TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Phân loại ý định người dùng theo ngữ cảnh trong hệ thống hội thoại

NGUYỄN THỊ MỪNG
mung.nt162768@sis.hust.edu.vn

Ngành Kỹ thuật phần mềm

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang

Bộ môn: Công nghệ phần mềm
Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 01/2021

Lời cam kết

Họ và tên sinh viên: Nguyễn Thị Mừng

Điện thoại liên lạc: 0394338777 Email: mung.nt162768@sis.hust.edu.vn

Lớp: CNTT2.01 Hệ đào tạo: Kỹ sư chính quy

Tôi – *Nguyễn Thị Mừng* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Thị Thu Trang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 08 tháng 01 năm 2021

Nguyễn Thị Mừng

Lời cảm ơn

Lời đầu tiên, em xin phép gửi lời cảm ơn chân thành và sâu sắc nhất đến TS. Nguyễn Thị Thu Trang. Bằng cả sự tâm huyết và kiên nhẫn của mình, cô đã giúp em định hướng đề tài, đưa ra những gợi ý, chỉ dẫn chi tiết và tạo điều kiện tốt nhất cho em hoàn thành đồ án này. Cô giống như người mẹ ân cần và chu đáo nhưng cũng đầy nghiêm khắc, đôi khi lại như người bạn để chúng em có thể dễ dàng tâm sự và chia sẻ những khó khăn của mình. Dưới sự hướng dẫn của cô, em cảm thấy bản thân mình đã tiến bộ hơn rất nhiều.

Em xin gửi lời cảm ơn chân thành tới ban lãnh đạo, các thầy cô giáo trong trường Đại học Bách Khoa Hà Nội nói chung và trong Viện Công nghệ thông tin và Truyền thông nói riêng đã giúp em có cơ hội được học tập trong một môi trường bổ ích và đáng nhớ trong quãng đời sinh viên của mình.

Em cũng muốn gửi lời cảm ơn đến các anh chị, bạn bè, các em trong phòng thí nghiệm 914 và các đối tác, đặc biệt là anh Nguyễn Hoàng Kỳ, em Đặng Trung Đức Anh và các anh chị chuyên viên dữ liệu về ngân hàng. Cảm ơn mọi người đã hướng dẫn chi tiết, giúp đỡ tận tình và động viên em trong thời gian em tham gia phòng thí nghiệm cũng như trong thời gian làm đồ án. Cùng với đó, em xin gửi lời cảm ơn đến bạn bè trong và ngoài Viện Công nghệ thông tin và Truyền thông đã quan tâm, chia sẻ và giúp đỡ em trong thời gian qua.

Cuối cùng, em muốn gửi gắm lời một cảm ơn thân thương đến gia đình của mình. Con cảm ơn bố mẹ, bác dâu và gia đình đã luôn yêu thương, quan tâm và là chỗ dựa tinh thần, là nguồn động lực lớn lao để con có thể vượt qua những khó khăn, thử thách của bản thân.

Trong quá trình làm đồ án tốt nghiệp, dù đã đã cố gắng hết sức nhưng vẫn sẽ không thể tránh được sai sót, em rất mong nhận được góp ý của thầy cô và các bạn để em sẽ không gặp phải những lỗi này về sau.

Một lần nữa, em xin chân thành cảm ơn!

Tóm tắt

Trong hội thoại giữa người và máy, con người thường bày tỏ ý định của mình thông qua các câu nói. Chính vì vậy, việc dự đoán ý định của con người là một trong những bài toán xử lý ngôn ngữ tự nhiên chính trong những hội thoại như thế. Bài toán này đặc biệt có thêm thách thức khi phải nhớ được các ngữ cảnh từ những trao đổi trước đó. Các nghiên cứu gần đây sử dụng các mô hình học sâu hiện đại để học ngữ cảnh ẩn thông qua dữ liệu dạng hội thoại. Tuy nhiên, với cách tiếp cận này, ta cần cung cấp một lượng lớn dữ liệu huấn luyện đồng thời không thể trực tiếp kiểm soát hoặc đưa vào các ngữ cảnh mong muốn, mà hoàn toàn phục thuộc vào mô hình đã huấn luyện.

Để giải quyết bài toán trên, đồ án lựa chọn hướng tiếp cận quản lý và mô hình hoá ngữ cảnh một cách trực tiếp, lấy ý tưởng từ cách Google Dialogflow quản lý ngữ cảnh. Đồ án đề xuất mô hình dự đoán ý định của người dùng, và đặc biệt có thể dự đoán được ý định sử dụng các thông tin ngữ cảnh được nhớ trong những trao đổi trước đó trong cuộc hội thoại.

Đồ án đã nghiên cứu và thực nghiệm các thuật toán để xây dựng mô hình dự đoán ý định của người dùng trong hội thoại, trong đó phân tích về độ tự tin của phép đoán, nhằm cung cấp thông tin để thực hiện các hành động phản hồi cho người dùng trong cuộc hội thoại. Đồ án tiến hành đánh giá các mô hình học máy và học sâu dựa trên tiêu chí về chênh lệch độ tự tin giữa các dự đoán đúng và sai, độ chính xác và thời gian dự đoán của mô hình. Kết quả đánh giá cho thấy, Véc-tơ máy hỗ trợ (Support Vector Machine – SVM) [1] là mô hình phù hợp nhất với độ chính xác 93,25%, chênh lệch độ tự tin 0,43 và thời gian dự đoán 0,03 giây.

Để có thể sử dụng được các ngữ cảnh trước đó của hội thoại, đồ án đề xuất mô hình 2 bước như sau: bước 1 sử dụng câu nói đầu vào cũng các thực thể xuất hiện trong câu để dự đoán ý định cho các câu đầy đủ thông tin và phát hiện nhập nhằng, bước 2 đưa ngữ cảnh vào dự đoán khi có nhập nhàng xảy ra. Kết quả cho thấy mô hình đề xuất đạt độ chính xác 80,03% đối với các câu dễ nhập nhằng giữa nhiều ý định trong khi mô hình trước đó hoàn toàn không xử lý được. Mô hình cũng hoạt động ổn định như với mô hình dự đoán không dùng ngữ cảnh trên tập dữ liệu có thông tin rõ ràng với độ chính xác 87,88%.

Mô hình dự đoán hai bước hiện đang được triển khai thử nghiệm vào nền tảng Smartdialog [2], ứng dụng trong việc xây dựng hệ thống hội thoại trong lĩnh vực ngân hàng.

Mục lục

Danh mục hình vẽ

Danh mục bảng

Danh mục công thức

Danh mục các từ viết tắt

NLP	Natural Language Processing Xử lý ngôn ngữ tự nhiên
SVM	Support Vector Machine Véc tơ máy hỗ trợ
BiLSTM	Bidirectional Long-Short Term Memory Mạng bộ nhớ dài ngắn hai chiều
CNN	Convolutional Neural Network Mạng nơ-ron tích chập
RNN	Recurrent Neural Network Mạng nơ-ron hồi quy
LSTM	Long-Short Term Memnory Mạng bộ nhớ dài ngắn
ASR	Automatic Speech Recognition Nhận dạng giọng nói tự động
STT	Speech to text Chuyển giọng nói sang văn bản
TF-IDF	Term frequency – inverse document frequency

TTS	Text to Speech Chuyển từ văn bản sang giọng nói
BOW	Bag-of-words Túi từ
CBOW	Common Bag of Word
NLU	Natural Language Understanding Hiểu ngôn ngữ tự nhiên

Danh mục thuật ngữ

Hội thoại	Là một hệ thống có khả năng giao tiếp với con người thông qua ngôn ngữ tự nhiên.
Ý định	Mục đích trò chuyện của người dùng.
Thực thể	Các thông tin quan trọng trong các câu nói của người dùng.
Ngữ cảnh	Là bối cảnh ngôn ngữ, ở đó, sản phẩn ngôn ngữ (văn bản) được tạo ra tron hoạt động giao tiếp, đồng thời là bối cảnh cần dựa vào để lĩnh hội và thấu hiểu sản phẩm ngôn ngữ đó.
Mô hình	Một đối tượng được huấn luyện để có thể nhận ra một số mẫu nhất định.

Chương 1 Bài toán phân loại ý định người dùng theo ngữ cảnh

1.1 Đặt vấn đề

Hệ thống hội thoại là một chương trình máy tính có khả năng giao tiếp với con người bằng ngôn ngữ tự nhiên [3]. Hiện nay, hệ thống hội thoại đang ngày càng nhận được sự quan tâm từ các công ty công nghệ trên khắp thế giới bởi khả năng ứng dụng của nó trong nhiều lĩnh vực khác nhau như nhà thông minh, xe tự hành, trợ lý ảo, chăm sóc sức khoẻ và chăm sóc khách hàng. Các trợ lý thông minh như Siri của Apple [4], Google Now của Google [5], Cortana của Microsoft,... đã và đang thay đổi cách thức chúng ta giao tiếp với máy tính và tìm kiếm thông tin. Các nền tảng hỗ trợ xây dựng các hệ thống hội thoại như Google Dialogflow [6], Microsoft Bot Framework, RASA [7], Oracle Digital Assistant [8]... giúp cho các cá nhân, tổ chức và doanh nghiệp có thể dễ dàng xây dựng một hệ thống hội thoại của riêng mình, từ đó nâng cao khả năng tiếp cận tập khách hàng mục tiêu, giảm chi phí nhân lực và nâng cao trải nghiệm người dùng.

Một hệ thống hội thoại, với đầu vào là văn bản, cử chỉ hoặc giọng nói, gồm nhiều thành phần được kết nối với nhau, trong đó, thành phần hiểu ngôn ngữ tự nhiên (Natural Language Understanding – NLU) và thành phần quản lý hội thoại (Dialog Manager) là hai thành phần có vai trò quan trọng nhất. Thành phần hiểu ngôn ngữ tự nhiên có nhiệm vụ dự đoán ý định và trích xuất các thông tin cần thiết từ câu nói của người dùng. Các thông tin này còn được gọi là thực thể. Dựa vào ý định và các thực thể được trích xuất, thành phần quản lý hội thoại sẽ phân tích để quyết định hành động tiếp theo của hệ thống. Ví dụ, khi một người hỏi "Hôm nay thời tiết ở Hà Nội thế nào", thành phần NLU đưa ra phán đoán về ý định của người dùng là "Hỏi thời tiết" và trích xuất các thông tin quan trọng bao gồm thời gian ("hôm nay") và địa điểm ("Hà Nội"). NLU sẽ chuyển các thông tin này cho thành phần quản lý hội thoại. Khi nhận được thông tin, dựa vào thời gian và địa điểm đã biết, thành phần quản lý hội thoại tiến hành tra cứu thông tin và đưa ra phản hồi "Mô tả thời tiết", tương ứng với ý định "Hỏi thời tiết" của người dùng.

Việc phán đoán sai ý định của người dùng có thể dẫn đến việc đưa ra phản hồi không chính xác. Vì thế nên hệ thống cần biết về độ tự tin của phép đoán để đưa ra phản hồi phù hợp. Dựa trên độ tự tin của phép đoán, hệ thống sẽ quyết định hành động cần thực hiện tiếp theo

là gì. Chẳng hạn như với Google Dialogflow và RASA, họ đưa ra một ngưỡng tự tin mà tại ngưỡng đó để xác định mức độ tin cậy tối thiểu cần thiết để phân loại câu nói của người dùng vào một ý định [9], [10]. Nếu như không có ý định nào có độ tự tin vượt qua được ngưỡng này thì hệ thống sẽ thực hiện một hành động mặc định. Hành động này có thể là một lời xin lỗi như "Xin lỗi, mình chưa hiểu ý của bạn" hoặc chuyển tiếp cuộc hội thoại cho nhân viên chăm sóc khách hàng. Nếu có nhiều hơn một ý định có độ tự tin vượt ngưỡng, hệ thống chọn ý định có độ tự tin cao nhất, sử dụng một tập luật để tìm ra ý định phù hợp nhất [9] hoặc hỏi lại người dùng [11]. Tuy nhiên, việc xác định một ngưỡng hợp lý không phải là một việc dễ dàng. Thậm chí chúng ta có thể không tìm được ngưỡng phù hợp nếu như thuật toán phân loại luôn đưa ra độ tự tin rất cao hoặc rất thấp cho các phán đoán bất kể phán đoán đó đúng hay sai. Do đó, chúng ta cần đánh giá thuật toán phân loại dựa trên các tiêu chí về độ tự tin để có thể lựa chọn một thuật toán vừa đảm bảo về chất lượng vừa có khả năng giúp chúng ta tìm ra một ngưỡng phù hợp. Hiện nay, các nghiên cứu về bài toán phân loại ý định người dùng sử dụng các thuật toán học máy như Naïve Bayes [12], AdaBoost [13], Random Forest [14], Véc-tơ máy hỗ trợ (Support Vector Machine – SVM) [15], Logistic Regression [16] và các thuật toán học sâu như mạng nơ-ron tích chập (Convolution Neural Networks – CNN) [17], mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN), mạng bộ nhớ dài ngắn (Long Short-Term Memory - LSTM) [18], Bidirectional Long-Short Term Memory (BiLSTM) [19], RNN và cơ chế chú ý (Attention Mechanism) [20] và Capsule Networks [21] chủ yếu sử dụng độ chính xác hoặc điểm F_1 (F_1-score) để đánh giá hiệu quả của thuật toán phân loại. Những độ đo này chỉ thể hiện một cách khái quát tỷ lệ các phán đoán đúng mà chưa đánh giá được độ tự tin của các phán đoán. Vì vậy, trong đồ án này, em đề xuất một cách đánh giá các thuật toán phân loại dựa trên độ tự tin để lựa chọn ra thuật toán phù hợp nhất trong việc xây dựng một hệ thống hội thoại.

Bên cạnh đó, việc dự đoán ý định trong nhiều trường hợp cần sử dụng những thông tin đã trao đổi trước đó giữa người dùng và hệ thống hội thoại. Ví dụ như với một hệ thống hội thoại trong lĩnh vực ngân hàng, khi người dùng hỏi "Đăng kí internet banking như thế nào?", hệ thống có thể lập phán đoán được ý định của người dùng là "Đăng kí Internet Banking" và đưa ra hướng dẫn. Sau đó, người dùng tiếp tục hỏi "Thế còn sms banking thì sao?". Lúc này, hệ thống cần hiểu rằng người dùng đang muốn đăng kí dịch vụ SMS Banking. "Đăng kí" trong ví dụ này được gọi là ngữ cảnh. Hiện tại có hai phương pháp chính để sử dụng ngữ cảnh trong việc dự đoán ý định người dùng là phương pháp trọn gói (end-to-end) học ngữ cảnh ẩn thông qua dữ liệu dạng hội thoại và phương pháp quản lý và mô hình hóa ngữ cảnh một cách trực tiếp như cách mà Google Dialogflow quản lý ngữ cảnh trong hệ thống.

Đi theo cách tiếp cận thứ nhất, M. Mensio và cộng sự sử dụng mô hình RNN hai lớp để học thông tin ngữ cảnh thông qua nội dung trò chuyện của một đoạn hội thoại [22]. Trong nghiên cứu đó, câu nói của người dùng được ghép với câu trả lời ngay trước đó của hệ thống và

được đưa qua một lớp RNN để học ra một véc-tơ ngữ nghĩa cho cặp câu nói và câu trả lời phía trước này. Các cặp câu nói và câu trả lời phía trước trong đoạn hội thoại sẽ tạo ra một chuỗi các véc-tơ ngữ nghĩa tương ứng. Lớp RNN thứ hai sẽ học một véc-tơ ngữ cảnh ẩn cho cả đoạn hội thoại dựa trên chuỗi véc-tơ ngữ nghĩa này để phân loại ý định cho câu nói của người dùng. Trong một nghiên cứu khác, A. Sharma coi câu nói của người dùng chỉ phụ thuộc vào ý định trước đó của họ [23]. Nghiên cứu này sử dụng mô hình BiLSTM để học được véc-tơ ngữ nghĩa cho câu nói đầu vào và biểu diễn ý định phía trước bằng một véc-tơ one-hot. Hai véc-tơ này được kết hợp với nhau và được đưa qua mô hình GRU hoặc một mô hình mạng nơ-ron kết nối đầy đủ để học được ngữ cảnh ẩn và sử dụng vào nhiệm vụ phân loại ý định. A. Gupta và cộng sự sử dụng nhiều thông tin khác nhau bao gồm những ý định đã phát hiện phía trước đó, các thực thể đã thu thập được, hành động của hệ thống và câu nói tại thời điểm dự đoán trong một cửa sổ hội thoại và sử dụng cơ chế chú ý để đưa ra ngữ cảnh ẩn [24]. Nhìn chung, các tiếp cận này có ưu điểm là hệ thống có thể tự học được thông tin về ngữ cảnh dựa trên dữ liệu đầu vào. Tuy nhiên, chúng ta cần cung cấp một lượng dữ liệu rất lớn, đồng thời không thể trực tiếp kiểm soát hoặc đưa thêm các ngữ cảnh mong muốn mà hoàn toàn phụ thuộc vào kết quả mô hình đưa ra.

Dựa trên khái niệm về ngữ cảnh trong thực tế. Google Dialogflow đưa ra cách quản lý ngữ cảnh để điều khiển luồng hội thoại [25]. Ví dụ nếu một người nói rằng “Chúng có màu xanh”, chúng ta cần ngữ cảnh để xác định “chúng” ở đây là gì. Tương tự như vậy, mỗi ý định cần được cung cấp thông tin về ngữ cảnh để hệ thống có thể dự đoán đúng ý định trong các trường hợp khác nhau. Các ngữ cảnh này còn được gọi là ngữ cảnh đầu vào (input context). Mặt khác, khi một người nói “Tôi muốn xem ảnh của một chú mèo”, rồi sau đó anh ấy/ cô ấy lại hỏi “Nó kêu như thế nào nhỉ?”, bằng cách dựa vào thông tin đã có ở câu trước, chúng ta có thể hiểu ngay “nó” ở đây là “con mèo”. Do đó, các ý định có thể được cấu hình để đưa ra các ngữ cảnh giúp lưu giữ thông tin cho các lượt hội thoại tiếp theo. Các ngữ cảnh này còn được gọi là ngữ cảnh đầu ra (output context). Các xây dựng và quản lý ngữ cảnh của Google Dialogflow khá rõ ràng, tuy nhiên, giải pháp sử dụng các ngữ cảnh này để hiểu ý định người dùng hoàn toàn không được công bố.

Từ những phân tích trên, bên cạnh việc đề xuất tiêu chí đánh giá các thuật toán phân loại dựa trên độ tự tin, trong đồ án này, em tiến hành nghiên cứu xây dựng mô hình phân loại ý định người dùng có sử dụng ngữ cảnh dựa trên cách xây dựng và thiết kế ngữ cảnh của Google DialogFlow để xác định chính xác mong muốn thực sự của người dùng.

1.2 Định hướng giải pháp

Để giải quyết bài toán phân loại ý định người dùng trong hệ thống hội thoại, chúng ta cần biết về độ tự tin của phép đoán nhằm đưa ra cách hành xử phù hợp. Do đó, em sẽ đưa ra tiêu chí đánh giá mô hình phân loại ý định dựa trên độ tự tin và thực nghiệm đánh giá các mô

hình phân loại trên tiêu chí này. Các mô hình mà em sẽ sử dụng để đánh giá gồm các mô hình học máy như Random Forest, SVM và các mạng nơ-ron sâu, cụ thể là CNN, LSTM, LSTM sử dụng cơ chế chú ý, BiLSTM và BiLSTM sử dụng cơ chế chú ý. Kết quả của nghiên cứu đưa ra một giải pháp nhằm chọn ra một mô hình phân loại có chất lượng tốt và phù hợp. Mô hình này sẽ giúp cho hệ thống dễ dàng chọn lựa cách phản hồi hợp lý dựa trên kết quả dự đoán ý định. Mô hình có thể được sử dụng trong các cuộc hội thoại ngắn, các trường hợp hỏi đáp thông thường, trong đó, hệ thống có thể chưa cần sử dụng đến ngữ cảnh mà có thể hỏi lại người dùng để xác nhận thông tin từ phía người dùng.

