ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hiểu ngôn ngữ dạng nói tiếng Việt

TRẦN ĐĂNG TUYẾN
tuyen.td194407@sis.hust.edu.vn

Ngành: Kỹ thuật máy tính

Giảng viên hướng dẫn: TS. Trần Hoàng Hải
TS. Nguyễn Thị Thu Trang

Khoa: Kỹ thuật máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2023

LỜI CAM KẾT

Họ và tên sinh viên: Trần Đăng Tuyến
Điện thoại liên lạc: 0362468233
Email: tuyen.td194407@sis.hust.edu.vn
Lớp: KTMT03-K64
Hệ đào tạo: Cử nhân

Tôi – *Trần Đăng Tuyến* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Ts. Trần Hoàng Hải và 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 tháng năm

Tác giả ĐATN

Trần Đăng Tuyến

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất tới TS. Trần Hoàng Hải và TS. Nguyễn Thị Thu Trang vì sự tận tâm và hỗ trợ mà thầy và cô đã dành cho em trong suốt thời gian qua. Hai thầy cô không chỉ là những giảng viên hướng dẫn mang đến cho em những kiến thức chuyên môn cùng các định hướng chính xác, mà còn là những người bạn đồng hành, là nguồn cảm hứng quan trọng của em trong quá trình thực hiện đồ án tốt nghiệp và nghiên cứu tại Lab. Với tất cả sự quan tâm và kinh nghiệm của mình, 2 thầy cô đã giúp em định hướng đề tài phù hợp với bản thân và đồng thời luôn sát cánh bên em, cho em những chỉ dẫn chính xác để giúp em hoàn thành được đồ án này.

Em xin gửi lời cảm ơn chân thành tới các thầy cô tại Đại học Bách Khoa Hà Nội và Trường công nghệ thông tin và truyền thông - SoICT, những người đã dành thời gian và tâm huyết để truyền dạy cho em những kiến thức quý báu trong suốt quá trình học tập 4 năm tại trường.

Em cũng xin gửi lời cảm ơn sâu sắc đến các anh chị, bạn bè và các em trong lab nghiên cứu 914. Cảm ơn mọi người đã luôn đồng hành, giúp đỡ và động viên em trong suốt quá trình em tham gia nghiên cứu tại lab và thực hiện đồ án.

Cuối cùng em xin cảm ơn toàn thể mọi người trong gia đình của mình vì đã luôn là nguồn động lực và ủng hộ em về cả tài chính lẫn tinh thần để em có thể vượt qua những khó khăn và thử thách của bản thân suốt những chặng đường học hành cho đến hiện tại.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Sự phát triển của công nghệ trí tuệ nhân tạo đem đến cho các hệ thống, máy móc khả năng có thể giao tiếp và hiểu được các yêu cầu của con người thông qua nhiều phương tiện, một trong số đó là sử dụng giọng nói. Vì vậy bài toán hiểu ngôn ngữ tự nhiên dạng nói (Spoken language understanding) đang được các nhà nghiên cứu trên thế giới đặc biệt quan tâm và phát triển. Đối với bài toán SLU có hiện có hai hướng giải quyết chính, thứ nhất là sử dụng hệ thống tích hợp hai mô hình Automatic speech recognition (ASR) và Natural language understanding (NLU) tuy nhiên việc sử dụng mô hình tích hợp này lại yêu cầu tiêu tốn nhiều tài nguyên, vì vậy xu hướng hiện nay tập trung vào hướng giải quyết thứ hai là mô hình đầu cuối SLU (End-to-end SLU) tiêu tốn ít tài nguyên hơn nhưng vẫn đem lại hiệu quả tương đương.

Hiện nay các nghiên cứu về các mô hình và bộ dữ liệu cho bài toán SLU đối với tiếng Việt là rất ít và không có nhiều đóng góp. Vì thế với đồ án này tác giả sẽ xây dựng một bộ dữ liệu SLU cho tiếng Việt là VN-SLU với quy trình bao gồm bốn bước từ chuẩn bị dữ liệu, sinh dữ liệu, thu thập dữ liệu đến xử lý dữ liệu. Bộ dữ liệu là tập hợp các yêu cầu, tương tác của người dùng với thiết bị trong nhà thông minh.

Với sự tham gia thu âm của 130 người, bộ dữ liệu VN-SLU thu được bao gồm 10000 câu nói với tổng dung lượng là 12 giờ, 9 ý định và 7 loại thực thể khác nhau.

Ngoài xây dựng bộ dữ liệu, tác giả cũng thực nghiệm và đưa ra một só cải tiến cho mô hình End-to-end SLU trên chính bộ dữ liệu VN-SLU. Trong các ngôn ngữ thanh điệu (tonal language) như tiếng Việt, độ cao (pitch) khi phát âm có ảnh hưởng lớn tới ý nghĩa của câu nói, trong mô hình cải tiến tác giả tích hợp thêm khối trích xuất độ cao của âm thanh nhằm bổ sung thêm thông tin cho quá trình giải mã của mô hình. Ngoài ra một phương pháp trích xuất thông tin và nén âm thanh dưới dạng các vec-tơ đặc trưng nhưng lại không làm mất nhiều thông tin quan trọng của tiếng nói cũng được tích hợp trong mô hình cải tiến này.

Kết quả thực nghiệm mô hình cải tiến trên bộ dữ liệu tiếng Việt đạt kết quả tốt với chỉ số Intent Accuracy đạt 93.33% cho nhiệm vụ phân loại ý định và chỉ số SLU-f1 đạt 71.32% cho nghiệm vụ điền khung ngữ nghĩa. Kết quả này có mức cải thiện đáng kể khi so với kết quả mà mô hình cơ sở đạt được với mức tăng 3.81% và Intent Accuracy và 2.94% SLU-F1.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
AM	Mô hình ngữ âm (ACOUSTIC MODEL)
ANN	Mạng trí tuệ nhân tạo (Artificial Neural Networks)
ASR	Nhận dạng tiếng nói tự động (Automatic speech recognition)
CNN	Mạng tích chập (CONVOLUTION NEURAL NETWORK)
CTC	Phân lớp thời gian kết nối (Connectionist Temporal Classification)
FFNN	Mạng lan truyền tiến (Feed forward Neural Networks)
GPT	Generative Pre-trained Transformer
GPU	Đơn vị xử lý đồ họa (Graphics Processing Unit)
GRU	Đơn vị cổng tái phát (Gated Recurrent Unit)
LSTM	Bộ nhớ dài-ngắn hạn (LONG-SHORT TERM MEMORY)
MFCCs	Hệ số Mel-Frequency Cepstral (Mel-frequency cepstral coefficients)
NLP	Xử lý ngôn ngữ tự nhiên (NATURAL LANGUAGE PROCESSING)
NLU	Hiểu ngôn ngữ tự nhiên (NATURAL LANGUAGE UNDERSTANDING)
RNN	Mạng tái phát (RECURRENT NEURAL NETWORK)
Seq2Seq	Mô hình chuỗi-chuỗi (Sequence-to-sequence)
SLU	Hiểu ngôn ngữ tự nhiên dạng nói (SPOKEN LANGUAGE UNDERSTANDING)

Thuật ngữ	**Ý nghĩa**
STT	Chuyển tiếng nói thành văn bản (Speech to text)

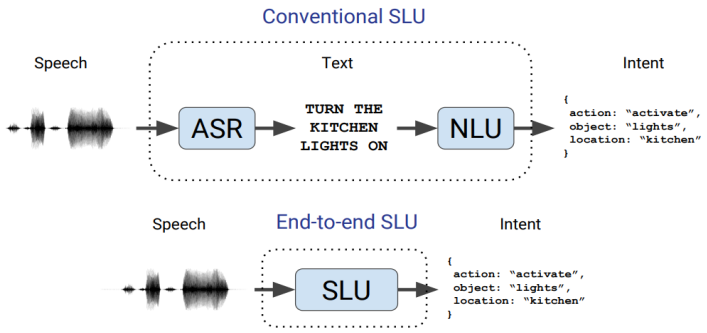
CHƯƠNG 1. BÀI TOÁN HIỂU NGÔN NGỮ TỰ NHIÊN DẠNG NÓI

1.1 Bài toán hiểu ngôn ngữ tự nhiên dạng nói

Trong cuộc sống hiện đại ngày nay với sự phát triển mạnh mẽ của lĩnh vực trí tuệ nhân tạo, ngày càng có nhiều công việc sử dụng trí tuệ nhân tạo để thay thế con người. Một trong số các lĩnh vực đang cực kì nổi bật hiện nay là về xử lý ngôn ngữ tự nhiên. Các chương trình máy tính được huấn luyện để có khả năng giao tiếp với con người với mục tiêu có thể hỗ trợ một phần hoặc có thể thay thế con người trong nhiều lĩnh vực. Hiện nay người dùng có thể dễ dàng sử dụng và tiếp cận với hệ thống giao tiếp điển hình là các trợ lý ảo thông minh trên các thiết bị đa phương tiện như trợ lý ảo Siri của Apple, Google Assistant của Google hay Amazon Alexa của Amazon,... Ngoài ra còn một số hệ thống hỗ trợ và chăm sóc khách hàng của nhiều công ty cũng dần được thay thế bởi các hệ thống trợ lý ảo, callbot để có thể giải phóng bớt sức lao động của con người, điều này đem lại lợi ích cực kì lớn cho các công ty, doanh nghiệp phải xử lý và giải đáp lượng lớn phản hồi từ khách hàng. Với những ứng dụng to lớn và có thể áp dụng trong nhiều lĩnh vực của cuộc sống, việc nghiên cứu và phát triển các chương trình máy tính giao tiếp hiệu quả được với con người đang được các công ty, tập đoàn và các nhà nghiên cứu quan tâm, đầu tư phát triển.

Hiện nay việc giao tiếp giữa người và máy có hai hình thức chính là thông qua văn bản hoặc lời nói. Đối với giao tiếp bằng văn bản, chương trình cần phải hiểu và xử lý câu nói của con người dưới dạng văn bản, để đáp ứng có yêu cầu này con người đã phát triển thành phần Hiểu ngôn ngữ tự nhiên dạng văn bản (Natural language understanding - NLU). Trong hệ thống của NLU, mục tiêu của hệ thống là phải hiểu được ý nghĩa hay ý định của câu nói người dùng cung cấp thông qua văn bản từ đó có thể nắm bắt và phân tích ý định của người dùng và đưa ra các phản hồi tương ứng. Các hệ thống NLU đã được phát triển khá lâu và hiện đã có nhiều hệ thống phát triển có thể nắm bắt được ý định lời nói của con người qua văn bản một cách chính xác cao như là Google Dialogflow, IBM Watson Assistant, RASA... Trong nhiều lĩnh vực, việc trao đổi giao tiếp với máy qua văn bản diễn ra phức tạp và cần nhiều thao tác hơn đối với con người vì vậy hệ thống tương tự NLU nhưng thay vì giao tiếp bằng văn bản thì người và máy sẽ giao tiếp với nhau bằng giọng nói, tương tự như việc nói chuyện giữa người với người và hệ thống này được gọi là Hiểu ngôn ngữ tự nhiên dạng nói (Spoken language understanding - SLU). Đối với hệ thống SLU, người dùng sẽ đưa ra lời nói và hệ thống sẽ phải xác định ý định hoặc mục tiêu của người dùng thông qua lời nói đó. Có hai nhiệm vụ chính

mà hệ thống SLU cần giải quyết là: Hiểu ý định của câu (intent understanding) và nhận dạng các thực thể trong câu (Entity recognition). Ví dụ: người dùng nói câu: "Tôi muốn bật cái điều hòa ở mức nhiệt 26 độ C", hệ thống sẽ nhận biết ý định của câu là "Bật điều hòa nhiệt độ" và nhận dạng các thực thể trong câu bao gồm "hành động" là "bật", "thiết bị" là "điều hòa", "mức nhiệt độ" là "26 độ C". Từ các thông tin phân tích được, có thể cung cấp cho các hệ thống, thiết bị tương ứng thực hiện yêu cầu của người dùng. Bài toán đặt ra cho con người là làm sao để



Hình 1.1: Hướng giải quyết bài toán SLU [1]

có một hệ thống SLU chính xác, hiệu quả. Hiện nay có 2 hướng chính giải quyết bài toán SLU là: thứ nhất là mô hình hiểu ngôn ngữ tự nhiên dạng nói truyền thống (Conventional SLU) là mô hình dạng chuỗi các mô hình nhỏ hơn để giải quyết 2 bài toán nhỏ là Chuyển đổi giọng nói thành văn bản (STT - Speech to text) và Hiểu ngôn ngữ tự nhiên (NLU - Natural language processing), mô hình được minh họa trong phần phía trên của hình 1.1. Trong bài toán Chuyển đổi giọng nói thành văn bản đầu vào của mô hình sẽ là một đoạn âm thanh và đầu ra là phiên âm của đoạn âm thanh đó, phần văn bản được sinh ra sẽ làm đầu vào cho bài toán hiểu ngôn ngữ tự nhiên để phán đoán ý định của câu nói và nhận dạng các thực thể trong câu. Trong Phương pháp này nhờ việc đã đạt những kết quả cực tốt ở hai bài toán nhỏ nên các mô hình SLU dạng này cũng đạt được những kết quả khá tốt trên các bộ dữ liệu SLU phổ biến. Tuy nhiên, với việc sử dụng 2 mô hình nhỏ phục vụ 2 nhiệm vụ khác nhau sẽ gây tốn tài nguyên huấn luyện khá lớn như thời gian huấn luyện lâu, lượng dữ liệu lớn,... và để khắc phục tình trạng này thì cách giải quyết thứ 2 cho bài toán SLU được phát triển đó là sử dụng mô hình đầu cuối (End-to-end SLU) cho bài toán hiểu ngôn ngữ tự nhiên dạng nói, với đầu vào của mô hình là đoạn âm thanh giọng nói và được một mô hình duy nhất xử lý âm thanh đầu vào để cho ra được đầu ra là ý định của đoạn âm thanh và các thực thể xuất hiện trong đó. Mô hình End-to-end SLU làm tăng tốc độ huấn luyện mô hình và cần ít dữ liệu hơn so với mô hình Conventional SLU mà vẫn có thể đạt được kết quả tốt tương đương.

1.2 Các nghiên cứu về hiểu ngôn ngữ tự nhiên dạng nói

1.2.1 Mô hình hiểu ngôn ngữ tự nhiên dạng nói truyền thống

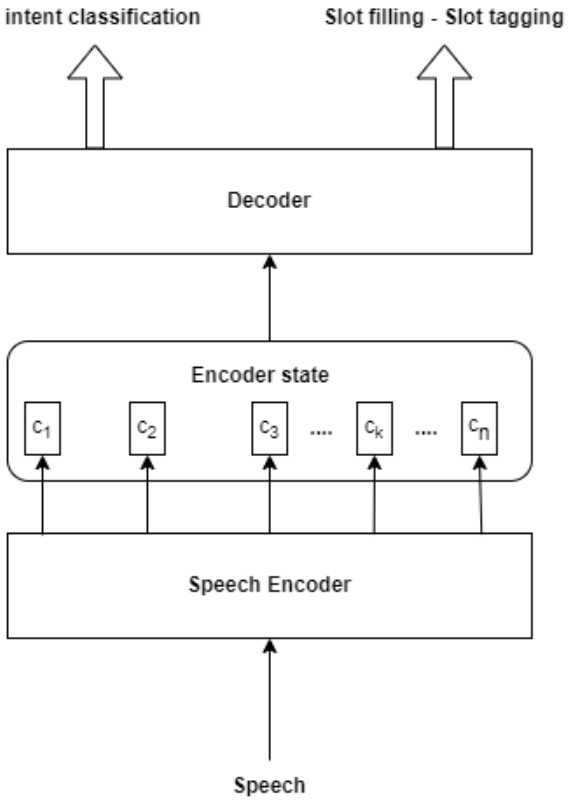
Như đã đề cập ở phần 1.1 mô hình hiểu ngôn ngữ tự nhiên dạng nói truyền thống sẽ có 2 phần, thứ nhất là mô hình hình chuyển giọng nói thành văn bản (STT) và thứ hai là mô hình hiểu ngôn ngữ tự nhiên dạng văn bản (NLU). Đối với mô hình STT khi áp dụng trong bài toán SLU các nhà nghiên cứu thường sử dụng mô hình nhận dạng giọng nói tự động (Automatic speech recognition - ASR) là một nhánh của mô hình STT. Mô hình ASR tập trung vào việc nhận dạng giọng nói con người và xử lý chuyển giọng nói thành văn bản. Hệ thống ASR gồm 4 thành phần chính là: đầu tiên là phần trích xuất đặc trưng (Feature extraction), khối này thực hiện việc chuyển đổi âm thanh thô ban đầu sang dạng một chuỗi các vec-tơ biểu thị các đặc trưng của âm thanh, phương pháp thường được sử dụng để trích xuất đặc trưng là Mel-frequency cepstral coefficients (MFCCs),... Thành phần thứ 2 là mô hình âm học (Acoustic model - AM), mô hình này đóng vai trò tính xác suất các vector đặc trưng với các đơn vị âm vị (như là âm, âm vị,..) hoặc các thành phần của từ (subword) phù hợp. Acoustic model thường có thể là mô hình mạng nơ ron hồi quy (Recurrent neural network - RNN) như Long Short-Term Memory (LSTM) [2] hay Gated Recurrent Unit (GRU) [3], acoustic model cũng có thể là các mô hình xây dựng dựa trên mang nơ ron tích chập (Convolutional Neural Network - CNN). Phần thứ 3 là Mô hình ngôn ngữ (Language model - LM), mô hình này cung cấp xác suất của các từ và cụm từ trong ngôn ngữ được huấn luyện, mô hình ngôn ngữ phổ biến được sử dụng hiện nay như là GPT-3 (Generative Pre-trained Transformer 3) [4], BERT (Bidirectional Encoder Representations from Transformers) [5],... Cuối cùng là khối giải mã (decoder), thành phần này kết hợp các thông tin từ Acoustic model và Language model để đưa ra dự đoán văn bản đầu ra. Các kiến trúc decoder thường dùng hiện nay có thể kể đến như Transformers Decoder [6], Connectionist Temporal Classification (CTC) [7],... để sinh ra chuỗi là kết quả dạng văn bản của câu nói được nhận dạng từ đầu vào.

Thành phần thứ 2 trong mô hình hiểu ngôn ngữ tự nhiên dạng nói thông thường là mô hình hiểu ngôn ngữ dạng nói (NLU). Như đã đề cập ở trên mô hình NLU thường giải quyết 2 công việc: phân loại ý định (intent classification) và nhận dạng thực thể (entity recognition). Tuy nhiên trong bài toán SLU, mô hình NLU được sử dụng để phân loại ý định và điền khoảng ngữ nghĩa (slot filling) hoặc gán nhãn giá trị (slot tagging). Các mô hình NLU hiện nay thường sử dụng kiến trúc transformers cho cả phần mã hóa (encoder) và giải mã (decoder) và các kết quả đạt được đều tốt trên nhiều bộ dữ liệu SLU.

Việc sử dụng mô hình SLU dựa trên sự ghép nối của mô hình ASR và NLU hiện nay là một trong 2 hướng nghiên cứu chính cho bài toán SLU, các nghiên cứu mang lại các kết quả rất tốt trên nhiều bộ dữ liệu SLU bao bao gồm các bộ dữ liệu SLURP [8] (A Spoken Language Understanding Resource Package), bộ dữ liệu FSC [9] (Fluent Speech Commands) và bộ dữ liệu ATIS [10] (Airline Travel Information Systems),...

1.2.2 Mô hình hiểu ngôn ngữ tự nhiên dạng nói đầu cuối

Mặc dù các mô hình hiểu ngôn ngữ tự nhiên truyền thống (Conventional SLU) đem lại các kết quả tốt, tuy nhiên việc sử dụng 2 mô hình ghép nối với nhau là ASR và NLU muốn đem lại kết quả tốt cho bài toán SLU thì cần phải tiền huấn luyện các mô hình này trước vì vậy lượng dữ liệu và tài nguyên tiêu tốn để huấn luyện là rất lớn. Để giải quyết vấn đề giảm bớt tiêu tốn nhiều tài nguyên và dữ liệu, mô hình đầu cuối hiểu ngôn ngữ tự nhiên dang nói (End-to-end SLU) đang ngày càng được quan tâm phát triển. Hiện nay có nhiều nghiên cứu về mô hình End-to-end SLU nhưng



Hình 1.2: Kiến trúc chung cho mô hình End-to-end SLU

nhìn chung mô hình phổ biến có kiến trúc như hình 0.2. Với đầu vào là một đoạn âm thanh sẽ đi qua khối Speech Encoder, khối này thường là các pretrained model như Wav2vec2 [11], Data2vec [12], Transformers encoder [6], Conformer encoder [13],... Sử dụng các mô hình huấn luyện trước (pre-trained model) đã được huấn luyện trước trên cùng ngôn ngữ sẽ khiến mô hình tăng cường khả năng trích xuất

thông tin, tiết kiệm thời gian tính toán, cải thiện khả năng tổng quát hóa và tránh overfitting. Đầu ra của khối encoder là thông tin về câu nói đã được trích xuất gọi là encoder state, dùng encoder state làm đầu vào của khối Decoder. Khối decoder thường sẽ gồm 2 phần là một mạng giải mã thường là mạng RNN, LSTM, GRU,.. [2] [3] sẽ sử dụng encoder state để giải mã ra các thông tin cần thiết được tạm gọi là decoder state, sau phần giải mã là các lớp dự đoán (predict layer) để dự đoán ra các nhãn kết quả. Kết quả có thể là ý định (intent) hay các thực thể (entities),... Các mô hình decoder thường được sử dụng trong các mô hình đem lại kết quả rất tốt hiện nay có thể kể đến như Transformers-based decoder, LSTM-based decoder,...

Sự hiệu quả của các mô hình End-to-end SLU được chứng thực bằng chứng là các mô hình end-to-end SLU đạt được được các kết quả tốt trên nhiều bộ dữ liệu, thậm chí có các nghiên cứu có độ chính xác lên đến hơn 99% trên bộ dữ liệu FSC [9] cho phần phân loại ý định[1], ngang bằng với các mô hình ASR-NLU thông thường.

