BỘ GIÁO DỤC VÀ ĐÀO TẠO
VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM
HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

NGUYỄN THỊ THU HIỀN

NGHIÊN CỨU PHƯƠNG PHÁP CHUẨN HOÁ VĂN BẢN VÀ NHẬN DẠNG THỰC THỂ ĐỊNH DANH TRONG NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT

LUẬN ÁN TIẾN SĨ NGÀNH MÁY TÍNH

HÀ NỘI - 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO **VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM**

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

NGUYỄN THỊ THU HIỀN

NGHIÊN CỨU PHƯƠNG PHÁP CHUẨN HOÁ VĂN BẢN VÀ NHẬN DẠNG THỰC THỂ ĐỊNH DANH TRONG NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT

LUẬN ÁN TIẾN SĨ NGÀNH MÁY TÍNH

Chuyên ngành: Hệ thống thông tin

Mã số: 9 48 01 04

Xác nhận của Học viện	**Người hướng dẫn 1**	**Người hướng dẫn 2**
Khoa học và Công nghệ	*(Ký, ghi rõ họ tên)*	*(Ký, ghi rõ họ tên)*

HÀ NỘI - 2023

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các kết quả được viết chung với các tác giả khác đều được sự đồng ý của các đồng tác giả trước khi đưa vào luận án. Các kết quả nêu trong luận án là trung thực và chưa từng được công bố trong các công trình nào khác.

Tác giả

Nguyễn Thị Thu Hiền

LỜI CẢM ƠN

Luận án của tác giả được thực hiện tại Học viện Khoa học và Công nghệ - Viện Hàn lâm Khoa học và Công nghệ Việt Nam, dưới sự hướng dẫn tận tình của PGS.TS. Lương Chi Mai và TS. Nguyễn Thị Minh Huyền. Tôi xin được bày tỏ lòng biết ơn sâu sắc đến hai Cô về những định hướng nghiên cứu, sự động viên và hướng dẫn tận tình giúp tôi vượt qua những khó khăn để hoàn thành luận án này.

Tôi cũng xin gửi lời cảm ơn chân thành đến các nhà khoa học, các đồng tác giả của các công trình nghiên cứu đã được trích dẫn trong luận án. Đây là những tư liệu quý báu có liên quan giúp tôi hoàn thành luận án.

Tôi xin chân thành cảm ơn đến Ban lãnh đạo Học viện Khoa học và Công nghệ, Viện Công nghệ Thông tin đã tạo điều kiện thuận lợi cho tôi trong quá trình học tập, nghiên cứu.

Tôi xin chân thành cảm ơn Ban giám hiệu trường Đại học Sư phạm - ĐH Thái Nguyên, Khoa Toán, Bộ môn Khoa học máy tính - Hệ thống thông tin và các đồng nghiệp đã giúp đỡ và tạo điều kiện thuận lợi để tôi có thể thực hiện kế hoạch nghiên cứu, hoàn thành luận án.

Tôi xin được bày tỏ tình cảm và lòng biết ơn vô hạn tới những người thân trong Gia đình, những người luôn dành cho tôi sự động viên, khích lệ, sẻ chia, giúp đỡ trong những lúc khó khăn.

Tác giả

Nguyễn Thị Thu Hiền

MỤC LỤC

Trang

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ tiếng Anh	Ý nghĩa tiếng Việt
1	ASR	Automatic Speech Recognition	Nhận dạng tiếng nói tự động
2	BERT	Bidirectional Encoder Representations from Transformers	Mã hóa biểu diễn hai chiều dựa trên Transformers
3	BiLSTM	Bidirectional Long Short Term Memory	Mô hình bộ nhớ ngắn-dài hạn hai chiều
4	BPE	Byte-Pair-Encoding	Mã hoá cặp byte
5	CaPu	Recovering Capitalization and Punctuation model	Mô hình khôi phục dấu câu và chữ hoa
6	CBOW	Continuous Bag of Words	Mô hình nhúng từ "Túi từ liên tục"
7	CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
8	CRF	Conditional Random Fields	Trường ngẫu nhiên có điều kiện
9	DL	Deep Learning	Học sâu
10	DNN	Deep Neural Networks	Mạng nơ-ron sâu
11	ELMO	Embeddings from Language Model	Nhúng từ từ mô hình ngôn ngữ
12	E2E	End-to-End	Mô hình đầu - cuối
13	GloVe	Global Véc-tơs for Word Representation	Mô hình nhúng từ dựa trên biểu diễn từ
14	GRU	Gated Recurrent Unit	Mạng hồi tiếp có cổng

15	GPT	Generative pre-trained transformer	Mô hình biến đổi được huấn luyện trước
16	HMM	Hidden Markov Model	Mô hình Markov ẩn
17	LM	Language Model	Mô hình ngôn ngữ
18	LSTM	Long Short Term Memory	Mô hình bộ nhớ ngắn-dài hạn
19	ME	Maximum Entropy	Mô hình Entropy cực đại
20	MEMM	Maximum Entropy Markov Model	Mô hình Markov Entropy cực đại
21	MTL	Multi-Task Learning	Học đa tác vụ
22	NER	Named Entity Recognition	Nhận dạng thực thể định danh
23	OOV	Out-of-Vocabulary	Từ nằm ngoài từ điển
24	RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
25	Seq2seq	Sequence-to-Sequence	Mô hình ánh xạ từ chuỗi sang chuỗi
26	SLU	Spoken Language Understanding	Hiểu ngôn ngữ nói
27	SVM	Support Véc-tơ Machine	Máy véc-tơ hỗ trợ
28	VLSP	Vietnamese Language and Speech Processing	Hội thảo xử lý ngôn ngữ và tiếng nói tiếng Việt
29	XLNNTN		Xử lý ngôn ngữ tự nhiên
30	TTS	Text To Speech	Hệ thống chuyển văn bản sang tiếng nói
31	WER	Word Error Rate	Tỉ lệ lỗi từ

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH VẼ

MỞ ĐẦU

Trong xã hội hiện đại, thông tin có thể dễ dàng được tiếp cận trên phạm vi toàn cầu nhờ hệ thống Internet rộng khắp. Bên cạnh thông tin dạng văn bản thì thông tin dạng âm thanh, phim ảnh ngày càng trở nên phổ biến và thu hút sự quan tâm của người sử dụng Internet nhờ hệ thống băng thông mạng ngày càng được mở rộng. Mặc dù vậy, thông tin dưới dạng văn bản vẫn có giá trị riêng biệt mà khó có dạng thức thông tin nào có thể thay thế được - nhất là trong các hoạt động giao tiếp thuộc các lĩnh vực như: kinh tế, chính trị, ngoại giao, khoa học... Kết quả các cuộc đàm phán, đối thoại song phương, đa phương bao giờ cũng được hiện thực hóa bằng các văn bản ghi nhớ của các bên liên quan.

Xử lý ngôn ngữ tự nhiên (XLNNTN) là lĩnh vực khoa học máy tính kết hợp giữa trí tuệ nhân tạo và ngôn ngữ học tính toán, nhằm xử lý tương tác giữa con người và máy tính sao cho máy tính có thể hiểu hay bắt chước được ngôn ngữ của con người. XLNNTN bao gồm hai nhánh lớn là xử lý tiếng nói (Speech processing) và xử lý văn bản (Text processing).

Một trong những bài toán quan trọng trong hiểu ngữ nghĩa văn bản viết hay nói là nhận dạng thực thể định danh (Named Entity Recognition - NER). Có thể nói, đây là một bài toán tiền đề cho các hệ thống về hiểu ngôn ngữ hay khai phá văn bản như trích xuất sự kiện, hỏi đáp tự động hay tìm kiếm ngữ nghĩa. Đã có nhiều nghiên cứu đạt được những kết quả rất khả quan cho bài toán NER với dữ liệu văn bản viết thông thường trong nhiều ngôn ngữ trên thế giới cũng như tiếng Việt. Trong khi đó, các nghiên cứu về nhận dạng thực thể định danh cho văn bản đầu ra của nhận dạng tiếng nói (Automatic Speech Recognition - ASR) có những khó khăn riêng so với văn bản viết, và có ít công trình nghiên cứu cho tiếng Việt.

Nhận dạng tiếng nói là một quá trình chuyển đổi tín hiệu tiếng nói của một ngôn ngữ cụ thể thành một chuỗi các từ có nội dung tương ứng ở định dạng văn

bản. Văn bản đầu ra của ASR thường không có cấu trúc, chẳng hạn như không có dấu câu, không viết hoa chữ cái đầu câu hoặc tên riêng, tên địa danh, ... Điều này dẫn đến khó khăn trong quá trình hiểu và hạn chế khả năng khai thác văn bản đầu ra của ASR trong hầu hết các ứng dụng. Việc nhận dạng thực thể định danh từ văn bản đầu ra của nhận dạng tiếng nói tự động do đó có những đặc trưng khác biệt vì nó luôn chứa nhiều lỗi nhận dạng, đặc biệt là các thực thể định danh nhiều khi nằm ngoài từ điển (Out-of-vocabulary - OOV). Các lỗi ASR thường xảy ra trong các từ cấu thành nên thực thể định danh hoặc trong ngữ cảnh của những từ đó, do vậy làm ảnh hưởng trực tiếp đến hiệu suất của NER. Ngoài ra, các hệ thống NER phải đối mặt với những vấn đề về sự thiếu hụt một số dấu hiệu quan trọng như chữ viết hoa, dấu chấm câu. Bên cạnh đó, để cải thiện kết quả đầu ra của ASR, người ta cần chuẩn hóa văn bản bằng cách loại bỏ các từ vô nghĩa, chuẩn hóa dữ liệu kiểu số, ngày, tháng, khôi phục dấu câu và viết hoa, xử lý từ nước ngoài, ... Sau xử lý, văn bản cuối sẽ có cấu trúc tốt và dễ hiểu hơn so với văn bản đầu ra của ASR, đồng thời khi đưa vào triển khai trong các ứng dụng thực tế (tạo phụ đề phim, tạo văn bản các cuộc họp trực tuyến, trích xuất thông tin khách hàng, ...) đạt hiệu quả cao hơn.

Như vậy, việc phát triển các giải pháp chuẩn hoá văn bản và nhận dạng thực thể định danh từ văn bản đầu ra của ASR là cần thiết để cải thiện chất lượng tổng thể của hệ thống ASR.

Tuy nhiên, việc chuẩn hoá văn bản đầu ra của ASR, cụ thể là vấn đề khôi phục dấu câu, chữ hoa vẫn còn không ít vấn đề cần cải thiện. Có thể kể đến như: tính toán việc cắt chuỗi câu dài để lấy được nhiều nhất ngữ cảnh các từ xung quanh đoạn cắt; xử lý trên văn bản có chứa lỗi đầu ra ASR (chèn, xóa, thay thế từ); kết hợp khôi phục dấu câu và chữ hoa trong một mô hình như thế nào để đạt được hiệu quả tối ưu; đặc biệt, một trong những vấn đề khó khăn nhất của các nghiên cứu về xử lý tiếng nói là nguồn dữ liệu. Việc sở hữu một nguồn dữ liệu phong phú, đủ lớn cho việc huấn luyện các mô hình học sâu là vô cùng cần thiết. Đến thời điểm hiện tại, chưa có nhiều công bố nghiên cứu

về khôi phục dấu câu và chữ hoa cho văn bản đầu ra của ASR tiếng Việt, do vậy, việc xây dựng bộ dữ liệu và đề xuất mô hình giải quyết bài toán này là cần thiết, giúp cải tiến chất lượng hệ thống ASR tiếng Việt.

Bên cạnh ý nghĩa trong việc cải thiện chất lượng đầu ra của ASR thì dấu câu, chữ hoa cũng là một trong những thông tin quan trọng, hữu ích cho bài toán nhận dạng thực thể định danh. Có thể thấy, không phải tất cả các từ viết hoa trong tiếng Việt đều được coi là thực thể định danh (ví dụ các từ viết hoa đầu câu). Ngược lại, thực thể định danh cũng không nhất thiết là các từ/cụm từ viết hoa đầy đủ (ví dụ: Uỷ ban nhân dân Thành phố Hà Nội, Bộ Giao thông vận tải, ...). Đặc biệt, cũng là thực thể định danh nhưng được phân loại thành các dạng thực thể khác nhau (ví dụ, thành phố Hồ Chí Minh, đường mòn Hồ Chí Minh là tên địa điểm, nhưng, lãnh tụ Hồ Chí Minh là tên người). Do đó, việc khôi phục dấu câu, chữ hoa là một trong các yếu tố quan trọng giúp tối ưu hóa hệ thống nhận dạng thực thể định danh trong văn bản đầu ra ASR.

Trong thực tế, đã có nhiều phương pháp xử lý NER cho văn bản đầu ra ASR nhưng chủ yếu tập trung ở ngôn ngữ giàu tài nguyên như tiếng Anh, tiếng Trung, tiếng Nhật. Có rất ít nghiên cứu áp dụng NER cho ASR tiếng Việt và các nghiên cứu này cũng mới chỉ tập trung cho văn bản hội thoại ngắn. Từ những thách thức đó, nghiên cứu sinh đã lựa chọn nghiên cứu đề tài *"Nghiên cứu phương pháp chuẩn hoá văn bản và nhận dạng thực thể định danh trong nhận dạng tiếng nói tiếng Việt"*.

Mục tiêu nghiên cứu

Luận án tập trung đề xuất giải pháp và triển khai thực nghiệm cho hai mục tiêu cụ thể. *Thứ nhất* là chuẩn hóa văn bản đầu ra của hệ thống ASR tiếng Việt bằng cách khôi phục dấu câu, chữ hoa. *Thứ hai* là nhận dạng thực thể định danh trên văn bản đầu ra của hệ thống ASR tiếng Việt.

Nội dung nghiên cứu

Để thực hiện các nhiệm vụ trên, trước tiên, luận án nghiên cứu đặc thù dữ liệu và lỗi đầu ra của các hệ thống ASR tiếng Việt, tìm hiểu các vấn đề cơ bản

của bài toán NER cũng như các thách thức của bài toán NER với văn bản đầu ra của ASR tiếng Việt. Một nội dung không thể thiếu được là xây dựng bộ dữ liệu phục vụ cho việc huấn luyện và đánh giá các mô hình học máy để giải quyết bài toán đặt ra. Trên cơ sở đó, luận án đề xuất mô hình khôi phục dấu câu và chữ hoa phục vụ chuẩn hóa văn bản đầu ra của ASR tiếng Việt. Bài toán NER cho văn bản đầu ra của ASR tiếng Việt được nghiên cứu giải quyết theo hai hướng. *Một là* hướng tiếp cận xây dựng hệ thống đường ống (Pipeline) bao gồm một số mô hình con đơn lập ghép nối tuần tự. *Hai là* hướng tiếp cận xây dựng hệ thống đầu - cuối (End-to-End - E2E) gồm các mô hình con kết hợp thành một mô hình học máy phức hợp với một luồng tính toán duy nhất.

Phạm vi nghiên cứu

Các nghiên cứu chuẩn hoá văn bản và nhận dạng thực thể định danh trong nội dung tiếng nói thường được tiếp cận theo hai cách: (1) chỉ sử dụng đặc trưng từ vựng trong văn bản đầu ra của hệ thống ASR hoặc (2) sử dụng trực tiếp các đặc trưng âm thanh, trong đó có thông tin nhiễu khi thu âm, cao độ người nói, khoảng ngắt nghỉ, ... Trong phạm vi luận án, nghiên cứu sẽ tập trung vào hướng giải quyết các vấn đề liên quan đến xử lý văn bản đầu ra của ASR với văn bản tiếng nói dài, khó xử lý.

Bên cạnh đó, với vấn đề chuẩn hóa văn bản đầu ra của ASR, nghiên cứu chỉ tập trung thiết kế mô hình dự đoán dấu câu, chữ hoa và coi hệ thống ASR có tỉ lệ lỗi từ (Word Error Rate - WER) bằng 0%. Về bài toán NER, luận án sử dụng hệ thống ASR thực tế có WER là 4.85% để đánh giá mô hình.

Trong luận án này, nghiên cứu sinh sử dụng nhiều thuật ngữ bằng tiếng Anh được trình bày trong bảng danh mục từ viết tắt và thuật ngữ. Để thuận tiện cho việc theo dõi luận án, các thuật ngữ đã được giải thích về nghĩa trong bảng này sẽ được dùng từ tiếng Anh.

Phương pháp nghiên cứu, triển khai

Luận án đã thực hiện nghiên cứu lý thuyết, bao gồm tổng quan về các bài toán cần giải quyết, các phương pháp, kĩ thuật đã được sử dụng để giải quyết

các bài toán này và hiệu quả của chúng. Trên cơ sở đó, luận án đề xuất các giải pháp để khắc phục một số vấn đề còn tồn tại. Luận án cũng chú trọng triển khai phương pháp thực nghiệm nhằm đo lường, đánh giá các mô hình đề xuất giải quyết bài toán, so sánh với các phương pháp khác.

Về dữ liệu thực nghiệm, luận án cần xây dựng các bộ dữ liệu văn bản kết hợp với tiếng nói tương ứng nhằm đáp ứng các bài toán đặt ra.

Các đóng góp của luận án

Luận án đã có những đóng góp chính sau:

-Xây dựng các bộ dữ liệu văn bản kết hợp với tiếng nói cho huấn luyện và đánh giá các mô hình chuẩn hoá và nhận dạng thực thể định danh cho văn bản đầu ra của các hệ thống ASR. Các dữ liệu này được mô tả trong các công trình [CT1, CT2, CT4, CT6];

-Đề xuất và cải tiến mô hình khôi phục dấu câu và chữ hoa giúp chuẩn hoá văn bản đầu ra của ASR tiếng Việt. Mô hình này được đưa ra, đánh giá và cải tiến trong các công trình [CT2, CT3, CT5];

-Đề xuất hai giải pháp nhận dạng thực thể định danh trong văn bản đầu ra của ASR tiếng Việt theo hướng tiếp cận đường ống và E2E. Các giải pháp này được trình bày và đánh giá trong các công trình [CT4, CT6].

Bố cục luận án

Ngoài phần mở đầu và kết luận, luận án được cấu trúc thành 4 chương. Chương 1 trình bày tổng quan các vấn đề nghiên cứu. Chương này phát biểu và nêu ý nghĩa ứng dụng của các bài toán, chỉ ra các thách thức cần giải quyết, khảo sát các nghiên cứu về nhận dạng tiếng nói, nhận dạng thực thể định danh từ tiếng nói nói chung và đối với tiếng Việt nói riêng. Chương 2 - Kiến thức cơ sở, trình bày những kiến thức nền tảng được sử dụng để định hướng và là cơ sở để đề xuất mô hình chuẩn hoá và nhận dạng thực thể định danh cho văn bản đầu ra của ASR. Tiếp theo, chương 3 sẽ giới thiệu về bài toán khôi phục dấu câu và chữ hoa cho hệ thống ASR tiếng Việt. Trong chương này, luận án trình bày mô hình đề xuất, dữ liệu và các kết quả thực

nghiệm cho bài toán. Cuối cùng, chương 4 đề xuất phương pháp nhận dạng thực thể định danh cho văn bản đầu ra của ASR tiếng Việt theo hai hướng tiếp cận đường ống và tiếp cận đầu-cuối, trình bày các kết quả thực nghiệm, và so sánh hai cách tiếp cận.

CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

NER là một bài toán quan trọng trong XLNNTN. Bài toán này đã và đang được nghiên cứu, đạt hiệu suất cao đối với văn bản viết thông thường. Tuy nhiên, với văn bản đầu ra của ASR, các thông tin đặc trưng về dấu câu, chữ hoa cho NER không còn tồn tại, gây nhiều khó khăn cho xử lý. Điều này khiến cho các nghiên cứu về NER trong văn bản đầu ra của ASR còn hạn chế. Chính vì vậy, việc nghiên cứu, xử lý và chuẩn hóa văn bản đầu ra của ASR, giúp cải tiến hệ thống ASR và phục vụ cho đầu vào của hệ thống NER là quan trọng và có ý nghĩa. Chương này trước hết sẽ trình bày tổng quan về XLNNTN, những khó khăn khi xử lý ngôn ngữ tiếng Việt. Tiếp đó là phần tìm hiểu chung về hệ thống ASR, những đặc trưng trong văn bản đầu ra của hệ thống ASR và các nghiên cứu liên quan đến việc chuẩn hóa văn bản đầu ra của ASR giúp hỗ trợ cho mô hình NER. Cuối chương, luận án mô tả bài toán NER, những khó khăn khi xử lý NER cho tiếng nói tiếng Việt và các nghiên cứu liên quan.

1.1. Xử lý ngôn ngữ tự nhiên

1.1.1. Giới thiệu

Ngôn ngữ là một trong những khía cạnh nhận thức quan trọng nhất của con người. Ngôn ngữ tự nhiên đề cập đến bất kỳ ngôn ngữ viết hoặc nói được phát triển một cách tự nhiên để con người có thể giao tiếp với nhau [1]. XLNNTN là một lĩnh vực con trong khoa học máy tính, kết hợp giữa trí tuệ nhân tạo và ngôn ngữ học tính toán. XLNNTN tập trung xử lý tương tác giữa con người và máy tính sao cho máy tính có thể hiểu hay bắt chước được ngôn ngữ của con người. Ra đời vào những năm 40 của thế kỷ 20, XLNNTN trải qua các giai đoạn phát triển tương ứng với các phương pháp, mô hình xử lý khác nhau như: dựa vào tập luật, dựa vào thống kê, dựa vào học máy, và đặc biệt là học sâu trong thập kỉ vừa qua.

Các công cụ như phân tích, nhận dạng cảm xúc, nhận dạng thực thể định danh, phân tích cú pháp, ngữ nghĩa, ... đã giúp XLNNTN trở thành chủ đề hấp

dẫn để nghiên cứu trong nhiều lĩnh vực khác nhau như dịch máy, trích xuất thông tin, tóm tắt văn bản, trả lời câu hỏi tự động, ... Nhiều ứng dụng XLNNTN trên các thiết bị thông minh xuất hiện ở khắp mọi nơi, thu hút được nhiều sự quan tâm của cộng đồng như Siri của Apple, Google Translate của Google, hay Alexa của Amazon, hệ thống trợ lý ảo Intelligent Personal Agent của Hyundai, nhà thông minh Xiaomi, ...

XLNNTN có thể được chia ra thành hai nhánh lớn, bao gồm xử lý tiếng nói và xử lý văn bản. Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói. Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản, hay kiểm tra lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu văn bản liên quan tới các bài toán phân tích văn bản thì sinh văn bản liên quan tới nhiệm vụ tạo ra văn bản mới [2].

Xử lý tiếng nói và xử lý văn bản không hoàn toàn độc lập mà có mối liên quan với nhau. Văn bản được xử lý tốt giúp hệ thống tổng hợp tiếng nói được thuận lợi, nâng cao độ chính xác. Xử lý tiếng nói cũng tạo ra các văn bản với các đặc điểm riêng. Vấn đề xử lý văn bản sau nhận dạng tiếng nói là một thách thức cần được giải quyết. Luận án cũng đặt ra vấn đề cần chuẩn hoá văn bản và nhận dạng thực thể định danh cho văn bản đầu ra của nhận dạng tiếng nói tiếng Việt.

1.1.2. Xử lý ngôn ngữ tự nhiên tiếng Việt

Theo xu thế phát triển chung của thế giới, XLNNTN tiếng Việt cũng được nghiên cứu hơn một thập kỉ qua với nhiều bài toán khác nhau cho cả xử

lý văn bản và xử lý tiếng nói. Đồng thời, nhiều công cụ đã được công bố giúp hỗ trợ tốt hơn cho các nghiên cứu như: vnTokenizer (hệ tách từ tiếng Việt), Viettagger (hệ gán nhãn từ loại tiếng Việt), VietChunker (hệ phân tích cụm từ tiếng Việt),…

Cộng đồng nghiên cứu đã phát triển mạnh mẽ, có tính gắn kết hơn kể từ khi hội thảo xử lý ngôn ngữ và tiếng nói tiếng Việt *(Vietnamese Language and Speech Processing - VLSP)* được tổ chức lần đầu tiên vào năm 2012. Hội thảo đã trở thành diễn đàn thường niên của cộng đồng nghiên cứu về tiếng Việt. Đây là nơi chia sẻ các kết quả nghiên cứu, tổ chức các cuộc thi đánh giá hiệu quả của các công cụ xử lí tiếng Việt, thu hút được rất nhiều đội tham gia và cho thấy sự lớn mạnh của cộng đồng qua từng năm.

Đáng chú ý là thông qua VLSP, những bộ dữ liệu chuẩn có gán nhãn đã được cung cấp nhằm phục vụ cộng đồng nghiên cứu về xử lý ngôn ngữ và tiếng nói tiếng Việt. Luận án đã sử dụng bộ dữ liệu của VLSP 2018 cho mục đích nghiên cứu.

Mặc dù, XLNNTN đã mang đến công cụ mạnh mẽ với những lợi ích to lớn và đã có những tiến bộ vượt bậc trong những năm gần đây, tuy nhiên, XLNNTN vẫn còn nhiều thách thức, đặc biệt, với ngôn ngữ tiếng Việt.

1.1.3. Những thách thức trong xử lý ngôn ngữ tự nhiên

Kaddari và các cộng sự [3] đã đưa ra một số thách thức đối với lĩnh vực XLNNTN, bao gồm:

Trong hiểu ngôn ngữ tự nhiên, những khó khăn đến từ việc trích xuất ngữ nghĩa từ văn bản, nắm bắt các mối quan hệ ngôn ngữ hoặc ngữ nghĩa giữa các cặp thuật ngữ từ vựng, xác định ngữ cảnh và nghĩa của một từ theo ngữ cảnh, xác định và hiểu ngôn ngữ theo các cách diễn đạt khác nhau, ...

Đối với sinh ngôn ngữ tự nhiên, vấn đề khó khăn gặp phải là thiếu dữ liệu và văn bản tạo ra thiếu mạch lạc, nhất quán.

Ngoài ra, thách thức cho các nghiên cứu trong lĩnh vực này là thiếu bộ dữ liệu, đặc biệt đối với ngôn ngữ có nguồn ngữ liệu hạn chế. Việc sử dụng

các kỹ thuật xử lý ngôn ngữ trên các ngôn ngữ này không mang lại kết quả khả quan như với các ngôn ngữ có tài nguyên phong phú. Thách thức này hiện đang được giải quyết từ nhiều góc độ như sử dụng kĩ thuật học chuyển giao, học tăng cường,...

Các mô hình học sâu cho XLNNTN không đưa ra lời giải thích cho các dự đoán, đây là lý do tại sao các mô hình học sâu này được coi là "hộp đen".

Đồng thời, các mô hình XLNNTN hiện tại không có khả năng phát hiện và diễn giải cảm xúc được thể hiện qua ngôn ngữ, vấn đề này đặc biệt quan trọng trong các hệ thống xử lý tiếng nói.

Bên cạnh những thách thức chung, ngôn ngữ tiếng Việt còn mang những đặc thù riêng của một ngôn ngữ đơn lập, có thanh điệu và các đặc trưng khác gây khó khăn khi xử lý. Cụ thể:

Ngôn ngữ tiếng Việt chứa đựng các từ đồng âm, từ đồng nghĩa, từ mỉa mai, châm biếm. Bên cạnh các từ thuần Việt, tiếng Việt còn có rất nhiều từ vay mượn từ các ngôn ngữ khác để tạo ra từ mới, cũng là một yếu tố khiến ngôn ngữ tiếng Việt trở nên phức tạp hơn. Ngoài ra, đặc trưng vùng miền cũng là một trở ngại trong xử lý tiếng Việt khi có rất nhiều các từ, cụm từ mang tính địa phương cao, chỉ được sử dụng hạn chế ở một số vùng miền (Nghệ An, Hà Tĩnh, Quảng Ngãi, Huế, ...).

Việc nghiên cứu cấu trúc từ (một hay nhiều âm tiết) đóng vai trò rất quan trọng trong quá trình nghiên cứu tiếng Việt. Trong các hệ thống tìm kiếm thông tin văn bản trên các tiếng Châu Âu, người ta có thể xác định các từ nhờ vào các khoảng trắng phân cách từ và chọn các từ đặc trưng cho nội dung văn bản (dựa vào tần suất xuất hiện của từ) làm chỉ mục mà hiệu quả tìm kiếm vẫn chấp nhận được. Đối với tiếng Việt, điều này trở nên khó khăn bởi nếu chỉ xác định từ dựa vào các khoảng trắng phân cách thì có thể chỉ nhận được các tiếng vô nghĩa, do đó độ chính xác của hệ thống sẽ rất thấp. Theo các nhà ngôn ngữ học đã thống kê, tiếng Việt có đến 80% là các từ hai tiếng.

Vấn đề khó khăn tiếp theo có thể kể đến chính là xác định từ loại cho từ trong tiếng Việt phức tạp hơn các tiếng châu Âu do không thể dựa vào các đặc tính đặc biệt về hình thái học của từ để xác định loại từ.

Mặc dù XLNNTN gặp rất nhiều khó khăn, thách thức, nhưng vẫn cho thấy tiềm năng và lợi ích to lớn trên phạm vi rộng cho bất kỳ doanh nghiệp, lĩnh vực nào, với các ứng dụng cụ thể như nhận dạng chữ viết, nhận dạng tiếng nói, tổng hợp tiếng nói, dịch tự động, tóm tắt văn bản, tự động thêm dấu, tách từ, ... Luận án tập trung nghiên cứu một trong những ứng dụng quan trọng trong XLNNTN là nhận dạng thực thể định danh trong văn bản đầu ra của ASR tiếng Việt. Phần tiếp theo sẽ trình bày sơ lược về hệ thống ASR, các đặc trưng của văn bản đầu ra của ASR có thể ảnh hưởng tới nhận dạng thực thể định danh và các nghiên cứu liên quan tới việc chuẩn hóa dạng văn bản này.

1.2. Nhận dạng tiếng nói

1.2.1. Giới thiệu sơ lược về nhận dạng tiếng nói

Nhận dạng tiếng nói được Yu và Deng [4] định nghĩa: *"là một thuật ngữ được sử dụng để mô tả các quy trình, công nghệ và phương pháp cho phép tương tác giữa người và máy tính tốt hơn thông qua việc dịch tiếng nói của con người sang định dạng văn bản"*. Nói một cách ngắn gọn, ASR là cách để máy tính nhận dạng và dịch ngôn ngữ nói thành văn bản. Đó là một cách để con người tương tác với máy tính bằng giọng nói giống như cách con người tương tác với nhau, giúp cho máy tính có thể hiểu mọi từ được nói, trong bất kỳ môi trường nói nào, hoặc bởi bất kỳ người nói nào.

Các nghiên cứu về ASR đã thu hút nhiều sự quan tâm trong nhiều thập kỷ qua nhờ các tiềm năng ứng dụng của nó. Nhiều tiến bộ quan trọng trong công nghệ ASR đã từng bước được chinh phục và ngày càng trở nên phổ biến trong nhiều ứng dụng. Có thể kể đến ở đây là các hệ thống dịch máy tự động như phần mềm Siri của Apple, Google Translate của Google; hệ thống ASR có thể đánh giá độ phát âm chính xác của người học như phần mềm học tiếng

Anh Elsa Speak; tương tác rảnh tay với các thiết bị điện thoại thông minh, ô tô, thiết bị tự động trong gia đình như hệ thống nhà thông minh Xiaomi, trợ lý ảo Intelligent Personal Agent của Hyundai. Ngoài ra, ASR còn được sử dụng để xây dựng các tổng đài trả lời tự động, hệ thống hỗ trợ liên lạc thông tin, ...

Trong các ứng dụng đó, việc xử lý hiệu quả hệ thống ASR liên quan rất nhiều đến việc đánh giá văn bản đầu ra. Một cách phổ biến nhất thường được sử dụng để đánh giá hiệu suất của hệ thống ASR chính là WER. Số liệu WER dựa trên khoảng cách Levenshtein, đo lường số lần chèn, xóa và thay thế trong một chuỗi [5]. Tỷ lệ lỗi từ được tính như sau:

$$WER = \frac{I + D + S}{N} * 100 \qquad (1.1)$$

trong đó, I là số lần chèn, D là số lần xóa, S là số lần thay thế và N là số từ trong văn bản.

Đôi khi, tỷ lệ nhận dạng từ (Word Recognition Rate - WRR) là một biến thể của WER cũng có thể là được sử dụng để đánh giá hiệu suất của ASR và được tính bằng công thức sau:

$$WRR = 1 - WER = \frac{N - S - D - I}{N} = \frac{H - I}{N} \qquad (1.2)$$

trong đó, $H = N - (S + D)$ là tổng số từ được nhận dạng đúng.

1.2.2. Xử lý văn bản đầu ra của hệ thống nhận dạng tiếng nói

Hệ thống ASR đã đạt đến một mức độ tin cậy nhất định, tuy nhiên, văn bản đầu ra của hệ thống ASR còn chứa một số lỗi từ, như:

- Chèn từ: ví dụ "*vấn nạn tin giả trong đợt dịch Covid-19"* nhận dạng thành "*vấn nạn tin giả **mạo** trong đợt dịch Covid-19"*

- Xóa từ: ví dụ "*Thu hẹp khoảng cách số để tiến tới một Việt Nam **số** toàn diện"* nhận dạng thành "*Thu hẹp khoảng cách số để tiến tới một Việt Nam toàn diện"*

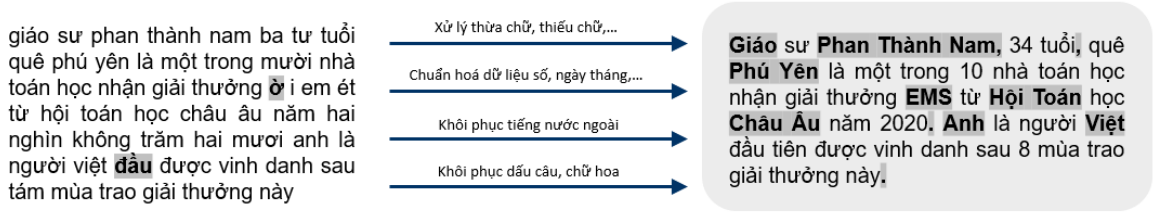
- Thay thế từ: ví dụ "*miền Trung gồng mình tránh bão"* nhận dạng thành "*miền Trung **đồng hành** tránh bão"*

Ngoài ra, các văn bản đầu ra của hệ thống lõi ASR cũng cần được xử lý để có thể sử dụng như văn bản viết thông thường. Bảng 1.1 dưới đây cho thấy các điểm khác biệt giữa văn bản đầu ra ASR và văn bản viết dạng chuẩn, với các ví dụ cụ thể trong tiếng Việt.

Bảng 1.1: Điểm khác biệt giữa văn bản đầu ra ASR và văn bản viết dạng chuẩn

Điểm khác biệt	**Ví dụ**	
	Văn bản gốc	**Văn bản đầu ra ASR**
Văn bản không chứa dấu câu và chữ hoa	*Gần đây, Việt Nam đang tích cực triển khai Giáo dục STEM trong Chương trình Giáo dục Phổ thông.*	*gần đây* ***việt nam*** *đang tích cực triển khai* ***giáo*** *dục* ***stem*** *trong* ***chương*** *trình* ***giáo*** *dục* ***phổ*** *thông*
Các từ tên riêng nước ngoài, các chữ viết tắt không được nhận dạng chính xác	*- kênh Youtube* *- IBM*	*- kênh diu tút/ kênh diu túp/ kênh iu túp/ kênh diu tu be, ...* *- ây bi em/ i bê mờ*
Kiểu số, kiểu tiền tệ nhận dạng thành kiểu chữ cái	*- Việt Nam hướng đến mục tiêu trở thành Quốc gia số vào năm 2030* *- 28$*	*- việt nam hướng đến mục tiêu trở thành quốc gia số vào năm* ***hai nghìn không trăm ba mươi (/hai không ba mươi)*** *- hai tám đô/ hai mươi tám đô la, ...*
Địa chỉ email hoặc địa chỉ website hay các siêu liên kết thường là một cụm từ một liên tục và có quy chuẩn nhưng bị nhận dạng thành các từ, cụm từ không tuân theo quy tắc chuẩn, rất dài và rời rạc	*- vietnamnet.vn là báo điện tử chính thống của Việt Nam* *- địa chỉ email của tôi là hien.math@tnue.edu.vn*	***- việt nam nét chấm vi en (/việt nam nét chấm vê nờ)*** *là báo điện tử chính thống của việt nam* *- địa chỉ email của tôi là* ***hiền chấm mát a còng tê nờ u e chấm e đu chấm vi en***

Tất cả các điểm khác biệt này dẫn đến văn bản ASR khó hiểu và hạn chế khả năng sử dụng văn bản ASR trong rất nhiều ứng dụng XLNNTN như dịch máy, trả lời câu hỏi, trích xuất thông tin, ... Chính vì vậy, để cải thiện khả năng hiểu và sử dụng cho các mục đích tiếp theo, văn bản ASR cần phải được xử lý các lỗi từ, loại bỏ các từ vô nghĩa (ví dụ: à, ừ, ờ) và chuẩn hóa lại bằng cách chuẩn hóa dữ liệu kiểu số, ngày tháng, chuẩn hoá ngôn ngữ nước ngoài và khôi phục dấu câu, viết hoa. Văn bản cuối sẽ có cấu trúc tốt và dễ hiểu hơn so với văn bản ban đầu được tạo bằng ASR. Hình 1.1 dưới đây minh họa về các vấn đề cần thực hiện để tăng chất lượng văn bản đầu ra của hệ thống ASR:



Hình 1.1: Minh họa các vấn đề cần thực hiện để tăng chất lượng văn bản đầu ra của ASR

1.2.3. Hệ thống nhận dạng tiếng nói tiếng Việt

Mặc dù phải đối mặt với nhiều vấn đề khó khăn, đặc biệt là sự hạn chế tài nguyên ngôn ngữ, nhưng với sự nỗ lực của các nhà nghiên cứu, các công ty, tập đoàn trong nước trong thời gian qua như VAIS (Vietnam AI System), Viettel, Zalo, FPT,… các hệ thống ASR tiếng Việt ngày càng được nâng cao chất lượng và đã đạt đến một mức độ tin cậy nhất định. Hiện nay, Việt Nam đã có một số hệ thống nhận dạng tiếng nói như Origin-STT, Viettel[1], Vbee… Năm 2021, trong nghiên cứu đối sánh giữa các hệ thống ASR tiếng Việt tại Việt Nam, Cao Hồng Nga và các cộng sự [6] đã đánh giá các hệ thống ASR tiếng Việt từ các công ty hàng đầu của Việt Nam hiện nay như VAIS, Viettel, Zalo, FPT và công ty hàng đầu thế giới Google cho tin tức, phỏng vấn và âm

[1] https://viettelgroup.ai/service/asr

nhạc. Mặc dù số lượng mẫu còn khiêm tốn nhưng cũng đã cho thấy sự vượt trội của VAIS và Viettel so với các hệ thống còn lại (Bảng 1.2).