Với bài toán phân loại ý định người dùng theo ngữ cảnh, dựa trên ý tưởng về quản lý ngữ cảnh của Google Dialogflow, em sẽ đưa ra cách mô hình hóa các ngữ cảnh có trong hệ thống và xây dựng mô hình phân loại dùng các ngữ cảnh này. Mô hình phân loại được đề xuất có thể được sử dụng trong các hội thoại dài và phức tạp để người dùng không phải nhắc lại những thông tin đã trao đổi từ trước.

1.3 Bố cục đồ án

Trên đây, em đã đưa ra những nghiên cứu và vấn đề còn tồn tại đối với bài toán phân loại ý định người dùng trong hệ thống hội thoại và đưa ra định hướng giải pháp cho các vấn đề đó. Phần còn lại của đồ án sẽ được tổ chức như dưới đây.

Chương 2 trình bày các cơ sở lý thuyết có liên quan đến bài toán phân loại ý định người dùng, bao gồm các thuật toán phân loại và các phương pháp trích chọn đặc trưng nhằm chuyển đổi văn bản sang dạng véc-tơ số.

Chương 3 sẽ đưa ra giải thích về tầm quan trọng của độ tự tin trong dự đoán ý định của người dùng. Dựa vào cơ cở lý thuyết đã trình bày ở chương 2, trong chương 3, em cũng tiến hành xây dựng, thử nghiệm và đánh giá các thuật toán phân loại dựa trên độ tự tin. Mô hình phù hợp nhất sẽ được sử dụng trong thử nghiệm tiếp theo

Khi đã lựa chọn được mô hình phù hợp, trong chương 4, em đề xuất về sơ đồ dự đoán hai bước sử dụng ngữ cảnh và cách thử nghiệm, đánh giá về sơ đồ dự đoán này

Cuối cùng, chương 5 kết luận lại nội dung của đồ án, bao gồm những đóng góp của đồ án, những vấn đề còn tồn tại và hướng phát triển trong tương lai

Chương 2 Cơ sở lý thuyết

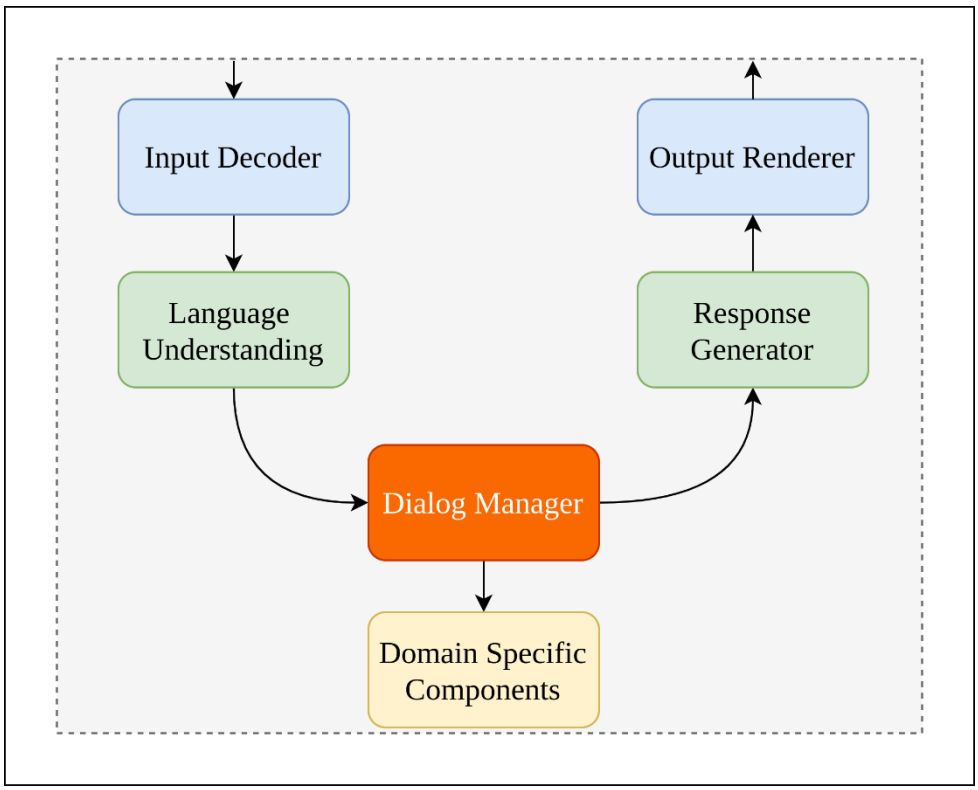
Chương 2 trình bày cơ sở lý thuyết về hệ thống hội thoại, các thuật toán phân loại ý định và các cách trích chọn đặc trưng để chuyển một văn bản thành một véc-tơ số. Đây là tiền để xây dựng các mô hình phân loại trong chương 3 và chương 4.

2.1 Hệ thống hội thoại

Hệ thống hội thoại là một chương trình máy tính có khả năng giao tiếp với người dùng bằng ngôn ngữ tự nhiên [3]. Các hệ thống này có thể nhận đầu vào là văn bản, cử chỉ hoặc giọng nói. Khi nhắc đến hệ thống hội thoại, đa phần chúng ta muốn nói tới đến hệ thống hội thoại hướng nhiệm vụ. Hệ thống hội thoại hướng nhiệm vụ tập trung vào việc giải quyết một số nhiệm vụ cụ thể trong một hoặc nhiều lĩnh vực [26]. Có hai phương pháp chính để xây dựng một hệ thống hội thoại là phương pháp trọn gói (end-to-end) và phương pháp đường ống.

Phương pháp trọn gói (end-to-end) sử dụng một mô hình duy nhất, lấy ngữ cảnh tự nhiên làm đầu vào và phản hồi bằng ngôn ngữ tự nhiên [26]. Vì chỉ cần một mô hình duy nhất nên phương pháp này yêu cầu ít công sức hơn trong việc xây dựng và triển khai mô hình. Tuy nhiên, cả hệ thống lúc này như một hộp đen, người phát triển hệ thống rất khó điều khiển và kiểm soát kết quả đầu ra của hệ thống.

Trong khi đó, phương pháp đường ống chia hệ thống thành các mô-đun nhỏ hơn, mỗi mô-đun có một nhiệm vụ khác nhau, giúp cho hệ thống dễ hiểu và ổn định hơn so với phương pháp end-to-end nên hầu hết các hệ thống thương mại trên thế giới đều sử dụng cách này [26]. Các hệ thống khác nhau có thể có thiết kế khác nhau tùy vào mục đích sử dụng, nhưng nhìn chung, một hệ thống hội thoại thường có sáu thành phần chính, bao gồm chuyển hóa thông tin đầu vào (Input Decoder), hiểu ngôn ngữ tự nhiên (Natural Language Understanding – NLU), quản lý hội thoại (Dialog Manager), thành phần của từng lĩnh vực cụ thể (Domain Specific Components), sinh thông tin phản hồi (Response Generator) và chuyển hóa thông tin đầu ra (Output Renderer) [3]. Vị trí và thứ tự xử lý của các thành phần được thể hiện trong Hình 1.



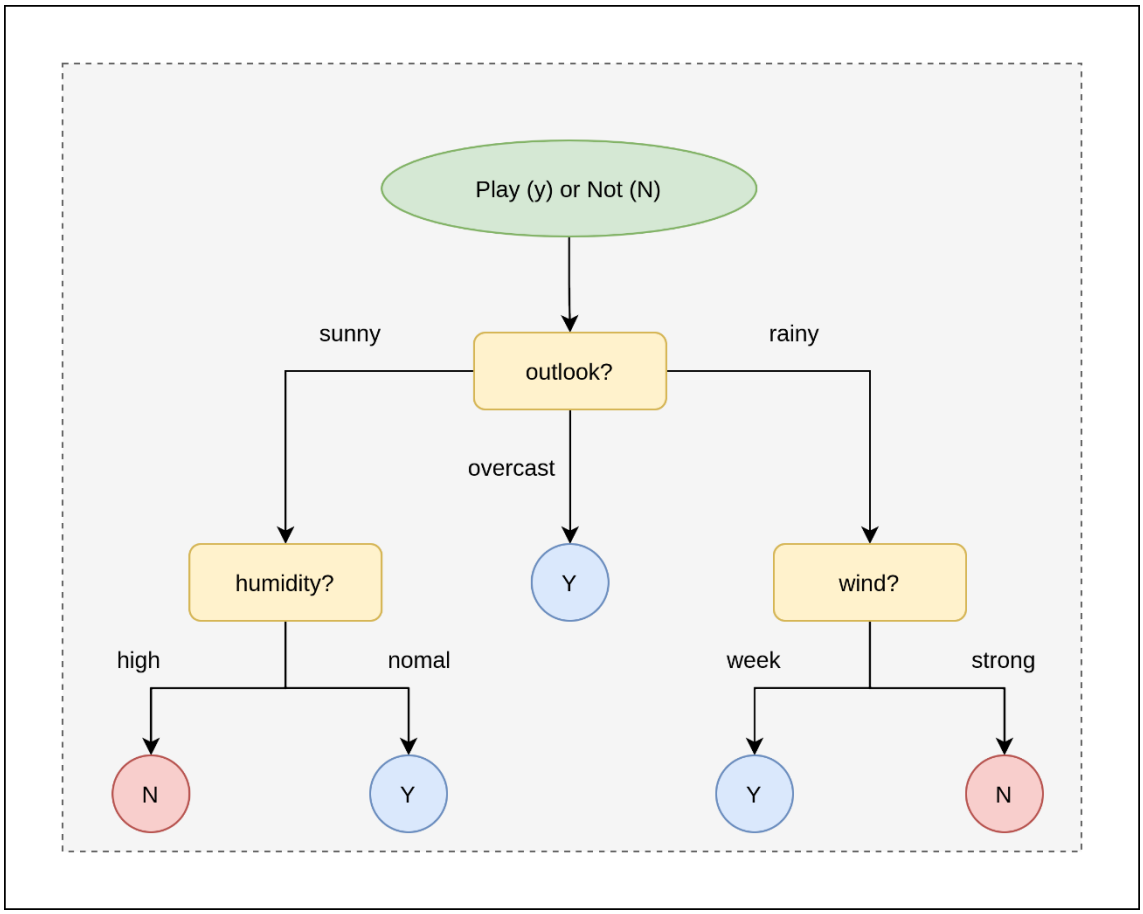
Hình 1 Kiến trúc hệ thống hội thoại.

Thành phần chuyển hóa thông tin đầu vào (Input Decoder) có vai trò chuyển hóa thông tin nhận được từ người dùng thành văn bản. Trong trường hợp đầu vào là giọng nói, Input Decoder cần chuyển đổi âm thanh nhận được thành dạng văn bản (một chuỗi các kí tự). Các hệ thống nhận dạng giọng nói như thế còn được gọi là Automatic Speech Recognition (ASR) hay Speech to Text (STT). Văn bản nhận được sẽ chuyển tiếp đến thành phần hiểu ngôn ngữ tự nhiên (Natural Language Understanding – NLU). Tại đây, chúng được phân tích thành biểu diễn có ngữ nghĩa mà hệ thống có thể hiểu được, sau đó được chuyển đến thành phần quản lý hội thoại (Dialog Manager). Dialog Manager sẽ tìm cách phù hợp biểu diễn đó với ngữ cảnh tổng thể và quyết định phản hồi cho đầu vào. Việc ra quyết định nhiều khi cần truy xuất thông tin bên ngoài hệ thống, lúc này Dialog Manager cần gọi đến thành phần của từng lĩnh vực cụ thể (Domain Specific Components). Domain Specific Components cho phép hệ thống có thể truy cập vào cơ sở dữ liệu hoặc các hệ thống chuyên gia. Sau khi có đầy đủ thông tin cần thiết, thành phần sinh thông tin phản hồi (Response Generator) sẽ dựa vào đó để xây dựng thông điệp cho người dùng. Cuối cùng, thành phần chuyển hóa thông tin đầu ra (Output Renderer) sẽ chuyển đổi thông điệp phản hồi thành dạng phù hợp. Đối với các hệ thống hội thoại dựa trên giọng nói, thông điệp đầu ra có thể là một câu nói được thu âm từ trước hoặc được tổng hợp bằng cách sử dụng kĩ thuật chuyển đổi văn bản thành giọng nói (Text to Speech – TTS) [3].

2.2 Các thuật toán phân loại

2.2.1 Thuật toán Rừng ngẫu nhiên

Giống như tên gọi của nó, Rừng ngẫu nhiên (Random Forest) [14] là một tập hợp của các cây quyết định (Decision Tree) [27], với số lượng cây có thể lên đến hàng trăm hoặc hàng nghìn cây. Một cây quyết định là cây phân cấp có cấu trúc được phân lớp dựa vào dãy các luật. Hình 2 mô tả một cây quyết định được xây dựng trong trường hợp một bạn nam muốn đi đá bóng hoặc ở nhà dựa theo thời tiết, độ ẩm và gió.



Hình 2 Ví dụ về một cây quyết định.

Qua sơ đồ trong Hình 2, ta có thể thấy rằng nếu trời nắng, độ ẩm bình thường thì khả năng cao bạn nam sẽ đi chơi bóng. Ngược lại, bạn ấy sẽ ở nhà nếu như trời mưa và có gió mạnh.

Để xây dựng một cây quyết định cho một bài toán phân loại, chúng ta sử dụng thuật toán Iterative Dichotomiser 3 (ID3) [27]. Một dữ liệu thường có nhiều thuộc tính, ở ví dụ trên thì một dữ liệu sẽ bao gồm các thông tin về thời tiết (nắng/mưa), độ ẩm (cao/ bình thường), sức gió (mạnh/ nhẹ). Ý tưởng của ID3 là xác định thứ tự các thuộc tính tại mỗi bước. Tại một bước cụ thể, thuộc tính tốt nhất được lựa chọn thông qua một phép đo. Một phép phân chia

được coi là tốt nếu như dữ liệu tại bước đó nằm hoàn toàn trong một lớp. Ngược lại, nếu dữ liệu vẫn bị lẫn vào nhau theo một tỷ lệ lớn thì phép chia chưa thực sự tốt.

Mỗi cây quyết định trong rừng sẽ được sinh ra một cách ngẫu nhiên, phụ thuộc vào các phán đoán trong quá trình học. Kết quả cuối cùng sẽ được tính bằng trung bình các phán đoán của các cây trong rừng.

2.2.2 Thuật toán Véc-tơ máy hỗ trợ

Véc-tơ máy hỗ trợ (Support Vector Machine – SVM) [1] là phương pháp phân loại tuyến tính, được dùng để phân chia dữ liệu thành các lớp riêng biệt. Với một bài toán phân loại nhị phân, giả sử tập dữ liệu ban đầu có N điểm:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

Trong đó x_i là một vector đầu vào được biểu diễn trong không gian $X \subseteq R^d$ còn y_i là nhãn tương ứng với vector đầu vào, $y_i \in \{1; -1\}$, với $y_i = 1$ tức là dữ liệu thuộc về lớp dương (lớp *positive*) và $y_i = -1$ tức là dữ liệu thuộc về lớp âm (lớp *negative*).

Mục tiêu của SVM là xác định một hàm phân tách tuyến tính giữa hai lớp dữ liệu:

$$f(x) = \langle w.x \rangle + b$$

Công thức 1 Hàm phân tích tuyến tính giữa hai lớp dữ liệu.

Với w là vector trọng số các thuộc tính, còn b là một giá trị số thực. Dựa vào hàm $f(x)$**,** ta xác định giá trị đầu ra của mô hình như sau:

$$y_i = \{1 \; nếu \; \langle w.x_i \rangle + b \geq 0 \;\; -1 \; nếu \; \langle w.x_i \rangle + b < 0$$

Công thức 2 Công thức tính nhãn cho một điểm dữ liệu.

Giả sử $(x^+, 1)$ là điểm thuộc lớp dương và $(x^-, -1)$ là điểm thuộc lớp âm, gần nhất với siêu phẳng phân tách H_0. Gọi H_+, H_- là hai siêu phẳng lề song song với nhau, trong đó H_+ đi qua $(x^+, 1)$ và song song với H_0 và H_- đi qua $(x^-, -1)$ và song song với H_0. Mức lề (margin) là khoảng cách giữa hai siêu phẳng H_+ và H_-. Để tối thiểu hóa giới hạn lỗi mắc phải trong quá trình phân tách, chúng ta cần chọn siêu phẳng có lề lớn nhất, siêu phẳng như vậy được gọi là siêu phẳng có lề cực đại.

Khoảng cách từ x^+ đến H_0 là:

$$d_+ = \frac{|\langle w.x^+ \rangle + b|}{\|w\|} = \frac{|1|}{\|w\|} = \frac{1}{\|w\|}$$

Công thức 3 Công thức tính khoảng cách từ điểm gần nhất thuộc lớp dương đến siêu phẳng phân cách.

Khoảng cách từ x^- đến H_0 là:

$$d_- = \frac{|\langle w.x^- \rangle + b|}{\|w\|} = \frac{|-1|}{\|w\|} = \frac{1}{\|w\|}$$

Công thức 4 Công thức tính khoảng cách từ điểm gần nhất thuộc lớp âm đến siêu phẳng phân cách.

Mức lề giữa hai siêu phẳng H_+ và H_- được xác định theo công thức:

$$margin = d_+ + d_- = \frac{2}{\|w\|}$$

Công thức 5 Công thức tính mức lề.

Do đó, bài toán xác định mức lề lớn nhất giữa hai siêu phẳng được đưa về việc xác định w và b sao cho $margin = \frac{2}{\|w\|}$ đạt giá trị lớn nhất và thỏa mãn điều kiện:

$$\{\langle w.x_i \rangle + b \geq 1 \; nếu \; y_i = 1 \; \langle w.x_i \rangle + b < 0 \; nếu \; y_i = -1$$

(do x^+, x^- là các điểm gần nhất với mặt phẳng phân tách và lần lượt thuộc

H_+: $\langle w.x \rangle + b = 1$, H_-: $\langle w.x \rangle + b$ = -1):

$(w, b) = \frac{2}{\|w\|}$, điều kiện: $\{ \langle w.x_i \rangle + b \geq 1 \; nếu \; y_i = 1 \; \langle w.x_i \rangle + b \leq -1 \; nếu \; y_i = -1$

Công thức 6 Công thức tìm hệ số trong thuật toán SVM. (1)

Tương đương với:

$(w, b) = \frac{\|w\|^2}{2} = \frac{\langle w.w \rangle}{2}$, điều kiện: $1 - y_i(\langle w.x_i \rangle + b) \leq 0$

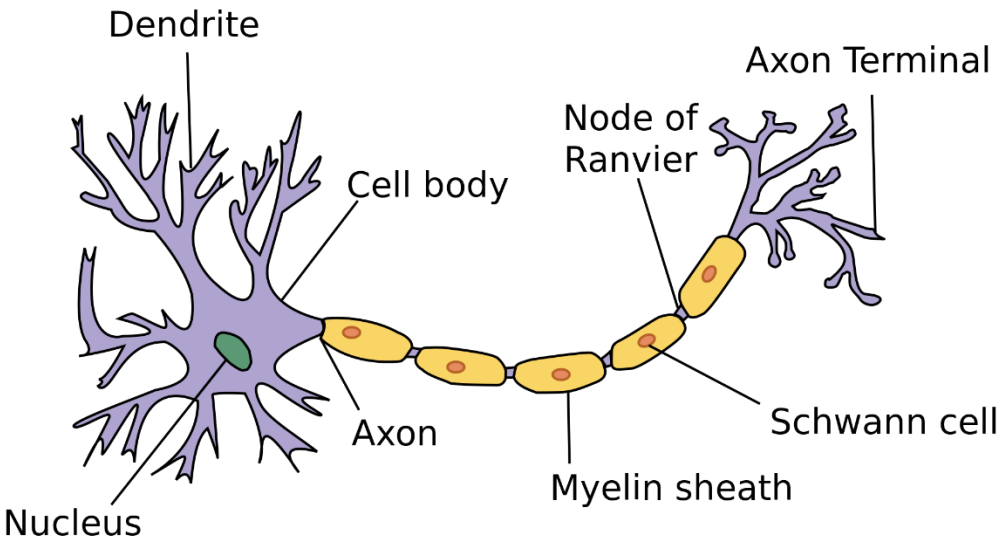
Công thức 7 Công thức tìm hệ số trong thuật toán SVM. (2)

Thuật toán phía trên được áp dụng trong trường hợp dữ liệu tuyến tính. Đối với dũ liệu không tuyến tính, SVM sử dụng các hàm nhân (kernel function) để biến đổi dữ liệu sang không gian mới, trong không gian đó, dữ liệu thu được là phân biệt tuyến tính. Một số hàm nhân thông dụng có thể kể đến như hàm tuyến tính, polynomial, Radial Basis Function (RBF), hàm sigmoid, [28]…

Với bài toán phân loại đa lớp sử dụng SVM, có nhiều cách để chúng ta đưa về bài toán phân loại nhị phân, trong đó, phương pháp thường được sử dụng nhiều nhất là one-vs-rest (hay còn được gọi là one-vs-all, one-against-all) [28]. Cụ thể, nếu có C lớp thì ta sẽ xây dựng C mô hình, trong đó, mỗi mô hình tương ứng với một lớp. Mỗi mô hình này sẽ giúp phân biệt một điểm dữ liệu có thuộc vào lớp đó hay không hoặc tính xác suất để một điểm rơi vào lớp đó là bao nhiêu. Kết quả cuối cùng có thể là lớp có xác xuất cao nhất.

