là thu thập dữ liệu – thu thập dữ liệu từ Internet hoặc tái sử dụng kết quả tư vấn sau pha phân tích và trả lời. Mục tiêu của pha là lấy được càng nhiều dữ liệu càng tốt, dữ liệu cần phải được chọn lọc và có nội dung trong miền thương mại. Pha thứ 2 là tự học - cho phép tự học thông qua kho dữ liệu người dùng (kết quả của pha thu thập dữ liệu) hoặc từ dữ liệu thu thập trên mạng. Mục tiêu của pha là xây dựng mô hình ngôn ngữ có khả năng sinh văn bản dựa trên các “tri thức” học được từ pha thu thập dữ liệu. Cuối cùng là pha phân tích/sinh câu trả lời - mô hình được sinh ra sau quá trình tự học sẽ được áp dụng để hiểu ngữ cảnh câu hỏi đầu vào, từ đó sinh văn bản để tạo câu trả lời hợp lý. Ngoài ra, những kết quả hỏi – trả lời sinh ra cần được thu thập lại và lưu trữ vào kho dữ liệu huấn luyện để làm đầu vào bổ sung cho pha tự học, điều này sẽ xử lý được vấn đề tái sử dụng dữ liệu tư vấn.



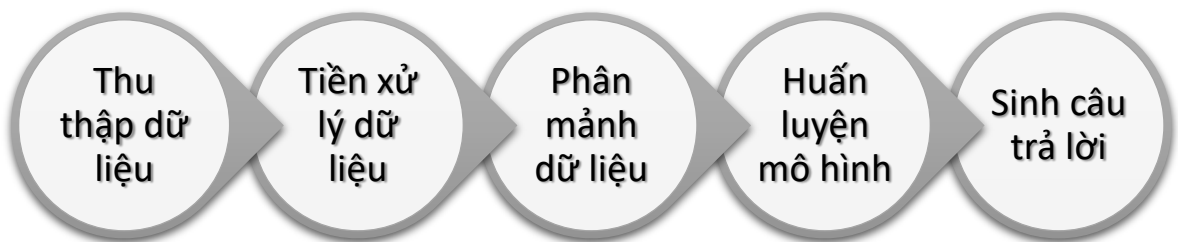
*Hình 4.1: Các pha trong mô hình bán hàng tự động*

Với mô hình này có thể giải quyết được 1 số vấn đề đã nêu trong mục 2.2, chương 2. Đó là vấn đề phát sinh những câu hỏi nằm ngoài “tập dữ liệu mẫu”, vấn đề câu hỏi dài và phức tạp, vấn đề tái sử dụng dữ liệu tư vấn và vấn đề “tự động”. Tuy nhiên, để giải quyết vấn đề hiểu được với ngôn ngữ tiếng Việt mô hình cần bổ sung bước tiền xử lý ngôn ngữ. Dữ liệu thu thập được cần xử lý để loại bỏ ký

tự không có ý nghĩa, câu trùng lặp, từ ngữ bất thường trong tiếng Việt. Thử nghiệm mô hình bán hàng tự động không sử dụng pha tiền xử lý và phân mảnh dữ liệu, kết quả của quá trình huấn luyện dữ liệu tạo ra các câu không có ý nghĩa trong tiếng Việt. Nguyên nhân do âm tiết của tiếng Việt có thể không có nghĩa, và khi ghép với các âm tiết khác sẽ gây ra hiểu lầm nghiêm trọng.

Ví dụ, trong câu hỏi “*Thiết bị truyền dẫn giá bao nhiêu tiền?*”. Với bài thử nghiệm khi không sử dụng pha tiền xử lý và phân mảnh: câu hỏi trên được tách 8 từ đơn: “thiết”, “bị”, “truyền”, “dẫn”, “giá”, “bao”, “nhiều”, “tiền”. Trong mô hình được huấn luyện với dữ liệu không phân mảnh, datasets sẽ hiểu đó là 8 word riêng biệt. Kết quả khi ứng dụng sẽ ra câu trả lời không rõ nghĩa như: “*Truyền bị dẫn giá thiết 120,000*” hoặc “*Truyền dẫn bị giá thiết 120,000*”....Cùng phương pháp huấn luyện, nhưng với nguồn dữ liệu được phân mảnh, datasets hiểu câu hỏi trên chỉ với 5 từ: “Thiết\_bị”, “truyền\_dẫn”, “giá”, “bao\_nhiều”, “tiền”. Kết quả sau khi ứng dụng có thể ra các câu trả lời như sau: “*Truyền\_dẫn thiết\_bị giá 120,000.*”, “*Thiết\_bị truyền\_dẫn giá 120,000.*”...Rõ ràng, số lượng câu trả lời có thể sinh ra bởi cùng 1 câu hỏi cho 2 bài thử nghiệm nêu trên là khác nhau. Ở bài thử nghiệm không sử dụng 2 pha cho ra nhiều kết quả tối nghĩa hơn. Bài có sử dụng sẽ thu hẹp kết quả với những kết quả ý nghĩa hơn.

Để giải quyết vấn đề còn tồn đọng ở trên, giải pháp iSales được đề xuất gồm 5 pha:



Hình 4.2: Sơ đồ quy trình của mô hình đề xuất

- Pha thu thập dữ liệu: sử dụng graphAPI để crawl dữ liệu từ page trên Facebook, trích xuất dữ liệu, chỉ lấy những comment trong các post nhằm có được dữ liệu thô cho quá trình huấn luyện. Kết quả pha được lưu trữ lại trong file text.

- Pha tiền xử lý dữ liệu: pha này cần đọc dữ liệu từ file kết quả của pha thu thập, tiến hành xóa bỏ những cặp câu bất thường/vô nghĩa, xóa bỏ những cặp câu bị trùng/có khoảng trắng và ghi kết quả vào file.
- Pha tiền phân mảnh dữ liệu: nhiệm vụ pha là đọc dữ liệu từ file kết quả của pha xử lý dữ liệu, tách từ theo nghĩa tiếng Việt. Lưu kết quả cho quá trình huấn luyện.
- Pha huấn luyện dữ liệu: cho phép đọc dữ liệu từ file kết quả của pha phân mảnh dữ liệu, bóc tách Word2Id và tạo ra từng cặp hội thoại, đưa kết quả vào mạng LSTM để huấn luyện.
- Pha sinh câu trả lời: nhận dữ liệu input là câu hỏi đầu vào, tiến hành phân tích câu hỏi dựa trên các ngữ cảnh học được. Sử dụng mô hình sinh ra câu trả lời và hiển thị kết quả.

Dựa trên nghiệp vụ của từng pha, tôi đề xuất sử dụng những công cụ sau để thực hiện xây dựng mô hình đề xuất:

*Bảng 4.1 Các công cụ xây dựng mô hình iSales*

| Tên công cụ | Chú giải  | Sử dụng ở pha nào                                      |
|-------------|---|--|
| GraphAPI    | Công cụ thu thập dữ liệu facebook   | Pha thu thập dữ liệu                                   |
| NLTK        | Công cụ xử lý ngôn ngữ tự nhiên mã nguồn mở   | Pha huấn luyện và sinh câu trả lời                     |
| VNTK        | Vietnamese language toolkit, tự phát triển để xử lý các vấn đề cơ bản của tiếng Việt. | Pha tiền xử lý   |
| Dongdu      | Thư viện hỗ trợ tách từ tiếng Việt [11], của tác giả Lưu Tuấn Anh                     | Pha phân mảnh dữ liệu                                  |
| Tensorflow  | Framework machine learning  | Pha huấn luyện và sinh câu trả lời                     |
| Python      | Ngôn ngữ lập trình  | Pha huấn luyện và sinh câu trả lời                     |
| NodeJS      | Ngôn ngữ lập trình  | Pha thu thập dữ liệu, tiền xử lý và phân mảnh dữ liệu. |

## 4.2 Nguồn dữ liệu huấn luyện

Ở các phần trên đã nêu ra được các kiến thức cần để xây dựng mô hình bán hàng tự động sử dụng mạng nơ-ron và phương pháp học chuỗi seq2seq. Tuy nhiên,

để có thể hiểu ngữ cảnh, phương pháp học chuỗi seq2seq yêu cầu có lượng dữ liệu lớn cho nghiệp vụ huấn luyện. Tôi đã tìm hiểu, lựa chọn facebook và sử dụng công graphAPI trong quá trình nghiên cứu đề tài.

#### 4.2.1 Facebook

Facebook, một website truy cập miễn phí do công ty Facebook điều hành. Người dùng có thể tham gia các mạng lưới được tổ chức theo thành phố, nơi làm việc, trường học và khu vực để liên kết, giao tiếp với người khác. Mọi người cũng có thể kết bạn và gửi tin nhắn cho họ, và cập nhật trang hồ sơ cá nhân của mình để thông báo cho bạn bè biết về chúng. Page trên Facebook được hiểu là một hồ sơ thông tin của một doanh nghiệp hoặc một tổ chức bất kì chứ không mang tư cách là cá nhân cụ thể nào, mọi người có thể đầu tư và phát triển thương hiệu cũng như sản phẩm đến người dùng một cách nhanh nhất, hiệu quả nhất hiện nay. Ưu điểm dễ quản lý, dễ sử dụng, dễ trao đổi, và hoàn toàn miễn phí, page đã vượt qua các trang web thương mại đơn giản để trở thành kênh bán hàng trên internet phổ biến nhất hiện nay. Post trên page là khái niệm chỉ một bài viết của người bán hàng. Người bán hàng khi muốn bán một đơn hàng nào đó, sẽ cần tạo ra một post. Các thông tin cần có: giới thiệu về đơn hàng, hình ảnh minh họa, giá cả nếu cần thiết..., thông tin được cung cấp càng nhiều, mọi người sẽ càng hiểu về đơn hàng. Comment, là những dòng câu hỏi, trả lời trên mỗi post. Comment có thể do người mua hàng hỏi về thông tin sản phẩm, về thông tin cần tư vấn, cũng như quyết định mua hay không. Comment cũng có thể từ người bán hàng tư vấn, trả lời. Nói cách khác, comment là những dòng chat trên một post.

Quay lại với mô hình bán hàng tự động, để cho hệ thống có thể tự học, thì cần một lượng dữ liệu đầu vào khổng lồ để máy tính có thể học hỏi. Với tần suất sử dụng liên tục, người dùng đã giúp Facebook có lượng dữ liệu khổng lồ. Theo thống kê của Jay Parikh - phó Chủ tịch phụ trách cơ sở hạ tầng kỹ thuật, thì trong một ngày, Facebook có 2,7 tỉ lượt “like” hàng ngày, 300 triệu bức ảnh được tải lên Facebook, 700.000 truy vấn được thực hiện bởi người sử dụng tới hệ thống, và 500 TB dữ liệu mới được xử lí. Không chỉ có vậy, thống kê còn chỉ ra cứ sau 18 tháng, dữ liệu điện tử toàn cầu của facebook lại tăng gấp đôi [9].

