ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Thị Lan

NHẬN DẠNG THỰC THỂ TRONG VĂN BẢN TIẾNG VIỆT SỬ DỤNG MÔ HÌNH HỌC SÂU SUỐT ĐỜI MỨC KÝ TỰ

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC CHÍNH QUY

Ngành: Công nghệ thông tin

HÀ NỘI – 2018

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Thị Lan

NHẬN DẠNG THỰC THỂ TRONG VĂN BẢN TIẾNG VIỆT SỬ DỤNG MÔ HÌNH HỌC SÂU SUỐT ĐỜI MỨC KÝ TỰ

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC CHÍNH QUY

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: PGS. TS. Hà Quang Thuỵ

Cán bộ đồng hướng dẫn: ThS. Trần Mai Vũ

HÀ NỘI - 2018

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Nguyen Thi Lan

NAMED ENTITY RECOGNITION IN VIETNAMESE TEXT USING CHARACTER LEVEL DEEP LIFELONG LEARNING MODEL

A THESIS PRESENTED FOR THE DEGREE BACHELOR

Major: Information and Technology

Supervisor: Assoc. Prof. Ha Quang Thuy

Co-supervisor: MSc. PhD. Tran Mai Vu

HA NOI - 2018

LỜI CAM ĐOAN

Tôi xin cam đoan các kỹ thuật sử dụng trong bài toán nhận dạng thực thể sử dụng mô hình học sâu suốt đời mức ký tự được trình bày trong khoá luận này là do tôi thực hiện dưới sự hướng dẫn của PGS.TS Hà Quang Thuỵ và ThS.Trần Mai Vũ.

Tất cả những tài liệu tham khảo từ các nghiên cứu liên quan đều được trích dẫn nguồn gốc rõ ràng từ danh mục tài liệu tham khảo của khoá luận. Trong khoá luận này, không có việc sao chép tài liệu, các công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo.

Nếu phát hiện có bất kì sự gian lận nào, tôi xin hoàn toàn chịu trách nhiệm trước hội đồng cũng như kết quả khóa luận tốt nghiệp của mình.

Hà Nội, ngày 26 tháng 04 năm 2018

Sinh viên

Nguyễn Thị Lan

LỜI CẢM ƠN

Đầu tiên, em xin bày tỏ lòng biết ơn chân thành và sâu sắc tới PGS.TS. Hà Quang Thuỵ, người đã mang đến cho em nguồn cảm hứng vô tận trong nghiên cứu khoa học. Em thật sự biết ơn những giúp đỡ, lời khuyên và sự tận tình hướng dẫn của thầy trong khóa luận cũng như định hướng nghiên cứu trong tương lai.

Em muốn gửi lời cảm ơn sâu sắc đến ThS. Trần Mai Vũ, người đã tận tình chỉ bảo, hướng dẫn, động viên và giúp đỡ em không chỉ trong quá trình thực hiện đề tài khóa luận này mà còn trong suốt quãng thời gian học tập và nghiên cứu tại Phòng Thí nghiệm và Công nghệ tri thức (DS&KT Lab) - Đại học Công nghệ, Đại học quốc gia Hà Nội.

Em xin gửi lời cảm ơn sâu sắc tới quý thầy cô giáo trong Khoa Công nghệ thông tin nói riêng và trường Đại học Công nghệ - Đại học Quốc gia Hà Nội nói chung, đã truyền đạt kiến thức quý báu cho em trong những năm tháng ngồi trên ghế nhà trường.

Em xin gửi lời cảm ơn tới các thầy cô, anh chị và các bạn trong DS&KTLab, đặc biệt là anh Nguyễn Minh Đức và chị Nguyễn Thị Cẩm Vân đã giúp đỡ em rất nhiều trong việc hỗ trợ kiến thức chuyên môn đề hoàn thành khoá luận tốt nghiệp.

Con xin nói lên lòng biết ơn vô hạn đối với bố mẹ, những người luôn luôn chăm sóc, là nguồn động viên, khích lệ con, giúp con vượt qua những khó khăn trong cuộc sống.

Cuối cùng, tôi xin gửi lời cảm ơn tới bạn bè, đặc biệt là tập thể lớp K59C-CLC đã ủng hộ, giúp đỡ tôi trong suốt quá trình học tập trên giảng đường đại học.

Tôi xin chân thành cảm ơn!

TÓM TẮT

Tóm tắt: Học máy suốt đời (Lifelong Machine Learning) hay Học suốt đời (Lifelong Learning) là một mô hình học máy tiên tiến, quá trình học được thực hiện liên tục, tích lũy tri thức đã học từ các bài toán trước đó và sử dụng các tri thức này hỗ trợ cho bài toán học trong tương lai. Bên cạnh đó, học sâu (Deep Learning) cũng là nhánh của học máy, sử dụng mạng nơron nhân tạo và các thuật toán để giải quyết các bài toán phức tạp mà các mô hình học máy truyền thống khó có thể giải quyết được. Cả học suốt đời và học sâu đều mô phỏng lại quá trình học tập, kiến trúc và hành vi bộ não người, do đó đều có thể đưa trí tuệ nhân tạo (Artificial Intelligence) ngày một gần hơn với trí thông minh của con người.

Hiện nay cũng đã có những nghiên cứu kết hợp học suốt đời với học sâu như nghiên cứu của Parisi và cộng sự (2017) về nhận diện hành động của con người, hay nghiên cứu của Chen và cộng sự (2016) trong trò chơi điện tử và đạt được những tiến bộ đáng kể. Tuy nhiên phương pháp học sâu suốt đời còn khá mới mẻ và các nỗ lực nghiên cứu sâu rộng là thực sự cần thiết cho sự phát triển trí tuệ nhân tạo.

Với mong muốn đóng góp công sức cho cộng đồng nghiên cứu, khoá luận tập trung vào việc tìm hiểu và kết hợp hai phương pháp học sâu và học suốt đời, sau đó áp dụng mô hình này vào việc giải quyết bài toán nhận dạng thực thể trong văn bản tiếng Việt. Cụ thể hơn khoá luận đã tiến hành xây dựng một mô hình học sâu suốt đời mức ký tự cho nhận dạng thực thể trong văn bản tiếng Việt. Để đánh giá mô hình, khoá luận đã tiến hành thực nghiệm trên tập dữ liệu VLSP2018, đồng thời sử dụng tập dữ liệu thu thập từ trang báo điện tử Dân trí để trích xuất đặc trưng suốt đời. Bằng thực nghiệm, khoá luận đã thu được những kết quả khả quan ban đầu qua đó chứng minh được tính hiệu quả của mô hình đề xuất.