Bảng 1.2: Tỉ lệ lỗi từ của một số hệ thống nhận dạng tiếng nói tiếng Việt

Hệ thống ASR	**Bộ dữ liệu đánh giá**	**WER**
VAIS	VLSP 2018	4.85%
	VLSP 2019	15.09%
FPT	FPT-test	9.71%
	VLSP 2018	14.41%
Viettel	Viettel-test	17.44%
	VLSP 2018	6.90%

Có thể nói, tại thời điểm nghiên cứu, hệ thống ASR của VAIS là một trong các hệ thống cho kết quả tốt trên bộ dữ liệu VLSP. Đồng thời, nghiên cứu sinh cũng đã được công ty VAIS đồng ý hỗ trợ sử dụng hệ thống ASR cho mục đích nghiên cứu liên quan đến văn bản đầu ra của hệ thống ASR. Do vậy, các thực nghiệm trong luận án đã sử dụng hệ thống này để đánh giá các mô hình đề xuất.

Đối với hệ thống ASR tiếng Việt, tại VLSP đã sử dụng tỷ lệ lỗi âm tiết (*SyER*) thay vì tỷ lệ lỗi từ để đánh giá hiệu suất của hệ thống ASR [7]. Nguyên nhân là do trong hệ thống chữ viết tiếng Việt, dấu cách được dùng để ngăn cách giữa các âm tiết thay cho các từ. Một từ có thể bao gồm từ một đến sáu âm tiết, và nhiệm vụ tìm ra ranh giới giữa các từ là vô cùng quan trọng. Tỷ lệ lỗi âm tiết được tính như sau:

$$SyER = \frac{S + D + I}{N} \tag{1.3}$$

trong đó, *S* là số lần thay thế, *D* là số lần xóa, *I* là số lần chèn, *C* là số lượng âm tiết đúng và *N* là số lượng âm tiết trong văn bản *N = (S + D + C)*.

Bên cạnh việc tăng hiệu suất của hệ thống ASR thì việc chuẩn hóa văn bản đầu ra của ASR cũng là một vấn đề được nhiều nhà nghiên cứu tập trung cải thiện. Phần tiếp theo, nghiên cứu sẽ trình bày tổng quan về vấn đề này.

1.3. Chuẩn hóa văn bản

1.3.1. Vấn đề khôi phục dấu câu, chữ hoa

Các lỗi chèn, xóa, thay thế từ trong văn bản đầu ra của ASR có thể được cải thiện khi gia tăng hiệu suất của hệ thống ASR. Một khi hệ thống ASR đạt hiệu quả cao thì tỉ lệ lỗi từ sẽ giảm đi đáng kể. Bên cạnh yêu cầu cải thiện hệ thống ASR thì vấn đề khó khăn nhất và luôn được các nhà nghiên cứu tập trung xử lý đó là việc khôi phục dấu câu, chữ hoa. Những dấu hiệu này hoàn toàn bị bỏ qua trong văn bản đầu ra của ASR [8] nhưng lại rất hữu ích trong dịch máy, tóm tắt văn bản hay trích xuất thông tin, ... Việc khôi phục viết hoa bao gồm khôi phục từ đầu tiên của một câu và các danh từ riêng. Viết hoa chính là việc xác định chính xác dạng của từ, phân biệt giữa bốn loại: tất cả các chữ cái viết thường, tất cả các chữ cái viết hoa, chỉ viết hoa chữ cái đầu tiên của âm tiết và chữ hoa hỗn hợp bao gồm một số chữ cái viết hoa và một số chữ cái viết. Đồng thời, trong ngôn ngữ, đối với những câu dài, một cấu trúc ngữ pháp sử dụng nhiều dấu câu sẽ tốt hơn một cấu trúc ngữ pháp tương tự mà bỏ qua các dấu câu. Khôi phục dấu câu là nhiệm vụ chèn chúng vào các vị trí thích hợp trong một văn bản đầu vào không có bất kỳ dấu câu nào.

Hệ thống ASR xử lý đối với hai dạng tiếng nói, một là, tiếng nói dài như bản tin thời sự, bài phát biểu họp Quốc hội, ... hai là, các đoạn hội thoại ngắn như trò chuyện, tin nhắn thoại,...Theo Coniam [9], trong việc xây dựng giao diện người - máy sử dụng ngôn ngữ tự nhiên, hay còn được gọi là *"chatbots"*, một trong những điều khó khăn gặp phải là người sử dụng không nhất quán dấu câu và cách viết hoa. Đồng thời, tác giả lập luận rằng "đối với các câu ngắn do chatbots tạo ra liệu những vấn đề khôi phục dấu câu, chữ hoa có thể được coi là quan trọng nữa hay không". Đặc biệt, trong trường hợp tin nhắn văn bản ngắn (SMS), trò chuyện, hoặc các hoạt động blog khác, mọi người cũng thường bỏ qua cách viết hoa và dấu câu [10]. Chính vì điều này, nghiên cứu trong luận án cũng chỉ tập trung xử lý trên văn bản đầu ra của tiếng nói dài.

Với hệ thống ASR xử lý tiếng nói dài, văn bản đầu ra của ASR không có dấu câu nên thường là các chuỗi dài vô hạn, rất khó để xử lý. Các nhà nghiên cứu khi xử lý vấn đề khôi phục dấu câu, chữ hoa cũng đặc biệt quan tâm tới việc phân đoạn chuỗi câu đầu vào và thường cắt ngẫu nhiên trong khoảng 20-30 từ [11], hay 20-50 từ [12], độ dài tối đa 100 từ [13], 128 từ [14], 150 từ [15],... Việc cắt bao nhiêu thì hợp lý là một vấn đề cần phải xem xét.

Trong ngôn ngữ, đối với những câu dài, một cấu trúc ngữ pháp sử dụng nhiều dấu câu sẽ tốt hơn một cấu trúc ngữ pháp tương tự mà bỏ qua các dấu câu. Khôi phục dấu câu là nhiệm vụ chèn các dấu câu như dấu chấm, dấu phẩy, dấu chấm hỏi, dấu gạch ngang, dấu chấm than,... vào các vị trí thích hợp trong một văn bản đầu vào không có bất kỳ dấu câu nào. Tuy nhiên, vì tần suất dấu phẩy và dấu chấm xuất hiện nhiều hơn những dấu khác nên hầu hết nghiên cứu chỉ tập trung vào những dấu này [16], [17], [18], ...

Viết hoa chính là việc xác định chính xác dạng của từ. Có bốn dạng từ: tất cả các chữ cái viết thường, tất cả các chữ cái viết hoa (thường là trường hợp cho một số cụm từ viết tắt nhất định), chỉ viết hoa chữ cái đầu tiên của âm tiết (các âm tiết bắt đầu của câu và các âm tiết trong các danh từ riêng) và chữ hoa hỗn hợp bao gồm một số chữ cái viết hoa và một số chữ cái viết thường (đây là trường hợp đối với một số danh từ riêng, như “McDonald”). Việc khôi phục viết hoa bao gồm khôi phục từ đầu tiên của một câu và các danh từ riêng (tên của người, tổ chức, địa điểm, ...) [19].

Mặt khác, quy tắc viết hoa chữ cái đầu âm tiết thứ nhất của một câu hoàn chỉnh: sau dấu chấm, sau dấu chấm hỏi, sau dấu chấm than, điều này cho thấy sự liên quan giữa chữ hoa và dấu câu. Các nghiên cứu thường chỉ tập trung giải quyết một nhiệm vụ cụ thể là khôi phục dấu câu hoặc chữ hoa. Kết quả nghiên cứu xử lý đơn lẻ như vậy không thể giúp cải thiện hiệu quả văn bản đầu ra của ASR, dẫn đến gần đây xuất hiện các hướng nghiên cứu tích hợp cả hai nhiệm vụ. Ngay cả khi xử lý tích hợp thì việc xác định khôi phục dấu câu hay chữ hoa trước cũng là một vấn đề vì thứ tự xử lý cũng có thể sẽ

ảnh hưởng lẫn nhau cũng như đến kết quả cuối cùng [15]. Phần tiếp theo, luận án sẽ trình bày về các phương pháp xử lý theo các hướng này.

1.3.2. Các phương pháp xử lý

Một trong những phướng pháp triển khai ban đầu cho viết hoa tự động là dựa trên tập luật, nghĩa là sử dụng nguyên tắc xác định phần bắt đầu của một câu mới để chỉ ra kí tự được viết hoa [20]. Ngoài viết hoa kí tự đầu câu, kí tự đầu tiên của các âm tiết bên trong câu cũng có thể được viết hoa trong trường hợp tên riêng nên cách tiếp cận khả thi hơn đó là dựa vào từ điển. Tuy nhiên, theo Mikheev [21] rất khó để xác định được đúng các danh từ riêng. Chính vì vậy, tác giả đã đề xuất đánh giá các từ khó xác định này trong toàn bộ tài liệu và đưa ra quyết định viết hoa dựa trên kết quả thu thập được. Các nghiên cứu chỉ ra rằng, hệ thống dựa trên luật khó duy trì vì chúng có thể liên tục yêu cầu bổ sung các luật mới.

Mô hình ngôn ngữ là mô hình tính xác suất giúp dự đoán từ tiếp theo trong chuỗi các từ. Mô hình ngôn ngữ tính xác suất của một từ w_k cho trước trong ngữ cảnh của *n-1* từ trước đó $w_{k-1}, w_{k-2}, ..., w_{k-(n-1)}$. Xác suất này có thể được biểu thị bởi $P(w_k| w_{k-1}, w_{k-2}, ..., w_{k-(n-1)})$. Các nghiên cứu về khôi phục dấu câu và mô hình kết hợp dựa trên mô hình ngôn ngữ *n*-gram đã được đề xuất [22]. Các nghiên cứu cho rằng nhược điểm của mô hình *n*-gram là không đánh giá được ngữ cảnh của toàn bộ câu, do đó, trong nhiều trường hợp không thể đưa ra một xác suất chính xác. Đồng thời, ngay cả với các tài nguyên máy tính ngày nay về khả năng lưu trữ và xử lý, các mô hình có số *n* cao vẫn khó xử lý do yêu cầu lưu trữ của chúng. Để sử dụng dễ dàng hơn các mô hình *n*-gram lớn hơn, một số phương pháp cắt dữ liệu đã được đề xuất [23].

Theo các nhà nghiên cứu, viết hoa hay dấu câu có thể được coi là một vấn đề gán nhãn tuần tự. Với một chuỗi $W=w_ow_1w_2...w_n$, mô hình dự đoán chuỗi viết hoa $C=c_0c_1c_2...c_n$ với c_i là AL (All Lowercase), FU (First Uppercase), AU (All Uppercase), MC (Mixed Case) tương ứng với tất cả viết

thường, viết hoa chữ đầu tiên, viết hoa tất cả và viết hoa trộn lẫn. Tương tự, dự đoán dấu câu $E=e_0e_1e_2...e_n$ trong đó e_i biểu thị một dấu câu hoặc không có dấu câu nào. Một số nghiên cứu sử dụng mô hình Entropy cực đại (Maximum Entropy - ME) [24], mô hình Markov ẩn (Hidden Markov Model - HMM) [25] và mô hình Markov Entropy cực đại (Maximum Entropy Markov Model - MEMM) [26] cho cả hai nhiệm vụ. Mặc dù, HMM, MEMM đều là mô hình hữu hạn trạng thái theo xác suất, nhưng nếu HMM chỉ phụ thuộc vào trạng thái hiện tại thì MEMM còn phụ thuộc vào các trạng thái trước đó. Điều đó giúp cho MEMM giải quyết được hạn chế nói trên của mô hình HMM. Tuy nhiên khi áp dụng vào thực tế, với tập dữ liệu huấn luyện khá lớn, khả năng phân nhánh của các trạng thái cao thì tính chính xác của mô hình bị ảnh hưởng rất lớn. Đây chính là hạn chế lớn nhất của mô hình MEMM.

Trường ngẫu nhiên có điều kiện (Conditional Random Field - CRF) cũng là mô hình xác suất được sử dụng để phân đoạn và gán nhãn dữ liệu chuỗi [27]. CRF có ưu điểm hơn so với MEMM và các mô hình Markov khác do CRF là một mô hình đồ thị vô hướng, cho phép CRF có thể định nghĩa phân phối xác suất của toàn bộ trạng thái. Các mô hình sử dụng CRF được đề xuất khôi phục dấu câu [28], viết hoa [29] được cho là cải thiện hơn rất nhiều so với *n*-gram cho cả tiếng Anh và tiếng Trung. Tuy nhiên, hầu như các nghiên cứu khôi phục dấu câu, chữ hoa thường sử dụng kết hợp CRF ở lớp cuối cùng của kiến trúc mạng nơ-ron.

Gần đây, các nghiên cứu đã sử dụng kiến trúc mạng nơ-ron cho bài toán khôi phục dấu câu, chữ hoa. Với tiếp cận mạng nơ-ron, có thể đưa ra mô hình mạng cho cả mức từ và mức ký tự. Trong trường hợp thứ nhất, đầu ra thường được coi như dấu câu theo sau một từ đầu vào. Trường hợp mức ký tự, mô hình dự đoán dấu câu sẽ đưa ra cùng với ký tự trống (dấu cách). Hơn nữa, trong trường hợp đầu vào là các từ, các giá trị mã hóa từ thường được sử dụng. Giải pháp này cho phép tái sử dụng các bộ mã hóa từ đã được tiền huấn luyện giúp nâng cao hiệu năng của mô hình với lượng dữ liệu huấn luyện hạn chế cho bài toán cụ thể.

Susanto và các cộng sự [30] đã đề xuất sử dụng mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) ở cấp ký tự để xử lý sai lệch trong các trường hợp viết hoa trộn lẫn (ví dụ: MacKenzie). RNN đã chứng minh sự hữu ích trong việc lập mô hình dữ liệu tuần tự. Tại mỗi thời điểm bước *t*, nó nhận một véc-tơ đầu vào x_t và trạng thái ẩn trước đó h_{t-1}, và tạo ra trạng thái ẩn tiếp theo h_t. Các công thức lặp lại khác nhau dẫn đến các mô hình RNN khác nhau. Các kết quả cho thấy, phương pháp tiếp cận mức kí tự khả thi cho viết hoa và RNN có hiệu suất cạnh tranh hơn so với CRF ở cùng cấp ký tự. Ngoài ra, nó còn giải quyết hiệu quả những từ nằm ngoài từ điển nhưng khó khăn khi xử lý các câu dài.

Mô hình mạng nơ-ron hồi quy hai chiều (Bidirectional Recurrent Neural Network) có thêm một tầng ẩn cho phép xử lý dữ liệu theo ngữ cảnh dài với chiều ngược lại một cách linh hoạt hơn so với RNN truyền thống. Tilk và các cộng sự [31] đã kết hợp mô hình này với cơ chế chú ý để hướng sự sự chú ý khi cần thiết giúp khôi phục dấu chấm câu đạt hiệu quả tốt hơn trên các tập dữ liệu về tiếng Anh (IWSLT2011) và tiếng Estonia trước đây.

Kể từ năm 2017, với sự ra đời của kiến trúc Transformer [32], các phiên bản khác nhau BERT [33], RoBERTa [34] đã mở ra nhiều hướng nghiên cứu mới. Rei và các cộng sự [35] đã ứng dụng khôi phục viết hoa phụ đề video được tạo bởi hệ thống ASR sử dụng mô hình BERT. Cách tiếp cận này dựa trên mã hóa từ theo ngữ cảnh được huấn luyện trước và áp dụng tinh chỉnh bằng các mô hình tinh chỉnh (fine-tuning). Phương pháp này chứng minh sự vượt trội so với các phương pháp tiếp cận khác không chỉ về hiệu suất mà còn về thời gian tính toán. Nhóm nghiên cứu của Alam [36] đã thử nghiệm một số mô hình Transformer như BERT, RoBERTa, ALBERT, DistilBERT, mBERT, XLM-RoBERTa cho ngôn ngữ giàu tài nguyên (tiếng Anh) và ngôn ngữ hạn chế tài nguyên (tiếng Bangla). Đối với tiếng Anh, các kết quả tốt nhất quan sát được trên mô hình $\text{RoBERTa}_{\text{LARGE}}$ khi khôi phục tốt dấu chấm, tuy nhiên hiệu quả xử lý dấu phẩy và dấu chấm hỏi lại tương đối

thấp. Hiệu suất quan sát được đối với tiếng Bangla thấp hơn so với tiếng Anh được dễ dàng giải thích do thiếu nguồn tài nguyên để huấn luyện.

1.3.3. Khôi phục dấu câu, chữ hoa cho tiếng Việt

1.3.3.1. Đặc điểm dấu câu, chữ hoa tiếng Việt

Trong văn bản, dấu câu giúp xác định rõ cấu tạo ngữ pháp bằng cách chỉ ranh giới giữa các câu, giữa những thành phần của câu đơn, giữa các vế của câu ghép. Trong nhiều trường hợp, dấu câu không chỉ là một phương tiện ngữ pháp, mà còn là một trong những phương tiện để biểu thị những sắc thái tế nhị về nghĩa của câu, về tư tưởng, tình cảm, thái độ của người viết. Khi sử dụng dấu một cách thích hợp thì văn bản sẽ dễ hiểu, ngược lại sẽ dễ gây ra hiểu lầm. Có nhiều trường hợp vì sử dụng sai dấu câu mà thành ra sai nghĩa, thậm chí sai cả ngữ pháp.

Dấu câu trong tiếng Việt đôi khi cũng có những "sự không thống nhất", gây khó khăn cho việc chèn dấu câu một cách chính xác, ngay cả trong văn bản viết. Dưới đây là một số ví dụ các dấu câu thường dùng như dấu chấm, dấu phẩy, dấu hỏi để thấy được những khó khăn riêng của tiếng Việt [37].

- Dấu chấm: Đặt sai vị trí dấu chấm

Ví dụ: *Hồi còn trẻ, học ở trường. Ông là học sinh xuất sắc.*

Câu đúng phải là: *Hồi còn trẻ, học ở trường, ông là học sinh xuất sắc.*

- Dấu phẩy: Trong tiếng Việt, dấu phẩy được sử dụng thường xuyên nhất. Dấu phẩy dùng để xác định ranh giới bộ phận nòng cốt với thành phần ngoài nòng cốt câu.

Ví dụ: *Tôi trở về, thành phố Hồ Chí Minh, thành phố thân yêu của tôi.*

So với: *Tôi trở về thành phố Hồ Chí Minh, thành phố thân yêu của tôi.*

Tuy nhiên, việc chèn dấu phẩy không đúng khiến cho đoạn văn lủng củng, sai nghĩa.

Ví dụ: *Thằng bé di di chân lên mặt, đất không nói gì cả.*

So với: *Thằng bé di di chân lên mặt đất, không nói gì cả.*

Dấu phẩy còn dùng do nhịp điệu trong từng câu, nhất là khi nhịp điệu có tác dụng biểu cảm.

Ví dụ: *Vẫn có Bác, ung dung, trông xuống, dịu dàng.*

- Dấu hỏi: thường được sử dụng ở cuối của mỗi câu nghi vấn.

Tuy nhiên, cũng có trường hợp một vế của câu ghép được cấu tạo theo kiểu câu nghi vấn nhưng không phải dùng để hỏi mà để nêu lên tiền đề, trường hợp này thì việc sử dụng các dấu trong tiếng Việt sẽ không sử dụng câu hỏi

Ví dụ: *Văn học nghệ thuật là gì, xưa nay người ta định nghĩa nhiều rồi.*

Có trường hợp tự đặt ra câu hỏi và tự trả lời

Ví dụ: *Mấy đời bánh đúc có xương?*

Ở Việt Nam, trong công cuộc *"Giữ gìn sự trong sáng của tiếng Việt"* nhằm mục đích thống nhất và chuẩn hóa ngôn ngữ tiếng Việt, vấn đề viết hoa cũng là nội dung quan trọng và được nhiều người quan tâm. Viết hoa đúng theo quy định của tiếng Việt không phải là chuyện đơn giản vì các quy tắc viết hoa liên quan đến viết hoa từ đầu câu, tu từ, danh từ riêng tên người, địa điểm, tên tên tổ chức, đặc biệt là xu hướng viết hoa không theo âm tiết mà theo từ, ... Chỉ xét riêng quy tắc viết hoa cho các danh từ riêng cũng có nhiều nhập nhằng so với các ngôn ngữ khác.

- Cách viết tên người, tên địa điểm sẽ viết hoa chữ cái đầu là phụ âm/âm đầu không dùng gạch nối. Ví dụ: Vũng Tàu, Hà Nội, ... Nhưng thực tế, nhiều người vẫn băn khoăn viết miền Nam hay Miền Nam, Bắc Bộ hay Bắc bộ. Đặc biệt thêm tọa độ như miền cực Nam Trung Bộ hay Miền Cực Nam Trung Bộ hay miền cực nam Trung Bộ, sông Hồng hay Sông Hồng, Đồng Bằng Sông Cửu Long hay đồng bằng sông Cửu Long.

- Tên riêng có kèm theo chức danh cũng là một khó khăn, ví dụ: Nhà giáo Nhân dân, Nhà giáo Ưu tú Lê Thanh Nhàn, ...

- Trong ngôn ngữ dân tộc thiểu số ở Việt Nam, tên riêng không phải tiếng Kinh cũng khó có sự thống nhất. Nhiều tên riêng được viết theo các kiểu khác nhau vẫn tồn tại như Moskva/Moscou/Moscow/Mát-xcơ-va/Matxcơva.

- Tên riêng cơ quan, tổ chức cũng gây nhiều khó khăn do trong tiếng Việt nhiều khi tên gọi của các cơ quan, xí nghiệp, đoàn thể thường rất dài, bao gồm đầy đủ cấp độ của tổ chức, cơ quan đó trong hệ thống. Ví dụ: Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Trường Đại học Sư phạm Thành phố Hồ Chí Minh, ... Có trường hợp viết Nhà hát Tuồng Đào Tấn lại dễ gây ngộ nhận nên đôi khi cần viết là Nhà hát tuồng Đào Tấn, ...

- Xu hướng viết hoa không theo âm tiết mà theo từ ví dụ thay vì Hà Nội, Việt Nam thì có cách viết Hanoi, Vietnam, ...

1.3.3.2. Các nghiên cứu liên quan và thách thức

Vấn đề nghiên cứu khôi phục dấu câu, chữ hoa đối với văn bản đầu ra tiếng nói tiếng Việt vẫn còn khá mới mẻ nên số lượng các công bố nghiên cứu còn hạn chế. Các nghiên cứu khôi phục dấu câu [13], [14], hay kết hợp khôi phục dấu câu và chữ hoa [15], [38] cho tiếng Việt đều sử dụng mô hình mạng nơ-ron học sâu. Điều đáng chú ý là để mô hình nắm bắt được các cấu trúc dữ liệu phức tạp hơn, Thuy Nguyen và cộng sự [13] đã nghiên cứu tích hợp một cơ chế chú ý trên đầu mô hình BiLSTM, giúp tập trung vào các âm tiết cụ thể trong khi dự đoán dấu câu. Hay, Hieu Dinh và cộng sự [14] đã sử dụng mô hình Transformer và thử nghiệm thêm các lớp BiLSTM, lớp CRF trên các mô hình được đề xuất và nâng cao đáng kể hiệu suất khôi phục dấu câu. Bài toán tích hợp hai nhiệm vụ khôi phục dấu câu và chữ hoa gây khó khăn hơn. Các nghiên cứu đều thực hiện theo kiến trúc đường ống, nghĩa là khôi phục chữ hoa trước sau đó mới đến lớp khôi phục dấu câu [15], [38]. Uyen và các cộng sự [15] cũng nhận thấy rằng, một mô hình ngôn ngữ được huấn luyện trước Transformer như vậy sẽ có tham số lớn, gây khó khăn trong mô hình do sự gia tăng độ trễ. Năm 2022, Luong Tran và các cộng sự [38] đã công bố mô hình BARTpho dựa trên BART - là mô hình mới nhất hiện nay cho XLNNTN. Các tác giả đã thử nghiệm để so sánh BARTpho với mBART trong nhiệm vụ khôi phục viết hoa, dấu câu tiếng Việt và nhận thấy rằng BARTpho hiệu quả hơn mBART trong cả hai tác vụ.

Các nghiên cứu cũng thường chỉ sử dụng phân đoạn với độ dài cố định, ví dụ, độ dài 100 [13], độ dài tối đa 128 từ [14], 150 từ [15], ...

Bên cạnh ý nghĩa trong việc cải thiện chất lượng đầu ra của ASR thì dấu câu, chữ hoa cũng là một trong những thông tin quan trọng, hữu ích giúp tối ưu hóa hệ thống nhận dạng thực thể định danh trong văn bản đầu ra ASR. Phần tiếp theo, luận án sẽ trình bày chi tiết về bài toán NER, những khó khăn của bài toán này đối với văn bản đầu ra của ASR tiếng Việt và các vấn đề liên quan trong xử lý bài toán.

1.4. Nhận dạng thực thể định danh

NER là một bài toán tiền đề cho các hệ thống về hiểu ngôn ngữ hay khai phá văn bản, đã được quan tâm nghiên cứu trên thế giới từ đầu những năm 1990. Đến năm 1995, hội thảo quốc tế chuyên đề Message Understanding Conference - MUC lần thứ 6 mới bắt đầu tổ chức đánh giá các hệ thống NER cho tiếng Anh. Tại hội thảo CoNLL năm 2002 và 2003, các hệ thống NER cũng đánh giá cho tiếng Hà Lan, Tây Ban Nha, Đức và Anh. Gần đây, tiếp tục có các cuộc thi về NER được tổ chức như GermEval 2014 cho tiếng Đức hay VLSP cho tiếng Việt từ năm 2012.

1.4.1. Định nghĩa

Trong ngôn ngữ học không có một định nghĩa chính thức thế nào là một thực thể định danh. Với ý tưởng là tìm kiếm trong văn bản tên người, tên tổ chức, địa điểm, thời gian, tiền tệ, ... và mục tiêu là trích chọn trong văn bản các từ, cụm từ có cùng một thể loại, thuật ngữ này được hai tác giả Sundheim và Grishman giới thiệu lần đầu tiên tại hội nghị MUC-6 [39]: *"Nhận dạng thực thể định danh là một quá trình xác định tìm kiếm các từ hoặc cụm từ có nghĩa từ văn bản ngôn ngữ tự nhiên phân loại thành các nhóm duy nhất được định nghĩa trước đó như: tên người, tên tổ chức, ngày giờ, địa điểm, con số, tiền tệ…"*

Aggarwal, C. C [40] cũng đã phát biểu về bài toán nhận dạng thực thể định danh như sau:*"Bài toán nhận dạng thực thể định danh là bài toán xác*

định thực thể có tên từ các văn bản dưới dạng tự do và phân lớp chúng vào một tập các kiểu được định nghĩa trước như tên người, tổ chức và địa điểm."

Thực thể định danh có rất nhiều kiểu khác nhau phụ thuộc vào đặc trưng của loại dữ liệu, miền dữ liệu hay mục đích của hệ thống ứng dụng nhận dạng thực thể. Năm 2011, dự án Quaero đã đưa ra một định nghĩa mở rộng về thực thể định danh, trong đó, các thực thể cơ sở được kết hợp để xác định những thực thể phức tạp hơn. Ví dụ, thực thể tên tổ chức được chia chi tiết hơn là tên tổ chức chính phủ, tổ chức giáo dục hay tổ chức thương mại. Định nghĩa mở rộng được phát biểu như sau: *"nhận dạng thực thể định danh bao gồm việc phát hiện, phân loại và phân tách các thực thể"* [41]. Ngoài các loại thực thể định danh thông thường, các loại thực thể định danh có dạng văn bản của các ngành đặc biệt như y sinh, quân sự cũng nhận được nhiều sự quan tâm.

1.4.2. Tầm quan trọng của bài toán nhận dạng thực thể định danh

Thực thể định danh là một trong những thông tin chính thường được trích chọn để ứng dụng trong các nhiều lĩnh vực khác nhau.

Trong hệ thống hỏi đáp tự động, mục tiêu là tìm câu trả lời trong một đoạn văn bản. Điều quan trọng là phải phát hiện các thực thể định danh trong văn bản vì các câu trả lời thường liên quan đến các thực thể định danh. Theo nghĩa đó, hầu hết các hệ thống hỏi đáp đều kết hợp một số dạng công cụ nhận dạng thực thể định danh, giúp đơn giản hóa công việc một cách đáng kể.

Khi thực hiện khai thác thông tin, nhiều mối quan hệ là sự liên kết giữa các thực thể định danh. Phát hiện ra các thực thể định danh là điều quan trọng đối với hệ thống để có thể trích xuất thông tin liên quan. Việc phân loại sai một thực thể định danh có thể dẫn đến việc trích xuất thông tin sai. Các thực thể định danh cũng có vai trò quan trọng trong quá trình dịch máy. Hệ thống cần phải nhận ra chúng một cách chính xác vì dịch sai hoặc bỏ một thực thể định danh có thể thay đổi ý nghĩa của câu.

Trong tóm tắt văn bản, mục tiêu là trích xuất thông tin liên quan từ các tài liệu. Thông tin liên quan thường bao gồm ngày tháng, địa điểm, con người và tổ chức. Tất cả các danh mục này có thể được phát hiện bằng hệ thống NER. Điều này sẽ đảm bảo rằng hệ thống sẽ không loại trừ thông tin có liên quan quan trọng trong phần tóm tắt.

Đối với hệ thống ASR, theo Yadav và các cộng sự [42] thông tin về thực thể định danh cũng có ý nghĩa quan trọng trong hệ thống khai thác thông tin và hữu ích trong nhiều ứng dụng như tối ưu công cụ tìm kiếm, phân loại nội dung cho các nhà cung cấp tin tức và đề xuất nội dung. Đôi khi, NER từ tiếng nói còn sử dụng cho ứng dụng hỗ trợ quyền riêng tư, ví dụ trong các bản ghi âm y tế cần sử dụng thông tin NER để ẩn thông tin tên bệnh nhân [43].

Hầu hết các công ty, đánh giá trực tuyến được dùng để thu thập phản hồi của khách hàng nhằm phát triển kinh doanh. Ví dụ: sử dụng hệ thống NER để phát hiện các vị trí được đề cập thường xuyên nhất trong phản hồi tiêu cực của khách hàng, điều này có thể giúp chủ doanh nghiệp tập trung vào một chi nhánh văn phòng cụ thể.

Nhiều ứng dụng hiện đại như Netflix, YouTube, Facebook, ... dựa vào hệ thống khuyến nghị để tạo ra trải nghiệm khách hàng tối ưu. Rất nhiều hệ thống này dựa vào nhận dạng thực thể định danh để đưa ra đề xuất dựa trên lịch sử tìm kiếm của người dùng.

1.4.3. Đánh giá hệ thống nhận dạng thực thể định danh

Thước đo đánh giá thích hợp cho hệ thống NER có thể giúp chúng ta phân tích điểm mạnh và điểm yếu của hệ thống và so sánh giữa các kiến trúc với nhau.

Các số đo đánh giá điển hình được sử dụng cho nhận dạng thực thể là độ chính xác (*precision - P*), độ phủ (*recall - R*) và độ đo F1 (*F1-measure*) [44].

$$P = \frac{NE\_true}{NE\_sys} \tag{1.4}$$

$$R = \frac{NE\_true}{NE\_ref}$$

$$F1 = \frac{2 * P * R}{P + R}$$

trong đó: *NE_ref*: là số thực thể trong dữ liệu gốc, *NE_sys*: là số thực thể được đưa ra bởi hệ thống, *NE_true*: là số thực thể được hệ thống gán nhãn đúng.

1.4.4. Thách thức cho bài toán NER trong văn bản đầu ra của ASR tiếng Việt

Để đạt được kết quả tốt, hệ thống NER yêu cầu một lượng dữ liệu đáng kể cho mục đích huấn luyện. Đối với các ngôn ngữ nhiều tài nguyên như tiếng Anh, tiếng Trung, việc lấy dữ liệu không khó, tuy nhiên, điều này không dễ đối với tiếng Việt do chưa có dữ liệu văn bản đầu ra ASR có gán nhãn NER đủ lớn phục vụ cho huấn luyện, đánh giá. Đã có rất nhiều nghiên cứu về NER cho văn bản tiếng Việt thông thường, tuy nhiên, xử lý bài toán này cho văn bản đầu ra của ASR tiếng Việt lại rất hạn chế, điều này khiến cho việc có một bộ dữ liệu công bố chuẩn hay việc so sánh các kết quả thử nghiệm gặp nhiều khó khăn. Những thách thức cho bài toán NER trong văn bản đầu ra của ASR tiếng Việt có thể kể đến như sau:

Các thực thể định danh thường được viết hoa, vì vậy hệ thống dựa vào cách viết hoa để phát hiện chúng. Trong khi đó, các văn bản đầu ra của ASR, việc viết hoa bị bỏ qua gây khó khăn cho hệ thống. Đồng thời, các văn bản đầu ra của ASR không có cấu trúc câu. Vấn đề trong câu không tồn tại bất kỳ một loại dấu câu nào thực sự là một khó khăn và không dễ dàng để phân đoạn hoặc phân tích câu được chính xác.

Việc xác định biên của một từ trong tiếng Việt khó khăn hơn so với các ngôn ngữ khác, do tiếng Việt thuộc loại hình ngôn ngữ đơn lập, tức là, một từ có thể được tạo nên bởi một hoặc nhiều tiếng, ví dụ: *thủ_đô, câu_lạc_bộ, uỷ_ban_nhân_dân, ...*

Yêu cầu hệ thống có khả năng phân biệt loại thực thể. Ví dụ: câu nói *"tôi yêu hà giang"* thì *hà giang* có thể đề cập đến tên người hoặc tên địa danh, tùy thuộc vào ngữ cảnh mà đối tượng đó xuất hiện.

Tên riêng cũng đặt ra những thách thức nhất định cho hệ thống NER. Do không có nhiều ràng buộc về tên riêng nên có thể khiến hệ thống bỏ qua hoặc nhầm nó với một thực thể khác. Ví dụ: *"đây là củ chi"* thì cũng có thể đó là tên của một địa danh là *Củ Chi*, nhưng cũng có thể đó là một câu hỏi *đây là củ gì* theo tiếng địa phương của người miền Trung.

Đặc biệt, lỗi ASR làm cho các thực thể định danh bị bỏ sót hoặc các thực thể định danh bị nhận dạng sai. Nếu một hoặc nhiều từ cấu thành thực thể định danh bị nhận dạng sai thì rất khó để nhận ra đúng thực thể định danh. Ngược lại, ngay cả khi tất cả các từ cấu thành thực thể định danh được nhận dạng chính xác, cũng có thể không nhận ra đúng thực thể định danh do thiếu ngữ cảnh trong văn bản đầu ra của ASR. Ví dụ: *"thời thanh xuân đã qua"* qua hệ thống ASR nhận dạng sai thành *thời anh xuân đã qua* và hệ thống NER nhận dạng *anh xuân* là thực thể định danh.

Tên nước ngoài, tên viết tắt trong văn bản đầu ra ASR cũng có thể bị nhận dạng theo nhiều phiên bản khác nhau, ví dụ: *"Cộng hòa Angola"* khi qua hệ thống ASR có thể nhận dạng thành *cộng hòa ăng gô la/ cộng hòa an gô la/ cộng hòa ăn goa la/ cộng hòa ăng la*, ...

Hiện tượng đồng âm khác nghĩa trong tiếng Việt phổ biến hơn các ngôn ngữ Ấn- Âu, ví dụ: *"trường tôi có nhiều lan"* thì *lan* có thể là thực thể định danh chỉ tên người, nhưng cũng có thể chỉ là cây lan, hoa lan.