Tất nhiên, với những con số nêu trên, thì lĩnh vực bán hàng và tư vấn bán hàng trên Facebook có lượng dữ liệu không hề nhỏ. Đây là nguồn thông tin đầu vào khá phù hợp cho mô hình bán hàng tự động.

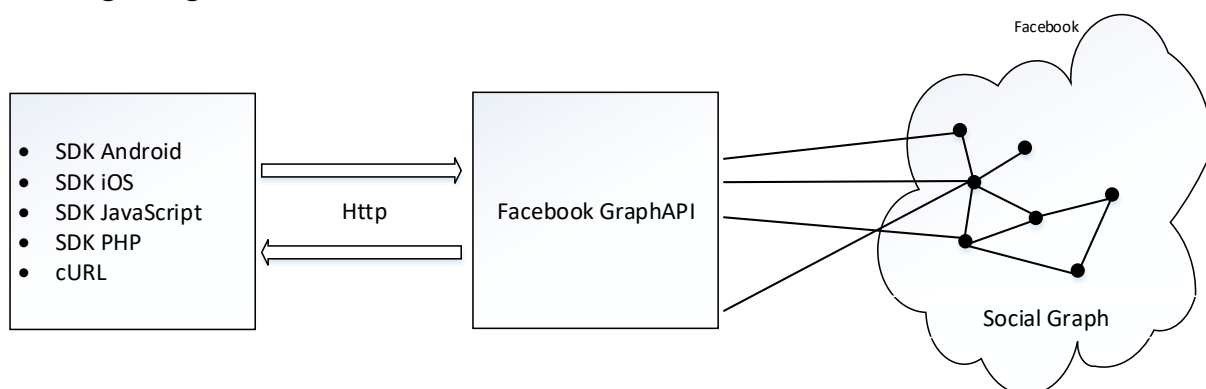
### 4.2.2 GraphAPI

GraphAPI [10], được Facebook công bố từ 21/04/2010 với phiên bản đầu tiên v1.0. Phát triển được hơn 6 năm, hiện tại đã tới phiên bản 2.8, GraphAPI cho phép kỹ sư lập trình tải dữ liệu và hoặc lấy dữ liệu ra khỏi nền tảng của Facebook.

*Bảng 4.2 Các phiên bản graphAPI Facebook*

| Phiên bản | Đường dẫn      | Ngày giới thiệu | Khả dụng đến                        |
|-----------|----------------|-----------------|-------------------------------------|
| v2.8      | /v2.8/{object} | Ngày 5/10/2016  | Tối thiểu đến tháng 10/2018         |
| v2.7      | /v2.7/{object} | Ngày 13/7/2016  | Ngày 5/10/2018                      |
| v2.6      | /v2.6/{object} | Ngày 12/4/2016  | Ngày 13/7/2018                      |
| v2.5      | /v2.5/{object} | Ngày 7/10/2015  | Ngày 12/4/2018                      |
| v2.4      | /v2.4/{object} | Ngày 8/7/2015   | Ngày 7/10/2017                      |
| v2.3      | /v2.3/{object} | Ngày 25/3/2015  | Ngày 8/7/2017                       |
| v2.2      | /v2.2/{object} | Ngày 30/10/2014 | Ngày 25/3/2017                      |
| v2.1      | /v2.1/{object} | Ngày 7/8/2014   | Ngày 30/10/2016                     |
| v2.0      | /v2.0/{object} | Ngày 30/4/2014  | Không khả dụng kể từ ngày 7/8/2016  |
| v1.0      | /v1.0/{object} | Ngày 21/4/2010  | Không khả dụng kể từ ngày 30/4/2015 |

GraphAPI là hệ thống các API dựa trên HTTP cấp thấp có thể sử dụng để truy vấn dữ liệu, đăng tin mới, quản lý quảng cáo, tải ảnh lên và nhiều tác vụ khác mà ứng dụng có thể cần thực hiện.

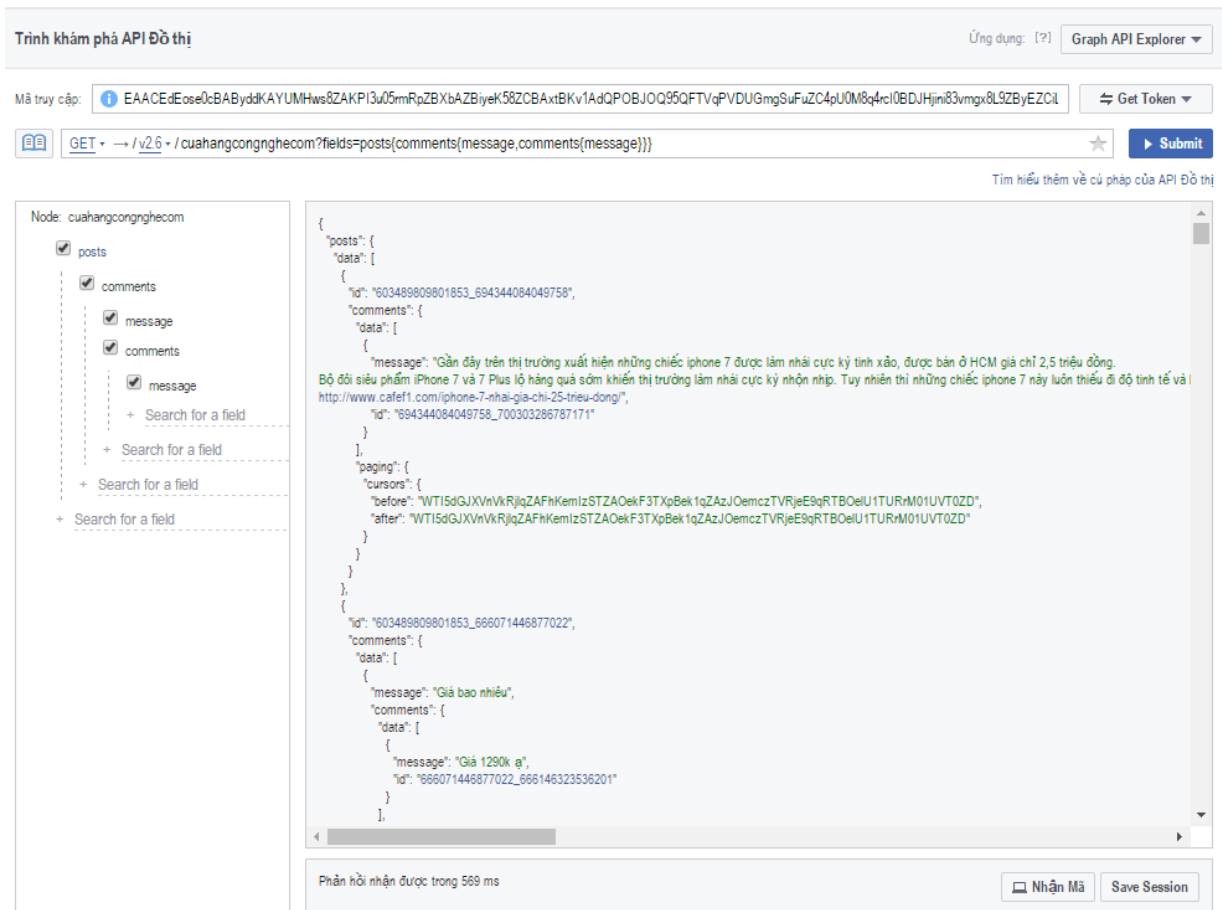


*Hình 4.3: Mô hình giao tiếp sử dụng graphAPI Facebook*

Về cơ bản, “đồ thị xã hội” GraphAPI bao gồm các nút (node) thể hiện các mục như người dùng, ảnh, trang, bình luận. Các cạnh (edge) là kết nối giữa các mục đó, như ảnh của trang hoặc bình luận của ảnh. Các thông tin về các mục đó, như ngày sinh của người dùng hoặc tên của trang gọi là trường (field). Muốn sử dụng graphAPI, yêu cầu đầu tiên cần một mã truy cập (*user access token*). Người



dùng muốn lấy được dữ liệu cần phải đăng ký trở thành nhà phát triển, mỗi khi sử dụng graphAPI, Facebook sẽ cấp một mã truy cập có thời hạn. Sau khi người dùng khai báo đúng mã truy cập trên ứng dụng, dữ liệu được thu thập được thông qua ứng dụng. GraphAPI cho phép bóc tách dữ liệu theo kiểu Json.



Hình 4.4: Sử dụng graphAPI v2.6 để thu thập dữ liệu page

Ngoài ra, graphAPI còn hỗ trợ mã tích hợp cho các loại ngôn ngữ, hoặc là trên những nền tảng khác nhau như Android, iOS, javaScript, PHP, cURL... Với những hỗ trợ nêu trên, việc lấy dữ liệu từ vấn bán hàng của một hay nhiều page bán hàng trở nên dễ dàng với GraphAPI.

### 4.3 Xây dựng mô hình iSales

Phần này trình bày phương pháp xây dựng từng pha trong mô hình đề xuất iSales. Ở mỗi pha, tôi sẽ đưa ra mục tiêu, vấn đề pha xử lý, nghiệp vụ chi tiết, các công cụ hỗ trợ xây dựng và kết quả đạt được.