Từ khoá: *học sâu, học suốt đời, nhận dạng thực thể.*

ABSTRACT

Abstract: Lifelong machine learning (LML) or lifelong learning is an advanced machine learning paradigm that learns continuously, accumulates the knowledge learned in previous tasks, and uses it to help future learning. In the process, the learner becomes more and more knowledgeable and effective at learning. This is learning ability is one of the hallmarks of human intelligence. In addition, Deep learning is also a branch of machine learning, using artificial intelligence and algorithms to resolve complex tasks that traditional machine learning models can not resolve. Even LML and Deep learning reproceduce the learning process, architecture and behavior of the brain, so that they can bring Arfiticial intelligence closed to human intelligence.

There are now reseachs that combine LML and deep learning such as Human action recognition (Parisi, et al, 2017), video game (Chen, et al 2016) and achived . Although significant advances have been made in domain-specific continual lifelong learning with neural networks, this method is quite novel and extensive research efforts are required for the development of artificial intelligence.

With the desire to contribute to the research community, this thesis focuses on understanding and combining deep learning and lifelong machine learning then applying the model on Named entity recognition in Vietnamese text. Thesis has conducted a character level deep lifelong learning model for Named entity recognition in Vietnamese text and experiments on VLSP2018 dataset and use the collected dataset from Dantri for lifelong extraction. The effective of the model was demonstrated by the experiments and achived positive results.

Keywords: *deep learning, lifelong learning, named entity recognization.*

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Từ viết tắt	Cụm từ tiếng Anh	Cụm từ tiếng Việt
1	Bi-LSTM	Bi-directional Long-Short Term Memory	Bộ nhớ dài ngắn 2 chiều
2	CNN	Convolutional Neural Network	Mạng nơron tích chập
3	CRF	Conditional Random Fields	Trường điều kiện ngẫu nhiên
4	LML	Lifelong Machine Learning	Học máy suốt đời
5	LSTM	Long-Short Term Memory	Bộ nhớ dài ngắn
6	ML	Machine Learning	Học máy
7	NER	Named Entity Recognition	Nhận dạng thực thể
8	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
9	POS	Part-of-speech	Từ loại
10	RNN	Recurrent Neural Network	Mạng nơron hồi quy

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG

MỞ ĐẦU

Học máy (Machine Learning - ML) đã trở thành công cụ cho những tiến bộ của phân tích dữ liệu và trí tuệ nhân tạo (Artificial Intelligence). Những thành công gần đây của học sâu (Deep Learning) đã đưa nó lên một tầm cao mới. Các thuật toán ML được sử dụng trong hầu hết lĩnh vực về khoa học máy tính và nhiều lĩnh vực khoa học tự nhiên, kỹ thuật và khoa học xã hội. Thậm chí các ứng dụng thực tế của học máy còn phổ biến hơn. Có thể nói rằng nếu không có các thuật toán ML hiệu quả, nhiều ngành công nghiệp sẽ không phát triển mạnh, ví dụ như thương mại điện tử và tìm kiếm Web. Tuy nhiên, đối với phương pháp học máy giám sát thường cần một lượng lớn các ví dụ huấn luyện, do đó việc gán nhãn dữ liệu huấn luyện thường được thực hiện bằng tay là rất tốn kém và mất thời gian. Hơn nữa, dữ liệu trên Internet ngày càng lớn và luôn luôn thay đổi và việc gán nhãn như vậy cần được thực hiện liên tục. Ngay cả đối với học không giám sát, việc thu thập một khối lượng dữ liệu lớn có thể không khả thi trong nhiều trường hợp. Bởi vậy các hệ thống hay các tác nhân luôn cần phải tự học, ghi nhớ nhiều tác vụ và có khả năng tinh chỉnh, chuyển giao kiến thức trong thời gian dài. Khả năng học tập liên tục gọi là học suốt đời. Học máy suốt đời (Lifelong machine learning - LML) (hay đơn giản là học suốt đời) nhằm bắt chước quá trình và khả năng học của con người, tích luỹ và duy trì tri thức đã học được từ các bài toán trước và không ngừng sử dụng tri thức đó để học và giải quyết bài toán mới. Tuy nhiên nhiệm vụ học liên tục là một thách thức lâu dài đối với học máy và mạng nơron và sự phát triển của các hệ thống trí tuệ nhân tạo.

Nhận dạng thực thể (Named Entity Recognization - NER) là một bài toán con trong bài toán trích xuất thông tin, thuộc lĩnh vực xử lý ngôn ngữ tự nhiên và thường được giải quyết bằng các kỹ thuật học máy và đặc biệt là học sâu. Tuy là bài toán cơ bản, nhưng NER được coi như một tác vụ tiền đề cho các bài toán phức tạp hơn trong trích xuất thông tin như trích xuất quan hệ hay trích xuất sự kiện. Các nghiên cứu gần đây **Error! Reference source not found.**[9][11] đã cho thấy nhận dạng thực thể sử dụng học sâu trong miền có giám sát đang đạt được những kết quả khả quan. Bên cạnh đó, đã có một vài nghiên cứu về việc kết hợp học suốt đời và học sâu trong các bài toán khác như: nhận diện hành động của con người [9], nhận diện hình ảnh[12], phân lớp văn bản [13] hay trong lĩnh vực y sinh học [8], tuy nhiên các nghiên cứu sử dụng học suốt đời trong bài toán gán nhãn chuỗi vẫn chỉ dừng lại ở các phương pháp không sử dụng học sâu và hiện chưa có nghiên cứu cụ thể nào cho bài toán NER. Do đó, sự kết hợp giữa học suốt đời và

học sâu mở ra một hướng nghiên cứu mới và mang tính đột phá trong bài toán NER nói chung và bài toán NER trong ngôn ngữ tiếng Việt nói riêng.

Mục tiêu của khoá luận là khảo sát, nghiên cứu để đưa ra một mô hình học sâu suốt đời mức ký tự cho nhận dạng thực thể trong văn bản tiếng Việt. Để tiếp cận mục tiêu này, khoá luận nghiên cứu và giới thiệu các phương pháp học sâu và học học suốt đời đã tồn tại có liên quan trực tiếp tới nhận dạng dạng thực thể. Từ đó, khoá luận đề xuất một mô hình nhận dạng thực thể sử dụng mạng bộ nhớ dài ngắn kết hợp với trường điều kiện ngẫu nhiên đồng thời lưu giữ và chuyển giao kiến thức từ các bài toán cũ sang bài toán mới.