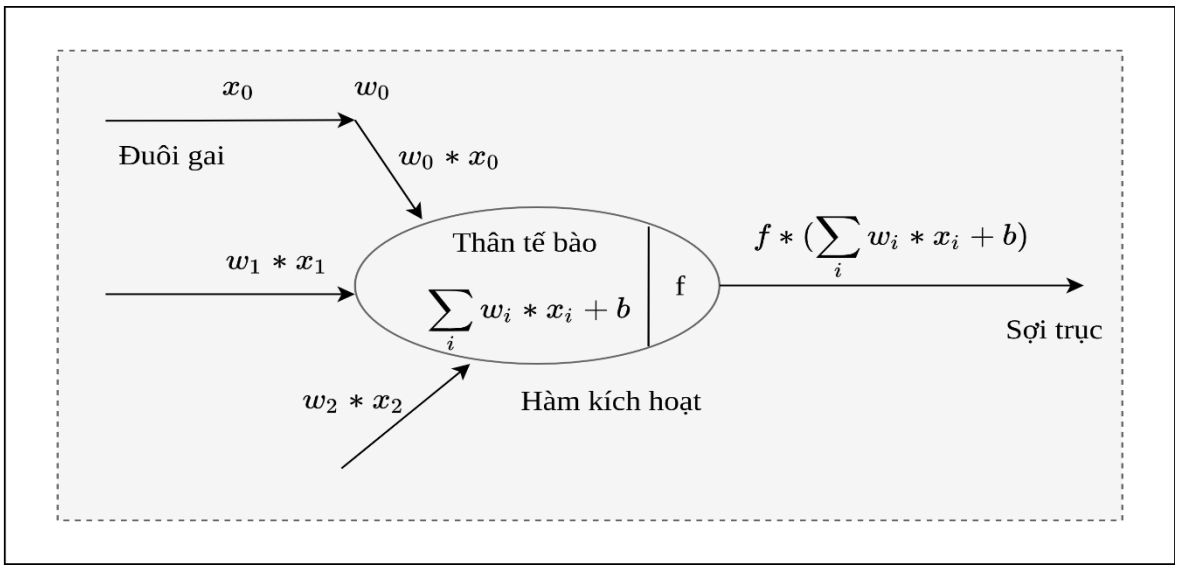
2.2.3 Mạng nơ-ron

Mạng nơ-ron được hình thành từ các nơ-ron đơn lẻ hay còn gọi là perceptron, các nơ-ron này được lấy ý tưởng từ tế bào nơ-ron sinh học của con người. **Hình 3** mô tả cấu trúc của một nơ-ron sinh học.



Hình 3 Nơ-ron sinh học.

Mỗi nơ-ron sinh học bao gồm ba thành phần chính: (i) thân tế bào (cell body) là chỗ phình to của nơ-ron, chứa nhân tế bào, có vai trò cung cấp dinh dưỡng cho nơ-ron, có thể phát sinh xung thần kinh và có thể tiếp nhận xung thần kinh truyền tới nơ-ron, (ii) đuôi gai (dendrite) là các tua ngắn phát triển từ thân tế bào, có chức năng tiếp nhận xung thần kinh từ các nơ-ron khác, truyền tới thân tế bào và (iii) sợi trục (axon) là sợi thần kinh đơn dài, làm nhiệm vụ truyền tính hiệu từ thân tế bào tới các nơ-ron khác. Lấy ý tưởng từ đây, nơ-ron nhân tạo được thiết kế với cấu trúc như mô tả trong Hình 4.

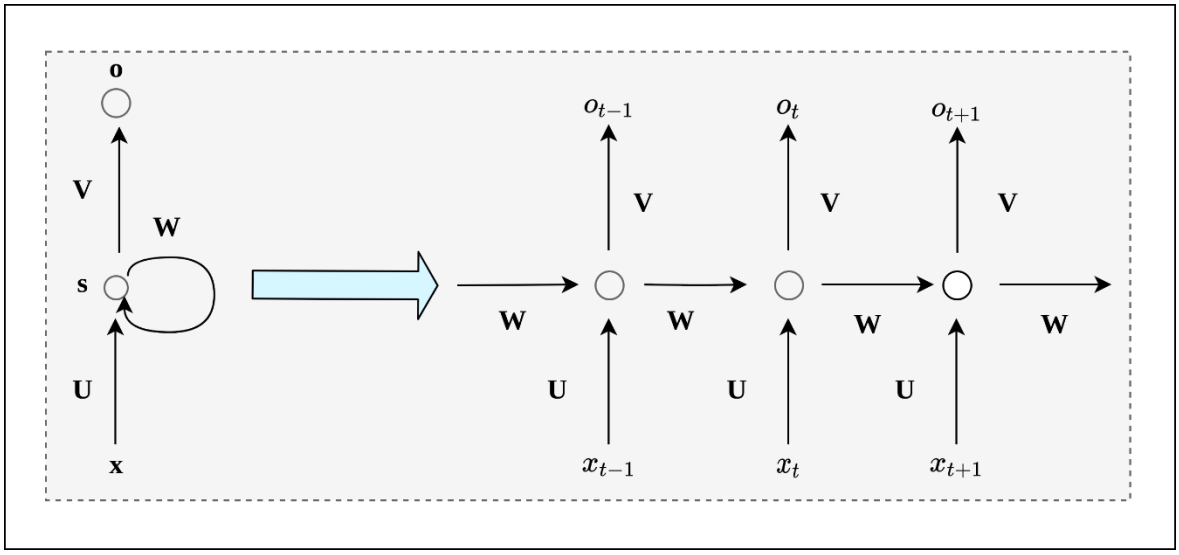


Hình 4 Nơ-ron nhân tạo.

Mỗi nơ-ron gồm các đầu vào tương ứng với các đuôi gai, một bộ xử lý sử dụng các hàm kích hoạt tương ứng với các thân tế bào và đầu ra của nơ-ron thì tương ứng với sợi trục. Hàm kích hoạt thường là hàm phi tuyến như hàm sigmoid, hàm tanh, hàm ReLU, hàm sign, [29]…Nhiều nơ-ron kết hợp với nhau sẽ trở thành mạng nơ-ron. Mạng nơ-ron thường có 3 tầng: tầng đầu vào nhận dữ liệu đầu vào từ tập dữ liệu, tầng đầu ra cho biết giá trị dự đoán của mô hình đối với dữ liệu đầu vào còn tầng ẩn là các tầng nằm giữa tầng đầu vào và tầng đầu ra.

2.2.4 Mạng nơ-ron hồi quy (Recurrent Neural Network)

Với mạng nơ-ron thông thường thì tất cả đầu vào độc lập với nhau, nên giữa chúng không có sự liên kết chuỗi. Trong bài toán xử lý văn bản, thì thứ tự trước sau trong một văn bản là rất quan trọng. Dựa vào điều này, mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) xác định giá trị của phần tử tiếp theo dựa trên các phép tính trước đó. Hình 5 mô tả cấu trúc mạng RNN.



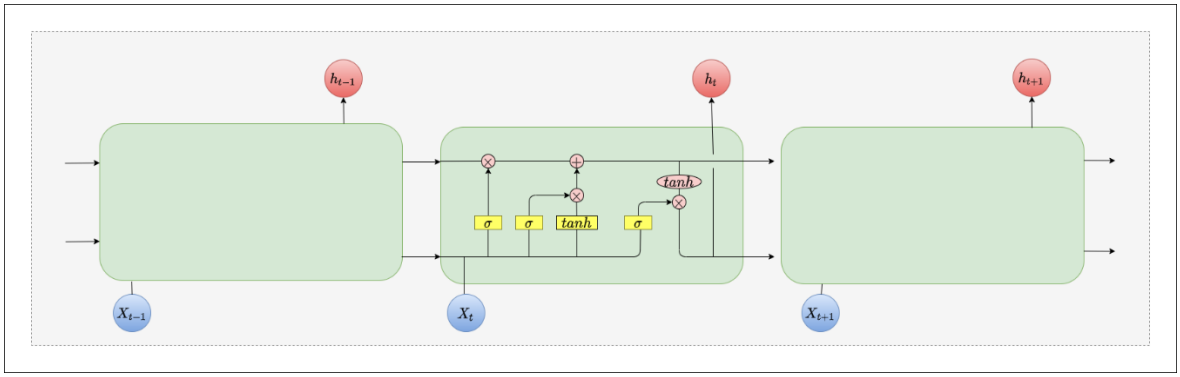
Hình 5 Cấu trúc mạng RNN.

Việc tính toán bên trong mạng tại mỗi bước được thực hiện dựa vào đầu vào tại mỗi bước, trạng thái ẩn, được tính dựa trên các trạng thái ẩn và đầu vào tại các bước trước đó thông qua một số hàm như tanh hoặc ReLU,… và đầu ra tại bước này, thường sử dụng hàm softmax.

2.2.5 Long-Short Term Memory

Lý thuyết đã chứng minh rằng đối với các bước ở xa thì mạng RNN gặp phải vấn đề phụ thuộc xa, tức là mạng chỉ nhớ được trong một khoảng nhỏ. Điều này xảy ra do hiện tượng vanishing gradient. Đây là hiện tượng xảy ra khi giá trị gradients sẽ có giá trị nhỏ dần khi đi xuống các lớp thấp hơn, dẫn đến việc cập nhật thực hiện bởi Gradient Descent không làm

thay đổi nhiều trọng số của các lớp đó, khiến chúng không thể hội tụ và RNN sẽ không thu được kết quả tốt. Mạng Long-Short Term Memory (LSTM) [30] ra đời nhằm khắc phục hạn chế này. LSTM cũng có kiến trúc dạng chuỗi tương tự RNN nhưng thay vì chỉ có một tầng mạng nơ-ron, chúng có tới bốn tầng tương tác với nhau một cách rất đặc biệt:



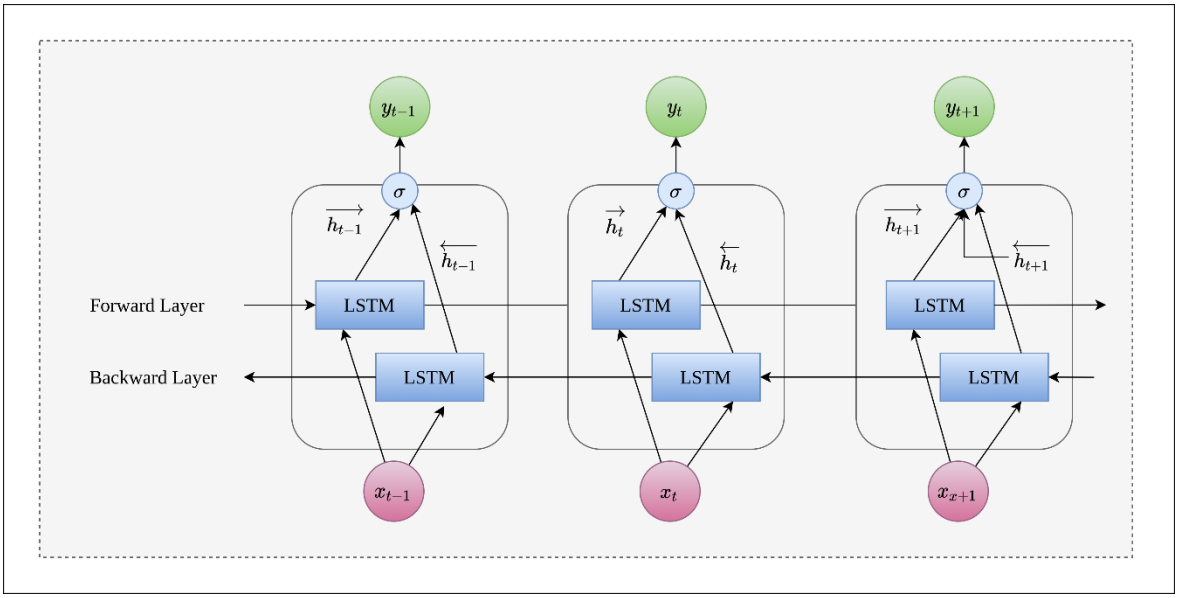
Hình 6 Cấu trúc mạng LSTM.

Điểm đặc biệt của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ. Đây được coi là bộ nhớ của mạng. Tại mỗi tế bào, LSTM có thể thêm vào hoặc bỏ đi các thông tin cần thiết thông quan các cổng: cổng quên, cổng vào và cổng ra lần lượt được thể hiện trong hình vẽ. Cổng quên sẽ quyết định xem những thông tin nào là không cần thiết và cần bỏ đi tại trạng thái này. Cổng vào cho biết những thông tin nào sẽ được thêm vào trạng thái tế bào. Cổng ra quyết định kết quả đầu của tế bào này. Với cấu tạo như vậy, mạng LSTM có khả năng nhớ các state ở xa hơn, từ đó tạo hiệu quả tốt hơn so với mạng RNN.

2.2.6 Bidirectional Long-Short Term Memory (BiLSTM)

Tuy nhiên, một mạng LSTM truyền thống chỉ có thể học được biểu diễn của trạng thái hiện tại dựa trên các trạng thái trước đó. Tuy nhiên, các từ trong câu mà chúng ta nói đến đến sau này có thể mang thông tin bổ sung cho từ hiện tại. Do đó, thông tin của các từ phía sau, hay còn gọi là thông tin từ tương lai, cũng là một ngữ cảnh quan trọng để xác định ý nghĩa của một từ. BiLSTM [19] đã được tạo ra để khắc phục điểm yếu này.

BiLSTM có sự kết hợp của hai mạng LSTM theo hai chiều khác nhau: một mạng theo chiều từ trái sang phải (forward LSTM) và mạng kia theo chiều ngược lại (backwark LSTM). Chiều từ trái sang phải giúp cho mô hình có thể học được thông tin ngữ cảnh từ quá khứ. Ngược lại, chiều từ phải sang trái khai thác thông tin đến từ tương lai. Việc khai thác thông tin từ cả hai phía như vậy giúp cho mô hình có thể hiểu ngữ cảnh của một văn bản tốt hơn, giúp tăng khả năng dự đoán của mô hình. Hình 7 mô tả kiến trúc của một mạng BiLSTM.



Hình 7 Cấu trúc mạng BiLSTM.

Trong Hình 7, x_t là biểu diễn véc-tơ của từ thứ t trong văn bản, $\overrightarrow{h_t}$ và $\overleftarrow{h_t}$ lần lượt là trạng thái ẩn trong trạng thái tế bào theo chiều tiến và lùi, y_t là kết quả đầu ra tại bước thứ t và σ là hàm sigmoid. Các trạng thái ẩn $\overrightarrow{h_t}$ và $\overleftarrow{h_t}$ thường được nối với nhau để tạo thành một trạng thái ẩn duy nhất. Nhờ kiến trúc theo hai chiều, BiLSTM có khả năng học được thông tin đầy đủ hơn so với LSTM.

2.2.7 Cơ chế chú ý (Attention)

Cơ chế chú ý (Attention) [31] cho phép mô hình tập trung vào một phần thông tin quan trọng thay vì phải tập trung vào toàn bộ nội dung câu. Có nhiều loại cơ chế chú ý khác nhau, tuy nhiên, tổng quan chúng ta có hai loại: hard-attention và soft-attention. Với hard-attention, mô hình sẽ chọn ngẫu nhiên một vùng thông tin để chú ý nên có ưu điểm là giảm được tài nguyên máy tính cần để xử lý. Soft-attention sẽ học trọng số để chú ý trên tất cả các phần thông tin của câu giúp tổng hợp thông tin cần thiết để đưa ra dự đoán. Tổng hợp thông tin này được tính bằng trung bình cộng có trọng số của tất cả các phần thông tin. Vì dễ tối ưu và không phức tạp trong lúc cài đặt nên soft-attention, được cộng đồng tập trung phát triển rất nhiều và có khá nhiều phiên bản cải tiến khác nhau. Một số phiên bản khác nhau của soft-attention bao gồm:

(i) Learn-to-align trong bài toán dịch máy của Bahdanau [32] có thể xem như là phiên bản đầu tiên được mọi người chú ý, sử dụng soft-attention để học cách tổng hợp thông tin từ câu được dịch để phát sinh câu đích.

(ii) Global và Local attention: global-attention tương tự như mô hình do Bahdanau đề xuất, còn local-attention lấy ý tưởng từ hard attention, tức là học thêm vị trí cần được chú ý.

(iii) Self-attention cho phép học mối tương quan giữ thông tin của từ hiện tại với những từ trước đó. Đây là một cơ chế chú ý liên quan đến các vị trí khác nhau của một chuỗi đơn lẻ để tính toán biểu diễn của cùng một chuỗi. Nó đã được chứng minh là rất hữu ích trong việc đọc máy, tóm tắt trừu tượng hoặc tạo mô tả hình ảnh.

2.3 Các phép trích chọn đặc trưng

Trích chọn đặc trưng là việc lựa chọn ra các thuộc tính của văn bản và véc-tơ hóa chúng sang một không gian véc-tơ để máy tính có thể dễ dàng xử lý. Sau đây, em xin trình bày các phép trích chọn đặc trưng.

2.3.1 Term Frequency – Inverse Document Frequency (TF-IDF)

Giá trị $TF - IDF$ [33] của một từ thể hiện mức độ quan trọng của một từ trong một văn bản. Trong đó, TF ($Term\ Frequency$) là tần số xuất hiện của một từ trong một văn bản và được tính theo công thức:

$$TF(t,d) = \frac{f(t,d)}{\sum_{t' \in d} \quad f(t',d)}$$

Công thức 8 Công thức tính tần số xuất hiện của từ.

Với $f(t,d)$ là số lần xuất hiện của từ t trong văn bản d. Mẫu số trong công thức trên chính là tổng số từ có trong văn bản.