1.2.3 Các bộ dữ liệu hiểu ngôn ngữ tự nhiên dạng nói phổ biến

Sự phát triển của các nghiên cứu về mô hình hiểu ngôn ngữ tự nhiên dạng nói kéo theo sự ra đời cùa các bộ dữ liệu phục vụ cho bài toán này. Hiện nay, các bộ dữ liệu SLU khá phổ biến trên nhiều ngôn ngữ nhưng tiếng Anh vẫn là ngôn ngữ được phát triển nhiều nhất với nhiều bộ dữ liệu khác nhau với chủ để ứng dụng đa dạng. Một số bộ dữ liệu tiêu biểu được nhiều nghiên cứu thực nghiệm hiện nay có thể nói đến như: bộ dữ liệu SLURP [8] (A Spoken Language Understanding Resource Package dataset), bộ dữ liệu SNIPS [14], bộ dữ liệu ATIS [10] (Airline Travel Information Systems dataset), FSC [9] (Fluent Speech Commands dataset),...

a, Bộ dữ liệu SNIPS

Bộ dữ liệu SNIPS [14] (Speech Natural Language Inference and Processing System dataset) được ra đời vào năm 2017 bởi công ty SNIPS là một công ty chuyên phát triển các hệ thống trí tuệ nhân tạo trong xử lý ngôn ngữ tự nhiên. Bộ dữ liệu được xây dựng dựa trên các tác vụ của người dùng với các thiết bị thông minh trong nhà với dung lượng là 16000 câu truy vấn, tuy nhiên phiên bản đầu tiên này của bộ dữ liệu chỉ có dạng văn bản để phục vụ cho bài toán NLU, sau khi được mua lại bởi công ty Sonos vào năm 2019, bộ SNIPS được phát triển thêm các bộ dữ liệu mới cho bài toán SLU với 2 bộ dữ liệu là SNIPS-SmartLights và SNIPS-Smartspeaker [15]. Bộ SNIPS-smartlight [15] là bộ dữ liệu có 6 ý định cho phép người dùng đưa ra các yêu cầu điều khiển các loại đèn chiếu sáng trong tiếng Anh. Bộ SNIPS-Smartspeaker [15] bao gồm 9 ý định đối với tiếng Anh và 8 ý định

[1] https://paperswithcode.com/sota/spoken-language-understanding-on-fluent

đối với tiếng Pháp, bộ này bao gồm các yêu cầu điều khiển các thiết bị âm thanh trong nhà.

b, Bộ dữ liệu ATIS

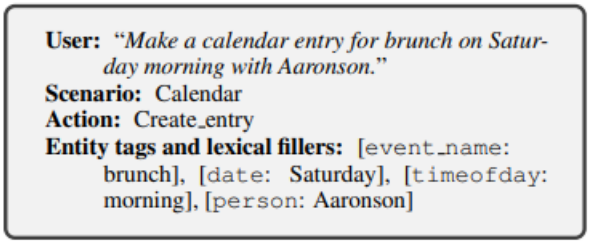
Bộ dữ liệu ATIS [10] (Airline Travel Information Systems dataset) là bộ dữ liệu bao gồm các bản thu âm với phiên âm tương ứng về những truy vấn thông tin chuyến bay như việc đặt vé, lịch trình, chỗ ngồi và các dịch vụ hàng không khác trên các hệ thống truy vấn hành trình tự động của các hãng hàng không. Bộ dữ liệu bao gồm 26 ý định khác nhau, 129 slot label với gần 6000 tệp âm thanh. Bộ dữ liệu ATIS cũng thường được sử dụng có cả 2 bài toán NLU và SLU.

c, Bộ dữ liệu FSC

Bộ dữ liệu FSC [9] (Fluent Speech Commands dataset) được phát triển bởi công ty Fluent.ai trên ngôn ngữ tiếng Anh trong lĩnh vực điều khiển các thiết bị trong nhà thông minh hoặc trợ lý ảo. Bộ dữ liệu được thu âm từ những người Mỹ và Canada, họ sẽ nói các câu yêu cầu đã được lên kịch bản sẵn bởi nhà phát triển. Mỗi câu trong bộ dữ liệu sẽ bao gồm các trường thông tin là action (bao gồm 'change language', 'activate', 'deactivate', 'increase', 'decrease', 'bring'), object (bao gồm 'none', 'music', 'lights', 'volume', 'heat', 'lamp', 'newspaper', 'juice', 'socks', 'shoes', 'Chinese', 'Korean', 'English', 'German') và location (bao gồm 'none', 'kitchen', 'bedroom', 'washroom'). Người thu âm sẽ được cung cấp một cụm các thông tin (ví dụ: action: "activate", object: "lights", location: "none") và đưa ra câu nói dựa trên các thông tin này, điều này đảm bảo tính đa dạng và tự nhiên của các câu nói mà không bị dập khuôn như đọc câu có sẵn. Sau quá trình thu thập và xử lý dữ liệu, bộ dữ liệu FSC thu được bao gồm 30043 tệp âm thanh từ 248 cụm thông tin, với 31 ý định khác nhau và được thu âm bởi 97 người tham gia.

d, Bộ dữ liệu SLURP

Bộ dữ liệu SLURP [8] (A Spoken Language Understanding Resource Package dataset) là bộ dữ liệu hiểu ngôn ngữ tự nhiên dạng nói tiếng Anh, được ra đời vào năm 2020 với quy mô và độ đa dạng hơn hẳn so với các bộ dữ liệu hiện có lúc đó và thậm chí là đến hiện nay. SLURP [8] là bộ dữ liệu được thu âm bởi hơn 100 người dưới hình thức đọc lại các câu đã được thiết kế từ trước, nó bao gồm 72000 tệp âm thanh ghi âm của một người dùng tương tác với trợ lý ảo trong nhà. Ngoài ra các nhà phát triển bộ dữ liệu còn sử dụng thêm hệ thống Google's Text-to-Speech với 34 giọng tổng hợp tiếng nói khác nhau để sinh ra hơn 50000 tệp âm thanh nâng tổng quy mô của bộ SLURP lên hơn 120000 tệp âm thanh. Bộ dữ liệu gồm 3 mức biểu diễn ngữ nghĩa là Scenario, Action and Entities (Ví dụ trong hình 3). Trong đó có 18 Scenario khác nhau, 54 action và 56 loại entity khác nhau. Ý



Hình 1.3: Ví dụ một bộ chú thích cho một câu nói trong bộ SLURP

định trong bộ dữ liệu là 91 ý định (intent) được sinh ra bằng cách ghép nối cái cặp "Scenario_Action".

	FSC	Snips	SLURP	SLURP -synt
Scenarios	2	2	**18**	18
Actions	6	7	**46**	54
Entities	2	4	**56**	56
Tot. Entities	334	2,870	**16,792**	14,623
Entity/Sentence	**1.35**	0.98	0.97	0.65
Unique Entities	16	1,348	**5,613**	4619

Bảng 1.1: Bảng so sánh số lượng actions, scenarios và entities giữa các bộ dữ liệu

Trong bảng thống kê hình 1.3 có thể thấy rõ độ đa dạng của action, scenario hay entities trong bộ SLURP [8] đều vượt trội với số lượng gấp nhiều lần so với các bộ dữ liệu khác. Trong bài báo nghiên cứu công bố bộ dữ liệu này, nhà phát triển cũng để xuất một chỉ số đánh giá mới cho bài toán SF là SLU-F1 [8]. Với quy mô và độ đa dạng của mình, bộ dữ liệu SLURP [8] là một trong những bộ dữ liệu khó nhất hiện nay, bằng chứng là các nhiệm vụ của bài toán SLU như phân loại ý định (IC) hiện chỉ ở mức 91%-92% intent Accuracy hay điền khung ngữ nghĩa (SF) thì chỉ đạt mức 0.82 SLU-F1[2].

1.3 Bài toán hiểu ngôn ngữ tự nhiên dạng nói tiếng Việt

Sự phát triển của lĩnh vực hiểu ngôn ngữ tự nhiên dạng nói trên thế giới đã đạt được nhiều thành công, tuy nhiên các nghiên cứu thường chỉ đánh giá trên các ngôn ngữ phổ biến như tiếng Anh, tiếng Trung,... Nhưng với một ngôn ngữ nghèo tài nguyên (Low-resource language) như tiếng Việt sự phát triển của các nghiên cứu là hoàn toàn thấp hơn khi so sánh với các ngôn ngữ phổ biến. Đối với bài toán hiểu ngôn ngữ tự nhiên dạng nói trong tiếng Việt hiện tại có khá ít các nghiên cứu và kết quả cũng thường chưa được tốt. Vì vậy trong đồ án này em sẽ thực hiện xây dựng một bộ dữ liệu tiếng Việt cho bài toán hiểu ngôn ngữ tự nhiên dạng nói và

[2]https://github.com/NVIDIA/NeMo/tree/main/examples/slu/speech_intent_slot

thử nghiệm, cải tiến các mô hình End-to-end SLU trên bộ dữ liệu này.

1.4 Mục tiêu và phạm vi đề tài

Với các vấn đề về bài toán hiểu ngôn ngữ tự nhiên dạng nói đối với tiếng Việt, đồ án sẽ có 2 nhiệm vụ cần giải quyết chính: (i) đề xuất quy trình xây dựng mộ dữ liệu SLU trong tiếng Việt, (ii) đề xuất các cải tiến mô hình End-to-end SLU đối với tiếng Việt.

Trong nhiệm vụ thứ nhất, tác giả cần đề xuất được một quy trình xây dựng bộ dữ liệu SLU cho tiếng Việt gồm bốn bước chính: đầu tiên là chuẩn bị các dữ liệu dựa theo các khảo thực tế, thứ hai là bước sinh các dữ liệu mà ở đây là thiết kế thuật toán có thể sinh các tổ hợp dữ liệu gợi ý cho một câu nói dựa trên dữ liệu đã chuẩn bị được ở bước 1, thứ 3 là bước thu thập dữ liệu và cuối cùng là xử lý toàn bộ dữ liệu đã thu được. Quy trình này có thể áp dụng cho nhiều chủ đề khác nhau cho bài toán SLU nhưng trong phạm vi đồ án này, tác giả sẽ áp dụng quy trình trên với các yêu cầu và tương tác giữa con người với thiết bị thông minh trong nhà. Bộ dữ liệu thu được sau khi trải qua toàn bộ quy trình xây dựng trên dự kiến đạt ít nhất 50 câu nói với mỗi người tham gia thu âm và tổng thời lượng là khoảng 10 giờ.

Nhiệm vụ thứ hai của đồ án là đề xuất các cải tiến trên mô hình End-to-end SLU đối với tiếng Việt. Tận dụng các đặc trưng có ảnh hưởng lớn trong giọng nói như cao độ đối với tiếng Việt để thực hiện các cải tiến phụ thuộc vào đặc trưng đó. Ngoài ra, đồ án còn sử dụng thêm mô hình mã hóa âm học để trích xuất được các thông tin của âm thanh có thể bị mất mát trong quá trình nén âm thanh thành các vec-tơ đặc trưng. Các cải tiến trên đều sẽ phải được thực nghiệm trên bộ dữ liệu tiếng Việt, kết quả thu được mô hình cải tiến cần đạt mức tăng từ 1-3% cho mỗi loại nhiệm vụ trong bài toán SLU bao gồm phân loại ý định và điền khung ngữ nghĩa.

1.5 Bố cục đồ án

Tại chương đầu tiên của đồ án, tác giả đã thực hiện giới thiệu và thảo luận về tình hình các nghiên cứu và các bộ dữ liệu phổ biến của bài toán hiểu ngôn ngữ tự nhiên dạng nói trên thế giới và trên tiếng Việt.

Chương 2 của đồ án sẽ trình bày về các cơ sở lý thuyết áp dụng cho bài toán hiểu ngôn ngữ tự nhiên dạng nói. Phần này sẽ bao gồm các kiến thức cơ bản của mạng thần kinh nhân tạo, các mô hình xử lý đặc trưng ngôn ngữ nâng cao và các kiến trúc thường được áp dụng trong các mô hình SLU hiện nay.

Chương 3 của đồ án sẽ trình bày chi tiết quy trình xây dựng bộ dữ liệu SLU tiếng Việt bao gồm các bước từ việc chuẩn bị dữ liệu, sinh dữ liệu cho đến thu thập và

xử lý dữ liệu. Phần cuối của chương sẽ trình bày kết quả và các thống kê phân tích về bộ dữ liệu thu được.

Chương 4 của đồ án trình bày về mô hình cơ sở, mô hình đề xuất của tác giả và kết quả thực nghiệm của các mô hình trên bộ dữ liệu tiếng Việt.

Chương 5 của đồ án tổng kết các kết quả đã đạt được và đưa ra định hướng phát triển trong tương lai.

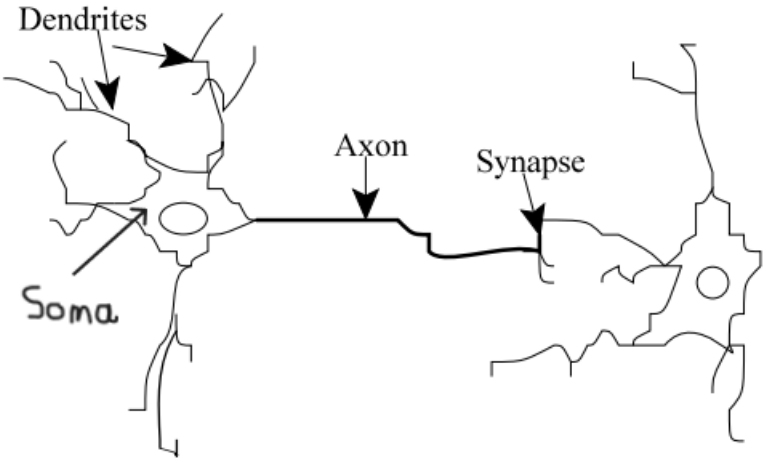
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương 2 trong đồ án sẽ là phần trình bày về lý thuyết cơ bản về mạng nơ-ron nhân tạo. Thêm nữa, phần này cũng sẽ trình bày về các mô hình và kiến trúc phục vụ cho bài toán hiểu ngôn ngữ tự nhiên. Đây chính là cơ sở để xây dựng phương pháp giải quyết bài toán SLU được trình bày trong chương 4.

2.1 Mạng nơ-ron nhân tạo

2.1.1 Mạng nơ-ron nhân tạo

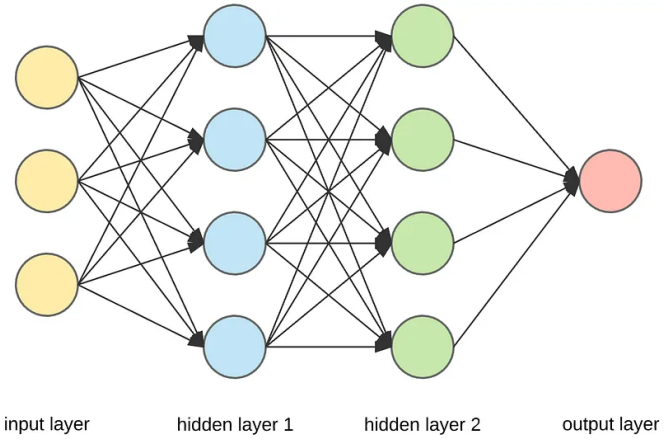
Mạng nơ ron nhân tạo (Neural network) là mô hình tính toán mô phỏng cấu trúc và cách hoạt động của não người.



Hình 2.1: Cấu trúc mạng nơ-ron sinh học [16]

Mạng nơ ron sinh học bao gồm 3 phần chính là: Thân tế bào (Soma), các tế bào tua gai (dendrites) và axon (hình 2.1). Ban đầu, các dendrite sẽ nhận tín hiệu từ các tế bào xung quanh, sau đó tín hiệu sẽ được Soma xử lý và tổng hợp. Axon sẽ đóng vai trò truyền tin hiệu đã được xử lý đến đầu ra. Trong mạng NN, thành phần Perceptron cấu tạo cũng tương tự như vậy. Trong một mạng Perceptron sẽ bao gồm các lớp đầu vào, tập các trọng số và hàm kích hoạt, cuối cùng là lớp đầu ra, cai trò của chúng tương đương với vai trò của dendrite, soma và axon. Mạng nơ ron nhân tạo được tạo từ nhiều lớp xếp chồng lên nhau, mỗi lớp gồm nhiều nơ ron hay perceptron nhưng chung quy lại mạng nơ ron cấu tạo từ 3 thành phần chính: lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer). Trong một mạng nơ-ron nhân tạo có L lớp xếp chồng lên nhau, tại mỗi lớp thứ i sẽ có một tập các nơ-ron tương ứng đặt là a^i, hai lớp liền kề nhau có chỉ số là i và i+1 được kết

[1]https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a.



Hình 2.2: Cấu trúc mạng nơ-ron nhân tạo[1]

nối với nhau thông qua một ma trận trọng số W^i và một vec-tơ bias b^i. Quá trình tính toán trên các nơ-ron từ đầu vào đến đầu ra được biểu diễn toán học với công thức 2.1:

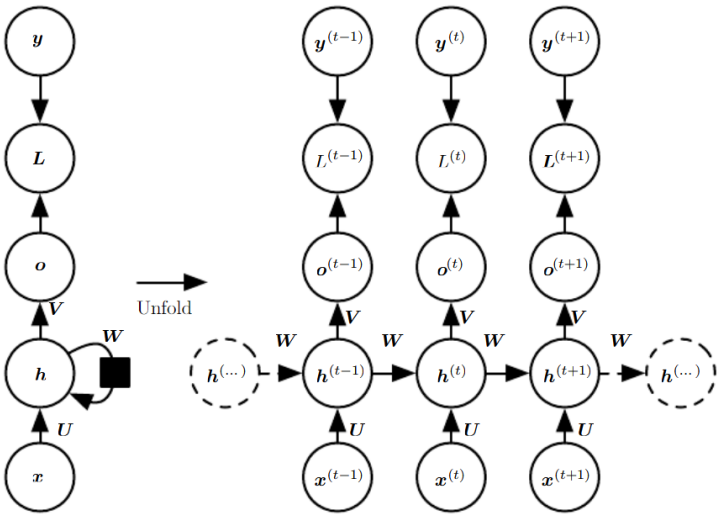
$$a^i = \sigma(W^i a^{i-1} + b^i) \tag{2.1}$$

Trong công thức, σ được gọi là hàm kích hoạt (activate function), hàm kích hoạt phải là hàm phi tuyến (non-linear function) vì mạng nơ-ron cần xử lý và học các biểu diễn phức tạp của dữ liệu, nếu sử dụng một hàm tuyến tính (linear function) mạng nơ-ron sẽ chỉ là một tổ hợp tuyến tính dẫn đến không có khả năng học được nhiều thông tin từ dữ liệu. Các hàm kích hoạt thông dụng hiện nay là sigmoid, tanh, ReLU (Rectified Linear Unit) [17], Softmax,... Quá trình lần lượt tính toán mạng nơ-ron từ lớp đầu vào đến lớp đầu ra được gọi là lan truyền tiến (feed forward) [18]

Việc thực hiện huấn luyện mạng nơ-ron thực chất là quá trình điều chỉnh các trọng số **W** và vec-tơ bias **b**, việc này được thực hiện thông qua quá trình lan truyền ngược (backpropagation) [19]. Để đo đạc độ sai khác giữa nhãn đầu vào và kết quả đầu ra của mạng nơ-ron, ta sử dụng các hàm mất mát (loss function). Tùy vào bài toán cần giải quyết mà sử dụng các hàm mất mát khác nhau, ví dụ bài toán hồi quy sử dụng trung bình bình phương sai số (Mean Squared Error - MSE), bài toán phân loại sủ dụng entropy chéo (Cross Entropy - CE),... Ngoài kiến trúc mạng nơ-ron nhân tạo cơ bản vừa nêu ở trên, đã có nhiều mạng nơ-ron nhân tạo khác phức tạp hơn được tạo ra với mục đích phục vụ cho từng bài toán đặc thù. Các mạng nơ-ron nổi bật có thể kể đến như mạng nơ-ron hồi quy (Recurrent Neural Network - RNN), mạng nơ-ron tích chập (Convolution Neural network - CNN), mạng bộ nhớ ngắn-dài hạn (Long-Short Term Memory - LSTM) [2],...

2.1.2 Mạng nơ-ron hồi quy

Mạng nơ-ron hồi quy (RNN) được ra đời với mục đích giải quyết các bài toán liên quan đến chuỗi dữ liệu. Đầu vào của mô hình RNN là chuỗi dữ liệu x(t) = x(1)...x(n) với chỉ số thời gian t nằm trong khoảng từ 1 đến n. Trong bài toán xử lý ngôn ngữ tự nhiên (NLP), việc dự đoán từ tiếp theo trong câu thì phải dựa trên thông tin của các từ trước nó. Mạng RNN được gọi là hồi quy vì nó thực hiện cùng một tác vụ cho tất cả phần tử của chuỗi với đầu ra là phụ thuộc vào các tính toán trước đó. RNN có một bộ nhớ ghi lại tất cả thông tin đã được tính toán trước đó. Trong mạng nơ-ron cơ bản, các đầu vào đều độc lập với nhau nhưng trong mạng RNN mọi đầu vào đều có liên quan đến nhau. Hình 2.3 là 2 hình thức biểu diễn



Hình 2.3: Kiến trúc cơ bản mạng RNN[2]

của mô hình RNN với bên trái là mô hình RNN thu gọn còn bên phải là hình biểu diễn mô hình RNN sau khi được duỗi thẳng (unfold). Đầu vào của mô hình là một chuỗi với x(t) là đầu vào của mạng tại bước thời gian t, ví dụ: x(1) có thể là vector 1 chiều biểu diễn 1 từ trong câu. h(t) là trạng thái ẩn tại thời điểm t và đóng vai trò là bộ nhớ của mạng. h(t) được tính toán đầu vào hiện tại và trạng thái ẩn của của thời điểm trước đó:

$$h^{(t)} = f(Ux^{(t)} + Wh^{(t-1)}) \tag{2.2}$$

f là hàm non-linear như ReLU hoặc tanh. Trong mạng RNN bộ tham số trạng thái ẩn giữa đầu vào và trạng thái ẩn là tập trọng số U, bộ tham số giữa các trạng thái thái ẩn là W và trọng số giữa trạng thái ẩn với đầu ra là V, bộ ba trọng số (U, W, V) này được tính toán kết hợp chéo với nhau theo thời gian. Đầu ra của mạng RNN là

[2]https://www.deeplearningbook.org/contents/rnn.html

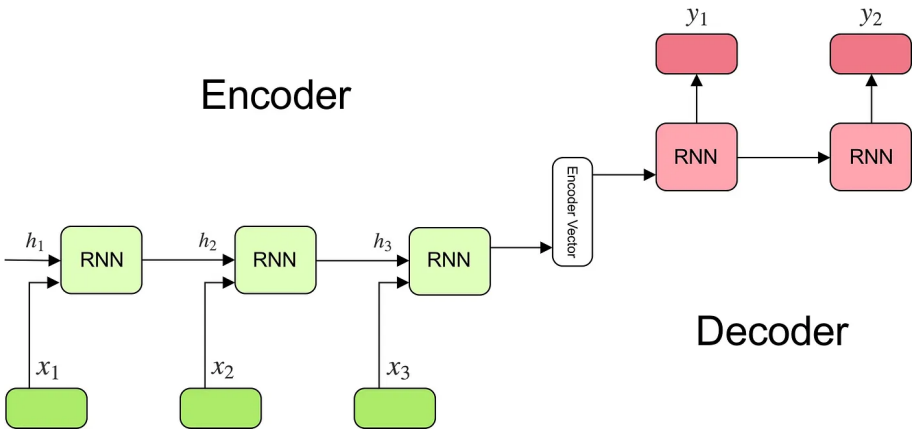
$o^{(t)}$ tại thời điểm t, mỗi đầu vào $x^{(t)}$ sẽ cho ra tương ứng đầu ra $o^{(t)}$, đầu ra $o^{(t)}$ được tính theo công thức 2.3.

$$o^{(t)} = c + Vh^{(t)} \tag{2.3}$$

ta sẽ sử dụng hàm mất mát L để tính toán sự sai khác giữa kết quả đầu ra o(t) và nhãn y(t).