Nội dung của khoá luận được chia thành các chương như sau:

Chương 1: Chương này sẽ trình bày một số kiến thức cơ bản và kỹ thuật nổi bật của hai phương pháp học sâu và học suốt đời đồng thời trình bày về bài toán nhận dạng thực thể trong văn bản tiếng Việt của khoá luận

Chương 2: Chương này sẽ trình bày một số mô hình đã tồn tại để giải quyết bài toán theo phương pháp học sâu và học suốt đời. Cụ thể, đối với phương pháp học sâu, khóa luận sẽ trình bày mô hình Bi-LSTM+CRF cho nhận dạng thực thể trong tiếng Việt và mô hình trích xuất khía cạnh suốt đời đối với phương pháp học suốt đời.

Chương 3: Chương này sẽ trình bày kiến trúc và các pha của mô hình học sâu suốt đời mức ký tự cho nhận dạng thực thể trong văn bản tiếng Việt mà khoá luận đề xuất.

Chương 4: Chương này sẽ mô tả về dữ liệu thực nghiệm, cụ thể là tập dữ liệu VLSP2018 và tập dữ liệu chưa gán nhãn thu thập từ trang báo điện tử Dân trí, các tham số thực nghiệm, môi trường và kết quả thực nghiệm của khoá luận.

Phần kết luận và hướng phát triển của khoá luận: Tóm lược những điểm chính của khoá luận. Chỉ ra những điểm chưa làm được và những hạn chế cần khắc phục, đồng thời đưa ra những hướng nghiên cứu trong thời gian sắp tới.

CHƯƠNG 1: ĐẶT VẤN ĐỀ VÀ PHÁT BIỂU BÀI TOÁN

1.1. Giới thiệu về học sâu

1.1.1. Giới thiệu chung

Học sâu (Deep Learning) là phạm trù nhỏ của học máy (Machine Learning - ML) dựa trên việc sử dụng mạng nơron và một tập hợp các thuật toán để mô hình hoá dữ liệu ở các mức trừu tượng khác nhau, qua đó giải quyết được nhiều bài toán mà các mô hình học không sâu truyền thống khó có thể giải quyết được như thị giác máy tính, nhận diện giọng nói, xử lý ngôn ngữ tự nhiên, nhận dạng âm thanh ngôn ngữ và tin sinh học.

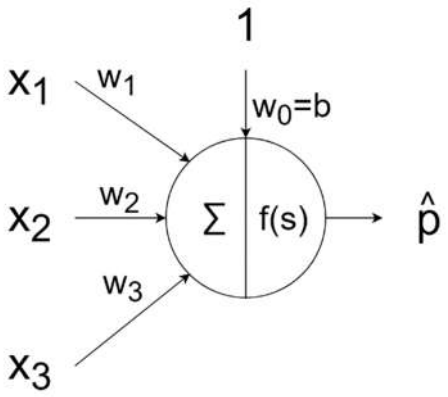
Các mô hình học sâu có thể đạt được độ chính xác cao, đôi khi vượt quá hiệu suất của con người. Các mô hình được huấn luyện bằng cách sử dụng một bộ dữ liệu có nhãn và các cấu trúc mạng thần kinh có nhiều lớp. Các mô hình học sâu không chỉ có khả năng mở rộng mạng nơron mà còn có cả tính năng học tập – khai thác các đặc trưng tự động từ dữ liệu thô, nên nó đòi hỏi số lượng lớn dữ liệu có nhãn và sức mạnh tính toán đáng kể.

Kiến trúc cơ bản của học sâu là mạng nơron nhân tạo và có rất nhiều biến thể từ chúng, hầu hết là các nhánh sinh ra từ kiến trúc ban đầu như: mang nơron sâu (Deep Neural Network), mạng niềm tin sâu (Deep Belief Network), Mạng nơron tích chập (Convolutional neural networks - CNN), mạng niềm tin sâu tích chập (Convolutional Deep Belief Network), mạng nơron lưu trữ và truy xuất bộ nhớ lớn (Large Memory Storage And Retrieval Neural Network), các máy Deep Boltzmann,…

1.1.2. Mạng nơron nhân tạo

Mạng nơron nhân tạo là một mô hình toán học được xây dựng để mô phỏng lại kiến trúc và hành vi của nơron sinh học trong não người. Nó là một hệ thống các nơron nhân tạo nối với nhau thành các lớp và xử lý thông tin bằng cách truyền theo các kết nối giữa các nơron.

Để dễ dàng giải thích các thành phần của mạng nơron, tôi sẽ lấy ví dụ về một mạng nơron đơn giản là mạng perceptron đơn (xem Hình 1.1) do Rosenblatt đưa ra vào năm 1957. Kiến trúc và hành vi của perceptron rất giống với nơron sinh học và thường được coi là dạng cơ bản nhất của mạng nơron. Các loại mạng nơron khác đã được phát triển dựa trên perceptron, và chúng vẫn đang tiếp tục phát triển cho tới hiện nay.



Hình 1.1: Mạng perceptron đơn

a) Nơron

Tương tự như kiến trúc và hành vi của nơron sinh học, một nơron nói chung và một perceptron nói riêng có các đầu vào và các đầu ra. Thông tin từ đầu vào đi qua nơron sẽ được biến đổi, sau đó đi ra tại đầu ra. Nói cách khác, một nơron là một tập hợp các hàm biến đổi toán học để biến đổi đầu vào thành đầu ra mong muốn. Trong ví dụ trên, mạng perceptron đơn được cấu tạo từ một perceptron duy nhất, sử dụng hàm tính tổng và một hàm phi tuyến f, hoạt động như một bộ phân lớp nhị phân với đầu vào là một vectơ đặc trưng $[x_1, x_2, x_3]$ và đầu ra là xác suất p của một sự kiện nhất định.

b) Trọng số

Mỗi đầu vào trong vectơ đặc trưng được gán với một trọng số tương đối (w) thể hiện ảnh hưởng của nó đối với hàm tính tổng. Trong số các đầu vào, một số cái có ảnh hưởng lớn hơn sẽ có trọng số lớn hơn, ngược lại thì trọng số sẽ nhỏ hơn. Độ lệch $w_o = b$ cũng được tính vào tổng như một trọng số. Giá trị tổng s được tính như sau:

$$s = w_0 + w_1x_1 + w_2x_2 + w_3x_3 = [w \quad b][x \quad 1]^T$$

c) Hàm kích hoạt

Kết quả của hàm tính tổng được biến đổi thành một đầu ra mong muốn bằng cách sử dụng một hàm phi tuyến f (non-linear function), còn gọi là hàm kích hoạt. Bảng 1.1 dưới đây liệt kê một số hàm kích hoạt thường gặp.