IDF (Inverse Document Frequency) là tần số nghịch của một từ trong tập văn bản. Mục đích của IDF là giảm các từ thường xuyên xuất hiện trong văn bản nhưng lại không mang nhiều ý nghĩa. Công thức tính IDF như sau:

$$IDF(t,D) = log\ log\ \frac{|D|}{|\{d \in D : t \in d\}|}$$

Công thức 9 Công thức tính tần số nghịch của một từ trong tập văn bản.

Trong đó: $|D|$ là tổng số văn bản có trong tập D, còn mẫu số là tổng số văn bản trong tâp D có chứa từ t. Giá trị $IF - IDF$ được tính như sau:

$$TF - IDF(t,d,D) = IF(t,d).IDF(t,D)$$

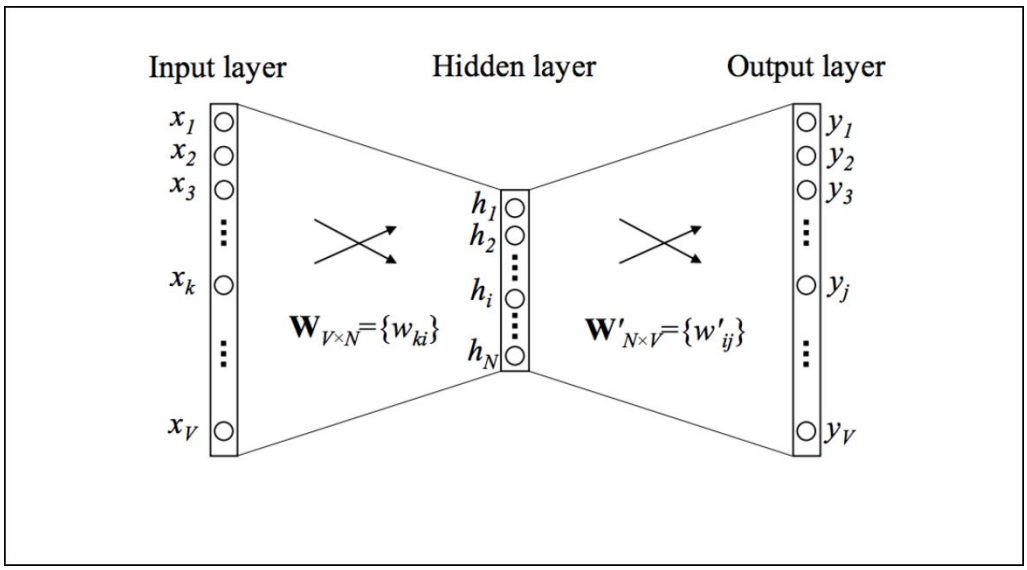
Công thức 10 Công thức tính $TF - IDF$.

Những từ có giá trị $TF - IDF$ cao là những từ xuất hiện nhiều trong văn bản này và xuất hiện ít trong văn bản khác. Giá trị này giúp chúng ta lọc ra được những từ phổ biến và giữ lại các từ có giá trị cao (các từ khóa của văn bản). $TF - IDF$ là một cách đơn giản để véc-tơ hóa dữ liệu dạng văn bản, tuy nhiên độ lớn của véc-tơ bằng số lượng từ vựng, làm gia tăng khối lượng tính toán. Với các từ nằm ngoài tập từ điển. Hơn nữa, cách biểu diễn từ bằng $TF - IDF$ gặp phải vấn đề không biểu diễn được các từ nằm ngoài từ điển và không thể hiện được mối quan hệ giữa các từ.

2.3.2 Word2Vec

Word2Vec là một phương pháp ánh xạ các từ vào trong không gian véc-tơ có số chiều nhỏ hơn số chiều của từ điển mà vẫn giữ được mối quan hệ ngữ nghĩa của các từ. Nó có thể được xây dựng bằng hai phương pháp: Skip Gram và Common Bag of Word (CBOW) [34].

Mô hình CBOW sử dụng ngữ cảnh xung quanh mỗi từ làm đầu vào và cố gắng dự đoán từ tương ứng với ngữ cảnh đó. Kiến trúc của mô hình được thể hiện trong Hình 8.



Hình 8 Cấu trúc mô hình CBOW một từ đầu vào.

Đầu vào hay từ ngữ cảnh là một véc-tơ $X = [x_1\ x_2\ ...\ x_V]$ được mã hóa dưới dạng one-hot với kích thước V, là kích thước của từ điển, trong đó $x_i = 1$ với i là vị trí của từ trong từ điển, ngược lại $x_i = 0$. Tầng ẩn bao gồm N nơ-ron và tầng đầu ra cũng là một véc-tơ one-hot $Y = [y_1\ y_2\ ...\ y_V]$ có kích thước V. $W_{V\times N}$ là ma trận trọng số giữa tầng đầu vào và tầng ẩn. $W'_{N\times V}$ là ma trận trọng số giữa tầng ẩn và tầng đầu ra. Các nơ-ron của lớp ẩn chỉ sao chép tổng trọng số của các đầu vào sang lớp tiếp theo. Không có kích hoạt như sigmoid, tanh hoặc ReLU. Sự phi tuyến tính duy nhất là các tính toán softmax trong lớp đầu ra. Trong quá trình dự đoán từ đích, chúng ta học cách biểu diễn vectơ của từ đích. SkipGram hoạt động tương tự như mô hình CBOW, mô hình SkipGram nhận đầu vào là một từ và cố gắng dự đoán ngữ cảnh xung quanh nó.

So với IF-IDF, các mô hình Word2Vec đã biểu diễn được mối quan hệ ngữ nghĩa giữa các từ. Khoảng cách giữa các cặp từ có ngữ nghĩa gần giống nhau sẽ xấp xỉ nhau, ví dụ như "king" – "queen" và "man" – "woman". Tuy nhiên, Word2Vec vẫn chưa giải quyết được vấn đề biểu diễn các từ không có trong từ điển.

2.3.3 FastText

Tương tự như các mô hình Word2Vec, FastText [35] cũng sử dụng mô hình CBOW và SkipGram để huấn luyện mô hình. Tuy nhiên, thay vì tách các câu thành một tập hợp các từ đơn dẻ, FastText phân chia các câu theo n-grams mức kí tự [38]. Cụ thể với từ "Chào", sử dụng 3-grams, FastText biến đổi đưa từ này tương đương với "<ch", "hào",

“ào>”, “<chào>”. Kí hiệu “<” và “>” cho biết bắt đầu một từ. Từ “<chào>” được thêm vào để phân biệt đây là từ gốc.

Sau khi phân tách được các n-grams, các từ này được cho qua mô hình CBOW hoặc Skip Gram để học được biểu diễn của các n-grams. Biểu diễn của một từ sau đó được tính toán bằng trung bình biểu diễn của các n-grams [38]. Do đó, bên cạnh biểu diễn được ngữ nghĩa và mối tương quan giữa các từ trong từ điển, FastText còn có thể biểu diễn các từ không có trong từ điển thông qua các n-grams của nó.

Như vậy, qua chương 2, chúng ta đã hiểu về các thành phần trong hệ thống hội thoại. Đồng thời, chương 2 cũng đưa ra những thuật toán và các phép trích chọn đặc trưng thường được sử dụng trong một bài toán phân loại. Tiếp theo, trong chương 3, em sẽ trình bày về mô hình phân loại ý định người dùng trong hệ thống hội thoại.

Chương 3 Phân loại ý định người dùng trong hội thoại

Việc dự đoán ý định sai sẽ làm cho hệ thống đưa ra hành động không chính xác, ảnh hưởng đến trải nghiệm người dùng. Do đó, chương này đưa ra mô hình phân loại ý định người dùng sử dụng độ tự tin nhằm giảm thiểu rủi ro trong của việc dự đoán sai, đồng thời đánh giá các thuật toán phân loại khác nhau để lựa chọn thuật toán phù hợp nhất cho mô hình đề xuất.

3.1 Đề xuất mô hình phân loại ý định trong hệ thống hội thoại

Như đã trình bày trong phần 2.1, mô-đun NLU sẽ đưa ra các phán đoán về ý định và thực thể xuất hiện trong câu nói của người dùng và gửi các thông tin này đến mô-đun quản lý hội thoại nhằm xác định hành động phù hợp. Thông thường, một mô hình phân loại sẽ trả về một nhãn lớp tương ứng với dữ liệu đầu vào kèm theo một chỉ số là độ tự tin thể hiện mức độ tin cậy của mô hình với phán đoán của nó. Tuy nhiên, trong giao tiếp, không phải lúc nào người dùng cũng trò chuyện một cách rõ ràng và thậm chí có thể nói về những chủ đề mà không nằm trong hiểu biết của hệ thống. Nếu như hệ thống luôn lấy kết quả của mô hình trả về để đưa ra hành động tiếp theo thì có thể dẫn đến phản hồi không chính xác. Bảng 1 đưa ra các ví dụ về các trường hợp có thể xảy ra trong một hệ thống hội thoại về lĩnh vực ngân hàng.

Bảng 1 Ví dụ về câu nói của người dùng

	Câu nói của người dùng	**Ý định có thể khớp với câu nói đầu vào**
Đầy đủ thông tin	Tôi muốn hỏi thủ tục làm thẻ tín dụng	Hồ sơ mở thẻ
Thiếu thông tin	Hồ sơ cần những gì em ơi	hồ sơ mở thẻ, hồ sơ vay vốn hồ sơ đăng kí ngân hàng điện tử
Trao đổi ngoài hệ thống	Hủy vé máy bay thì làm như thế nào em nhỉ?	Không khớp với ý định nào

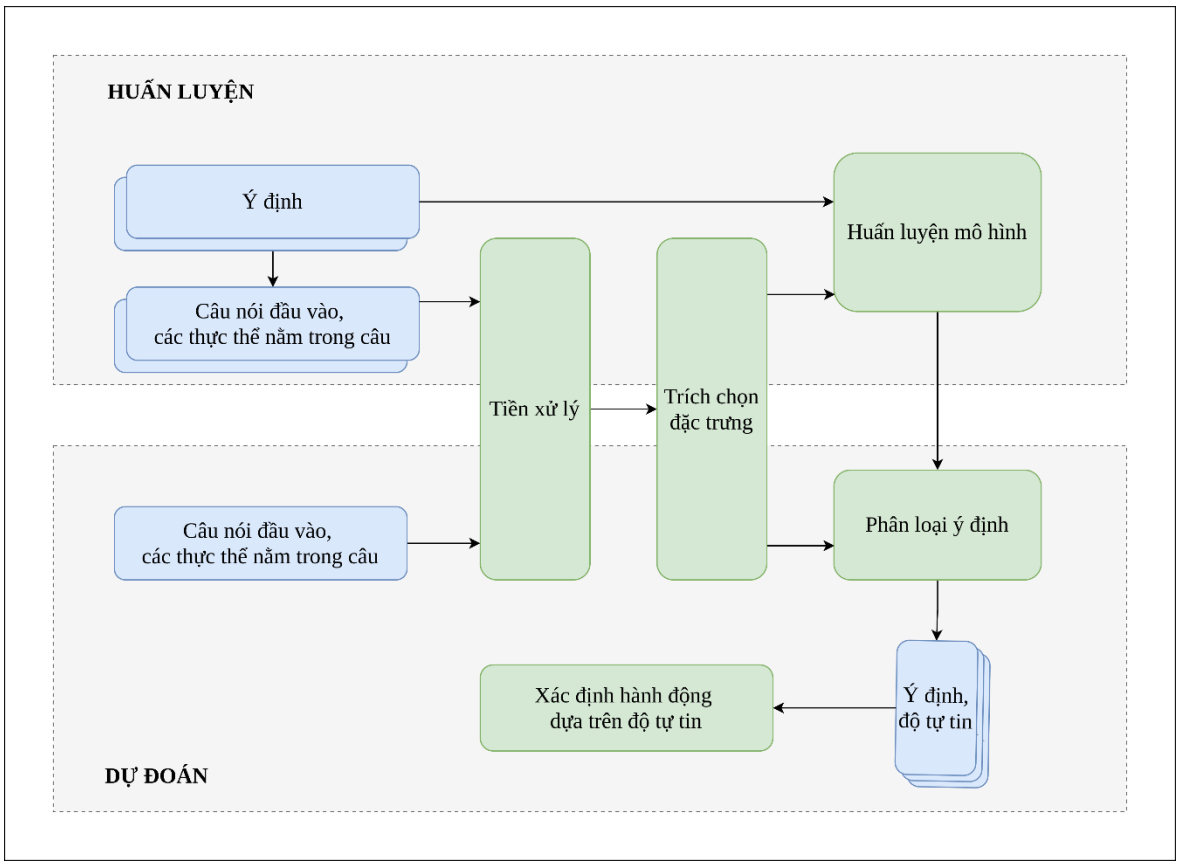
Trong trường hợp người dùng cung cấp đầy đủ thông tin như "Tôi muốn hỏi thủ tục làm thẻ", hệ thống có thể dễ dàng xác định được mong muốn của người dùng là mở thẻ với mức tin tưởng cao và hệ thống có thể đưa ra hành động tương ứng là hướng dẫn họ cách mở thẻ tín dụng như thế nào.

Tuy nhiên, với trường hợp thiếu thông tin, câu nói "Hồ sơ cần những gì em ơi" có thể thuộc vào rất nhiều ý định khác nhau. Giả sử như ý định thực sự của người dùng là hỏi về hồ sơ mở thẻ, nhưng mô hình phân loại dự đoán vào ý định "Hỏi hồ sơ vay vốn". Rõ ràng, phán đoán của mô hình trong trường hợp này sẽ có độ tự tin không cao vì câu nói này có thể thể hiện nhiều ý định khác nhau. Nếu hệ thống tin tưởng hoàn toàn vào kết quả phán đoán của mô hình và hướng dẫn họ cách để vay vốn tại ngân hàng thì hệ thống sẽ không đáp ứng được nhu cầu thực sự của khách hàng. Trong nhiều nghiệp vụ phức tạp hơn, việc sử dụng phán đoán sai có thể gây ra những lỗi lầm nghiêm trọng làm mất lòng tin của người dùng. Để có thể xác định đúng ý muốn thực sự của người dùng, trước khi đưa ra hướng dẫn, hệ thống cần hỏi lại để xác nhận xem người dùng muốn hỏi về hồ sơ mở thẻ, hồ sơ vay vốn hay hồ sơ đăng kí ngân hàng điện tử.

Khi người dùng hỏi về một nghiệp vụ nằm ngoài hệ thống như hủy vé máy bay, mô hình phân loại có thể đưa ra phán đoán về một ý định như "Phương thức hủy dịch vụ ngân hàng điện tử". Tuy nhiên, độ tự tin trong trường hợp này sẽ rất thấp vì câu nói không hướng đến bất cứ ý định nào trong lĩnh vực ngân hàng. Việc đưa ra phản hồi tương ứng với ý định do mô hình phân loại là không đúng. Thực tế, vì nghiệp vụ này nằm ngoài hệ thống, nên hệ thống sẽ không thể hiểu ý định thực sự của người dùng. Trong trường hợp này, hệ thống có thể đưa câu nói này khớp với một ý định mặc định (Default FallBack Intent) để có thể xin lỗi và hướng người dùng vào các nghiệp vụ mà một ngân hàng cung cấp hoặc chuyển tiếp cho nhân viên chăm sóc khách hàng.

Từ các ví dụ trên cho thấy, để hệ thống hội thoại có những hành xử hợp lý, chúng ta có thể sử dụng độ tự tin của phép đoán với hai ngưỡng MIN và MAX. Hai ngưỡng này cho phép hệ thống dựa trên độ tự tin để quyết định việc trả về ý định, hỏi lại người dùng hay kích hoạt ý định mặc định. Trong đó, ngưỡng MIN đảm bảo rằng độ tự tin của ý định được trả về luôn lớn hơn một ngưỡng cho trước, còn ngưỡng MAX được đưa vào để xác định khoảng nhập nhằng khi có nhiều hơn hai ý định nằm giữa khoảng MIN và MAX.

Trước khi có thể đưa ra dự đoán, một mô hình phân loại cần được huấn luyện trên một tập dữ liệu cho trước. Tập dữ liệu này bao gồm nhiều ý định với các cách diễn đạt khác mà người dùng có thể sử dụng để đưa ra ý định đó. Chi tiết các bước huấn luyện mô hình và và dự đoán kết quả sử dụng hai ngưỡng này được mô tả trong Hình 9.



Hình 9 Sơ đồ huấn luyện và dự đoán ý định sử dụng độ tự tin.

Tại bước huấn luyện, mỗi dữ liệu huấn luyện bao gồm câu nói của người dùng và các thực thể được trích xuất từ trong câu nói đó. Thực thể được sử dụng trong việc huấn luyện mô hình dự đoán ý định vì đây là các thông tin quan trọng trong câu, do đó, chúng cũng góp phần phân biệt các ý định khác nhau. Mỗi dữ liệu được gán với một ý định cụ thể. Đầu tiên, câu nói đầu vào sẽ được tiền xử lý để làm sạch văn bản. Bước tiền xử lý bao gồm loại bỏ từ dừng, chuyển văn bản thành dạng viết thường, chuẩn hóa từ. Từ dừng là các từ xuất hiện nhiều trong một ngôn ngữ nhưng không mang nhiều ý nghĩa. Chuẩn hóa từ là việc đưa dấu câu của một từ về đúng vị trí. Ví dụ người dùng có thể viết "hóa" hoặc "hoá". Mặc dù hai từ này có cùng một ý nghĩa, nhưng vì có cách viết khác nhau nên khi trích rút đặc trưng, chúng sẽ được coi là hai từ khác nhau. Do vậy, chúng ta cần chuẩn hóa để đưa hai từ này về cũng một từ "hóa". Sau khi tiền xử lý, hệ thống tiến hành trích chọn đặc trưng cho dữ liệu đầu vào. Để đơn giản, các thực thể sẽ được ghép nối trực tiếp vào câu nói và được coi như một từ trong văn bản. Tiếp đó, văn bản sau khi được ghép nối sẽ được trích chọn đặc trưng. Cuối dùng một thuật toán phân loại sẽ được chọn lựa để huấn luyện mô hình.

Trong bước dự đoán, các thực thể có trong câu nói của người dùng cũng được ghép nối trực tiếp vào văn bản. Văn bản sau khi ghép nối cũng sẽ trải qua quá trình tiền xử lý và trích chọn đặc trưng như trong bước huấn luyện để chuyển đổi thành đầu vào cho mô hình phân loại. Mô hình sau đó sẽ tiến hành dự đoán ý định cho đầu vào này. Kết quả thu được là độ tự tin

cho từng ý định. Dựa trên độ tự tin, mô hình tiến hành lựa chọn hàng động trả về. Chi tiết cách lựa chọn hành động được mô tả trong thuật toán sau.

Thuật toán Xác định hành động dựa trên độ tự tin

Đầu vào: Các ý định y_i, $i = 1,2, \dots, n$ và độ tự tin d_i tương ứng, với n là tổng ý định có trong hệ thống, ngưỡng tự tin MAX và MIN.