2.2 Mô hình giải mã chuỗi - chuỗi

Trong các bài toán xử lý ngôn ngữ tự nhiên hiện nay việc các mô hình nhận đầu vào là một chuỗi dữ liệu và nhận lại đầu ra cũng là một chuỗi kết quả đang thực sự phổ biến. Nó có thể áp dụng vào các hệ thống dịch máy (machine translation), hệ thống hiểu ngôn ngữ tự nhiên NLU,... Với đầu vào là một chuỗi ký tự mô hình sequence-to-sequence (Seq2Seq) [20] có thể cho kết quả là câu đầu với kích thước của câu khác kích thước câu đầu vào. Mô hình Seq2Seq có cấu trúc dạng encoder - decoder là cấu trúc học sâu thường được áp dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên hoặc thị giác máy tính. Cấu trúc gồm 2 phần chính là bộ mã hóa (encoder) và bộ giải mã (decoder). Khối Encoder có nhiện vụ chuyển các đặc trưng có thể học được trong tùy loại yêu cầu của bài toán. Đối với mạng ANN, encoder thường là các hidden layer. Đối với mô hình CNN, encoder sẽ là các chuỗi Convolutional + Maxpooling layer. Và trong mô hình RNN cấu trúc của mạng sẽ là các lớp RNN (lớp LSTM [2] hoặc GRU [3] thường cho kết quả tốt nhất) và lớp embedding. Đầu ra của khối mã hóa chính là trạng thái ẩn cuối cùng trong khối gọi là trạng thái mã hóa (encoder state), vec-tơ trạng thái này nằm bắt và đóng gói tất cả thông tin quan trọng để cung cấp cho bộ giải mã có thể đưa ra dự đoán đúng. Bộ giải mã nhận



Hình 2.4: Minh họa cấu trúc mô hình sequence-to-sequence[3]

encoder vec-tơ làm đầu vào để thực hiện nhiệm vụ giải mã, tại mỗi điểm thời gian

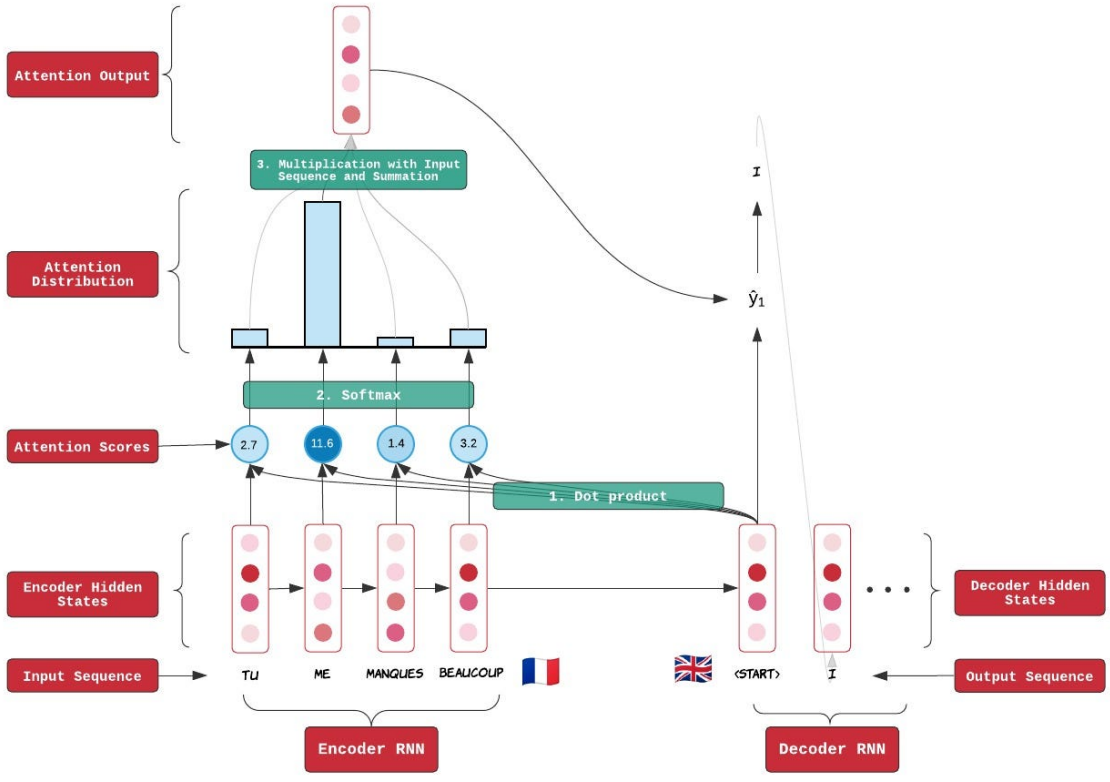
[3]https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

bộ giải mã sử dụng cả trạng thái ẩn hiện tại, encoder vec-tơ và ký tự mã hóa (token) sinh ra trước đó để tính xác suất cho các ký tự mã hóa sinh ra tiếp theo. Cuối cùng token nào có xác suất cao nhất được chọn làm đầu ra, cứ tiếp tục như vậy cho đến khi nhận được hoàn toàn chuỗi đầu ra.

2.3 Mô hình Transformers

2.3.1 Cơ chế chú ý (Attention)

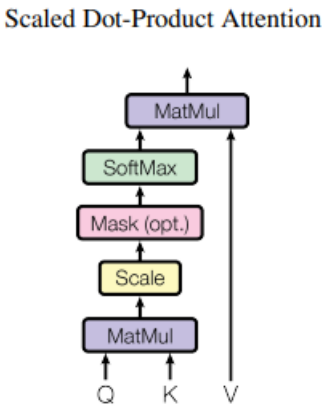
Trong mô hình Seq2Seq [20] sử dụng mạng RNN đã trình bày ở trên có một vài hạn chế khi áp dụng đối với các chuỗi dữ liệu dài như là hiện tượng tiêu biến gradient (vanishing gradient) và bùng nổ gradient (exploding gradient). Mạng LSTM ra đời đã phần nào khắc phục được 2 tình trạng này trong mô hình seq2seq, tuy nhiên lại phát sinh các vấn đề khác khi sử dụng mạng LSTM [2] là tốn nhiều thời gian huấn luyện và khó huấn luyện. Nhưng nếu sử dụng mô hình RNN để huấn luyện sẽ gây ra tình trạng bộ mã hóa trong mô hình sẽ nén toàn bộ thông tin của chuỗi đầu vào vào trong một vec-tơ duy nhất để cung cấp cho bộ giải mã, khi chuỗi dữ liệu dài thông tin sẽ càng trở nên trừu tượng và sẽ rất khó để bộ giải mã có thể đoán đúng chính xác đầu ra cần tìm. Vì vậy để giải quyết những vấn đề này cơ chế chú ý (attention) được ra đời vào năm 2014 bởi Dzmitry Bahdanau trong bài báo nghiên cứu về dịch máy (machine translation) [21].



Hình 2.5: Minh họa mô hình squence-to-squence sử dụng attention[4]

Cơ chế attention cho phép mô hình có thể chú ý vào từng phần tử quan trọng của chuỗi đầu vào, từ đó không cần phải nén toàn bộ thông tin vào một vec-tơ duy nhất. Cơ chế chú ý sẽ tập trung vào những phần quan trọng trong chuỗi bằng cách tính toán và gán các trọng số cho từng phần tử trong chuỗi. Các phần tử quan trọng sẽ mang trọng số cao hơn, từ đó có thể dễ dàng đặc trưng hóa mạnh mẽ hơn các phần tử quan trọng cho phù hợp với từng tác vụ. Ví dụ trong hình 2.5, mô hình seq2seq được áp dụng cho bài toán machine translation, khi áp dụng cơ chế chú ý có thể thấy rõ Từ "I" trong tiếng Anh và từ "me" trong tiếng Pháp có cùng nghĩa nên được lớp attention nâng trọng số chú ý cao hơn hẳn trọng số với các từ còn lại, điều này giúp quá trình dự đoán đầu ra của các từ trong tiếng Anh và tiếng Pháp sẽ dễ dàng hơn rất nhiều.

Cơ chế chú ý được triển khai với nhiều biến thể khác nhau tùy thuộc vào yêu cầu của từng bài toán. Ví dụ có thể nói đến cơ chế tự chú ý (Self-attention). Cơ chế Self-Attention cho phép mô hình tập trung vào các phần tử quan trọng trong cùng một dữ liệu đầu vào, thường là các từ trong câu hoặc chuỗi từ, để hiểu và tạo ra các đặc trưng phụ thuộc vào ngữ cảnh, mô hình Dot product attention [6] ở trong hình 2.6 là một loại self attention.



Hình 2.6: Minh họa cơ chế attention cơ bản [6]

Một mô hình attention cơ bản sẽ có 3 đầu vào là Query (Q), Key (K), Value (V). Q là một vec-tơ ma trận đại diện cho các phần tử xuất hiện trong chuỗi đầu ra, K cũng tương tự vậy nhưng đai diện cho các phần tử trong chuỗi đầu vào, V là vec-tơ biểu diễn nội dung ngữ nghĩa của từng phần tử. Cơ chế Attention sẽ tính toán các điểm tương quan giữa Query và Key để xác định mức độ tương quan giữa các phần tử trong dữ liệu đầu vào và dữ liệu đầu ra, sau đó sử dụng hàm softmax để chuẩn hóa các trọng số chú ý. Vec-tơ chú ý sẽ được tính dựa trên trung bình có trọng số

[4]https://towardsdatascience.com/the-definitive-guide-to-bidaf-part-3-attention-92352bbdcb07

của vec-tơ value trong ma trận V với trọng số chú ý vừa tính được từ Q và K. Vec-tơ chú ý đầu ra sẽ biểu diễn ngữ cảnh và các phần tử quan trọng nhất của dữ liệu đầu vào so với dữ liệu đầu ra. Công thức tính vec-tơ chú ý như sau:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.4}$$

Hiện nay cơ chế chú ý là một công cụ rất mạnh của công nghệ trí tuệ nhân tạo, nó cho phép các mô hình tập trung vào các phần tử quan trọng trong dữ liệu làm cho quá trình tính toán trở nên chính xác và dễ dàng hơn. Tạo điều kiện cho các nghiên cứu về dữ liệu dạng chuỗi có thể ngày càng đạt được những thành tựu lo lớn hơn nữa.

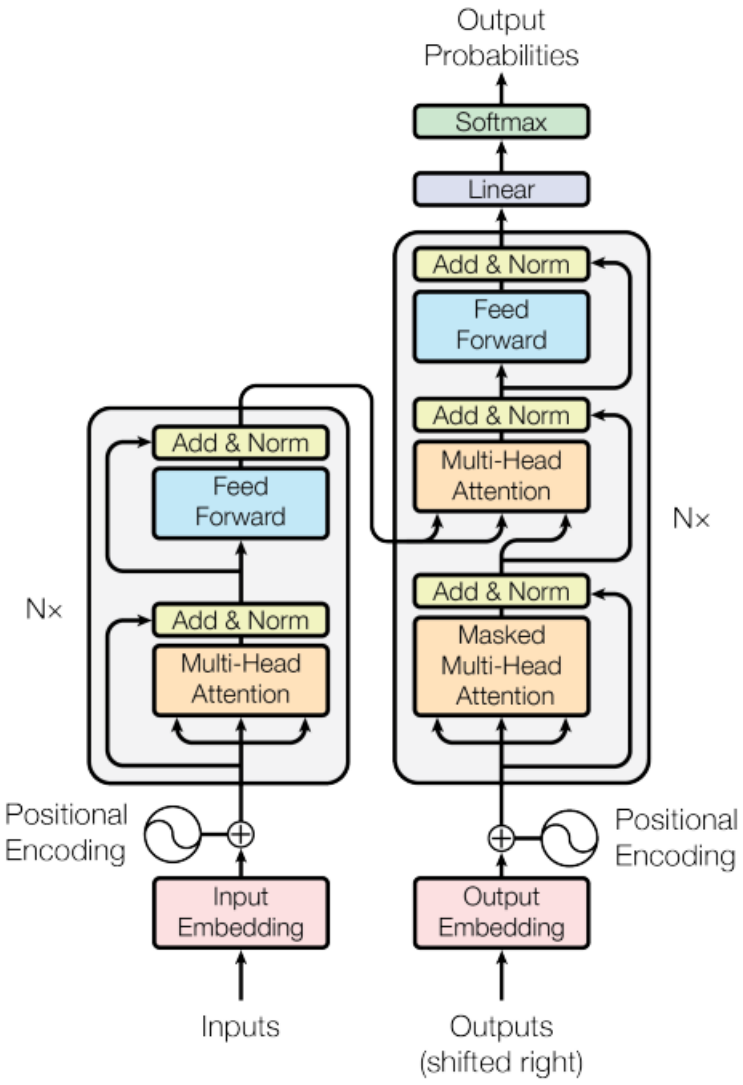
2.3.2 Mô hình Transformers

Transformer là kiến trúc được Google công bố trong bài báo "Attention is all you need" vào năm 2017 [6]. Đây được coi như là một bước ngoặt, đột phá của lĩnh vực trí tuệ nhân tạo và đặc biệt là trong lĩnh vực xử lý ngôn ngữ tự nhiên NLP. Trước khi transformer ra đời hầu hết các tác vụ trong xử lý ngôn ngữ tự nhiên đều phải sử dụng kiến trúc mạng nơ ron hồi quy (RNNs). Việc chỉ có thể xử lý các chuỗi đầu vào một cách tuần tự khiến tốc độ xử lý của mạng RNNs chậm chạp và còn không thể bao quát sự phụ thuộc của phần tử đứng xa nhau trong một chuỗi. Khi mô hình transformer xuất hiện các vấn đề trên đã được giải quyết một cách triệt để. Những năm gần đây transformer chính là nền tảng của rất nhiều mô hình phát triển trong nhiều lĩnh vực về trí tuệ nhân tạo

Mô hình transformer [6] dựa trên cơ chế chú ý, điều này làm cho mô hình transformer có thể tập trung vào các phần tử quan trọng trong một chuỗi dữ liệu. Trong kiến trúc của mình transformer chứa nhiều khối encoder và decoder mà trong mỗi khối encoder đều chứa một lớp self attention và mạng lan truyền tiến FFNN [18], với self attention mô hình encoder có thể nhìn vào toàn bộ chuỗi trong khi đang tập trung vào một phần tử cụ thể, điều này giúp mô hình transformer [6] hiểu được mối liên quan giữa các phần tử trong chuỗi kể cả chúng có đứng xa nhau. Trong các khối decoder cũng chứa các thành phần tương tự nhưng sẽ có sự bổ sung thêm của một multi-head attention [6] để học được mối liên hệ giữa phần tử đang được chú ý và phần tử ở chuỗi nguồn.

2.4 Kiến trúc Wav2vec2

Wav2vec là kiến trúc tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên được Facebook AI Research công bố vào năm 2019 [22], sau 1 năm nhóm nghiên cứu tiếp tục cho ra phiên bản nâng cấp của mô hình wav2vec gọi là wav2vec2.0 [11].

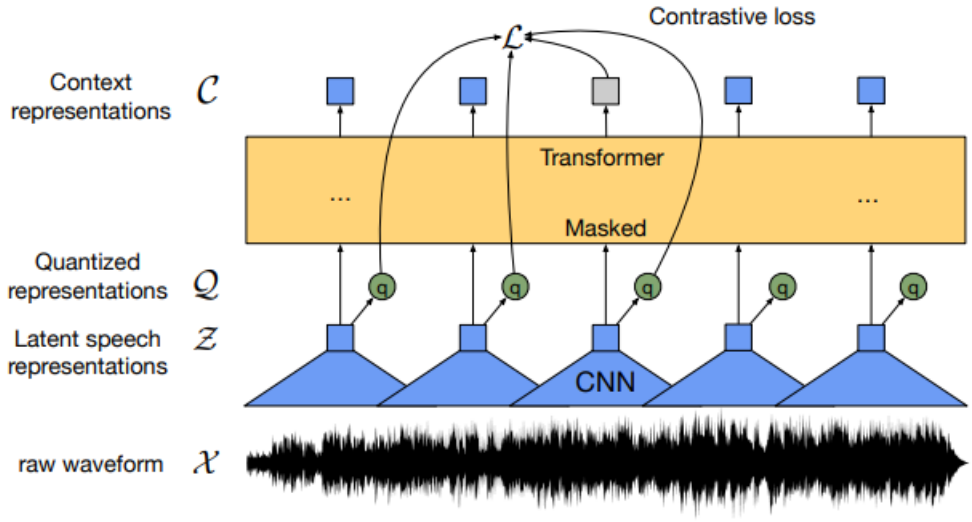


Hình 2.7: Kiến trúc mô hình transformer [6]

Wav2vec 2.0 là một mô hình tự học (self-supervised learning) trong bài toán nhận dạng tiếng nói (speech representation) Kiến trúc wav2vec2 gồm các phần như sau: Mã hóa đặc trưng (Feature encoder): Lớp này gồm nhiều lớp CNN theo sau bởi 1 lớp bình thường hóa dữ liệu (normalization layer) [23] và hàm kích hoạt GELU [24]. Dạng sóng âm thanh thô $\mathcal{X}$ đi qua sẽ được trích xuất các thông tin là các biểu diễn tiềm ẩn $\mathcal{Z} = z_1, ..., z_T$ của âm thanh (latent speech representations) với T là bước thời gian.

Đầu ra của khối mã hóa đặc trưng được đưa vào mạng ngữ cảnh (context network) dựa theo kiến trúc transformer [6], [25] để thu được các đặc trưng $C = c_1, ..., c_T$ mang thông tin của chuỗi đầu vào.

Khối lượng tử hóa: (Quantized module): Trong quá trình huấn luyện tự giám sát (self supervised training), mô hình rời rạc hóa các đặc trưng tiềm ẩn z thành



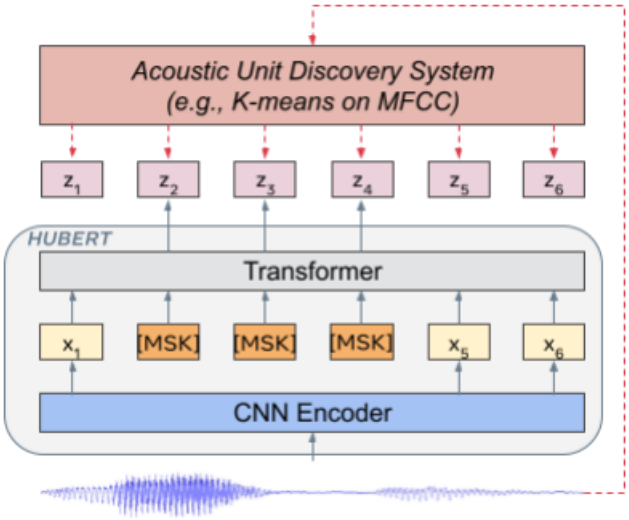
Hình 2.8: Kiến trúc mô hình wav2vec 2.0 [11]

tập hợp hữu hạn các biểu diễn âm thanh thông qua phép lượng tử hóa (product quantization) [26]. Sau đó chọn các biểu diễn rời rạc trong các bảng mã (codebook) và ghép chúng lại và cho qua lớp tuyến tính (linear layer) kết quả thu được chuỗi đặc trưng lượng tử hóa $q_1, ..., q_T$.

Ở mô hình huấn luyện trước (pre-trained model), các latent feature được che (mask) đi ngẫu nhiên tại các vị trí ngẫu nhiên chiếm khoảng 50% số vec-tơ tiềm ẩn, từ đây mô hình sẽ học được các đặc trưng tiềm ẩn thông qua việc sử dụng hàm mất mát Contrastive loss [27]. Tại đây mô hình sẽ cố gắng kéo các đặc trưng giống nhau lại gần nhau và cho chúng học thêm các đặc trưng cấp cao hơn của nhau, còn các đặc trưng khác xa nhau sẽ bị đẩy nhau ra xa đến mức tối đa. Sau quá trình huấn luyện mô hình cùng một số cải tiến khác so với mô hình wav2vec [22], các nhà nghiên cứu đã thu được một mô hình có hiệu quả cao và có thể sử dụng trong các bài toán xuôi (downstream task) như nhận dạng tiếng nói (automatic speech recognition), nhận dạng cảm xúc (emotion detection), nhận dạng người nói (speaker recognition),...

2.5 Kiến trúc HuBERT

Mô hình BERT [5] được ra đời vào năm 2018 với kiến trúc là một mô hình NLP tự giám sát hai chiều (bi-directional self-supervised NLP model) dựa trên kiến trúc transformer. Với sự đột phá trong kiến trúc, BERT là một mô hình NLP thực sự mạnh mẽ trong việc tiền xử lý dữ liệu và trích xuất đặc trưng ngôn ngữ. Tuy nhiên, khi áp dụng BERT hay các mô hình NLP tự giám sát để học cá biểu diễn của tiếng nói lại đối mặt với 3 vấn đề: (i) có rất nhiều âm thanh trong mỗi câu nói



Hình 2.9: Kiến trúc mô hình HuBERT [28]

đầu vào, (ii) không có một từ điển tập hợp đầy đủ các đơn vị âm thanh đầu vào trong quá trình tiền huấn luyện, (iii) độ dài mỗi đơn vị âm thanh luôn biến thiên và không phân đoạn rõ ràng. Vì vậy để giải quyết 3 vấn đề nêu trên, các nhà nghiên cứu đã cho ra đời mô hình HuBERT [28], mô hình được công bố trong bài báo "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units" [28] bởi các nhà khoa học Wei-Ning Hsu, Benjamin Bolte, Yao-Hung HuBERT Tsai, Kushal Lakhotia, Ruslan Salakhutdinov and Abdelrahman Mohamed.

Nhìn vào hình 2.9, có thể thấy rõ kiến trúc HuBERT có sự tuân theo kiến trúc Wav2vec2 [11] ở mức nhất định bao gồm các khối: Convolution encoder, BERT encoder [5], projection layer và Code embedding layer. Số lượng các lớp trong từng thành phần này thay đổi giữa các biến thể khác nhau của mô hình.

Quá trình huấn luyện mô hình HuBERT gồm 2 bước chính: (i) Tạo các đơn vị ẩn (hidden unit) và (ii) dự đoán thông tin bị che (Masked prediction).