1.4.5. Tình hình nghiên cứu NER cho văn bản đầu ra của ASR

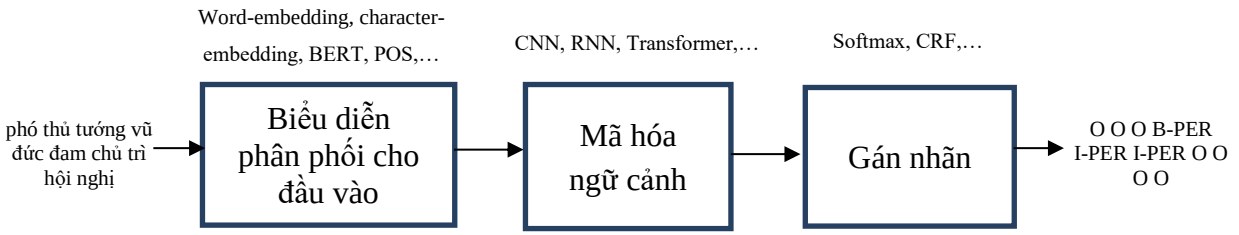
NER là một trong những nền tảng chính để hiểu ngôn ngữ nói. Phương pháp phổ biến để trích xuất các thực thể định danh từ tiếng nói là thông qua phương pháp đường ống. Cách tiếp cận này tuân theo quy trình hai bước, (i) xử lý tín hiệu tiếng nói bằng cách sử dụng hệ thống ASR và xuất ra văn bản tương ứng và (ii) gắn thẻ NER trên văn bản được tạo ra bởi hệ thống ASR.

Gần đây, cách tiếp cận E2E đã được đề xuất với mục đích là gán nhãn trực tiếp các thực thể định danh từ hệ thống ASR [45]. Tổng quan nghiên cứu được trình bày theo từng hướng tiếp cận.

1.4.5.1. Các nghiên cứu liên quan theo hướng tiếp cận đường ống

Theo mô hình đường ống, các nghiên cứu NER cho văn bản đầu ra của ASR được tiếp cận theo cách truyền thống như dựa trên luật, học máy và cách tiếp cận dựa trên học sâu. Trong giai đoạn đầu tiên, Kim và cộng sự [46] đã đề xuất nhận dạng thực thể định danh trên văn bản đầu ra của ASR dựa trên tập luật. Ưu điểm của phương pháp là yêu cầu lưu trữ nhỏ, có thể mở rộng các luật. Tuy nhiên, nhược điểm là các quy tắc cần được xây dựng thủ công, đặc biệt khi đầu vào là văn bản đầu ra của ASR thì thông tin viết hoa cho thực thể định danh sẽ không còn nữa, do đó việc lấy thông tin ngôn ngữ cần thiết để xây dựng các luật sẽ khó khăn. Để khắc phục điều này, rất nhiều các nghiên cứu dựa trên học máy đã được các nhà nghiên cứu đề xuất như mô hình HMM [47], mô hình entropy cực đại (ME) [48], CRF [49], [50], HMM-CRF [51], máy véc-tơ hỗ trợ (SVM) [52] và tập trung chủ yếu cho tiếng Anh, tiếng Trung, tiếng Nhật, tiếng Pháp. Việc kết hợp sử dụng phân đoạn lại *(re-segmentation)*, phân lớp sau *(post-classification)*, sử dụng *n*-best từ hệ thống ASR hay kiến trúc đa tầng cho phép gán nhãn NER theo từng cấp độ đã giúp cải thiện đáng kể các mô hình. Các nghiên cứu cũng chỉ ra rằng cần kết hợp thêm các đặc trưng về âm tiết, kết hợp các thông tin dấu câu, chữ hoa và cải thiện lỗi trong văn bản đầu ra của ASR để tăng hiệu suất NER.

Gần đây, với sự phát triển của học sâu, các nghiên cứu hiện nay về NER tập trung chủ yếu theo hướng này bởi các ưu điểm vượt trội trong khả năng biểu diễn véc-tơ, khả năng tính toán, khả năng ánh xạ phi tuyến tính từ đầu vào đến đầu ra, khả năng học thông tin ngữ nghĩa tiềm ẩn có số chiều lớn và khả năng huấn luyện E2E. Hình 1.2 trình bày mô hình NER dựa trên học sâu, bao gồm biểu diễn phân phối cho đầu vào, mã hóa ngữ cảnh và giải mã nhãn [53].



Hình 1.2: Mô hình NER dựa trên học sâu

- *Biểu diễn phân phối cho đầu vào:* thực chất là biểu diễn văn bản trong một không gian *n* chiều. Biểu diễn từ như GloVe, Word2Vec là các phương pháp gán cùng một véc-tơ được tiền huấn luyện cho cùng một từ không phân biệt ngữ cảnh. Do hiện tượng đa nghĩa cũng như sự phức tạp ngữ nghĩa trong ngôn ngữ tự nhiên nên việc biểu diễn độc lập ngữ cảnh như vậy bị hạn chế. Điều này thúc đẩy sự phát triển của các biểu diễn từ nhạy ngữ cảnh (context-sensitive) - biểu diễn của từ phụ thuộc vào ngữ cảnh của từ đó, phổ biến là biểu diễn ELMo (Embeddings from Language Model) và GPT (Generative pre-training transformers). Trong đó, ELMo là một phương pháp biểu diễn một chuỗi từ dưới dạng một chuỗi véc-tơ tương ứng từ mô hình ngôn ngữ. Cách nhúng ELMo nhạy cảm với ngữ cảnh, tạo ra các cách biểu diễn khác nhau cho các từ có cùng cách viết nhưng có ý nghĩa khác nhau (đồng âm). ELMo mã hóa ngữ cảnh hai chiều nhưng sử dụng các kiến trúc đặc thù cho từng tác vụ. GPT là một họ các mô hình ngôn ngữ của OpenAI thường được huấn luyện trên một khối lượng lớn dữ liệu văn bản để tạo ra văn bản giống con người. GPT có kiến trúc không phân biệt tác vụ nhưng chỉ mã hóa ngữ cảnh từ trái sang phải. Kết hợp những điều tốt nhất của hai phương pháp trên, BERT (Bidirectional Encoder Representations from Transformers) mã hóa ngữ cảnh theo hai chiều và chỉ yêu cầu vài thay đổi kiến trúc tối thiểu cho một loạt các tác vụ XLNNTN [33]. Chính vì vậy, gần đây, biểu diễn véc-tơ từ theo ngữ cảnh được huấn luyện trước như BERT là cách biểu diễn đầu vào được nhiều nhà nghiên cứu ưu tiên lựa chọn cho bài toán NER. Ngoài ra, biểu diễn kí tự cũng thường được sử dụng trong tiếng Anh, trong đó chuỗi kí tự của từ

được mã hóa bằng các mô hình mã hóa tuần tự như RNN, CNN, Transformer để có được biểu diễn cấp kí tự của từ.

- *Mã hóa ngữ cảnh*: Phần mã hóa ngữ cảnh sẽ thực hiện các thao tác tiếp theo trên véc-tơ ngữ nghĩa đã biến đổi, mã hóa các đặc trưng ban đầu, sau đó biểu diễn thông tin ngữ nghĩa của câu. Mã hóa ngữ cảnh chủ yếu được chia thành nhiều phương pháp: CNN, RNN, Transformer, ... Là một công cụ mã hóa ngữ nghĩa, CNN có thể trích xuất thông tin về các ký tự khóa liền kề thông qua hoạt động tích chập, tương tự như ý tưởng của *n*-gram. Tuy nhiên, nó không thể trích xuất mối quan hệ phụ thuộc dài của các từ và thường được sử dụng như một phần của trích xuất đặc trưng để kết hợp với các đặc trưng được trích xuất từ các cấu trúc mạng khác. RNN có thể mô hình hóa sự phụ thuộc đầu vào thông qua lớp ẩn và đầu ra đại diện cho ngữ nghĩa của câu, nhưng do cấu trúc lặp nên sẽ chạy chậm hơn. Trong những năm gần đây, từ mô hình Transformer đã đề xuất một loạt các phương pháp mới cho việc mã hóa ngữ cảnh như GPT, BERT, XLNET, ALBERT, ... và đạt được hiệu quả trong lĩnh vực NER hiện tại.

- *Gán nhãn*: là giai đoạn cuối cùng trong mô hình NER. Các phương pháp của gán nhãn thường sử dụng nhất là Softmax, CRF. Softmax coi vấn đề NER như một bài toán phân loại và dự đoán nhãn của từng từ trong câu. Tuy nhiên, phương pháp xử lý này không hiệu quả vì không tính đến mối liên kết và thông tin trình tự giữa các nhãn. CRF coi nhãn được dự đoán dưới dạng một chuỗi và hiện đang được xem là cách tốt nhất trong bước giải mã nhãn thực thể định danh.

Đối với dữ liệu, Porjazovski và các cộng sự [54] nhận thấy rằng với dữ liệu sau khi xóa dấu câu, chuyển chữ hoa thành chữ thường theo định dạng văn bản đầu ra của ASR đạt kết quả tốt hơn. Như vậy, lỗi của văn bản đầu ra ASR luôn là một thách thức và dữ liệu lớn giúp mô hình đạt hiệu suất cao hơn.

Bên cạnh đó, Mayhew và các cộng sự [55] cũng đề xuất giải quyết vấn đề của hệ thống NER đối với dữ liệu bằng cách huấn luyện trước dự đoán viết hoa trong văn bản trước khi kết hợp với mô hình BiLSTM-CRF cho NER. Đặc biệt, các tác giả đã chọn huấn luyện riêng biệt từng mô hình vì cho rằng, không rõ ràng mô hình chữ hoa mã hóa những gì mô hình NER cần và đảm bảo mô hình chữ hoa được hoạt động độc lập bình thường. Các thử nghiệm theo cả hai hướng BiLSTM-CRF+GloVe và BiLSTM-CRF+BERT có và không có khôi phục chữ hoa. Kết quả cho thấy, không có khôi phục chữ hoa, mô hình có BERT vẫn cho hiệu suất tốt hơn GloVe, tuy nhiên, sử dụng BERT kết hợp với khôi phục chữ hoa, hiệu suất của mô hình được cải thiện hơn. Điều này cho thấy việc kết hợp mô hình khôi phục chữ hoa với mô hình NER có thể cung cấp cho mô hình thông tin bổ sung mà BERT không nắm bắt được.

Có thể nhận thấy, với cách tiếp cận đường ống, thành phần NER phải đối phó với một văn bản không chuẩn hóa như văn bản thông thường và chứa nhiễu (theo nghĩa là trật tự từ có thể bị đảo, các từ có thể bị thiếu hoặc sai chính tả, ...), do đó có tác động lớn đến hiệu suất NER [56]. Cách tiếp cận này sẽ chịu ảnh hưởng của lỗi văn bản đầu ra của ASR và sự lan truyền lỗi qua từng bước [57]. Để xử lý, gần đây các nhà nghiên cứu đã quan tâm tới phương án tiếp cận E2E với mục đích gán nhãn NER trực tiếp từ hệ thống ASR. Mặc dù vậy, phương pháp này vẫn đang còn khá mới mẻ và các công bố còn khá khiêm tốn, ngay cả với ngôn ngữ giàu tài nguyên như tiếng Anh, tiếng Pháp.

1.4.5.2. Các nghiên cứu liên quan theo hướng tiếp cận E2E

Từ suy luận rằng cách tiếp cận đường ống có một số nhược điểm và phương pháp tiếp cận tích hợp sẽ hơn tốt hơn so với các phương pháp tuần tự, các nghiên cứu theo hướng tiếp cận E2E gần đây đã được đề xuất [57], [58], [59], ... Bên cạnh việc đề xuất các mô hình học sâu với kiến trúc đa tầng thì các nghiên cứu cũng huấn luyện E2E bằng cách sử dụng hàm CTC-loss (Connectionist Temporal Classification Loss) [10]. Đây là một hàm mất mát

phổ biến được sử dụng trong các mô hình học sâu, với mục đích giải quyết vấn đề căn chỉnh giữa đầu vào và đầu ra. CTC-loss sẽ tính toán sự mất mát giữa chuỗi thời gian liên tục (không phân đoạn) và chuỗi mục tiêu. Điều này được thực hiện bằng cách tính tổng xác suất sắp xếp có thể có của đầu vào nhằm tạo ra một giá trị tổn thất có thể phân biệt được đối với từng nút đầu vào.

Mặc dù, một số nghiên cứu đã cho thấy hiệu quả của mô hình E2E khi kết hợp với mô hình ngôn ngữ [60] hoặc gia tăng dữ liệu huấn luyện [61] thì hầu hết các nghiên cứu đều cho thấy mô hình E2E chưa thực sự tốt hơn mô hình đường ống về mặt hiệu suất [58], [59]. Cách tiếp cận đường ống giúp đơn giản hóa việc thiết kế mô hình cũng như tận dụng hiệu quả các mô hình đã được xây dựng cho từng bài toán thành phần như nhận dạng dấu câu, chữ hoa và nhận dạng thực thể định danh. Theo Chan và các cộng sự [57], mặc dù các mô-đun trong mô hình đường ống có thể bị ảnh hưởng bởi sự lan truyền lỗi, chúng vẫn có thể tận dụng việc huấn luyện trước để tăng hiệu suất, đặc biệt khi hệ thống ASR được cải thiện tốt.

Tuy nhiên, khi lượng dữ liệu huấn luyện đủ lớn thì các hệ thống lại cần hướng tới xây dựng các mô hình E2E. Điều này giúp tối ưu hóa quá trình huấn luyện, tất cả các tham số của mô hình được huấn luyện đồng thời, các sai số phát sinh giữa các thành phần đều được tính toán do đó giảm thiểu được lỗi lan truyền qua từng mô-đun. Việc huấn luyện và suy luận sử dụng mô hình E2E thuận tiện hơn cho việc đưa mô hình nhận dạng vào ứng dụng. Mặc dù vậy, việc thiết kế mô hình E2E sẽ đòi hỏi sự tích hợp mức độ cao các mô hình thành phần vào một mô hình chung nhất, bỏ qua các khâu trung gian, khiến cho quá trình thiết kế khó khăn hơn. Đồng thời, nó đòi hỏi các thuật toán huấn luyện mô hình nâng cao như phương pháp chia sẻ trọng số (weight tying), huấn luyện đa tác vụ (multitask-learning), ...

1.4.6. Nghiên cứu NER cho văn bản đầu ra của ASR tiếng Việt

Các nghiên cứu ứng dụng NER cho tiếng nói tiếng Việt cũng được các nhà nghiên cứu đề xuất nhưng không nhiều, có thể kể đến ứng dụng trong

tương tác với điện thoại thông minh [62], tuy nhiên chỉ xử lý trên các câu hội thoại ngắn. Theo kết quả tìm kiếm trên Google Schoolar và các nguồn khác, hầu như chưa có các công bố cho các bài toán NER từ văn bản đầu ra của ASR tiếng Việt. Dữ liệu cho tiếng nói tiếng Việt gán nhãn NER đủ lớn là một thách thức khi nghiên cứu vấn đề này. Trong luận án, nghiên cứu sinh cũng đề xuất mô hình đường ống và E2E cho văn bản đầu ra của ASR tiếng Việt để có những đối sánh cụ thể. Các mô hình kết hợp với mô hình khôi phục dấu câu, chữ hoa cho văn bản đầu ra của ASR trước khi đưa vào hệ thống NER với mong muốn bổ sung thêm thông tin hữu ích cho quá trình nhận dạng thực thể định danh. Đồng thời, mô hình E2E đề xuất cũng không theo nghĩa trích xuất trực tiếp NER từ tiếng nói mà trực tiếp từ văn bản đầu ra của hệ thống ASR.

Có thể thấy, thách thức đặt ra cho các bài toán chuẩn hoá văn bản đầu ra của ASR và nhận dạng thực thể định danh theo hướng tiếp cận đường ống, E2E là xây dựng bộ dữ liệu gán nhãn tiếng Việt đủ lớn để thực nghiệm. Phần tiếp theo sẽ giới thiệu tổng quan về các bộ dữ liệu sử dụng trong luận án, các diễn giải chi tiết về từng bộ dữ liệu cho từng mục đích huấn luyện, kiểm thử cho từng mô hình đề xuất sẽ được trình bày cụ thể trong các Chương 3, 4.

1.5. Tổng quan về dữ liệu

Để có nguồn dữ liệu lớn cho mục đích huấn luyện các mô hình, các nghiên cứu đã có nhiều phương án khác nhau. Với bài toán khôi phục dấu câu, chữ hoa, hầu hết các nghiên cứu thực hiện thu thập dữ liệu từ các trang tiểu thuyết [13], tin tức [14], [38], sau đó, các văn bản được bỏ dấu câu, chữ hoa với tỉ lệ lỗi từ là 0% [63], [16], [64].

Về dữ liệu thực nghiệm cho bài toán NER, Mdhaffar và các cộng sự [65] nhận định rằng các mô hình hiểu ngôn ngữ nói (Spoken Language Understanding - SLU) cần một lượng lớn dữ liệu để huấn luyện, trong khi đó, các nghiên cứu phải đối mặt với trường hợp không có sẵn dữ liệu huấn luyện từ tiếng nói và văn bản gán nhãn NER tương ứng. Việc để có thể có một lượng dữ liệu tiếng nói có gán nhãn lớn vẫn còn gặp rất nhiều khó khăn và

không kinh tế. Một số giải pháp gần đây đã được đề xuất để khắc phục vấn đề này. Caubrière và các cộng sự [66] đề xuất áp dụng phương pháp học tập chuyển giao để tận dụng các dữ liệu gán nhãn sẵn có cho các nhiệm vụ SLU chung cho từng nhiệm vụ cụ thể. Trong [67], các tác giả đề xuất tạo tiếng nói tổng hợp để mở rộng tập dữ liệu nhỏ có liên quan tới dữ liệu gán nhãn. Cách tiếp cận này cũng đã được đề xuất để tăng dữ liệu trong ASR [68] hoặc với văn bản đầu ra của tiếng nói [69].

Hiện nay, chưa có một bộ dữ liệu văn bản đầu ra ASR cho tiếng Việt có gán nhãn dấu câu, chữ hoa hay thực thể định danh chuẩn, phục vụ cho mục đính nghiên cứu. Chính vì vậy, luận án cần xây dựng các bộ dữ liệu phù hợp để có thể huấn luyện cho các mô hình đề xuất trong luận án.

Để phục vụ cho mục đích huấn luyện và đánh giá mô hình chuẩn hoá văn bản đầu ra của hệ thống ASR trong Chương 3, nghiên cứu cần xây dựng bộ dữ liệu lớn, tập văn bản này được xóa định dạng (bỏ dấu câu, chuyển chữ hoa thành chữ thường).

Bộ dữ liệu văn bản và âm thanh đã gán nhãn mẫu phục vụ mục đích huấn luyện và đánh giá mô hình cho bài toán NER theo hướng tiếp cận đường ống và E2E trong Chương 4 được tận dụng từ bộ dữ liệu văn bản NER VLSP 2018[2]. Tương ứng với tập văn bản chuẩn này là tập văn bản được xóa định dạng và dữ liệu thu âm với các giọng đọc khác nhau, trong môi trường khác nhau. Đồng thời, để tiết kiệm chi phí thu âm, tất cả dữ liệu văn bản của VLSP sẽ sử dụng hệ thống TTS của Google để tạo ra dữ liệu âm thanh tổng hợp. Sau đó, bộ dữ liệu âm thanh tổng hợp sẽ qua hệ thống ASR của VAIS để được bộ dữ liệu văn bản phục vụ huấn luyện mô hình NER E2E.

Chi tiết về các bộ dữ liệu sẽ được mô tả cụ thể trong Chương 3 và Chương 4.

---

[2] Dữ liệu từ cuộc thi NER tại Hội thảo VLSP (Vietnamese Language and Speech Processing) 2018: https://vlsp.org.vn/vlsp2018/ner

1.6. Kết luận Chương 1

Trong Chương 1 nghiên cứu sinh đã trình bày tổng quan về XLNNTN, các khó khăn trong xử lý ngôn ngữ tiếng Việt. Nhận dạng thực thể định danh là một bài toán quan trọng trong XLNNTN, nhưng lại gặp phải nhiều khó khăn đối với văn bản đầu ra của ASR. Do đó, những nghiên cứu về đặc trưng văn bản đầu ra ASR, các vấn đề cần giải quyết và tổng quan các nghiên cứu liên quan giúp chuẩn hóa văn bản đầu ra ASR đã được trình bày. Bên cạnh việc giới thiệu cơ bản về bài toán NER, tầm quan trọng của bài toán và cách thức đánh giá hệ thống, nghiên cứu cũng đưa ra những thách thức đối với bài toán NER trong văn bản đầu ra của ASR tiếng Việt và các nghiên cứu liên quan để từ đó xác định những nội dung cần giải quyết. Đồng thời, Chương 1 cũng đã giới thiệu tổng quan về các bộ dữ liệu sử dụng trong luận án, việc triển khai chi tiết bộ dữ liệu này tương ứng với từng bài toán và các bộ dữ liệu đặc trưng khác sẽ được giới thiệu cụ thể trong các Chương 3 và Chương 4.

Phần tiếp theo, chương 2 sẽ trình bày những kiến thức nền tảng cho việc nghiên cứu, phát triển các phương pháp hiệu quả cho các mô hình học sâu trong việc chuẩn hoá văn bản và nhận dạng thực thể định danh trong nhận dạng tiếng nói tiếng Việt.

Vấn đề khôi phục chữ hoa, dấu câu cho văn bản đầu ra của ASR giúp tối ưu hệ thống ASR sẽ được tiếp tục trình bày trong Chương 3. Trọng tâm của luận án về nhận dạng thực thể định danh cho văn bản đầu ra của ASR cũng được đề xuất theo hai hướng tiếp cận đường ống, tiếp cận đầu-cuối, trong đó chứng minh được giả thuyết việc kết hợp mô hình khôi phục dấu câu, chữ hoa sẽ giúp cải thiện hiệu suất mô hình NER và các thực nghiệm, kết quả, đối sánh sẽ được giới thiệu chi tiết trong Chương 4.

CHƯƠNG 2: KIẾN THỨC CƠ SỞ

Hiện nay, có rất nhiều mô hình học sâu đã được áp dụng thành công và chứng tỏ hiệu suất cao trong nhiều lĩnh vực và bài toán khác nhau. Mô hình học sâu đóng vai trò quan trọng trong XLNNTN như hiểu ngôn ngữ tự nhiên, dịch máy, phân loại văn bản, sinh văn bản tự động,… Chương 2 trình bày chi tiết về một số mô hình học sâu cho xử lý chuỗi, mô hình biểu diễn từ và mô hình gán nhãn chuỗi. Những kiến thức nền tảng này là cơ sở quan trọng để định hướng việc đề xuất các mô hình chuẩn hoá và nhận dạng thực thể định danh cho văn bản đầu ra của ASR tiếng Việt trong Chương 3, Chương 4. Đồng thời, Chương 2 cũng giới thiệu về phương pháp học đa tác vụ, cho phép một mô hình học được nhiều tác vụ cùng một lúc, giúp mô hình có thể học được nhiều thông tin từ các tác vụ khác nhau và cải thiện khả năng tổng quát hóa. Chương 4 sẽ áp dụng phương pháp này để thiết kế một mô hình nhận dạng thực thể định danh theo hướng E2E.

2.1. Mô hình xử lý chuỗi

Có nhiều mô hình được sử dụng để xử lý chuỗi trong lĩnh vực XLNNTN như HMM, RNN, LSTM. Mô hình RNN và LSTM có khả năng duy trì thông tin ngữ cảnh qua các trạng thái ẩn, nhưng vẫn tồn tại vấn đề mất mát thông tin dài hạn. Với các chuỗi dài, thông tin từ các vị trí xa nhau có thể bị mất đi hoặc không đủ để ảnh hưởng đến quá trình dự đoán. Trong quá trình lan truyền ngược (backpropagation) và huấn luyện, việc tính toán đối với các chuỗi dài có thể trở nên phức tạp và tốn nhiều thời gian. Ngoài ra, cả HMM, RNN và LSTM đều không có khả năng chú trọng vào ngữ cảnh toàn cục trong chuỗi, chỉ xem xét thông tin từ vị trí trước hoặc gần đó. Điều này có thể hạn chế khả năng mô hình hóa mối quan hệ phức tạp giữa các từ trong ngôn ngữ tự nhiên. Mặc dù LSTM được thiết kế để giải quyết vấn đề mất mát thông tin dài hạn trong RNN, nhưng cũng không phải là giải pháp hoàn hảo cho xử lý các chuỗi dữ liệu rất dài.

GRU (Gated Recurrent Unit) là một biến thể của RNN nhằm giải quyết một số hạn chế của RNN và LSTM, như: cấu trúc đơn giản, ít tham số hơn, do đó, có thể thực hiện tính toán nhanh hơn so với LSTM, có khả năng xử lý các chuỗi dữ liệu dài tốt hơn so với RNN truyền thống. Do đó, luận án đã sử dụng GRU cho thiết kế mô hình nhận dạng thực thể định danh theo hướng tiếp cận đường ống. Phần tiếp theo sẽ trình bày chi tiết về mô hình này.

2.1.1. GRU

RNN thích hợp để nắm bắt các mối quan hệ giữa các kiểu dữ liệu tuần tự và có trạng thái ẩn lặp lại (*recurrent hidden state*) như sau [70]:

$$h_t = g(Wx_t + Uh_{t-1} + b) \qquad (2.2)$$

trong đó, x_t là véc-tơ đầu vào *m-chiều* tại thời điểm *t*, h_t là trạng thái ẩn *n-chiều*, g là hàm kích hoạt (theo điểm), chẳng hạn như hàm logistic, hàm tiếp tuyến hyperbol hoặc đơn vị tuyến tính được chỉnh lưu (Rectified Linear Unit - ReLU), và W, U và b lần lượt là các tham số có kích thước thích hợp (hai trọng số và độ lệch). Cụ thể, trong trường hợp này, W là ma trận $n \times m$, U là ma trận $n \times n$, và b là ma trận (hoặc véc-tơ) $n \times 1$.

Có thể nhận thấy rằng, rất khó để nắm bắt sự phụ thuộc khoảng cách xa (*long-term*) bằng cách sử dụng mô hình RNN vì các gradient có xu hướng suy biến hoặc loại bỏ với các chuỗi dài. Mô hình GRU [70] đã được đề xuất để giải quyết vấn đề này. Sự khác biệt chính giữa RNN thông thường và GRU là GRU hỗ trợ việc kiểm soát trạng thái ẩn. Điều này có nghĩa là có các cơ chế để quyết định khi nào nên cập nhật và khi nào nên xóa trạng thái ẩn.

Mô hình GRU giảm tín hiệu cổng thành hai so với mô hình LSTM. Hai cổng được gọi là cổng cập nhật (update gate) z_t và một cổng đặt lại (reset gate) r_t. Dưới đây là công thức tính toán cho hai cổng này:

a. Cổng cập nhật

Đầu vào:

- Đặt h tại thời điểm trước đó: *h(t-1)*
- Đầu vào hiện tại: *x(t)*

Công thức tính toán:

$$z(t) = \sigma\ (W_z * x(t) + U_z * h(t-1) + b_z) \quad (2.3)$$

Trong đó:

- W_z, U_z: ma trận trọng số cho đầu vào $x(t)$ và $h(t-1)$
- b_z: véc-tơ độ lệch (bias)

b. Cổng đặt lại

Công thức tính toán:

$$r(t) = \sigma\ (W_r * x(t) + U_r * h(t-1) + b_r) \quad (2.4)$$

Sau khi tính toán cổng cập nhật và cổng đặt lại, chúng được sử dụng để tính toán hidden state mới tại thời điểm hiện tại:

Công thức tính toán hidden state mới:

$$h'(t) = tanh(W * x(t) + U * (r(t) \odot h(t-1)) + b) \quad (2.5)$$

Trong đó:

- W, U: ma trận trọng số cho đầu vào $x(t)$ và $h(t-1)$
- b: véc-tơ độ lệch (bias)
- $\odot$: phép nhân vô hướng (element-wise multiplication)

Cuối cùng, hidden state mới $h(t)$ được tính bằng cách kết hợp hidden state trước đó và hidden state mới:

$$h(t) = (1 - z(t)) \odot h(t-1) + z(t) \odot h'(t) \quad (2.6)$$

Đây là công thức cơ bản để tính toán cổng cập nhật và cổng đặt lại trong mô hình GRU. Các tham số W, U và b là các ma trận trọng số và véc-tơ độ lệch được học trong quá trình huấn luyện mô hình.

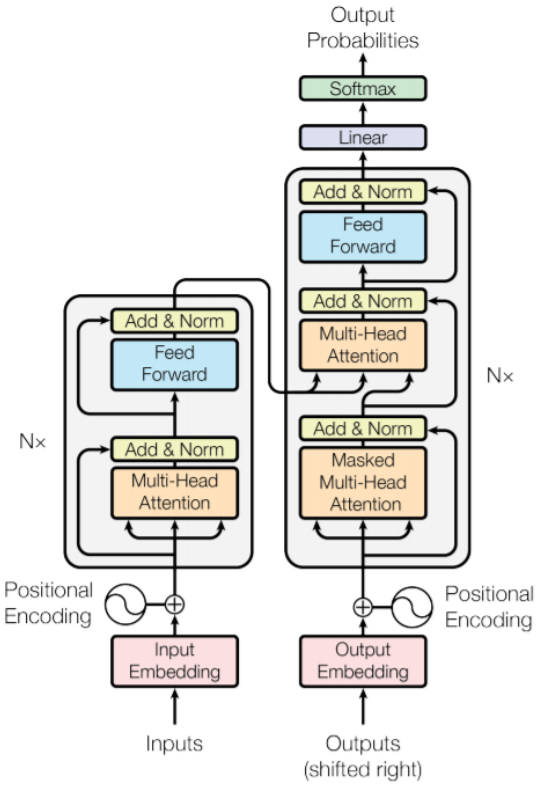
Mặc dù vậy, GRU cũng tồn tại một số hạn chế khi xử lý các chuỗi dữ liệu rất dài như: có khả năng mất mát thông tin quan trọng trong quá trình xử lý chuỗi, vẫn giới hạn về khả năng mô hình hóa mối quan hệ phức tạp trong chuỗi, cần nhiều tham số để huấn luyện, do đó làm tăng yêu cầu về lượng dữ liệu huấn luyện và tài nguyên tính toán.

Sự ra đời của mô hình Transformer đã tạo ra bước đột phá mới, giúp mô hình xử lý hiệu quả với nhiều tác vụ khác nhau, đồng thời hạn chế được

một số nhược điểm của RNN và các biến thể của nó như LSTM hay GRU. Transformer có khả năng chú trọng tất cả các từ trong chuỗi đầu vào, cho phép mô hình có cái nhìn rõ ràng và toàn diện về ngữ cảnh trong chuỗi. Đồng thời, Transformer có khả năng học các mối quan hệ không phụ thuộc chuỗi trong dữ liệu, học biểu diễn từ ngữ phức tạp và mô hình hóa sự tương tác phức tạp giữa các thành phần của chuỗi. Đặc biệt, Transformer có thể được huấn luyện và tính toán song song một cách hiệu quả trên phần cứng nhờ vào sự phụ thuộc không đáng kể giữa các vị trí trong chuỗi, điều này giúp tăng tốc quá trình huấn luyện và dự đoán của mô hình. Chính vì vậy, luận án đã áp dụng mô hình Transformer trong thiết kế mô hình chuẩn hoá văn bản đầu ra của ASR tiếng Việt ở Chương 3. Phần tiếp theo sẽ giới thiệu về kiến trúc và các cơ chế đặc trưng riêng của mô hình.

2.1.2. Transformer

Hình 2.1 mô tả chi tiết kiến trúc mô hình học chuyển giao Transformer do Vaswani và các cộng sự đề xuất [32].



Hình 2.1: Mô hình Transformer [32]

Transformer là mô hình học sâu, trong đó sử dụng cơ chế chú ý (attention) để tính toán ảnh hưởng của các biến đầu vào đến kết quả đầu ra. Mô hình này được dùng phổ biến trong lĩnh vực XLNNTN, tuy nhiên gần đây còn được phát triển cho các ứng dụng khác như thị giác máy, xử lý tiếng nói.

Giống như những mô hình dịch máy khác, kiến trúc tổng quan của mô hình Transformer bao gồm hai phần chính là bộ mã hóa *(Encoder)* và bộ giải mã *(Decoder).* Trong mô hình Transformer, bộ mã hoá chịu trách nhiệm xử lý đầu vào và biểu diễn các từ hoặc câu thành các véc-tơ biểu diễn có ý nghĩa. Bộ giải mã có nhiệm vụ chuyển đổi biểu diễn của đầu vào thành một chuỗi đầu ra.

Mô hình Transformer sử dụng nhiều khối mã hóa và khối giải mã để xử lý dữ liệu. Mỗi khối bao gồm một tầng tự chú ý đa đỉnh (multi-head self-attention) và mạng nơ-ron truyền thẳng (feed-forward network). Tầng tự chú ý đa đỉnh cho phép mô hình học các biểu diễn đa chiều của câu, trong khi mạng nơ-ron truyền thẳng học các biểu diễn phi tuyến của từng vị trí.

Tự chú ý (Self-Attention): là một cơ chế quan trọng trong mô hình Transformer, cho phép mô hình xác định mức độ quan trọng của các từ trong câu bằng cách tính toán một trọng số cho mỗi từ dựa trên tương quan với các từ khác. Điều này giúp mô hình hiểu được mối quan hệ ngữ nghĩa và cú pháp trong câu.

Cơ chế chú ý đa đỉnh (Multi-Head Attention): Trong mô hình Transformer, mỗi tầng tự chú ý sử dụng cơ chế chú ý đa đỉnh. Cơ chế này cho phép mô hình học các biểu diễn đa chiều của câu bằng cách tính toán chú ý từ nhiều không gian biểu diễn khác nhau, giúp tăng khả năng học các mối quan hệ phức tạp trong câu. Việc sử dụng cơ chế chú ý đa đỉnh giúp mô hình học được nhiều khía cạnh khác nhau của câu và cung cấp biểu diễn phong phú hơn cho dữ liệu đầu vào.

a. Bộ mã hoá

Dưới đây là chi tiết về bộ mã hoá:

- Đầu vào và biểu diễn từ (Input Embeddings): Đầu vào của bộ mã hoá là một chuỗi các từ hoặc câu được biểu diễn dưới dạng các véc-tơ từ. Trước khi đi vào bộ mã hoá, các từ đầu vào được chuyển thành các véc-tơ biểu diễn từ. Các véc-tơ từ này có thể được học từ dữ liệu huấn luyện hoặc sử dụng các phương pháp như Word2Vec hoặc GloVe.

- Mã hóa vị trí (Positional Encoding): Trước khi được đưa vào bộ mã hoá, các véc-tơ biểu diễn từ được kết hợp với mã hóa vị trí để cung cấp thông tin về vị trí của từ trong câu. Mã hóa vị trí là một loạt véc-tơ có cùng kích thước với véc-tơ từ và được tính toán dựa trên vị trí tương ứng của từ trong câu.

- Multi-head Self-Attention: Trong quá trình này, mỗi từ trong câu tương tác với các từ khác trong cùng một câu để tính toán trọng số attention cho từng từ. Quá trình attention cho phép bộ mã hoá biết được mức độ quan trọng của các từ trong câu và xây dựng biểu diễn có ý nghĩa.

- Mạng nơ-ron truyền thẳng (Feed-forward Network): Sau quá trình tự chú ý, biểu diễn từ tiếp tục được đưa qua một mạng nơ-ron gọi là mạng nơ-ron truyền thẳng. Mạng này bao gồm hai lớp liên kết đầy đủ với một hàm kích hoạt như ReLU, giúp tăng cường khả năng biểu diễn và khái quát hóa của bộ mã hoá.

- Kết hợp thông tin (Residual Connections): Trong mỗi tầng của bộ mã hoá, thông tin đầu vào ban đầu được kết hợp với đầu ra của quá trình tự chú ý và mạng nơ-ron truyền thẳng thông qua kết hợp thông tin. Kết hợp thông tin giúp truyền thông tin từ đầu vào qua các tầng mã hoá và đảm bảo rằng thông tin quan trọng không bị mất mát trong quá trình biểu diễn.

b. Bộ giải mã

Trong mô hình Transformer, bộ giải mã bao gồm các thành phần:

- Đầu vào và biểu diễn từ: Đầu vào của bộ giải mã là chuỗi các véc-tơ biểu diễn từ, thường là đầu ra của bộ mã hoá hoặc là chuỗi đầu ra đã được sinh ra ở các bước trước trong quá trình giải mã. Tương tự như bộ mã hoá, các véc-tơ biểu diễn từ có thể được học từ dữ liệu huấn luyện hoặc sử dụng các phương pháp như Word2Vec hoặc GloVe.

- Mã hóa vị trí (Positional Encoding): Các véc-tơ biểu diễn từ đầu vào được kết hợp với mã hóa vị trí để cung cấp thông tin về vị trí của từ trong chuỗi. Mã hóa vị trí được tính toán dựa trên vị trí tương ứng của từ trong chuỗi giải mã.

- Tự chú ý: Tương tự như bộ mã hoá, bộ giải mã cũng sử dụng cơ chế tự chú ý để tương tác giữa các từ trong chuỗi đầu vào của mình. Tuy nhiên, bộ giải mã cần chú ý đến tương lai ẩn, có nghĩa là một từ trong chuỗi đầu ra không thể "nhìn thấy" các từ sau nó. Để đạt được điều này, mô hình sử dụng mặt nạ attention.

- Tầng mã hoá và tầng tổng hợp (Encoder and Decoder Layers): Mô hình Transformer sử dụng cả tầng mã hoá và tầng tổng hợp trong bộ giải mã. Mỗi tầng tổng hợp bao gồm quá trình tự chú ý và chú ý giữa bộ giải mã và bộ mã hoá để lấy thông tin từ cả hai phía. Tầng tổng hợp cũng có một mạng nơ-ron truyền thẳng như tầng mã hoá để tăng khả năng biểu diễn của mô hình.