Đầu ra: ý định tốt nhất y^*, hành động hỏi lại ask_again, hoặc ý định mặc định y^{**}

Giải thuật:

begin
$imax \leftarrow 1$
$count\_between\_thresholds \leftarrow 1$
for $i \leftarrow 1$ **to** n **do**
  if $d_i > d_{imax}$ **then**
    $imax \leftarrow i$
  endif

  if $MIN \leq d_i < MAX$ **then**
    $count\_between\_thresholds \leftarrow count\_between\_thresholds + 1$
endfor
if $d_{imax} < MIN$ ***then***
  return y^{**}
else if $d_{imax} \geq MAX$ *or* $(d_{imax} < MAX$ *and* $coun\_between\_thresholds = 1)$ **then**
  $y^* \leftarrow y_{imax}$
  return y^*
else
  return ask_again
endif
end

Dựa các ý định và độ tự tin của chúng được tính toán qua bước phân loại ý định, mô hình sử dụng thuật toán xác định hành động dựa trên độ tự tin để tìn ra hành động phù hợp nhất. Thuật toán nhận đầu vào là danh sách các ý định kèm theo độ tự tin, ngưỡng MAX và MIN. Ý định mặc định sẽ được kích hoạt nếu như không có ý định nào có độ tự tin vượt qua ngưỡng MIN. Lúc này, mô hình được coi là không đoán được ý định của người dùng. Ngược lại, thuật toán sẽ trả về một ý định nếu như ý định đó thỏa mãn một trong hai điều kiện: (i) ý định có độ tự tin lớn nhất và độ tự tin này vượt qua ngưỡng MAX hoặc (ii) ý định có độ tự tin lớn nhất và là ý định duy nhất có độ tự tin nằm giữa ngưỡng MAX và MIN. Trong trường hợp còn lại, khi không có ý định nào có độ tự tin vượt ngưỡng MAX và tồn tại ít nhất hai ý định có độ tự tin nằm giữa khoảng MAX và MIN, mô hình đưa ra yêu cầu hỏi lại. Mô hình lúc này được coi là đã phát hiện ra có sự nhập nhằng trong câu nói của người dùng.

Cách lựa chọn ý định như vậy đảm bảo rằng ý định được trả về luôn có độ tự tin lớn hơn một ngưỡng MIN cho trước. Việc đưa ra hai ngưỡng tự tin cho phép chúng ta áo thể xác định một khoảng giá trị mà tại đó, hệ thống có thể phát hiện có sự nhập nhằng hay không. Do đó, khoảng MIN – MAX còn được gọi là khoảng nhập nhằng.

3.2 Tiêu chí đánh giá mô hình phân loại dựa trên độ tự tin

Một ý định sẽ được trả về khi nó có độ tự tin cao nhất và vượt ngưỡng MAX hoặc là ý định duy nhất có độ tự tin nằm trong khoảng nhập nhằng. Trong quá trình kiểm thử mô hình phân loại, để một dự đoán đúng được trả về, nó cần có độ tự tin ở mức cao hoặc trung bình. Mặc khác, khi ý định có độ tự tin nhỏ hơn ngưỡng MIN, mô hình kích hoạt mộ ý định mặc định. Do vậy, nếu độ tự tin của các câu sai là thấp, mô hình có thể làm giảm ảnh hưởng tiêu cực của nó bằng cách kích hoạt ý định mặc định hoặc đưa ra yêu cầu hỏi lại. Vì thế cho nên, một thuật toán phân loại tốt cần đưa ra độ tự tin cao cho các dự đoán đúng và độ tự tin thấp cho các dự đoán sai. Một mô hình như vậy sẽ làm cho độ chênh lệch giữa giá trị trung bình của các câu dự đoán đúng và sai là lớn. Với lập luận trên, em đưa ra tiêu chí đánh giá một mô hình phân loại dựa trên độ tự tin là độ chênh lệch độ tự tin trung bình của các dự đoán đúng và sai. Chỉ số này càng lớn thì mô hình càng có sự phân tách giữa các dự đoán đúng và sai.

Bên cạnh độ tự tin, em đưa ra hai tiêu chí là độ chính xác và thời gian dự đoán trung bình để có thể đánh giá một mô hình phân loại một cách đầy đủ và toàn diện. Độ chính xác cho biết một cách khái quát tỷ lệ các trường hợp dự đoán đúng trên tổng số các trường hợp đưa vào dự đoán và được tính bằng số dự đoán đúng trong tất cả các dự đoán trên tập kiểm tra. Độ chính xác càng cao thì hiệu quả của mô hình càng tốt. Về thời gian dự đoán trung bình, hệ thống đưa ra phản hồi kịp thời sẽ giúp cho người dùng cảm thấy vấn đề của bản thân được ưu tiên, được quan tâm, từ đó tăng sự hài lòng và củng cố niềm tin với doanh nghiệp. Vì thế, việc phản hồi kịp thời là một yếu tố rất quan trọng để tăng trải nghiệm của khách hàng. Ngoài việc dự đoán ý định, hệ thống hội thoại còn cần thực hiện nhiều công việc khác như lựa chọn hành động phù hợp, tạo kết quả đầu ra nên thời gian trung bình cho một dự đoán càng nhanh càng tốt.

Như vậy, để lựa chọn mô hình phân loại phù hợp đưa vào triển khai trong thực tế, có khả năng giúp nhà phát triển xây dựng các chiến lược khác nhau để giảm thiểu rủi ro trong việc đoán sai ý định của người dùng, chúng ta xem xét ba tiêu chí đánh giá: độ chênh lệch giữa giá trị trung bình của các dự đoán đúng và sai, độ chính xác và thời gian dự đoán trung bình. Trong đó, độ chênh lệch và độ chính xác có giá trị càng cao thì càng tốt, còn thời gian dự đoán càng ngắn càng tốt.

3.3 Lựa chọn mô hình phân loại

Trong thí nghiệm này, em tiến thành thực nghiệm trên các mô hình học máy và học sâu khác nhau. Các mô hình học máy bao gồm: Random Forest và SVM. Với việc tổng hợp ý kiến từ nhiều cây khác nhau, Random Forest đưa ra phán đoán một cách khách quan, do đó, mang lại kết quả chính xác cao. SVM sử dụng các véc-tơ hỗ trợ, có thể nhanh chóng xác định phân lớp của văn bản đầu vào. Các mô hình học sâu gồm có: CNN, LSTM, LSTM sử dụng cơ chế chú ý (LSTM – Attention), BiLSTM, BiLSTM sử dụng cơ chế chú ý (BiLSTM – Attention). Đây đều là các mô hình mạng có khả năng học các đặc trưng ẩn từ văn bản đầu vào. Ngoài ra, các mạng xử lý chuỗi như LSTM và BiLSTM có khả năng nắm bắt thông tin ngữ nghĩa cho cả câu theo trình tự thời gian. Cơ chế chú ý cho phép mô hình chú ý đến các từ ngữ quan trọng để phân loại ý định.

Các thuật toán học máy, bao gồm Random Forest và SVM, được điều chỉnh tham số bằng cách sử dụng thuật toán Grid Search. Grid Search là thuật toán điều chỉnh tham số bằng cách liệt kê tất cả các tổ hợp của các tham số, sau đó kiểm thử tất cả các tổ hợp này để chọn ra bộ tham số cho kết quả tốt nhất.

3.4 Trích xuất đặc trưng

Đầu vào của các mô hình bao gồm các câu mẫu và ý định tương ứng. Trong mỗi câu mẫu còn chứa thông tin thực thể nằm trong câu. Vì thực thể là một thành phần nắm giữ các thông tin quan trọng trong câu, do đó, bên cạnh nội dung của văn bản, các thực thể cũng đóng vai trò quan trọng trọng việc xác định ý định của người dùng. Để có thể khai thác được thông tin này, tên của thực thể được ghép trực tiếp vào trong câu, có vai trò như một từ giúp phân loại ý định của câu. Ví dụ, một câu mẫu được thể hiện trong Bảng 2:

Bảng 2 Ví dụ về câu mẫu

Câu mẫu	Đơn phát hành	tài khoản
Thực thể		loại_tài_khoản
Ý định	Hỏi biểu mẫu	

Bảng 2 câu nói "Đơn phát hành tài khoản" bao gồm một thực thể *loại_tài_khoản* có giá trị là "tài khoản". Tên của thực thể được ghép vào câu mẫu, khi đó, cả cụm "Đơn phát hành tài khoản loại_tài_khoản" được sử dụng để dự đoán ý định "Hỏi biểu mẫu".

Các đặc trưng được sử dụng để phân loại ý định bao gồm các từ trong câu đầu vào của người dùng và các thực thể trích xuất được từ câu nói đó. Một cách đơn giản để có thể véc-

tơ hóa các thông tin này là coi thực thể như một từ trong văn bản và ghép chúng vào nội dung của văn bản và thực hiện véc-tơ hóa. Để không bị nhầm lẫn với các từ trong văn bản, các thực thể được viết dưới dạng gạch nối giữa các tiếng và thêm tiếp đầu ngữ "alias". Ví dụ trong Bảng 2 sẽ trở thành một điểm dữ liệu bao gồm văn bản và nhãn tương ứng được cho trong Bảng 3.

Bảng 3 Ví dụ về một điểm dữ liệu

Văn bản	Đơn phát hành tài khoản alias_loại_tài_khoản
Nhãn	Biểu mẫu

Các phép trích chọn đặc trưng được sử dụng trong bài thử nghiệm này bao gồm: TF-IDF, Word2Vec, Fasttext. Trong đó, TT-IDF được sử dụng cùng các thuật toán học máy. Word2Vec, FastText được sử dụng trong cả thuật toán học máy và các thuật toán học sâu gồm CNN, LSTM, LSTM-Attention, BiLSTM và BiLSTM-Attention. Word2Vec và Fasttext biểu diễn một từ bằng một véc-tơ trong không gian nhiều chiều.

Các thuật toán học máy nhận đầu vào là một véc-tơ một chiều. Do đó, để có thể mã hóa văn bản bằng Word2Vec và Fasttext trong các thử nghiệm với các thuật toán học máy, chúng ta véc-tơ hóa văn bản bằng một véc-tơ có độ dài bằng độ dài từ điển, trong đó giá trị của phần tử thứ i tương ứng với từ thứ i trong từ điển. Giá trị này bằng 0 nếu từ không xuất hiện trong văn bản và bằng giá trị trung bình của véc-tơ từ nếu từ nằm trong văn bản.

3.5 Thực nghiệm và đánh giá

3.5.1 Dữ liệu

Tập dữ liệu được sử dụng trong thử nghiệm này được xây dựng trong lĩnh vực ngân hàng, bao gồm các câu nói mà một người có thể hỏi về nghiệp vụ của một ngân hàng. Bộ dữ liệu được xây dựng với dữ liệu thô được thu thập tự động trên các bình luận, trao đổi trong các nhóm và diễn đàn trên mạng xã hội bởi các chuyên viên dữ liệu về ngân hàng. Từ đó, các câu nói được tạo ra và gán nhãn dựa trên nền tảng hội thoại thông minh SmartDialog [2]. Chi tiết thông tin về bộ dữ liệu được cho trong Bảng 4.

Bảng 4 Thông tin về bộ dữ liệu ngân hàng

Số lượng ý định	121
Số lượng thực thể	18
Số lượng câu	14.150

Số câu ít nhất trong một ý định	1
Số câu nhiều nhất trong một ý định	874
Số câu trung bình trong một ý định	116

Bộ dữ liệu trên bao gồm 121 ý định với tổng số 14.150 câu mẫu và 14 thực thể. Số câu nhiều nhất trong một ý định là 874 câu và số câu ít nhất là 1 câu. Trung binh mỗi ý định chưa 116 câu mẫu. Giống như hầu hết các tập dữ liệu thực tế, bộ dữ liệu này gặp phải vấn đề mất cân bằng, vì trung bình số câu mẫu trong một ý định là 116 câu, trong khi lớp có ít câu mẫu nhất chỉ có 1 câu thì lớp có nhiều mẫu nhất là 874 câu.

Mất cân bằng làm cho kết quả dự đoán sẽ nghiêng về lớp có nhiều dữ liệu hơn, ảnh hưởng đến tính chính xác của mô hình. Để khắc phục điều này, trong thử nghiệm, em sử dụng phương pháp oversampling. Phương pháp này là một cách làm giảm cân bằng dữ liệu bằng cách thêm ngẫu nhiên các câu mẫu vào trong mỗi ý định sao cho số lượng câu của mỗi ý định bằng với số câu lớn nhất của một ý định hoặc số câu trung bình trên một ý định. Ở đây, em lựa chọn việc thêm số lượng câu sao cho số câu trong một ý định bằng với số câu trung bình, trong trường hợp ý định đó có ít hơn số câu trung bình trên một ý định.

3.5.2 Đánh giá các thuật toán học máy

Kết quả thử nghiệm với các thuật toán học máy và các phép trích chọn đặc trưng khác nhau được cho trong **Bảng 5**.

Bảng 5 Kết quả so sánh các mô hình học máy và các phép trích chọn đặc trưng khác nhau

Chú thích Độ tự tin T là trung bình độ tự tin của tất cả các dự đoán đúng. Độ tự tin F là trung bình của tất cả các dự đoán sai. RF: RandomForest. W2V: Word2Vec.

	Độ chính xác (%)	**Độ tự tin T**	**Độ tự tin F**	**Chênh lệch**	**Thời gian xử lý (giây)**
RF - TF-IDF	88,35	0,70	0,45	0,25	0,15
RF - Fasttext	87,67	0,70	0,50	0,21	0,17
RF - W2V	87,90	0,71	0,50	0,21	0,22
SVM – TF-IDF	**93,25**	0,79	0,36	0,43	**0,03**

	Độ chính xác (%)	Độ tự tin T	Độ tự tin F	Chênh lệch	Thời gian xử lý (giây)
SVM - Fasttext	43,54	0,74	0,15	**0,59**	0,06
SVM - W2V	65,66	0,79	0,28	0,51	0,06

Bảng 5 cho thấy thuật toán SVM và phép trích chọn đặc trưng Fasttext cho độ chênh lệch giữa hai giá trị trung bình của độ tự tin là tốt nhất 0,59, tuy nhiên độ chính xác lại quá thấp, chỉ có 43,54%. SVM và TF-IDF cho kết quả độ chính xác cao nhất, 93,25%, với độ chênh lệnh giữa hai giá trị trung bình ở mức cao 0.43 và thời gian dự đoán nhanh nhất 0,03 giây. Mô hình Random Forest và TF-IDF có thời gian dự đoán trung bình nhanh 0,15 giây, tuy nhiên độ chênh lệch giữa hai giá trị trung bình của độ tự tin chỉ có 0.25.

Kết quả thực nghiệm này cũng cho thấy TF-IDF là phép trích chọn đặc trưng tốt cho các thuật toán học máy khi độ chính xác với SVM là Random Forest lần lượt là 93,25% và 88,35%. Word2Vec và Fasttext không phải phép trích chọn đặc trưng phù hợp vì việc lấy giá trị trung bình của véc-tơ từ đã làm giảm đi ý nghĩa véc-tơ của từ.

Trong thực nghiệm này, SVM và TF-IDF, Random và TF-IDF là hai thuật toán cho kết quả tốt nhất xét trên cả ba tiêu chí đánh giá.

3.5.3 Đánh giá các thuật toán học sâu

Nhóm các thuật toán CNN, LSTM, LSTM-Attention, BiLSTM, BiLSTM-Attentin và phép trích chọn đặc trưng Word2Vec

Kết quả so sánh các mô hình học sâu cùng phép trích chọn đặc trưng Word2Vec được mô tả trong Bảng 6.

Bảng 6 Kết quả so sánh các mô hình học sâu cùng phép trích chọn đặc trưng Word2Vec

Chú thích Độ tự tin T là trung bình độ tự tin của tất cả các dự đoán đúng. Độ tự tin F là trung bình của tất cả các dự đoán sai.

	Độ chính xác (%)	Độ tự tin T	Độ tự tin F	Chênh lệch	Thời gian dự đoán (giây)
CNN	84,44	0,99	0,93	0,06	0,02
LSTM	91,28	0,99	0,86	0,13	0,02

	Độ chính xác (%)	Độ tự tin T	Độ tự tin F	Chênh lệch	Thời gian dự đoán (giây)
LSTM-Attention	89,57	0,99	0,86	0,13	0,02
BiLSTM	**93,99**	0,99	0,83	**0,16**	0,02
BiLSTM-Attention	92,59	0,99	0,86	0,13	0,02

Nhìn chung, các thuật toán học sâu đều có thời gian dự đoán rất nhanh, khoảng 0,02s. Các thuật toán này đều đạt độ chính xác rất cao, từ 84,44% trở lên, trong đó BiLSTM đạt độ chính xác cao nhất với 93,99%. Tuy nhiên, các mô hình này lại cho độ chênh lệch giữa hai giá trị trung bình của độ tự tin thấp, chi khoảng 0,13 – 0,16. Độ tự tin trên trung bình trên các câu đúng và các câu sai của các mô hình mạng cũng rất cao, trong đó độ tự tin trung bình trên các câu đúng là 0.99 còn trên các câu sai từ 0,83 – 0,93. Hai mô hình tốt nhất được chọn trong thử nghiệm này là BiLSTM và BiLSTM-Attention.

Nhóm các thuật toán CNN, LSTM, LSTM-Attention, BiLSTM, BiLSTM-Attentin và phép trích chọn đặc trưng Fasttext

Kết quả so sánh các mô hình học sâu cùng phép trích chọn đặc trưng Fasttext được mô tả trong Bảng 7.

Bảng 7 Kết quả so sánh các mô hình học sâu cùng phép trích chọn đặc trưng Fasttext

Chú thích Độ tự tin T là trung bình độ tự tin của tất cả các dự đoán đúng. Độ tự tin F là trung bình của tất cả các dự đoán sai.

	Độ chính xác (%)	Độ tự tin T	Độ tự tin F	Chênh lệch	Thời gian dự đoán (giây)
CNN	78,09	0,96	0,76	**0,20**	0,02
LSTM	78,21	0,98	0,85	0,13	0,02
LSTM-Attention	76,56	0,98	0,91	0,07	0,02
BiLSTM	80,29	0,96	0,81	0,15	0,02
BiLSTM-Attention	**83,60**	0,98	0,87	0,11	0,02

Trong thử nghiệm này, các mô hình đều cho thời gian dự đoán rất nhanh, vào khoảng 0,02s. Độ chính xác của các mô hình rơi vào khoảng 78,09 – 83,60%. Giống như thử nghiệm trước, các mô hình này cũng cho độ chênh lệch giữa hai giá trị trung bình của độ tự tin rất thấp, chỉ khoảng 0,07 – 0,20. Kết quả này cho thấy Word2Vec phù hợp hơn so với Fasttext trong các mô hình học sâu. Hai mô hình tốt nhất là BiLSTM và BiLSTM – Attention.

3.5.4 Đánh giá các thuật toán tốt nhất

Kết quả so sánh các mô hình tốt nhất trong các thử nghiệm trên được trình bày trong Bảng 8.

Bảng 8 Kết quả so sánh giữa các thuật toán tốt nhất

Độ tự tin T là trung bình độ tự tin của tất cả các dự đoán đúng. Độ tự tin F là trung bình của tất cả các dự đoán sai. RF: Random Forest. W2V: Word2Vec.

	Độ chính xác (%)	**Độ tự tin T**	**Độ tự tin F**	**Chênh lệch**	**Thời gian dự đoán (giây)**
RF - TF-IDF	88,35	0,70	0,45	0,25	**0,15**
SVM – TF-IDF	**93,25**	0,79	0,36	**0,43**	0,03
BiLSTM-W2V	**93,99**	0,99	0,83	0,16	0,02
BiLSTM-Attention-W2V	92,59	0,99	0,86	0,13	0,02
BiLSTM-Fasttext	80,29	0,96	0,81	0,15	0,02
BiLSTM-Attention-Fasttext	83,60	0,98	0,87	0,11	0,02

Bảng 8 cho thấy thuật toán BiLSTM – Word2Vec cho kết quả độ chính xác cao nhất 93,99%, tuy nhiên, độ chênh lệch giữa giá trị trung bình của các câu đúng và sai quá thấp, chỉ là 0,13 nên không phù hợp để sử dụng trong hệ thống hội thoại. SVM – TF-IDF là mô hình đạt độ chính xác cao 93,25%, đồng thời có độ chênh lệch giữa các giá trị trung bình của độ tự tin cao nhất, 0,43 và có thời gian dự đoán tương đối nhanh 0,03 giây. Random Forest – TF-IDF có thời gian dự đoán nhanh nhất nhưng độ chênh lệch giữa hai giá trị trung bình của độ tự tin chỉ có 0,25. Các mô hình mạng đạt kết quả cao về độ chính xác, nhưng độ tự tin của chúng quá cao nên không có khả năng phân hóa giữa các câu đúng và sai dựa trên độ tự tin.