Trong bước thứ nhất, mô hình sử dụng mô hình khám phá đơn vị âm học (acoustic unit discovery models) để cung cấp cho các mục tiêu xử lý cấp khung (frame-level), đầu vào là một chuỗi âm thanh có dạng X = $[x_1, ..., x_T]$ với T khung âm thanh, qua phương pháp trích xuất MFCCs thu được các đơn vị ẩn $h(\mathcal{X})$ = Z = $[z_1, ..., z_T]$ với $z_t \in [C]$ là một biến phân loại gồm C lớp và h là một mô hình phân cụm (ví dụ: k-means).

Bước thứ 2 trong mô hình là bước huấn luyện mô hình tương tự mô hình BERT [5] sử dụng mô hình hóa ngôn ngữ ẩn (masked language model). Đầu tiên âm thanh thô được cho qua lớp CNN encoder để thu được các đặc trưng, sau đó che

ngẫu nhiên các đặc trưng đi và đưa chúng vào bộ mã hóa BERT. BERT encoder [5] sẽ cho ra một chuỗi các đặc trưng điền vào các đặc trưng đã bị che trước đó. Đầu ra này sẽ được tính toán độ tương đồng cosine giữa chúng và các đơn vị vec-tơ nhúng Z được tạo ra từ bước đầu tiên. Cuối cùng sử dụng hàm mất mát cross-entropy để tính trên các thời điểm bị ẩn L_m và không bị ẩn L_u. Công thức tính L_m như sau:

$$L_m(f : X, M, Z) = \sum_{t \in M} \log p_f(z_t | \hat{X}, t) \tag{2.5}$$

trong đó $M \subset [T]$ là tập hợp các chỉ số bị che dấu trong chuỗi X có độ dài T. $\hat{X}$ là phiên bản bị che của X với x_t bị thay thế bởi một vec-tơ nhúng bị che $\hat{x}$. f là một mô hình dự đoán dạng che giấu (masked prediction model).

Công thức tính L_u cũng tương tự công thức 2.5 nhưng với các thời điểm $t \notin M$. Hàm mất mát cuối cùng được tính dựa trên tổng trọng số của hai thành phần L_m và L_u:

$$L = \alpha L_m + (1 - \alpha) L_u \tag{2.6}$$

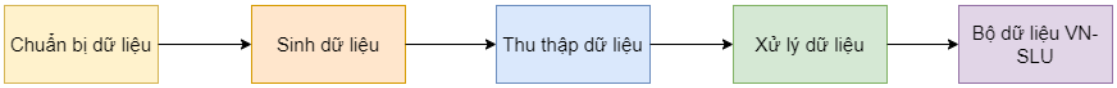
Trong trường hợp cực đoan $\alpha = 0$, hàm mất mát chỉ tính trên các thời điểm không bị che giấu, mô hình lúc này sẽ tương tự các mô hình nhận dạng giọng nói hỗn hợp. Còn trong một trường hợp cực đoan khác là $\alpha = 1$, lúc này hàm mất mát chỉ tính trên các thời điểm bị che giấu, tại đây mô hình phải đưa ra các dự đoán các mục tiêu tương ứng với các khung thời gian bị che từ ngữ cảnh, hoạt động này tương tự như trong mô hình ngôn ngữ. Điều này cho thấy để có thể thu được kết quả tốt mô hình cần học cả những đặc trưng âm học không bị che và cấu trúc về độ dài thời gian của dữ liệu âm thanh.

Trải qua quá trình tiền huấn luyện, mô hình HuBERT trở nên mạnh mẽ với khả năng biểu diễn thông tin phong phú từ dữ liệu giọng nói và ngôn ngữ. Với công dụng của mình mô hình HuBERT có thể được sử dụng trong nhiều bài toán khác nhau mang lại hiệu quả cao như Nhận dạng giọng nói (Speaker recognition), tổng hợp giọng nói (Speech synthesis), hiểu ngôn ngữ tự nhiên dạng nói (SLU),...

CHƯƠNG 3. ĐỀ XUẤT PHƯƠNG PHÁP XÂY DỰNG BỘ DỮ LIỆU TIẾNG VIỆT

3.1 Xây dựng bộ dữ liệu SLU tiếng Việt VN-SLU

Bài toán hiểu ngôn ngữ tự nhiên dạng nói là một trong những bài toán cho tính ứng dụng lớn hiện nay, sự giao tiếp giữa người và máy đang ngày càng tối ưu hơn, thay vì sử dụng giao tiếp thông qua thao tác bằng tay việc sử dụng giọng nói để ra giao tiếp đang được phát triển nhanh chóng. Tuy nhiên, sự phát triển này đang chỉ diễn ra nhanh chóng ở các ngôn ngữ giàu tài nguyên như tiếng Anh và tiếng Trung, còn đối với một ngôn ngữ nghèo tài nguyên như tiếng Việt sự phát triển này đang bị thua kém khá nhiều. Hiện nay không có bộ dữ liệu hiểu ngôn ngữ tự nhiên dạng nói lớn nào được công bố rộng rãi trên tiếng Việt, dẫn tới các mô hình được nghiên cứu về bài toán này trên tiếng Việt cũng hầu như không có. Vì vậy trong đồ án này, tác giả sẽ xây dựng một bộ dữ liệu cho bài toán SLU tiếng Việt với chủ đề là tương tác của con người với các thiết bị trong nhà thông minh. Sau đây sẽ là quá trình xây dựng chi tiết bộ dữ liệu. Quy trình xây trình bộ dữ liệu gồm 4 quá trình lần lượt là: Chuẩn bị dữ liệu, sinh dữ liệu, thu thập dữ liệu, tiền xử lý dữ liệu.



Hình 3.1: Quy trình xây dựng bộ dữ liệu

3.2 Chuẩn bị dữ liệu

Bước đầu tiên để xây dựng một bộ dữ liệu là phải chuẩn bị được các ngữ liệu, ngữ liệu phải đảm bảo độ chính xác, đa dạng với chủ đề của bộ dữ liệu. Đối với bộ SLU tiếng Việt lần này với chủ đề chính là tương tác giữa người dùng và các thiết bị trong nhà thông minh tác giả đề xuất 3 trường thông tin chính bao gồm: Các loại thiết bị trong nhà, các địa điểm, các hành động tương tác với thiết bị,...

Thứ nhất, với dữ liệu về Các loại thiết bị trong nhà, tác giả đã thực hiện khảo sát trên các dự án nhà thông minh hiện tại. Sau khi thực hiện khảo sát và thu thập dữ liệu có khoảng 100 thiết bị thông dụng trong nhà. Minh họa một vài thiết bị tại bảng 1.

Thông tin quan trọng tiếp theo là về các không gian, địa điểm trong và ngoài nhà. Dựa trên những khảo sát về kiến trúc nhà thông minh, khách sạn, nhà hàng,... và những liên hệ thực tế với không gian nhà ở thông thường, tác giả đã chọn lọc ra 72 địa điểm phổ biến được minh họa trong bảng 2. Ngoài các địa điểm cơ bản

máy mát xa	màn cuốn	máy hút ẩm	laptop	bóng âm trần	lò nướng
máy chơi game	màn cửa	máy hút bụi	máy tính	bóng chùm	lò vi sóng
ti vi	mành	máy hút mùi	máy tính để bàn	bóng compact	máy pha cafe
rèm	mành cuốn	máy rửa bát	máy tính xách tay	bóng để bàn	tủ lạnh
rèm cửa	máy giặt	quạt hút mùi	bóng	bóng đứng	máy fax

Bảng 3.1: Bảng minh họa các thiết bị trong nhà

đã liệt kê thì tác giả còn sử dụng một số tên riêng cho các địa điểm, có thể kể đến như: phòng Trường sa, phòng tình yêu,.. hay phòng có gắn thêm tên người ở trong đó như phòng của Nam, phòng của Tuấn,...

ban công	đại sảnh	hành lang	nhà bếp	nhà xe	phòng học
bếp	đầu hè	hầm rượu	nhà để xe	phòng	phòng họp
cầu thang	gác xép	hè	nhà giữ đồ	phòng bếp	phòng họp gia đình
cổng	garage	hiên trước	nhà tắm	phòng chính	phòng khách
cổng chính	gầm cầu thang	hiên	nhà vệ sinh	phòng chờ	phòng làm việc

Bảng 3.2: Bảng minh họa các địa điểm trong nhà

Thông tin tiếp theo là về hành động thực hiện trên các thiết bị gọi là "command", bộ dữ liệu có tổng cộng 7 command bao gồm: bật, tắt, tăng, giảm, đóng, mở, kiểm tra tình trạng. Từ 3 thông tin về thiết bị, địa điểm và hành động tác giả đề xuất 9 ý định tương ứng với 9 hành động bao gồm: bật thiết bị, tắt thiết bị, mở thiết bị, đóng thiết bị, tăng thiết bị, giảm thiết bị, kiểm tra tình trạng thiết bị. Ngoài ra các hành động cơ bản bộ dữ liệu còn có thêm 2 hành động đặc biệt là kích hoạt cảnh và hủy hoạt cảnh, với hoạt cảnh là một trường hợp đặc biệt thay vì chỉ thực hiện một hành động hoạt cảnh sẽ thực hiện một loạt hành động khi được thực hiện. Có tổng cộng 10 hoạt cảnh là: Đi ngủ, đi tắm, ra khỏi phòng, ra ngoài, thư giãn, về nhà, riêng tư, khách tới nhà, tiệc tùng và lãng mãn. Trong cuộc sống hàng ngày khi ở trong những trường hợp như thế này người dùng sẽ không cần phải thực hiện một loạt các yêu cầu điều khiển mà thay vào đó chỉ cần cho trợ lý ảo biết mình đang ở trong hoạt cảnh có thì tự khắc một loạt hoạt động tương ứng với các hoạt cảnh sẽ được thực hiện.

3.3 Chiến lược sinh dữ liệu

Với những ngữ liệu đã chuẩn bị xong, tác giả thực hiện quá trình sinh các tổ hợp gợi ý (combination) để thực hiện hành động bao gồm các trường hợp có thể xảy ra trong bảng 3. Các thực thể xuất hiện trong tổ hợp gợi ý bao gồm: hành động (command), thiết bị (device), địa điểm (location), giá trị thay đổi (changing value), giá trị hướng đến (target value), khoảng thời gian thực hiện (duration), mốc thời gian (time at).

command_device	command_device_time at
command_device_location	command_device_location_duration
command_device_changing value	command_device_location_time at
command_device_target value	command_device_location_changing value_duration
command_device_location_changing value	command_device_location_target value_duration
command_device_location_target value	command_device_location_target value_time at
command_device_duration	command_device_location_changing value_time at

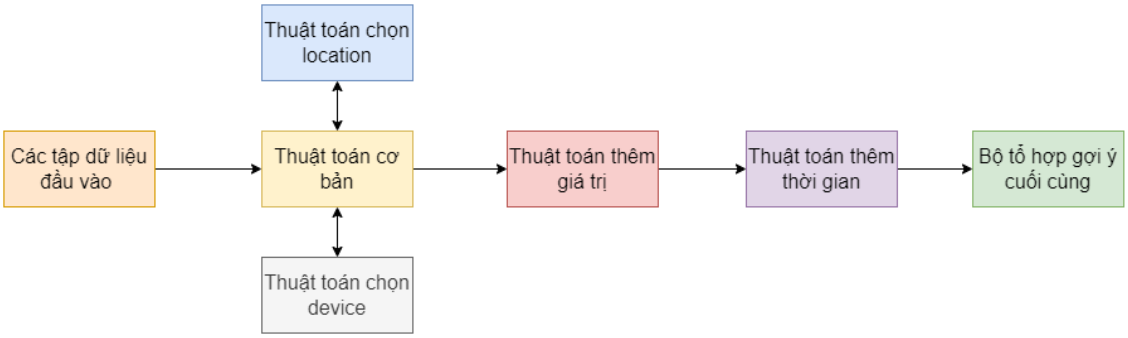
Bảng 3.3: Bảng thống kê các tổ hợp gợi ý

Thuật toán sinh các tổ hợp sẽ được chia thành 5 thuật toán nhỏ là Thuật toán cơ bản, thuật toán chọn location, thuật toán chọn device, thuật toán thêm giá trị, thuật toán thêm thời gian, trong đó với thuật toán cơ bản là thuật toán tổng hợp cả 2 thuật toán chọn location và thuật toán chọn device. Hình 3.2 minh họa cho toàn bộ quy trình sinh dữ liệu.

Bảng 4 ở dưới đây miêu tả các tập dữ liệu đầu vào trong thuật toán:

No	Field_name	Component type	Required?	Description
1	command	dict	yes	Tập dạng dictionary với key là command và value là số lần xuất hiện, mặc định là 0.
2	device	dict	yes	Tập dạng dictionary với key là device và value là số lần xuất hiện, mặc định là 0.
3	locaion	dict		Tập dạng dictionary với key là location và value là số lần xuất hiện, mặc định là 0.
4	device_changing	dict		Tập dạng dictionary gồm các thiết bị có thể thay đổi với khoảng giá trị, trong đó key là tên device và value là giá trị gắn kèm với nó (ví dụ: 'key': 'bếp', 'value': 'khoảng 1 - 10'
5	device_target	dict		Tập dạng dictionary gồm các thiết bị có thể đạt với mức giá trị, trong đó key là tên device và value là giá trị gắn kèm với nó (ví dụ: 'key': 'bếp', 'value': 'mức 1 - 10'
6	command_device	dictionary		Dictionary lưu các giá trị (key, value) với key là các giá trị command (type: String), value là danh sách các giá trị device tương ứng (type: List)
7	device_location	dictionary		Dictionary lưu các giá trị (key, value) với key là các giá trị device (type: String), value là danh sách các giá trị location tương ứng (type: List)
8	Time	set		Tập các thiết bị có thể bị tác động về mặt thời gian (type: String)

Bảng 3.4: Bảng miêu tả các giá trị đầu vào



Hình 3.2: Toàn bộ quy trình sinh dữ liệu

Hoạt động của thuật toán diễn ra như sau:

Thuật toán cơ bản được minh họa trong hình 3.3. Đây là thuật toán sử dụng để khởi tạo, xây dựng các tổ hợp gợi ý cơ bản chỉ bao gồm 3 thực thể là command, device, location. Trước khi thực hiện thuật toán, các giá trị của device sẽ được ghép với các command tương ứng, mỗi giá trị command sẽ có các thiết bị có thể thực hiện tương ứng (ví dụ: "điều hòa" có thể thực hiện hành động "bật", "tắt" hoặc "tăng", "giảm" nhưng không thể có hành động "đóng"), các giá trị của device cũng được ghép với các địa điểm, tại mỗi địa điểm cũng sẽ chỉ có các thiết bị phù hợp. Danh sách ghép cặp sẽ được sử dụng làm đầu vào của thuật toán dưới dạng các dictionary với key là device và value là hành động hoặc địa điểm tương ứng đã được ghép. Quy trình diễn ra thuật toán như sau:

- Bước 1: Tạo vòng lặp với điều kiện dừng là tất cả các thiết bị đều đã xuất hiện trong các combination > 6 lần.
- Bước 2: Chọn một trong 2 template là command_device, command_device_location với tỷ lệ chọn là $10:6$.
- Bước 3: Chọn ngẫu nhiên một hành động trong tập command.
- Bước 4: Chọn ngẫu nhiên 1 trong 3 giá trị device có số lần xuất hiện ít nhất theo "thuật toán chọn device".
- Bước 5: Kiểm tra nếu template có chứa location, nếu có thì chọn giá trị location theo "thuật toán chọn location". Sau khi chọn xong location hoặc template không chứa location thực hiện kiểm tra combination thu được đã tồn tại hay chưa. Nếu chưa tồn tại lưu vào tập Combinations tổng, nếu đã tồn tại quay lại tiếp tục vòng lặp.
- Bước 6: Kết thúc thuật toán trả về tập Combinations chứa các combination cơ bản.

Trong mỗi template đều luôn chứa device, quy trình chọn device diễn ra trong

bước 4 của thuật toán cơ sở được miêu tả trong thuật toán chọn device được minh họa trong hình 3.5. Các giá trị ít xuất hiện nhất trong các tổ hợp gợi ý trước đó sẽ có tỷ lệ lựa chọn cao nhất.Thuật toán chọn device diễn ra như sau:

- Bước 1: Chọn ngẫu nhiên giá trị name_device trong khoảng từ [0:1], nếu $name\_random < 0.1$ và device có chứa tên riêng thì thêm tên riêng cho location. Nếu $name\_random >= 0.1$ tiếp tục bước 2, ngược lại chuyển đến bước 3.
- Bước 2: Chọn ngẫu nhiên giá trị name_device_number trong khoảng [0:1]. Nếu $name\_device\_number < 0.5$ thì thêm tên riêng dạng chữ cho device, ngược lại thêm tên riêng dạng số cho device.
- Bước 3: Trả về giá trị của device

Tại bước 5 trong thuật toán cơ bản, nếu trong template có chứa location, thuật toán sẽ thực hiện lựa chọn location thông qua thuật toán chọn location. Tại thuật toán, giá trị của location sẽ được lựa chọn dựa trên số lần đã xuất hiện của chúng, location nào càng ít sẽ càng có tỷ lệ được lựa chọn nhiều nhất để các tổ hợp thu được có độ đa dạng giá trị nhất có thể. Thuật toán chọn location được minh họa trong hình 3.4, quy trình diễn ra thuật toán như sau:

- Bước 1: Chọn ngẫu nhiên 1 trong 3 giá trị location có số lần xuất hiện ít nhất
- Bước 2: Chọn ngẫu nhiên giá trị name_random trong khoảng từ [0:1], nếu $name\_random < 0.3$ và location có chứa tên riêng thì thêm tên riêng cho location. Nếu $name\_random >= 0.3$ hoặc location đã được thêm tên riêng thì trả về giá trị của location.

Sau khi kết thúc thuật toán cơ sở, thu được tập các combination cơ bản bao gồm các thông tin về hành động, thiết bị và địa điểm. Các combination này sẽ được sử dụng làm đầu vào của thuật toán 4 để tiếp tục quá trình sinh combination.

Trong thuật toán thêm giá trị (minh họa hình 3.6), các combination được sinh ra từ thuật toán 1 được sử dụng làm đầu vào, nhưng combination nào có chứa các hành động "Bật", "Tăng", "Giảm" mới có thể xuất hiện thêm các giá trị changing value và target value. Các thiết bị đều được định nghĩa trước các giá trị có thể thay đổi gắn liền với chúng (ví dụ: đèn tương ứng % độ sáng, bếp tương ứng với độ C,...). Tuy nhiên chỉ có khoảng 30% cả các combination đều sẽ được bổ sung thêm các thông tin về changing value và target value. Quy trình diễn ra thuật toán như sau:

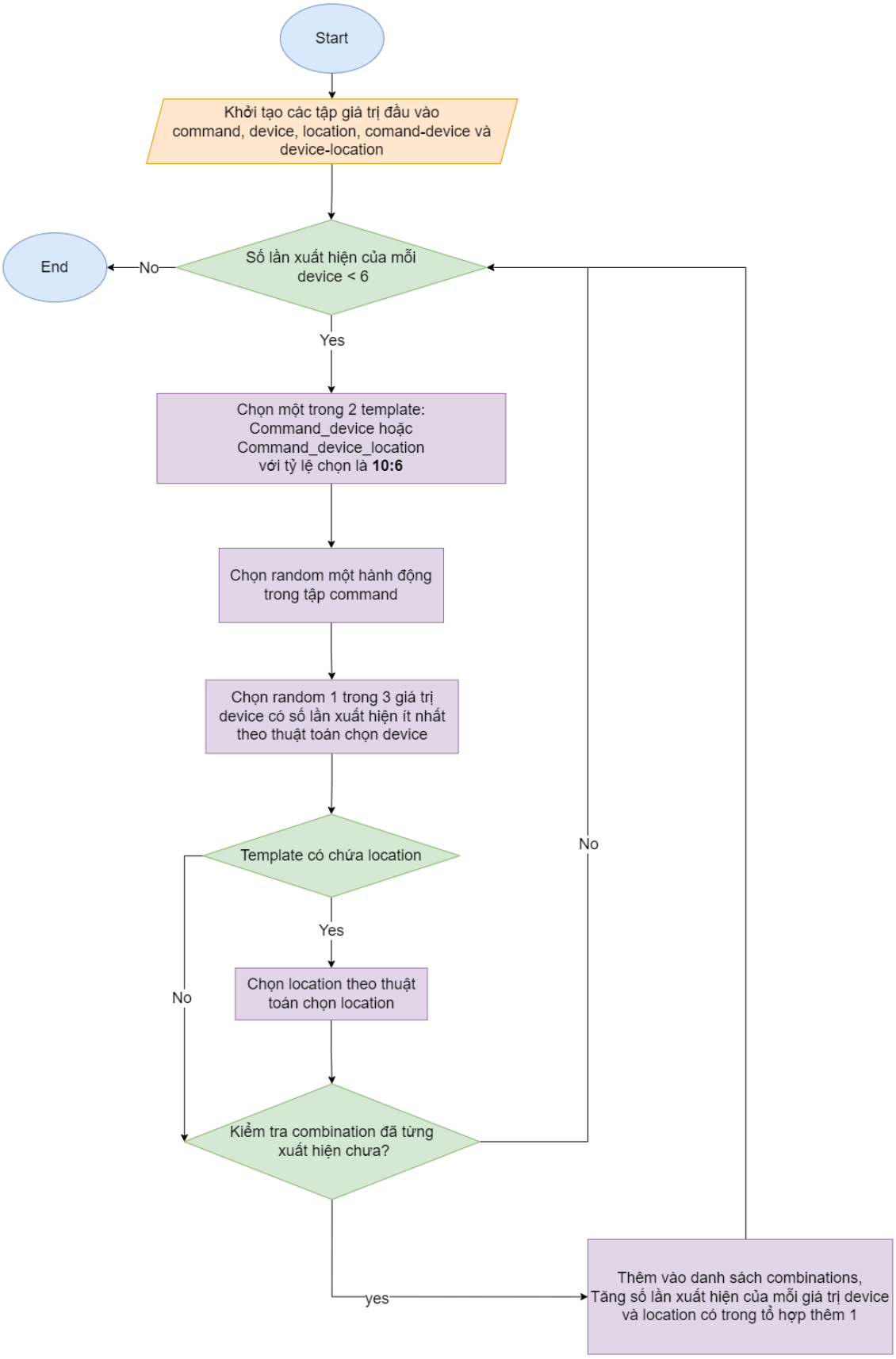
- Bước 1: Thêm tập Combinations thu được trong bước cơ sở va các tập device_changing và device_target vào.

- Bước 2: Duyệt tập combinations đầu vào
- Bước 3: Xét giá trị của command trong combination, nếu thuộc một trong các giá trị "Tăng", "Giảm", "Bật" thì tiếp tục thực hiện bước 3. Nếu không chuyển đến bước 5.
- Bước 4: Xét device trong combination có thuộc các tập set_changing hoặc set_target hay không. nếu có thêm target number hoặc changing value tương ứng với mỗi thiết bị được lưu trong tập set_changing và set_target cho combination đang xét, nếu không chuyển đến bước 5.
- Bước 5: Trả về combination.