- Mô hình tự hồi quy (Autoregressive Model): Bộ giải mã trong mô hình Transformer được thiết kế dưới dạng một mô hình tự hồi quy. Nghĩa là quá trình giải mã được thực hiện một từ tại một thời điểm. Trong mỗi bước giải mã, từ được dự đoán tiếp theo dựa trên các từ đã được sinh ra trước đó. Quá trình này được lặp lại cho đến khi kết thúc chuỗi đầu ra hoặc đạt đến một giới hạn độ dài đã cho trước.

- Cơ chế chú ý giữa bộ giải mã và bộ mã hoá (Encoder-Decoder Attention): Trong mỗi tầng tổng hợp của bộ giải mã, mô hình thực hiện cơ chế chú ý giữa chuỗi đầu vào của bộ giải mã và đầu ra của bộ mã hoá. Điều này cho phép bộ giải mã truy cập thông tin quan trọng từ chuỗi đầu vào và hướng dẫn quá trình giải mã.

- Mạng nơ-ron truyền thẳng (Feed-forward Network): Tương tự như bộ mã hoá, bộ giải mã cũng sử dụng mạng nơ-ron truyền thẳng sau quá trình chú ý để tăng cường khả năng biểu diễn. Mạng nơ-ron truyền thẳng này giúp mô hình học được các mẫu và đặc trưng phức tạp trong quá trình giải mã.

- Kết hợp thông tin (Residual Connections): Như bộ mã hoá, bộ giải mã cũng sử dụng kết hợp thông tin để truyền thông tin từ đầu vào qua các tầng. Kết hợp thông tin giúp đảm bảo rằng thông tin quan trọng không bị mất mát trong quá trình giải mã và cải thiện khả năng học và khái quát hóa của mô hình.

Mô hình Transformer đã được sử dụng thành công trong nhiều lĩnh vực của xử lý ngôn ngữ tự nhiên, bao gồm dịch máy, tạo tiêu đề, nhận dạng ngôn ngữ, xử lý câu hỏi và trả lời, tóm tắt văn bản, và nhiều ứng dụng khác. Đặc điểm linh hoạt và hiệu quả của Transformer đã giúp nó trở thành một trong những kiến trúc quan trọng trong lĩnh vực XLNNTN.

2.2. Mô hình biểu diễn từ

Trong lĩnh vực XLNNTN, biểu diễn từ (word embedding) là quá trình biểu diễn từ thành các véc-tơ số thực trong không gian đa chiều. Mỗi thành phần trong véc-tơ biểu diễn mô tả một thuộc tính nào đó của từ, ví dụ như ý nghĩa, ngữ cảnh, tần suất xuất hiện, v.v. Việc mã hóa dạng véc-tơ số thực cho các từ giúp máy tính có thể xử lý được các tác vụ liên quan đến ngôn ngữ tự nhiên, đặc biệt là khả năng ứng dụng các mô hình học máy cho XLNNTN..

Biểu diễn từ có thể được tạo ra bằng cách sử dụng nhiều phương pháp khác nhau như Word2Vec, GloVe, FastText, BERT,… được xây dựng sao cho các phép toán véc-tơ giữa các từ cũng có ý nghĩa, các từ có ý nghĩa gần gũi hoặc thường xuất hiện cùng nhau trong văn bản sẽ có các véc-tơ gần nhau. Điều này cho phép mô hình hiểu được các mối quan hệ ngữ nghĩa phức tạp giữa các từ.

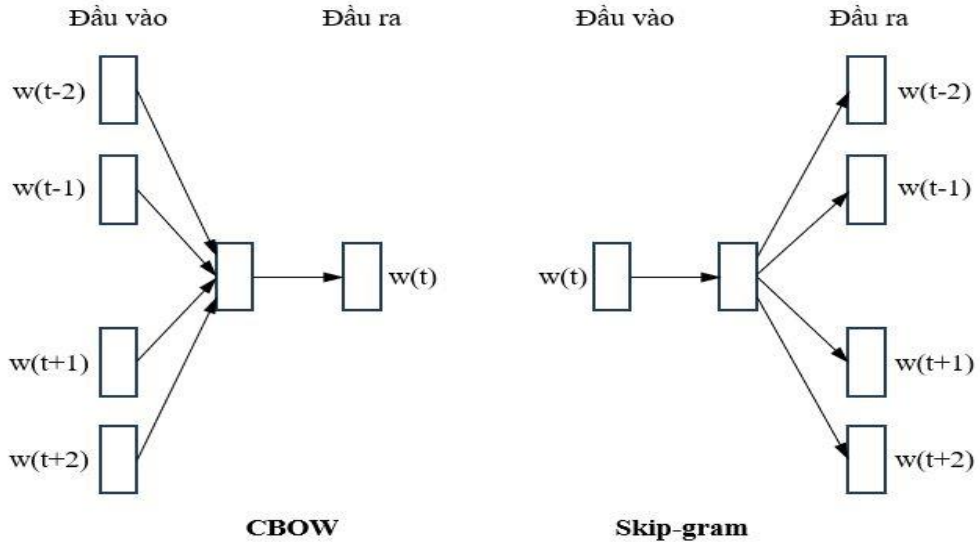
Có thể nhận thấy, việc lựa chọn Word2Vec hay GloVe phụ thuộc vào nhiều yếu tố như tác vụ cụ thể, kích thước dữ liệu, và ngôn ngữ được sử dụng. Về quy mô dữ liệu, Word2Vec thường hoạt động tốt trên các bộ dữ liệu nhỏ và có thể đạt kết quả tốt khi dữ liệu ít. Trong khi đó, GloVe thường được sử dụng trên các tập dữ liệu lớn hơn và có thể tạo ra các biểu diễn từ vựng phong phú hơn. Đối với tính cú pháp và ngữ nghĩa, nếu Word2Vec thường cho kết quả tốt hơn trong việc “bắt chước” các mối quan hệ ngữ nghĩa như "vua - nữ hoàng" hay "ông - bà" thì GloVe có xu hướng tạo ra các biểu diễn vector chứa nhiều thông tin về cú pháp và tần suất từ. Mặt khác, Word2Vec cần một số lượng lớn

vòng lặp để huấn luyện, trong khi GloVe thường huấn luyện nhanh hơn vì sử dụng một ma trận tần số từ có kích thước nhỏ hơn. Đặc biệt, Word2Vec thường không lưu trữ nhiều thông tin về cú pháp của từ, tập trung chủ yếu vào ngữ nghĩa. Chính vì vậy, để phù hợp với bộ dữ liệu và mục đích chuẩn hoá văn bản đầu ra của ASR tiếng Việt, luận án đã lựa chọn sử dụng Word2Vec cho các mô hình đề xuất, phần tiếp theo sẽ trình bày chi tiết về kĩ thuật này.

2.2.1. Word2Vec

Được phát triển bởi Tomas Mikolov và các cộng sự tại Google vào năm 2013, Word2Vec là một kỹ thuật biểu diễn véc-tơ từ để giải quyết các vấn đề XLNNTN nâng cao. Nó có thể lặp lại trên một kho văn bản lớn để tìm hiểu các liên kết hoặc sự phụ thuộc giữa các từ. Word2Vec xác định mối quan hệ ngữ nghĩa giữa từ bằng cách dự đoán từ hiện tại dựa trên ngữ cảnh xung quanh nó hoặc ngược lại. Kết quả của Word2Vec là các biểu diễn véc-tơ từ, có thể được sử dụng trong các mô hình học máy khác nhau [71].

Word2Vec cung cấp hai biến thể dựa trên mạng nơ-ron: CBOW (Continuous Bag of Words) và Skip-gram. Hình 2.2 dưới đây minh hoạ về hoạt động của CBOW và Skip-gram [71].



Hình 2.2: Minh hoạ hoạt động của CBOW và Ship-Gram

CBOW dự đoán từ hiện tại dựa trên ngữ cảnh xung quanh nó. Đầu vào của CBOW là một cửa sổ các từ xung quanh từ hiện tại và mục tiêu là dự

đoán từ hiện tại. Skip-gram, ngược lại với CBOW, Skip-gram cố gắng dự đoán ngữ cảnh xung quanh từ hiện tại dựa trên từ hiện tại. Skip-gram lấy từ hiện tại và dự đoán các từ trong ngữ cảnh xung quanh nó.

Cả CBOW và Skip-gram đều xây dựng trên ý tưởng rằng các từ có xu hướng xuất hiện cùng nhau trong cùng một ngữ cảnh sẽ có ý nghĩa tương đồng. Khi mô hình Word2Vec được huấn luyện, các véc-tơ embedding từ được học sao cho các từ có cùng ngữ cảnh gần nhau trong không gian embedding.

Để huấn luyện Word2Vec, cần thực hiện các bước sau:

– Chuẩn bị dữ liệu: Dữ liệu huấn luyện cho Word2Vec là một tập văn bản lớn. Dữ liệu này có thể là một tập các văn bản tự do, từ các nguồn như sách, bài báo, trang web, v.v. Trước khi huấn luyện, dữ liệu cần được tiền xử lý bằng cách loại bỏ các ký tự đặc biệt, chuyển đổi chữ hoa thành chữ thường,… và thực hiện các bước xử lý ngôn ngữ tự nhiên khác.

– Xây dựng từ điển: Trước khi huấn luyện Word2Vec, cần xây dựng một từ điển từ vựng từ tập dữ liệu huấn luyện. Từ điển này sẽ định danh và gán một chỉ số duy nhất cho mỗi từ trong tập dữ liệu.

– Tạo cặp từ - ngữ cảnh: Trong quá trình huấn luyện Word2Vec, cặp từ - ngữ cảnh được tạo từ các câu trong tập dữ liệu. Một cặp từ - ngữ cảnh gồm một từ đích (target word) và các từ xung quanh nó trong ngữ cảnh. Kích thước của ngữ cảnh được xác định bằng cửa sổ trượt (window size), ví dụ: nếu cửa sổ trượt là 2, thì các từ xung quanh từ đích trong khoảng 2 từ sẽ được lấy làm ngữ cảnh.

– Xây dựng mô hình CBOW hoặc skip-gram: Sau khi tạo các cặp từ - ngữ cảnh, ta sẽ sử dụng chúng để huấn luyện mô hình Word2Vec. Mô hình CBOW và skip-gram được xây dựng dựa trên mạng nơ-ron đa tầng. Trong quá trình huấn luyện, các biểu diễn véc-tơ từ sẽ được cập nhật để giảm thiểu sai số giữa dự đoán và mục tiêu thực tế. Quá trình huấn luyện thường sử dụng các phương pháp tối ưu hóa như stochastic gradient descent (SGD) để điều chỉnh các trọng số.

– Trích xuất biểu diễn véc-tơ từ: Sau khi huấn luyện hoàn thành, các biểu diễn véc-tơ từ có thể được trích xuất từ mô hình. Các véc-tơ này có thể được sử dụng để biểu diễn từng từ trong không gian.

Sau khi đã trích xuất các biểu diễn véc-tơ từ từ mô hình Word2Vec, chúng có thể được sử dụng để thực hiện các tác vụ trong XLNNTN.

Khi có một lượng dữ liệu lớn và cần mô hình học biểu diễn từ ngữ phức tạp, giúp nắm bắt được các mối quan hệ tương quan giữa từ trong câu, hiểu được ý nghĩa của từ trong ngữ cảnh cụ thể và tạo ra các biểu diễn phù hợp thì các mô hình học sâu trở lên phù hợp hơn. Với sự ra đời của mô hình Transformer, nhiều biến thể mới được mở rộng và đạt được nhiều thành công trong nhiều tác vụ XLNNTN, bao gồm phân loại văn bản, dịch máy, trích xuất thông tin và nhiều tác vụ khác. Luận án đã cải tiến mô hình BERT cho dữ liệu tiếng Việt khi đề xuất mô hình nhận dạng thực thể định danh. Phần tiếp theo sẽ trình bày kiến thức cơ sở về BERT.

2.2.2. Mô hình BERT

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình ngôn ngữ học sâu, được giới thiệu bởi Jacob Devlin và các cộng sự tại Google Research vào năm 2018.

Kiến trúc chung: Mô hình BERT có kiến trúc mạng học sâu sử dụng nhiều tầng mã hoá Transformer. Tuy nhiên, điểm đặc biệt của BERT là sử dụng hai biểu diễn từ: biểu diễn từ đầu vào (input representation) và biểu diễn từ đầu ra (output representation) [72].

Tiền huấn luyện: BERT được huấn luyện trước trên dữ liệu lớn và không có nhãn sẵn, quá trình này gọi là tiền huấn luyện. Trong tiền huấn luyện, mô hình học cách dự đoán từ bị ẩn đi trong một ngữ cảnh câu. Điều này giúp mô hình hiểu được mối quan hệ giữa các từ trong câu và xây dựng một biểu diễn từ phong phú.

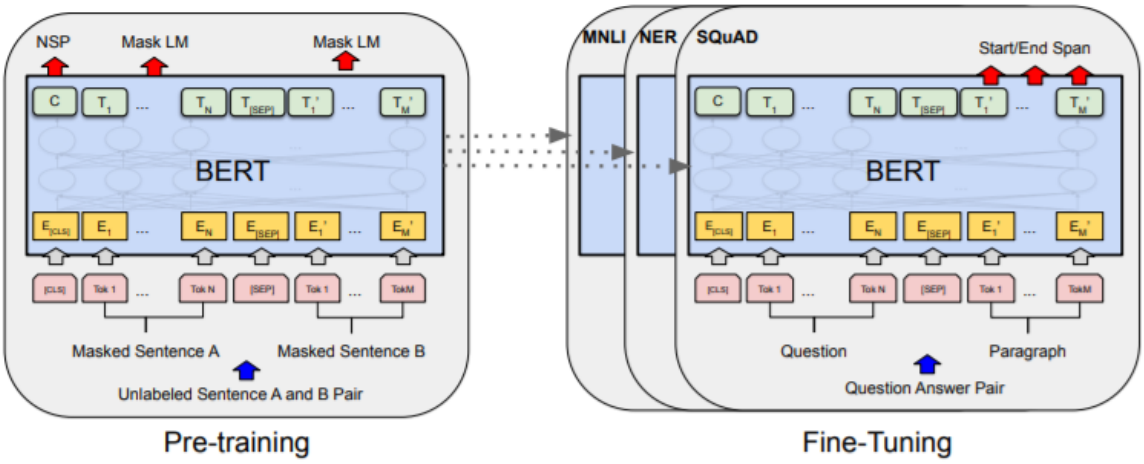
Tinh chỉnh (Fine-tuning): Sau quá trình tiền huấn luyện, mô hình BERT được tinh chỉnh trên các tác vụ cụ thể. Tinh chỉnh là quá trình huấn luyện tiếp

theo trên một tập dữ liệu có nhãn sẵn cho các tác vụ như phân loại văn bản, dịch máy, trích xuất thông tin, và nhiều tác vụ khác. Quá trình tinh chỉnh giúp mô hình BERT chuyển đổi biểu diễn từ thông qua việc điều chỉnh các tham số để phù hợp với các tác vụ cụ thể.

Biểu diễn từ đầu vào: Để biểu diễn một câu đầu vào, mô hình BERT sử dụng sự kết hợp của hai thành phần: biểu diễn từ (word embedding) và biểu diễn vị trí (position embedding). Biểu diễn từ là một véc-tơ số hóa từ đưa vào mô hình, còn biểu diễn vị trí là một véc-tơ số hóa vị trí của các từ trong câu. Mô hình BERT sử dụng phép cộng của hai thành phần này để tạo ra biểu diễn từ vào.

Biểu diễn từ đầu ra: Biểu diễn từ ra trong BERT là biểu diễn của các từ được dự đoán trong quá trình tiền huấn luyện. Khi huấn luyện BERT, một số từ trong câu đầu vào được ngẫu nhiên che đi và mục tiêu của mô hình là dự đoán các từ bị che bởi các từ xung quanh. Điều này giúp mô hình học được biểu diễn ngữ nghĩa của từ dựa trên ngữ cảnh xung quanh.

BERT là một phương pháp mới để tiền huấn luyện các bộ biểu diễn véc-tơ từ. Một điểm đặc biệt ở BERT mà các mô hình biểu diễn véc-tơ từ trước đây chưa từng có đó là kết quả huấn luyện có thể tinh chỉnh được. Hình 2.3 mô tả quy trình tiền huấn luyện và tinh chỉnh cho BERT [33].



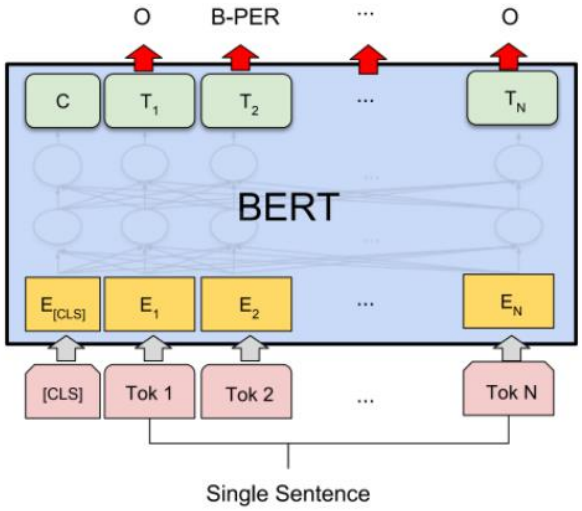
Hình 2.3: Tổng thể quy trình tiền huấn luyện và tinh chỉnh cho BERT [33]

Quy trình tiền huấn luyện và tinh chỉnh trong BERT cho phép mô hình học cách biểu diễn ngôn ngữ tổng quát thông qua huấn luyện không giám sát và

sau đó áp dụng kiến thức này vào các tác vụ cụ thể. Điều này giúp cải thiện khả năng hiểu và xử lý ngôn ngữ tự nhiên của mô hình trên nhiều tác vụ khác nhau.

Khi BERT được tinh chỉnh trong một nhiệm vụ nào đó, bộ Transformer tiền huấn luyện sẽ hoạt động như một bộ mã hóa và một bộ phân loại được khởi tạo ngẫu nhiên được thêm vào trên cùng. Trong trường hợp NER, trình phân loại chỉ đơn giản là một phép chiếu từ kích thước các từ đến kích thước tập nhãn, toán tử Softmax tiếp theo thực hiện chuyển điểm số thành xác suất của nhãn.

Taher và các cộng sự [73] đã minh họa BERT trên nhiều nhiệm vụ khác nhau này. Hình 2.4 mô tả quá trình tinh chỉnh cho NER.



Hình 2.4: Tinh chỉnh BERT cho nhiệm vụ NER [33]

Có nhiều phiên bản khác nhau của mô hình BERT, các phiên bản đều dựa trên việc thay đổi kiến trúc của Transformer tập trung ở ba tham số: L: số lượng các block sub-layers trong Transformer, H: kích thước của biểu diễn véc-tơ từ (hay còn gọi là kích thước ẩn), A: Số lượng đỉnh trong lớp chú ý đa đỉnh, mỗi một đỉnh sẽ thực hiện một thao tác tự chú ý. Tên gọi của 2 kiến trúc bao gồm:

BERTBASE(L=12, H=768, A=12): Có 110 triệu tham số

BERTLARGE(L=24, H=1024, A=16): Có 340 triệu tham số

Các kiến trúc biến thể mới của BERT hiện tại vẫn đang được nghiên cứu và tiếp tục phát triển như RoBERTa [34], ALBERT, CameBERT, ...

2.3. Mô hình gán nhãn chuỗi

2.3.1. Softmax

Softmax là một hàm kích hoạt thường được sử dụng trong các mô hình phân loại đa lớp để chuyển đổi đầu ra của mạng thành một phân phối xác suất. Softmax thường được áp dụng cho lớp đầu ra cuối cùng của mô hình để tính toán xác suất dự đoán cho mỗi lớp.

Hàm softmax được định nghĩa cho một véc-tơ đầu vào có kích thước *K* như sau [74]:

$$S_i = \frac{e^{x_i}}{\sum_j^{\infty} e^{x_j}} \qquad (2.7)$$

trong đó, x_i đại diện cho đầu vào của một véc-tơ gồm *K* số thực, S_i đại diện cho kết quả chuẩn hóa các đầu vào thành một phân phối xác suất gồm *K* xác suất tỉ lệ với các giá trị mũ. Do phép mũ, S_i luôn là giá trị dương. Khi áp dụng cho phân phối xác suất từ một số lượng lớn đặc trưng, hàm softmax thường được đặt sau lớp fully connected.

Hàm softmax chuyển đổi các giá trị đầu vào thành một phân phối xác suất, trong đó giá trị đầu ra của mỗi lớp nằm trong khoảng từ 0 đến 1 và tổng của tất cả các giá trị đầu ra bằng 1. Điều này cho phép coi các giá trị đầu ra như xác suất dự đoán cho mỗi lớp.

Khi sử dụng hàm softmax trong một mô hình phân loại đa lớp, đầu ra của mô hình sẽ là một vectơ xác suất có cùng kích thước với số lượng lớp. Giá trị tương ứng với mỗi lớp trong vectơ đầu ra thể hiện xác suất dự đoán cho lớp đó. Lớp với xác suất cao nhất sẽ được chọn là lớp dự đoán.

Hàm softmax thường được sử dụng kết hợp với hàm cross-entropy để đo lường sự khác biệt giữa phân phối xác suất dự đoán và phân phối xác suất thực tế của các lớp. Việc tối thiểu hóa hàm cross-entropy thông qua việc điều chỉnh các trọng số mạng sẽ tạo ra một mô hình phân loại có khả năng dự đoán tốt.

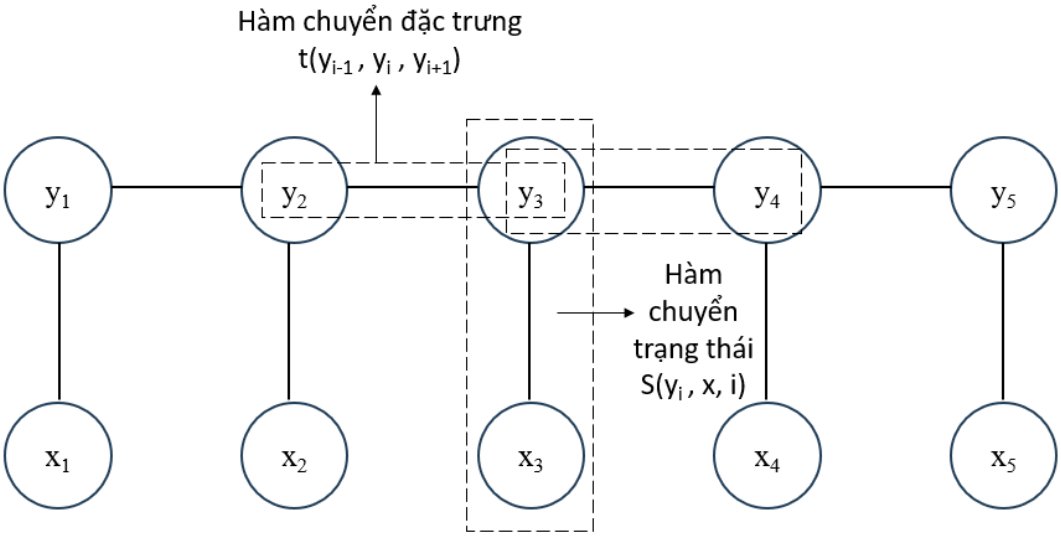
Một lợi ích quan trọng của hàm softmax là nó là một hàm liên tục và khả vi, điều này rất hữu ích trong việc tính toán đạo hàm để cập nhật các

trọng số trong quá trình huấn luyện mạng nơ-ron. Việc sử dụng hàm softmax không chỉ hữu ích trong các tác vụ phân loại đa lớp, mà còn có thể được áp dụng trong các bài toán khác như xác định mức độ tin cậy của dự đoán hoặc tạo ra một phân phối xác suất từ các giá trị đầu vào.

Tuy nhiên, hàm softmax cũng có một số hạn chế. Khi số lượng lớp rất lớn, việc tính toán và xử lý đồng thời các giá trị mũ có thể trở nên phức tạp và tốn nhiều thời gian tính toán. Đồng thời, hàm softmax không kháng nhiễu, có nghĩa là nếu có sự biến động mạnh trong giá trị đầu vào, các giá trị xác suất đầu ra có thể dễ dàng bị lệch và dẫn đến sai lệch trong dự đoán.

2.3.2. Trường ngẫu nhiên có điều kiện

Trường ngẫu nhiên có điều kiện (Conditional Random Fields - CRFs) được đề xuất bởi Lafferty và đồng nghiệp vào năm 2001. Đây là một mô hình đồ thị xác suất vô hướng, kết hợp các đặc điểm của mô hình Markov ẩn và mô hình entropy tối đa. CRFs là một trường hợp đặc biệt của mô hình Markov ngẫu nhiên, giải quyết vấn đề thiên vị nhãn do mô hình Markov ẩn gây ra. Ngoài ra, đặc điểm ngữ cảnh có thể được xem xét để lựa chọn đặc trưng tốt hơn. CRFs được sử dụng để tính toán mật độ phân phối xác suất điều kiện của một tập hợp biến ngẫu nhiên đầu ra khác dựa trên một tập hợp biến ngẫu nhiên đầu vào. Một mô hình CRFs chung được thể hiện trong Hình 2.5 [75]



Hình 2.5: Mô hình Conditional Random Fields

Trong mô hình trên, chuỗi quan sát được biểu diễn bởi x_1 , x_2 ,..., x_T và chuỗi trạng thái ẩn được biểu diễn bởi y_1, y_2,..., y_T , vì vậy hình ảnh trên đại diện cho giá trị quan sát x_i và y_{i-1}, y_i, y_{i+1} có liên quan.

Ý tưởng cơ bản của CRF là mô hình phân bố xác suất có điều kiện của các biến đầu ra (ví dụ: nhãn) cho trước các biến đầu vào (ví dụ: đặc trưng). Biểu diễn toán học như sau:

$$p(y|x) = \frac{1}{Z(x)} * \exp\left(\sum i \, \varphi_{i(y_i, y_{i-1}, x_i)}\right) \quad (2.8)$$

trong đó:

$y = (y_1, \dots, y_n)$ là một chuỗi các biến đầu ra (ví dụ: nhãn)

$x = (x_1, \dots, x_n)$ là một chuỗi các biến đầu vào (ví dụ: đặc trưng)

Z(x) là một hệ số chuẩn hóa đảm bảo các xác suất có tổng bằng 1 trên tất cả các chuỗi đầu ra có thể có

- $\varphi_{i(y_i, y_{i-1}, x_i)}$ là một hàm đặc trưng ánh xạ biến đầu ra hiện tại y_i, biến đầu ra trước đó $y_{\{i-1\}}$ và biến đầu vào tương ứng x_i thành một điểm số có giá trị thực.

Có thể nhận thấy, tùy thuộc vào yêu cầu cụ thể của tác vụ để có thể lựa chọn CRF hoặc Softmax. Với Softmax, quá trình huấn luyện và dự đoán thường nhanh hơn do tính đơn giản và không yêu cầu tính toán phức tạp như CRF. Đồng thời, Softmax thường được sử dụng cho các tác vụ đơn giản hơn và có dữ liệu huấn luyện ít hơn. Với CRF, giải quyết được sự phụ thuộc ngữ cảnh trong chuỗi và tạo ra các chuỗi nhãn liên tục hơn, phù hợp với các tác vụ gán nhãn chuỗi. Trong khi đó, Softmax xử lý mỗi nhãn độc lập, không có khả năng mô hình hóa mối quan hệ giữa các nhãn. Mặt khác, CRF có khả năng xử lý các chuỗi dữ liệu dài hơn so với Softmax. Nếu ở Softmax, mỗi nhãn độc lập được dự đoán độc lập và không có thông tin về ngữ cảnh toàn bộ chuỗi thì CRF có khả năng xem xét các nhãn trước đó trong chuỗi, giúp tạo ra các chuỗi nhãn liên tục và giải quyết các vấn đề như hiện tượng phụ thuộc trên phạm vi dài (long-range dependencies). Đặc

biệt, CRF thường được sử dụng để đánh giá cùng lúc nhiều nhãn trong chuỗi dữ liệu. Điều này có lợi khi cần đánh giá và tối ưu toàn bộ chuỗi nhãn một cách toàn diện, thay vì chỉ xem xét từng nhãn độc lập. Chính vì vậy, luận án đã lựa chọn sử dụng CRF để gán chuỗi trong các mô hình đề xuất của các bài toán chuẩn hoá văn bản và nhận dạng thực thể định danh cho văn bản đầu ra ASR tiếng Việt.

2.4. Học đa tác vụ

Con người có thể học nhiều nhiệm vụ cùng một lúc. Trong quá trình học tập, con người có thể sử dụng những kiến thức đã học trong một nhiệm vụ để học một nhiệm vụ khác. Lấy cảm hứng từ khả năng học tập của con người, học đa tác vụ có mục đích là cùng học nhiều nhiệm vụ liên quan để kiến thức chứa trong một nhiệm vụ có thể được tận dụng bởi các nhiệm vụ khác với hy vọng cải thiện hiệu suất tổng quát hóa của tất cả các nhiệm vụ [76].

Với giả thuyết rằng, mô hình khôi phục dấu câu, chữ hoa có thể cung cấp thêm các thông tin, hỗ trợ tốt hơn và giúp nâng cao hiệu quả nhận dạng thực thể định danh, luận án đã tận dụng tri thức về các phương pháp học tập đa tác vụ và tác vụ phụ trợ để đề xuất mô hình nhận dạng thực thể định danh cho văn bản đầu ra của ASR theo hướng E2E. Vậy MTL là gì? có những phương pháp nào? ý nghĩa của tác vụ phụ trợ? sẽ tiếp tục được nghiên cứu trình bày dưới đây.

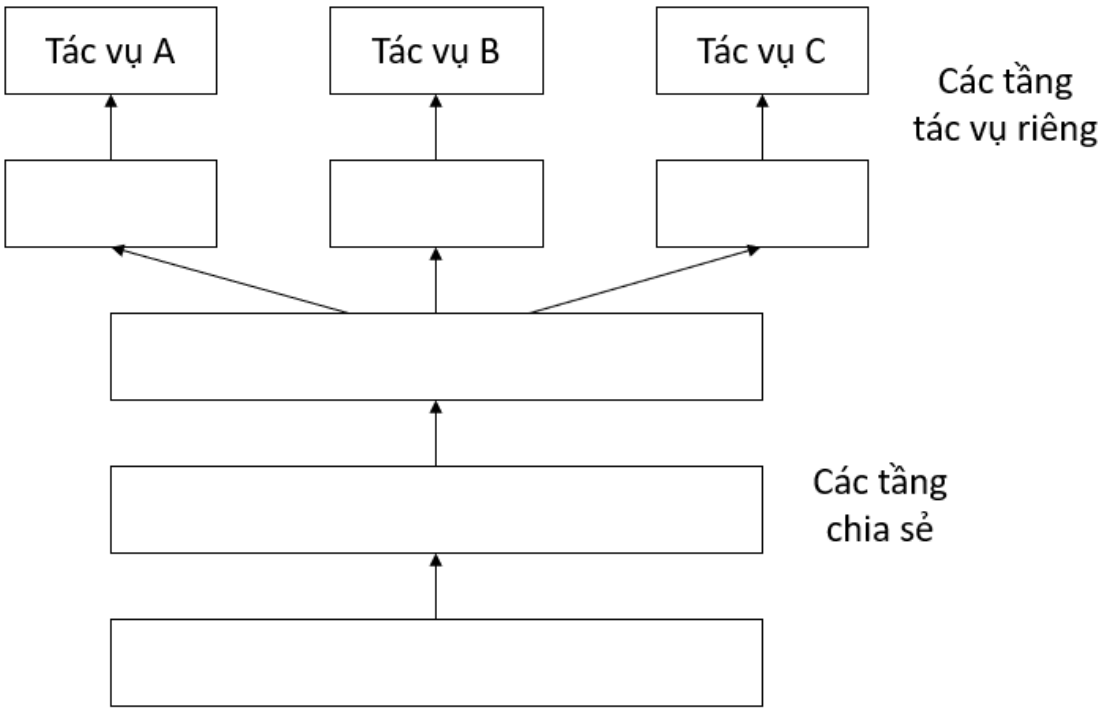
Theo Zang và cộng sự, MTL được định nghĩa như sau: "Với *m* nhiệm vụ học $\{T_i\}_m^{i=1}$ trong đó tất cả các nhiệm vụ hoặc một tập hợp con của chúng có liên quan với nhau, học đa tác vụ nhằm mục đích học *m* nhiệm vụ cùng nhau để cải thiện việc học mô hình cho từng nhiệm vụ T_i bằng cách sử dụng kiến thức có trong tất cả hoặc một số nhiệm vụ." [77]

MTL có rất nhiều cách sử dụng khác nhau, tuy nhiên trong học sâu thường sử dụng hai phương pháp là chia sẻ tham số cứng (Hard Parameter Sharing) và chia sẻ tham số mềm (Soft Parameter Sharing) [78].

2.4.1. Chia sẻ tham số cứng

Chia sẻ tham số cứng là một phương pháp được sử dụng rất nhiều trong mạng Nơ-ron. Phương pháp này được thực hiện bằng cách chia sẻ các tầng ẩn giữa tất cả các tác vụ, trong khi vẫn giữ một số tầng đầu ra dành riêng cho tác vụ, như có thể thấy trong Hình 2.6.

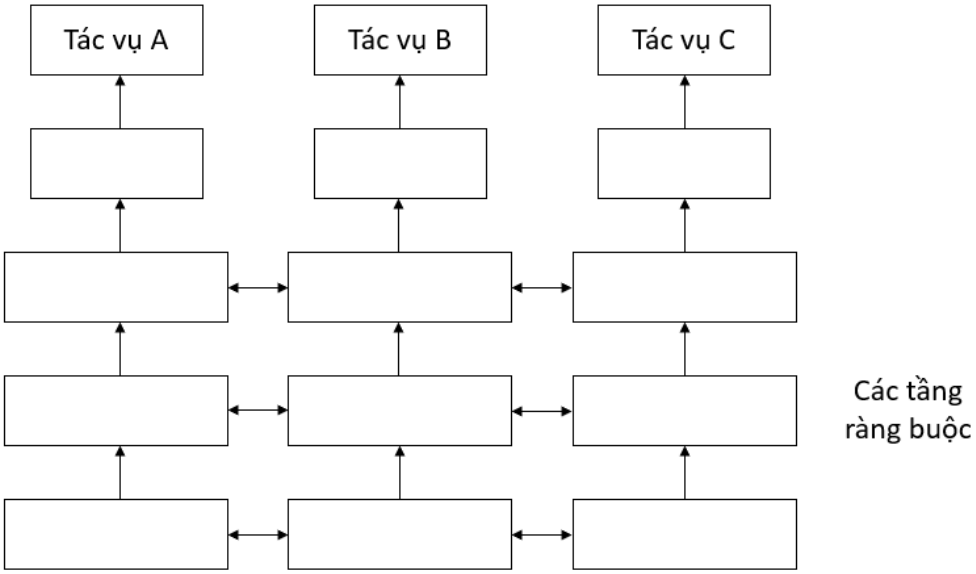
Chia sẻ tham số cứng giảm hiện tượng quá khớp (overfitting) rất tốt. Việc chia sẻ các tầng ẩn giữa các nhiệm vụ sẽ ép buộc mô hình phải học những biểu diễn tổng quát thích hợp ở trên nhiều nhiệm vụ, nhờ vậy mà khả năng overfitting vào một nhiệm vụ cụ thể nào đó sẽ giảm đi rất nhiều.



Hình 2.6: Mô hình phương pháp chia sẻ tham số cứng

2.4.2. Chia sẻ tham số mềm

Trong chia sẻ tham số mềm, mỗi tác vụ có mô hình riêng với các tham số riêng, tuy nhiên khoảng cách của các tham số giữa các nhiệm vụ sau đó sẽ được ràng buộc để khiến các tham số này có mức độ tương đồng cao giữa các nhiệm vụ, như trong Hình 2.7.



Hình 2.7: Mô hình phương pháp chia sẻ tham số mềm

2.4.3. Tác vụ phụ trợ

Trong nhiều trường hợp, mô hình chỉ quan tâm tới hiệu suất của một tác vụ cụ thể, tuy nhiên để tận dụng được những lợi ích mà MTL mang lại, có thể thêm vào một số tác vụ liên quan với mục đích là cải thiện thêm hiệu suất trên tác vụ chính. Các tác vụ này được gọi là các tác vụ phụ trợ (Auxiliary task). Việc sử dụng các tác vụ phụ trợ như thế nào là vấn đề đã được nghiên cứu từ lâu, tuy nhiên không có bằng chứng lý thuyết chắc chắn việc sử dụng các tác vụ phụ trợ nào sẽ đem lại sự cải thiện cho tác vụ chính.

Một trong những cân nhắc chính khi sử dụng học tập đa tác vụ với mạng nơ-ron học sâu là xác định tầng nào sẽ được chia sẻ. Trong XLNNTN, công việc gần đây tập trung vào việc tìm kiếm các hệ thống phân cấp nhiệm vụ để học đa tác vụ được tốt hơn.