Bảng 1.1: Một số hàm kích hoạt thường gặp

Hàm kích hoạt	Công thức	Khoảng giá trị
Identity	$f(x) = x$	$(-\infty, +\infty)$
Logistic (Sigmoid)	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$(0,1)$
TanH	$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	$(-1,1)$
Rectified linear unit (ReLU)	$f(x) = \begin{cases} 0\ (x < 0) \\ x\ (x \geq 0) \end{cases} = \max(0, x)$	$[0, +\infty)$
Softmax	$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}, i \in [1, J]$	$(0,1)$

Vì đầu ra mong muốn trong trường hợp này là xác suất của một sự kiện, ta có thể sử dụng hàm sigmoid để giới hạn giá trị tổng s trong khoảng $(0,1)$

$$\hat{p} = f(s)$$

1.1.3. Các thuật toán huấn luyện

Như đã đề cập, bên cạnh mạng nơron, một mô hình học sâu cần có các thuật toán để huấn luyện mạng nơron đó.

a) Sai số và hàm mất mát

Trong hầu hết các mạng nơron, sai số (error) được tính toán bằng hiệu giữa đầu ra mong muốn và đầu ra dự đoán.

$$J(w) = p - \hat{p}$$

Hàm được sử dụng để tính sai số được gọi là hàm mất mát (loss function) $J(.)$. Hàm mất mát khác nhau sẽ cho ra sai số khác nhau trên cùng một dự đoán của mô hình, do đó nó có ảnh hưởng tới hiệu năng của mô hình. Một trong những hàm mất mát được dùng rộng rãi nhất là hàm trung bình của sai số bình phương. Hàm mất mát sẽ được chọn tuỳ vào từng bài toán.

b) Lan truyền ngược và hàm tối ưu hoá

Sai số $J(w)$ là một hàm với đầu vào là các tham số nội mô hình (các trọng số và độ lệch). Để dự đoán chính xác, ta cần giảm thiểu sai số, tức tìm w để $J(w)$ đạt giá trị cực tiểu. Trong mạng nơron, điều này được thực hiện bằng lan truyền ngược. Sai số tại lớp hiện tại thường được truyền ngược lại lớp trước đó để thay đổi các trọng số và độ lệch sao cho sai số giảm đi. Các trọng số được thay đổi bằng cách sử dụng một hàm gọi là hàm tối ưu hoá.

Các hàm tối ưu hoá thường tính độ dốc (gradient), tức là tính đạo hàm riêng của hàm mất mát đối với trọng số, và trọng số được thay đổi theo hướng ngược lại của độ dốc tính được. Việc này được lặp lại cho đến khi chúng ta đạt đến giá trị cực tiểu của hàm mất mát.

$$W^{(k+1)} = W^{(k)} - \frac{\partial}{\partial W^{(k)}} J(W)$$

1.1.4. Một số mô hình mạng nơron điển hình

Việc xây dựng mạng nơron chỉ dựa trên perceptron sẽ khiến số lượng trọng số (weight) của mô hình trở nên rất lớp, giữa hai lớp có n và m nơron sẽ tồn tại $n * m$ kết nối giữa các nơron. Bên cạnh đó, các nơron trong cùng một lớp nơron lại không hề có kết nối. Do vậy, sau này các nhà nghiên cứu đã tạo ra một số mô hình mạng nơron để giải quyết những vấn đề này.

a) Mạng nơron tích chập

Mạng nơron tích chập (Convolutional Neural Network – CNN) là một tập hợp các lớp tích chập (Convolutional layer), thường được sử dụng để nắm bắt các đặc trưng ở mức cụm n từ (n-gram).

Các lớp tích chập hoạt động như sau. Đầu vào là các câu x dưới dạng một vectơ $x = \{w_1, w_2, \dots, w_m\}$, $w_i \in \mathbb{R}^d$, giả sử l là kích thước cửa sổ của nơron trong lớp tích chập (hay còn gọi là nhân tích chập – convolutional kernel) thì vectơ của cửa sổ thứ i ($q_i \in \mathbb{R}^{d \times l}$) được tính bằng cách nối các vectơ đầu vào trong cửa sổ đó,,

$$q_i = w_{i:i+l-1}; (1 \le i \le m - l + 1) \quad (1)$$

Một nhân tích chập đơn có thể bao gồm một vectơ trọng số $W \in \mathbb{R}^{d \times l}$ và một độ lệch (bias) $b \in \mathbb{R}$, và đầu ra của cửa sổ thứ i có công thức:

$$p_i = f(W' q_i + b)$$

trong đó f là hàm kích hoạt (activation function). Đầu ra của nhân tích chập p sẽ có dạng $p \in \mathbb{R}^{m-l+1}$. Một lớp tích chập có thể bao gồm d_c nhân tích chập, khiến đầu ra của lớp tích chập có dạng $\mathbb{R}^{d_c \times (m-l+1)}$.

b) Mạng nơron hồi quy

Mạng nơron hồi quy (Recurrent Neural Network - RNN) có thể xử lý các chuỗi đầu vào có độ dài tuỳ ý thông qua ứng dụng đệ quy (recursive application) của một hàm chuyển tiếp trên một vectơ trạng thái ẩn h_t.

Tại thời điểm t, trạng thái ẩn h_t là một hàm của vectơ đầu vào x_t mà mạng nhận được tại thời điểm t và trạng thái ẩn trước đó của nó là h_{t-1}. Ví dụ, vectơ đầu vào x_t có thể là vectơ đại diện của từ thứ t trong câu. Trạng thái ẩn $h_t \in \mathbb{R}^d$ có thể hiểu như là một biểu diễn phân tán d chiều của chuỗi các dấu hiệu quan sát được đến thời điểm t.