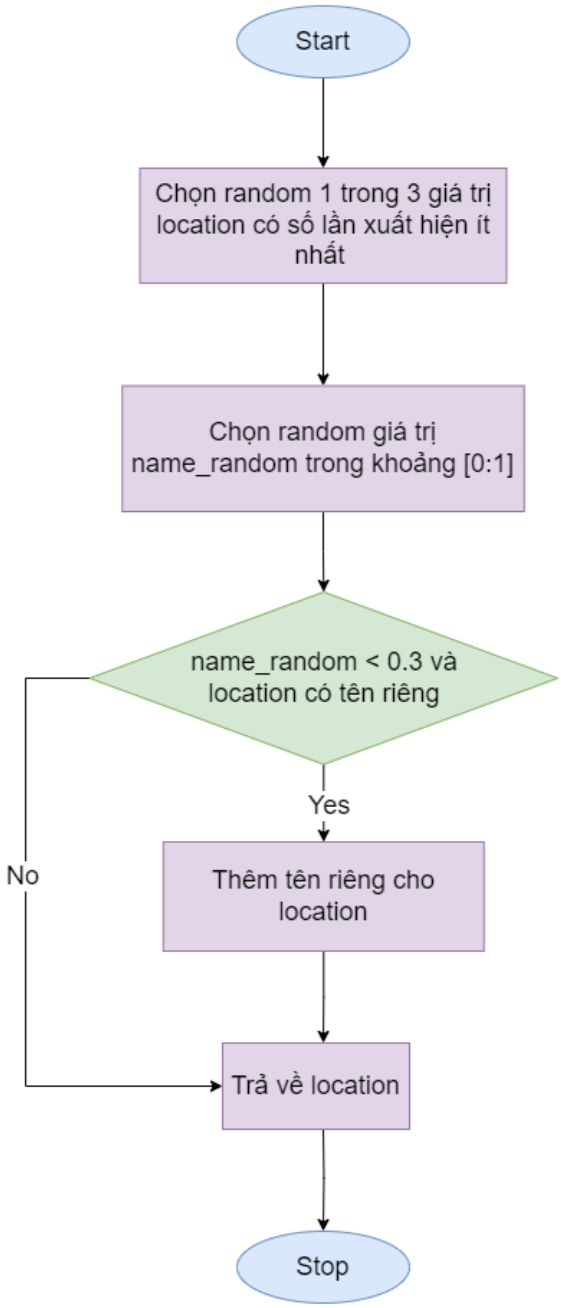
Bước cuối cùng trong quy trình sinh các combination diễn ra trong thuật toán thêm thời gian (minh họa hình 3.7). Tại đây đầu vào được sử dụng là toàn bộ combination sau khi thực hiện xong thuật toán thêm giá trị. Các thông tin về khoảng thời gian thực hiện (duration) và mốc thời gian (time at) được bổ sung vào cuối cùng. Thời gian được chọn là ngẫu nhiên nhưng vẫn sẽ nằm trong một khoảng thời gian được quy định phù hợp với mỗi thiết bị. Quy trình diễn ra thuật toán như sau:

- Bước 1: Truyền vào tập combinations đã hoàn thành trong thuật toán thêm giá trị trước đó và tập Time chứa các thiết bị có thể tác động về mặt thời gian.
- Bước 2: Thực hiện duyệt tập combinations
- Bước 3: Xét device có thuộc tập Time hay không, nếu có tiếp tục thực hiện bước 4, nếu không chuyển đến bước 6.
- Bước 5: Chọn ngẫu nhiên giá trị p trong khoảng [0:1]. Nếu $p < 0.2$ sinh giá trị timeAt, nếu $0.2 <= p < 0.3$ sinh giá trị duration, nếu $0.3 <= p < 0.35$ sinh cả 2 giá trị duration và timeAt. Sau đó thêm các giá trị sinh được vào combination. Nếu $p >= 0.35$ quay lại bước 2.
- Bước 6: Trả về combination

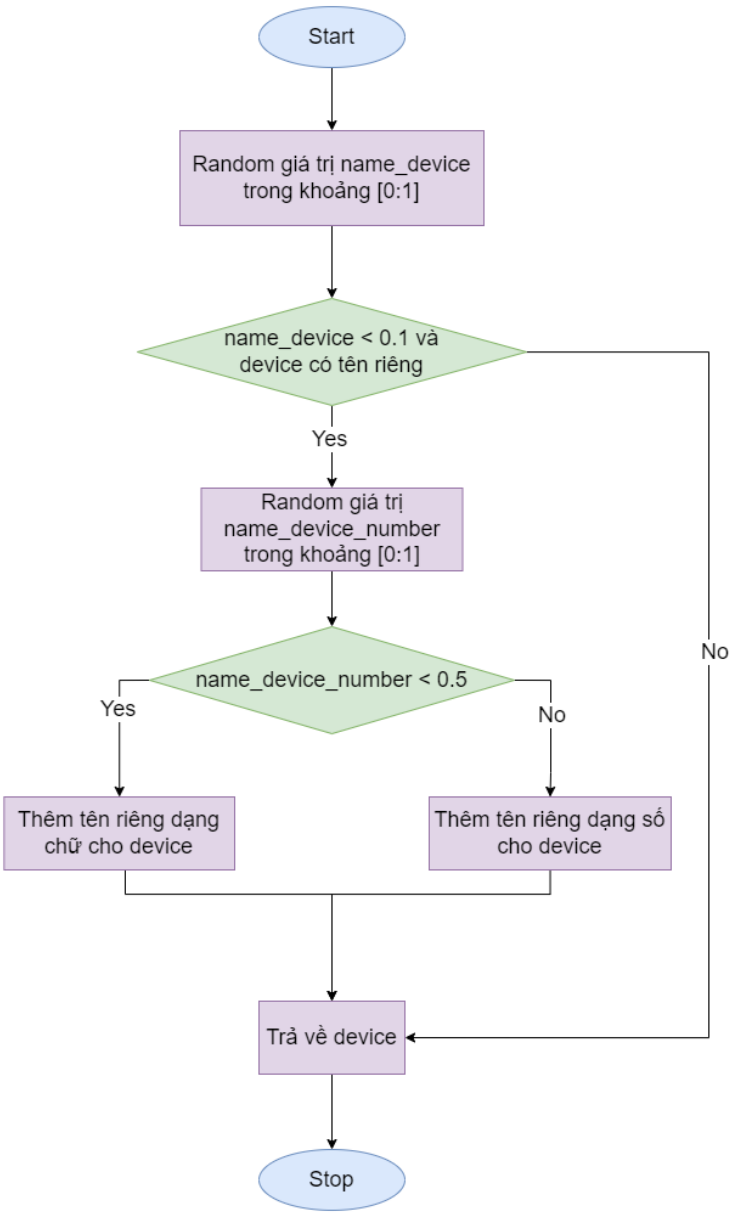
Kết quả thu được là một bộ các tổ hợp gợi ý (combination) với tổng cộng 650 combination khác nhau với độ đa dạng giá trị ở mức tối đa. Các combination này sẽ đóng vai trò là các gợi ý cho người tham gia thu âm được thực hiện trong sau là phần thu thập dữ liệu.



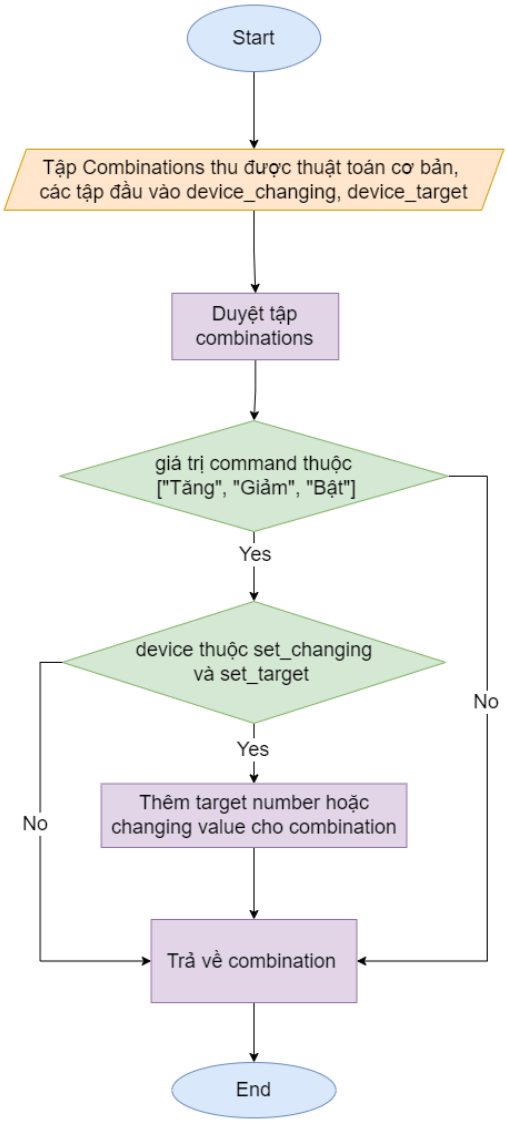
Hình 3.3: Thuật toán sinh dữ liệu cơ bản



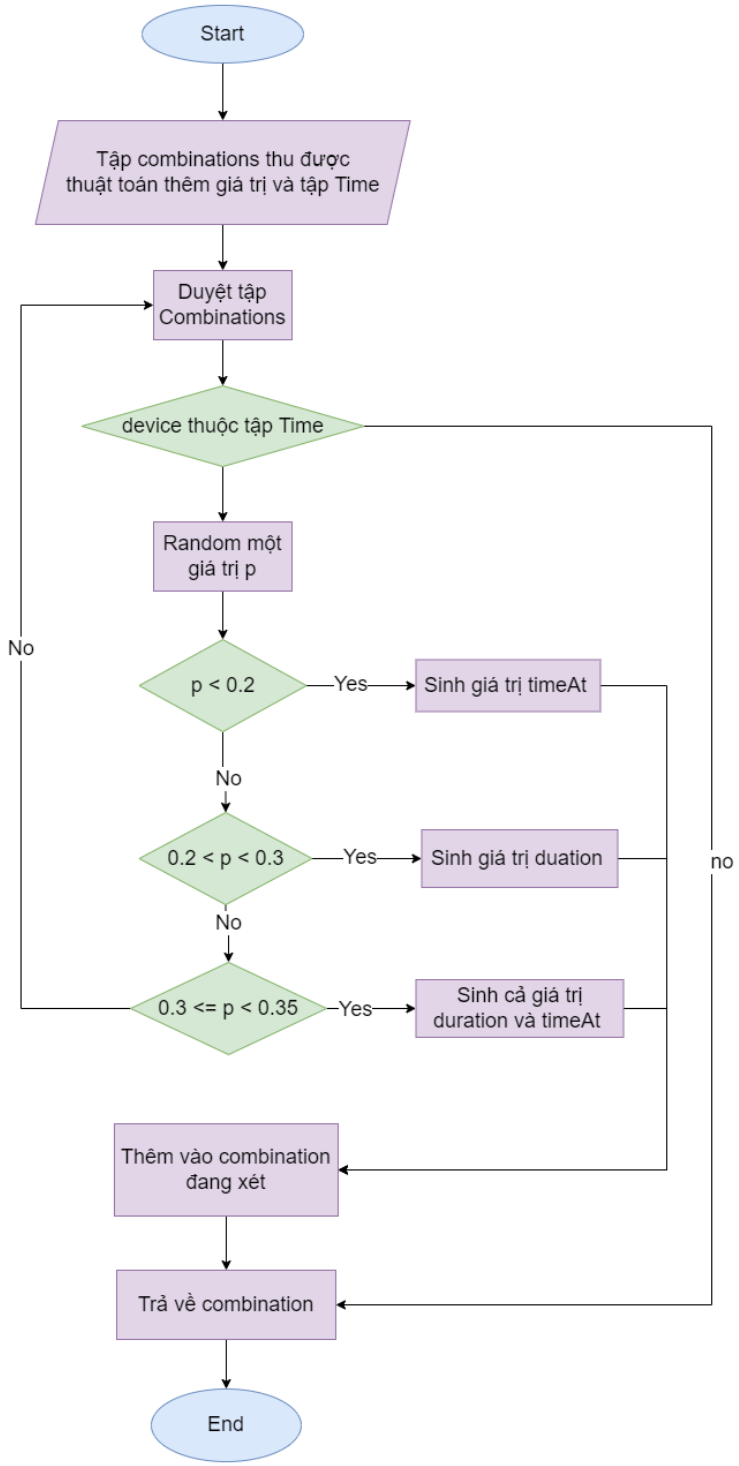
Hình 3.4: Thuật toán chọn location



Hình 3.5: Thuật toán chọn device



Hình 3.6: Thuật toán thêm giá trị



Hình 3.7: Thuật toán thêm thời gian

3.4 Thu thập dữ liệu

Bộ dữ liệu SLU-VN được thu thập dưới dạng cuộc hội thoại giao tiếp giữa người và máy với mỗi cuộc hội thoại sẽ bao gồm 2 người thực hiện, một người đóng vai trò là người chủ nhà và người còn lại đóng vai trò là trợ lý ảo. Chủ nhà sẽ là người đưa ra các yêu cầu cho trợ ý ảo và trợ lý ảo sẽ là người thực hiện nó. Vai trò cụ thể của 2 nhân vật như sau:

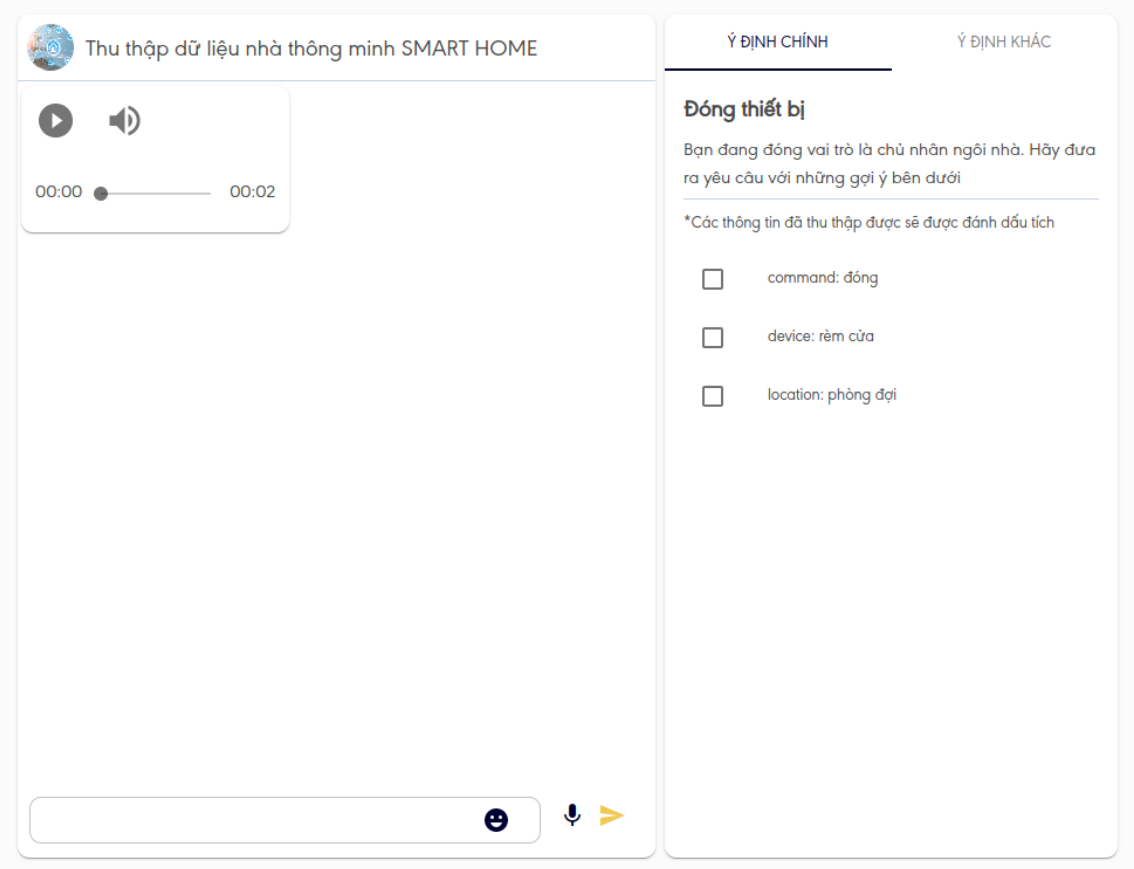
Chủ nhà: Đây là người sẽ đưa ra câu nói của mình để thực hiện hành động với các thiết bị trong nhà. Họ sẽ được cung cấp các thông tin bao gồm: hành động (command), thiết bị (device), địa điểm thực hiện (location), hoạt cảnh (scene). Người nói phải đưa ra câu nói một cách tự nhiên nhất có thể và phải chứa đầy đủ các thông tin đã được cung cấp.

Trợ lý ảo: đây là người cần phải nghe lại câu nói của người cặp với mình, sau khi nghe xong người nghe cần kiểm tra lại phần văn bản của câu nói đó được sinh ra, nếu chưa chính xác với câu nói thì cần chỉnh sửa lại cho đúng, sau đó chọn đúng các thông tin bao gồm: hành động (command), thiết bị (device), địa điểm thực hiện (location),hoạt cảnh (scene) đã nghe được ở câu nói đó. Nếu câu nói của người người đồng hành chưa bao gồm đủ các thông tin như gợi ý, trợ lý có thể hỏi lại người đồng hành của mình để họ cung cấp thêm thông tin.

Trong quá trình thu âm những người tham gia sẽ tìm và kết hợp ngẫu nhiên với nhau tạo thành một cặp chủ nhà - trợ lý ảo, sau đó quá trình thực hiện thu âm sẽ diễn ra như sau:

a) Vai trò chủ nhà

1. Bước 1: Chủ nhà sẽ nhận được các gợi ý của câu nói bao gồm: ý định (intent), hành động (command), địa điểm (location), khoảng thời gian (duration),... thay vì được cung cấp 1 câu nói được tạo sẵn, điều này giúp cho các câu nói sẽ có độ đa dạng và tự nhiên do mỗi người sẽ có một cách nói khác nhau nhưng vẫn bảo đảm được độ chính xác bởi vì chủ nhà vẫn phải đảm bảo nói đúng và đủ tất cả các thông tin gợi ý được cung cấp. Hình 3.8 mô tả giao diện của người chủ nhà như sau: thứ nhất là vai trò của người tham gia là chủ nhà trong đoạn hội thoại. Thứ hai là các gợi ý của đoạn hội thoại mà bạn được cung cấp. Các gợi ý bao gồm “intent” là mục đích chính của đoạn hội thoại, "device" là thiết bị thực hiện hành động, "command" hành động,... những gợi ý trên chính là các tổ hợp gợi ý đã sinh được ở phần 3.3. Ngoài ra, các thông tin đã xuất hiện trong cuộc hội thoại sẽ được tích vàng. Sau khi nắm được các thông tin gợi ý và suy nghĩ xong câu cần nói. Chủ nhà sẽ thực hiện thu âm để nói chuyện qua lại với trợ lý ảo bằng cách bấm vào biểu thực micro này để thu âm



Hình 3.8: Giao diện thu âm của chủ nhà

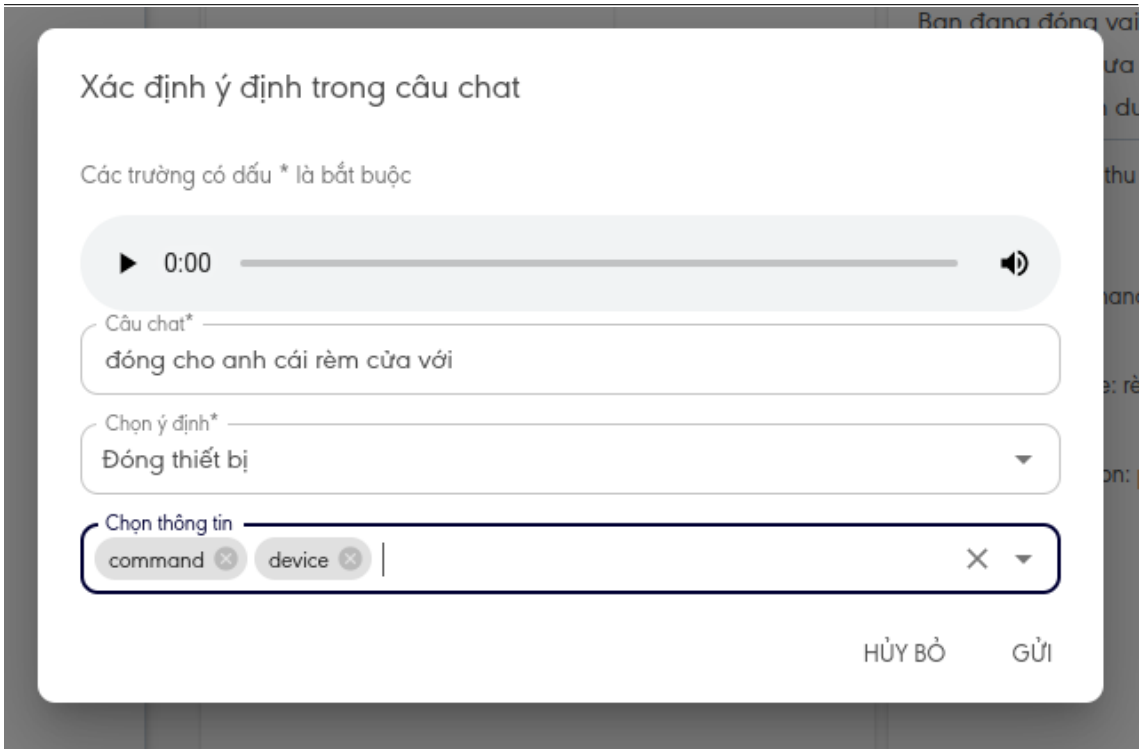
câu nói. Sau khi thu âm xong câu nói chủ nhà bấm gửi câu nói cho trợ lý để xác nhận.

2. Bước 2: Sau khi chủ nhà bấm gửi câu nói sang cho trợ lý ảo cần phải thực hiện một bước xác minh thông tin như hình 3.9 dưới đây:

   Đầu tiên chủ nhà cần nghe lại và nhập chính xác câu nói của mình vào ô "Câu chát". Thứ hai, chủ nhà cần chọn ý định chính cho câu nói của mình ở mục "Chọn ý định". Cuối cùng, chủ nhà chọn các trường thông tin xuất hiện trong câu nói của mình ở mục "Chọn thông tin"

   Sau khi đã xác minh toàn bộ thông tin, chủ nhà bấm "send" để gửi câu nói của mình cho người trợ lý ảo.