Học đa tác vụ có hàm mất mát cuối cùng là tổng trọng số của các hàm mất mát thành phần

$$L_{final} = \sum_{i=1}^{T} \lambda_i \mathcal{L}_i \qquad (2.9)$$

trong đó T là số lượng tác vụ, λ_i là trọng số của mỗi hàm mất mát. Việc chọn các trọng số λ_i thích hợp cho mỗi tác vụ là rất quan trọng. Lựa chọn mặc định

là coi tất cả các nhiệm vụ như nhau bằng cách đặt $\lambda_1 = \cdots = \lambda_T = c$ với c là hằng số tùy ý.

Ngoài ra, cũng có thể thay đổi việc lấy mẫu các tác vụ. Với hai tác vụ $\mathcal{T}_1$ và $\mathcal{T}_2$ được lấy mẫu với xác suất lần lượt là $p_1$và p_2 nếu $p_1 = 2p_2$, xác định $\mathcal{T}_1$ với $\lambda_1 = 2\lambda_2$. Do đó, việc điều chỉnh tỷ lệ lấy mẫu của các tác vụ khác nhau có tác dụng tương tự như việc gán các trọng số khác nhau.

Việc tìm kiếm một tác vụ phụ trợ phần lớn dựa trên giả định rằng tác vụ phụ trợ phải liên quan đến nhiệm vụ chính theo một cách nào đó và nó sẽ hữu ích cho việc dự đoán tác vụ chính.

2.5. Kết luận chương 2

Chương 2 đã trình bày những kiến thức nền tảng về các kĩ thuật biểu diễn từ như Word2Vec, GloVe, BERT. Mô tả chi tiết về đặc điểm, kiến trúc của một số mô hình xử lý chuỗi như Transformer, GRU. Đồng thời, các mô hình gán nhãn như softmax, CRF cũng được giới thiệu. Đặc biệt, phương pháp chia sẻ tham số cứng, chia sẻ tham số mềm và tác vụ phụ trợ trong học đa tác vụ cũng được trình bày. Những mô hình được giới thiệu trong chương này sẽ là cơ sở để hướng tới xây dựng mô hình cho bài toán chuẩn hoá và nhận dạng thực thể định danh cho văn bản đầu ra ASR tiếng Việt được trình bày ở Chương 3, Chương 4.

CHƯƠNG 3: CHUẨN HÓA VĂN BẢN ĐẦU RA CỦA HỆ THỐNG NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT

Văn bản đầu ra của ASR là một văn bản thô, thường ở định dạng chữ thường, không có dấu câu, các kí tự số, ngày tháng, tiền tệ nhận dạng thành dạng chữ, tên riêng nước ngoài nhận dạng thành các chuỗi dài khó xử lý, đặc biệt là có sự xuất hiện của các lỗi như chèn, xóa, thay thế từ. Bên cạnh yêu cầu cải tiến hệ thống ASR để giảm thiểu lỗi từ thì chuẩn hóa văn bản đầu ra của hệ thống ASR bao gồm khôi phục dấu câu, chữ hoa cũng sẽ giúp văn bản dễ hiểu và cung cấp các thông tin quan trọng cho nhiều ứng dụng như tạo phụ đề hay sản xuất nội dung đa phương tiện. Trong phạm vi nghiên cứu luận án, nghiên cứu sinh cũng đặt giả thuyết rằng việc kết hợp khôi phục, dấu câu chữ hoa sẽ hỗ trợ cho mô hình NER đạt hiệu suất cao hơn. Chương 3 này sẽ trình bày về bài toán khôi phục dấu câu, chữ hoa trong văn bản đầu ra tiếng nói tiếng Việt, những khó khăn, hạn chế khi thực hiện nhiệm vụ này và từ đó đề xuất giải pháp, cách thức xây dựng dữ liệu, thiết lập mô hình và các kết quả thực nghiệm. Kết quả nghiên cứu về hai cách tiếp cận được công bố trong công trình [CT2], [CT3], [CT5].

3.1. Bài toán

Như đã trình bày trong mục 1.3, khôi phục dấu câu và chữ hoa đối với văn bản đầu ra của ASR là cần thiết, giúp văn bản dễ hiểu và được coi như bước tiền xử lý quan trọng để áp dụng cho các bài toán XLNNTN khác. Luận án xác định những vấn đề chính trong nghiên cứu và các giải pháp cụ thể khi xử lý bài toán này như sau:

Đầu vào: văn bản đầu ra của hệ thống ASR tiếng Việt

Đầu ra: văn bản được khôi phục dấu câu, chữ hoa

Phạm vi nghiên cứu:

- Về dữ liệu: Xây dựng bộ dữ liệu lớn phục vụ cho mục đích huấn luyện theo mô hình học sâu. Nghiên cứu khôi phục dấu câu, chữ hoa trên các đoạn văn bản tiếng nói dài như bản tin thời sự, bài phát biểu họp Quốc hội, ...

- Về dấu câu: Tập trung khôi phục ba loại dấu câu là dấu chấm, dấu phẩy, dấu chấm hỏi.

- Về chữ hoa: Phân biệt 2 nhãn chính là chữ thường, chữ hoa. Xử lý khôi phục viết hoa chữ cái đầu tiên của âm tiết. Không xử lý các nhãn như chữ hoa trộn lẫn (McDonald, TOUSlesJOURS, ...) hay chữ hoa toàn bộ (FPT, IBM, ...)

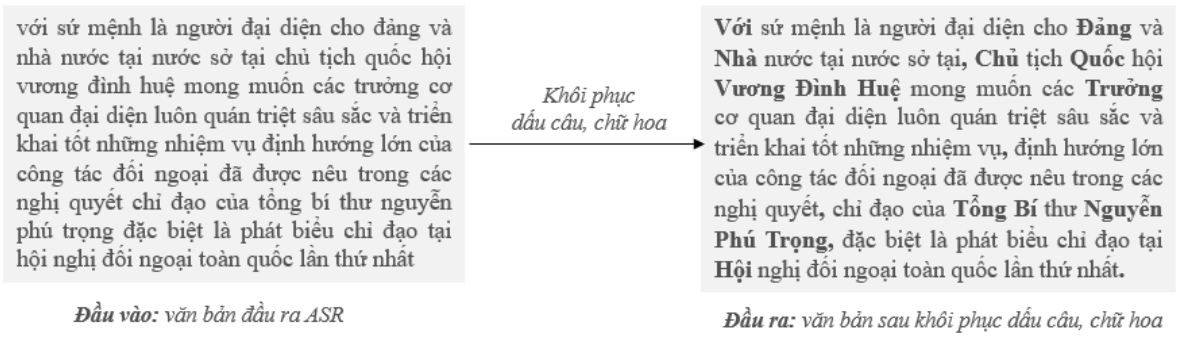
Hướng giải quyết:

- Đề xuất một cách phân đoạn chuỗi đầu vào và hợp nhất đầu ra, trong đó quan tâm tới ngữ cảnh của các từ xung quanh đoạn cắt.

- Thiết kế mô hình học sâu để kết hợp khôi phục dấu câu, chữ hoa.

- Xây dựng bộ dữ liệu phục vụ mục đích nghiên cứu từ các trang báo mạng chính thống của Việt Nam, với tỉ lệ lỗi từ trong văn bản là 0%.

Hình 3.1 dưới đây là một ví dụ minh hoạ[3] mô tả đầu vào, đầu ra của khôi phục dấu câu và chữ hoa đối với văn bản đầu ra ASR.



Hình 3.1: Minh hoạ đầu vào, đầu ra của khôi phục dấu câu, chữ hoa đối với văn bản đầu ra ASR

3.2. Xây dựng dữ liệu

3.2.1. Thu thập dữ liệu văn bản từ Internet

Để có nguồn dữ liệu văn bản đầu ra của ASR tiếng Việt đủ lớn cho nghiên cứu và tập trung chính cho việc huấn luyện mô hình CaPu, bộ dữ liệu $Text_{CaPu}$ được nghiên cứu thu thập từ các trang tin tức điện tử Việt Nam bao

[3] https://vietnamnet.vn/dai-su-truong-co-quan-dai-dien-phai-luon-neu-cao-tinh-than-vi-nhan-dan-phuc-vu-2120064.html

gồm *vietnamnet.vn*, *dantri.com.vn*, *vnexpress.net*. Đây là các tài liệu tin tức chính thống và sử dụng ngôn ngữ, ngữ pháp chuẩn.

3.2.2. Chuẩn hóa dữ liệu

Bộ dữ liệu $Text_{CaPu}$ được chuyển về chữ thường và loại bỏ các dấu câu để mô phỏng giống với đầu ra của ASR. Nghiên cứu cũng giữ nguyên các dữ liệu kiểu số, ngày tháng và không có lỗi từ (chèn, xóa, thay thế từ) trong văn bản. Bộ dữ liệu này cũng được chia thành bộ huấn luyện $Text_{CaPu\text{-}train}$, bộ đánh giá $Text_{CaPu\text{-}vl}$ và bộ kiểm tra $Text_{CaPu\text{-}test}$.

Một số lượng lớn các dấu câu có thể được xem xét cho văn bản đầu ra của ASR, bao gồm: dấu phẩy, dấu chấm, dấu chấm than, dấu chấm hỏi, dấu hai chấm, dấu chấm phẩy, dấu gạch ngang, dấu ngoặc đơn và dấu ngoặc kép. Tuy nhiên, hầu hết các dấu hiếm khi xảy ra trong văn bản. Do đó, hầu hết các nghiên cứu, kể cả với ngôn ngữ giàu tài nguyên, đều tập trung vào khôi phục các dấu cơ bản như dấu chấm, dấu phẩy và có thể thêm dấu chấm hỏi [18], [64]. Nghiên cứu cũng chỉ tập trung cho ba dấu câu cơ bản là dấu chấm, dấu phẩy, dấu chấm hỏi. Bảng 3.1 cung cấp thông tin số lượng nhãn cho từng loại dấu câu và viết hoa, viết thường trong bộ dữ liệu huấn luyện và bộ dữ liệu kiểm tra, bao gồm, *chữ hoa (U), chữ thường (L), không chứa dấu câu ($), dấu chấm (.), dấu phẩy (,) và dấu chấm hỏi (?)*

Bảng 3.1: Thông tin bộ dữ liệu

Nhãn	**Bộ dữ liệu huấn luyện(*)**	**Bộ dữ liệu kiểm tra(*)**
U	15.400	74
L	69.300	507
$	76.600	525
.	2.700	24
,	5.300	30
?	53	2.6

(*) Đơn vị: 1.000

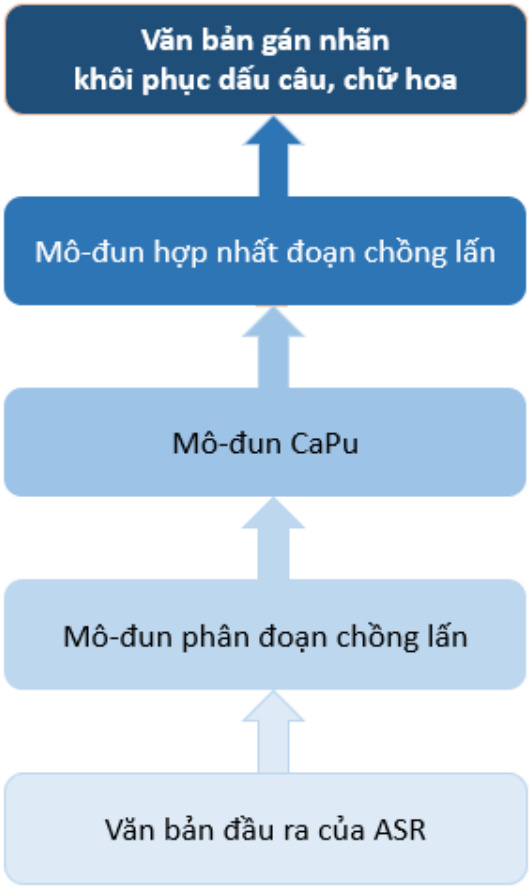
3.3. Kiến trúc mô hình

Hình 3.2 mô tả kiến trúc mô hình xử lý được tiến hành theo các bước sau:

(1) Bước một, văn bản đầu ra của ASR tiếng Việt sẽ được đưa qua mô-đun phân đoạn chồng lấn để cắt chuỗi đầu vào.

(2) Bước hai, mô hình khôi phục dấu câu, chữ hoa *(Recovering Capitalization and Punctuation - CaPu)* sẽ lấy các phân đoạn được cắt xử lý song song và tạo ra một danh sách nhãn dấu câu, chữ hoa đầu ra.

(3) Cuối cùng, sử dụng mô-đun hợp nhất đoạn chồng lấn để hợp nhất kết quả đầu ra được gán nhãn tương ứng với văn bản đầu vào.



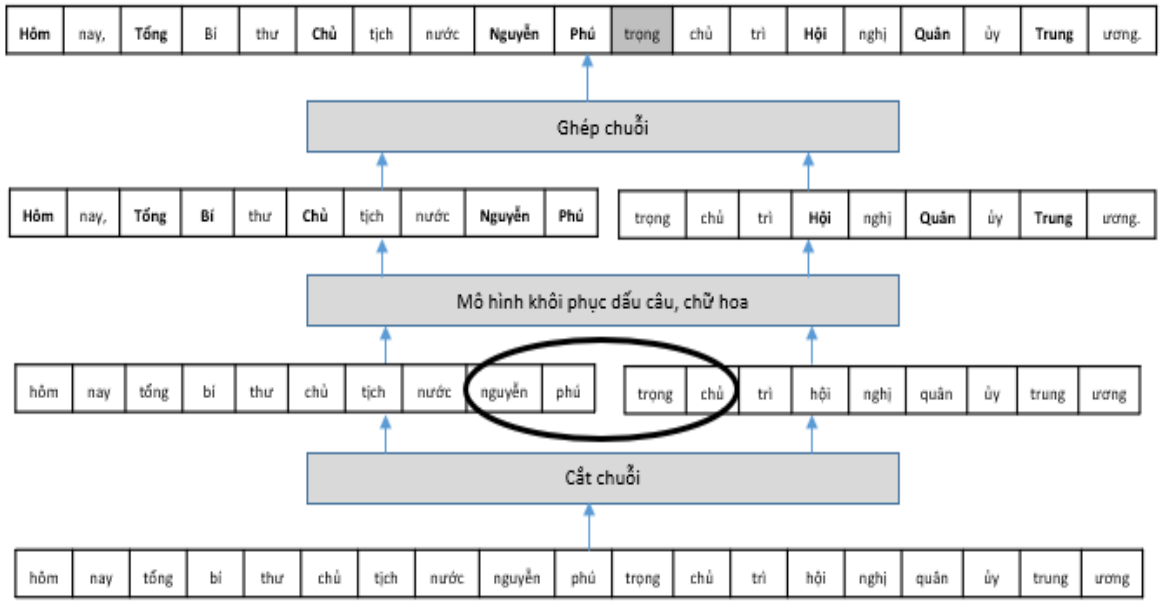
Hình 3.2: Kiến trúc mô hình

Trong đó, luận án đã ***đề xuất một kĩ thuật mới xử lý việc cắt chuỗi văn bản đầu vào, hợp nhất chuỗi đầu ra***, đồng thời, ***thiết kế một mô hình học sâu cho mục đích khôi phục dấu câu, chữ hoa***. Chi tiết mô hình và các đề xuất được trình bày chi tiết ở phần 3.3.1 và 3.3.2.

3.3.1. Đề xuất xử lý phân đoạn chuỗi đầu vào và hợp nhất chuỗi đầu ra

Đầu vào của mô hình CaPu là văn bản đầu ra của ASR. Văn bản này không có dấu câu nên thường là một chuỗi dài bất định, rất khó để các mô hình xử lý. Do đó, trước khi đưa vào mô hình, chuỗi đầu vào thường được cắt thành các đoạn có độ dài cố định, giúp cải thiện khả năng xử lý độc lập hoặc các phần song song.

Các nghiên cứu có liên quan đặc biệt quan tâm tới việc phân đoạn chuỗi câu đầu vào và thường xử lý theo hướng cắt ngẫu nhiên trong khoảng 20-30 từ [11], hay 20-50 từ [12]. Tuy nhiên, theo cách tiếp cận này, các từ xung quanh ranh giới của phần cắt không có đủ thông tin ngữ cảnh nên dự đoán thường thiếu chính xác.

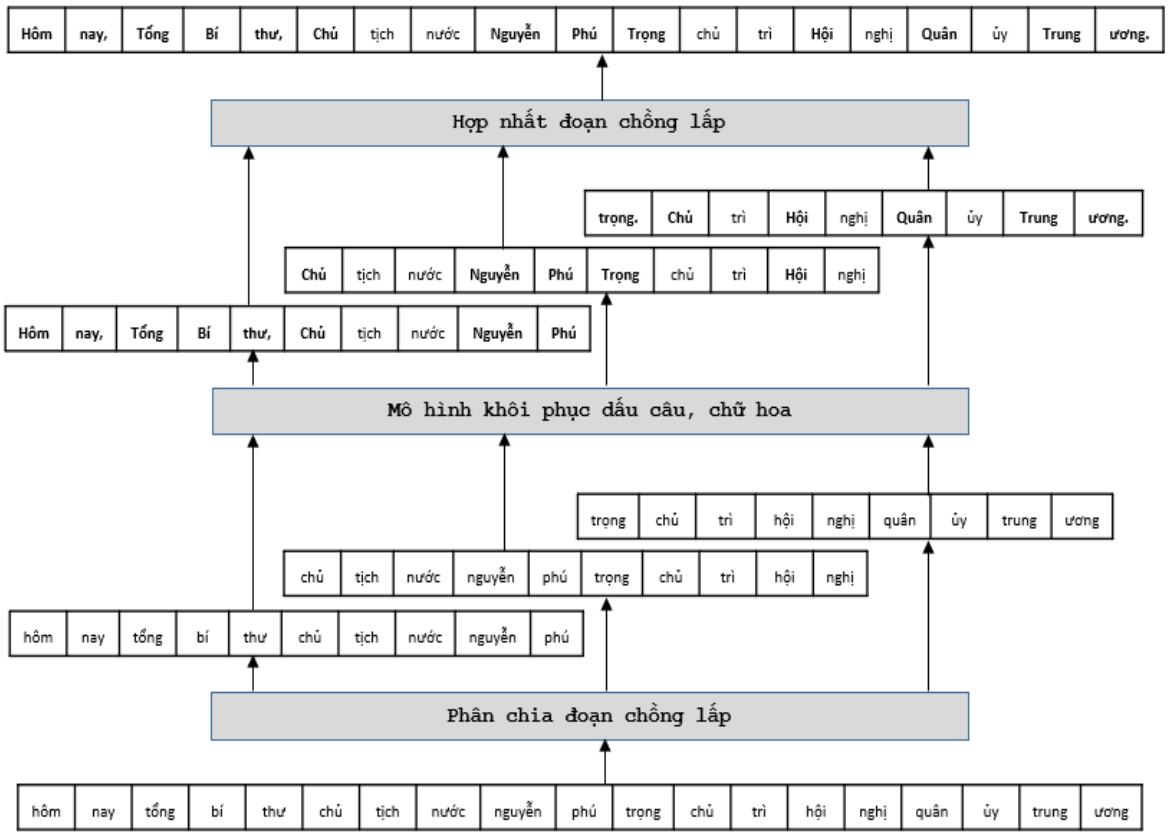


Hình 3.3: Mô hình xử lý chuỗi đầu vào, đầu ra thông thường

Ví dụ minh họa trong Hình 3.3 cho thấy từ *"trọng"* trong đoạn cắt thứ hai không đủ ngữ cảnh xung quanh để khôi phục đúng chữ hoa.

Để khắc phục hạn chế đó, nghiên cứu đã đề xuất một kỹ thuật mới nhằm xử lý cắt, ghép chuỗi bằng cách cắt có chồng lấn với ***ý tưởng chính là nhằm đảm bảo các đoạn cắt thu được có đủ ngữ cảnh của các từ để mô hình CaPu dự đoán tốt nhất***. Sau khi xử lý các đoạn cắt có chồng lấn, thực hiện hợp nhất các đoạn này thành chuỗi đầu ra của chuỗi ban đầu.

Hình 3.4 mô tả chi tiết về kiến trúc này, bao gồm ba thành phần: phân chia đoạn chồng lấn, mô hình CaPu, và hợp nhất các đoạn chồng lấn. Có thể thấy, câu đầu vào được chia thành ba đoạn, các đoạn được xếp chồng. Sau khi qua mô hình CaPu, các đoạn được nhận dạng, trong đó cụm từ *"Nguyễn Phú Trọng"* ở giữa đoạn thứ hai có nhiều ngữ cảnh xung quanh nên nhận dạng chính xác hơn các từ *"phú"* ở đoạn 1 và *"trọng"* ở đoạn 3. Cuối cùng, các đoạn sẽ được hợp nhất chồng lấn để ra câu sau khôi phục.

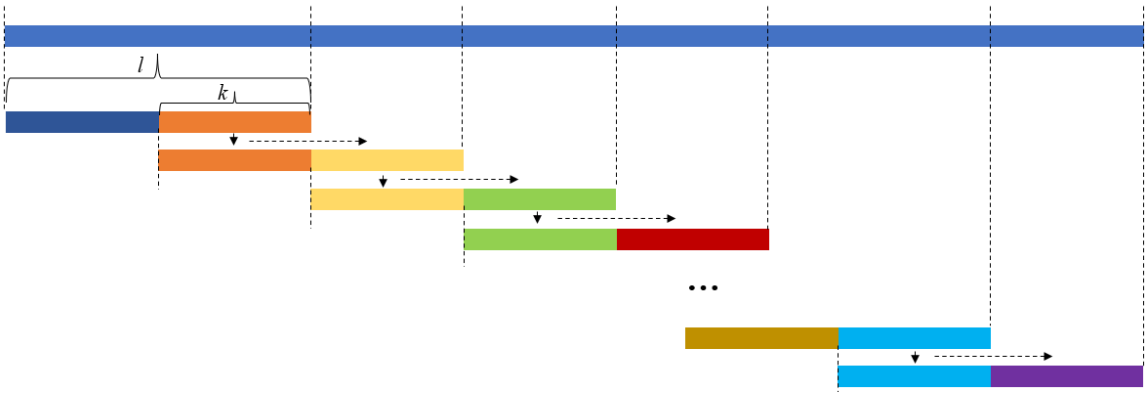


Hình 3.4: Đề xuất mô hình phân chia/hợp nhất đoạn chồng lấn

Phần tiếp theo sẽ trình bày cụ thể phương pháp phân đoạn chồng lấn và cách thức hợp nhất kết quả đầu ra.

3.3.1.1.Phân đoạn chồng lấn

Đối với mô-đun phân đoạn chồng lấn, hướng giải quyết được đề xuất là chia nhỏ chuỗi đầu vào thành các đoạn có kích thước cố định, với phần chồng lấn chiếm một nửa độ dài đoạn cắt. Hình 3.5 dưới đây mô tả cách phân đoạn chồng lấn.



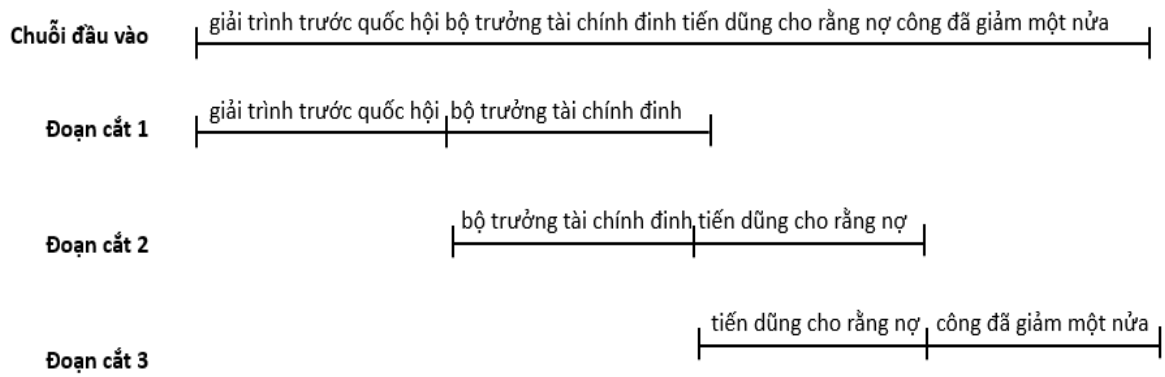
Hình 3.5: Mô tả phân chia đoạn chồng lấn

Có thể mô tả hình thức cách phân đoạn chồng lấn như sau:

Độ dài đoạn cắt được chọn là một số chẵn các từ. Gọi l là độ dài đoạn cắt, k là độ dài đoạn chồng lấn, khi đó ta có $l=2k$.

Mỗi chuỗi từ đầu vào S chứa n từ kí hiệu là w_1, w_2,..., w_n sẽ được cắt thành $\lceil n/l \rceil + \lceil (n-k)/l \rceil$ đoạn chồng lấn, trong đó, đoạn cắt thứ i là chuỗi con các từ $[w_{(i-1)k+1}, \ldots, w_{(i+1)k}]$. Trong nghiên cứu đã khảo sát các giá trị của l, k và bằng thực nghiệm đã lựa chọn các giá trị này cho phù hợp.

Hình 3.6 minh hoạ bằng một ví dụ cụ thể:



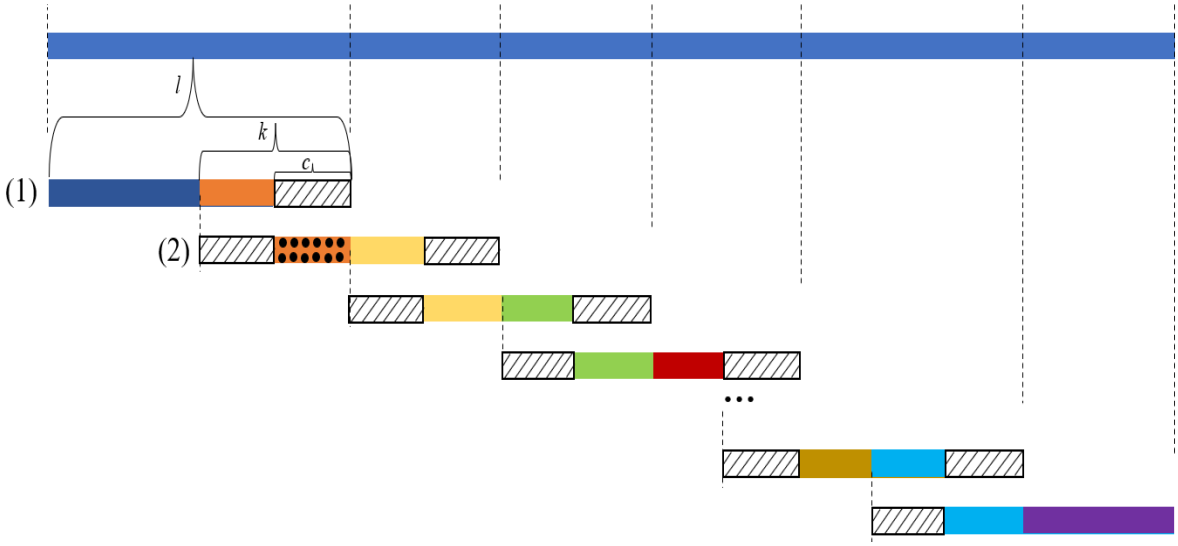
Hình 3.6: Ví dụ phân chia đoạn chồng lấn với l = 10 và k = 5

Sau khi xử lý, cần hợp nhất các đoạn như trong mục sau.

3.3.1.2.Hợp nhất đoạn chồng lấn

Vì câu đầu vào được phân chia thành các đoạn chồng lấn, do đó, với vấn đề hợp nhất các đoạn chồng lấn, cần phải xác định được những từ nào sẽ được bỏ đi và từ nào sẽ được giữ trong phần hợp nhất của câu cuối cùng.

Gọi c là độ dài đoạn sẽ giữ lại hay loại bỏ trong các đoạn chồng lấn. Để đơn giản cho tính toán, lấy $c = \lfloor k/2 \rfloor$. Theo quan sát, các từ cuối của đoạn chồng lần thứ nhất và các từ đầu tiên trong đoạn chồng lấn thứ hai (các từ xung quanh đoạn cắt) sẽ không có nhiều ngữ cảnh. Do vậy, thuật toán sẽ loại bỏ đoạn c thuộc cuối đoạn chồng lấn (1) (phần gạch chéo) và giữ lại đoạn c ở đoạn chồng lấn (2) (phấn chấm). Theo đó, các từ còn lại của đầu đoạn chồng lấn (1) được giữ lại và các từ còn lại ở đầu đoạn chồng lấn (2) sẽ bị loại bỏ. Điều này đảm bảo cho các từ ở phần chống lấn được giữ lại luôn ở giữa các đoạn, sẽ có nhiều ngữ cảnh giúp cho việc khôi phục được chính xác hơn. Các đoạn loại bỏ và giữ lại của các phần chồng lấn sẽ được lặp lại cho các phân đoạn chồng lấn tiếp theo.

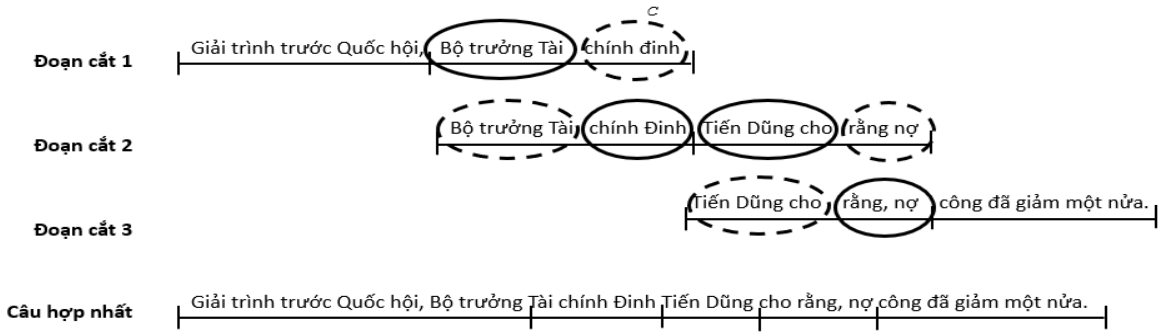


Hình 3.7: Mô tả cách ghép nối

Như vậy, theo Hình 3.7, các đoạn loại bỏ sẽ được gạch chéo. Phần hợp nhất sau ghép nối được mô tả như sau.

$$[w_1, \dots w_{2k-c}] + \sum_{i=2}^{n-1} [w_{(i-1)k+c}, \dots w_{ik+c}] + [w_{n-2k+c}, \dots w_n] \qquad (3.1)$$

Hình 3.8 mô tả các đoạn c trong khoanh nét đứt sẽ bị loại bỏ và các đoạn c trong khoanh nét liền sẽ được giữ lại. Điều này giúp cho các từ ở các phần chồng lấn được giữ lại có nhiều ngữ cảnh để mô hình dự đoán đạt hiệu suất cao hơn. Do đó, câu ghép nối cuối, từ "Đinh" và dấu phẩy được nhận dạng chính xác trong câu hợp nhất cuối cùng.



Hình 3.8: Hợp nhất các đoạn chồng chéo dựa trên tham số c

3.3.2. Thiết kế mô hình học sâu cho mục đích khôi phục dấu câu, chữ hoa

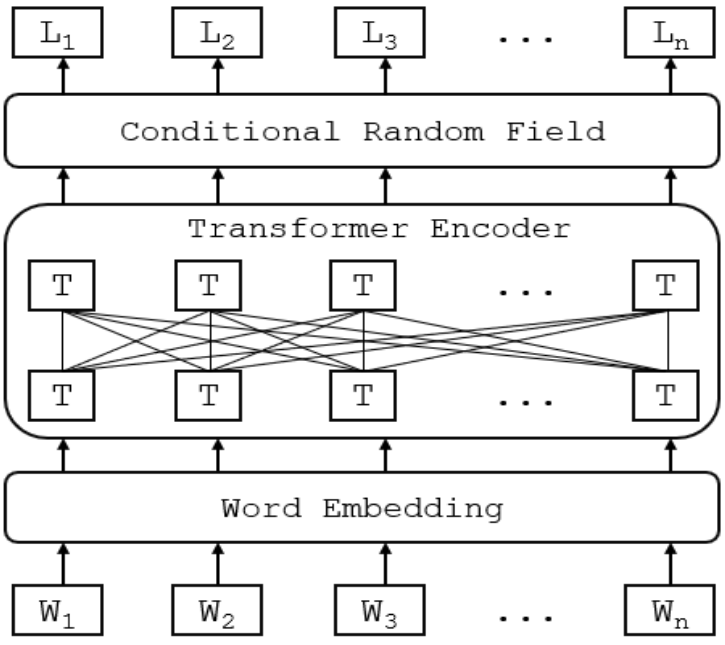
Tổng quan các nghiên cứu về khôi phục dấu câu, chữ hoa cho văn bản đầu ra của ASR đã được trình bày chi tiết trong mục 1.3 của Chương 1. Có thể thấy, các hướng nghiên cứu trước đây thường tập trung xử lý một nhiệm vụ cụ thể, khôi phục dấu câu, hoặc khôi phục chữ hoa. Điều này khiến cho việc cải thiện mô hình ASR không đạt được hiệu quả tối ưu. Một số các nghiên cứu gần đây đã xử lý kết hợp khôi phục dấu câu, chữ hoa trong một mô hình, tuy nhiên tiếp cận theo hướng khôi phục chữ hoa trước, sau đó khôi phục dấu câu và ngược lại. Điều này ảnh hưởng đến kết quả của mô-đun sau và rất khó để xác định nên thực hiện mô-đun nào trước, mô-đun nào sau [15]. Chính vì vậy, luận án tích hợp đồng thời khôi phục dấu câu và chữ hoa trong cùng một mô-đun.

Mặt khác, các mô hình học sâu gần đây đã chứng minh được tính hiệu quả trong nhiều tác vụ XLNNTN trong đó có xử lý khôi phục dấu câu, chữ hoa [18], [79]. Các mô hình truyền thống trong XLNNTN chủ yếu sử dụng kiến trúc tuần tự chuỗi tới chuỗi (Sequence-to-Sequence) dựa trên các mạng nơ-ron hồi quy (RNN). Nhược điểm của các mạng RNN là tốc độ xử lý chậm do phải xử lý câu đầu vào một cách tuần tự, đồng thời cũng hạn chế trong việc biểu diễn sự phụ thuộc xa giữa các từ trong một câu. Mô hình Transformer [32] có thể giải quyết gần như triệt để các vấn đề nói trên. Transformer không xử lý các phần tử trong một chuỗi một cách tuần tự. Nếu dữ liệu đầu vào là một câu ngôn ngữ tự nhiên, Transformer không cần phải xử lý phần đầu câu trước rồi mới tới phần cuối câu. Do tính năng này, Transformer có thể tận dụng khả năng tính toán song song của GPU và giảm thời gian xử lý đáng kể.

Để gán nhãn cho chuỗi đầu ra của Transformer, có thể sử dụng một lớp Softmax hoặc một lớp CRF. Tuy nhiên, CRF thường được sử dụng trong các bài toán gán nhãn chuỗi vì khả năng mô hình hóa các ràng buộc giữa các nhãn liên tiếp trong chuỗi. CRF có thể giải quyết được nhược điểm sai lệch nhãn do các nhãn độc lập với nhau của mô hình Markov ẩn. Trong khi đó, Softmax thường được sử dụng trong các bài toán phân loại do khả năng tính xác suất cho mỗi lớp. CRFs là một lớp các phương pháp mô hình hóa thống kê thường được áp dụng trong nhận dạng mẫu và học máy, và được sử dụng để dự đoán cấu trúc [27], [80]. Trong mô hình CRF, các nút chứa dữ liệu đầu vào và các nút chứa dữ liệu đầu ra được kết nối trực tiếp với nhau, trái với kiến trúc của LSTM hoặc BiLSTM trong đó các đầu vào và đầu ra được kết nối gián tiếp qua các ô nhớ. CRF có thể được sử dụng để gán nhãn tên riêng với đầu vào là các đặc trưng của một từ được rút trích thủ công.

Luận án thiết kế sử dụng mô hình Transformer Encoder kết hợp với CRF để khôi phục dấu câu và chữ hoa cho văn bản đầu ra của ASR tiếng Việt.

Hình 3.9 giới thiệu mô hình CaPu đề xuất cho bài toán khôi phục dấu câu và chữ hoa cho văn bản đầu ra ASR tiếng Việt gồm các thành phần: bộ biểu diễn véc-tơ từ (Word Embedding), Transformer Encoder và CRF.



Hình 3.9: Mô hình CaPu đề xuất cho văn bản đầu ra của ASR tiếng Việt

Mô hình đề xuất sử dụng cấp độ từ để đưa vào lớp biểu diễn véc-tơ từ (Word Embedding). Đây là bước ánh xạ các từ sang dạng véc-tơ để mô tả tất cả các từ trong từ điển sang một không gian véc-tơ biểu diễn ngôn ngữ hay cũng có thể hiểu là một hình thức mã hóa từ. Ý tưởng chính là đưa các từ qua một tầng biểu diễn véc-tơ từ trước khi được đưa vào các tầng khác của mạng. Điều này giúp các mô hình học sâu có thể xử lý các từ ngữ trong văn bản trên không gian véc-tơ biểu diễn ngôn ngữ, đồng thời giảm bớt ảnh hưởng về chiều đối với các mô hình ngôn ngữ [81].

Trong nghiên cứu của luận án, mô hình CaPu đề xuất không sử dụng toàn bộ kiến trúc Transformer (mục 2.1.2) mà chỉ sử dụng bộ mã hóa trong mô hình này giúp hạn chế không chỉ về mặt thời gian, mà còn giải quyết một vấn đề trong quá trình mã hóa là số từ mã hóa nhiều hơn số từ đầu vào.