Từ kết quả thực nghiệm trên cho thấy, mô hình mạng tuy đạt độ chính xác cao nhưng thiếu khả năng phân hóa giữa các câu đúng và sai. Mô hình SVM – TF-IDF cho độ chính xác tốt, có khả năng phân hóa và thời gian dự đoán nhanh. Vì vậy, SVM – TF-IDF là mô hình phù hợp để đưa vào triển khai trong thực tế. SVM sử dụng TF-IDF để trích chọn đặc trưng cũng là mô hình được lựa chọn để sử dụng trong thử nghiệm thứ hai của đồ án – thử nghiệm phân loại ý định người dùng dựa theo ngữ cảnh.

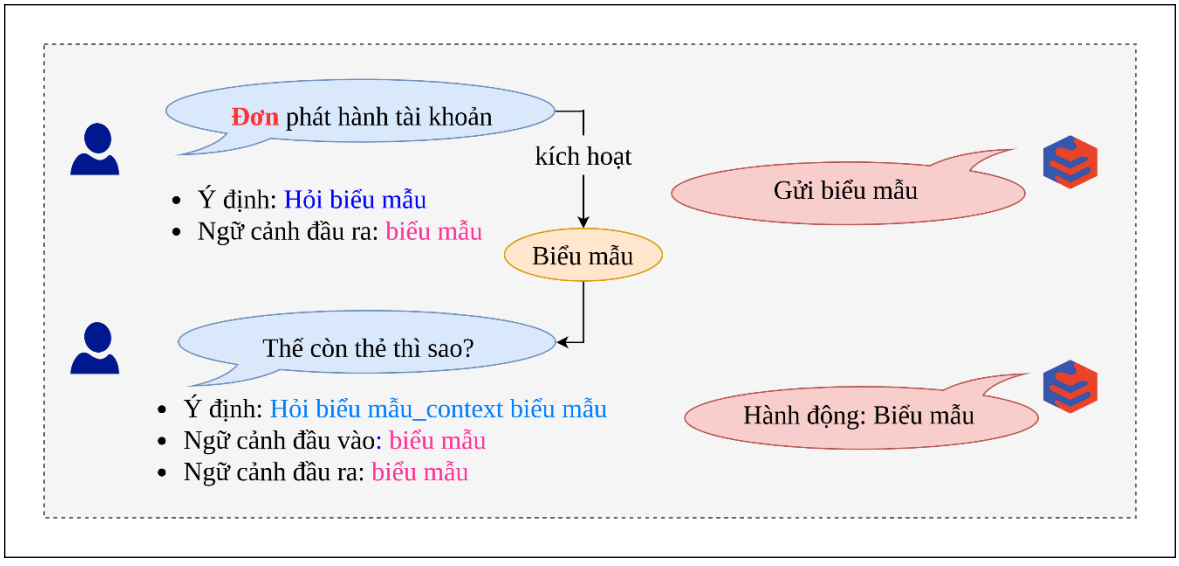
Chương 4 Mô hình dự đoán ý định có ngữ cảnh trong hội thoại

Từ kết quả thuật toán được đánh giá trong chương 3, chương này đưa ra mô hình phân loại người dùng có sử dụng ngữ cảnh theo hai bước: bước 1 sử dụng ngữ cảnh và bước 2 không sử dụng ngữ cảnh và đánh giá tính hiệu quả của mô hình đã đề xuất.

4.1 Ngữ cảnh

Khái niệm ngữ cảnh (context) được xây dựng dựa trên ngữ cảnh trong thực tế. Nếu người dùng hỏi rằng "Thế còn thẻ thì sao?", chúng ta cần phải biết người dùng đang mong muốn biết về nghiệp vụ gì của thẻ (ví dụ như mở thẻ, đóng thẻ, khóa thẻ, ...). Trong các cuộc hội thoại, thông tin này thường sẽ được đề cập trong các trao đổi trước đó. Do đó, chúng ta sử dụng ngữ cảnh như một cách để lưu lại thông tin này.

Để có thể hiểu được nghiệp vụ của thẻ đang để cập đến, chúng ta định nghĩa các ngữ cảnh đầu vào (input context) và ngữ cảnh đầu ra (output context) cho mỗi ý định. Khi một ý định được trả về, ngữ cảnh đầu ra của ý định đó được kích hoạt, ngữ cảnh sau khi được kích hoạt gọi là ngữ cảnh kích hoạt (active context). Ý định tiếp theo muốn được lựa chọn cần có ngữ cảnh đầu vào là tập con của tập các ngữ cảnh kích hoạt đang có tại thời điểm đoán. Ví dụ trong Hình 10 mô tả cách sử dụng ngữ cảnh trong hội thoại.



Hình 10 Ví dụ về ngữ cảnh.

Đầu tiên, người dùng hỏi về đơn phát hành một tài khoản. Từ câu đầu vào, hệ thống sẽ phán đoán ý định của người dùng qua câu nói này là "Hỏi biểu mẫu" và trả về hành động tương ứng là "Gửi biểu mẫu". Ý định này có một ngữ cảnh đầu ra là "biểu mẫu" nên ngữ cảnh "biểu mẫu" được kích hoạt sau khi ý định được trả về. Tiếp theo, người dùng tiếp hỏi "Thế còn thẻ thì sao?". Lúc này, hệ thống sẽ dự đoán ý định người dùng muốn hỏi về biểu mẫu của thẻ, tương ứng với ý định "Hỏi biểu mẫu_context biểu mẫu" vì ý định này có ngữ cảnh đầu vào là "biểu mẫu", là một tập con của tập ngữ cảnh kích hoạt hiện tại. Cuối cùng, hệ thống trả về hành động dựa trên ý định được đoán ra: "Gửi biểu mẫu".

4.1.1 Ngữ cảnh đầu ra

Ngữ cảnh đầu ra (output context) được sử dụng để tạo ra các ngữ cảnh kích hoạt. Khi một ý định được đoán vào, tất cả các ngữ cảnh đầu ra của nó đều được kích hoạt. Một ý định có thể có một hoặc nhiều ngữ cảnh đầu ra. Mỗi ngữ cảnh này sẽ có thời gian sống (lifeSpan), được định nghĩa là số lượt hội thoại mà ngữ cảnh được kích hoạt. Khi một ý định được đoán và ngữ cảnh đầu ra của nó đã được kích hoạt trước đó, thời gian sống và tuổi thọ (lifeTime) của ngữ cảnh đó sẽ được đặt lại. Bảng 9 đưa ra một ví dụ về ngữ cảnh đầu ra:

Bảng 9 Ví dụ về ngữ cảnh đầu ra

Ý định	Câu mẫu	Ngữ cảnh đầu vào	Ngữ cảnh đầu ra	Phản hồi
pet-init	- "Chúng trông như thế nào?" - "Chúng kêu như thế nào?" - "Chúng nặng như thế nào?"	-	-	"Bạn muốn biết về con vật nào?"
pet-select-dogs	- "Mình thích chó"	-	dogs (lifeSpan = 2)	"Bạn muốn biết điều gì về chó?"
pet-select-cats	"Mình thích mèo"	-	cats (lifeSpan = 2)	"Bạn muốn biết gì về mèo"
dog-show	"Chúng trông như thế nào"	dogs	-	"Đây là hình ảnh của một con chó"
cat-show	"Chúng trông như thế nào"	cats	-	"Đây là hình ảnh của một con mèo"
dog-sound	"Chúng kêu như thế nào"	dogs	-	"Con chó kêu như thế này nhé"
cat-sound	"Chúng kêu như thế nào"	cats	-	"Con mèo kêu như thế này nè"

Ý định	Câu mẫu	Ngữ cảnh đầu vào	Ngữ cảnh đầu ra	Phản hồi
dog-size	"Chúng nặng như thế nào"	dogs	-	"Một chú chó có thể to đến như thế này"
cat-size	"Chúng nặng như thế nào"	cats	-	"Một chú mèo có thể to như thế này"

Nhờ việc sử dụng ngữ cảnh, một câu usersay có thể thuộc về nhiều ý định khác nhau. Ở trong Bảng 9, "Chúng trông như thế nào?" là câu mẫu thuộc về ba ý định khác nhau "pet-init", "dog-show", "cat-show" và "Chúng kêu như thế nào?" cũng thuộc về ba ý định khác nhau là "pet-init", "dog-sound" và "cat-sound". Bảng 10 đưa ra một ví dụ và giải thích tương ứng qua từng lượt hội thoại về các ngữ cảnh đầu ra và tời gian sống của ngữ cảnh trong một phiên hội thoại.

Bảng 10 Ví dụ sử dụng ngữ cảnh trong một phiên hội thoại

Lượt	Hội thoại	Giải thích
1	Người dùng: Chúng trông như thế nào? Bot: Bạn muốn biết về con vật nào?	Ý định "pet-init" được trả về
2	Người dùng: Mình thích chó Bot: Bạn muốn biết điều gì về con chó	Ý định "pet-select-dogs" được trả về, "dogs" là ngữ cảnh đầu vào của ý định này, do đó, nó trở thành ngữ cảnh kích hoạt.
3	Người dùng: Chúng trông như thế nào? Bot: Đây là hình ảnh của một con chó	Ý định "dog-show" có ngữ cảnh đầu ra là "dogs", do đó nó được trả về, "dogs" là ngữ cảnh kích hoạt với tuổi thọ thời gian sống bằng 1
4	Người dùng: Chúng kêu như thế nào? Bot: Con chó kêu như thế này nhé	Ý định "dog-sound" có ngữ cảnh đầu vào là "dogs" nên được trả về, tuổi thọ của "dogs" là 2, bằng với thời gian sống nên "dogs" bị xóa khỏi tập ngữ cảnh kích hoạt.
5	Người dùng: Chúng nặng bao nhiêu cân? Bot: Bạn muốn hỏi về con vật nào?	Lúc này, tập ngữ cảnh kích hoạt rỗng nên ý định "pet-init" được trả về.

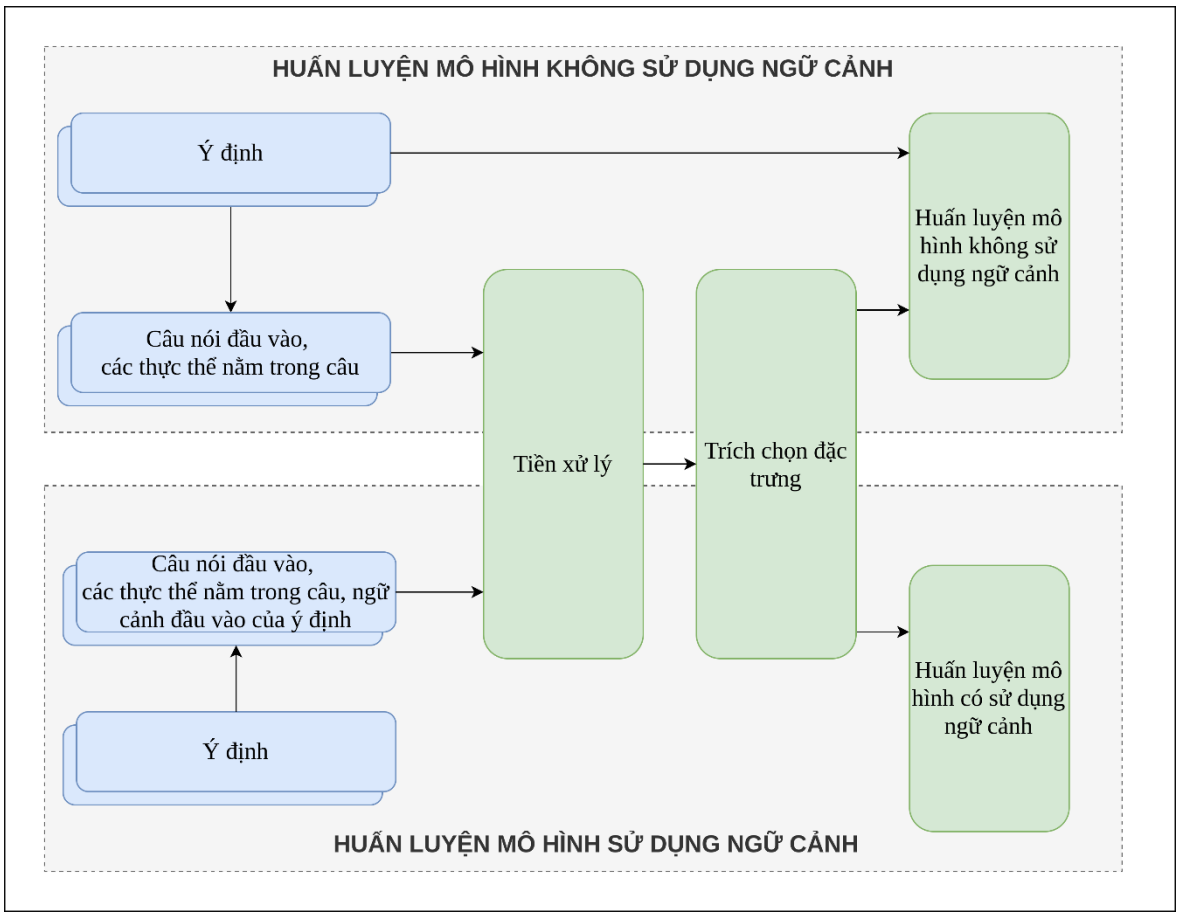
4.1.2 Ngữ cảnh đầu vào

Một ý định sẽ được trả về khi nó thỏa mãn các điều kiện sau: (i) ngữ cảnh đầu vào của ý định đó phải là tập con của tập các ngữ cảnh kích hoạt, (ii) ý định có độ tự tin cao nhất và (iii) có độ tự tin vượt ngưỡng MAX hoặc là ý định duy nhất có độ tự tin nằm giữa khoảng MAX và

MIN. Trong đó, MAX và MIN là hai ngưỡng về độ tự tin do người phát triển hệ thống đặt ra. Nếu tất cả các ý định đều có độ tự tin dưới ngưỡng MIN thì hệ thống sẽ trả về ý định mặc định, đồng nghĩa với việc hệ thống đang không hiểu điều mà người dùng nói. Trường hợp tất cả các ý định có độ tự tin dưới ngưỡng MAX, nếu có ít nhất hai ý định có ngữ cảnh đầu vào là tập con của tập các ngữ cảnh kích hoạt, có độ tự tin nằm trên ngưỡng MIN thì hệ thống sẽ lựa chọn 3 ý định có độ tự tin cao nhất trong các ý định này để hỏi lại.

4.2 Mô hình đề xuất

Ý tưởng xây dựng mô hình bắt nguồn từ thực tế. Nếu đối phương nói một câu hoàn chỉnh và đầy đủ thông tin, chúng ta có thể ngay lập tức hiểu được ý muốn của họ. Ngược lại, nếu đối phương chỉ đưa ra một câu nói chung chung, chúng ta sẽ liên hệ ngữ cảnh cuộc trò chuyện để đoán ý định của đối phương. Do vậy, em xây dựng hai mô hình dự đoán, một mô hình sử dụng ngữ cảnh và một mô hình không sử dụng ngữ cảnh. Để có thể đưa ra được dự đoán, đầu tiên, các mô hình cần được huấn luyện với dữ liệu. Sơ đồ huấn luyện các mô hình được miêu tả trong Hình 11.



Hình 11 Sơ đồ huấn luyện mô hình dự đoán ý định sử dụng ngữ cảnh.

Trong sơ đồ được mô tả ở Hình 11, đầu tiên, chúng ta huấn luyện mô hình không sử dụng ngữ cảnh. Chúng ta mong muốn rằng những câu đầy đủ thông tin sẽ được dự đoán với một độ tự tin cao. Trong trường hợp có thể xảy ra nhập nhằng giữa các ý định khác nhau, mô hình có thể đưa ra độ tự tin thấp cho các ý định dễ nhập nhằng với nhau. Do đó, mô hình này không sử dụng ngữ cảnh đưa vào huấn luyện, đồng thời được huấn luyện với tất cả dữ liệu, bao gồm dữ liệu của các ý định có ngữ cảnh đầu vào và không có ngữ cảnh đầu vào. Tương tự như ở chương trước, mô hình này sử dụng tập dữ liệu với mỗi điểm dữ liệu là một câu nói của người dùng cùng các thực thể được trích xuất từ câu nói đó. Mỗi điểm dữ liệu tương ứng với một ý định. Trong bước này, văn bản cũng trải qua quá trình tiền xử lý và trích xuất đặc trưng rồi được đưa vào một thuật toán phân loại để học ra một mô hình. Các bước tiền xử lý bao gồm loại bỏ các từ dừng, đưa dữ liệu về dạng viết thường và chuẩn hóa dữ liệu. Các thực thể được ghép nối trực tiếp vào văn bản với tiếp đầu ngữ "alias". Trong các trường hợp câu nói của người dùng là rõ ràng, mô hình này được sử dụng để dự đoán ý muốn của người dùng. Với các trường hợp không rõ ràng, mô hình được sử dụng để phát hiện nhập nhằng xảy ra.

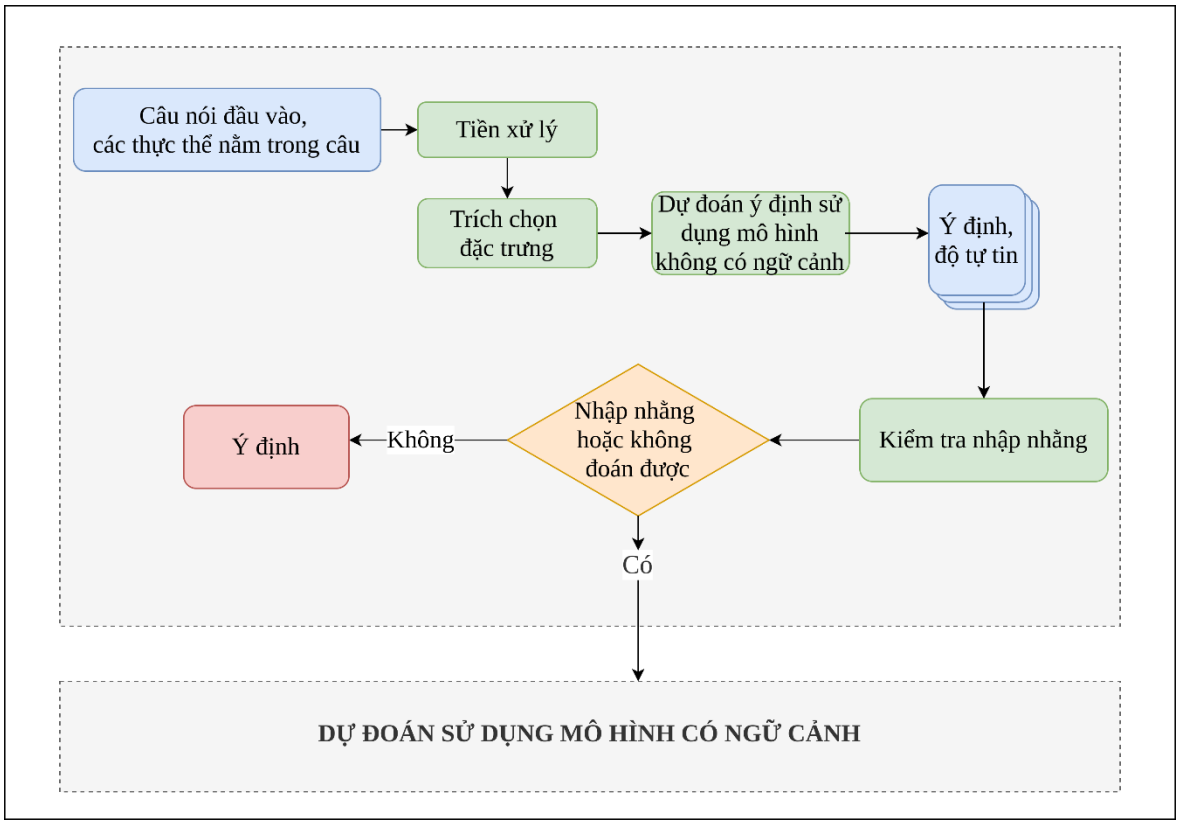
Mô hình có ngữ cảnh cũng trải qua các bước huấn luyện tương tự. Mô hình này được xây dựng với mục đích có thể phân biệt các ý định khác nhau tùy vào ngữ cảnh. Do đó, mô hình được huấn luyện với tập dữ liệu của các ý định có ngữ cảnh đầu vào. Ngữ cảnh đầu vào của ý định được đưa vào để làm đặc trưng huấn luyện mô hình. Lý do cho việc này vì chúng ta lý tưởng hóa rằng tại câu nói của người dùng đang được đặt vào đúng ngữ cảnh này. Cũng giống như thực thể, các ngữ cảnh đầu vào được ghép nối trực tiếp với văn bản, sử dụng tiếp đầu ngữ "context". Trong trường hợp mô hình không sử dụng ngữ cảnh phát hiện có nhập nhằng hoặc không đoán được ý muốn của người dùng, mô hình có ngữ cảnh sẽ được sử dụng. Vai trò của mô hình này là phát hiện ý định của người dùng trong các ngữ cảnh khác nhau.