3. Bước 3: Người đóng vai trò trợ lý ảo sẽ nhận được yêu cầu của chủ nhà và thực hiện các bước xác minh. Sau khi xác thực hoàn thành và xác nhận câu nói của chủ nhà có đúng ý định và chứa đầy đủ các thông tin, chủ nhà sẽ được thông báo là câu nói thu âm thành công như hình 3.10. Các thông tin được xác nhận sẽ được đánh tích vàng ở bên cạnh. Nếu câu nói được trợ lý ảo xác minh là chưa đúng ở bất ký trường thông tin nào thì câu nói sẽ bị hủy và chủ nhà phải thực hiện lại câu nói đó.

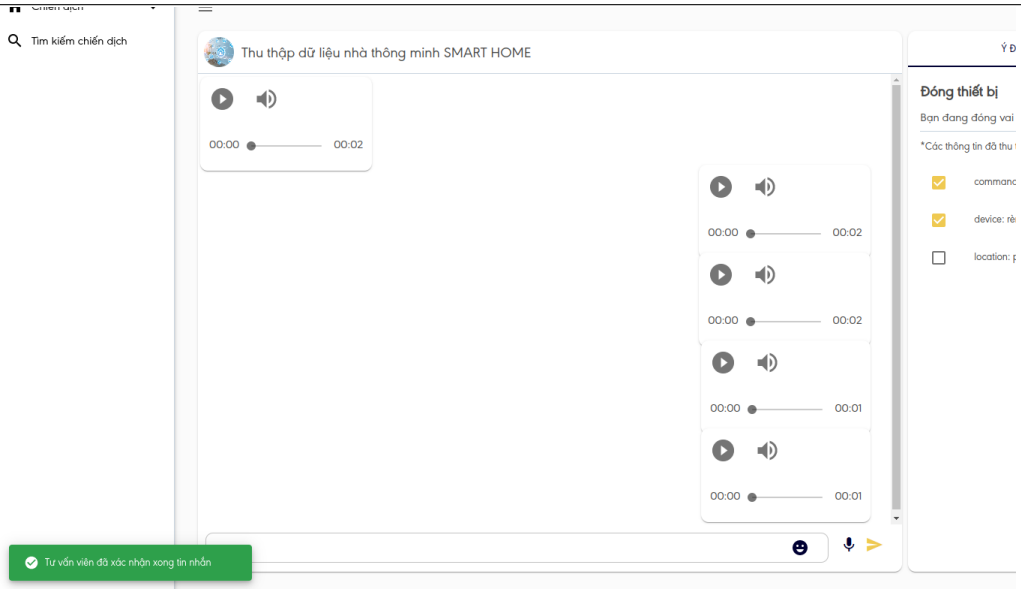


Hình 3.9: Giao diện xác minh thông tin của chủ nhà

Nếu chưa nói đủ thông tin gợi ý cung cấp hoặc nói sai chủ nhà cần quay lại bước 1 và nói tiếp yêu cầu cho các thông tin còn thiếu đến khi tất cả thông tin gợi ý được cung cấp đều đã được xác nhận là xuất hiện. Sau khi hoàn thành cuộc hội thoại sẽ được thông báo thành công như hình 3.11.

b) Vai trò trợ lý ảo

1. Bước 1: Sau khi được ghép cặp với một người chủ nhà trợ lý ảo sẽ ở trạng thái chờ trong khi chủ nhà đưa ra câu nói của mình, tuy nhiên trợ lý ảo cũng có thể đưa ra câu nói theo kiểu chào hỏi trước cho phía chủ nhà. Tại giao diện chờ, trợ lý ảo sẽ được cung cấp các thông tin hỗ trợ quá trình xác minh và các chức năng bao gồm: Đầu tiên là vai trò của mình trong cuộc hội thoại, thứ hai là ý định chính của hội thoại và các trường thông tin sẽ có trong cuộc hội thoại (không biết giá trị của thông tin), thứ ba là biểu tưởng micro chính là nút ghi âm lại câu nói, cuối cùng là biểu tượng nút mũi tên để gửi câu nói (minh họa mình 3.12).
2. Bước 2: Khi nhận được câu nói từ phía chủ nhà, trợ lý ảo sẽ thực hiện các bước xác minh. Bước xác minh đầu tiên, giao diện của trợ lý ảo sẽ hiện như hình 3.13, trợ lý ảo sẽ nghe câu nói của chủ nhà sau đó kiểm tra phần văn bản của câu nói đã chính xác hay chưa, nếu chưa chính xác thì cần sửa lại câu nói vào ô sửa ở ngay bên dưới, nếu chính xác rồi thì không cần sửa gì. Lựa chọn ý định của câu nói mà mình nghe được ở ô "Chọn ý định", nếu ý định của trợ lý

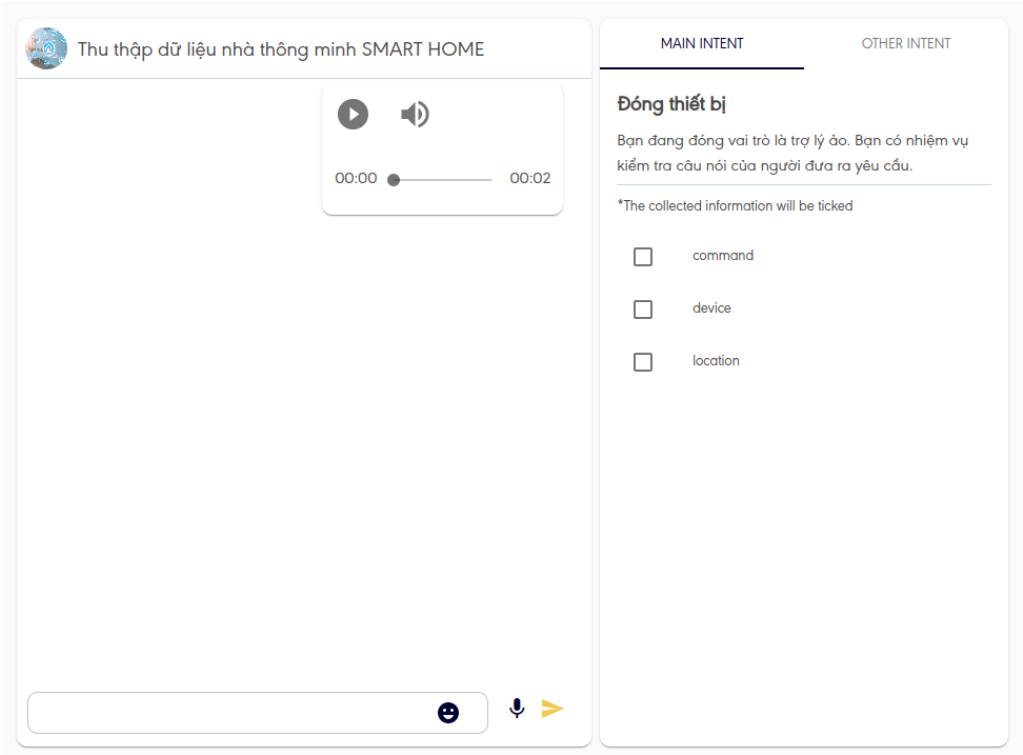


Hình 3.10: Giao diện thông báo đã xác nhận câu nói thành công của chủ nhà

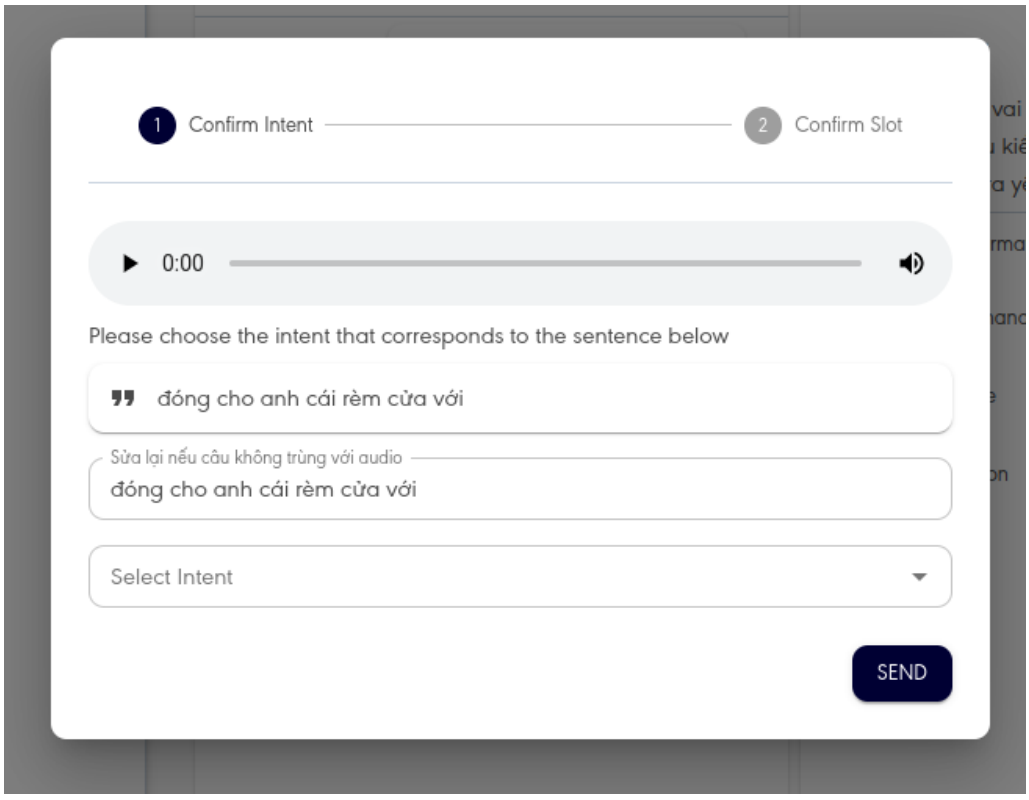
Hình 3.11: Giao diện thu âm thành công cuộc hội thoại

nghe được và lựa chọn không trùng khớp với cả ý định mà chủ nhà lựa chọn câu nói sẽ bị hủy và chủ nhà sẽ phải thực hiện lại câu nói còn trợ lý ảo sẽ quay lại trạng thái chờ. Nếu ý định của 2 bên là trùng khớp sẽ chuyển qua bước xác minh thứ hai (Hình 3.14).

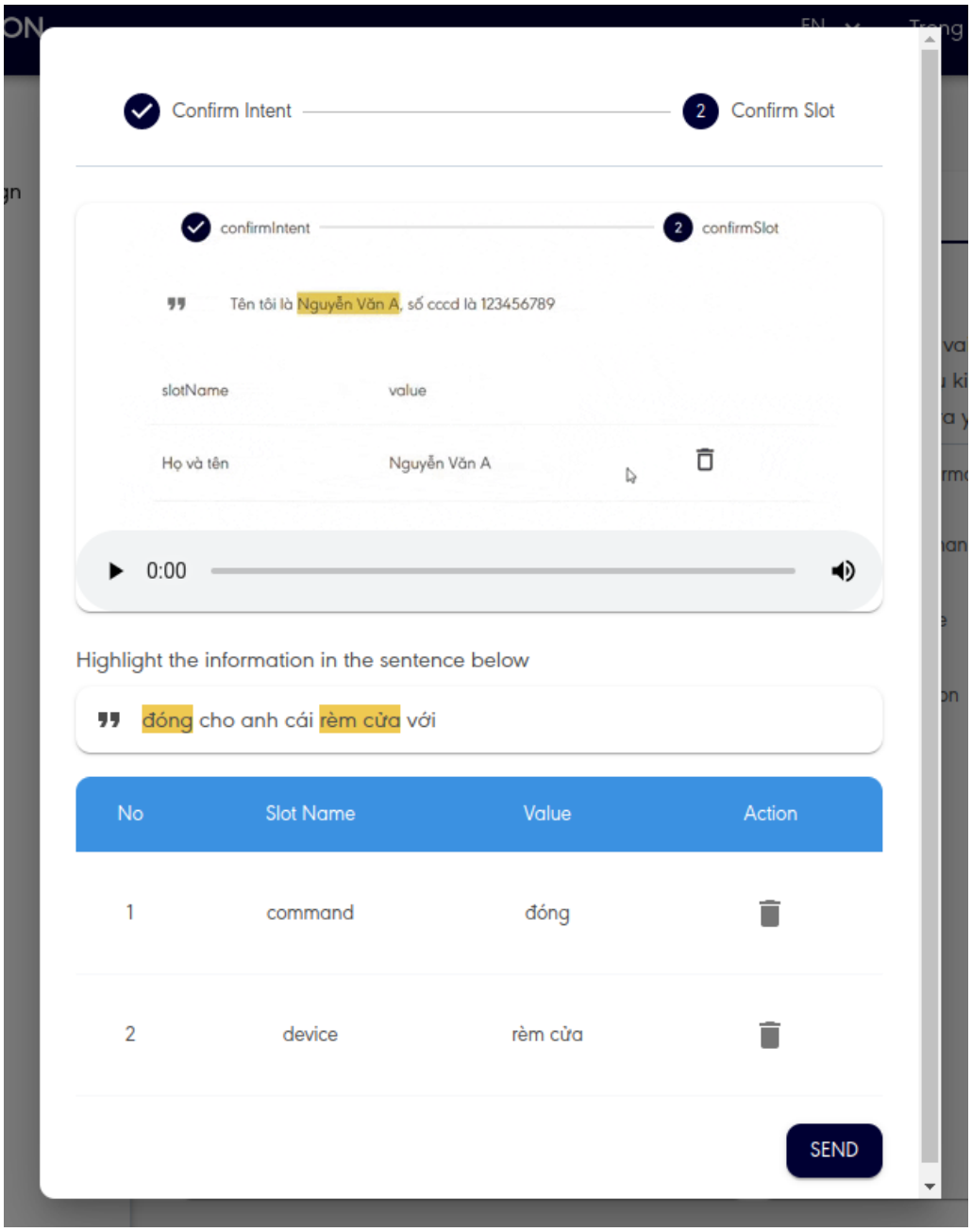
Tại bước này, trợ lý ảo sẽ thực hiện xác minh các thông tin gợi ý có trong câu nói. Tại ô văn bản của câu nói trợ lý ảo thực hiện bôi đen để đánh nhãn giá trị của các trường thông tin gợi ý được cung cấp ban đầu, sau khi hoàn thành tất cả thông tin bấm "Gửi" (Hình 3.14).



Hình 3.12: Giao diện chờ của trợ lý ảo



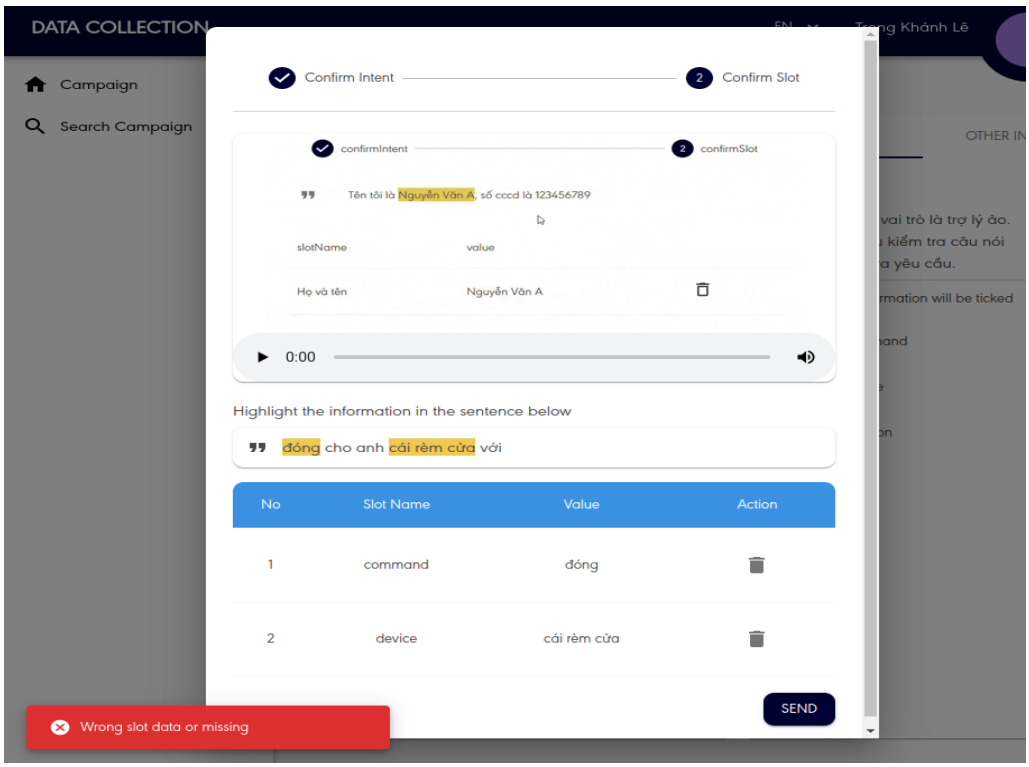
Hình 3.13: Giao diện bước xác minh thông tin thứ nhất của trợ lý ảo



Hình 3.14: Giao diện bước xác minh thông tin thứ hai của trợ lý ảo

3. Bước 3: Nếu tất cả giá trị đều trùng khớp với thông tin mà chủ nhà cung cấp câu nói của chủ nhà sẽ được xác thực là chính xác và thu âm thành công. Nếu có bất kỳ giá trị nào bị sai, sẽ được thông báo sai thông tin (Hình 3.15) và phải đánh lại nhãn một lần nữa, nếu sau 3 lần sai câu nói sẽ bị hủy và chủ nhà sẽ phải thực hiện lại câu nói còn trợ lý ảo sẽ quay lại trạng thái chờ. Hoặc câu nói trước của chủ nhà vẫn chưa đầy đủ thông tin gợi ý cung cấp thì trợ lý ảo cũng sẽ quay lại trạng thái chờ người chủ nhà tiếp tục ra yêu cầu.

   Nếu tất cả trường thông tin đã được xác minh đầy đủ cuộc hội thoại được tính là thành công. (Hình 3.16)



Hình 3.15: Giao diện thông báo thông tin bị sai

Quy trình thu âm cơ bản của một cuộc hội thoại được mô tả hoàn chỉnh với quy trình dưới đây.

1. Bước 1: Chủ nhà ra yêu cầu đưa ra yêu cầu. Ví dụ: "Em mới bật cho anh cái bình nóng lạnh lên mức 100 độ C nhé, tý nữa anh tắm".
2. Bước 2: Chủ nhà xác nhận các thông tin mình nói là chính xác và gửi cho trợ lý ảo.
3. Bước 3: Trợ lý ảo nhận và nghe câu nói, thực hiện 2 bước xác mình câu nói, nếu thông tin là không trùng khớp giữa 2 người, chủ nhà phải thực hiện lại câu nói. Nếu thông tin là trùng khớp câu nói được tính thành công. Nếu thông tin chủ nhà cung cấp trong câu nói trước đó chưa đầy đủ tất cả thông tin gợi ý

Hình 3.16: Giao diện thu âm thành công cuộc hội thoại

thực hiện tiếp bước 4. Nếu đã đủ thông tin hội thoại được tính đã hoàn thành và kết thúc.

4. Bước 4: Trợ lý ảo đặt lại câu hỏi cho chủ nhà về các thông tin còn thiếu. Trong ví dụ ở bước 1 nếu có thêm gợi ý về địa điểm nhưng chủ nhà lại chưa nói thì trợ lý ảo cần hỏi lại chủ nhà thêm thông tin về địa điểm. Câu hỏi có thể là: "Anh muốn bật bình nóng lạnh ở đâu ạ?"
5. Bước 5: Chủ nhà tiếp tục đưa ra yêu cầu đi kèm địa điểm mong muốn để cung cấp thêm các thông tin còn thiếu với cách thức như trong bước 1. Ví dụ: "Bật cho anh cái bình nóng lạnh ở nhà tắm tầng 1 nhé.". Sau đó lại xác minh thông tin như bước 2 và gửi cho trợ lý ảo.
6. bước 6: Trợ lý ảo thực hiện công việc như bước 3.

Một cuộc hội thoại sẽ có các bước cơ bản như vậy trong đó các bước 5, 6 thực chất là thực hiện lại vòng lặp của các bước 1, 2, 3, vòng lặp các bước 4, 5, 6 sẽ tiếp tục cho đến khi thông tin gợi ý được nói đầy đủ, nếu có sự sai sót xảy ra tại bước nào quá trình sẽ quay lại bước 1. Tại bước đặt câu nói của chủ nhà, nếu số trường thông tin gợi ý là nhiều (có thể 3, 4 hoặc 5 trường thông tin) thì chủ nhà có thể không nói hết các thông tin trong 1 câu vì sẽ làm cho câu nói quá dài và thiếu tự nhiên. Một cuộc hội thoại được thu thành công nếu tất cả các trường thông tin gợi ý của đều được xác nhận đầy đủ và chính xác vì vậy quy trình thu âm sẽ có thể lặp lại nhiều

lần cho đến khi tất cả thông tin đã được xác nhận. Tuy nhiên cũng sẽ có các cuộc hội thoại có ít thông tin gợi ý và chỉ cần 1 câu nói đã bao quát toàn bộ thông tin và vai trò của trợ lý ảo lúc này là xác minh độ chính xác trong câu nói của chủ nhà, nếu không có sai sót cuộc hội thoại tính là thành công.

3.5 Tiền xử lý dữ liệu

Mặc dù thu âm dưới dạng cuộc hội thoại nhưng trong đồ án lần này tác giả chỉ xử lý dữ liệu dưới dạng câu đơn để phù hợp với mô hình triển khai. Vì trong quá trình thu âm việc kiểm tra các trường thông tin được kiểm tra qua lại giữa 2 người tham gia và chỉ khi chính xác, đầy đủ mọi thông tin cuộc hội thoại mới tính là thành công nên quá trình tiền xử lý dữ liệu cũng sẽ không quá phức tạp.

Tại phần này tác giả sẽ sử dụng mô hình Automatic Speech Recognition (ASR) đã được huấn luyện trên tiếng Việt để kiểm tra tệp ghi âm của mỗi có đủ chất lượng hay không. Tại đây các file âm thanh nào trả về kết quả tỉ lệ lỗi của từ là WER[1] > 0.7 đều bị lược bỏ.