Đối với lớp đầu ra, mô hình sẽ gán nhãn bao gồm '*U*' để biểu thị chữ hoa (Uppercase) '*L*' để biểu thị chữ thường (Lowercase) và nhãn '*$*' (không chứa dấu câu) ; '.' (dấu chấm) ; ',' (dấu phẩy) ; '*?*' (dấu chấm hỏi) để thêm dấu câu cho từ đầu vào. Như vậy, mỗi từ sẽ được gán với một trong 8 nhãn sau: {*U$* ; *L$* ; *U.* ; *L.* ; *U,* ; *L,* ; *U?* ; *L?*}.

Khối mã hóa của Transformer có thể sử dụng toán tử Softmax để xác định xem nhãn nào được sử dụng cho từ đầu vào. Tuy nhiên, Softmax không quan tâm đến thứ tự của nhãn, do đó có thể xuất ra hai nhãn *U.* và *L.* đứng cạnh nhau, hầu như vô nghĩa trong mọi trường hợp. Do đó, để xử lý thứ tự của nhãn đầu ra, luận án sẽ sử dụng trường ngẫu nhiên có điều kiện (Conditional Random Field - CRF) là một loại mô hình đồ thị được sử dụng cho các tác vụ dự đoán cấu trúc, chẳng hạn như gán nhãn chuỗi hoặc phân đoạn ảnh. Chúng thường được sử dụng trong xử lý ngôn ngữ tự nhiên, thị giác máy tính và các lĩnh vực khác nơi dữ liệu có cấu trúc là quan trọng.

Nghiên cứu cũng sử dụng định dạng văn bản được gãn nhãn (b) để huấn luyện mô hình, ví dụ được đưa ra trong Hình 3.10. Định dạng gán nhãn (b) có thể suy diễn nhanh hơn văn bản thông thường (a) vì số nhãn cố định

nên khi mã hóa vốn từ vựng sẽ được thu hẹp. Tuy nhiên, nó có hạn chế là không có nhiều thông tin ngữ cảnh của các từ xung quanh.

Hình 3.10: Mô tả đầu ra nhận dạng dạng văn bản và dạng nhãn

3.4. Kết quả thực nghiệm

3.4.1. Thiết lập mô hình

Nghiên cứu đã tiến hành thực nghiệm trên các mô hình LSTM, Transformer và mô hình mới đề xuất Transformer Encoder - CRF. Các mô hình được xây dựng dựa trên thư viện Fairseq [82]. LSTM và Transformer là mô hình mã hóa-giải mã. Mỗi mô hình có hai tầng mã hóa, hai tầng giải mã và có kích thước lớp ẩn giống nhau là 1024. Một điểm khác biệt của Transformer so với LSTM là Transformer có số đỉnh chú ý là 4.

Để so sánh trong cùng điều kiện, Transformer Encoder - CRF cũng có số tầng mã hóa là 4, mỗi tầng có 4 đỉnh chú ý và có cùng kích thước ẩn là 1024. Kích thước biểu diễn véc-tơ cả ba mô hình là 256. Bảng 3.2 cho thấy số lượng các tham số của ba mô hình, qua đó cho thấy số tham số của mô hình đề xuất tương đương với mô hình LSTM và chỉ bằng 1/5 số tham số của mô hình Transformer.

Thực nghiệm được huấn luyện trên GPU NVIDIA 2080Ti. Kho ngữ liệu bao gồm 85 triệu từ. Kích thước đoạn ngẫu nhiên là 4 đến 22 từ.

Bảng 3.2: Số lượng tham số của các mô hình

Mô hình	Văn bản mã hóa(*)	Văn bản thường(*)
LSTM	6.500	11.300
Transformer	3.700	42.000
Transformer Encoder-CRF	7.400	-

(*) Đơn vị: 1.000

Để huấn luyện mô hình, nghiên cứu sử dụng thuật toán tối ưu Adam [83] với hàm mất mát là giá trị âm của logarit hàm hợp lí (negative log-likelihood). Độ đo để đánh giá là độ đo F1. Các tham số huấn luyện được cho ở bảng 3.3 dưới đây:

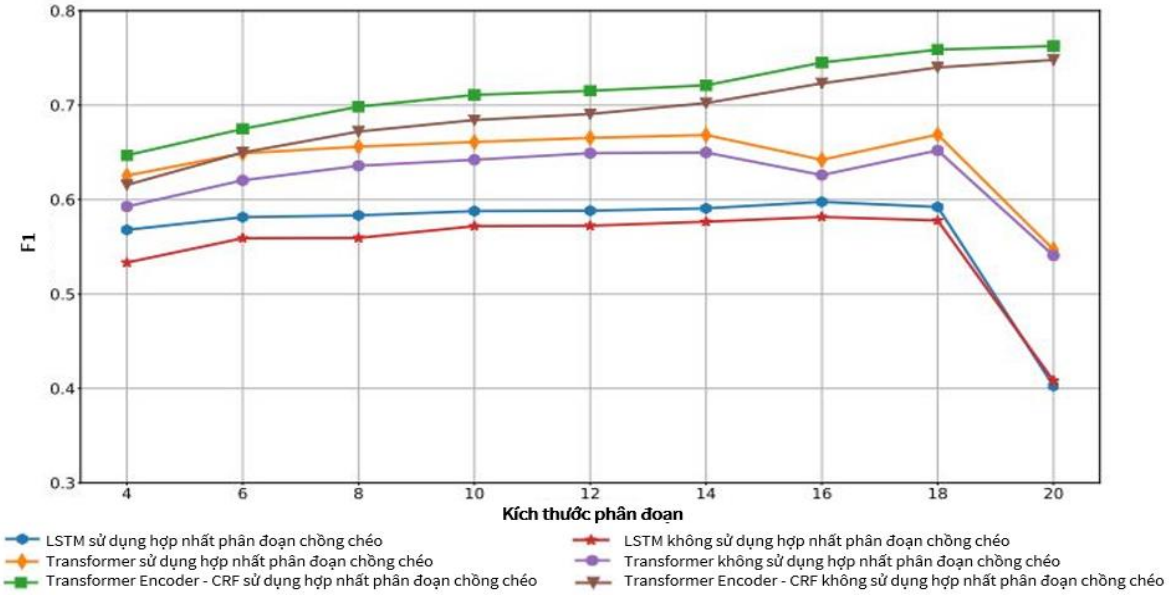
Bảng 3.3: Các tham số huấn luyện mô hình

Tham số	Giá trị
Tốc độ học	5×10^{-4}
Tỉ lệ dropout	3×10^{-1}
Weight decay	10^{-4}
Số lần lặp tối đa	15×10^{4}
Số lần lặp khởi động	4×10^{3}
Tốc độ học khởi động	10^{-7}
Tốc độ học tối thiểu	10^{-9}

3.4.2. Đánh giá về sử dụng hợp nhất đoạn chồng lấn

Hình 3.11 dưới đây hiển thị biểu đồ so sánh với kết quả của các mô hình LSTM, Transformer, Transformer Encoder - CRF với các kích thước phân

đoạn khác nhau, từ 4 đến 20 từ và trong các trường hợp sử dụng hoặc không sử dụng hợp nhất đoạn chồng lấn.



Hình 3.11: Kết quả của các mô hình sử dụng và không sử dụng hợp nhất đoạn chồng lấn

Có thể nhận thấy rằng, các mô hình sử dụng hợp nhất đoạn chồng lấn luôn cho kết quả tốt hơn. Đặc biệt, ở mô hình đề xuất là Transformer Encoder - CRF, kết quả sử dụng hợp nhất có kết quả cao nhất là 0.88. Kết quả xác nhận giả thuyết của nghiên cứu rằng việc bổ sung thêm ngữ cảnh bằng cách phân đoạn, hợp nhất các đoạn chồng lấn sẽ giúp cải thiện mô hình.

Nghiên cứu trình bày kết quả của mô hình đề xuất Transformer Encoder - CRF khi áp dụng hoặc không áp dụng hợp nhất đoạn chồng lấn và cũng chỉ thống kê trong các nhãn (‘*U*’ ‘.’ ‘,’ ‘*?*’), bỏ qua các nhãn (‘*L*’ ‘*$*’), vì số lượng chính xác nhiều, nên không cần thiết để so sánh hiệu quả.

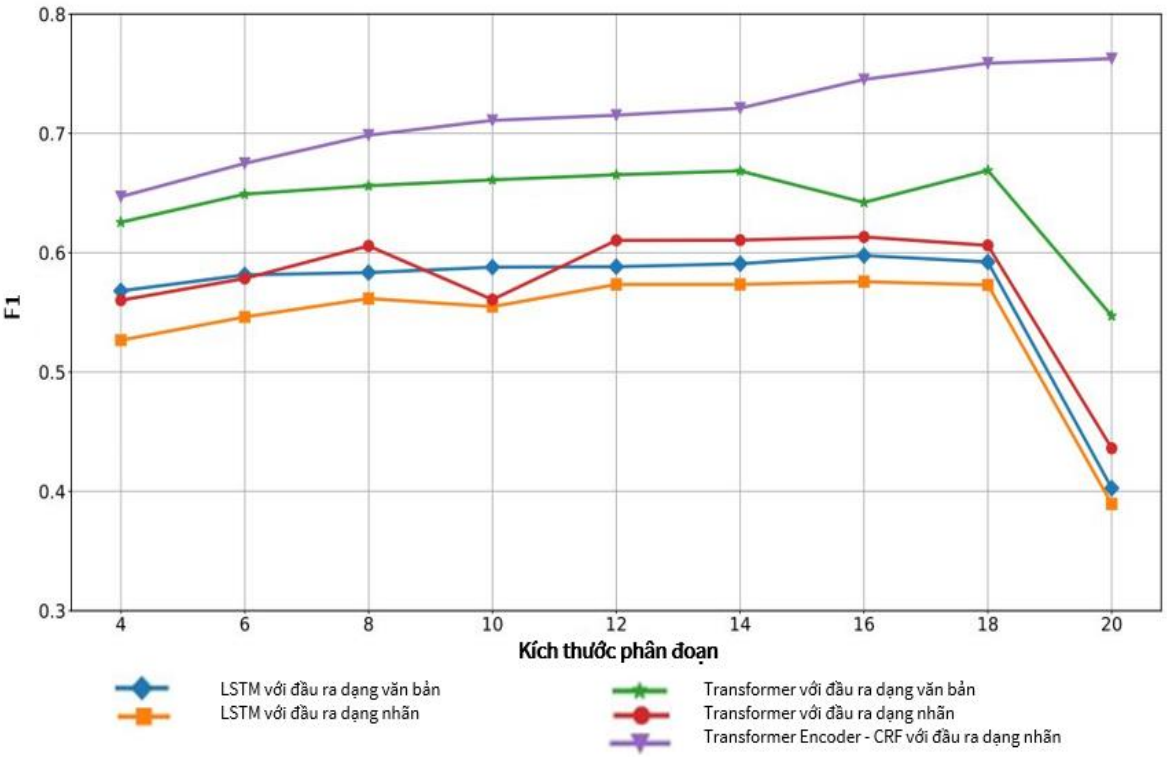
Bảng 3.4 trình bày sự so sánh giữa mô hình Transformer Encoder - CRF khi áp dụng và không áp dụng hợp nhất chồng lấn cho thấy sự vượt trội của phương pháp hợp nhất đoạn chồng lấn so với không sử dụng khi điểm F1 trên tất cả các lớp được cải thiện đáng kể từ 0.01 đến 0.05.

Kết quả cho thấy rằng các từ ở đoạn giữa phần xếp chồng lấn cung cấp cho mô hình nhiều thông tin dự đoán hơn và quá trình hợp nhất có thể chọn phần thích hợp của khu vực xếp chồng này.

Bảng 3.4: So sánh kết quả mô hình Transformer Encoder - CRF khi áp dụng và không áp dụng hợp nhất chồng lấn

Mô hình	Nhãn	Precision	Recall	F1
Transformer Encoder-CRF áp dụng hợp nhất chồng lấn	U	0.90	0.86	**0.88**
	.	0.71	0.57	**0.63**
	,	0.66	0.53	**0.59**
	?	0.75	0.52	**0.62**
Transformer Encoder-CRF không áp dụng hợp nhất chồng lấn	U	0.89	0.85	0.87
	.	0.69	0.54	0.61
	,	0.65	0.50	0.57
	?	0.74	0.47	0.58

3.4.3. Đánh giá đầu ra văn bản mã hóa và văn bản thô



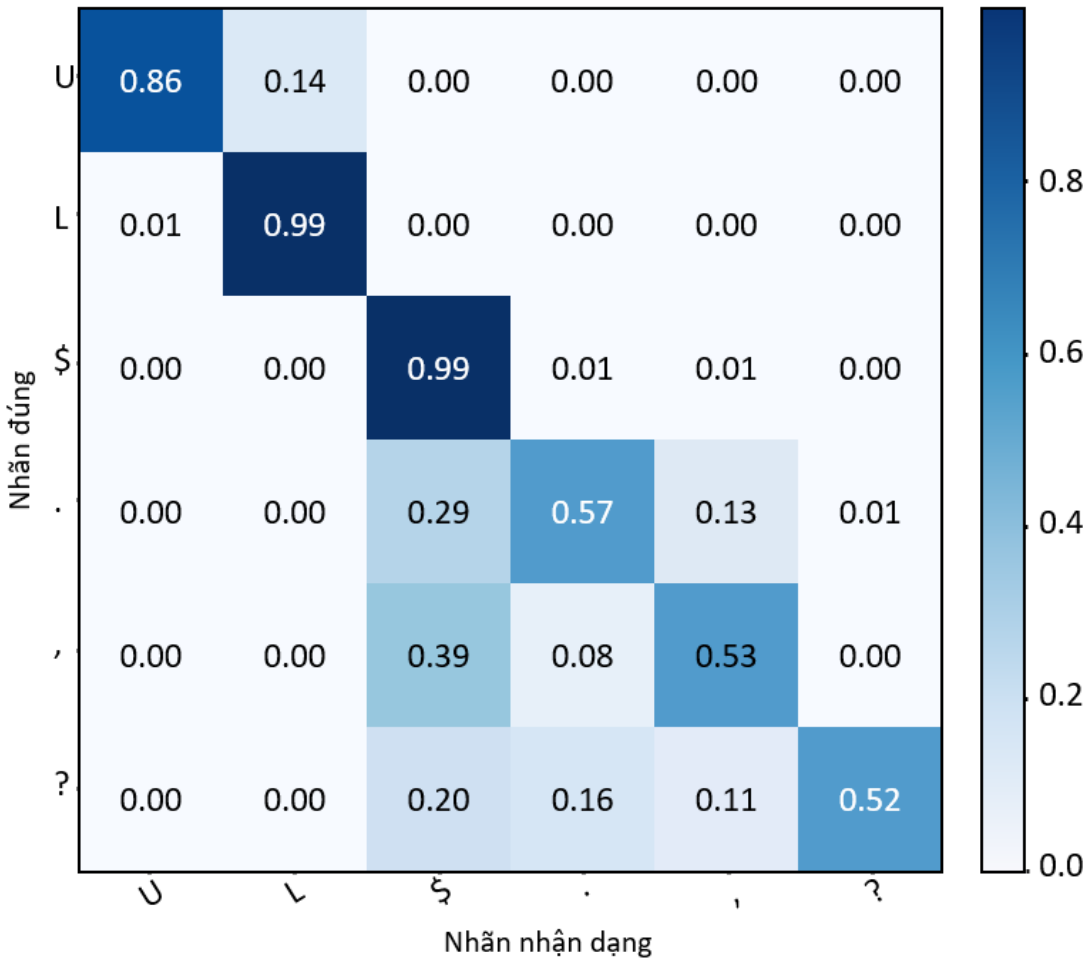
Hình 3.12: Kết quả của các mô hình với đầu ra là dạng văn bản hoặc dạng nhãn

Kết quả cho các mô hình sử dụng đầu ra gán nhãn và văn bản thông thường được so sánh trong Hình 3.12, trong đó, mô hình LSTM và mô hình

Transformer với văn bản thông thường có kết quả tốt hơn so với sử dụng đầu ra gán nhãn.

Nghiên cứu cho thấy, mô hình sử dụng đầu ra gán nhãn được giảm kích thước và suy luận nhanh hơn. Vì vậy, nghiên cứu chỉ tập trung đánh giá mô hình đề xuất - Transformer Encoder - CRF với đầu ra này. Biểu đồ cho thấy mô hình đề xuất cho kết quả tốt nhất.

Đồng thời, ma trận lỗi *(Confusion matrix)* trong Hình 3.13 cũng cho thấy phần trăm dự đoán đúng/sai lệch các nhãn dấu câu, chữ hoa cho mô hình đề xuất Transformer Encoder - CRF.



Hình 3.13: Ma trận lỗi cho mô hình Transformer Encoder - CRF

Ma trận lỗi chứng minh khả năng khôi phục đúng chữ thường, chữ hoa và không dấu rất cao (0.86-0.99), sau đó giảm dần với các dấu chấm, dấu phẩy và dấu chấm hỏi.

3.4.4. Đánh giá tốc độ xử lý

Kết quả so sánh thời gian thực thi của ba mô hình có đầu ra được gán nhãn và văn bản chuẩn hóa thông thường được hiển thị trong Bảng 3.5s với 2080 ti (GPU), batch_size 128. Với đầu ra gán nhãn, các mô hình có thời gian xử lý nhanh hơn, thậm chí còn cho thấy hiệu suất vượt trội khi nó được sử dụng với mô hình được đề xuất Transformer Encoder - CRF.

Bảng 3.5: So sánh tốc độ xử lý (tokens/second)

Đầu ra	**Transformer**	**LSTM**	**Transformer Encoder - CRF**
Dạng gán nhãn	263*s* → 2209*t/s*	217*s* → 2678*t/s*	90*s* → 6457*t/s*
Dạng văn bản	355*s* → 1637*t/s*	230*s* → 2526*t/s*	-

3.5. Kết luận Chương 3

Chương 3 đã xây dựng mô hình kết hợp Transformer Encoder và CRF cho mục đích khôi phục viết hoa và dấu câu với văn bản đầu ra của ASR tiếng Việt. Có thể nói, đóng góp chính của nghiên cứu là đề xuất giải pháp phân chia và hợp nhất đoạn chồng lấn trong chuỗi đầu vào, đầu ra. Cách tiếp cận này nhằm mục đích cải thiện khả năng trích xuất thông tin theo ngữ cảnh và hiệu suất làm việc với văn bản dài. Sau khi đánh giá, phương pháp đề xuất thể hiện hiệu suất vượt trội cả về tốc độ và độ chính xác. Trong cùng điều kiện với mô hình Transformer, thì Transformer Encoder - CRF cung cấp một số lượng tham số nhỏ hơn đáng kể, từ đó giúp làm tăng tốc độ xử lý. Phương pháp hợp nhất đoạn chồng lấn cho thấy hiệu suất tốt hơn việc không sử dụng hợp nhất từ 0.01 đến 0.05 của độ đo F1. Ngoài ra, việc sử dụng văn bản đầu ra được gán nhãn cũng cải thiện hiệu suất của hệ thống.

Tuy nhiên, mô hình vẫn còn tồn tại một số hạn chế cần cải tiến trong thời gian tới bao gồm việc khôi phục trên văn bản có chứa lỗi từ của đầu ra hệ

thống ASR, đồng thời, thử nghiệm trên bộ dữ liệu của ngôn ngữ khác để có đối sánh giữa các phương pháp.

Trong những chương tiếp theo, nghiên cứu đề xuất tích hợp mô-đun CaPu với mô hình NER cho văn bản đầu ra của ASR tiếng Việt và giả thuyết rằng việc kết hợp như vậy sẽ giúp cải thiện hiệu suất mô hình NER.

CHƯƠNG 4: NHẬN DẠNG THỰC THỂ ĐỊNH DANH CHO VĂN BẢN ĐẦU RA CỦA HỆ THỐNG NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT

Nhận dạng thực thể định danh (NER) là một nhiệm vụ quan trọng làm tiền đề cho nhiều lĩnh vực XLNNTN như truy xuất thông tin, tóm tắt văn bản, dịch máy, ... Tuy nhiên, bên cạnh những thành tựu đạt được từ NER cho các văn bản viết, vấn đề NER với văn bản đầu ra của ASR vẫn còn gặp nhiều khó khăn do phải đối mặt với các lỗi phiên âm, từ ngoài từ điển hay thiếu các đặc trưng quan trọng của thực thể định danh. Các nghiên cứu cho vấn đề này mới chủ yếu tập trung cho các ngôn ngữ giàu tài nguyên như tiếng Anh, tiếng Pháp, tiếng Trung Quốc. Việc nghiên cứu về NER cho ASR tiếng Việt - được coi là ngôn ngữ hạn chế tài nguyên, với nhiều đặc trưng riêng là cần thiết và có ý nghĩa trong các ứng dụng thực tiễn. Trong nội dung Chương 3 này sẽ trình bày chi tiết về bài toán NER và đề xuất mô hình, xây dựng dữ liệu, đưa ra kết quả thực nghiệm nhằm đánh giá, so sánh các giải pháp cho NER của văn bản đầu ra của ASR tiếng Việt theo cách tiếp cận đường ống truyền thống và cách tiếp cận E2E.

Cách tiếp cận đường ống truyền thống dựa trên giả thuyết rằng việc kết hợp một mô hình khôi phục dấu câu và chữ hoa như mô hình CaPu sẽ cung cấp thông tin hữu ích làm đầu vào giúp mô hình NER đạt hiệu suất cao hơn. Cách tiếp cận E2E là một quy trình phức hợp từ đầu đến cuối, giúp hệ thống hoạt động thuận tiện hơn, tránh được những lỗi lan truyền qua các bước giải các bài toán trung gian. Giải pháp E2E cho bài toán NER đề xuất mô hình giải quyết đồng thời cả hai bài toán khôi phục dấu câu, chữ hoa và nhận dạng thực thể định danh. Kết quả nghiên cứu về hai cách tiếp cận được công bố trong công trình [CT4], [CT6].

4.1. Bài toán

Đầu vào: Văn bản đầu ra của ASR tiếng Việt.

Đầu ra: Gán nhãn thực thể định danh theo hướng tiếp cận đường ống và E2E.

Phạm vi nghiên cứu:

- Về dữ liệu: Văn bản dài, từ vựng lớn. Hệ thống ASR phục vụ đánh giá có WER là 4.85%.

- Về thực thể định danh: Nhận dạng ba loại thực thể chính là tên người, tên tổ chức và tên địa điểm.

Hướng nghiên cứu:

- Xây dựng bộ dữ liệu phù hợp cho mục đích huấn luyện và đánh giá mô hình.

- Đối với cách tiếp cận đường ống, nghiên cứu đề xuất kết hợp mô hình CaPu vào hệ thống với mục đích nâng cao hiệu suất mô hình NER. Cách tiếp cận E2E, sử dụng tiền huấn luyện mô-đun CaPu cho mô hình.

- Đề xuất kiến trúc NER sử dụng các mô hình học sâu.

4.2. Tổng quan dữ liệu

4.2.1. Bộ dữ liệu huấn luyện

Bộ dữ liệu thứ nhất, $Text_{CaPu}$, là một bộ dữ liệu lớn bao gồm các văn bản tin tức được lấy từ các trang báo điện tử của Việt Nam. Tập văn bản này được xóa định dạng (bỏ dấu câu, chuyển chữ hoa thành chữ thường) và gán nhãn dấu câu, chữ hoa phục vụ cho mục đích huấn luyện mô hình chuẩn hoá văn bản đầu ra của hệ thống ASR.

Bộ dữ liệu thứ hai, $Text_{ViBERT}$, là bộ dữ liệu huấn luyện mô hình ViBERT thu thập từ nhiều miền trên Internet bao gồm tin tức, luật, giải trí, Wikipedia,...

Bộ dữ liệu thứ ba, $Text_{VLSP}$, là bộ dữ liệu văn bản đã gán nhãn NER của VLSP 2018. Tập văn bản chuẩn này được sử dụng để huấn luyện mô hình NER theo cách tiếp cận đường ống.

Bộ dữ liệu thứ tư, $Text_{VLSP\text{-}TTS\text{-}ASR}$, là bộ dữ liệu để huấn luyện mô hình NER theo tiếp cận E2E. Đầu tiên, dữ liệu tiếng nói được tổng hợp từ văn bản huấn luyện của bộ dữ liệu NER VLSP 2018 sử dụng hệ thống TTS của Google. Sau đó dữ liệu tiếng nói này được đưa qua hệ thống ASR của VAIS để thu được văn bản đầu ra ASR.

4.2.2. Bộ dữ liệu kiểm tra

Cả hai cách tiếp cận đường ống và E2E đều sử dụng một bộ dữ liệu thu âm bởi bốn giọng đọc trong môi trường khác nhau từ bộ dữ liệu kiểm tra NER của VLSP 2018 với 26 giờ âm thanh. Sau đó, bộ dữ liệu âm thanh này được đưa qua hệ thống ASR của VAIS (với WER bằng 4.85%) để nhận được bộ dữ liệu văn bản đầu ra của ASR, $Text_{VLSP\text{-}Audio\text{-}ASR}$ để phục vụ cho mục đích đánh giá các mô hình đề xuất.

Đồng thời, bộ dữ liệu kiểm tra VLSP chuẩn $Text_{VLSP\text{-}test}$ hay bộ dữ liệu VLSP được xóa định dạng $Text_{VLSP\text{-}UnCaPu}$, cũng được sử dụng để đánh giá và so sánh mô hình trong các điều kiện đầu vào khác nhau.

Chi tiết xây dựng dữ liệu cho từng cách tiếp cận sẽ được trình bày cụ thể trong mục 4.3.3 và mục 4.4.2.

4.3. Nhận dạng thực thể định danh theo hướng tiếp cận Đường ống

Mục 1.4.5.1, Chương 1 của luận án đã trình bày tổng quan về các phương pháp NER cho tiếng nói theo hướng tiếp cận Đường ống. Quá trình nhận dạng thực thể định danh từ tiếng nói thực hiện tuần tự qua các bước: đầu tiên hệ thống ASR tạo ra các văn bản, sau đó, hệ thống NER gắn thẻ các thực thể định danh từ văn bản đầu ra của ASR. Có thể nói, hướng tiếp cận này được ưu tiên lựa chọn nghiên cứu bởi tính đơn giản của hệ thống bằng cách chia nhỏ để xử lý từng mô-đun con, dễ xử lý từng phần và không đòi hỏi hệ thống tính toán lớn, đặc biệt, đối với các phương pháp học sâu.

4.3.1. Đề xuất mô hình

4.3.1.1. Tổng quan hệ thống

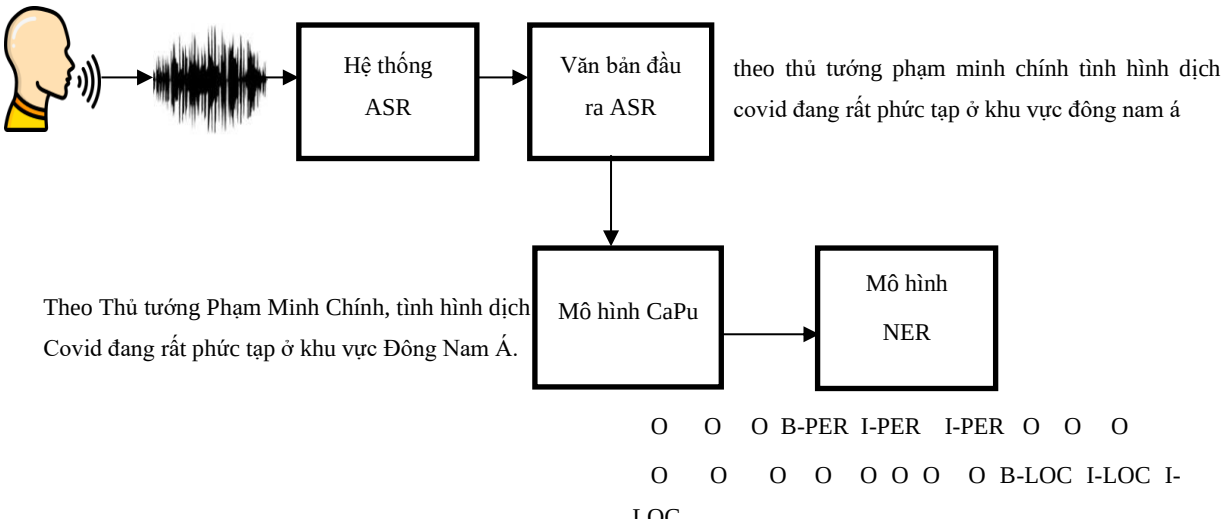
Đề xuất kiến trúc tổng quát hệ thống NER trong văn bản đầu ra của ASR tiếng Việt theo hướng tiếp cận đường ống được mô tả trong Hình 4.1.

Hệ thống đường ống thực hiện theo trình tự các bước sau:

(1) Hệ thống ASR sẽ chuyển tín hiệu tiếng nói sang dạng văn bản.

(2) Tiếp theo, qua mô hình CaPu, văn bản đầu ra của ASR sẽ được khôi phục dấu câu, chữ hoa.

(3) Cuối cùng, từ mô hình CaPu, thông tin của các thực thể được gán nhãn bằng cách sử dụng mô hình NER.



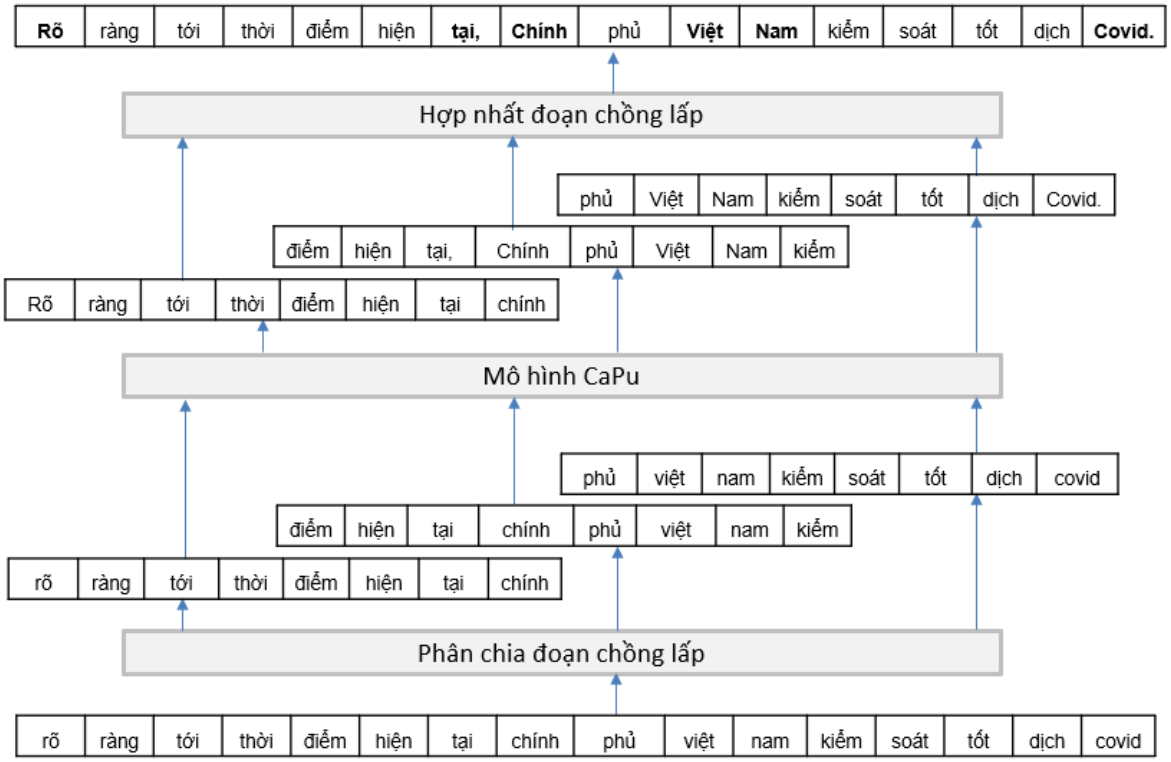
Hình 4.1: Mô tả kiến trúc NER tổng quát theo cách tiếp cận đường ống

Phần tiếp theo sẽ trình bày chi tiết về hai mô hình CaPu và NER.

4.3.1.2. Mô hình khôi phục dấu câu và chữ hoa

Dấu câu và chữ hoa đóng một vai trò quan trọng trong việc cung cấp ý nghĩa của câu, là một trong những thông tin không thể thiếu cần cung cấp trong mô hình NER, tuy nhiên, thông tin này thường bị bỏ qua trong hệ thống ASR. Năm 2020, Mayhew và các cộng sự [84] đã thử nghiệm tiền huấn luyện bộ nhận dạng chữ hoa trong văn bản trước khi kết hợp với mô hình NER đối với dữ liệu tiếng Anh và cho thấy mô hình khôi phục chữ hoa có thể cung cấp thông tin bổ sung giúp hệ thống cải thiện ít nhất 0.3 điểm F1. Chính vì vậy, trong nghiên cứu này, nghiên cứu sinh cũng đặt giả thuyết mô hình CaPu sẽ hỗ trợ tăng hiệu suất mô hình NER cho văn bản đầu ra ASR tiếng Việt.

Hình 4.2 biểu diễn mô hình CaPu được đề xuất nhằm khôi phục dấu câu và chữ hoa cho văn bản đầu ra của ASR. Mô hình đề xuất và các kết quả thực nghiệm đã được trình bày chi tiết trong Chương 3 của luận án, đồng thời được công bố trong các công trình (CT2), (CT3), (CT5) của nghiên cứu sinh và các cộng sự.



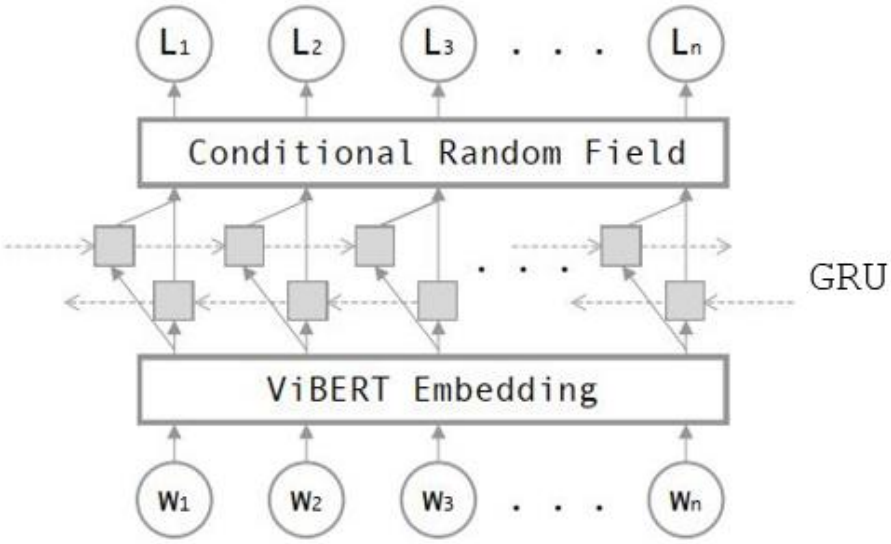
Hình 4.2: Mô hình CaPu cho văn bản đầu ra của ASR

4.3.1.3. Thiết kế mô hình học sâu cho nhận dạng thực thể định danh

Các mô hình học sâu cho XLNNTN cần một lượng dữ liệu rất lớn để có thể cho ra kết quả tốt. Vì vậy, vấn đề đặt ra: làm thế nào để tận dụng được nguồn dữ liệu vô cùng lớn có sẵn để giải quyết bài toán? Đây là tiền đề cho kỹ thuật mới là học chuyển giao (Transfer Learning) ra đời. Với học chuyển giao, các mô hình "chung" nhất với tập dữ liệu khổng lồ trên Internet được huấn luyện trước và có thể được "tinh chỉnh" cho các bài toán cụ thể. Nhờ có kỹ thuật này mà kết quả cho các bài toán được cải thiện rõ rệt, không chỉ trong XLNNTN mà còn trong các lĩnh vực khác như thị giác máy, ... BERT là một trong những mô hình được sử dụng nhiều trong học chuyển giao bởi có thể áp dụng trong nhiều bài toán khác nhau.

Có thể nói, BERT và các biến thể mô hình đang trở thành xu hướng và sẽ định hướng các thuật toán XLNNTN trong tương lai. Điều này thúc đẩy sử dụng mô hình BERT trong nghiên cứu luận án để xây dựng bộ biểu diễn ngôn ngữ cho tiếng Việt.

Cụ thể, nghiên cứu này đã đề xuất sử dụng kiến trúc RoBERTa [34] (một công thức cải tiến cho huấn luyện mô hình BERT) và huấn luyện trên kho ngữ liệu tiếng Việt để tạo ra một mô hình ngôn ngữ được huấn luyện trước (Pre-trained language models). Do giới hạn về năng lực tính toán, mô hình huấn luyện đã giảm số lượng lớp ẩn, số đỉnh chú ý và kích thước từ biểu diễn véc-tơ từ mô hình kiến trúc cơ sở RoBERTa và được đặt tên là ViBERT. Hình 4.3 mô tả thiết kế mô hình NER, trong đó, ViBERT được sử dụng để nhúng câu đầu vào. Các mô hình GRU hai chiều và các lớp CRF được gắn vào đầu ViBERT để phân loại nhãn thực thể của mỗi từ đầu vào.