Thông thường, hàm chuyển tiếp của RNN là một chuyển tiếp toàn vẹn (affine transformation) theo sau bởi một phi tuyến rời rạc (pointwise non-linearity) như hàm tiếp tuyến hyperbol

$$h_t = \tanh(W x_t + U h_{t-1} + b)$$

Thật không may, một vấn đề với RNN với các hàm chuyển tiếp dưới dạng này là trong quá trình huấn luyện, các thành phần của vectơ gradient có thể phát triển hoặc phân rã theo cấp số mũ trên các chuỗi dài. Vấn đề bùng nổ hoặc biến mất gradient làm cho mô hình RNN khó có thể học các tương quan có khoảng cách lớn trong một chuỗi.

c) Mạng bộ nhớ dài ngắn

Kiến trúc bộ nhớ dài-ngắn (Long-Short Term Memory – LSTM) **Error! Reference source not found.** giải quyết vấn đề học phụ thuộc lâu dài bằng cách giới thiệu một tế bào nhớ có khả năng bảo toàn trạng thái trong một thời gian dài. Trong khi nhiều biến thể LSTM đã được mô tả, khóa luận sẽ mô tả phiên bản được sử dụng bởi Tai et al.(2015)[15].

Ta định nghĩa đơn vị (unit) LSTM tại mỗi thời điểm t là một tập các vectơ trong $\mathbb{R}^d$: một cổng vào (input gate) i_t, một cổng quên (forget gate) f_t, một cổng ra (output gate) o_t, một tế bào nhớ (memory cell) c_t và một trạng thái ẩn h_t. Các đầu vào của các

vectơ cổng i_t, f_t và o_t có giá trị trong đoạn [0,1]. Ta gọi d là chiều nhớ (memory dimension) của LSTM.

Các phương trình chuyển tiếp của LSTM như sau:

$$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right), \quad (1)$$

$$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right),$$

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right),$$

$$u_t = \tanh\left(W^{(u)}x_t + U^{(u)}h^{t-1} + b^{(u)}\right),$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1},$$

$$h_t = o_t \odot \tanh(c_t),$$

trong đó x_t là đầu vào tại thời điểm hiện tại, σ biểu thị hàm logistic sigmoid và $\odot$ biểu thị phép nhân các phần tử. Một cách trực quan, cổng quên điều khiển mức độ mà các tế bào nhớ trước đó bị lãng quên, cổng vào kiểm soát mỗi đơn vị được cập nhật bao nhiêu, và cổng ra kiểm soát sự thể hiện ra ngoài của trạng thái bộ nhớ trong. Vì thế, vectơ trạng thái ẩn trong một đơn vị LSTM phản ánh một phần trạng thái của tế bào nhớ trong của đơn vị. Vì giá trị của các biến cổng thay đổi cho mỗi phần tử vectơ nên mô hình có thể học để biểu diễn thông tin trên nhiều khoảng thời gian.

Bộ nhớ dài-ngắn hai chiều (Bi-directional LSTM – Bi-LSTM)[15] là một biến thể của kiến trúc LSTM cơ bản. Bi-LSTM bao gồm hai LSTM chạy song song: một trên chuỗi đầu vào và một trên nghịch đảo của chuỗi đầu vào. Tại mỗi thời điểm, trạng thái ẩn của Bi-LSTM được nối từ các trạng thái thái ẩn phía trước và phía sau. Thiết lập này cho phép trạng thái ẩn nắm bắt cả thông tin trong quá khứ lẫn tương lai.

Bộ nhớ dài-ngắn nhiều lớp (Multilayer LSTM)[15]: Trong kiến trúc bộ nhớ dài ngắn nhiều lớp, trạng thái ẩn của một đơn vị LSTM trong lớp l được sử dụng như đầu vào của lớp LSTM $l + 1$ trong cùng thời điểm. Ở đây, ý tưởng này để cho các lớp cao hơn nắm bắt các phụ thuộc dài hơn của chuỗi đầu vào.

1.2. Giới thiệu về học suốt đời

1.2.1. Tổng quan về học suốt đời

a) Định nghĩa

Khái niệm học suốt đời (LML) được Thrun và Mitchell [1995] đề xuất vào khoảng năm 1995 và định nghĩa đầu tiên của LML [16] được phát biểu như sau: Cho một hệ thống đã thực hiện N bài toán. Khi đối mặt với bài toán thứ $N+1$, nó sử dụng tri thức thu được từ N bài toán để trợ giúp bài toán $N+1$. Sau đó, Chen và Liu [2] mở rộng định nghĩa này bằng cách bổ sung thêm một cơ sở tri thức (Knowledge base: KB) để nhấn mạnh tầm quan trọng của việc tích lũy tri thức và chuyển đổi các tri thức mức độ cao hơn được thêm vào từ tri thức thu được trong quá trình học trước đó.

Định nghĩa: *Học máy suốt đời* (*Lifelong Machine Learning*: LML) là một quá trình học liên tục. Tại thời điểm bất kỳ, bộ học đã thực hiện một chuỗi *N* bài toán học, $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_N$. Các bài toán này, còn được gọi là các bài toán trước (*previous tasks*) có các tập dữ liệu tương ứng là $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N$. Các bài toán có thể cùng kiểu hoặc thuộc các kiểu khác nhau và từ cùng một miền ứng dụng hoặc các miền ứng dụng khác nhau. Khi đối mặt với bài toán thứ *N*+1, $\mathcal{T}_{N+1}$ (được gọi là bài toán mới hoặc bài toán hiện tại) với dữ liệu $\mathcal{D}_{N+1}$, bộ học có thể tận dụng tri thức quá khứ trong cơ sở tri thức (KB) để hỗ trợ học bài toán $\mathcal{T}_{N+1}$

Mục tiêu của LML thường là tối ưu hóa hiệu năng của bài toán mới $\mathcal{T}_{N+1}$, song nó có thể tối ưu hóa bất kỳ bài toán nào bằng cách xử lý các bài toán còn lại như các bài toán trước đó. Cơ sở tri thức (KB) duy trì tri thức đã được học và được tích lũy từ việc học các bài toán trước đó. Sau khi hoàn thành bài toán học $\mathcal{T}_{N+1}$, tri thức được cập nhật vào KB (chẳng hạn, kết quả trung gian cũng như các kết quả cuối cùng) thu được từ bài toán học $\mathcal{T}_{N+1}$. Việc cập nhật tri thức có thể bao gồm liên quan đến kiểm tra tính nhất quán, lập luận và biến đổi của tri thức mức cao bổ sung vào KB.

b) Đặc điểm

LML có 3 đặc điểm chính: (1) Quá trình học liên tục, (2) Tích luỹ và lưu giữ tri thức trong cơ sở tri thức (KB), (3) Khả năng sử dụng các tri thức đã học trước đó để xử lý các bài toán mới.