3.6 Bộ dữ liệu VN-SLU

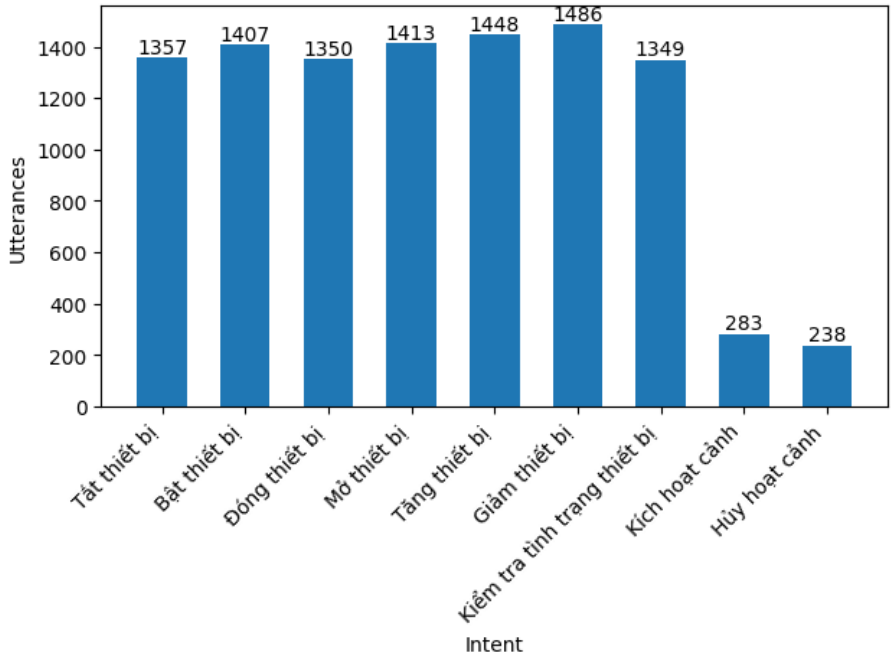
3.6.1 Bộ dữ liệu VN-SLU

Sau các quá trình từ chuẩn bị dữ liệu cho đến xử lý dữ liệu, bộ dữ liệu hiểu ngôn ngữ dạng nói tiếng Việt với chủ để tương tác với các thiết bị trong nhà thông minh đã hoàn thành và tạm đặt tên là VN-SLU. Bộ dữ liệu gồm các hiệu lệnh tương tác giữa người và các thiết bị thông minh trong nhà, hiệu lệnh trong bộ dữ liệu cũng rất đa dạng với tập các tổ hợp thông tin gợi ý lên tới hơn 600 tổ hợp khác nhau. Cuối cùng bộ dữ liệu xây dựng được bao gồm khoảng 10000 câu nói với sự tham gia của 130 người tham gia thu âm, có tổng thời lượng khoảng 12 tiếng và bao gồm 9 ý định (intent), 8 thực thể (entity).

3.6.2 Thống kê và phân tích dữ liệu

a, Phân bố của các ý định

Bộ dữ liệu VN-SLU bao gồm 9 ý định khác nhau là bật thiết bị, tắt thiết bị, đóng thiết bị, mở thiết bị, tăng thiết bị, giảm thiết bị, kiểm tra tình trạng thiết bị, kích hoạt cảnh và hủy hoạt cảnh. Số lượng câu nói của mỗi ý định được minh họa trong hình 3.15 là khá tương đồng, giao động trong khoảng 1350 đến 1490, riêng chỉ có 2 ý định là kích hoạt cảnh và hủy hoạt cảnh sẽ ít hơn do 2 ý định này trong thực tế có mật độ xuất hiện ít hơn và ý nghĩa của chúng trong câu nói cũng mang tính trừu tượng cao hơn tương đối so với các ý định khác.



Hình 3.17: Phân bố số lượng câu nói của mỗi ý định

b, Phân bố của thực thể

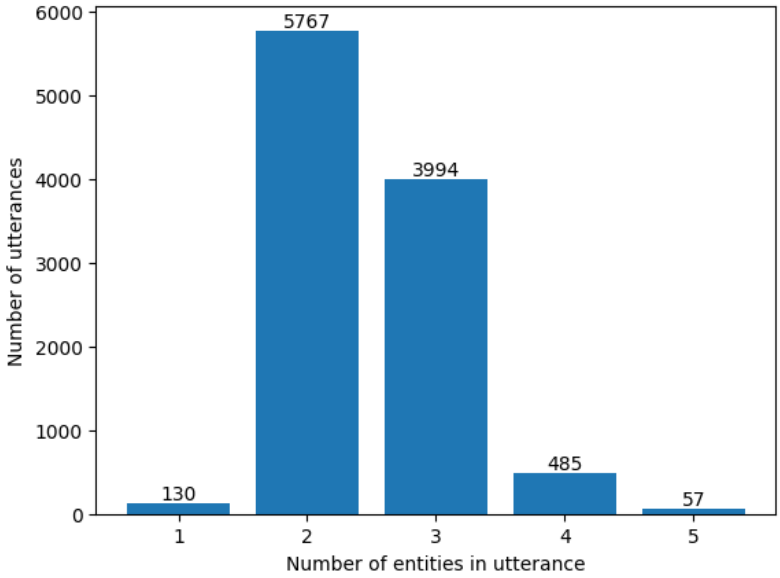
Trong hình 3.16, có thể thấy rõ số lượng câu có 1 thực thể hoặc 4, 5 thực thể có số lượng rất ít khi so sánh với số lượng câu có 2 hoặc 3 thực thể. Khi giao tiếp giữa người với người hay giữa người với máy trong thực tế thì số lượng thực thể xuất hiện trong một câu thường sẽ chỉ từ 2 đến 3 thực thể, điều này chứng minh các câu nói trong bộ dữ liệu về mặt hình thức đang có tính tự nhiên tương tự với thực tế. Trong bộ dữ liệu mà tác giả đề suất đối với câu có 2 thực thể sẽ thường là hành động và thiết bị, còn với câu có 3 thực thể sẽ đa dạng hơn là hành động, thiết bị kết hợp với địa điểm hoặc các giá trị thời gian, mức độ tính năng muốn thay đổi trên thiết bị.

c, Phân bố các thiết bị và địa điểm

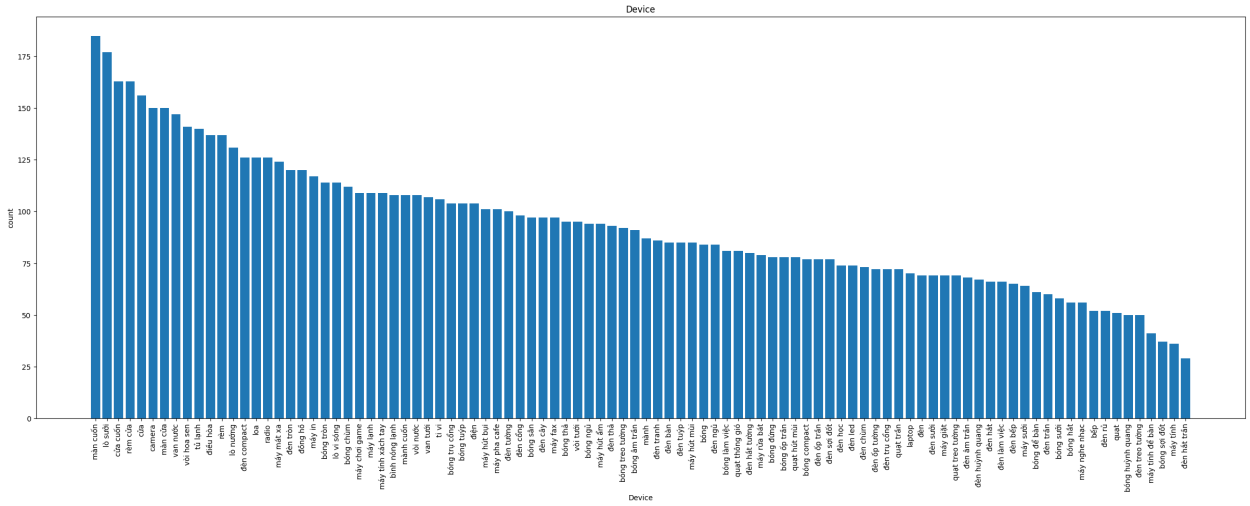
Trong hình 3.17, có thể thấy phân bố mật độ xuất hiện của các thiết bị không bị đồng đều mà có sự chênh lệch. Dựa vào thực tế, các thiết bị nào có khả năng và tần suất sử dụng nhiều trong mỗi ngôi nhà sẽ có số lượng xuất hiện trong các câu nói ít hơn. Các thiết bị như cửa, rèm, tủ lanh, điều hòa, các loại bóng đèn có số lượng xuất hiện trong các câu nói lớn nhất, điều này là đúng bì các thiết bị này có thể xuất hiện ở bất cứ căn nhà nào nên trong một hệ thống nhà thông minh chúng sẽ được ra lệnh điều khiển phổ biến là điều hiển nhiên. Ngoài ra các thiết bị như quạt, máy tính hay bếp có mật độ xuất hiện thấp nhất.

Ngoài các thiết bị, sự xuất hiện của các địa điểm trông một câu nói cũng có tỉ lệ

[1]https://en.wikipedia.org/wiki/Word$_e rror_r ate$



Hình 3.18: Phân bố số lượng thực thể (entities) trong mỗi câu nói

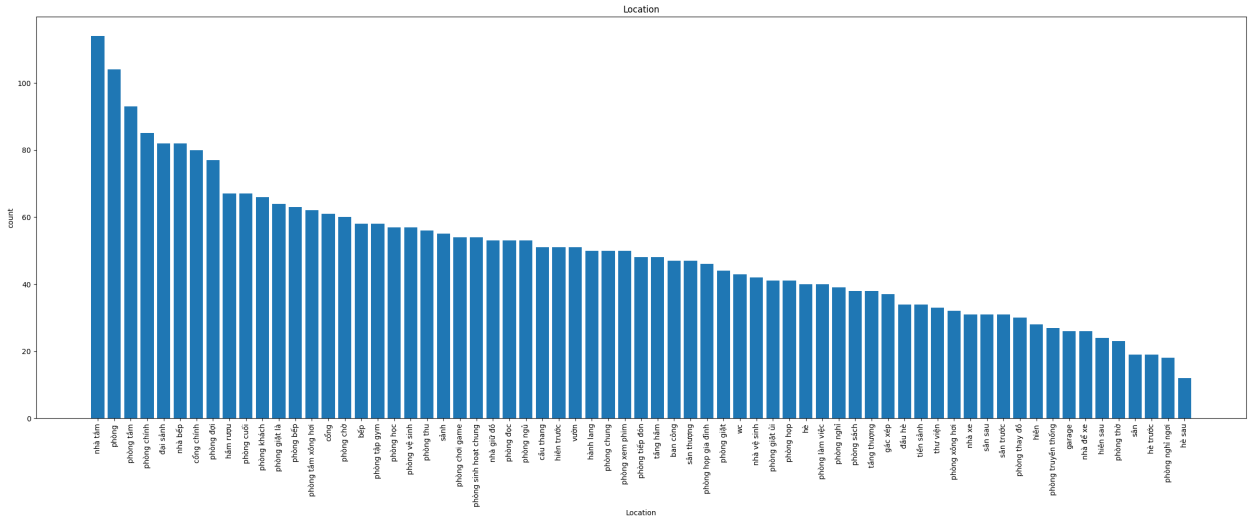


Hình 3.19: Phân bố thiết bị trong mỗi câu nói

khá cao. Để câu nói được tự nhiên và sát với thực tế nhất, các địa điểm có mật độ xuất hiện của các thiết bị cao sẽ có nhiều tỉ lệ xuất hiện trong các câu nói nhất, từ hình 3.18 có thể thấy các khu vực như nhà tắm, phòng nói chung, bếp có số lượng các câu nói nhiều nhất, những nơi này cũng là nơi có thể chứa nhiều thiết bị. Còn những khu vực như sân, hè, một số phòng chuyên dụng, nhà xe thường không có quá nhiều các thiết bị cần tương tác nên chúng sẽ có khả năng xuất hiện trong các câu kém hơn.

3.7 Tổng kết

Trong Chương 3 của đồ án, tác giả đã trình bày quá trình xây dựng một bộ dữ liệu hiểu ngôn ngữ dạng nói đối với tiếng Việt với chủ đề là các hành động, yêu cầu giữa người dùng với các thiết bị trong nhà thông minh. Bộ dữ liệu VN-SLU



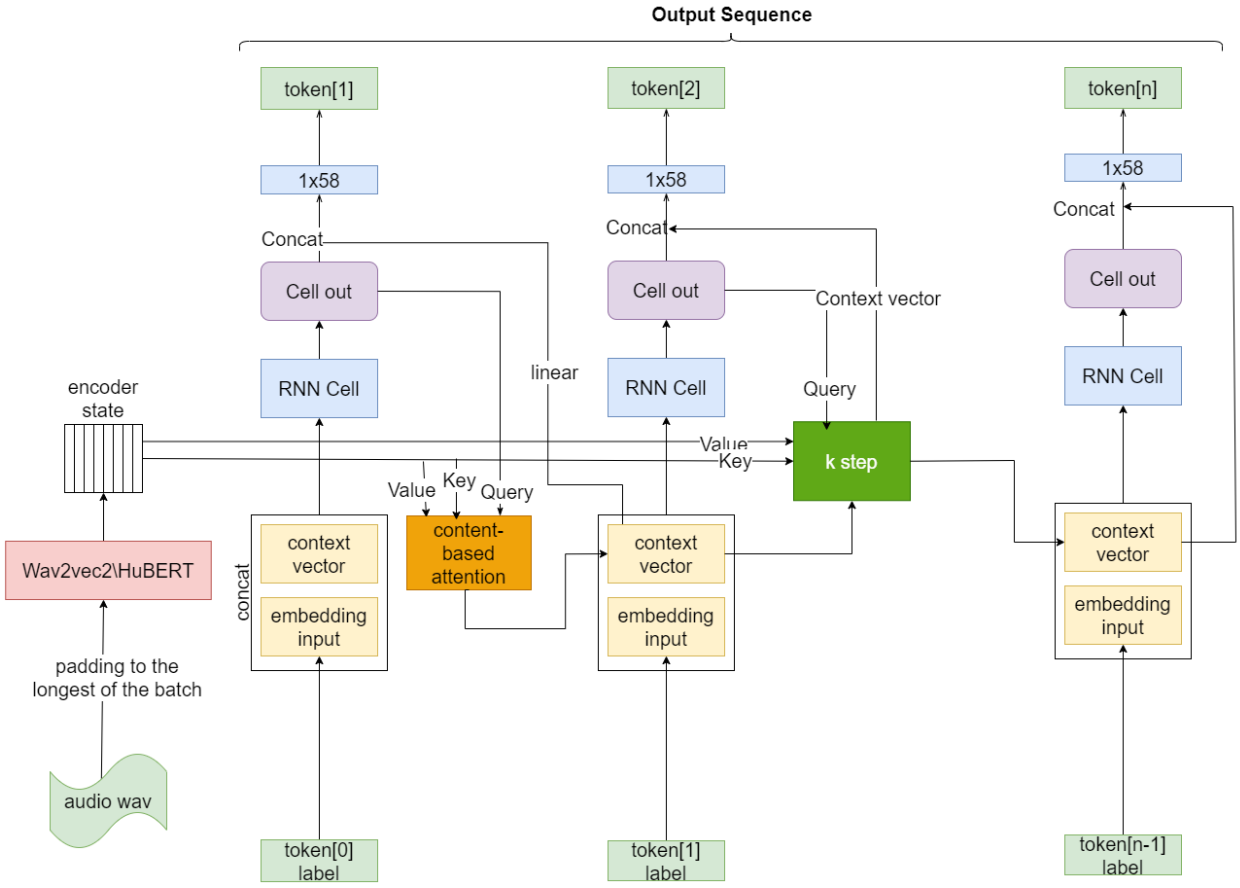
Hình 3.20: Phân bố địa điểm xuất hiện trong các câu nói

được thu âm dựa trên các gợi ý có sẵn được xây dựng thông qua các khảo sát thực tế, từ đó xây dựng thuật toán sinh các tổ hợp gợi ý với quy tắc chặt chẽ để thu được một bộ dữ liệu chất lượng. Với dung lượng khoảng 10000 câu và 12 giờ thì bộ dữ liệu VN-SLU đang có lượng dữ liệu không hề kém cạnh so với nhiều bộ dữ liệu SLU khác trên thế hiện nay và để đánh giá chất lượng đi kèm việc thực hiện các thử nghiệm trên bộ dữ liệu mới này tác giả sẽ thực nghiệm các mô hình và đưa ra một số đề xuất cải tiến trên bộ dữ liệu VN-SLU trong chương 4 của đồ án.

CHƯƠNG 4. ĐỀ XUẤT MÔ HÌNH SLU CHO TIẾNG VIỆT

4.1 Mô hình cơ sở

Mô hình cơ sở được tác giả sử dụng trong đồ án lần này là mô hình End-to-end spoken language understanding của 3 tác giả Yingzhi Wang, Abdelmoumene Boumadane, Abdelwahab Heba đồng nghiên cứu và công bố năm 2021 trong bài báo A Fine-tuned Wav2vec 2.0/HuBERT Benchmark For Speech Emotion Recognition, Speaker Verification and Spoken Language Understanding [29].



Hình 4.1: Kiến trúc mô hình huấn luyện cơ sở

Kiến trúc mô hình được tác giả nghiên cứu và biểu diễn lại trong hình 4.1. Mô hình cơ sở được chọn là một mô hình End-to-end SLU dưới dạng Sequence-to-Sequence với 2 khối chính là encoder và decoder, khối encoder được sử dụng là mô hình pre-trained Wav2vec2/HuBERT **HuBERT**, [11]. Tiếng nói khi qua mô hình encoder sẽ được trích xuất và mã hóa gọi là trạng thái mã hóa (encoder states), từ encoder states sẽ được sử dụng làm đầu vào của khối decoder. Khối decoder được sử dụng trong mô hình cơ sở là kiến trúc kết hợp của lớp RNN và cơ chế chú ý (Attention) [6], đầu vào của khối gồm 2 phần là encoder states và nhãn ký tự (token label), bên trong khối decoder sẽ được chia thành nhiều khối giải mã nhỏ

hơn. Mỗi khối giải mã nhỏ sử dụng đầu vào là một token label và encoder states, token label sẽ cộng với một đặc trưng khác là vec-tơ ngữ cảnh (context vector) qua lớp RNN để làm đầu vào của khối attention. Khối attention được sử dụng trong mô hình là content-based attention đã được áp dụng trong mạng nơ-ron dịch máy (neural machine translation) [21] và có thể được áp dụng trực tiếp trong mô hình này. Đặt encoder states là chuỗi các đặc trưng h = $(h_1, h_2, ..., h_T)$ với T là biến thời gian , context vector c_i được tính bằng tổng các trọng số của h_i:

$$c_i = \Sigma_{j=1}^{t} \alpha_{i,j} \, h_j \tag{4.1}$$

Trong đó trọng số chú ý α_{ij} của mỗi đặc trưng h_j được tính theo công thức hàm softmax là:

$$\alpha_{i,j} = \frac{\exp\left(e_{i,j}\right)}{\sum_{j=1}^{T} \exp\left(e_{i,j}\right)} \tag{4.2}$$

với

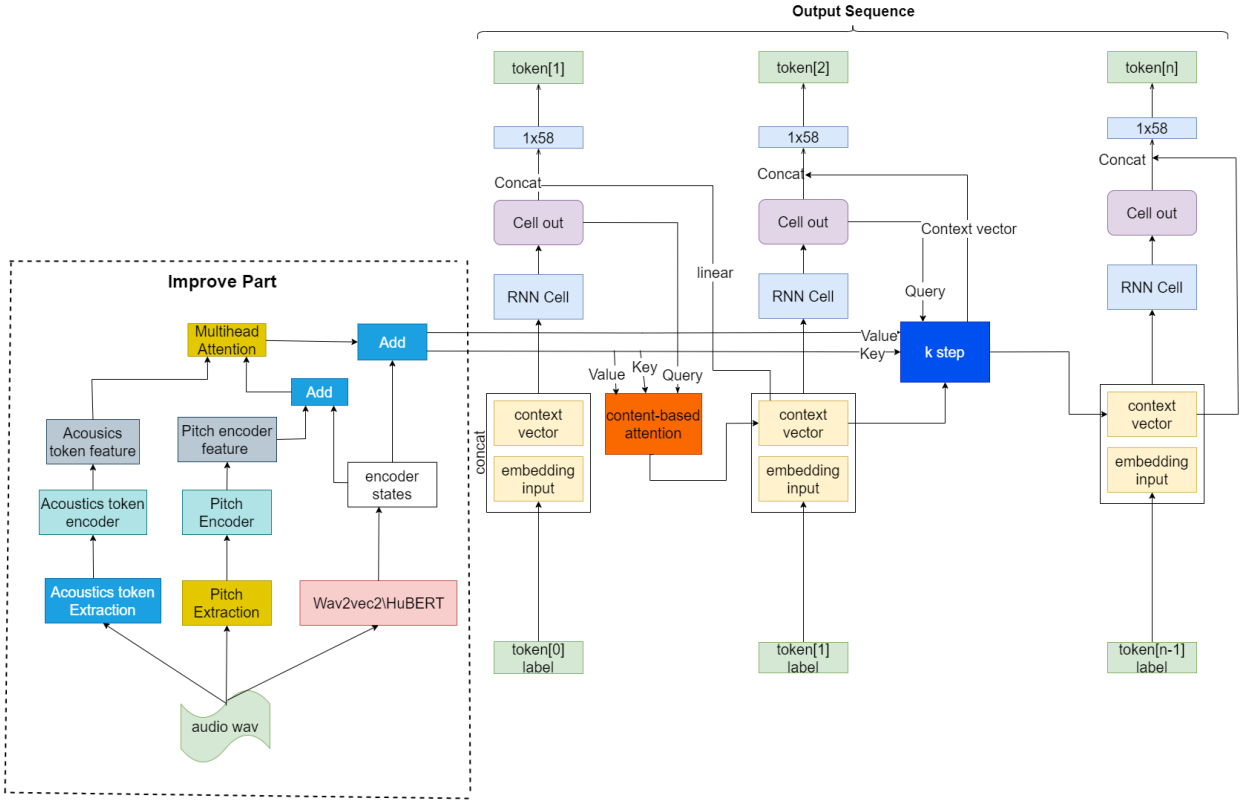
$$e_{i,j} = \mathbf{w}^{\top} \tanh\left(\mathbf{W}\mathbf{s}_{i-1} + \mathbf{V}\mathbf{h}_j + \mathbf{b}\right) \tag{4.3}$$

Với s_{i-1} và h_j lần lượt là vec-tơ trạng thái giải mã (decoder states) và vec-tơ trạng thái mã hóa (encoder states). Trong mô hình này s_{i-1} chính là đầu ra của khối Cell out đóng vai trò là Query còn h_j là encoder states đóng vai trò là Key, Value. Lần lượt tính bắt đầu từ ký tự bắt đầu câu (begin of sentence) cho đến ký tự kết thúc câu (end of sentence) trong chuỗi ký tự nhãn được cung cấp trong quá trình huấn luyện, các ký tự này sẽ được kết nối với context vector sau đó cho qua một lớp RNN được Cell out làm Query cho khối attention, đầu ra của khối attention là một context vector, từ context vector sẽ kết nối lại với Cell out ban đầu và cho qua các lớp tuyến tính và hàm softmax để tính xác xuất ký tự đầu ra, các ký tự của câu sẽ lần lượt được giải mã kết hợp với thuật toán Beam-search để tạo ra chuỗi đầu ra.