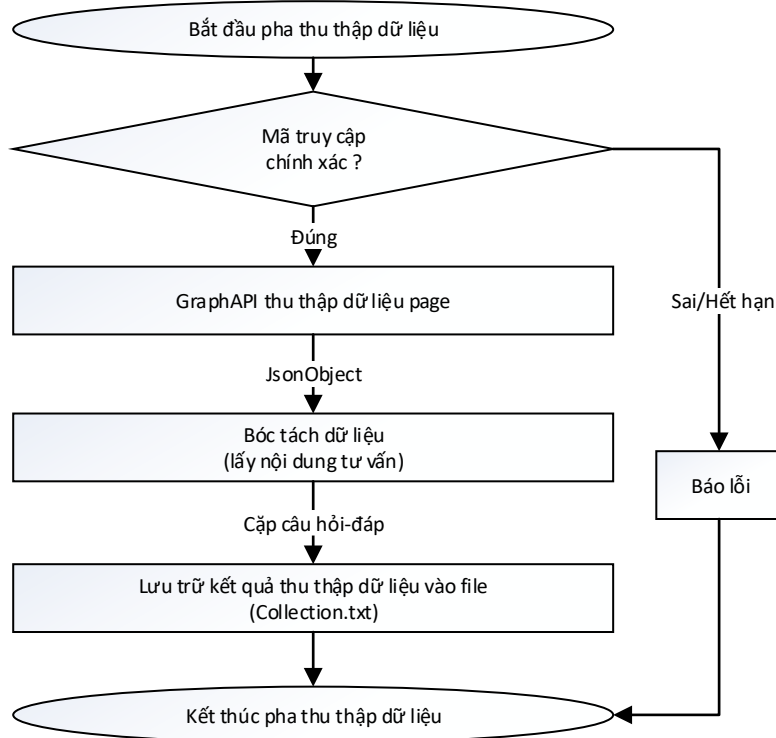
#### 4.3.1 Pha thu thập dữ liệu

Mục tiêu của pha là thu thập dữ liệu đầu vào cho mô hình đề xuất. Dữ liệu thu thập này sẽ quyết định khả năng học hỏi, xử lý. Giống như con người, mô hình càng được học nhiều, thì tính chính xác của mô hình càng cao. Các điều kiện bắt buộc cho nguồn dữ liệu đầu vào cần thỏa mãn gồm dữ liệu đầu vào phù hợp với nghiệp vụ thương mại điện tử với khối lượng dữ liệu thu thập lớn, đủ cho mô hình học. Ngoài ra, vì mô hình đề xuất phục vụ cho thương mại điện tử tại Việt Nam nên ngôn ngữ dữ liệu phải là Tiếng Việt và cần phải là những cuộc đối thoại

Như đã đề cập ở chương 3, dữ liệu Facebook là một trong những lời giải tốt cho pha này. Dữ liệu trao đổi, tư vấn giữa người mua và người bán trên những page nói chung và của những trang có ngôn ngữ Tiếng Việt nói riêng đáp ứng được các tiêu chí về dữ liệu lớn: tùy thuộc vào mỗi bài viết (post) trên page và tần xuất theo dõi của người mua cũng như sự nhiệt tình trả lời của người bán mà số lượng comment trên một page là khác nhau. Có những post có số lượng comment, tư vấn lên đến hàng nghìn, có những post chỉ dừng lại ở mức 20-30 lượt comment. Nhưng với tần xuất sử dụng của người dùng Facebook hiện nay, khối lượng dữ liệu sau khi trích xuất, thu thập có thể đáp ứng được tiêu chí này. Tiêu chí ngôn ngữ cũng đáp ứng tốt, với mỗi page, người bán hàng có thể đặt và sử dụng ngôn ngữ theo ý mình. Từ đó, người mua hàng cũng sẽ sử dụng ngôn ngữ đó để giao tiếp. Một kịch bản đơn giản, với page buôn bán đồ mỹ phẩm của người Việt, thì việc trao đổi thông tin, tư vấn mua hàng bằng Tiếng Việt là điều tất nhiên. Tiêu chí khả dụng ở đây có thể phân ra thành hai tiêu chí nhỏ là tiêu chí khả dụng về ngôn ngữ và tiêu chí khả thi trong việc thu thập. Tiêu chí khả dụng về ngôn ngữ, cụ thể hơn là dữ liệu trao đổi cần phải có cặp, từ là với một cặp 2 comment, có thể trở thành một câu đối thoại hỏi-trả lời. Còn tiêu chí khả thi trong việc thu thập, graphAPI cho phép người dùng có thể thu thập dữ liệu từ một page bất kỳ và trích xuất lấy dữ liệu cần thiết.

Phiên bản graphAPI version 2.6 được sử dụng trong quá trình nghiên cứu đề tài hỗ trợ người dùng có thể thu thập dữ liệu trên một page cụ thể, và trích xuất lấy thông tin đầy đủ của 25 posts gần nhất. Tùy theo nhu cầu của người sử dụng,

việc trích xuất thông tin để lấy comment trong từng posts hoàn toàn có thể thực hiện được. Đầu tiên, người dùng cần đăng nhập Facebook Developer để nhận mã xác thực. Sau đó, sử dụng mã xác thực để thu thập dữ liệu thông qua graphAPI, tiến hành bóc tách dữ liệu để lấy nội dung tư vấn và lưu trữ kết quả vào file collection.txt.



Hình 4.5: Luồng nghiệp vụ pha thu thập dữ liệu

Với luồng nghiệp vụ như trên, tôi đề xuất sử dụng ngôn ngữ lập trình nodeJS, các thư viện có thể sử dụng là thư viện fb - cho phép tích hợp, sử dụng các graphApi, thư viện fs - cho phép đọc ghi file.

```

1 // Using require() in ES5
2 var FB = require('fb');
3 var fs = require('fs');
4 var accessToken = '_Mã_xác_thực_';
5 //Get data from FB
6 FB.options({ version: 'v2.6' });
7 FB.setAccessToken(accessToken);
8 FB.api(
9   _Tên_page_,
10  'GET',
11  { "fields": "posts{comments{message,comments{message}}}" },//get comments and reply
12  function (response) {
13    //Bóc tách dữ liệu
14  }
15 );
  
```

Hình 4.6: Mã nguồn sample pha thu thập dữ liệu

Output của pha thu thập dữ liệu sẽ là một tập tin chứa các cặp câu hỏi đáp theo đúng thứ tự tư vấn trên page. Ví dụ file kết quả thu thập tại trang “*cuahangcongnghecom*”:

```
Giá bao nhiêu ?
Giá 1290k ạ ☺)
Giá 1290k ạ ☺)
Giá 1290k ạ ☺)
Có bảo hành k, bao lâu ?

Bảo hành 6 tháng anh nhé !
...
```

Hình 4.7: Kết quả sample pha thu thập dữ liệu

Ngoài phương án thu thập dữ liệu comment từ các page thương mại điện tử trên Internet, chúng ta có thể tái sử dụng nguồn dữ liệu tư vấn của người mua hàng và người bán hàng trên Messenger. Việc thu thập lịch sử tư vấn tương tự, dễ dàng thực hiện được với các graphAPI hỗ trợ.

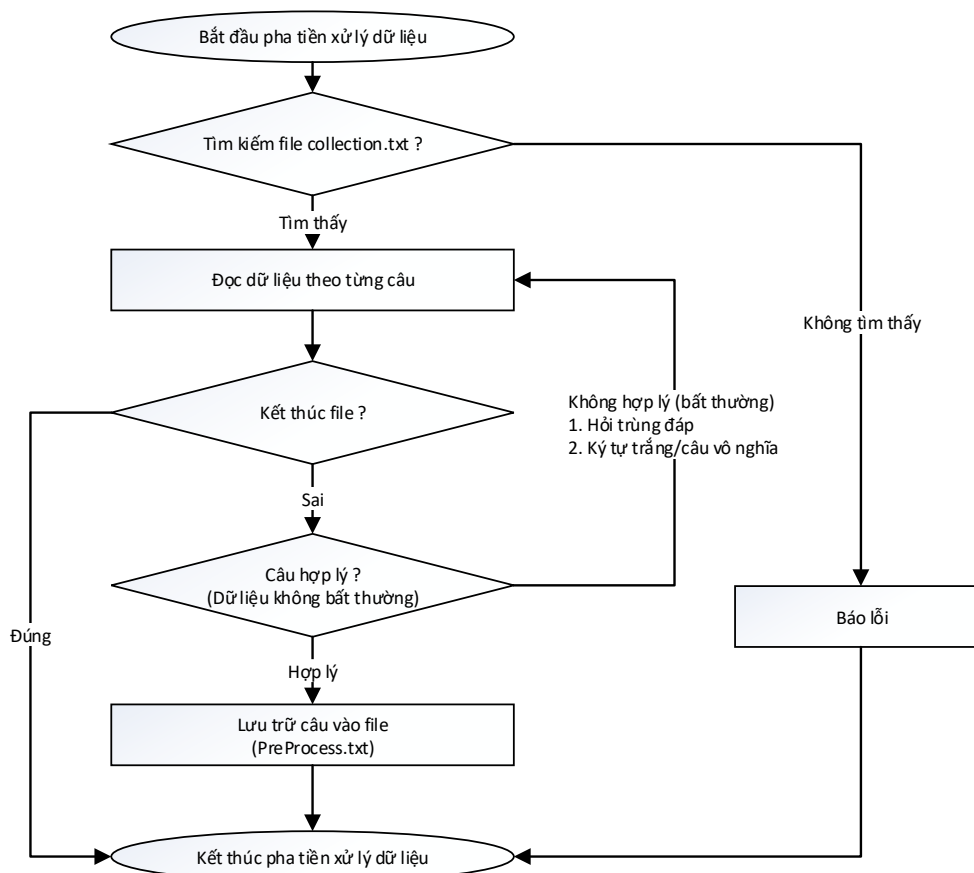
### 4.3.2 Pha tiền xử lý dữ liệu

Dữ liệu được thu thập và trích xuất từ page bán hàng có thể đáp ứng được những yêu cầu cơ bản để sử dụng huấn luyện trong mô hình bán hàng tự động. Tuy nhiên, với hiện trạng trên mạng, những comments trên các post đôi khi không có ý nghĩa, hoặc bị lặp từ như kết quả thử nghiệm tài phần 3.3.1. Việc tiền xử lý dữ liệu sẽ chọn lọc các thuộc tính phù hợp cho mô hình (*Filtering Attributes*) và làm sạch (*clean data*) dữ liệu thô được thu thập sau bước thu thập.

Bảng 4.3 Bảng nội dung làm sạch dữ liệu

| Nội dung thực hiện  | Ví dụ   |
|---|---|
| Loại bỏ các ký tự đặc biệt không phải chữ hoặc chữ số (bắt đầu, kết thúc và bên trong một câu tiếng Việt) | <i>Xin chào, các bạn!</i>                     |
| Xóa bỏ các ký tự phân tách câu không phải dấu chấm, dấu hỏi hoặc dấu chấm than                            | @#\$\$%^&*                                    |
| Xóa bỏ các bình luận, chú thích ý nghĩa các từ, thuật ngữ trong câu                                       | <i>Chatbot (chương trình tự động trả lời)</i> |
| Xóa bỏ các ký tự lặp, ký tự phân tách không có ý nghĩa  | -, , ...,                                     |
| Biến đổi bảng mã Unicode tổ hợp về dạng unicode dụng sẵn  | <i>Pha?i. La`12'...</i>                       |
| Loại bỏ các dòng trắng, vô nghĩa  |   |
| Loại bỏ các câu giống nhau  |   |

Nghệp vụ của pha khá đơn giản. Bước 1, Kiểm tra file đầu vào (collection.txt), nếu không có báo lỗi. Bước 2, đọc dữ liệu theo từng câu và làm sạch dữ liệu. Bước 3, lưu các câu đã xử lý vào file kết quả. Danh sách các thư viện cần thiết gồm có: thư viện fs - cho phép đọc ghi file, thư viện es - hay event-stream, hỗ trợ đọc ghi file, cho phép tạm dừng hoặc tiếp tục xử lý trong quá trình đọc ghi file, thư viện unicode - cho phép chuyển đổi từ mã Unicode tổ hợp về dạng unicode dựng sẵn và ngược lại, thư viện regex - cho phép biểu diễn các mẫu nhóm chứa ký tự.