Kiến trúc hệ thống học máy suốt đời được mô tả trong Hình 1.2 bao gồm 4 phần: Bộ quản lý bài toán (Task management), Cơ sở tri thức (Knowledge Base - KB), Bộ học dựa trên tri thức (Knowledge Base Learner - KBL) và Đầu ra (Output).

Bộ quản lý bài toán (**Task management**): Nhận và quản lý các bài toán xuất hiện trong hệ thống. Xử lý sự chuyển bài toán và trình bày bài toán học mới cho bộ học (KBL) theo phương pháp học suốt đời.

Cơ sở tri thức (Knowledge Base - KB): Lưu giữ lại các tri thức đã học được, gồm các thành phần:

Kho thông tin quá khứ (Past Information Store - PIS): Lưu trữ thông tin đã học trong quá khứ, bao gồm các mô hình kết quả, mẫu hoặc các dạng kết quả, PIS cũng có thể bao gồm các kho con chứa các thông tin như (1) dữ liệu ban đầu được sử dụng trong mỗi bài toán trước đó, (2) các kết quả trung gian từ mỗi bài toán trước, và (3) mô hình kết quả hoặc các mẫu học được từ mỗi bài toán trước đó. Những thông tin hoặc tri thức nào nên được giữ lại phụ thuộc vào bài toán học và thuật toán học. Trong một hệ thống cụ thể, người sử dụng cần quyết định những gì cần giữ lại để trợ giúp việc học trong tương lai.

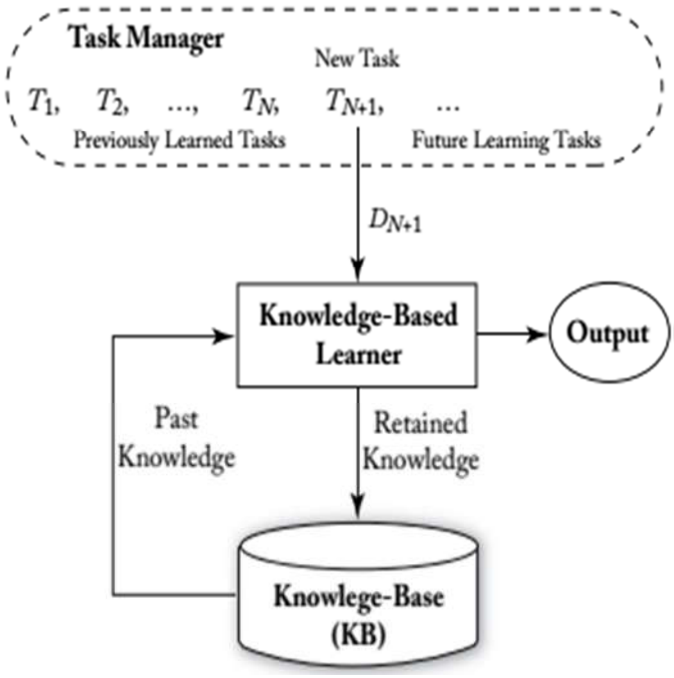
Kho tri thức (Knowledge Store - KS): Lưu trữ kiến thức được khai thác hoặc củng cố, tổng hợp từ PIS.

Bộ khai phá tri thức (Knowledge Miner - KM) : Khai thác dữ liệu từ PIS, Kết quả được lưu ở KS.

Bộ suy luận tri thức (Knowledge Resoner - KR): Suy luận dựa trên tri thức trong KB và PIS để tạo thêm tri thức bổ sung.

Bộ học dựa trên tri thức (Knowledge Base Learner - KBL): Nhận kiến thức từ KS, Bộ học của LML có thể tận dụng kiến thức và thông tin trong PIS để học bài toán mới.

Đầu ra (Output): Đây là kết quả học của người dùng, có thể là một mô hình dự báo hoặc bộ phân lớp trong học giám sát, các cụm hoặc chủ đề trong học không giám sát, chính sách trong học tăng cường,…



Hình 1.2: Kiến trúc chung của hệ thống học suốt đời [2]

c) Khó khăn

Đối với học máy suốt đời, việc giữ lại tri thức nào, cách sử dụng tri thức trước đây và cách duy trì cơ sở tri thức (KB) là các bài toán khó cần được giải quyết; đây chính là một thách thức rất lớn của LML. Dưới đây là 2 thách thức tiềm ẩn nhưng cơ bản của LML:

- *Tính chính xác của tri thức*: Tri thức sai rất bất lợi cho việc học mới. LML có thể được xem như là một quá trình khởi động (bootstrapping) liên tục. Lỗi có thể lan truyền từ các bài toán trước sang các bài toán sau tạo ra ngày càng nhiều lỗi hơn. Nhưng chúng ta dường như có ý tưởng tốt về những gì đúng hoặc những gì là sai.
- *Khả năng áp dụng tri thức*: Mặc dù một mẩu tri thức có thể đúng trong ngữ cảnh của một số bài toán trước đây, nhưng nó có thể không áp dụng được cho bài toán hiện tại. Việc áp dụng tri thức không thích hợp có hệ quả tiêu cực như trường hợp trên.

d) Phương pháp đánh giá

Trong học riêng biệt (cô lập) cổ điển, một thuật toán học được đánh giá dựa trên việc sử dụng dữ liệu từ cùng một miền của bài toán để huấn luyện và kiểm thử, LML đòi

hỏi một phương pháp đánh giá khác vì nó liên quan đến một dãy bài toán và chúng ta muốn thấy những cải tiến trong việc học của các bài toán mới. Đánh giá thử nghiệm một thuật toán LML trong nghiên cứu hiện nay thường được thực hiện bằng cách sử dụng các bước sau đây:

- *Chạy trên dữ liệu của các bài toán trước:* Đầu tiên, chúng ta chạy thuật toán trên dữ liệu của một tập các bài toán trước, mỗi lần thực hiện trên dữ liệu của một bài toán của dãy và giữ lại tri thức thu được ở cơ sở tri thức (KB).
- *Chạy trên dữ liệu của bài toán mới*: Chúng ta chạy thuật toán trên dữ liệu của bài toán mới bằng cách tận dụng tri thức trong Knowledge Base (tri thức tiên nghiệm thu được từ bước 1).
- *Chạy các thuật toán cơ sở*: Trong bước này, chúng ta lựa chọn một số thuật toán cơ sở để thực nghiệm; mục tiêu của bước này là so sánh kết quả được thực hiện bởi thuật toán LML với các thuật toán cơ sở. Thông thường có hai kiểu thuật toán cơ sở. (1) Các thuật toán học thực hiện riêng biệt trên dữ liệu mới không sử dụng bất kỳ tri thức quá khứ nào, và (2) các thuật toán LML hiện có.
- *Phân tích các kết quả*: Bước này so sánh các kết quả thực nghiệm của bước 2, bước 3 và phân tích các kết quả để đưa ra một số nhận xét, chẳng hạn như cần cho thấy các kết quả thực hiện của thuật toán LML trong bước 2 có tốt hơn các kết quả thực hiện từ các thuật toán cơ sở trong bước 3 hay không.