Trong bước dự đoán, lúc này, trong hệ thống sẽ có một tập các ngữ cảnh được kích hoạt hoặc không. Để dự đoán được ý định của người dùng với tập ngữ cảnh kích hoạt này, ta sử dụng mô hình dự đoán hai bước. Bước 1 có vai trò dự đoán những ý định đầy đủ thông tin và phát hiện sự nhập nhằng trong câu nói của người dùng. Trường hợp mô hình nhận thấy có nhập nhằng hoặc không đoán được ý định, nếu trong hệ thống tồn tại các ngữ cảnh kích hoạt, việc dự đoán được chuyển sang bước 2. Trong bước 2, câu nói đầu vào của người dùng sẽ được kết hợp với các ngữ cảnh kích hoạt để dự đoán ý định người dùng. Chi tiết thực hiện các bước được mô tả trong phần 4.2.1 và 4.2.2.

4.2.1 Bước 1: Dự đoán ý định người dùng không sử dụng ngữ cảnh

Một câu nói đầy đủ thông tin từ người dùng sẽ được mô hình dự đoán với độ tự tin cao. Mặt khác, nếu câu nói này có thể thuộc về nhiều ý định khác nhau tùy vào ngữ cảnh hội thoại, lúc này, việc sử dụng mô hình huấn luyện không dùng ngữ cảnh sẽ làm cho độ tự tin của các

ý định đó có độ tự tin xấp xỉ nhau. Do vậy, chúng sẽ rơi vào khoảng nhập nhằng giữa hai ngưỡng MAX và MIN hoặc thậm chí có độ tự tin nhỏ hơn ngưỡng MIN. Do đó, quá trình dự đoán ở bước 1 sử dụng mô hình không có ngữ cảnh với hai mục đích. Mục đích thứ nhất là dự đoán ý định của những câu nói mà thông tin đã được đưa ra đầy đủ. Mục đích thứ hai là phát hiện có nhập nhằng xảy ra giữa các ý định hay không để xem xét đưa ngữ cảnh vào dự đoán. Trường hợp có xảy ra nhập nhằng, câu nói của người dùng sẽ được chuyển đến bước 2. Sơ đồ dự đoán theo hai bước được mô tả trong Hình 12.

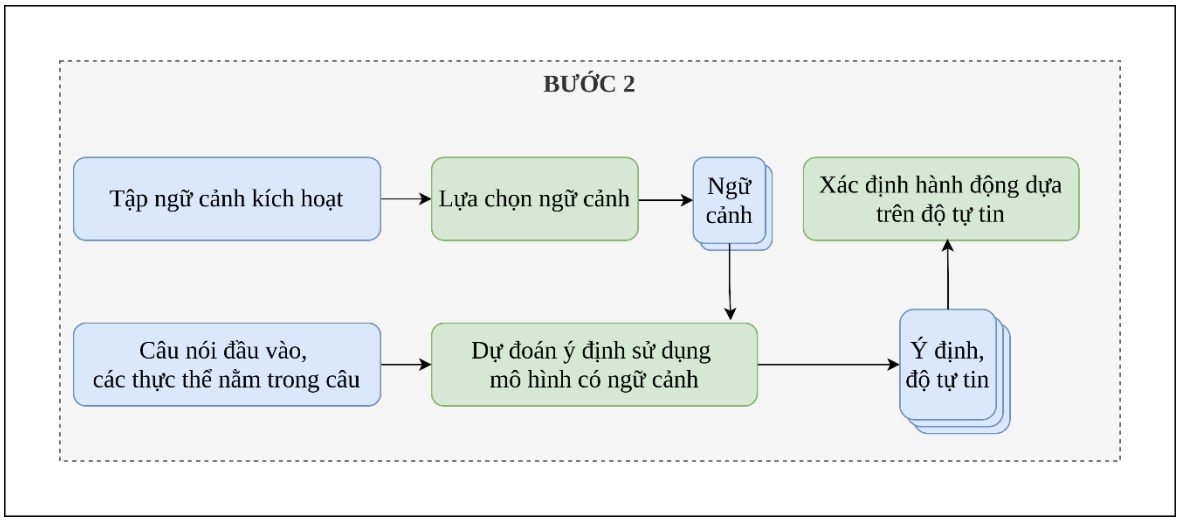


Hình 12 Sơ đồ dự đoán 2 bước.

Trong bước 1, câu nói đầu vào và các thực thể được trích xuất từ trong câu nói này được trải qua bước tiền xử lý và trích chọn đặc trưng giống như trong quá trình huấn luyện. Sau khi đưa qua mô hình phân loại, chúng ta thu được các ý định với độ tự tin tương ứng. Chúng ta chỉ xem xét các ý định có ngữ cảnh đầu là tập con của tập các ý định kích hoạt tồn tại trong hệ thống. Lúc này, nếu ý định có độ tự tin cao nhất vượt qua ngưỡng MAX hoặc là ý định duy nhất nằm giữa MAX và MIN, ý định sẽ được trả về. Ngược lại, mô hình sẽ kiểm tra xem có ngữ cảnh nào đang được kích hoạt hay không. Nếu có, mô hình sẽ tiến hành dự đoán lần 2. Nếu không, ý định có độ tự tin cao nhất sẽ sẽ được xem xét xem nó có vượt qua ngưỡng MIN hay không. Nếu không, ý định mặc định được kích hoạt. Ngược lại, mô hình đang thấy có sự nhập nhằng nên đưa ra yêu cầu hỏi lại.

4.2.2 Bước 2: Dự đoán ý định người dùng có sử dụng ngữ cảnh

Mục đích của việc dự đoán lần 2 là để dự đoán được ý định trong các ngữ cảnh khác nhau với các câu nói gần giống nhau. Trong mô hình này, chúng ta cũng sử dụng các ngưỡng MAX' và MIN' để lựa chọn ý định. Các giá trị này có thể giống hoặc khác so với lần đoán thứ nhất. Sơ đồ dự đoán trong bước 2 được mô tả trong Hình 13.



Hình 13 Sơ đồ dự đoán lần 2.

Trong lần 2, mô hình thực hiện dự đoán ý định qua các bước: lựa chọn ngữ cảnh, dự đoán ý định và xác định hành động dựa trên độ tự tin của phép đoán. Bước lựa chọn ngữ cảnh sẽ chọn ra các ngữ cảnh có liên quan nhất đến câu nói của người dùng trong tập tất cả các ngữ cảnh đang được kích hoạt. Ngữ cảnh được lựa chọn sẽ kết hợp với câu nói của người dùng và thực thể trong câu để mô hình có thể dự đoán ra một tập các ý định và độ tự tin tương ứng. Các ý định và độ tự tin này sẽ được sử dụng để xác định hành động tiếp theo là gì.

4.2.2.1 Lựa chọn ngữ cảnh

Tập ngữ cảnh kích hoạt tại một bước dự đoán có thể rất nhiều nên chúng ta cần lựa chọn các ngữ cảnh phù hợp nhất để đưa vào phán đoán ý định. Thông thường, ngữ cảnh quan trọng sẽ xuất hiện gần với thời điểm xuất hiện câu nói nhất nên ngữ cảnh kích hoạt nào có tuổi thọ ngắn hơn thì sẽ có nhiều khả năng là ngữ cảnh mà chúng ta cần quan tâm. Thời gian sống cũng là một yếu tố để lựa chọn ngữ cảnh. Nhà phát triển hệ thống sẽ đặt thời gian sống dài hơn cho những ngữ cảnh có tầm quan trọng với nội dung trao đổi sau đó. Vì vậy, em đưa ra công thức tính điểm cho một ngữ cảnh kích hoạt z theo Công thức 11.

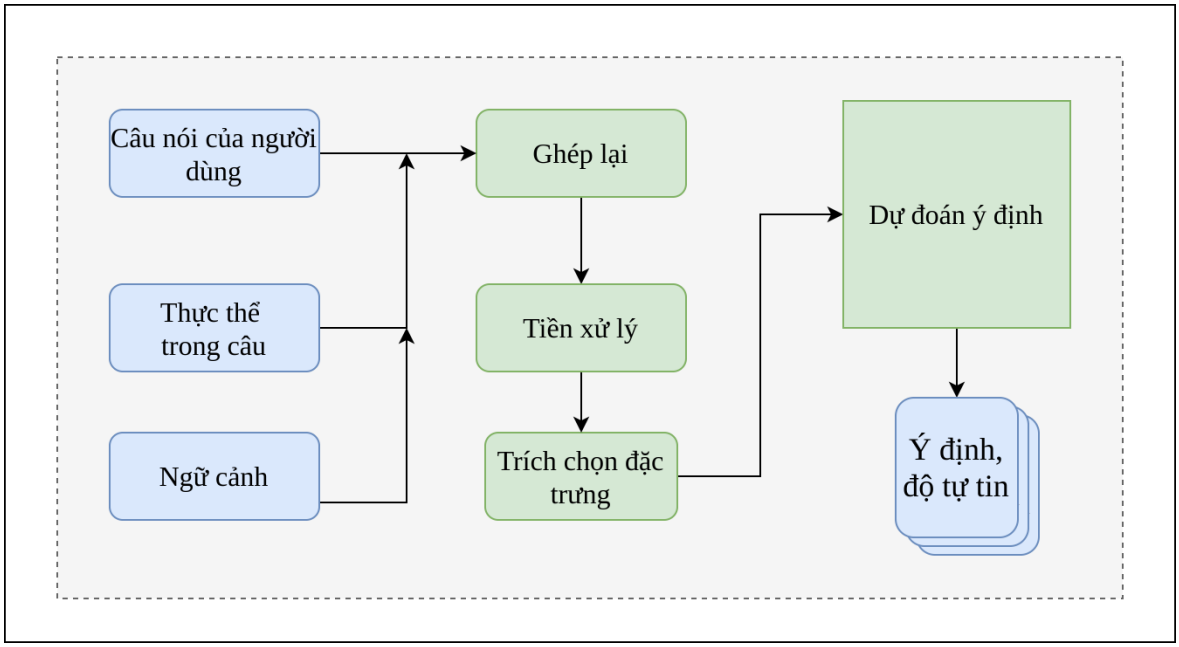
$$score = \frac{lifeSpan(z)}{lifeTime(z)}$$

Công thức 11 Công thức tính điểm cho một ngữ cảnh.

Trong đó, $score$ là điểm đánh giá ngữ cảnh, $lifeSpan(z)$ và $lifeTime(z)$ lần lượt là thời gian sống và tuổi thọ của ngữ cảnh z. Hai ngữ cảnh kích hoạt có độ điểm đánh giá cao nhất sẽ được lựa chọn để đưa vào dự đoán.

4.2.2.2 Dự đoán ý định

Sơ đồ dự đoán ý định được mô tả trong Hình 14.



Hình 14 Sơ đồ dự đoán ý định sử dụng mô hình có ngữ cảnh.

Sau khi lựa chọn được các ngữ cảnh tốt nhất, bước tiếp theo, chúng ta ghép các thực thể và ngữ cảnh vào văn bản, với tiếp đầu ngữ "context" cho ngữ cảnh và "alias" cho thực thể. Văn bản sẽ được tiền xử lý và trích rút đặc trưng để đưa vào dự đoán. Mô hình dự đoán dùng ngữ cảnh sẽ đưa ra độ tự tin cho từng ý định. Các ý định này bao gồm tất cả các ý định có ngữ cảnh đầu ra trong hệ thống. Các ý dịnh và độ tự tin tương ứng sẽ được dùng để xác định hành động dựa vào độ tự tin.

4.2.2.3 Xác định hành động dựa trên độ tự tin

Khi mô hình đưa ra kết quả dự đoán với các ý định kèm theo độ tự tin của chúng, chúng ta tiến hành dự đoán theo độ tự tin. Xét hai ngưỡng tự tin MAX_2 và MIN_2 để xác định có nhập nhằng hoặc mô hình không đoán được. Lúc này, một ý định sẽ được trả về nếu nó thỏa mãn các điều kiện: (i) có ngữ cảnh đầu vào là tập con của ngữ cảnh kích hoạt. ý định lúc này được gọi là thỏa mãn ngữ cảnh, (ii) là ý định có độ tự tin lớn nhất và (iii) có độ tự tin vượt ngưỡng MAX_2 hoặc là ý định duy nhất có độ tự tin nằm giữa hai ngưỡng MAX_2 và MIN_2.

Trong trường hợp không có ý định nào thỏa mãn cả ba điều kiện trên, mô hình sẽ kiểm tra xem có nhập nhằng xảy ra giữa các ý định có ngữ cảnh đầu ra là tập con của tập ngữ cảnh kích hoạt hay không. Trường hợp có ít nhất hai ý định nằm giữa ngưỡng MAX_2 và ngưỡng MIN_2, mô hình đưa ra một yêu cầu hỏi lại. Ngược lại, tất cả các ý định thỏa mãn ngữ cảnh đều có độ tự tin nhỏ hơn ngưỡng MIN_2, lúc này chúng ta xem xét kết quả đươc trả về ở mô hình không có ngữ cảnh để hạn chế việc hỏi lại người dùng.

Phép đoán sẽ chuyển sang bước thứ 2 nếu như trong bước 1 có nhập nhằng xảy ra hoặc mô hình không đoán được. Trường hợp mô hình không đoán được thì mô hình sẽ đưa ra yêu cầu kích hoạt ý định mặc định. Ngược lại, với các ý định thỏa mãn ngữ cảnh và rơi vào khoảng nhập nhằng, chúng ta xem xét hai trường hợp sau:

(i) Nếu chỉ có duy nhất một ý định có ngữ cảnh đầu vào, ý định được ưu tiên trả về.

(ii) Nếu có nhiều hơn một ý định có ngữ cảnh đầu vào, xét các ý định không có ngữ cảnh đầu vào. Nếu chỉ có một ý định không có ngữ cảnh đầu vào, ý định được ưu tiên trả về. Ngược lại, mô hình đưa ra hành động hỏi lại.

Kí hiệu:

y_{1i} một ý định trong hệ thống, với $i = 1, 2, \ldots, m$ và m là số ý định có trong hệ thống.

d_{1i} độ tự tin của ý định y_{1i} được trả về khi dự đoán theo mô hình không có ngữ cảnh.

MAX_1 các ngưỡng tự tin MAX được sử dụng để dự đoán theo mô hình không có ngữ cảnh.

MIN_1 các ngưỡng tự tin MAX được sử dụng để dự đoán theo mô hình không có ngữ cảnh.

y_{2i} ý định có ngữ cảnh đầu vào với $i = 1, 2, \ldots, n$ và n là số ý định.

d_{2i} độ tự tin của ý định y_{1i} được trả về bằng cách dự đoán theo mô hình có ngữ cảnh.

MAX_2 các ngưỡng tự tin MAX được sử dụng để dự đoán theo mô hình có ngữ cảnh.

MIN_2 các ngưỡng tự tin MAX được sử dụng để dự đoán theo mô hình có ngữ cảnh.

AC tập ngữ cảnh kích hoạt.

$IN(y)$ ngữ cảnh đầu vào của ý định y.

Thuật toán lựa chọn hành động trả về được sử dụng trong bước 2 như sau.

Thuật toán Xác định hành động dựa trên độ tự tin

Đầu vào: các y_{1i} và d_{1i} tương ứng, các y_{2i} và d_{2i}, MAX_1, MAX_2, MIN_1, MIN_2, AC

Đầu ra: ý định tốt nhất y^*, hành động hỏi lại ask_again, hoặc ý định mặc định y^{**}

Giải thuật:

begin
$imax \leftarrow 1$; $count\_between\_thresholds \leftarrow 1$
for $i \leftarrow 1$ **to** n **do**
  if $IN(y_i)$ not is subset of AC **then**
    continue
  if $d_i > d_{imax}$ **then**
    $imax \leftarrow i$
  endif
  if $MIN \leq d_i < MAX$ **then**
    $count\_between\_thresholds \leftarrow count\_between\_thresholds + 1$
endfor
if $d_{imax} \geq MAX$ *or* $(d_{imax} < MAX$ *and* $coun\_between\_thresholds = 1)$ **then**
  $y^* \leftarrow y_{imax}$
  return y^*
else if $d_{imax} < MAX$ *and* $coun\_between\_thresholds > 1$ **then**
  return ask_again
else
  $imax' \leftarrow 1$; $intent\_has\_context \leftarrow [\,]$; $intent\_no\_context \leftarrow [\,]$
  for $i \leftarrow 1$ **to** m **do**
    if $IN(y_{2i})$ not is subset of AC **then**
      continue
    if $d_{1i} > d_{imax}$ **then**
      $imax' \leftarrow i$
    endif
    if $MIN_1 \leq d_{1i} < MAX\_1$ **then**
      if $IC(y_{1i})$ is subset of AC then add y_i to $intent\_has\_context$
    else add y_i to $intent\_no\_context$
  if $y_{1imax'} < MIN\_1$ **then**
    return y^{**}
  else if $len(intent\_has\_context) = 1$ **then**
    $y^* = intent\_has\_context[0]$
    return y^*
  else if $len(intent\_has\_context) = 1$ **then**
    $y^* = intent\_no\_context[0]$
    return y^*
  else return ask_again
endif
end

4.3 Thực nghiệm và đánh giá

Trong phần này, em tiến hành thử nghiệm và đánh giá mô hình đã đề xuất. Mô hình đề xuất dựa trên mô hình dự đoán trình bày ở chương 3, do đó, thuật toán phân loại tốt ở chươn 3 sẽ có sự phân cách tốt giữa các dự đoán đúng và sai. Vì vậy, thuật toán phân loại được sử dụng là SVM và phép trích rút đặc trưng sử dụng trong phần này là IF-IDF. Ngưỡng MAX_1 và MAX_2 được lựa chọn là 0.6 và MIN_1 , MIN_2 là 0.2. Các ngưỡng này được em lựa chọn dựa trên thực nghiệm.

Thử nghiệm có 2 mục đích chính: đánh giá hiệu quả của mô hình dự đoán hai bước với các dữ liệu gây nhập nhằng và đánh giá ảnh hưởng của mô hình với các dữ liệu đầu đủ thông tin.

4.3.1 Dữ liệu

Tập dữ liệu này được mở rộng và bổ sung từ tập dữ liệu thực nghiệm trong Bảng 4. Chi tiết về tập dữ liệu được cho trong Bảng 11.