Hình 4.8: Luồng nghiệp vụ pha tiền xử lý dữ liệu

Output của pha tiền xử lý dữ liệu là một file dữ liệu đã được xử lý, tất cả những câu không có ý nghĩa, những ký tự bất thường sẽ bị xóa.

Giá bao nhiêu  
 Giá 1290k ạ  
 Có bảo hành k, bao lâu  
 Bảo hành 6 tháng anh nhé

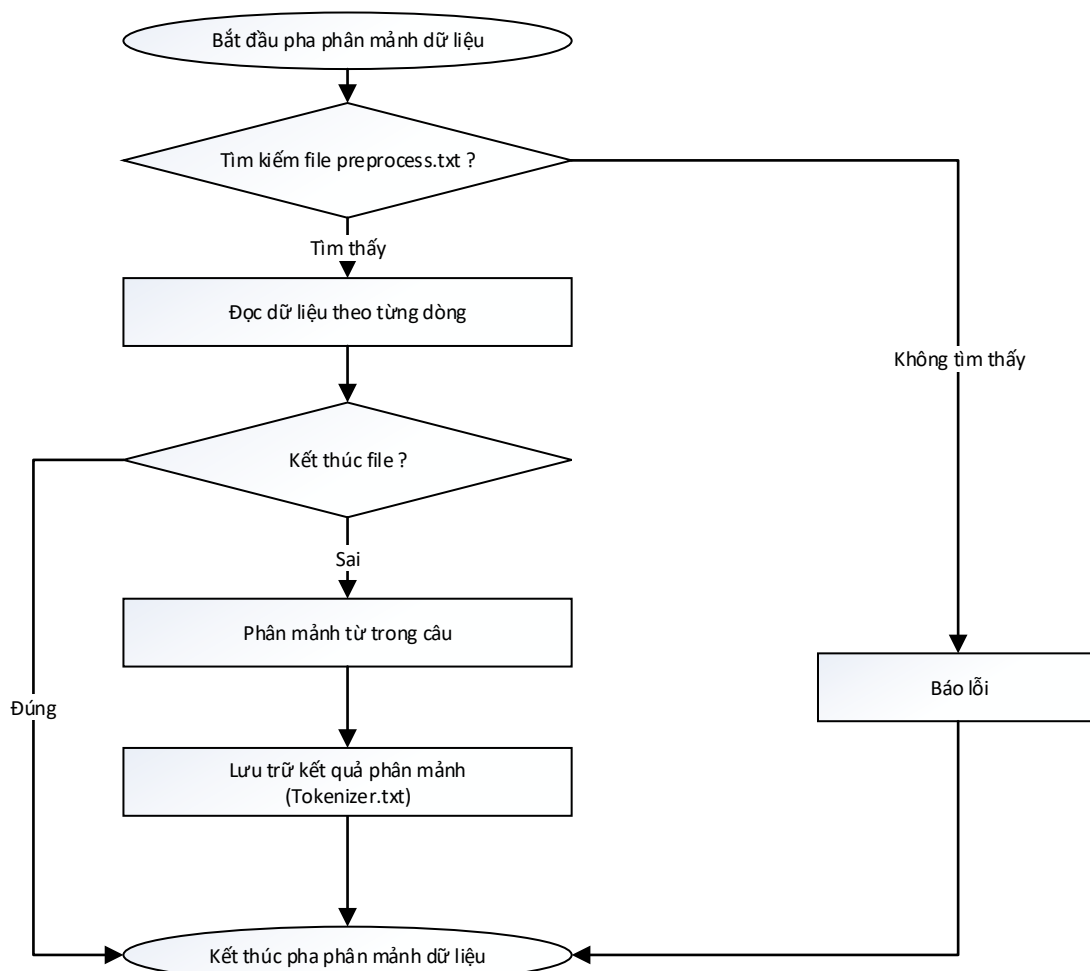
Hình 4.9: Kết quả sample pha tiền xử lý dữ liệu

### 4.3.3 Pha phân mảnh dữ liệu

Một trong những thử thách trong mô hình bán hàng tự động qua mạng tại Việt Nam là thử thách về ngôn. Tiếng Việt phức tạp hơn tiếng Anh ở nhiều khía cạnh như đã đề cập ở phần 3.1. Ở tiếng Anh, định nghĩa từ và sự kết hợp từ đơn giản, trong khi đó với Tiếng Việt lại khác. Ví dụ câu chào đơn giản: “Welcome to HaNoi” và “Chào mừng tới Hà Nội”, cùng một ý nghĩa nhưng phương pháp xử lý ở hai câu là khác nhau. Trong câu “Welcome to HaNoi”, việc tách từ rất đơn giản, thuật toán chỉ cần split các từ theo đúng kiểu “*word by word*”. Các từ được tách ra là: “Welcome”, “to”, “HaNoi”, hoàn toàn có thể đứng độc lập và vẫn mang ý nghĩa. Mặt khác, trong tiếng Việt, việc tách từ không chỉ dừng lại ở mức độ tách *word by word* – tách âm tiết, mà từ tách ra cần phải có ý nghĩa và có thể hiểu được. Theo như ví dụ trình bày ở trên, kết quả của câu “Chào mừng tới Hà Nội” là các từ “Chào\_mừng”, “tới”, “Hà\_Nội”.

Thư viện DongDu của tác giả Lưu Tuấn Anh cho phép hỗ trợ tách từ tiếng Việt với tỷ lệ chính xác tới 98.2% [11]. Sử dụng phương pháp mới được nghiên cứu gần đây là Pointwise, DongDu khắc phục được nhược điểm của những phương pháp học máy tham khảo nhãn (HMM, CRF và ME). Ưu điểm vượt trội so với các thư viện khác là chỉ cần từ điển có số lượng từ vừa phải, khá hiệu quả khi xác định những từ mới không có trong từ điển, ít bị ảnh hưởng bởi các ví dụ sai trong dữ liệu huấn luyện. Thư viện này còn có tốc độ xử lý rất nhanh (nhanh hơn vnTokenizer của Lê Hồng Phong 8 lần), có thể phân loại những bài toán có từ hàng triệu đến hàng chục triệu đặc trưng trong khi yêu cầu cấu hình máy thấp, máy tính cá nhân có thể hoạt động được.

Trở lại với yêu cầu của hệ thống, sau pha tiền xử lý dữ liệu, chúng ta đã có file kết quả chứa tập các câu tư vấn sử dụng cho hệ thống. Nghiệp vụ tiếp theo cần thực hiện là tách từ trong từng câu cho file đó.



Hình 4.10: Luồng nghiệp vụ pha phân mảnh dữ liệu

Tương tự như 2 pha trước, tôi cũng sử dụng ngôn ngữ lập trình nodeJS, các thư viện gồm có thư viện fs, cho phép đọc ghi file. Thư viện hỗ trợ đọc ghi file, cho phép tạm dừng hoặc tiếp tục xử lý trong quá trình đọc ghi file như event-stream. Một thư viện không thể thiếu là vntk - sử dụng thư viện hỗ trợ tách từ DongDu để xử lý tách từ trong của tiếng Việt.

Output nhận được là file segment chứa tập các câu đã được tách từ theo ngôn ngữ Tiếng Việt. Tuy kết quả chỉ được > 90%, nhưng rõ ràng có thể sử dụng làm input tốt cho pha huấn luyện dữ liệu.

```

Giá bao_ nhiều
Giá_1290k ạ
Có bảo_hành k, bao_ lâu
Bảo_hành 6 tháng anh nhé
  
```

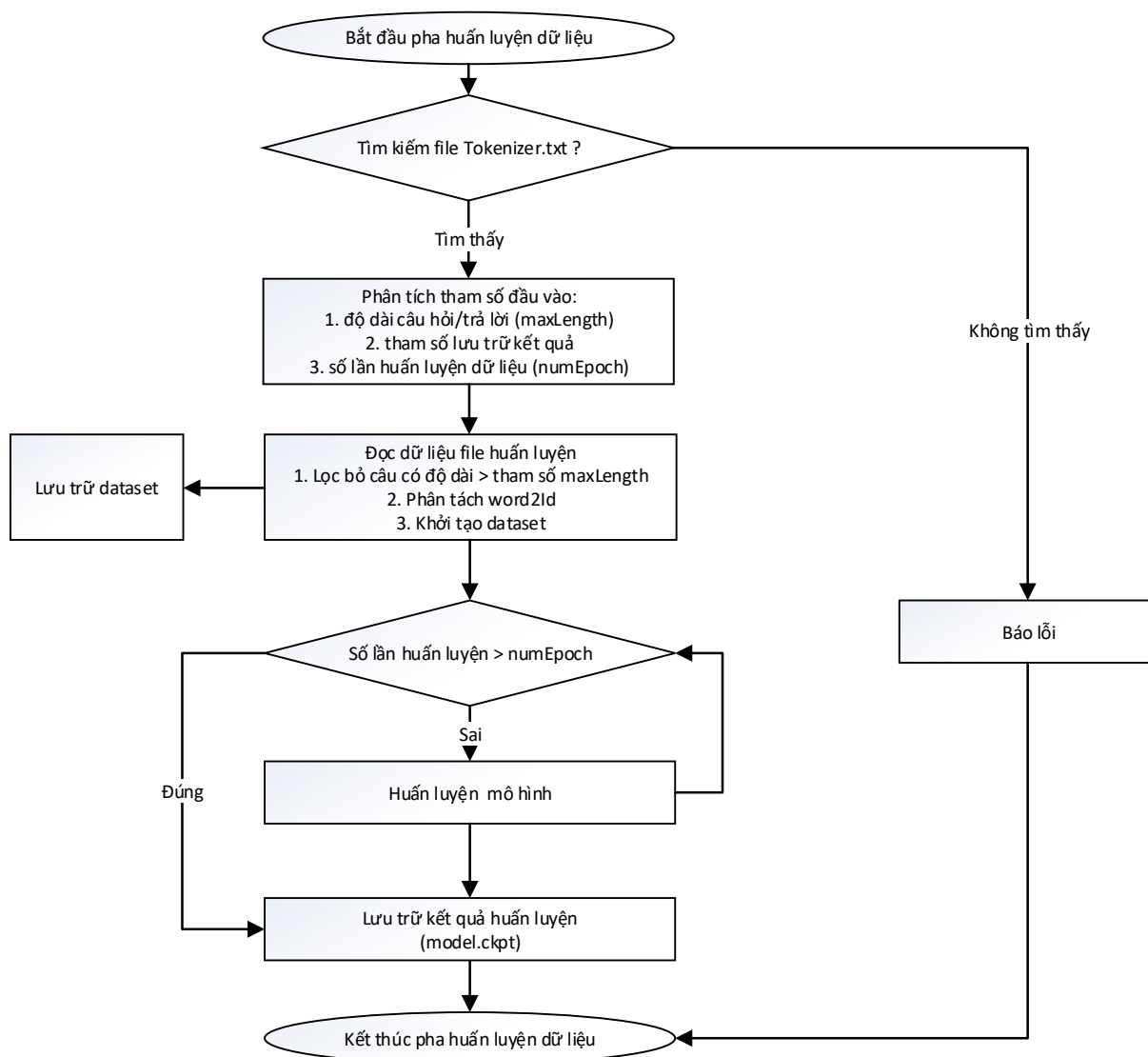
Hình 4.11: Kết quả sample pha phân mảnh dữ liệu