Hình 4.3: Đề xuất mô hình NER

4.3.2. Thiết lập mô hình

Nghiên cứu của chúng tôi đã giảm kích thước của mô hình $RoBERTa_{base}$ triển khai trong fairseq [82] để tạo ra ViBERT. Mô hình này chứa 4 tầng mã hóa tương ứng với với 4 tầng trong $RoBERTa_{base}$. Số lượng đỉnh tự chú ý cũng giảm từ 12 xuống 4 so với mô hình $RoBERTa_{base}$. Mỗi mẫu huấn luyện chứa tối đa 512 token.

ViBERT được huấn luyện bằng cách sử dụng kích thước mỗi batch là 512 và tốc độ học lớn nhất là 0.0003 với 3.000 bước cập nhật khởi động. Tổng các bước cập nhật là 800.000. Thực nghiệm sử dụng hai GPU Nvidia

2080Ti (12GB cho từng GPU) trong 5 tuần. Bảng 4.1 mô tả các tham số cấu trúc và huấn luyện mô hình ViBERT.

Bảng 4.1: Tham số cấu trúc và huấn luyện mô hình ViBERT

Tham số	**Giá trị**
Tầng mã hóa	4
Đỉnh tự chú ý	4
Kích thước batch	512
Tốc độ học lớn nhất	3×10^{-4}
Số bước cập nhật khởi động	3×10^{3}
Tổng các bước cập nhật	8×10^{5}
Thuật toán tối ưu	Adam

Cài đặt mô hình NER sử dụng ViBERT để biểu diễn từ và có 4 lớp GRU hai chiều với kích thước ẩn của ô GRU là 512. CRF được sử dụng trong lớp đầu ra để tạo ra 7 nhãn (B-X, I-X, O trong đó X trong bộ {ORG, PER, LOC}). Giống như [34], mô hình này cũng được tối ưu hóa bằng cách sử dụng Adam, kích thước batch là 64 và quá trình huấn luyện hội tụ sau 30 bước lặp.

4.3.3. Chi tiết xây dựng dữ liệu

Vấn đề dữ liệu NER cho tiếng nói gặp nhiều khó khăn, đến thời điểm hiện tại, chưa có một tập dữ liệu chuẩn cho nhiệm vụ NER của tiếng nói tiếng Việt. Có hai cách tiếp cận mà luận án xem xét là tạo bộ dữ liệu tiếng nói từ tập dữ liệu NER hoặc ngược lại. Việc ghi lại âm thanh từ việc đọc văn bản đã có gán nhãn NER dễ dàng hơn nhiều so với việc gắn thẻ NER trên bản ghi âm ASR. Nghiên cứu xây dựng các bộ dữ liệu riêng cho từng mục đích:

(1) Bộ dữ liệu huấn luyện mô hình CaPu

Trong mô hình CaPu, các dấu câu được xử lý bao gồm (“.” , “,” , “?”), nghiên cứu chia dữ liệu thành các phân đoạn có phạm vi ngẫu nhiên từ 4 đến 20 từ. Bộ dữ liệu $Text_{CaPu\text{-}train}$ được thu thập tự động từ các trang tin tức điện tử chính thống của Việt Nam bao gồm vietnamnet.vn, dantri.com.vn,

vnexpress.net, ... và được mã hóa như mô tả trong Chương 2. Tổng số dữ liệu sử dụng để huấn luyện mô hình này là hơn 300 triệu mẫu.

(2) Bộ dữ liệu huấn luyện mô hình ViBERT

Mô hình ViBERT cần một kho dữ liệu lớn để huấn luyện, nghiên cứu đã sử dụng bộ dữ liệu $Text_{ViBERT}$ với 50GB văn bản, khoảng 7.7 tỷ từ thu thập dữ liệu từ nhiều miền trên Internet bao gồm tin tức, luật, giải trí, Wikipedia, ... Do sự đa dạng của các kiểu gõ mã hóa tiếng Việt trên Internet, nghiên cứu cũng sử dụng thư viện Visen[4] để thống nhất phương pháp mã hóa. Mô hình ViBERT được huấn luyện bằng cách sử dụng kho dữ liệu xử lý theo thuật toán Byte-Pair-Encoding (BPE). BPE được thiết lập để xuất ra kích thước từ vựng 50 nghìn từ.

Năm 2016, phương pháp BPE được đề xuất [85], có khả năng tách từ theo mức nhỏ hơn từ và lớn hơn ký tự được gọi là từ con (subword). Phương pháp BPE sẽ thống kê tần suất xuất hiện của các từ con và tìm cách gộp chúng lại nếu tần suất xuất hiện là lớn nhất. Tiếp tục quá trình gộp từ con cho tới khi không tồn tại các từ con để gộp nữa, sẽ thu được tập các từ con cho toàn bộ văn bản mà mọi từ đều có thể biểu diễn được thông qua tập từ con này. Phương pháp đã được áp dụng ở hầu hết các phương pháp XLNNTN hiện đại như BERT, RoBERTa, DistilBERT, XLMNet. Kết quả áp dụng tokenize theo phương pháp mới đã cải thiện được độ chính xác trên nhiều tác vụ dịch máy, phân loại văn bản, dự báo câu tiếp theo, hỏi đáp, dự báo mối quan hệ văn bản.

(3) Bộ dữ liệu huấn luyện mô hình NER

Bộ dữ liệu văn bản chuẩn $Text_{VLSP}$ là bộ dữ liệu NER cho tiếng Việt đã được xây dựng trong VLSP. VLSP 2018 [44] là tập dữ liệu tốt nhất hiện nay để đánh giá hệ thống NER cho tiếng Việt.

---

[4] https://github.com/nguyenvulebinh/visen

Bộ dữ liệu NER của VLSP cung cấp một tập dữ liệu đáng tin cậy để huấn luyện và đánh giá hiệu suất của các mô hình NER tiếng Việt. Các tài nguyên này có sẵn cho mục đích nghiên cứu thông qua trang web VLSP vlsp.org.vn/resources [CT1]. Thống kê chi tiết của bộ dữ liệu này được thể hiện ở Bảng 4.2 dưới đây:

Bảng 4.2: Thống kê bộ dữ liệu NER của VLSP 2018

Tập dữ liệu	**PER**	**ORG**	**LOC**	**MICS**	**Tổng**
Huấn luyện	9002	10931	12304	1454	**33.691**
Đánh giá	356	524	576	46	**1.501**
Kiểm thử	1225	1383	1546	116	**4.270**

(4) *Bộ dữ liệu kiểm thử mô hình NER*

Sử dụng bộ dữ liệu $Text_{VLSP\text{-}Audio\text{-}ASR}$ thu âm từ bộ dữ liệu NER của VLSP 2018 và đưa qua hệ thống ASR, cụ thể: Dữ liệu văn bản đầu ra của ASR dùng để kiểm thử mô hình NER chính là đầu ra thu được qua bộ ASR của VAIS với dữ liệu đầu vào âm thanh tiếng nói được ghi âm của tập dữ liệu kiểm tra của dữ liệu NER VLSP 2018. Dữ liệu tiếng nói được tạo bởi bốn người đọc trong các môi trường khác nhau tạo ra tổng cộng hơn 26 giờ âm thanh. Các thực thể cần được trích xuất là tên người (PER), tên tổ chức (ORG) và tên địa điểm (LOC). Dữ liệu gốc ở định dạng XML và chứa các thực thể ở các cấp lồng nhau. Để dễ dàng so sánh với các kết quả nghiên cứu công bố trong [44], dữ liệu đã được chuyển đổi sang định dạng CoNLL NER và chỉ phát hiện thực thể ở cấp độ đầu tiên.

4.3.4. Độ đo đánh giá

Trong hệ thống NER thông thường, đầu vào là văn bản và đầu ra là nhãn cho mỗi từ trong văn bản đó. Tuy nhiên, trong hệ thống đề xuất, mô-đun NER sẽ trích xuất các thực thể từ văn bản đầu ra của hệ thống ASR. Vấn đề đặt ra là văn bản đầu ra của ASR có thể có một số loại lỗi như chèn, xóa, thay thế từ khiến độ dài của nhãn đầu ra giả thuyết có thể khác với

nhãn thật sự ban đầu, làm cho nó không thể tính điểm F1 như trong hệ thống NER thông thường.

Để bỏ qua sự không khớp này, văn bản đầu ra của ASR sẽ được so sánh với văn bản tham chiếu trong bộ dữ liệu NER. Nếu văn bản đầu ra ASR đúng **(T -** True**)**, thì nhãn thực thể giả thuyết vẫn được giữ nguyên. Nếu loại lỗi là xóa **(D -** Delete**)** hoặc thay thế **(S -** Substitute**)**, thì đầu ra giả thuyết của từ này sẽ trở thành nhãn **O**. Còn nếu loại lỗi là chèn **(I -** Insert**)**, thì nhãn sẽ bị xóa. Bằng cách đó sẽ làm cho kích thước của các nhãn tham chiếu bằng với kích thước của các nhãn giả thuyết.

Ví dụ, trong Hình 4.4, đầu ra ASR có một vài lỗi. Sau khi căn chỉnh, độ chính xác (P) là 100% và độ thu hồi (R) là 33,33% vì chỉ một trong ba nhãn là đúng F1 = 2*(P * R)/(P + R) = 50%

Reference text	Ông	Tedros	thuộc	WHO	đánh	giá	cao	Việt	Nam
Reference tag	O	B-PER	O	B-ORG	O	O	O	B-LOC	I-LOC
ASR + CAPU output	Ông	Dros	thuộc	_	đánh	giá	cao	tại	Việt	Nam
Hypothesis tag	O	B-PER	O	O	O	O	O	O	B-LOC	I-LOC
ASR error	T	S	T	D	T	T	T	I	T	T
Alignment text	Ông	_	thuộc	_	đánh	giá	cao	Việt	Nam	
Alignment tag	O	O	O	O	O	O	O	B-LOC	I-LOC	

Hình 4.4: Ví dụ về đầu ra của mô hình

4.3.5. Kết quả đánh giá

Trong mô hình NER, nghiên cứu kết hợp ViBERT với lớp GRU và lớp CRF cho thấy hiệu quả khi tạo ra kết quả F1 là 0.9018, cao hơn đáng kể khi so sánh với kết quả đã công bố trước đó (Bảng 4.3). Đây là kết quả được đánh giá trực tiếp bằng cách sử dụng bộ dữ liệu $Text_{VLSP\text{-}test}$ của NER VLSP 2018, văn bản có phân biệt chữ hoa chữ thường và đầy đủ dấu câu.

Bảng 4.3: Đánh giá các mô hình NER dựa trên bộ dữ liệu NER của VLSP 2018

Mô hình	**F1**
Vi Tokenizer + Bidirectional Inference [44]	0.8878
VNER [86]	0.7752
Multi layers LSTM [44]	0.8380
CRF/MEM + BS [44]	0.8408
ViBERT+GRU+CRF (mô hình đề xuất)	**0.9018**

Với tỷ lệ lỗi từ của hệ thống ASR là 4.85%, Bảng 4.4 cho thấy rằng nếu văn bản đầu ra của ASR được đưa trực tiếp vào mô hình NER, hiệu quả nhận dạng thực thể sẽ giảm từ 0.9018 xuống 0.6389. Tầm quan trọng của chữ hoa và dấu câu cũng được quan sát thấy trong thử nghiệm chạy mô hình NER trên văn bản bỏ dấu dấu câu và chữ hoa. Trong trường hợp này, điểm F1 giảm từ 0.9018 xuống 0.7535.

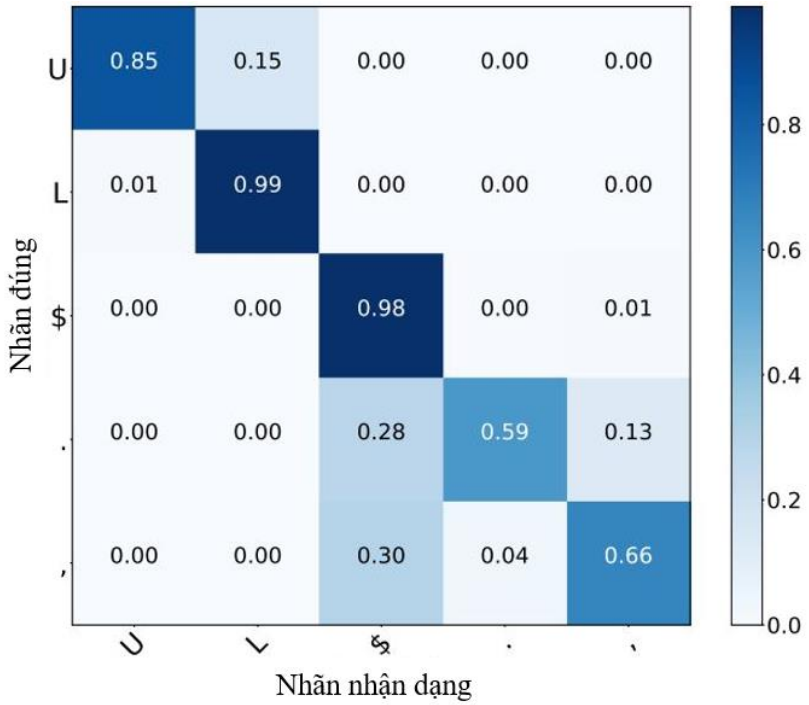
Bảng 4.4: Đánh giá mô hình NER đề xuất theo cách tiếp cận đường ống với các kiểu văn bản đầu vào khác nhau

Kiểu đầu vào	**F1**
Văn bản chuẩn ($Text_{VLSP\text{-}test}$)	0.9018
Văn bản đầu ra của ASR ($Text_{VLSP\text{-}Audio\text{-}ASR}$)	0.6319
Văn bản đầu ra của ASR + CaPu ($Text_{VLSP\text{-}Audio\text{-}ASR}$ + CaPu)	**0.6713**
Văn bản chuẩn bỏ dấu câu, chữ hoa ($Text_{VLSP\text{-}UnCaPu}$)	0.7535
Văn bản chuẩn bỏ dấu câu, chữ hoa + CaPu ($Text_{VSP\text{-}UnCaPu}$ + CaPu)	**0.8141**

Bảng 4.4 cũng chứng tỏ hiệu quả của mô hình CaPu trong việc cải thiện độ chính xác của mô hình NER làm việc trên văn bản đầu ra của ASR. Điểm F1 của mô hình NER tăng từ 0.6319 lên 0.6713 khi áp dụng mô hình này trên văn bản đầu ra của ASR và cải thiện hơn 0.06 điểm F1 (từ 0.7535 lên 0.8141) của mô hình NER khi áp dụng cho văn bản bỏ dấu câu và chữ hoa. Đặc biệt,

cùng với việc kết hợp mô hình CaPu thì với văn bản đầu ra văn bản chuẩn bỏ dấu câu, chữ hoa cho kết quả vượt trội (0.8141) so với văn bản đầu ra của ASR (0.6713), tương đương với 21.3%. Điều này chứng tỏ, lỗi từ văn bản đầu ra của ASR (chèn, xoá, thay thế từ,...) là một trong những yếu tố ảnh hưởng đến hiệu quả của mô hình. Do đó, việc tăng chất lượng của hệ thống ASR là nhu cầu cấp thiết.

Hình 4.5 chứng minh kết quả của mô hình CaPu trên văn bản chuẩn bỏ dấu câu và chữ hoa. Độ chính xác của khôi phục ký tự viết hoa là 0.85. Việc khôi phục dấu câu sẽ khó hơn, độ chính xác luôn duy trì ở mức gần 0.60 đối với dấu chấm (‘.’) và 0.66 đối với dấu phẩy (‘,’). Lỗi khôi phục dấu câu xảy ra khi mô hình CaPu không hiểu ý nghĩa của câu đầu vào và đặt dấu trống ($) sau những từ này.



Hình 4.5: Đánh giá mô hình CaPu trên văn bản chuẩn bỏ dấu câu và chữ hoa

Có thể nhận thấy, cách tiếp cận đường ống gặp phải những hạn chế nhất định. Hệ thống ASR, mô hình CaPu và mô hình NER được huấn luyện độc lập, dẫn đến ASR không được tối ưu hóa cho NER. Đồng thời, lỗi từ lan truyền qua các bước sẽ ảnh hưởng trực tiếp đến hiệu suất của hệ thống NER [56]. Do đó, gần đây, một số phương pháp tiếp cận E2E cho nhận dạng định

danh đã được tập trung nghiên cứu. Phần tiếp theo, luận án sẽ trình bày những kết quả nghiên cứu theo hướng tiếp cận này và có những đối sánh cụ thể.

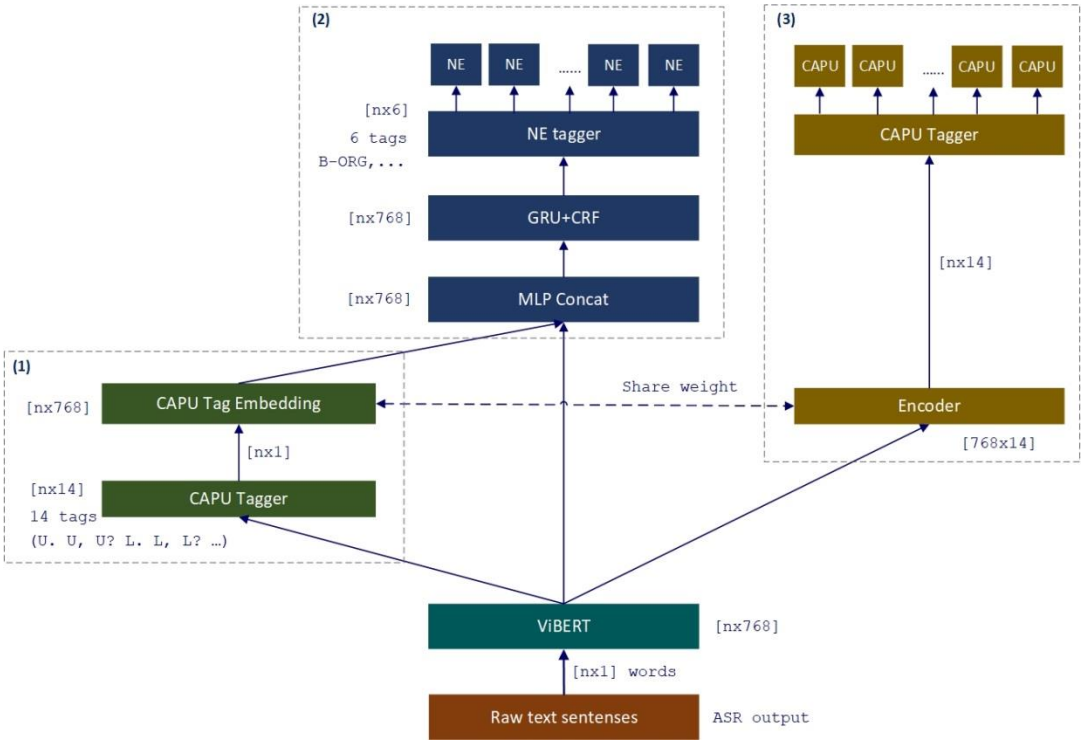
4.4. Nhận dạng thực thể định danh theo hướng tiếp cận E2E

Chương 1, mục 1.4.5.2 đã trình bày tổng quan nghiên cứu NER cho tiếng nói theo hướng E2E. Hầu hết các công bố đều có nguồn ngữ liệu phong phú như tiếng Anh, tiếng Pháp, tiếng Trung Quốc, đồng thời, các kết quả chưa cải thiện đáng kể so với tiếp cận đường ống. Các tác giả nhận định rằng lỗi của văn bản đầu ra ASR luôn là một thách thức và dữ liệu lớn giúp mô hình đạt hiệu suất cao hơn [54]. Đồng thời, việc kết hợp huấn luyện trước chữ hoa sẽ bổ sung thông tin giúp cải thiện mô hình NER [55]. Theo hiểu biết của nghiên cứu sinh, cho tới thời điểm hiện tại, chưa có công bố nào về NER cho văn bản đầu ra tiếng nói tiếng Việt theo hướng E2E. Mặc dù còn nhiều thách thức, nhưng có thể nhận thấy, khi lượng dữ liệu huấn luyện đủ lớn, mô hình E2E sẽ giúp tối ưu hóa quá trình huấn luyện, tất cả các tham số của mô hình được huấn luyện đồng thời, các sai số phát sinh giữa các thành phần đều được tính toán do đó giảm thiểu được lỗi lan truyền qua từng mô-đun. Việc huấn luyện và suy luận sử dụng mô hình E2E đơn giản hơn cũng như thuận tiện hơn cho việc đưa mô hình nhận dạng vào ứng dụng. Chính vì vậy, việc nghiên cứu mô hình E2E cho NER của tiếng nói tiếng Việt là cần thiết và có ý nghĩa thực tiễn. Mặc dù vậy, việc thiết kế mô hình E2E sẽ đòi hỏi sự tích hợp mức độ cao các mô hình thành phần vào một mô hình chung nhất, bỏ qua các khâu trung gian, khiến cho quá trình thiết kế khó khăn hơn. Đồng thời, nó đòi hỏi các thuật toán huấn luyện mô hình nâng cao như phương pháp chia sẻ trọng số (Weight tying), học đa tác vụ (Multi-task Learning), ... Cách tiếp cận huận luyện đa tác vụ đã được áp dụng để đề xuất mô hình E2E trong luận án. Phần tiếp theo, nghiên cứu sẽ trình bày về nội dung này.

4.4.1. Đề xuất mô hình

Luận án sử dụng ý tưởng từ học đa tác vụ cho nhiệm vụ nhận dạng thực thể định danh theo hướng E2E với mong muốn tác vụ khôi phục dấu câu, chữ hoa sẽ bổ trợ, giúp mô hình NER được nhận dạng tốt hơn. Phần tiếp theo sẽ trình bày mô hình đề xuất theo cách tiếp cận này.

Hình 4.6 biểu diễn mô hình E2E được đề xuất, bao gồm luồng NER chính dựa trên cấu trúc đường ống kết hợp với một luồng nhận dạng dấu câu, chữ hoa có vai trò bổ sung thông tin về dấu câu và chữ hoa cho khâu nhận dạng thực thể định danh. Dữ liệu đưa vào mô hình là văn bản đầu ra của ASR tiếng Việt không dấu câu, không chữ hoa có độ dài *n*. Trong quá trình nhận dạng, một số câu xuất hiện những lỗi như thay thế, chèn, và xóa khiến cho quá trình nhận dạng thực thể định danh trở nên khó khăn hơn. Câu đầu vào được đưa qua bộ biểu diễn ngôn ngữ tiếng Việt ViBERT. Ở nghiên cứu này, tiếp cận học chuyển giao được áp dụng với mô hình ViBERT là mô hình đã được tiền huấn luyện và được giữ nguyên trong mô hình E2E được đề xuất ở đây. Đầu ra của ViBERT là một ma trận có kích thước *(n x768)* là một biểu diễn dạng ma trận của câu đầu vào. Ma trận biểu diễn này được đưa đồng thời đến ba khối: (1) Khối trích xuất thông tin bổ trợ dấu câu, chữ hoa, (2) khối nhận dạng NER, và (3) khối học bổ trợ nhận dạng dấu câu chữ hoa CaPu theo cơ chế học đa tác vụ.



Hình 4.6: Đề xuất kiến trúc NER theo tiếp cận E2E

Kí hiệu chuỗi đầu vào được mã hóa $I=\{w_1,w_2,\ldots,w_n\}$, đầu ra của ViBERT sẽ là $E_{ViBERT} = \{E_1,E_2,E_3,\ldots,E_n\}$ có kích thước $[n,d] \in \mathbb{R}^{n\times d}$ trong

đó *n* là độ dài câu đầu vào và *d = 768* là kích thước lớp ẩn cuối cùng của bộ ViBERT.

$$E_{ViBERT} = ViBERT(I) \quad (3.2)$$

Đầu ra của bộ ViBERT chứa các mã nhúng theo ngữ cảnh của từng từ mã đầu vào w_i. Để tránh hiện tượng quá khớp, nghiên cứu đã thêm một lớp Dropout với tỷ lệ 0.1 trên đầu ra của bộ ViBERT.

(1) Khối trích xuất thông tin bổ trợ CaPu gồm có khối gán nhãn CaPu (CaPu tagger) là một mô hình đã được tiền huấn luyện theo phương pháp và cấu trúc như đã trình bày ở Chương 3, trong đó đầu ra bổ sung thêm một khối mã hóa CaPu có nhiệm vụ mã hóa các nhãn dấu câu, chữ hoa nhằm bổ sung thông tin về dấu câu, chữ hoa cho khối NER. Để phù hợp với dữ liệu là văn đầu ra của ASR, mô hình gãn nhãn CaPu này tiếp tục được tinh chỉnh trong quá trình huấn luyện chung của cả mô hình E2E. Đầu ra của khối mã hóa CaPu là véc tơ $T_{CaPu} \in \mathbb{R}^{n \times d}$.

$$T_{CaPu} = CaPu\_tagger(E_{ViBERT}) \quad (3.3)$$

Khối mã hóa CaPu (CaPu tag embedding) thực chất là một mạng nơ ron truyền thẳng hai lớp có đầu vào là một véc tơ T_{CaPu} có độ dài *N* là độ dài của câu đầu vào với các phần tử là các nhãn CaPu được xác định bởi khối gán nhãn CaPu. Đầu ra của khối mã hóa CaPu là một ma trận $E_{CaPu} \in \mathbb{R}^{n \times d}$ chứa thông tin mã hóa của các nhãn CaPu của cầu đầu vào.

$$E_{CaPu} = CaPu\_tag\_embedding(T_{CaPu}) \quad (3.4)$$

(2) Khối NER là khối đảm nhiệm tác vụ chính của mô hình. Khối này gồm có đầu vào là ma trận biểu diễn của câu đầu vào bởi ViBERT được kết hợp với ma trận mã hóa CaPu tag là đầu ra của khối (1). Việc kết hợp với đầu ra của khối (1) sẽ cung cấp thêm thông tin bổ trợ về dấu câu, chữ hoa, giúp cho việc gán nhãn NER sẽ chính xác hơn.

Việc kết hợp ma trận mã biểu diễn véc-tơ ViBERT với ma trận mã nhúng CaPu để bổ sung thông tin về dấu câu, chữ hoa được thực hiện bởi khối MLP_concat. Thay vì kết hợp thông tin bằng các phép toán học thông thường

giữa hai ma trận, trong mô hình này sử dụng một mạng nơ ron truyền thẳng hai lớp ẩn với đầu vào là ma trận ghép của hai ma trận mã nhúng $[E_{CaPu}\&E_{ViBERT}] \in \mathbb{R}^{n\times 2\times d}$. Đầu ra của khối MLP_concat là ma trận mã nhúng $E_{CaPu-ViBERT} \in \mathbb{R}^{n\times d}$. Việc kết hợp mã nhúng sử dụng mạng nơ ron truyền thẳng cho phép huấn luyện bộ kết hợp để có thể cập nhật các trọng số của nó một cách linh hoạt theo dữ liệu huấn luyện.

$$E_{CaPu\text{-}ViBERT} = MLP\_concat(E_{CaPu}, E_{ViBERT}) \quad (3.5)$$

Mạng GRU mã hóa chuỗi đầu vào thành một chuỗi các biến ẩn $(h_1, \ldots h_n)$ sử dụng các véc-tơ này cùng với một lớp Softmax để tạo ra một chuỗi các quyết định phân loại độc lập. Điều này đã khá thành công trong các nhiệm vụ như gắn thẻ POS. Tuy nhiên, NER là một nhiệm vụ phụ thuộc nhiều hơn vào ngữ pháp và các từ ngữ cảnh trong câu. Ví dụ: nếu một mã thông báo được đặt trước nhãn 'B-PERSON', thì khả năng nhãn tiếp theo là 'I-PERSON' là rất cao và ngược lại, khả năng nhãn tiếp theo là 'I-ORG' là rất nhỏ. Do đó, một lớp đầu ra cho phép dự đoán có cấu trúc có thể rất hữu ích.

Lớp CRF ngay sau lớp GRU cho phép dự đoán tuần tự đó. CRF sử dụng các véc-tơ ẩn là đầu ra từ GRU làm phép đo $P \in R^{n\times k}$ và ma trận chuyển trạng thái $A \in R^{(k+2)\times(k+2)}$, ma trận này có thể sử dụng các phép đo trước đó và trong tương lai để dự đoán lớp hiện tại. n là số từ trong chuỗi và k là số nhãn đầu ra. Ma trận A là ma trận vuông kích thước $(k + 2)$ vì y_0 và y_{n+1} là nhãn đầu và nhãn cuối. Với hai ma trận này, hàm đánh giá của một chuỗi đầu ra nhất định được tính bằng:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \quad (3.6)$$

Từ hàm đánh giá trên, xác suất có điều kiện $p(y|X)$ được tính theo công thức sau:

$$p(y|X) = \frac{\exp(s(X,y))}{Z(X)} \quad (3.7)$$

trong đó $Z(X)$ là tổng lũy tích của $exp(s(X, y))$ với tất cả khả năng y.

(3) Khối học bổ trợ nhận dạng dấu câu chữ hoa nhận ma trận biểu diễn đầu ra từ ViBERT và nhận dạng CaPu bằng cách trích xuất thông tin từ ma

trận này sử dụng bộ mã hóa là một mang nơ-ron truyền thẳng có ma trận trọng số chính là ma trận chuyển vị của khối mã hóa CaPu trong khối (1). Đầu ra của khối này chính là xác suất của các nhãn CaPu tương ứng với câu đầu vào và hàm mất mát được tính dựa trên các nhãn CaPu. Có thể nói rằng bộ mã hóa ở khối này và khối mã hóa CaPu có kết nối với nhau theo cơ chế chia sẻ tham số [87], trong đó, ma trận trọng số của CaPu Embedding được sao chép từ ma trận chuyển vị của bộ mã hóa ở khối (3) theo công thức sau.

$$W_{emb} = W_{enc}^{T} \quad (3.8)$$

trong đó, W_{emb} là ma trận trọng số của bộ mã hóa nhãn CaPu, W_{enc} là ma trận trọng số của bộ Encoder. Cơ chế chia sẻ trọng số giúp giảm bớt số lượng tham số của mô hình giúp quá trình huấn luyện nhanh hơn, đồng thời làm hạn chế hiện tượng quá khớp do số lượng tham số quá lớn.

Quá trình huấn luyện của mạng được thực hiện theo tiếp cận E2E với phương pháp huấn luyện đa tác vụ là tác vụ NER và CaPu. Trong đó, tác vụ NER được coi là tác vụ chính còn tác vụ CaPu là tác vụ phụ trợ (*Auxiliary task*). Giá trị mất mát của mô hình E2E được tính bằng tổng có trọng số của hai giá trị mất mát của hai tác vụ:

$$L_{mtl} = \alpha L_{NER} + \beta L_{CaPu} \quad (3.9)$$

trong đó, α là trọng số của giá trị mất mát của tác vụ NER và β là trọng số cho giá trị mất mát của tác vụ CaPu. Việc chọn α và β phụ thuộc vào mức độ quan trọng của từng tác vụ. Trong nghiên cứu này, tác vụ NER được coi là tác vụ chính, tác vụ CaPu là tác vụ phụ trợ, do đó α, β được lựa chọn $\alpha = 0.6$ và $\beta = 0.4$.

Mặc dù trong mô hình hợp nhất có sử dụng các mô hình tiền huấn luyện như ViBERT, CaPu theo tiếp cận học chuyển giao, tuy nhiên trong quá trình huấn luyện, toàn bộ các tham số mô hình được cập nhật theo thuật toán lan truyền ngược với cùng một hàm mất mát L_{mtl} và trên cùng một luồng đồ thị tính toán với tiếp cận học đa tác vụ, cho nên có thể nói mô hình được đề xuất là một mô hình hợp nhất. Hơn nữa, trong quá trình suy diễn để nhận

dạng thực thể định danh, dữ liệu là văn bản đầu ra của ASR được đưa qua luồng đồ thị tính toán duy nhất đến đầu ra nhận dạng NER mà không phải qua các bước trung gian làm phát sinh thời gian cũng như sai số. Do đó có thể nói mô hình được đề xuất là mô hình E2E đầy đủ. Quá trình huấn luyện mô hình theo tiếp cận E2E được mô tả trong thuật toán sau:

Thuật toán E2E

Input: Cho một tập các chuỗi các từ đầu vào *D* từ tập dữ liệu huấn luyện, $I = \{w_1, w_2, \dots, w_n\}$ là một đoạn trong tập *D*

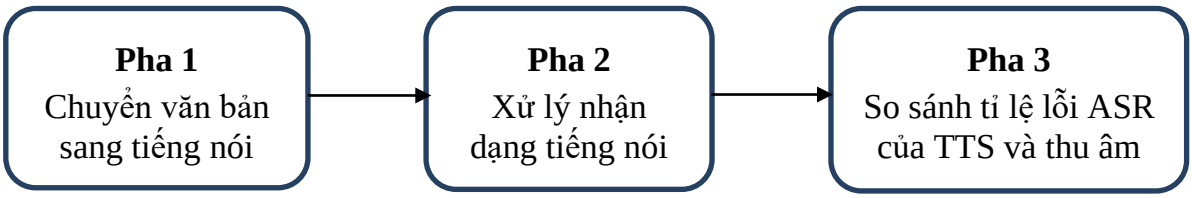
Output: Các tham số của mô hình đã được huấn luyện

1: Khởi tạo tất cả các tham số cần học θ
2: **repeat**
3: **for** b = 1:n_batches **do**
4: Sinh một tập các mẫu S_b từ *D*
5: **for** chuỗi *I* thuộc S_b **do**
6: E_{ViBERT} = ViBERT(I)
7: T_{caPu} = CaPu_Tagger(E_{ViBERT})
8: E_{CaPu} = CaPu_tag_embedding(T_{CaPu})
9: P_{NER} = NER_tagger(MLP_concat(E_{caPu}, E_{ViBERT})
10: P_{caPu} = CaPuTagger(Encoder(E_{ViBERT}))
11: **end for**
12: $L_{mtl} = \alpha L_{NER} + \beta L_{CaPu}$
13: Sử dụng thuật toán lan truyền ngược để cập nhật chung các tham số θ của mô hình E2E bằng cách tối thiểu hóa hàm mất mát L_{mtl} theo từng batch.
14: **end for**
15: **until** thỏa mãn điều kiện dừng

4.4.2. Chi tiết xây dựng dữ liệu

Hiện tại, chưa có nhiều bộ dữ liệu quy mô lớn công khai cho tiếng nói tiếng Việt, một số bộ được biết đến như MICA VNSpeechCorpus, AIlab

VIVOS, VOV, ... Tuy vậy, những kho dữ liệu đó không có âm thanh quy mô lớn, chất lượng cao. Bên cạnh đó, rất khó để thu thập thủ công kho ngữ liệu tiếng nói chất lượng cao vì tốn thời gian và chi phí, vì vậy nghiên cứu đề xuất thu thập tự động để giảm bớt công sức bằng cách thử nghiệm phương án sau với mong muốn có thể lấy được dữ liệu âm thanh tiếng nói từ hệ thống tổng hợp tiếng nói (Text To Speech - TTS) thay thế phương án thu âm.



Hình 4.7: Các pha trong quá trình thu thập, xử lý dữ liệu

Hình 4.7 mô tả các pha trong quá trình xây dựng bộ dữ liệu huấn luyện cho mô hình E2E được đề xuất.

Pha 1: Chuyển dữ liệu văn bản sang tiếng nói nhờ công cụ TTS

Tổng hợp tiếng nói là việc tạo ra tiếng nói của con người một cách nhân tạo. Một hệ thống máy tính thực hiện mục đích này được gọi là một hệ thống tổng hợp tiếng nói. Chất lượng của một hệ thống tổng hợp tiếng nói được đánh giá dựa trên độ "giống" đối với tiếng nói của người thật và khả năng để người nghe có thể hiểu được hết ý nghĩa của văn bản. Một hệ thống chuyển văn bản thành tiếng nói là một hệ thống có đầu vào là một văn bản và đầu ra là một sóng âm thanh.

TTS ở Việt Nam cũng đã được nghiên cứu từ khá lâu. Phần mềm VnSpeech được biết đến là hệ thống tổng hợp tiếng nói đầu tiên của Tiếng Việt, phần mềm này sử dụng phương pháp tổng hợp FORMANT. Hệ thống có thể đọc được hầu hết các âm tiết tiếng Việt ở mức nghe rõ nhưng mức độ tự nhiên không cao. Phần mềm VietSound được phát triển tại đại học Bách Khoa Thành phố Hồ Chí Minh, sử dụng giải thuật TD-PSOLA để tổng hợp các nguyên âm đơn và phương pháp tổng hợp FORMANT để tổng hợp các phụ âm, nguyên âm và âm vần đơn giản. Phần mềm này cũng chưa đạt đến mức

độ tự nhiên gần giống với tiếng nói con người. Cả hai phần mềm trên đều có nhược điểm là âm thanh thu được rời rạc, thiếu tự nhiên.