Bảng 11

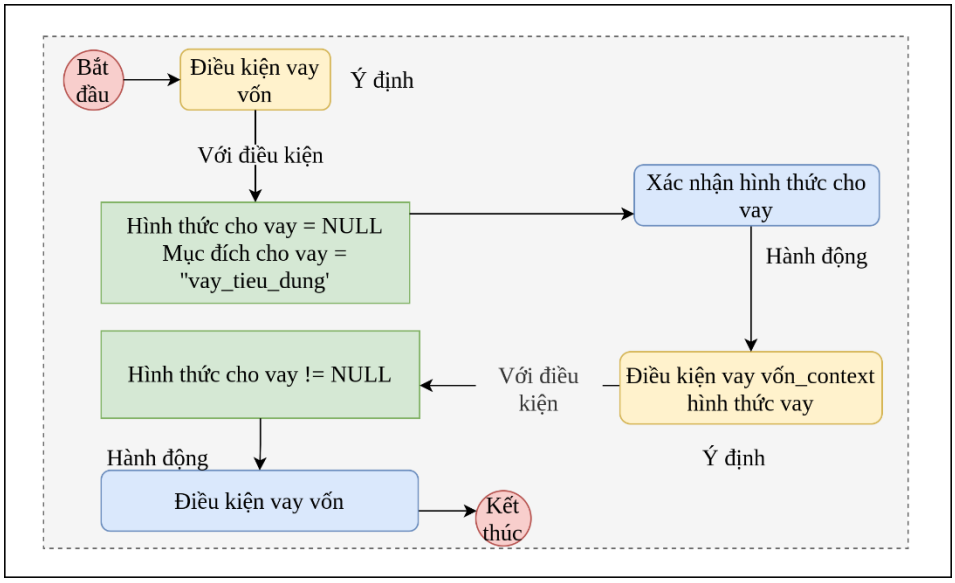
Bảng 11 Thông tin về bộ dữ liệu ngân hàng có ngữ cảnh

Số lượng ý định	**214**
Số lượng ý định có input context	**86**
Số lượng context	**36**
Số lượng thực thể	18
Số lượng câu	24.301
Số câu ít nhất trong một ý định	1
Số câu nhiều nhất trong một ý định	1.140
Số câu trung bình trong một ý định	113

Bộ dữ liệu trên bao gồm 214 ý định với tổng số 24.301 câu mẫu, trong đó có 36 ý định có input context, 36 context và 14 thực thể. Số câu nhiều nhất trong một ý định là 1.140 câu và số câu ít nhất là 1 câu. Trung binh mỗi ý định chứa 113 câu mẫu. Số ý định tăng lên do chúng

ta cần thiết kế các ý định tương ứng với các ngữ cảnh khác nhau. Tương tự như bộ dữ liệu sử dụng trong chương trước, bộ dữ liệu này cũng gặp phải vấn đề mất cân bằng và được xử lý bằng phương pháp oversampling.

Ngoài ra, trong thí nghiệm này, em sử dụng một tập luồng hội thoại để xây dựng các ngữ cảnh kích hoạt cho từng ý định. Tập các ngữ cảnh kích hoạt được ghép đôi một với từng câu nói của người dùng trong tập kiểm thử để xây dựng bộ dữ liệu kiểm thử. Tập dữ liệu luồng hội thoại bao gồm các hội thoại được xây dựng từ các lượt ý định – hành động của hệ thống, bao gồm 220 hội thoại. Hình 15 mô tả ví dụ về một luồng hội thoại:



Hình 15 Ví dụ một luồng hội thoại.

Trong đó, người dùng muốn hỏi về điều kiện vay vốn và cung cấp thông tin về mục đích cho vay là "vay_tiêu_dùng". Để có thể trả lời người dùng tiếp theo, hệ thống cần xác định hình thức cho vay qua hành động "Xác nhận hình thức cho vay". Người dùng trả lời tươn ứng với ý định "Điều kiện vay vốn_context hình thức cho vay". Sau khi xác nhận được hình thức vay, hệ thống trả lại thông tin cho người dùng. Dựa vào sơ đồ này, chúng ta có thể xác định được ngữ cảnh hội thoại hay tập các ngữ cảnh kích hoạt tại từng lượt hội thoại.

4.3.2 Kết quả và đánh giá

Mô hình dự đoán ý định được đánh giá trên hai loại dữ liệu khác nhau. Loại dữ liệu thứ nhất bao gồm dữ liệu có ngữ cảnh kích hoạt và ý định đúng của nó là ý định không có ngữ cảnh vào. Loại dữ liệu thứ hai bao gồm các dữ liệu có ngữ cảnh kích hoạt và ý định đúng của nó có ngữ cảnh vào. Thử nghiệm trên bộ dữ liệu thứ nhất giúp chúng ta xác định xem mô hình dự đoán hai bước có hoạt động hiệu quả như mô hình một bước hay không. Với loại dữ liệu thứ hai, chúng ta đánh giá xem liệu mô hình có dự đoán được những câu nói dễ gây nhập nhằng hay không. Bảng 12 mô tả kết quả thực nghiệm với bộ dữ liệu đầu tiên.

Bảng 12 Kết quả thử nghiệm trên loại dữ liệu không có ngữ cảnh kích hoạt của ý định không có ngữ cảnh vào

	Mô hình không sử dụng ngữ cảnh	Kết hợp hai mô hình
Độ chính xác (%)	**87,50**	**87,88**
Tỷ lệ hỏi lại (%)	1,14	0,51
Tỷ lệ trả lời mặc định (%)	8,26	8,97

Bảng 12 cho thấy việc kết hợp hai mô hình vẫn có khả năng dự đoán như mô hình một bước. Cụ thể mô hình đạt độ chính xác 87,88%, tỷ lệ hỏi lại là 0,51% và tỷ lệ ý định trả về rơi vào mặc định là 8,97% trên bộ dữ liệu của các ý định không có ngữ cảnh vào và không có ngữ cảnh kích hoạt tại thời điểm đoán. Kết quả này xấp xỉ so với mô hình trước khi sử dụng ngữ cảnh, trong đó mô hình trước khi sử dụng ngữ cảnh đạt độ chính xác là 87,50% với tỷ lệ hỏi lại và tỷ lệ ý định trả về rơi vào mặc định là 8,26%.

Bảng 13 mô tả kết quả thực nghiệm với bộ dữ liệu có ngữ cảnh kích hoạt của với các ý định đúng có ngữ cảnh vào.

Bảng 13 Kết quả thử nghiệm trên loại dữ liệu có ngữ cảnh kích hoạt với ý định có ngữ cảnh đầu vào

	Mô hình không sử dụng ngữ cảnh	**Kết hợp hai mô hình**
Độ chính xác (%)	**3,27**	**80,03**
Tỷ lệ hỏi lại (%)	19,07	0,00
Tỷ lệ trả lời mặc định (%)	31,47	7,58

Kết quả thử nghiệm cho thấy, mô hình dự đoán hai bước đã có thể xử lý tốt các trường hợp nhập nhằng với độ chính xác lên đến 80,03% trong khi mô hình không sử dụn ngữ cảnh xử lý rất tệ trong trường hợp này, chỉ đoán đúng 3,27% và tỷ lệ hỏi lại lên đến 19,07%, tỷ lệ trả lời mặc định là 31,47%. Như vậy, qua kết quả thử nghiệm trên hai loại dữ liệu khác nhau, ta có thể thấy việc kết hợp đoán 2 lần đạt độ chính xác xấp xỉ 80%. Mô hình đã giải quyết được các trường hợp nhập nhằng trong khi vẫn đảm bảo tỷ lệ dự đoán tốt trên những dữ liệu có đầy đủ thông tin.

Chương 5 Kết luận và hướng phát triển

5.1 Kết luận

Trong đồ án này, em đã đưa ra hai mô hình phân loại ý định người dùng trong trường hợp không sử dụng ngữ cảnh và có sử dụng ngữ cảnh để giảm thiểu rủi ro trong các trường hợp đoán sai ý định.

Với mô hình phân loại ý định người dùng không có ngữ cảnh, em sử dụng hai ngưỡng MAX và MIN để lựa chọn ý định được trả về, từ đó giúp hệ thống đưa ra phản hồi thích hợp. Cụ thể, nếu ý định có độ tự tin lớn nhất vượt ngưỡng MAX hoặc là ý định duy nhất có độ tự tin nằm giữa hai ngưỡng thì ý định sẽ được trả về. Nếu không có ý định nào có độ tự tin vượt qua ngưỡng MIN, ý định mặc định sẽ được kích hoạt. Ngược lại, hệ thống sẽ tiến hành hỏi lại để xác nhận lại ý định của người dùng. Để lựa chọn thuật toán phân loại và các ngưỡng phù hợp, em đưa ra tiêu chí đánh giá dựa trên độ tự tin là khoảng chênh lệnh giữa độ tự tin của các câu phán đoán đúng và sai lớn. Bên cạnh đó, em còn đánh giá thuật toán phân loại dựa trên tiêu chí về thời gian dự đoán và độ chính xác để lựa chọn ra mô hình phù hợp nhất. Kết quả thực nghiệm cho thấy, các mô hình học sâu tuy có độ chính xác rất cao nhưng lại không thỏa mãn điều kiên về độ tự tin. Các thuật toán hoc máy như SVM, Random Forest cho độ chính xác cao đồng thời có khả năng tách biệt giữa các dự đoán đúng và sai. Do vậy, các thuật toán này có thể giúp hệ thống hội thoại đưa ra cách hành xử hợp lý tùy theo độ tự tin. Mô hình này có thể được sử dụng trong các hội thoại ngắn, các câu hỏi FAQ mà không yêu cầu dùng ngữ cảnh.

Với việc lấy ý tưởng xây dựng các ngữ cảnh tự định nghĩa từ Google Dialogflow, trong mô hình sử dụng ngữ cảnh sử dụng để thực hiện dự đoán ý đinh, em tiến hành xây dựng hai mô hình dự đoán: mô hình không dùng ngữ cảnh và mô hình có dùng dử cảnh, sau đó tiến hành dự đoán theo hai bước. Bước đầu tiên không sử dụng ngữ cảnh hoạt động giống như trong mô hình không có ngữ cảnh. Tuy nhiên, nếu hệ thống phát hiện có sự nhập nhằng hoặc không thể dự đoán được, ngữ cảnh sẽ được đưa vào để phán đoán. Bước 2 cũng sử dụng hai ngưỡng MAX' và MIN' để xác định ý định được trả về, phát hiện nhập nhằng hoặc không đự đoán được. Trong trường hợp mô hình 2 không dự đoán được, em tiến hành lựa chọn kết quả dự đoán theo mô hình không có ngữ cảnh, có ưu tiên các ngữ cảnh có độ tương đồng gần với ngữ cảnh hội thoại nhất. Kết quả thử nghiệm cho thấy mô hình cho tốt trên các trường hợp

câu nói không rõ ràng hoặc dễ gây nhập nhằng với độ chính xác 80,03% và hoạt động ổn định như với mô hình dự đoán một bước với độ chính xác 87,88%.

Kết quả của nghiên cứu mở ra hai hướng đi trong việc quản lý hệ thống hội thoại. Hướng thứ nhất không sử dụng ngữ cảnh và dùng các chiến lược khác nhau dựa trên độ tự tin để có thể xác định đúng ý muốn của người dùng. Hướng đi thứ hai sử dụng các ngữ cảnh tự định nghĩa, kết hợp với các chiến lược về độ tự tin giúp hệ thống giảm đi số trường hợp hỏi lại người dùng để xác định mong muốn của họ. Điều này mở ra cơ hội xây dựng hệ thống hội thoại thông minh hơn và có khả năng đáp ứng tốt hơn.

Tuy nhiên, việc sử dụng ngữ cảnh tự định nghĩa làm cho số lượng ý định chứa ngữ cảnh vào trở lên bùng nổ, làm tăng lượng dữ liệu đào tạo. Việc này có thể khiến cho thời gian kéo dài hơn, làm cho nhà phát triển cảm thấy không hài lòng.

Hiện tại, mô hình đang được triển khai và thử nghiệm trên nền tảng Smart Dialog – một nền tảng cho phép xây dựng các hệ thống hội thoại thông minh.

5.2 Hướng phát triển

Sau khi đồ án hoàn thành, em sẽ tiếp tục nghiên cứu cải thiện mô hình để giải thời gian đào tạo. Ngoài ra, các thực thể trong câu cũng nằm giữ những thống tin quan trọng, vì thế, các thực thể này cũng có thể được sử dụng làm ngữ cảnh trong hệ thống. Các thực thể trong các tin nhắn khác nhau có thể khác nhau, do vậy, chúng ta có thể kích hoạt ngữ cảnh dựa trên tưng tin nhắn thay vì trên một ý định.

Mô hình hiện tại cũng chưa khai thác được mối quan hệ giữa các ý định được kết nối với nhau theo ngữ cảnh hay phân cấp các ý định. Trong thời gian tới, em hướng đến việc khai thác thông tin này để có thể giúp mô hình tối ưu hơn và đạt độ chính xác cao hơn.

Ngoài ra, chưa có một công thức nào để xác định xem cần chọn bao nhiêu ngữ cảnh tại thời điểm đoán và làm sao để lựa chọn ngữ cảnh một cách tối ưu nhất. Đây cũng là vấn đề mà đồ án hướng tới phát triển trong tương lai.

Tài liệu tham khảo

[1] C. Cortes và V. Vapnik, "Support-vector networks", *Mach. Learn.*, vol 20, số p.h 3, tr 273–297, tháng 9 1995, doi: 10.1007/BF00994018.

[2] "SmartDialog". https://smartdialog.vn/ (truy cập tháng 1 08, 2021).

[3] S. Arora, K. Batra, và S. Singh, "Dialogue System: A Brief Review", tr 4.

[4] "Siri", *Apple (CA)*. https://www.apple.com/ca/siri/ (truy cập tháng 12 29, 2020).

[5] "Google Assistant, your own personal Google". https://assistant.google.com/ (truy cập tháng 1 06, 2021).

[6] "Dialogflow | Google Cloud". https://cloud.google.com/dialogflow (truy cập tháng 12 29, 2020).

[7] "Open source conversational AI", *Rasa*, tháng 12 01, 2020. https://rasa.com/ (truy cập tháng 12 29, 2020).

[8] "Build AI Powered Chatbots". https://www.oracle.com/chatbots/ (truy cập tháng 1 07, 2021).

[9] "Intent matching | Dialogflow ES", *Google Cloud*. https://cloud.google.com/dialogflow/es/docs/intents-matching (truy cập tháng 1 07, 2021).

[10] "Fallback Actions". https://legacy-docs.rasa.com/docs/core/fallbacks (truy cập tháng 1 07, 2021).

[11] P. Keegan, C. Kutler, và J. Bassett, "Intents", *Oracle Help Center*. https://docs.oracle.com/en/cloud/paas/digital-assistant/use-chatbot/intents1.html#GUID-F380AD2B-9A7B-4422-9B3A-00E6C8D3D092 (truy cập tháng 1 07, 2021).

[12] W. Zhang và F. Gao, "An Improvement to Naive Bayes for Text Classification", *Procedia Eng.*, vol 15, tr 2160–2164, tháng 12 2011, doi: 10.1016/j.proeng.2011.08.404.

[13] "BoosTexter: A Boosting-based System for Text Categorization | SpringerLink". https://link.springer.com/article/10.1023/A:1007649029923 (truy cập tháng 1 05, 2021).

[14] L. Breiman, "Random Forests", *Mach. Learn.*, vol 45, số p.h 1, tr 5–32, tháng 10 2001, doi: 10.1023/A:1010933404324.

[15] "(PDF) Optimizing Svms For Complex Call Classification". https://www.researchgate.net/publication/2937139_Optimizing_Svms_For_Complex_Call_Classification (truy cập tháng 1 05, 2021).

[16] A. Genkin, D. D. Lewis, và D. Madigan, "Large-Scale Bayesian Logistic Regression for Text Categorization", *Technometrics*, vol 49, số p.h 3, tr 291–304, tháng 8 2007, doi: 10.1198/004017007000000245.

[17] H. B. Hashemi, A. Asiaee, và R. Kraft, "Query Intent Detection using Convolutional Neural Networks", tr 5.

[18] S. Ravuri và A. Stolcke, "Recurrent Neural Network and LSTM Models for Lexical Utterance Classification", tr 5.

[19] A. Graves và J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Netw.*, vol 18, số p.h 5–6, tr 602–610, tháng 7 2005, doi: 10.1016/j.neunet.2005.06.042.

[20] B. Liu và I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling", *ArXiv Prepr. ArXiv160901454*, 2016.

[21] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, và Z. Zhao, "Investigating capsule networks with dynamic routing for text classification", *ArXiv Prepr. ArXiv180400538*, 2018.

[22] M. Mensio và G. Rizzo, *Multi-turn QA: A RNN Contextual Approach to Intent Classification for Goal-oriented Systems*. 2018, tr 1080.

[23] A. Sharma, "Improving Intent Classification in an E-commerce Voice Assistant by Using Inter-Utterance Context", trong *Proceedings of The 3rd Workshop on e-Commerce and NLP*, Seattle, WA, USA, tháng 7 2020, tr 40–45, doi: 10.18653/v1/2020.ecnlp-1.6.

[24] A. Gupta, P. Zhang, G. Lalwani, và M. Diab, "CASA-NLU: Context-Aware Self-Attentive Natural Language Understanding for Task-Oriented Chatbots", *ArXiv190908705 Cs*, tháng 9 2019, Truy cập: tháng 12 29, 2020. [Online]. Available at: http://arxiv.org/abs/1909.08705.

[25] "Contexts | Dialogflow ES | Google Cloud". https://cloud.google.com/dialogflow/es/docs/contexts-overview (truy cập tháng 12 29, 2020).

[26] Z. Zhang, R. Takanobu, Q. Zhu, M. Huang, và X. Zhu, "Recent Advances and Challenges in Task-oriented Dialog System", *ArXiv200307490 Cs*, tháng 6 2020, Truy cập: tháng 12 30, 2020. [Online]. Available at: http://arxiv.org/abs/2003.07490.

[27] J. R. Quinlan, "Induction of decision trees", *Mach. Learn.*, vol 1, số p.h 1, tr 81–106, 1986.

[28] V. H. Tiệp, "Machine learning cơ bản", *Nhà Xuất Bản Khoa Học Và Kỹ Thuật*, 2018.

[29] P. Ramachandran, B. Zoph, và Q. V. Le, "Searching for activation functions", *ArXiv Prepr. ArXiv171005941*, 2017.

[30] S. Hochreiter và J. Schmidhuber, “Long Short-term Memory”, *Neural Comput.*, vol 9, tr 1735–80, tháng 12 1997, doi: 10.1162/neco.1997.9.8.1735.

[31] A. Vaswani *và c.s.*, “Attention is All you Need”, tr 11.

[32] “[1409.0473] Neural Machine Translation by Jointly Learning to Align and Translate”. https://arxiv.org/abs/1409.0473 (truy cập tháng 1 10, 2021).

[33] A. Aizawa, “An information-theoretic perspective of tf–idf measures”, *Inf. Process. Manag.*, vol 39, số p.h 1, tr 45–65, 2003.

[34] T. Mikolov, K. Chen, G. Corrado, và J. Dean, “Efficient Estimation of Word Representations in Vector Space”, *ArXiv13013781 Cs*, tháng 9 2013, Truy cập: tháng 1 06, 2021. [Online]. Available at: http://arxiv.org/abs/1301.3781.

[35] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, và A. Joulin, “Advances in Pre-Training Distributed Word Representations”, *ArXiv171209405 Cs*, tháng 12 2017, Truy cập: tháng 1 06, 2021. [Online]. Available at: http://arxiv.org/abs/1712.09405.