#### 4.3.4 Pha huấn luyện dữ liệu

Pha đóng vai trò quan trọng nhất trong mô hình đó là pha huấn luyện dữ liệu. Sau bước thu thập dữ liệu, 2 pha tiền xử lý và phân mảnh dữ liệu chỉ mang lại tác dụng làm cho kết quả của mô hình được tốt hơn, thì pha huấn luyện này là pha không thể thiếu. Dữ liệu đã được làm sạch, sẽ được sử dụng làm input cho quá trình huấn luyện dữ liệu. Ứng dụng phương pháp học chuỗi seq2seq, dữ liệu thu thập được sẽ được huấn luyện để tạo ra mô hình đề xuất.

Luồng nghiệp vụ của pha này khá phức tạp. Đầu tiên kiểm tra file đầu vào (tokenizer.txt), nếu không có báo lỗi. Bước 2 cần phân tích tham số đầu vào. Cụ thể, cần định nghĩa được một số tham số cho việc huấn luyện dữ liệu như độ dài tối đa của câu hỏi/trả lời sẽ được học, số lần huấn luyện dữ liệu qua mạng LSTM. Dựa vào những tham số này, pha huấn luyện sẽ lựa chọn những câu thỏa mãn yêu cầu, đưa vào mạng LSTM để học với thời gian cụ thể. Bước 3, đọc dữ liệu từ file huấn luyện, tạo thành những cặp câu đối thoại seq2seq như mô tả phần 3.4. Tập hợp của các cặp câu này được gọi là dataset. Bước 4, huấn luyện mô hình với dataset theo số lần yêu cầu được huấn luyện. Và bước cuối cùng là lưu trữ mô hình, hay model.





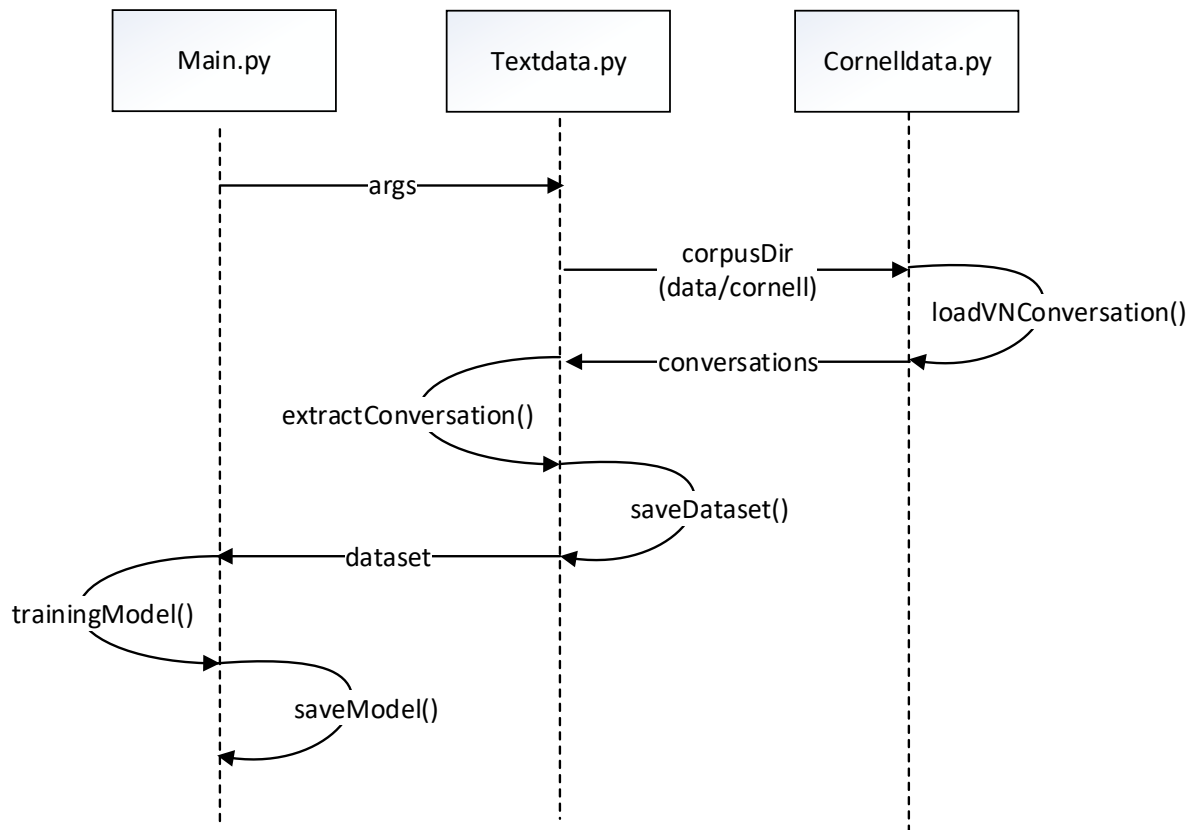
Hình 4.12: Luồng nghiệp vụ pha huấn luyện dữ liệu

Với luồng nghiệp vụ như trên, tôi đề xuất sử dụng ngôn ngữ lập trình python, các thư viện gồm có NLTK - công cụ xử lý ngôn ngữ tự nhiên mã nguồn mở, cho phép xử lý tách văn bản. Framework học máy Tensor Flow, trong phạm vi luận văn, tôi sử dụng phiên bản v0.9.0 được phát hành ngày 22 tháng 6 năm 2016. Kiến trúc phần mềm của pha huấn luyện gồm 3 lớp chính:

Bảng 4.4 Danh sách các cấu phần xử lý trong pha huấn luyện

| Tên lớp        | Mục đích  |
|----------------|---|
| Main.py        | Phân tích dữ liệu đầu vào   |
|                | Huấn luyện mô hình theo số lần yêu cầu                                  |
|                | Lưu trữ mô hình   |
| Cornelldata.py | Đọc dữ liệu từ file đầu vào, lưu trữ thành các câu hội thoại            |
| Textdata.py    | Biến đổi tập các câu hội thoại thành từng cặp mô hình đối thoại seq2seq |
|                | Sử dụng nltk phân tách câu hội thoại thành mã                           |
|                | Lưu trữ dataset   |

Từ quy trình nghiệp vụ của pha, kết hợp với danh mục các cấu phần xử lý, chúng ta có biểu đồ tuần tự như sau:



Hình 4.13: Biểu đồ tuần tự pha huấn luyện dữ liệu

Trong đó loadVNConversation, là hàm đọc dữ liệu từ file kết quả *tokenizer.txt* từ pha phân mảnh dữ liệu. Hàm sẽ đọc từng dòng và đưa vào mảng *conversations*. ExtractConversation là hàm thao tác xử lý mảng *conversations* nhằm tạo ra tập *trainingSamples* chứa các mẫu đối thoại seq2seq, loại bỏ dòng

cuối cùng (do để dòng cuối là câu hỏi thì sẽ không có câu trả lời), ghép từng cặp câu liền kề thành mô hình đối thoại seq2seq, lưu trữ các mô hình đối thoại trên vào mảng thư viện huấn luyện (*trainingSamples*). *SaveDataset*, là hàm thao tác lưu trữ dataset phục vụ huấn luyện. Các thông tin có trong 1 dataset gồm có word2id - object mapping từ sang mã, id2word - object mapping mã sang từ, *trainingSamples* - chứa các mẫu đối thoại seq2seq. Hàm cuối cùng, quan trọng nhất là *trainingModel*: hàm thao tác huấn luyện dựa trên dataset. Với tham số truyền vào *numEpochs*, hàm sẽ thực hiện tương ứng số lần huấn luyện mô hình. Trong mỗi lần huấn luyện, dataset được xáo trộn để tăng tính chính xác cuối cùng. Sau quá trình huấn luyện, hàm tiến hành lưu trữ model.

```

1 def mainTrain(self, sess):
2     """ Training loop. Args: sess: The current running session """
3     # Specific training dependent loading
4     self.textData.makeLighter(self.args.ratioDataset) # Limit the number of training samples
5
6     mergedSummaries = tf.merge_all_summaries() # Define the summary operator (Warning: Won't appear on the tensorboard graph)
7     if self.globStep == 0: # Not restoring from previous run
8         self.writer.add_graph(sess.graph) # First time only
9
10    # If restoring a model, restore the progression bar ? and current batch ?
11    print('Start training...')
12    for e in range(self.args.numEpochs):
13        print("-- Epoch {}/{} ; (lr={})".format(e, self.args.numEpochs, self.args.learningRate))
14        print()
15        batches = self.textData.getBatches()
16        # TODO: Also update learning parameters eventually
17        tic = time.perf_counter()
18        for nextBatch in tqdm(batches, desc="Training"):
19            # Training pass
20            ops, feedDict = self.model.step(nextBatch)
21            assert len(ops) == 2 # training, loss
22            _, loss, summary = sess.run(ops + (mergedSummaries,), feedDict)
23            self.writer.add_summary(summary, self.globStep)
24            self.globStep += 1
25
26            # Checkpoint
27            if self.globStep % self.args.saveEvery == 0:
28                tqdm.write('Checkpoint reached: saving model (don't stop the run)...')
29                self.saveModelParams()
30                self.saver.save(sess, self._getModelName()) # TODO: Put a limit size (ex: 3GB for the modelDir)
31                tqdm.write('Model saved.')
32        toc = time.perf_counter()
33        print("Epoch finished in: {}s".format(toc-tic)) # TODO: Better time format