1.2.2. Phương pháp học giám sát suốt đời

Định nghĩa: Học giám sát suốt đời là một quá trình học liên tục mà bộ học đã thực hiện một chuỗi các bài toán học giám sát, $\mathcal{T}_1, \mathcal{T}_2, \dots \mathcal{T}_N$, và giữ lại tri thức đã học được trong cơ sở tri thức (KB). Khi một bài toán mới $\mathcal{T}_{N+1}$ đến, bộ học sử dụng tri thức quá khứ trong KB để giúp học một mô hình mới f_{N+1} từ dữ liệu huấn luyện $\mathcal{D}_{N+1}$ của $\mathcal{T}_{N+1}$. Sau khi học $\mathcal{T}_{N+1}$, KB cũng được cập nhật các tri thức đã học được từ $\mathcal{T}_{N+1}$.

Học giám sát suốt đời bắt đầu từ bài báo của Thrun [1996b] với đề xuất một vài phương pháp LML ban đầu trong ngữ cảnh học theo ghi nhớ (memory-based learning) và mạng nơron. Cách tiếp cận mạng nơron đã được Silver và Mercer [1996, 2002], Silver và cộng sự [2015] cải tiến. Trong các bài báo này, mỗi bài toán mới tập trung vào việc học một khái niệm hoặc lớp mới. Mục tiêu của LML là tận dụng các dữ liệu trong quá khứ để giúp xây dựng một phân lớp nhị phân để xác định các thể hiện của lớp mới này. Trong

công trình của Fei và cộng sự [2016], một hình thức đặc biệt của LML được gọi là học tích luỹ được đề xuất.

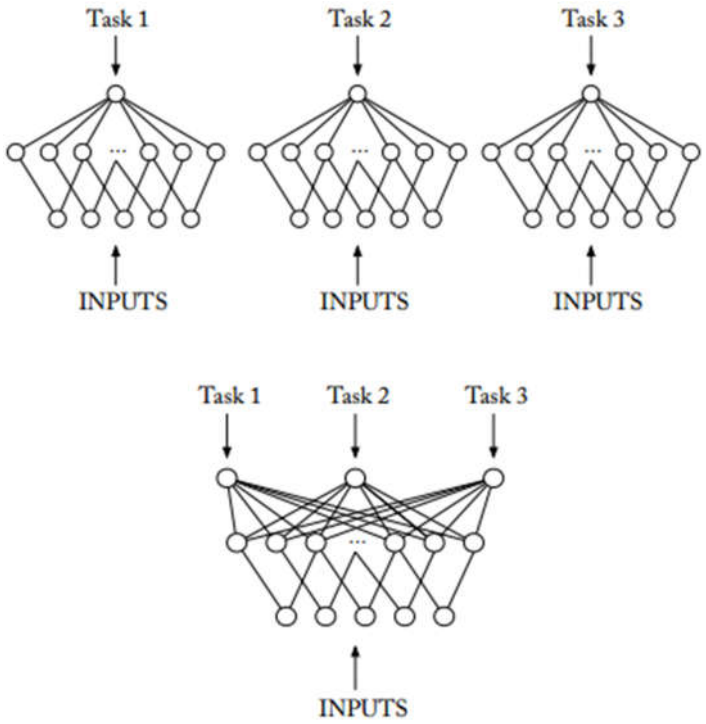
1.2.3. Mạng nơron suốt đời

Trong cuốn sách "Lifelong machine learning" của Chen và Bing Liu [2] có đề cập tới hai phương pháp tiếp cận mạng nơron ban đầu để học giám sát suốt đời. Dưới đây khoá luận sẽ trình bày cụ thể hai phương pháp này.

a) Mạng MTL (Học đa nhiệm với mạng nơron)

Mặc dù học đa nhiệm với Mạng nơron (*Multi-task learning with neural network: MTL net*) [1] được mô tả như là một phương thức học suốt đời mà Thrun trình bày trong công trình nghiên cứu năm 1996 [16], nó thực sự là một phương pháp học đa nhiệm theo lô (batch multi-task learning). Dựa trên định nghĩa của Bing Liu về học suốt đời, chúng là những mô hình học khác nhau.

Trong mạng MTL, thay vì xây dựng một mạng nơron cho mỗi bài toán riêng lẻ, nó xây dựng một mạng nơron tổng thể cho mọi bài toán (Hình 1.3). Mạng nơron tổng thể này sử dụng cùng một tầng đầu vào để làm đầu vào cho mọi bài toán và sử dụng một đơn vị đầu ra cho mỗi bài toán (hoặc lớp trong trường hợp này). Ngoài ra còn có một tầng ẩn dùng chung trong mạng MTL được huấn luyện song song bằng cách sử dụng lan truyền ngược (Back-Propagation trên mọi bài toán để giảm thiểu các lỗi của mọi bài toán. Tầng chia sẻ này cho phép các đặc trưng của một bài toán phát triển (mở rộng) được các bài toán khác sử dụng. Vì vậy, một số đặc trưng phát triển có thể đại diện cho các đặc điểm chung của các bài toán. Đối với một bài toán cụ thể, nó sẽ khởi động (kích hoạt) một số đơn vị ẩn có liên quan đến nó trong khi làm cho trọng số của các đơn vị ẩn khác không liên quan nhỏ đi. Về bản chất, giống như phương pháp học đa nhiệm theo lô thông thường, hệ thống sẽ tối ưu hóa đồng thời việc phân lớp mọi bài toán gồm bài toán quá khứ/bài toán trước đó và bài toán hiện tại/bài toán mới.



Hình 1.3: Các mạng nơron hàng trên được huấn luyện độc lập cho mỗi bài toán, và mạng nơron hàng dưới là mạng MTL của Caruana [1].

b) Mạng nơron dựa trên sự giải thích

Cách tiếp cận học suốt đời này trong ngữ cảnh của Mạng nơron dựa trên sự giải thích *(Explanation-Based Neural Network: EBNN)* của Thrun [16], một lần nữa thúc đẩy dữ liệu bài toán trước đó (hoặc tập hỗ trợ) để cải thiện việc học. Khái niệm *học* là mục tiêu của bài toán này, trong đó học một hàm $f: I \rightarrow \{0, 1\}$ để dự đoán nếu một đối tượng được biểu diễn bởi một vectơ đặc trưng $x \in I$ có thuộc về một khái niệm ($y = 1$) hay là không ($y = 0$).