Hiện nay, có một số hệ thống TTS rất mạnh đã được triển khai, ứng dụng rộng rãi. Có thể kể đến như: Dịch vụ chuyển đổi văn bản thành giọng nói của FPT.AI Text to Speech, ứng dụng công nghệ tổng hợp tiếng nói và công nghệ học sâu đã cho phép tổng hợp tiếng nói tự nhiên với các lựa chọn phong phú về giọng đọc (nam/nữ) và ngữ âm (Bắc, Trung, Nam). Viettel cũng cho ra đời một hệ thống TTS - VTCC.AI - giọng đọc tự nhiên, đa dạng vùng miền, ngắt nghỉ tự động và kết hợp biểu cảm chính xác. Đặc biệt, không thể không nhắc tới Vbee - đây là công ty công nghệ chuyên nghiên cứu, cung cấp các giải pháp, dịch vụ thông minh về tiếng nói trí tuệ nhân tạo. Vbee là đơn vị đầu tiên công bố và thương mại hóa giải pháp về tiếng nói trí tuệ nhân tạo tiếng Việt có cảm xúc tại Việt Nam. Tiếng nói trí tuệ nhân tạo của Vbee được thiết kế với rất nhiều giọng đọc tiếng Việt từ giọng nam đến giọng nữ, giọng miền Bắc hay miền Nam…, với chất giọng tự nhiên. Điều này cho phép người nhận thông tin thấy gần gũi hơn, thân thiện hơn và dễ nghe hơn. Ngoài ra, còn một số hệ thống TTS khác của MobiPhone, Google, ... Được xếp hạng là một trong các phần mềm TTS tốt nhất hiện nay, Google ngoài nổi tiếng với vai trò công cụ tìm kiếm, cũng cung cấp cho người dùng công cụ chuyển đổi văn bản thành giọng nói có âm thanh tự nhiên. Google sử dụng kiến thức chuyên môn về tổng hợp tiếng nói của DeepMind để cung cấp một tiếng nói chân thực mà người nghe sẽ khó phân biệt được. Chính vì vậy, hệ thống TTS của Google đã được lựa chọn với bộ dữ liệu VLSP 2018 cho thực nghiệm của nghiên cứu này.

Pha 2: Xử lý nhận dạng tiếng nói qua hệ thống ASR

Sau khi thu được bộ audio TTS, tiến hành đưa qua hệ thống ASR. Trong pha này hệ thống ASR của VAIS được sử dụng bởi các lý do sau:

-Trong bài đối sánh giữa các hệ thống nhận dạng tiếng nói tiếng Việt tại Việt Nam [6], các tác giả đã đánh giá các hệ thống ASR tiếng Việt từ các

công ty hàng đầu của Việt Nam hiện nay như VAIS, Viettel, Zalo, FPT và Google. Kết quả đánh giá với các mẫu là các bản tin truyền hình trên Youtube. Mặc dù số lượng mẫu còn khiêm tốn nhưng cũng đủ để thấy là kết quả thể hiện VAIS vượt trội hơn các hệ thống còn lại.

-Theo cách tiếp cận đường ống, nghiên cứu đã sử dụng hệ thống ASR VAIS để nhận dạng tiếng nói thu âm.

Do đó, để thuận tiện cho việc đối sánh kết quả, cách tiếp cận E2E này sẽ tiếp tục lựa chọn hệ thống ASR VAIS cho thực nghiệm.

Pha 3: So sánh văn bản TTS-ASR với văn bản thu âm-ASR (REC-ASR)

Thông thường, các lỗi đầu ra ASR tiếng Việt thường tập trung vào một số lỗi cơ bản sau:

-Lỗi về xử lý dữ liệu số, ngày tháng, tiền tệ. Ví dụ: 2004 - "hai không linh thư"; c302 - "c ba không hay"; 50 % miles - "năm mươi phần trăm vai"; 360kg - "ba trăm sáu mươi ki lô" hay "ba sáu mươi ki lô gam", ...

-Lỗi về các từ ngoại lai (outlier) bao gồm,

+Từ nước ngoài. Ví dụ: britney - "whitney"; christian aguilera - "gibson a gi lê ra"; nikola jokic - "nghi cô la du kích"; china - "chi nờ" hay "chai nờ" hay "trai ờ";…

+Từ viết tắt: twc - "tywin đắp liu si"; hlv - "ghét eo vi"; ubnd - "thu bên", csgt - "xi ét ti"; atgt - "à tê giê tê", ...

-Một số các lỗi khác, như: smartphone - "mát phôn"; king - "kinh"; gram - "ram" hay "giam" hay "gờ ram"; windows - "nguyên đâu" hay "huy đâu", ...

Chính vì vậy, nghiên cứu cũng tập trung so sánh các lỗi dựa trên ba tiêu chí thống kê này. Với 241.899 từ trong bộ dữ liệu, Bảng 4.5 cho biết tỉ lệ phần trăm lỗi theo cách TTS-ASR và REC-ASR.

Bảng 4.5: Tỉ lệ lỗi của TTS-ASR và REC-ASR trên dữ liệu kiểu số, dữ liệu ngoại lại và các lỗi khác

	Kiểu số	**Dữ liệu ngoại lai**	**Lỗi khác**	**Tổng (%)**
TTS-ASR	1.06	2.42	1.58	**5.07**
REC-ASR	1.43	2.37	2.23	**6.03**

Theo bảng thống kê, độ chênh lệch lỗi ASR giữa hai cách thu thập dữ liệu là không đáng kể. Chính vì vậy, để thu thập được lượng dữ liệu lớn cho huấn luyện, luận án đã sử dụng phương án thu thập dữ liệu sau đó đưa qua hệ thống TTS để được bộ dữ liệu âm thanh và đưa qua hệ thống ASR để được văn bản tương ứng.

Tất cả dữ liệu văn bản của VLSP sẽ sử dụng hệ thống TTS của Google để tạo ra dữ liệu âm thanh tổng hợp. Sau đó, bộ dữ liệu âm thanh tổng hợp sẽ qua hệ thống ASR của VAIS để được bộ dữ liệu văn bản $Text_{VLSP\text{-}TTS\text{-}ASR}$ phục vụ huấn luyện mô hình E2E.

Bên cạnh đó, các bộ dữ liệu $Text_{CaPu}$ và $Text_{ViBERT}$ vẫn được sử dụng để tiền huấn luyện mô hình khôi phục dấu câu, chữ hoa và huấn luyện mô hình ViBERT tương ứng.

Bộ dữ liệu $Text_{VLSP\text{-}Audio\text{-}ASR}$ là dữ liệu kiểm tra được ghi âm bởi bốn người đọc, với môi trường khác nhau, trong 26 giờ âm thanh, được nhận dạng thông qua hệ thống ASR của VAIS để thu được dữ liệu văn bản đầu ra ASR.

4.4.3. Thiết lập mô hình

Mô hình ViBERT được thiết lập và trình bày chi tiết ở mục 4.3.2.

Trong mô hình E2E, nghiên cứu đã sử dụng chung thiết lập cho bộ gán nhãn NER và CaPu. Mỗi bộ gán nhãn gồm có 4 lớp GRU hai chiều với 512 phần tử ẩn. Một lớp CRF ở đầu ra để tính toán xác suất các nhãn.

4.4.4. Kết quả thực nghiệm

Kết quả bảng 4.6 cho thấy rằng nếu văn bản đầu ra của ASR được đưa trực tiếp vào mô hình NER, kết quả nhận dạng thực thể sẽ giảm từ 0.9018 xuống 0.6319.

Bảng kết quả 4.6 chứng tỏ việc kết hợp mô hình học tập đa tác vụ với mô hình CaPu giúp cải thiện độ chính xác của mô hình NER trên văn bản đầu ra của ASR khi điểm F1 của mô hình NER tăng gần 0.05 từ 0.6319 lên 0.6780. Mô hình này cũng giúp cải thiện xấp xỉ 0.14 điểm F1 (từ 0.6780 lên 0.8178) khi áp dụng cho văn bản chuẩn bỏ dấu câu, chữ hoa so với văn bản

đầu ra của ASR. Điều này cũng cho thấy sự cần thiết phải cải tiến mô hình ASR để giảm các lỗi "dị thường" về chèn, xóa, thay thế và thêm từ trong văn bản đầu ra của ASR.

Bảng 4.6: Đánh giá mô hình NER đề xuất theo cách tiếp cận E2E với các kiểu văn bản đầu vào khác nhau

Các kiểu dữ liệu đầu vào	**F1**
Văn bản chuẩn (*Text*$_{VLSP\text{-}test}$)	0.9018
Văn bản đầu ra của ASR (*Text*$_{VLSP\text{-}Audio\text{-}ASR}$)	0.6319
Văn bản đầu ra của ASR+CaPu E2E (*Text*$_{VLSP\text{-}Audio\text{-}ASR}$ +CaPu E2E)	**0.6780**
Văn bản chuẩn bỏ dấu câu, chữ hoa+CaPu E2E (*Text*$_{VLSP\text{-}UnCaPu}$ +CaPu E2E)	**0.8178**

Điều này cho thấy việc "làm sạch" dữ liệu qua từng bước trong mô hình đường ống vẫn có hiệu quả nhất định và sẽ cải thiện kết quả nếu hệ thống ASR tốt.

Bảng kết quả 4.7 cho thấy mô hình E2E có kết quả xử lý tốt hơn so với mô hình đường ống, cụ thể, tăng 0.0067 với văn bản đầu ra của ASR (từ 0.6713 lên 0.6780) và 0.0037 (từ 0.8141 lên 0.8178) đối với văn bản chuẩn bỏ dấu câu, chữ hoa.

Bảng 4.7: So sánh mô hình E2E với mô hình đường ống

Hệ thống NER	**F1**
Văn bản đầu ra của ASR + CaPu Pipeline (*Text*$_{VLSP\text{-}Audio\text{-}ASR}$ +CaPu Pipeline)	0.6713
Văn bản đầu ra của ASR + CaPu E2E (*Text*$_{VLSP\text{-}Audio\text{-}ASR}$ +CaPu E2E)	**0.6780**
Văn bản chuẩn bỏ dấu câu, chữ hoa + CaPu Pipeline (*Text*$_{VLSP\text{-}UnCaPu}$ +CaPu Pipeline)	0.8141
Văn bản chuẩn bỏ dấu câu, chữ hoa + CaPu E2E (*Text*$_{VLSP\text{-}UnCaPu}$ +CaPu E2E)	**0.8178**

Mặc dù với kết quả chưa cải thiện tốt hơn nhiều, nhưng với mô hình đường ống, quá trình huấn luyện các thành phần riêng biệt, đòi hỏi các thuật toán huấn luyện riêng và hàm mất mát riêng với ứng với mỗi thành phần, do đó cần số lượng lớn siêu tham số (hyperparameter) dẫn đến phức tạp trong huấn luyện. Các sai số phát sinh trong mỗi thành phần không được tính toán khi kết hợp với các thành phần khác nên sai số tích lũy lớn. Ngược lại, với mô hình E2E, tất cả các tham số của mô hình được huấn luyện đồng thời với chỉ một hàm mất mát. Toàn bộ luồng đồ thị tính toán (computational flow graph) được tối ưu đồng thời bởi thuật toán lan truyền ngược. Các sai số phát sinh giữa các thành phần đều được tính toán do đó giảm thiểu sai số chung. Quá trình suy diễn cũng đơn giản và nhanh hơn khi không có những bước chuyển trung gian giữa các mô hình thành phần. Chính vì vậy, mô hình E2E vẫn có những lợi thế nhất định và việc tiếp tục cải tiến mô hình E2E cho bài toán NER tiếng nói tiếng Việt là cần thiết để đạt được hiệu suất cao hơn và tận dụng được tính ưu việt trong huấn luyện mô hình và trong triển khai ứng dụng vào thực tế.

4.5. Kết luận Chương 4

Chương 4 đã đề xuất mô hình NER cho hệ thống ASR tiếng Việt theo hướng tiếp cận đường ống và E2E. Thực nghiệm đã chứng minh hiệu quả của việc kết hợp mô hình CaPu giúp tăng hiệu suất mô hình NER. Luận án đã giới thiệu bộ dữ liệu đầu tiên cho nghiên cứu NER cho văn bản đầu ra của ASR tiếng Việt. Đồng thời, nghiên cứu cũng trình bày tác động hiệu quả của mô hình ngôn ngữ được huấn luyện trước cho ngôn ngữ tiếng Việt để áp dụng cho nhiệm vụ NER và đã đạt được kết quả khả quan trên bộ dữ liệu NER của VLSP 2018.

Mô hình E2E kết quả tốt hơn nhưng chưa đáng kể so với mô hình đường ống (0.0067 với văn bản đầu ra của ASR và 0.0037 đối với văn bản chuẩn bỏ dấu câu, chữ hoa). Việc kết hợp mô hình học tập đa tác vụ với mô hình khôi phục dấu chấm câu và chữ hoa đã tăng điểm F1 lên xấp xỉ 0.05 và cải thiện rõ rệt 0.14 điểm F1 của mô hình NER khi áp dụng cho văn bản chuẩn bỏ chữ hoa, dấu câu.

KẾT LUẬN

Văn bản đầu ra của một hệ thống nhận dạng tiếng nói thường cần được hậu xử lí, với các yêu cầu chuẩn hoá về dấu câu, chữ hoa, chữ thường. Bên cạnh đó, nhận dạng các thực thể định danh cũng là một bài toán quan trọng, cho phép khai thác văn bản thu được hiệu quả hơn. Luận án này tập trung đề xuất mô hình chuẩn hóa văn bản đầu ra của ASR tiếng Việt, các mô hình NER cho văn bản đầu ra của ASR tiếng Việt. Kết quả nghiên cứu chính của luận án được trình bày như sau:

1.Xây dựng các bộ dữ liệu ban đầu phục vụ cho thực nghiệm các mô hình chuẩn hoá và nhận dạng thực thể định danh cho văn bản đầu ra của hệ thống ASR tiếng Việt.

2.Thiết kế mô hình Transformer Encoder - CRF cho bài toán khôi phục viết hoa và dấu câu cho văn bản đầu ra của ASR tiếng Việt. Luận án đề xuất cách phân chia đoạn mới cho câu đầu vào sử dụng phân đoạn, hợp nhất các đoạn chồng lấn, giúp các từ xung quanh đoạn cắt có nhiều ngữ cảnh để nhận dạng được chính xác hơn. Đầu ra của mô hình là văn bản tiếng Việt có đầy đủ dấu câu, chữ hoa, giúp tăng độ chính xác của quá trình nhận dạng thực thể định danh ở bước tiếp theo.

3.Đề xuất mô hình biểu diễn ngôn ngữ tiền huấn luyện cho văn bản tiếng Việt với tên gọi ViBERT dựa theo kiến trúc RoBERTa. Mô hình được huấn luyện dựa trên tập dữ liệu lớn văn bản tiếng Việt chính thống để biểu diễn ngôn ngữ tiếng Việt trong không gian véc-tơ giúp tăng hiệu quả áp dụng các thuật toán học sâu trong XLNNTN tiếng Việt. Mô hình được áp dụng vào các mô-đun biểu diễn véc-tơ từ cho các mô hình NER tiếp theo.

4.Xây dựng mô hình đường ống cho bài toán NER tiếng nói tiếng Việt. Nghiên cứu cho thấy tác động hiệu quả của mô hình biểu diễn ngôn ngữ được tiền huấn luyện ViBERT để áp dụng cho nhiệm vụ NER trên văn bản đầu ra của ASR tiếng Việt và đã đạt được kết quả khả quan. Đồng thời nghiên cứu

cũng chứng tỏ được tầm quan trọng của việc kết hợp mô hình CaPu vào chuẩn hóa văn bản đầu vào cho mô hình NER giúp cải thiện đáng kể hiệu suất của mô hình.

5.Thiết kế mô hình E2E giải quyết bài toán NER cho tiếng nói tiếng Việt cùng với các đề xuất mới như kỹ thuật chia sẻ tham số, kỹ thuật huấn luyện đa tác vụ. Bên cạnh thực nghiệm cho thấy đạt hiệu suất tương đương mô hình đường ống, mô hình E2E còn cho thấy ưu thế của việc tích hợp hệ thống trên một mô hình duy nhất giúp thuận lợi cho quá trình huấn luyện, giảm thiểu sai số phát sinh giữa các thành phần, tăng tốc độ thực thi, tăng khả năng triển khai trong các ứng dụng thực tiễn.

Từ những kết quả đạt được, luận án cũng đặt ra các vấn đề cần tiếp tục được nghiên cứu trong thời gian tới:

1.Nghiên cứu giải pháp giảm thiểu sự ảnh hưởng của lỗi dữ liệu trong văn bản đầu ra của ASR, đồng thời, bổ sung bộ dữ liệu từ điển NER chuẩn mực phục vụ cho mục đích huấn luyện nhằm nâng cao chất lượng mô hình NER tiếng Việt.

2.Thực nghiệm NER cho khôi phục chữ hoa, giúp hệ thống E2E ASR được cải thiện hơn.

3.Thực nghiệm các mô hình đề xuất trong nghiên cứu này với các bộ dữ liệu tiếng Anh, Trung Quốc, ... đã công bố để có đối sánh về tính hiệu quả của mô hình.

4.Áp dụng mô hình đề xuất để nhận dạng thực thể định danh cho văn bản thuộc các lĩnh vực chuyên biệt, ví dụ như trong văn bản y sinh, họp Quốc hội, ... nhằm làm rõ tính khả thi của mô hình.

5.Tiếp tục cải tiến mô hình E2E và các thuật toán huấn luyện tương ứng để đạt hiệu suất tốt hơn cho bài toán NER tiếng nói tiếng Việt.

DANH MỤC CÔNG TRÌNH CỦA TÁC GIẢ

[CT1]. Nguyen Thi Minh Huyen, Ngo The Quyen, Vu Xuan Luong, Tran Mai Vu, **Nguyen Thi Thu Hien**, *"VLSP shared task: Named Entity Recognition"*, Journal of Computer Science and Cybernetics, V.34, N.4, p.283-294, 2018.

[CT2]. **Thu Hien Nguyen**, Thai Binh Nguyen, Vu Bao Hung Nguyen, Truong Quoc Do, Chi Mai Luong, Minh Huyen Nguyen, *"Recovering Capitalization for Automatic Speech Recognition of Vietnamese using Transformer and Chunk Merging"*, Proceedings of the 11th International conference on Knowledge and Systems Engineering (KSE), p.430-434, 2019.

[CT3]. Thai Binh Nguyen, Vu Bao Hung Nguyen, **Thu Hien Nguyen**, Ngoc Phuong Pham, The Loc Nguyen, Quoc Truong Do, Chi Mai Luong, *"Fast and Accurate Capitalization and Punctuation for Automatic Speech Recognition Using Transformer and Chunk Merging"*, Proceedings of the COCOSDA, Philippines, p. 1-5, doi: 10.1109/O-COCOSDA46868.2019.9041202, 2019.

[CT4]. Thai Binh Nguyen, Quang Minh Nguyen, **Thu Hien Nguyen,** Quoc Truong Do, Chi Mai Luong, *"Improving Vietnamese Named Entity Recognition from Speech Using Word Capitalization and Punctuation Recovery Models"*, Proceedings of the Interspeech, p.4263-4267, Shanghai, China, 2020.

[CT5]. **Thu Hien Nguyen,** Thai Binh Nguyen, Ngoc Phuong Pham, Quoc Truong Do, Tu Luc Le, Chi Mai Luong, *"Toward Human-Friendly ASR Systems: Recovering Capitalization and Punctuation for Vietnamese Text"*, IEICE TRANSACTIONS on Information and Systems, Vol.E104-D, No.8, p.1195-1203 (SCIE, Q3), 2021.

[CT6]. **Thu Hien Nguyen**, Thai Binh Nguyen, Quoc Truong Do, Tuan Linh Nguyen, *"End-to-End named entity recognition for Vietnamese speech"*, Proceeding in the 25th conference of the Oriental COCOSDA, p.193-197, 979-8-3503-9855-7 ©2022 IEEE 2022.

TÀI LIỆU THAM KHẢO

[1]. Nadkarni, P. M., Ohno-Machado, L., Chapman, W. W., *"Natural language processing: an introduction"*, Journal of the American Medical Informatics Association, https//doi.org/10.1136/amiajnl-2011-000464v, vol. 18, no. 5, pp. 544-551, 2011.

[2]. Khurana, D., Koli, A., Khatter, K., Singh, S., *"Natural language processing: State of the art, current trends and challenges"*, Multimedia tools and applications, 82(3), pp.3713-3744, 2023.

[3]. Kaddari, Z., Mellah, Y., Berrich, J., Belkasmi, M. G., Bouchentouf, T., *"Natural Language Processing: Challenges and Future Directions"*, Artificial Intelligence and Industrial Applications: Artificial Intelligence Techniques for Cyber-Physical, Digital Twin Systems and Engineering Applications, Springer International Publishing, vol. 144, pp. 236-246, 2021.

[4]. L. Yu, D. Deng, *"Automatic Speech Recognition"*, Vol. 1. Berlin: Springer London. https://doi.org/10.1007/978-1-4471-5779-3, 2016.

[5]. Morris, A. C., Maier, V., Green, P., *"From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition"*, The Eighth International Conference on Spoken Language Processing, 2004.

[6]. Nga, C. H., Li, C. T., Li, Y. H., Wang, J. C., *"A Survey of Vietnamese Automatic Speech Recognition",* 2021 9th International Conference on Orange Technology (ICOT), IEEE, pp. 1-4, 2021.

[7]. Thanh, P. V., Huy, D. D., Thanh, L. D., Tan, N. D., Anh, D. T. D., Trang, N. T. T., *"ASR-VLSP 2021: Semi-supervised Ensemble Model for Vietnamese Automatic Speech Recognition",* VNU Journal of Science: Computer Science and Communication Engineering, vol. 38, no. 1, 2022.

[8]. Batista, F., Caseiro, D., Mamede, N., Trancoso, I., *"Recovering capitalization and punctuation marks for automatic speech recognition: Case study for Portuguese broadcast news"*, Speech Communication, 50(10), pp. 847-862, 2008.

[9]. Coniam, D. , *"Evaluating the language resources of chatbots for their potential in English as a second language",* ReCALL, vol. 20, no. 1, pp. 98-116, 2008.

[10]. Nebhi, K., Bontcheva, K., Gorrell, G., *"Restoring capitalization in# tweets"*, Proceedings of the 24th International Conference on World Wide Web, pp. 1111-1115, 2015.

[11]. Cho, E., Niehues, J., Waibel, A., *"NMT-based segmentation and punctuation insertion for real-Time spoken language translation"*, Interspeech, pp. 2645-2649, doi: 10.21437/Interspeech.2017-1320, 2017.

[12]. Courtland, M., Faulkner, A., McElvain, G., *"Efficient automatic punctuation restoration using bidirectional transformers with robust inference",* Proceedings of the 17th International Conference on Spoken Language Translation, pp. 272-279, 2020.

[13]. Pham, T., Nguyen, N., Pham, Q., Cao, H., Nguyen, B., *"Vietnamese punctuation prediction using deep neural networks"*, SOFSEM 2020: Theory and Practice of Computer Science: 46th International Conference on Current Trends in Theory and Practice of Informatic, Proceedings 46, Springer International Publishing, pp. 388-400, 2020.

[14]. Tran, H., Dinh, C. V., Pham, Q., Nguyen, B. T., *"An Efficient Transformer-Based Model for Vietnamese Punctuation Prediction"*, Advances and Trends in Artificial Intelligence. From Theory to Practice: 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021, Proceedings, Part II 34, Springer International Publishing, pp. 47-58, 2021.

[15]. Thu Uyen, H. T., Tu, N. A., Huy, T. D., *"Vietnamese Capitalization and Punctuation Recovery Models"*, Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 3884-3888), 2022.

[16]. Lu, W., Ng, H. T., *"Better punctuation prediction with dynamic conditional random fields"*, Proceedings of the 2010 conference on empirical methods in natural language processing (EMNLP), pp. 177-186, 2010.

[17]. Batista, F., Caseiro, D., Mamede, N., Trancoso, I., *"Recovering punctuation marks for automatic speech recognition"*, Eighth Annual Conference of the International Speech Communication Association, Interspeech, vol. 3, pp. 1977-1980, 2007.

[18]. A. Vāravs, A., Salimbajevs, *"Restoring punctuation and capitalization using transformer models"*, Statistical Language and Speech Processing: 6th International Conference, Proceedings 6, Springer International Publishing, pp. 91-102, 2018.

[19]. Lita, L. V., Ittycheriah, A., Roukos, S., Kambhatla, N., *"Truecasing",* Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 152-159, 2003.

[20]. Rayson, S. J., Hachamovitch, D. J., Kwatinetz, A. L., Hirsch, S. M., *"Autocorrecting text typed into a word processing document"*, U.S. Patent No. 5,761,689. Washington, DC: U.S. Patent and Trademark Office, 1998.

[21]. Mikheev, A., *"A knowledge-free method for capitalized word disambiguation"*, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 159-166, 1999.

[22]. Caranica, A., Cucu, H., Buzo, A., Burileanu, C., *"Capitalization and punctuation restoration for Romanian language"*, University Politehnica of Bucharest Scientific Bulletin, 77(3), pp. 95-106, 2015.

[23]. Pauls, A., Klein, D., *"Faster and smaller n-gram language models"*, Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 258-267, 2011.

[24]. Batista, F., Trancoso, I., Mamede, N., *"Automatic recovery of punctuation marks and capitalization information for Iberian languages"*, I Joint SIG-IL/Microsoft Workshop on Speech An Language Technologies for Iberian Languages, Porto Salvo, Portugal, pp. 99-102, 2009.

[25]. Hasan, M., Doddipatla, R., Hain, T., *"Multi-pass sentence-end detection of lecture speech"*, Fifteenth Annual Conference of the International Speech Communication Association, Interspeech, pp. 2902-2906, 2014.

[26]. Chelba, C., Acero, A., *"Adaptation of maximum entropy capitalizer: Little data can help a lot"*, Computer Speech & Language, 20(4), pp. 382-399, 2006.

[27]. Lafferty, J., McCallum, A., Pereira, F. C., *"Conditional random felds: Probabilistic models for segmentation and labeling sequence data"*, Proceedings eighteenth International Conference on Machine Learning (ICML '01), Morgan Kaufmann Publ. Inc, pp. 282-289, 2001.

[28]. Lu, W., Ng, H. T., *"Better punctuation prediction with dynamic conditional random felds"*, Proceedings of the 2010 conference on empirical methods in natural language processing, pp. 177-186, 2010.

[29]. Wang, W., Knight, K., Marcu, D., *"Capitalizing machine translation"*, Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pp. 1-8, 2006.

[30]. Susanto, R. H., Chieu, H. L., Lu, W., *"Learning to capitalize with character-level recurrent neural networks: an empirical study"*, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2090-2095, 2016.

[31]. Tilk, O., Alumäe, T., *"Bidirectional recurrent neural network with attention mechanism for punctuation restoration"*, Interspeech, vol. 08-12-Sept, pp. 3047-3051, doi: 10.21437/Interspeech.2016-1517, 2016.

[32]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez,

A. N., Polosukhin, I., *"Attention Is All You Need"*, Advances in neural information processing systems, pp. 5998-6008, 2017.

[33]. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. , *"Bert: Pre-training of deep bidirectional transformers for language understanding"*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, Minneapolis, Minnesota. Association for Computational Linguistics. pp. 4171-4186, 2019.

[34]. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Stoyanov, V., *"Roberta: A robustly optimized bert pretraining approach"*, International Conference on Learning Representations (ICLR), 2020.

[35]. Rei, R., Guerreiro, N. M., Batista, F., *"Automatic truecasing of video subtitles using BERT: a multilingual adaptable approach"*, Information Processing and Management of Uncertainty in Knowledge-Based Systems: 18th International Conference, pp. 708-721, Springer International Publishing, 2020.

[36]. Alam, F., Khan, T., Alam, A., *"Punctuation Restoration using Transformer Models for Resource Rich and Poor Languages"*, Proceedings Sixth Work Noisy User-generated Text, pp. 132-142, 2020.

[37]. N. Đ. Dân, *Tiếng Việt (dùng cho đại học đại cương)*. Nhà xuất bản Giáo dục, 2000.

[38]. Tran, N. L., Le, D. M., Nguyen, D. Q., *"BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese"*, Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH , pp. 1751-1755, 2022.

[39]. Grishman, R., Sundheim, B. M., *"Message understanding conference-6: A brief history"*, COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, 1996.

[40]. Aggarwal, C. C., Aggarwal, C. C., *"Mining text data"*, Springer

International Publishing, pp. 429-455, 2012.

[41]. Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O., Quintard, L., *"Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview"*, Proceedings of the 5th linguistic annotation workshop, pp. 92-100, 2011.

[42]. Yadav, H., Ghosh, S., Yu, Y., Shah, R. R., *"End-to-end Named Entity Recognition from English Speech"*, Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 4268-4272, 2020.

[43]. Cohn, I., Laish, I., Beryozkin, G., Li, G., Shafran, I., Szpektor, I., Matias, Y., *"Audio de-identification: A new entity recognition task"*, NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, Vol. 2, pp. 197-204, 2019.

[44]. Nguyen, H. T., Ngo, Q. T., Vu, L. X., Tran, V. M., Nguyen, H. T., *"VLSP shared task: Named entity recognition"*, Journal of Computer Science and Cybernetics, 34(4), pp. 283-294, 2018.

[45]. Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A., Morin, E., *"End-to-end named entity and semantic concept extraction from speech"*, IEEE Spoken Language Technology Workshop (SLT), pp. 692-699, 2018.

[46]. Kim, J. H., Woodland, P. C., *"A rule-based named entity recognition system for speech input"*, Sixth International Conference on Spoken Language Processing, 2000.

[47]. Palmer, D. D., Ostendorf, M., Burger, J. D., *"Robust information extraction from spoken language data"*, Eurospeech, 1999.

[48]. Zhai, L., Fung, P., Schwartz, R., Carpuat, M., Wu, D., *"Using n-best lists for named entity recognition from chinese speech"*, Proceedings of HLT-NAACL 2004: Short Papers, pp. 37-40, 2004.

[49]. Hatmi, M., Jacquin, C., Morin, E., Meignier, S., *"Named entity*

recognition in speech transcripts following an extended taxonomy", First Workshop on Speech, Language and Audio in Multimedia, vol. 1012, pp. 61-65, 2013.

[50]. Paaß, G., Pilz, A., Schwenninger, J., *"Named entity recognition of spoken documents using subword units"*, IEEE International Conference on Semantic Computing, pp. 529-534, doi: 10.1109/ICSC.2009.78, 2009.

[51]. Alam, F., Zanoli, R., *"A combination of classifiers for named entity recognition on transcription"*, Evaluation of Natural Language and Speech Tools for Italian: International Workshop (EVALITA), pp. 107-115, 2012.

[52]. Sudoh, K., Tsukada, H., Isozaki, H., *"Incorporating speech recognition confidence into discriminative named entity recognition of speech data"*, Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 617-624, 2006.

[53]. Li, J., Sun, A., Han, J., Li, C., *"A Survey on Deep Learning for Named Entity Recognition"*, IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 50-70, 2020v.

[54]. Porjazovski, D., Leinonen, J., Kurimo, M. , *"Named Entity Recognition for Spoken Finnish"*, Proceedings of the 2nd International Workshop on AI for Smart TV Content Production, Access and Delivery, pp. 25-29, doi: 10.1145/3422839.3423066, 2020.

[55]. Mayhew, S., Nitish, G., Roth, D., *"Robust named entity recognition with truecasing pretraining"*, Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 05, pp. 8480-8487, 2020.

[56]. Jannet, M. A. B., Galibert, O., Adda-Decker, M., Rosset, S., *"How to evaluate ASR output for named entity recognition?"*, Sixteenth Annual Conference of the International Speech Communication Association, Interspeech, vol. 2015-Janua, no. 2, pp. 1289-1293, 2015.

[57]. Chen, B., Xu, G., Wang, X., Xie, P., Zhang, M., Huang, F., *"AISHELL-*

NER: Named Entity Recognition from Chinese Speech", ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8352-8356, 2022.

[58]. *Ghannay, S., Caubriere, A., Esteve, Y., Laurent, A., Morin, E.*, "End-to-end named entity extraction from speech", *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), https://doi.org/10.48550/arXiv.1805.12045, 2018*.

[59]. Caubrière, A., Rosset, S., Estève, Y., Laurent, A., Morin, E., *"Where are we in named entity recognition from speech?"*, Proceedings of the 12th Language Resources and Evaluation Conference, pp. 4514-4520, 2020.

[60]. *Yadav, H., Ghosh, S., Yu, Y., Shah, R. R.*, "End-to-end named entity recognition from English speech", *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 4268-4272, 2020.*

[61]. Pasad, A., Wu, F., Shon, S., Livescu, K., Han, K. J., "On the use of external data for spoken named entity recognition", *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, pp. 724-737, 2022*.

[62]. Tran, P. N., Ta, V. D., Truong, Q. T., Duong, Q. V., Nguyen, T. T., Phan, X. H., *"Named entity recognition for vietnamese spoken texts and its application in smart mobile voice interaction"*, Intelligent Information and Database Systems: 8th Asian Conference, ACIIDS 2016, Da Nang, Vietnam, March 14-16, 2016, Proceedings, Part I 8, pp. 170-180, doi: 10.1007/978-3-662-49381-6_17, 2016.

[63]. Gravano, A., Jansche, M., Bacchiani, M., *"Restoring punctuation and capitalization in transcribed speech"*, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4741-4744, 2009.

[64]. Rei, R., Batista, F., Guerreiro, N. M., Coheur, L., *"Multilingual*

simultaneous sentence end and punctuation prediction", Multilingual simultaneous sentence end and punctuation prediction, 2021.

[65]. Mdhaffar, S., Duret, J., Parcollet, T., Estève, Y., *"End-to-end model for named entity recognition from speech without paired training data",* Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 4068-4072, 2022.

[66]. Caubrière, A., Tomashenko, N., Laurent, A., Morin, E., Camelin, N., Esteve, Y. , *"Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability",* Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 1198-1202, 2019.

[67]. Lugosch, L., Meyer, B. H., Nowrouzezahrai, D., Ravanelli, M., *"Using speech synthesis to train end-to-end spoken language understanding models"*, ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8499-8503, 2020.

[68]. Laptev, A., Korostik, R., Svischev, A., Andrusenko, A., Medennikov, I., Rybin, S., *"You do not need more data: Improving end_to-end speech recognition by text-to-speech data augmentation"*, 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 439-444, 2020.

[69]. Kano, T., Sakti, S., Nakamura, S., *"End-to-end speech transla_tion with transcoding by multi-task learning for distant language pairs"*, IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1342-1355, 2020.

[70]. Dey, R., Salemt, F. M., *"Gate-variants of gated recurrent unit (GRU) neural networks"*, Midwest Symposium on Circuits and Systems, pp. 1597-1600, 2017.

[71]. Jiao, Q., Zhang, S., *"A Brief Survey of Word Embedding and Its Recent*

Development", IAEAC 2021 - IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference. Institute of Electrical and Electronics Engineers Inc., pp. 1697-1701, 2021.

[72]. Devlin, J., Chang, M. W., Lee, K., Toutanova, K., *"Bert: pre-training of deep bidirectional transformers for language understanding"*, NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1, pp. 4171-4186, 2019.

[73]. Taher, E., Hoseini, S. A., Shamsfard, M., *"Beheshti-NER: Persian named entity recognition using BERT"*, Proceedings of the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019, pp. 37-42, 2019.

[74]. Gao, Y., Liu, W., Lombardi, F., "*Design and implementation of an approximate softmax layer for deep neural networks*", Proceedings IEEE International Symposium on Circuits and Systems. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/iscas45731.2020.9180870, pp. 1-5, 2020.

[75]. Gao, W., Zhao, S., Zhu, S., Ren, S., *"Research on Entity Recognition in Aerospace Engine Fields Based on Conditional Random Fields"*, Journal of Physics: Conference Series (Vol. 1848). IOP Publishing Ltd. https://doi.org/10.1088/1742-6596/1848/1/012058, 2021.

[76]. Caruana, R., *"Multitask learning"*, Springer US, pp. 95-133, 1998.

[77]. Zhang, Y., Yang, Q., *"A survey on multi-task learning"*, IEEE Transactions on Knowledge and Data Engineering, vol. 34(12), pp. 5586-5609, 2021.

[78]. Ruder, S., *"Neural transfer learning for natural language processing"*, PhD Thesis. NUI Galw., 2019.

[79]. Christensen, H., Gotoh, Y., Renals, S., *"Punctuation annotation using statistical prosody models"*, Proceedings of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 35-40, 2001.

[80]. Panchendrarajan, R., Amaresan, A., *"Bidirectional LSTM-CRF for named*

entity recognition", Proceedings of the 32nd Pacific Asia conference on language, information and computation, pp. 531-540, 2018.

[81]. Bengio, Y., Ducharme, R., Vincent, P. , *"A neural probabilistic language model"*, Advances in neural information processing systems, 13, pp. 1137-1155, 2003.

[82]. Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Auli, M., *"fairseq: A fast, extensible toolkit for sequence modeling",* NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session, pp. 48-53, 2019.

[83]. Kingma, D. P., Ba, J.*, "ADAM: a method for stochastic optimization"*, 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

[84]. Mayhew, S., Nitish, G., Roth, D., *"Robust named entity recognition with truecasing pretraining"*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8480-8487, 2020.

[85]. Sennrich, R., Haddow, B., Birch, A., *"Neural machine translation of rare words with subword units"*, 54th Annual Meeting of the Association for Computational Linguistics, ACL, Vol. 3, pp. 1715-1725, 2016.

[86]. Nguyen, K. A., Dong, N., Nguyen, C. T., *"Attentive neural net_work for named entity recognition in vietnamese"*, IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 1-6, 2019.

[87]. Pappas, N., Werlen, L. M., Henderson, J., "Beyond weight tying: Learning joint input-output embeddings for neural machine translation", WMT 2018 - 3rd Conference on Machine Translation, Proceedings of the Conference, l. 1, pp. 73-83, 2018.