```

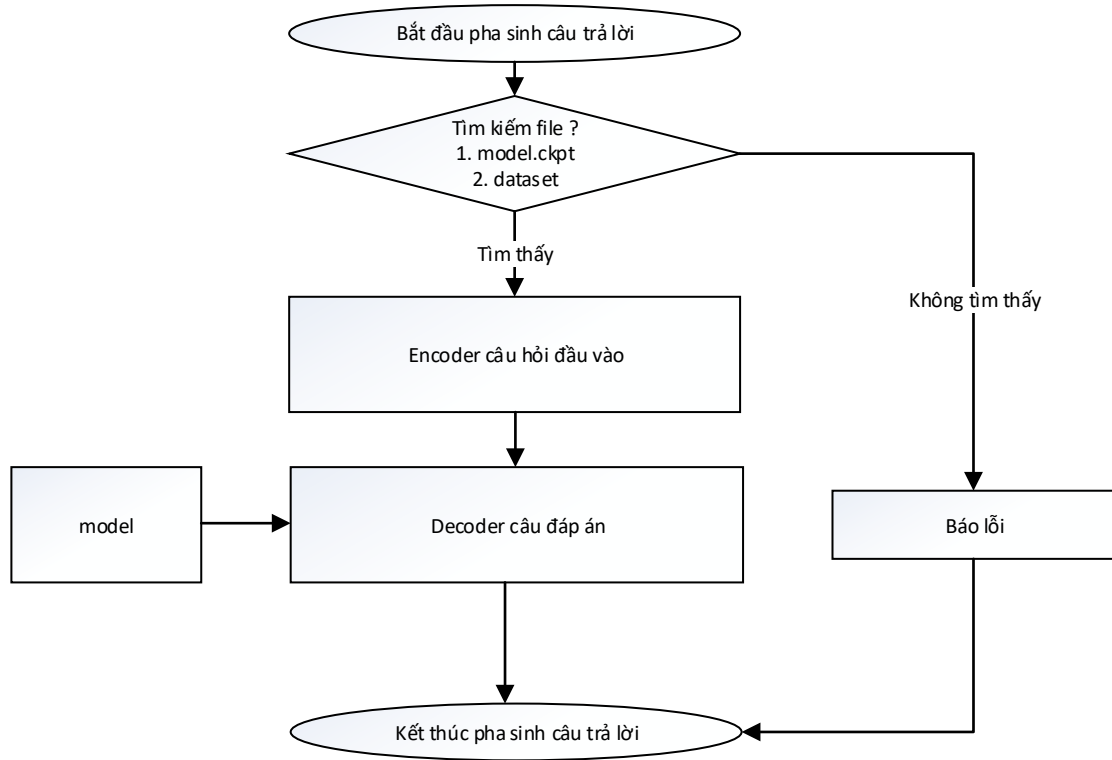
Hình 4.14: Mã nguồn sample cho huấn luyện dữ liệu

Output của pha là dataset được tạo ra từ dữ liệu thu thập và mô hình iSales đã qua huấn luyện. Mô hình này đã giải quyết vấn đề xây dựng một mô hình có thể hiểu được ngôn ngữ tiếng Việt. Ở hàm *extractConversation*, *nltk* được sử dụng để tách các từ thành mã (word2id). Các từ được tách bằng thành mã sẽ không phân biệt ngôn ngữ tiếng Anh hay tiếng Việt. Hơn nữa, pha phân mảnh dữ liệu đã đảm bảo các từ tách ra hoàn toàn dễ hiểu khi đứng một mình.

#### 4.3.5 Pha sinh câu trả lời

Dựa trên mô hình được xây dựng qua bước huấn luyện dữ liệu, có thể ứng dụng mô hình dữ liệu này để dùng cho việc sinh câu trả lời tư vấn cho các câu hỏi đầu vào. Khi câu hỏi được đặt ra, hệ thống sẽ phân tách câu hỏi ra từng từ,

mapping theo dataset được tạo trong pha trước đó. Từ đó sẽ sử dụng bộ *Encoder* và *Decoder* để sinh ra câu trả lời phù hợp với ngữ cảnh nhất.



*Hình 4.15: Luồng nghiệp vụ pha sinh câu trả lời*

Dựa trên hình trên, ta có thể mô tả luồng nghiệp vụ như sau. Bước 1, kiểm tra file đầu vào (model.ckpt/dataset), nếu không có báo lỗi. Bước 2, dựa trên model/dataset từ pha trước, tái kiến trúc lại LSTM. Bước 3, sử dụng bộ encoder để encode câu hỏi. Bước 4, đưa kết quả encode vào LSTM nêu trên. Bước 5, sử dụng bộ decoder của LSTM để sinh ra câu trả lời hợp lý nhất. Ta có thể xây dựng một hàm ứng dụng cụ thể như sau:

```

1 def mainTestInteractive(self, sess):
2     """ Try predicting the sentences that the user will enter in the console
3     Args:
4         sess: The current running session
5     """
6     while True:
7         question = input('Human:')
8         if question == '' or question == 'exit':
9             break
10
11         batch = self.textData.sentence2enco(question)
12         if not batch:
13             print('Warning: sentence too long, sorry. Maybe try a simpler sentence.')
14             continue # Back to the beginning, try again
15         print(self.textData.batchSeq2str(batch.encoderSeqs, clean=True, reverse=True))
16         ops, feedDict = self.model.step(batch)
17         output = sess.run(ops[0], feedDict)
18         answer = self.textData.deco2sentence(output)
19
20         print('iSales:', self.textData.sequence2str(answer, clean=True))
21         print(self.textData.sequence2str(answer))
22         print()

```

Hình 4.16: Mã nguồn sample cho pha sinh câu trả lời

Output của pha là câu trả lời được tạo ra từ mô hình dữ liệu đã qua huấn luyện. Rõ ràng, ở pha này, LSTM đã giúp giải quyết vấn đề phát sinh những câu hỏi nằm ngoài “tập dữ liệu mẫu”: việc huấn luyện mô hình với phương pháp LSTM cho phép mô hình có khả năng hiểu ngữ cảnh, sinh câu trả lời dựa trên các tri thức học được. Một vấn đề khác được xử lý là câu hỏi dài và phức tạp, mạng LSTM giải quyết được bài toán trí nhớ dài hạn. Bên cạnh đó, huấn luyện mô hình bằng Tensorflow cho phép tham số đầu vào là độ dài của câu hỏi/trả lời học. Đó là lời giải thích đáng cho vấn đề này.

iSales là giải pháp đề xuất để giải quyết bài toán mô hình bán hàng tự động. Mô hình này có tổng thể 5 pha, được xây dựng dựa trên phương pháp học chuỗi seq2seq và mạng nơ-ron. Trên lý thuyết, iSales đáp ứng các yêu cầu ban đầu, có khả năng tự học dữ liệu người dùng, tự sinh câu trả lời trong phạm vi thương mại điện tử mà không gặp khó khăn với các câu hỏi dài. Và để có những đánh giá khách quan nhất, tôi sẽ thử nghiệm iSales với dữ liệu thực tế trên mạng, với danh sách các câu hỏi bất kỳ trong chương 5.

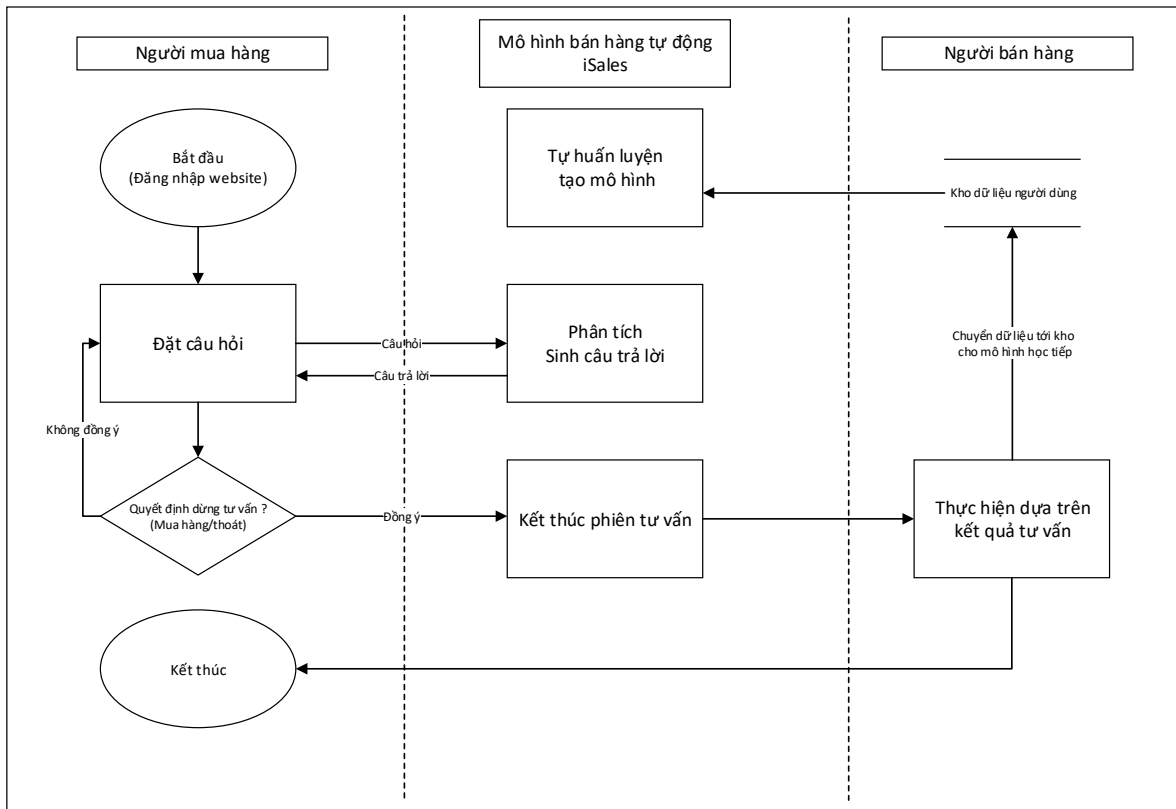
## CHƯƠNG 5: THỬ NGHIỆM, ĐÁNH GIÁ KẾT QUẢ

Ở Chương này, tôi thực hiện thử nghiệm iSales trong một usecase cụ thể. Kết quả thử nghiệm được minh họa thông qua mô hình thực nghiệm, với dữ liệu trên facebook và danh sách một loạt các câu hỏi bất kỳ. Môi trường thử nghiệm hoạt động trên máy tính cá nhân, đã cài đặt tất cả các pha hoạt động.