4.2 Mô hình đề xuất

4.2.1 Kiến trúc tổng quan

Kiến trúc tổng quan của mô hình mà tác giả đề xuất trong đồ án này được minh họa trong hình ... Trong ngôn ngữ tiếng Anh, cao độ thường không ảnh hưởng đến ý nghĩa của câu nói, tuy nhiên trong các ngôn ngữ thanh điệu (tonal language) như tiếng Việt, cao độ có ảnh hưởng lớn đến ý nghĩa của câu, từ được phát âm. Vì vậy để nâng cao hơn chất lượng mô hình cơ sở trên tiếng Việt tác giả đề xuất thêm phần trích xuất cao độ (pitch) của tiếng nói, đặc trưng về cao độ cung cấp thêm thông tin để mô hình giải mã có thể dự đoán chính xác hơn ý nghĩa của câu hay các từ được phát âm trong tiếng nói. Một hướng đề xuất cải tiến nữa cho mô hình cơ sở là trích xuất và sử dụng mã âm học (acoustic token) [30]. Kiến trúc mô hình cải tiến được



Hình 4.2: Kiến trúc mô hình cải tiến đề xuất

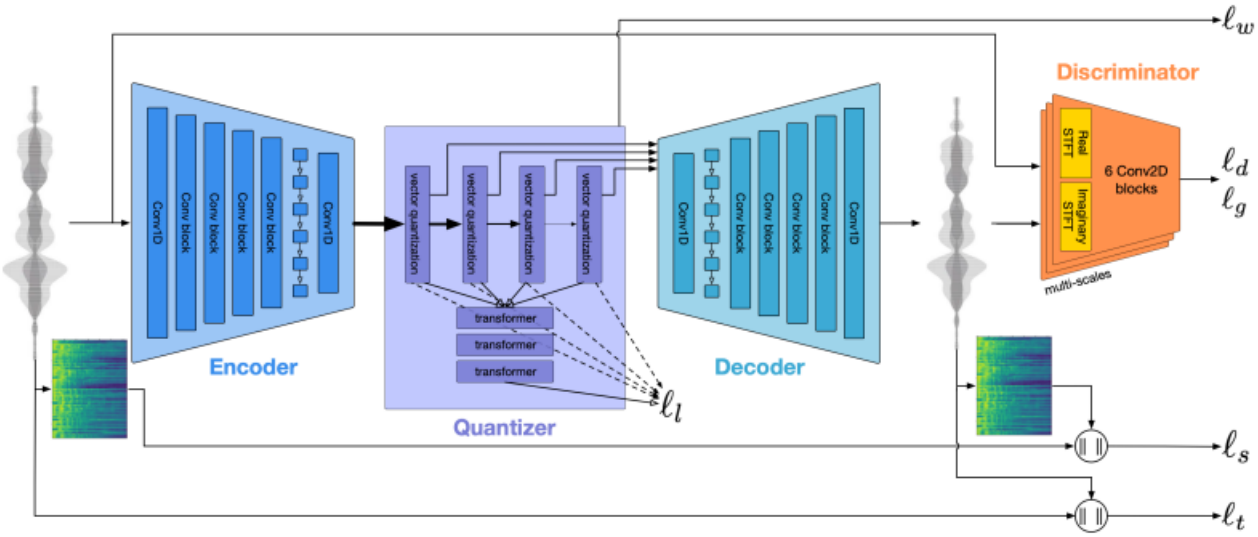
minh họa trong hình 4.2.

4.2.2 Thành phần mô hình hóa cao độ (pitch)

Để trích xuất cao độ trong tiếng nói, tác giả sử dụng phương pháp tương quan chéo chuẩn hóa (Normalized Cross-Correlation Function - NCCF) [31]. Phương pháp này được sử dụng để ước tính các tín hiệu cơ bản bao gồm cả cao độ trong các tín hiệu định kỳ như giọng nói. NCCF được tính toán bằng cách tính độ tương quan giữa các tín hiệu với các phiên bản bị trễ của chính đó ở các mức độ trễ khác nhau. Đối với các tín hiệu âm thanh, NCCF thường sẽ được sử dụng để trích xuất thông tin về cao độ. Trong đồ án này tác giả sử dụng công cụ trích xuất pitch là hàm torch.funtional.detect_pitch_frequency[1].

Trong mô hình cải tiến, tác giả sử dụng phép cộng đơn giản là cộng hai đặc trưng đầu ra của khối wav2vec2 và đầu ra của khối pitch encoder theo chiều thứ 2. Đầu ra của cả 2 khối đều có dạng [1, time, 786] với time là số bước thời gian của đoan âm thanh. Trong 768 đặc trưng tại 1 bước thời gian của encoder states chứa thông tin về ngữ nghĩa của tiếng nói còn 768 đặc trưng của đặc trưng pitch (pitch_feature) chứa thông tin về cao độ. Đặc trưng tổng của 2 đặc trưng này sẽ chứa cả thông tin

[1]https://pytorch.org/audio/stable/tutorials/audio$_{f}eature_{e}xtractions_{t}utorial.html$



Hình 4.3: Kiến trúc mô hình Encodec [30]

về ngữ nghĩa và cao độ và số chiều của đặc trưng mới vẫn giữ nguyên.

$$combine\_feature = encoder\_state + pitch\_encoder\_feature \tag{4.4}$$

4.2.3 Thành phần mô hình hóa âm học (Acoustic token)

Mô hình EnCodec: High Fidelity Neural Audio Compression [30] được Facebook giới thiệu vao năm 2022 với mục đích giải quyết các vấn đề xung quanh việc nén các tệp âm thanh (audio compression). Quá trình nén các dữ liệu âm thanh là việc loại bỏ hoặc giảm bớt các thông tin dư thừa trong âm thanh trong khi lưu trữ hoặc truyền âm thanh mà vẫn giữ được chất lượng âm thanh ở mức cần thiết. Trong các mô hình huấn luyện cũng phải có sự nén các âm thanh thành các đặc trưng tương ứng và sự mất mát thông tin là không thể tránh khỏi, vì vậy tác giả đề xuất sử dụng kiến trúc Encoder của bài báo này cho hệ thống cải tiến SLU, mục đích nhằm cung cấp thêm các thông tin khác đã có thể bị loại bỏ trong quá trình mã hóa dữ liệu giúp quá trình giải mã đạt kết quả cao hơn.

Nhìn vào hình 4.3, mô hình được sử dụng trong bài báo là một mô hình Encoder-Decoder với đầu vào là một đoạn âm thanh sau đó nén các đoạn âm thanh đó lại thành các vec-tơ đặc trưng và đầu ra là một đoạn âm thanh tương ứng nhận được khi giải nén các vec-tơ đặc trưng. Trong khối encoder bao gồm 1 lớp 1D convolution theo sau là nhiều khối convolution, sau khối convolution sẽ là 2 lớp LSTM và cuối cùng là 1 lớp 1D convolution. Từ đầu ra của bộ mã khóa, các đặc trưng trích xuất được sẽ được lượng tử hóa thông qua lớp Quantizer. Sau khi lượng tử hóa thu được các codebook chứa các đặc trưng được mã hóa và từ các codebook này thông tin

được giải mã để cung cấp thông tin cho mô hình đề xuất.

Acoustic token feature cũng có dạng tương tự encoder states và pitch encoder feature. Trong mô hình cải tiến, tác giả sử dụng khối Multihead attention với acoustic token feature làm Key và Value, còn đặc trưng tổng gồm pitch feature và encoder states làm Query. Mục đích để tính toán sự tương quan giữa các đặc trưng âm học (acoustic token feature) và tổng đặc trưng ngữ nghĩa và cao độ (encoder state + pitch feature), do đặc trưng âm học mang lượng thông tin nhiều hơn đặc trưng ngữ nghĩa vậy nên khối attention sẽ giúp so khớp tìm các thông tin quan trọng đã bị bỏ sót với mỗi đặc trưng ngữ nghĩa tương ứng của âm thanh có trong đặc trưng âm học. Đầu ra của khối Multihead attention là một context vector được tính theo công thức sau:

$$Multihead(Q, K, V) = ConCat(head_1, ..., head_h)W^o \tag{4.5}$$

với $head_i$ = $Attention(QW_i^Q, KW_i^K, VW_i^V)$ Trong đó Q: Acoustic token feature; K,V: Pitch encoder feature + Encoder states.

Cuối cùng vec-tơ ngữ cảnh (context vector) nhận được qua khối Multihead attention mang thông tin của các 3 đặc trưng sẽ cộng lại với encoder states theo chiều thứ 2 để thông tin ngữ nghĩa ban đầu vẫn được bảo toàn.

$$encoder\_feature = context\_vector + encoder\_state \tag{4.6}$$

Đặc trưng mới thu được sẽ mang cả 3 thông tin trong giọng nói đầu vào giúp quá trình giải mã trở nên dễ dàng và chính xác hơn.

4.3 Chuẩn bị thực nghiệm

4.3.1 Môi trường thực nghiệm

Trong khuôn khổ đồ án này, tác giả thực hiện huấn luyện mô hình trên GPU RTX2080, Google Colab và Kaggle.

4.3.2 Bộ dữ liệu thực nghiệm

Quá trình thực nghiệm sẽ diễn ra trên bộ dữ liệu đề xuất là VN-SLU với 10000 câu nói, 12 giờ dung lượng. Bộ dữ liệu hiểu ngôn ngữ tự nhiên dạng nói bao gồm 9 ý định và 7 thực thể. Bộ dữ liệu được phân chia thành 3 bộ train, dev và test tương ứng trong bàng 4.1.

Để phân chia bộ dữ liệu hiệu quả, tác giả lựa chọn các câu nói có combination chứa các device và location ít xuất hiện nhất cho tập test và tập dev, chia tập câu nói mới thu được với tỷ lệ $6:4$, tập test sẽ chiếm 60% bộ câu nói chọn lọc và các

	Train set	Dev set	Test set
Utterances	7592	1260	1483

Bảng 4.1: Phân chia bộ dữ liệu VN-SLU

câu có combination xuất hiện trong tập test sẽ không có trong tập train. Còn đối với tập dev, sẽ bao gồm 40% còn lại của bộ câu nói chọn lọc và thêm một phần dữ liệu được trích xuất từ tập train với tỷ lệ trùng lặp các combination giữa 2 tập là 25%.

4.3.3 Phương pháp đánh giá

Đối với bài toán hiểu ngôn ngữ tự nhiên dạng nói có 2 chỉ số đánh giá chính cho 2 nhiệm vụ: thứ nhất là độ chính xác của ý định (intent accuracy hay intent F1) và thứ 2 là đánh giá sự chính xác đối với nhiệm vụ điền khung ngữ nghĩa (slot filling), chỉ số để đánh giá đối với nhiệm vụ này được tác giả sử dụng là SLU-F1 [8] Cách tính Intent-F1 như sau: Công thức tính Intent-F1:

$$\text{Intent F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.1}$$

$$\text{Với: Precision} = \frac{TP}{TP + FP} \quad \text{và} \quad \text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

Trong đó - True Positive (TP): Số lượng intent dự đoán chính xác - False Positive (FP): Số lượng intent dự đoán bởi hệ thống nhưng không có trong groundtruth - False Negative (FN): Số lượng intent có trong groundtruth nhưng không được hệ thống dự đoán

Tương tự cách tính Intent-F1, cách tính SLU-F1 như sau:

$$\text{SLU-F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

$$\text{Với: Precision} = \frac{TP}{TP + FP} \quad \text{và} \quad \text{Recall} = \frac{TP}{TP + FN} \tag{4.4}$$

Trong đó - True Positive (TP): Số slot value dự đoán đúng - False Positive (FP): Số lượng slot value hệ thống dự đoán nhưng không có trong groundtruth - False Negative (FN): Số lượng slot value có trong groundtruth nhưng không được hệ thống dự đoán

4.4 Kết quả thực nghiệm

4.4.1 Kết quả thực nghiệm trên bộ dữ liệu VN-SLU

Trong khuôn khổ đồ án, tác giả thực hiện thử nghiệm 4 mô hình trên bộ dữ liệu VN-SLU: (i) Mô hình cơ sở: Là mô hình được sử dụng Pre-trained wav2vec encoder kết hợp với AttentionRNNDecoder. (ii) Mô hình cải tiến thêm pitch: Là mô hình có sử dụng thông tin về pitch và âm thanh làm đầu vào, đặc trưng của pitch cộng với encoder states sẽ được cho qua khối AttentionRNNDecoder để giải mã. (iii) Mô hình cải tiến thêm acoustic token: Là mô hình sử dụng thông tin về mã âm học (acoustic token) kết hợp với encoder states sẽ được cho qua khối AttentionRNNDecoder để giải mã. (iv) Mô hình cải tiến đề xuất: Tích hợp cả 3 thông tin về pitch, encoder states và acoustic token để cung cấp thông tin cho bộ giải mã.

Model	Intent Acc	SLU-F1
Baseline	89.52%	68.38%
Baseline + Acoustic	92.05%	70.00%
Baseline + Pitch	92.51%	70.77%
Baseline + Pitch + Acoustic	93.33%	71.32%

Bảng 4.2: Bảng kết quả thực nghiệm trên dữ liệu VN-SLU

Kết quả thực nghiệm được thống kê trong bảng 4.1. Mô hình cơ sở cho kết quả tệ nhất với độ chính xác đối với nhiệm vụ phân loại ý định chỉ đạt 89.52% Intent Acc và SLU-F1 đạt 68.38%, các mô hình cải tiến đều đạt kết quả tốt hơn so với mô hình cơ sở trên bộ dữ liệu VN-SLU. Kết quả tốt nhất của mô hình đề xuất đạt mức 93.33% intent Acc và 71.32% SLU-F1, cả 2 chỉ số trên lần lượt có mức tăng 3.81% và 2.94%.

Tại chương 4 này, tác giả đã đưa ra mô hình cơ sở và thực hiện các cải tiến trên mô hình đó. Các thực nghiệm được thực hiện trên bộ dữ liệu VN-SLU, các kết quả thực nghiệm đều đạt mức khả quan nhất định trên các tác vụ mà bài toán hiểu ngôn ngữ tự nhiên dạng nói đặt ra.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Trong đồ án này, tác giả đã thực hiện hai đóng góp là trình bày quy trình xây dựng bộ dữ liệu SLU cho tiếng Việt VN-SLU và thực hiện cải tiến mô hình End-to-end SLU trên bộ dữ liệu VN-SLU. Quy trình xây dựng dữ liệu bao gồm 4 bước chính là: chuẩn bị dữ liệu, sinh dữ liệu, thu thập dữ liệu và cuối cùng là xử lý dữ liệu. Với quy trình trên có thể thực hiện xây dựng bộ dữ liệu SLU trên nhiều lĩnh vực khác nhau, tuy nhiên trong khuôn khổ đồ án, tác giả đã áp dụng để xây dựng một bộ dữ liệu liệu SLU với chủ đề là tập hợp các yêu cầu, tương tác giữa người dùng với các thiết bị trong nhà thông minh. Bộ dữ liệu thu được có tổng dung lượng là 12 giờ với 10000 câu nói xây dựng dựa trên 9 ý định và 7 thực thể cùng sự tham gia thu âm của 130 người.

Đối với mô hình thực nghiệm, tác giả lựa chọn sử dụng mô hình baseline End-to-end SLU [29] dưới dạng sequence-to-sequence với đầu vào là một đoạn âm thanh của câu nói và đầu ra là ý định của câu đi kèm với các thực thể có trong câu nói. Trong mô hình cải tiến, tác giả đã đề xuất sử dụng đặc trưng về cao độ (pitch) cho ngôn ngữ thanh điệu như tiếng Việt, thêm nữa tác giả cũng đề xuất sử dụng thêm mô hình trích xuất đặc trưng về mã âm học (acoustic token) để bổ sung thêm các thông tin đã bị mất mát trong quá trình nén dữ liệu âm thanh thành các đặc trưng trong mô hình mã hóa dữ liệu. Kết hợp việc sử dụng các mô hình Wav2vec2 [11] và HuBERT [28] đã được huấn luyện từ trước với hiệu quả cao trên tiếng Anh cùng với hai đề xuất trên giúp bổ sung thêm thông tin cung cấp cho mô hình giải mã tiếng nói trở nên chính xác hơn. Kết quả đánh giá thực nghiệm trên bộ dữ liệu VN-SLU của mô hình cải tiến tốt nhất cho kết quả đạt mức 93.33% intent Acc và 71.32% SLU-F1, cả 2 chỉ số trên lần lượt có mức tăng 3.81% và 2.94%.

5.2 Hướng phát triển trong tương lai

Hiện tại bộ dữ liệu VN-SLU thu được có dung lượng ở mức tương đối, số lượng ý định và thực thể còn chưa nhiều nếu so sánh với các bộ dữ liệu SLU khác trên thế giới, thêm nữa bộ dữ liệu đang chỉ thu thập trên một lĩnh vực là tương tác giữa con người với các thiết bị thông minh trong nhà. Vì vậy trong tương lai, tác giả muốn phát triển bộ dữ liệu hiện tại thêm nhiều câu nói hơn với nhiều loại ý định và thực thể khác nhau để độ đa dạng của bộ dữ liệu trở nên phong phú và chất lượng hơn nữa. Ngoài ra tác giả cũng hướng đến việc tận dụng quy trình xây dựng dữ liệu sẵn có để xây dựng thêm các bộ dữ liệu SLU khác trên tiếng Việt trong nhiều lĩnh vực khác nhau.

Về mô hình, các kết quả của các phương pháp cải tiến tuy có tăng nhưng vẫn chưa thể đạt tới mức tối đa thêm nữa là các đề xuất hiện chỉ phục vụ cho việc cải tiến mô hình trên tiếng Việt và chưa đánh giá được liệu các cải tiến có hiệu quả trên các ngôn ngữ khác hay không. Cho nên trong tương lai tác giả sẽ nghiên cứu thêm các cải tiến để mô hình có thể mang lại hiệu quả cao hơn không chỉ trên ngôn ngữ tiếng Việt mà nhiều ngôn ngữ khác trên thế giới.

TÀI LIỆU THAM KHẢO

[1] P. I. V. S. T. Y. B. Loren Lugosch Mirco Ravanelli, "Speech model pre-training for end-to-end spoken language understanding," **in***Interspeech* 2019.

[2] J. S. Sepp Hochreiter, *Long Short-Term Memory*. Neural Computation (1997), vol-9, issue-8, pp 1735–1780, 1997.

[3] K. C.-Y. B. Junyoung Chung Caglar Gulcehre, "Empirical evaluation of gated recurrent neural networks on sequence modeling," **in***NIPS 2014 Deep Learning and Representation Learning Workshop* 2014.

[4] N. R. M. S. J. K. P. D. A. N. P. S. G. S. A. A. S. A. A. H.-V. G. K. T. H. R. C. A. R. D. M. Z. J T. B. Brown B. Mann **and** D. Amodei, "Language models are few-shot learners," **in***arXiv:2005.14165 [cs.CL]* 2020.

[5] K. L. K. T. Jacob Devlin Ming-Wei Chang, "Bert: Pre-training of deep bidirectional transformers for language understanding," **in***Empirical Methods in Natural Language Processing (EMNLP)* 2018.

[6] N. P. J. U. L. J. A. N. G. K. A. Vaswani N. Shazeer **and** I. Polosukhin, "Attention is all you need," **in***Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017), pp. 6000-6010* 2017.

[7] F. G. Alex Graves Santiago Fernandez **and** J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," **in***Proceedings of the 23rd International Conference on Machine Learning (ICML), pp. 369-376* 2006.

[8] E. Bastianelli, A. Vanzo, P. Swietojanski **and** V. Rieser, "Slurp: A spoken language understanding resource package," **in***Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 2020.

[9] P. I. V. S. T. Y. B. Loren Lugosch Mirco Ravanelli, "Speech model pre-training for end-to-end spoken language understanding," **in***Audio and Speech Processing (eess.AS)* 2019.

[10] C. T. Hemphill, J. J. Godfrey **and** G. R. Doddington, "The ATIS spoken language systems pilot corpus," **in***Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990* 1990. **url**: `https://aclanthology.org/H90-1021`.

[11] A. M. M. A. Alexei Baevski Henry Zhou, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," **in***Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 2020.

[12] Q. X. A. B. J. G. M. A. Alexei Baevski Wei-Ning Hsu, "Data2vec: A general framework for self-supervised learning in speech, vision and language," **in***Proceedings of the 39th International Conference on Machine Learning, in PMLR, vol.162, pp1298-1312* 2022.

[13] C.-C. C. N. P. Y. Z. J. Y. W. H. S. W. Z. Z. Y. W. R. P. Anmol Gulati James Qin, "Conformer: Convolution-augmented transformer for speech recognition," **in***Interspeech* 2020.

[14] A. B. T. B. A. C. D. L. C. D. T. G. F. C. T. L. M. P. J. D. Alice Coucke Alaa Saade, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," **in***Empirical Methods in Natural Language Processing (EMNLP)* 2018.

[15] A. C. J. D. A. B. T. B. D. L. C. D. T. G. F. C. T. L. M. P. Alaa Saade Alice Coucke, "Spoken language understanding on the edge," **in***Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing colocated with NeurIPS* 2019.

[16] C. Gershenson, *Artificial neural networks for beginners*, 2003. arXiv: `cs/0308031`.

[17] A. F. Agarap, *Deep learning using rectified linear units (relu)*, 2019. arXiv: `1803.08375`.

[18] G. Bebis **and** M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, **jourvol** 13, **number** 4, **pages** 27–31, 1994.

[19] D. E. Rumelhart, G. E. Hinton **and** R. J. Williams, "Learning representations by back-propagating errors," *Nature*, **jourvol** 323, **pages** 533–536, 1986.

[20] O. V. v. Q. V. L. Ilya Sutskever, "Sequence to sequence learning with neural networks," **in***Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol.2* 2014, 3104–3112.

[21] Y. B. Dzmitry Bahdanau Kyunghyun Cho, "Neural machine translation by jointly learning to align and translate," **in***The International Conference on Learning Representations - ICLR* 2015.

[22] A. M. M. A. Alexei Baevski Henry Zhou, "Wav2vec: Unsupervised pre-training for speech recognition," **in***Interspeech* 2019.

[23] J. L. Ba, J. R. Kiros **and** G. E. Hinton, *Layer normalization*, 2016. arXiv: `1607.06450`.

[24] D. Hendrycks **and** K. Gimpel, *Gaussian error linear units (gelus)*, 2023. arXiv: `1606.08415`.

[25] A. Baevski **and** M. Auli, "Adaptive input representations for neural language modeling," **in***7th International Conference on Learning Representations (ICLR)*.

[26] H. Jégou, M. Douze **and** C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **jourvol** 33, **number** 1, **pages** 117–128, 2011.

[27] A. van den Oord, Y. Li **and** O. Vinyals, *Representation learning with contrastive predictive coding*, 2019. arXiv: 1807.03748.

[28] Y.-H. H. T. K. L. R. S. Wei-Ning Hsu Benjamin Bolte **and** A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol.29, pp 3451–3460*, 2021.

[29] Y. Wang, A. Boumadane **and** A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *ArXiv*, **jourvol** abs/2111.02735, 2021.

[30] A. Défossez, J. Copet, G. Synnaeve **and** Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[31] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal **and** S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," **in***2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2494-2498* 2014.

PHỤ LỤC