### 5.1 Phát biểu usecase

Chương 4 đã đưa ra giải pháp và xây dựng mô hình đề xuất iSales. Để đánh giá khách quan cho mô hình, cần có một usecase áp dụng cụ thể. Sau quá trình khảo sát, tìm hiểu, tôi xin đưa ra được quy trình nghiệp vụ của việc bán hàng tự động trên Internet như sau:

- Bước 1: Người mua hàng truy cập vào đường link website bán hàng.
- Bước 2: Người mua hàng xem mặt hàng cần mua, đặt câu hỏi tư vấn.
- Bước 3: Câu hỏi sẽ được gửi đến iSales, iSales sẽ phân tích và trả lời câu hỏi một cách tự động. Người mua hàng nhận được câu trả lời, nếu không thỏa mãn, đặt câu hỏi tiếp theo, iSales tiếp tục công việc tư vấn. Phiên tư vấn này sẽ diễn ra liên tục cho đến khi người mua hàng đồng ý mua hàng hoặc rời khỏi hệ thống.
- Bước 4: Người bán hàng nhận kết quả tư vấn giữa iSales và người mua hàng. Hoặc là xử lý đơn hàng, hoặc là bỏ qua tùy thuộc vào kết quả tư vấn.
- Bước 5: Người bán hàng thực hiện sàng lọc, chuyển dữ liệu tư vấn vào kho dữ liệu cho iSales tự động học tiếp.



Hình 5.1: Quy trình bán hàng tự động iSales

Usecase được đề xuất là người mua hàng muốn tư vấn về sản phẩm “áo sơ mi”. Kịch bản đặt ra là người mua truy cập iSales, đưa ra câu hỏi về chiếc áo sơ mi, đặt các câu hỏi tình huống liên quan đến màu sắc, cỡ, giá của chiếc áo, tiếp đó quyết định mua chiếc áo và yêu cầu ship về địa chỉ nhà riêng. Yêu cầu đặt ra là hệ thống cần thực hiện đầy đủ các pha, phiên tư vấn cần ít nhất 10 câu hỏi.

Trên lý thuyết, phương pháp đánh giá cho những mô hình sử dụng trí tuệ nhân tạo hiện nay là phép thử Turing [15], là bài kiểm tra đánh giá mang tính chất cảm tính. Tuy nhiên, việc đánh giá cảm tính sẽ dựa trên sự hợp lý của các kết quả mà iSales cung cấp. Ngoài ra, đứng dưới phương diện người dùng, tôi sẽ so sánh kết quả của iSales và chatbot Skype để đưa ra thêm một số nhận xét.

## 5.2 Thử nghiệm iSales

Thử nghiệm mô hình iSales trên máy tính cá nhân với 5 pha: thu thập dữ liệu, tiền xử lý, phân mảnh, huấn luyện dữ liệu và sinh câu trả lời. Usecase đặt ra ở phần 5.1 yêu cầu thực hiện đầy đủ và cần ít nhất 10 câu hỏi. Tuy nhiên, khả năng trả lời câu hỏi của iSales cũng như đa phần các ứng dụng trí tuệ nhân tạo khác, đều bị ảnh hưởng bởi dữ liệu huấn luyện, thời gian huấn luyện và thuật toán

sử dụng. Để có kết quả chi tiết hơn, tôi sẽ áp dụng usecase trên với 3 kết quả huấn luyện khác nhau về thời gian.

Với mong muốn đảm bảo lượng dữ liệu tạm chấp nhận cho quá trình huấn luyện, tôi thực hiện pha thu thập dữ liệu trên hệ điều hành window với danh sách 10 page bán hàng như sau.

*Bảng 5.1 Danh sách các page facebook thu thập dữ liệu*

| Tên page            | Mô tả                    |
|---------------------|--------------------------|
| cuahangcongnghecom  | Trang bán hàng công nghệ |
| laroseshop1990      | Trang bán hàng quần áo   |
| Hilheoshop96        | Trang bán hàng quần áo   |
| 92wear              | Trang bán hàng quần áo   |
| banhangtructuyen360 | Trang bán hàng quần áo   |
| ChilloutVillage.vn  | Trang bán hàng quần áo   |
| ChoNinhHiep.net     | Trang bán hàng quần áo   |
| T.Shirt.Pro         | Trang bán hàng quần áo   |
| adoredressdesign    | Trang bán hàng quần áo   |
| cloud.cuckoo.shop   | Trang bán hàng quần áo   |

Kết quả của pha là file collection.txt với số lượng dòng comment thu được 5683 dòng. Nhận xét chung là dữ liệu xấu, có dòng trống, có comment không có ý nghĩa, xuất hiện các dòng lặp liên kề nhau. Thực hiện tiếp pha tiền xử lý dữ liệu, kết quả thu được là file preprocess.txt với 5148 dòng dữ liệu tạm ổn, không còn dòng trống, các dòng lặp bị xóa, comment không có ý nghĩa bị loại bỏ. Output của pha phân mảnh dữ liệu là tokenizer.txt, với số dòng dữ liệu không thay đổi, tuy nhiên dữ liệu không còn dòng trống, các dòng lặp bị xóa, comment không có ý nghĩa bị loại bỏ, các từ đã được tách theo chuẩn Tiếng Việt. Mặc dù còn có một số câu tách chưa chuẩn như “Dạ giá 850k\_à”...nhưng đánh giá sơ bộ kết quả này có thể áp dụng cho huấn luyện dữ liệu.

Hai pha cuối cùng là huấn luyện dữ liệu và sinh câu trả lời được thử nghiệm trên VMware giả lập Ubuntu 14.04. Do khác biệt môi trường nên cần chuyển file kết quả của pha phân mảnh dữ liệu (tokenizer.txt) sang thư mục /Home/Desktop/GR/DeepQA/data/cornell trong môi trường giả lập Ubuntu.



Theo như mở đầu của phần, để có những đánh giá tốt hơn, tôi sẽ thực hiện huấn luyện dữ liệu với 3 trường hợp sau. Trường hợp 1, chỉ chấp nhận những hỏi với độ dài hội thoại là 5 với số vòng huấn luyện là 10. Trường hợp 2, nâng độ dài hội thoại lên 10, nâng số vòng huấn luyện lên 20. Trường hợp 3, nâng độ dài hội thoại lên 15, nâng số vòng huấn luyện lên 30.

Kết quả huấn luyện có thể liệt kê như bảng sau. Rõ ràng, cùng một khối dữ liệu như nhau, nếu tăng các tham số huấn luyện lên kéo theo thời gian huấn luyện tăng lên.

*Bảng 5.2 Bảng kết quả huấn luyện dữ liệu*

|                      | Trường hợp 1                  | Trường hợp 2                   | Trường hợp 3                   |
|----------------------|-------------------------------|--------------------------------|--------------------------------|
| Tham số huấn luyện   | maxlength = 5<br>numEpochs=10 | maxlength = 10<br>numEpochs=20 | maxlength = 15<br>numEpochs=30 |
| Dữ liệu huấn luyện   | 4460 từ<br>3896 cặp tư vấn    | 4460 từ<br>3896 cặp tư vấn     | 4460 từ<br>3896 cặp tư vấn     |
| Thời gian huấn luyện | ~32 phút                      | ~ 1 giờ 14 phút                | ~ 3 giờ 20 phút                |

Thực hiện sinh câu trả lời với danh sách gồm 15 câu hỏi về usecase mua áo sơ mi, gồm các câu hỏi liên quan đến màu sắc, cỡ, giá cả của áo, yêu cầu các thông tin khi ship về địa phương. Các câu hỏi được đặt một cách ngẫu nhiên

*Bảng 5.3 Danh sách câu hỏi thử nghiệm*

| Câu hỏi thử nghiệm                   | Mô tả                   |
|--------------------------------------|-------------------------|
| xin chào                             | Câu chào hỏi            |
| mình muốn mua áo sơ mi               | Câu mô tả mục đích      |
| có màu trắng không                   | Yêu cầu tư vấn màu sắc  |
| còn màu đen không                    | Yêu cầu tư vấn màu sắc  |
| còn màu khác không                   | Yêu cầu tư vấn màu sắc  |
| cỡ áo l còn không                    | Yêu cầu tư vấn kích cỡ  |
| cỡ áo s hết à bạn                    | Yêu cầu tư vấn kích cỡ  |
| áo trắng bn ạ                        | Yêu cầu tư vấn giá      |
| giá màu đen bao nhiêu                | Yêu cầu tư vấn giá      |
| giá chiếc màu trắng bao nhiêu        | Yêu cầu tư vấn giá      |
| thế là mình tự phải ra mua đúng ko ạ | Yêu cầu tư vấn địa điểm |
| khi nào thì hàng đi vậy              | Yêu cầu tư vấn địa điểm |

|   |                         |
|---|-------------------------|
| nếu ship ra tỉnh khác thì mất thêm khoảng bao nhiêu vậy bạn | Yêu cầu tư vấn địa điểm |
| ship về vĩnh phúc hết bao lâu                               | Yêu cầu tư vấn địa điểm |
| chuyển qua bình xuyên, vĩnh phúc bạn nha                    | Yêu cầu tư vấn địa điểm |

Kết quả sinh câu trả lời không được tốt. Với các trường hợp test có maxLength nhỏ, thì iSales sẽ không trả lời. Những câu khác trả lời được nhưng câu trả lời không hợp lý.

*Bảng 5.4 Thử nghiệm sinh câu trả lời trong trường hợp 1*

|   |
|---|
| <b>Human:</b> xin chào  |
| <b>iSales (case01):</b> yeah  |
| <b>Human:</b> mình muốn mua áo sơ mi                                      |
| <b>iSales (case01):</b>   |
| <b>Human:</b> có màu trắng không  |
| <b>iSales (case01):</b> giá ạ   |
| <b>Human:</b> còn màu đen không   |
| <b>iSales (case01):</b> ok  |
| <b>Human:</b> còn màu khác không  |
| <b>iSales (case01):</b> ok  |
| <b>Human:</b> cỡ áo l còn không   |
| <b>iSales (case01):</b> 40  |
| <b>Human:</b> cỡ áo s hết à bạn   |
| <b>iSales (case01):</b>   |
| <b>Human:</b> áo trắng bn ạ   |
| <b>iSales (case01):</b> mũ 180k nhé                                       |
| <b>Human:</b> giá màu đen bao nhiêu                                       |
| <b>iSales (case01):</b> Áo 180k bn nha                                    |
| <b>Human:</b> giá chiếc màu trắng bao nhiêu                               |
| <b>iSales (case01):</b>   |
| <b>Human:</b> thế là mình tự phải ra mua đúng ko ạ                        |
| <b>iSales (case01):</b>   |
| <b>Human:</b> khi nào thì hàng đi vậy                                     |
| <b>iSales (case01):</b>   |
| <b>Human:</b> nếu ship ra tỉnh khác thì mất thêm khoảng bao nhiêu vậy bạn |
| <b>iSales (case01):</b>   |
| <b>Human:</b> ship về vĩnh phúc hết bao lâu                               |
| <b>iSales (case01):</b>   |
| <b>Human:</b> chuyển qua bình xuyên, vĩnh phúc bạn nha                    |
| <b>iSales (case01):</b>   |

Bảng 5.5 Thử nghiệm sinh câu trả lời trong trường hợp 2

|                          |  |
|--------------------------|--|
| <b>Human:</b>            | <i>xin chào</i>  |
| <b>iSales (case02):</b>  | <i>vy ngoc</i>   |
| <b>Human:</b>            | <i>mình muốn mua áo sơ mi</i>                                      |
| <b>iSales (case02) :</b> | <i>không</i>   |
| <b>Human:</b>            | <i>có màu trắng không</i>  |
| <b>iSales (case02):</b>  | <i>có nhé</i>  |
| <b>Human:</b>            | <i>còn màu đen không</i>   |
| <b>iSales (case02):</b>  | <i>bn bn shop</i>  |
| <b>Human:</b>            | <i>còn màu khác không</i>  |
| <b>iSales (case02):</b>  | <i>ok</i>  |
| <b>Human:</b>            | <i>cỡ áo l còn không</i>   |
| <b>iSales (case02):</b>  | <i>hiccc</i>   |
| <b>Human:</b>            | <i>cỡ áo s hết à bạn</i>   |
| <b>iSales (case02):</b>  | <i>hiccc</i>   |
| <b>Human:</b>            | <i>áo trắng bn ạ</i>   |
| <b>iSales (case02):</b>  | <i>mũ 180k nhé</i>   |
| <b>Human:</b>            | <i>giá màu đen bao nhiêu</i>                                       |
| <b>iSales (case02):</b>  | <i>chất riêng nhé</i>  |
| <b>Human:</b>            | <i>giá chiếc màu trắng bao nhiêu</i>                               |
| <b>iSales (case02):</b>  | <i>giá ạ</i>   |
| <b>Human:</b>            | <i>thế là mình tự phải ra mua đúng ko ạ</i>                        |
| <b>iSales (case02):</b>  | <i>40</i>  |
| <b>Human:</b>            | <i>khi nào thì hàng đi vậy</i>                                     |
| <b>iSales (case02):</b>  | <i>yeah</i>  |
| <b>Human:</b>            | <i>nếu ship ra tỉnh khác thì mất thêm khoảng bao nhiêu vậy bạn</i> |
| <b>iSales (case02):</b>  |  |
| <b>Human:</b>            | <i>ship về vĩnh phúc hết bao lâu</i>                               |
| <b>iSales (case02):</b>  | <i>giá 550k ạ</i>  |
| <b>Human:</b>            | <i>chuyển qua bình xuyên, vĩnh phúc bạn nha</i>                    |
| <b>iSales (case02):</b>  | <i>dạ không ạ</i>  |

Bảng 5.6 Thử nghiệm sinh câu trả lời trong trường hợp 3

|                          |  |
|--------------------------|--|
| <b>Human:</b>            | <i>xin chào</i>  |
| <b>iSales (case03):</b>  | <i>đẹp</i>   |
| <b>Human:</b>            | <i>mình muốn mua áo sơ mi</i>                                      |
| <b>iSales (case03) :</b> | <i>ok ạ</i>  |
| <b>Human:</b>            | <i>có màu trắng không</i>  |
| <b>iSales (case03):</b>  | <i>láng ah</i>   |
| <b>Human:</b>            | <i>còn màu đen không</i>   |
| <b>iSales (case03):</b>  | <i>ok</i>  |
| <b>Human:</b>            | <i>còn màu khác không</i>  |
| <b>iSales (case03):</b>  | <i>ok</i>  |
| <b>Human:</b>            | <i>cỡ áo l còn không</i>   |
| <b>iSales (case03):</b>  | <i>còn b nhé</i>   |
| <b>Human:</b>            | <i>cỡ áo s hết à bạn</i>   |
| <b>iSales (case03):</b>  | <i>hiccc</i>   |
| <b>Human:</b>            | <i>áo trắng bn ạ</i>   |
| <b>iSales (case03):</b>  | <i>mũ 80k ạ</i>  |
| <b>Human:</b>            | <i>giá màu đen bao nhiêu</i>                                       |
| <b>iSales (case03):</b>  | <i>sỉ vâng</i>   |
| <b>Human:</b>            | <i>giá chiếc màu trắng bao nhiêu</i>                               |
| <b>iSales (case03):</b>  | <i>40</i>  |
| <b>Human:</b>            | <i>thế là mình tự phải ra mua đúng ko ạ</i>                        |
| <b>iSales (case03):</b>  | <i>40</i>  |
| <b>Human:</b>            | <i>khi nào thì hàng đi vậy</i>                                     |
| <b>iSales (case03):</b>  | <i>yeah</i>  |
| <b>Human:</b>            | <i>nếu ship ra tỉnh khác thì mất thêm khoảng bao nhiêu vậy bạn</i> |
| <b>iSales (case03):</b>  | <i>dạ vâng ạ</i>   |
| <b>Human:</b>            | <i>ship về vĩnh phúc hết bao lâu</i>                               |
| <b>iSales (case03):</b>  | <i>giá 550k ạ</i>  |
| <b>Human:</b>            | <i>chuyển qua bình xuyên, vĩnh phúc bạn nha</i>                    |
| <b>iSales (case03):</b>  | <i>dạ không ạ</i>  |

### 5.3 Đánh giá kết quả

Sau khi thực hiện thử nghiệm mô hình iSales trên usecase mua áo sơ mi, kết quả thu được khi đặt câu hỏi cho iSales là tỷ lệ đáp án chấp nhận được quá thấp < 10%. Tuy nhiên, iSales đã giải quyết được 1 số vấn đề:

- Vấn đề phát sinh những câu hỏi nằm ngoài “tập dữ liệu mẫu”, phân câu hỏi đặt ra hoàn toàn mang tính chất khách quan. Tuy nhiên có thể thấy, iSales đã tự đưa ra câu trả lời được với tất cả các câu hỏi phù hợp.

- Vấn đề xây dựng một mô hình có thể hiểu được ngôn ngữ tiếng Việt, iSales đã có thể hiểu được câu hỏi bằng tiếng Việt, và trả lời bằng tiếng Việt.
- Vấn đề câu hỏi dài và phức tạp, tùy thuộc vào tham số độ dài huấn luyện, iSales đã thể hiện như trong 3 trường hợp. Huấn luyện với tham số độ dài càng cao, thì độ dài câu mà iSales càng có khả năng đưa ra câu trả lời.
- Vấn đề “tự động”, iSales đã làm được điều này, nhưng đáp án đưa ra chưa tốt, cần cải thiện rất nhiều.

Rõ ràng, với những kết quả hiện tại, iSales chưa thể trở thành ứng dụng trên thị trường. Luận văn thành công ở phương diện phương án xây dựng bằng một phương pháp mới, nhưng vẫn chưa đảm bảo được kết quả tốt. Nếu so sánh với chatbot Skype dưới con mắt người dùng, chatbot Skype vẫn tốt hơn, vì có thể hoàn thành được phiên giao dịch nếu người mua hàng làm đúng theo chỉ dẫn của nó. Tuy nhiên, iSales cũng là một thử thách hay trong lĩnh vực ứng dụng mạng nơ-ron nói riêng và trong ngành vực trí tuệ nhân tạo nói chung.

## KẾT LUẬN

Luận văn tập trung tìm hiểu thực trạng về các mô hình bán hàng trên thế giới và Việt Nam, đưa ra các nhược điểm, các thành phần cần cải tiến, từ đó đề xuất giải pháp, xây dựng mô hình bán hàng iSales có tính tự động, không phụ thuộc vào người bán hàng trong quá trình tư vấn dịch vụ. Mô hình iSales hiện tại, chưa thể đưa ra thành ứng dụng thương mại. Tuy có thể đáp ứng được các yêu cầu có khả năng hiểu Tiếng Việt và sinh câu trả lời tự động với độ dài câu hỏi không giới hạn, nhưng kết quả tư vấn iSales đưa ra còn ngô nghê, khó hiểu.

Hướng nghiên cứu tiếp theo của luận văn tập trung vào việc cải thiện chất lượng câu trả lời của iSales.

- Mở rộng khả năng thu thập và tiền xử lý dữ liệu làm cho dữ liệu thu thập được nhiều hơn, sạch hơn. Phương án đưa ra là sử dụng BigData để quản lý dữ liệu và cho phép tìm kiếm, lọc dữ liệu nhanh hơn [12], hoặc lựa chọn các nguồn dữ liệu như dữ liệu sau tư vấn được lưu trữ của các doanh nghiệp.
- Cải tiến pha huấn luyện dữ liệu để giảm thời gian huấn luyện và tăng hiệu quả đầu ra của mô hình.

Đồng thời, tôi muốn phối hợp nghiên cứu và triển khai ứng dụng với các đơn vị bên ngoài, đầu tư mở rộng phần cứng để giảm thời gian huấn luyện. Những hướng phát triển trên nhằm hướng tới một mục đích chung là phát triển hệ thống iSales có tầm thế. Từ đó đóng góp, phát triển nền trí tuệ nhân tạo thế giới nói chung và Việt Nam nói riêng .

## TÀI LIỆU THAM KHẢO

- [1] Facebook Messenger, September 2016, “Facebook Messenger”.
- [2] Microsoft, 31 Mar 2016, “Build 2016: Microsoft Skype bots preview announced”.
- [3] Martín Abadi, 14 Mar 2016, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”.
- [4] Google Blog, 09 November 2015, “TensorFlow - Google’s latest machine learning system, open sourced for everyone”.
- [5] S.M. Al-Alawi, H.A. Al-Hinai, May–August 1998, “An ANN-based approach for predicting global radiation in locations with no direct measurement instrumentation”.
- [6] Ondřej Dušek, Filip Jurčiček, 17 Jun 2016, “Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings”.
- [7] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, May 2013, “On the difficulty of training recurrent neural networks”.
- [8] James Ryan, September 2016, “Translating Player Dialogue into Meaning Representations Using LSTMs”.
- [9] Jay Parikh, August 2012, “Facebook processes more than 500 TB of data daily”.
- [10] Facebook, August 2016, “The Graph API”.
- [11] Lư Tuấn Anh, Yamamoto Kazuhide, 16 Feb 2013, “Pointwise for Vietnamese Word Segmentation”.
- [12] Xue-Wen Chen, Xiaotong Lin, 16 May 2014, “Big Data Deep Learning: Challenges and Perspectives”.
- [13] Francois Chaubard, Rohit Mundra, Richard Socher, Spring 2015, “CS 224D: Deep Learning for NLP”.
- [14] Sepp Hochreiter; Jürgen Schmidhuber, 1997, "Long short-term memory".
- [15] Feigenbaum, Edward A. 2003, “Some challenges and grand challenges for computational intelligence”.