Trong cách tiếp cận này, (1) đầu tiên hệ thống học một *hàm khoảng cách tổng quát*, $d: I \times I' \rightarrow [0, 1]$ xem xét tất cả dữ liệu quá khứ (hoặc tập hỗ trợ) và (2) sử dụng hàm khoảng cách này để chia sẻ hoặc chuyển tri thức của dữ liệu bài toán quá khứ thành bài toán mới $\mathcal{T}_{N+1}$. Cho hai vectơ đầu vào, gọi là x và x', hàm d tính xác suất của x và x' là các bộ phận của cùng một khái niệm (hoặc lớp), bất kể khái niệm là gì. Trong Thrun [1996b], d được học bằng cách sử dụng một mạng nơron được huấn luyện bằng lan truyền ngược. Dữ liệu huấn luyện để học hàm khoảng cách được tạo ra như sau: Đối với mỗi dữ liệu bài toán quá khứ $\mathcal{D}_i \in \mathcal{D}^p$, từng cặp ví dụ của khái niệm tạo ra một ví dụ huấn luyện tích cực hoặc tiêu cực.

Với hàm khoảng cách đã học, EBNN hoạt động như sau: EBNN ước tính độ nghiêng (đường tiếp tuyến) của hàm đích tại mỗi điểm dữ liệu x và thêm nó vào véc-tơ biểu diễn của điểm dữ liệu. Trong bài toán mới $\mathcal{T}_{N+1}$, một ví dụ huấn luyện có dạng, $\langle x, f_{N+1}(x), \nabla_x f_{N+1}(x)\rangle$, trong đó $f_{N+1}(x)$ là nhãn lớp gốc (ban đầu) của $x \in \mathcal{D}_{N+1}$ (dữ liệu bài toán mới). Hệ thống được huấn luyện bằng thuật toán Tangent-Prop. $\nabla_x f_{N+1}(x)$ được ước lượng bằng cách sử dụng gradient của khoảng cách d thu được từ mạng nơron, nghĩa là $\nabla_x f_{N+1}(x) \approx \frac{\partial d_{x'}(x)}{\partial x}$, trong đó $\langle x', y' = 1\rangle \in \mathcal{D}_{N+1}$ và $d_{x'}(x) = d(x, x')$. Lý do là khoảng cách giữa x và một ví dụ huấn luyện tích cực x' là ước tính xác suất của x là một ví dụ tích cực, xấp xỉ $f_{N+1}(x)$. Kết quả là, EBNN được xây dựng phù hợp cho cả dữ liệu bài toán hiện tại $\mathcal{D}_{N+1}$ và tập hỗ trợ thông qua $\nabla_x f_{N+1}(x)$ và d.

Tuy nhiên, EBNN suốt đời không giữ lại bất kỳ tri thức nào đã học được trong quá khứ mà chỉ tích lũy dữ liệu quá khứ, nó cũng không hiệu quả nếu số lượng các bài toán trước đó lớn bởi vì huấn luyện hàm khoảng cách d cần thực hiện lại bằng cách sử dụng tất cả dữ liệu quá khứ (tập hỗ trợ) bất cứ lúc nào bài toán mới xảy ra. Thêm nữa, vì mỗi cặp của các điểm dữ liệu trong mỗi tập dữ liệu của bài toán quá khứ tạo thành một ví dụ huấn luyện để học hàm khoảng cách d nên dữ liệu huấn luyện để học d có thể là khổng lồ.

1.2.4. Vấn đề lãng quên tri thức của mạng nơron suốt đời

Vấn đề chính của các mô hình tính toán liên quan đến việc học liên tục là chúng dễ bị lãng quên (forgetting) hoặc can thiệp (interference) nghiêm trọng, có nghĩa là huấn luyện một mô hình với thông tin mới sẽ cản trở tri thức đã học trước đó. Hiện tượng này thường dẫn tới giảm hiệu suất đột ngột hoặc trường hợp xấu nhất là tri thức cũ hoàn toàn bị ghi đề bởi tri thức mới. Các mô hình học sâu hiện tại đạt kết quả rất tốt đối với các bài toán phân lớp bằng cách huấn luyện mô hình với một chuỗi dữ liệu có nhãn. Tuy nhiên, lược đồ học tập này giả định rằng tất cả các mẫu có sẵn trong giai đoạn huấn luyện và do đó yêu cầu huấn luyện lại các thông số mạng trên toàn bộ tập dữ liệu để thích ứng với những thay đổi trong phân phối dữ liệu. Khi được huấn luyện về các nhiệm vụ tuần tự với các mẫu dần dần có sẵn theo thời gian, hiệu suất của các mô hình mạng nơron thông thường giảm đáng kể các nhiệm vụ đã học trước đó khi các nhiệm vụ mới được học. Mặc dù huấn luyện lại từ đầu tránh được sự can thiệp thảm khốc nhưng phương pháp này rất kém hiệu quả và cản trở việc học dữ liệu mới trong thời gian thực. Trong các tình huống học tập liên tục, ví dụ, các tác nhân tương tác tự trị, có thể không có sự phân biệt giữa các

giai đoạn huấn luyện hay đánh giá, yêu cầu các mô hình đồng thời tìm hiểu và kích hoạt kịp thời các hành vi phản hồi.

Để khắc phục sự can thiệp nghiêm trọng này, hệ thống học tập phải, một mặt thể hiện khả năng thu nhận kiến thức mới và tinh chỉnh tri thức hiện có trên cơ sở đầu vào liên tục, mặt khác phải ngăn cản các tri thức mới can thiệp quá nhiều vào tri thức hiện có. Mức độ mềm dẻo mà hệ thống có thể tích hợp thông tin mới ổn định và không can thiệp nghiêm trọng vào tri thức tổng hợp được gọi là sự tiến thoái lưỡng tính dẻo (stability-plasticity dilemma) và được nghiên cứu rộng rãi trong cả các mô hình tính toán và lĩnh vực sinh học.

1.3. Giới thiệu chung về bài toán nhận dạng thực thể

Bài toán nhận dạng thực thể, hay còn gọi là bài toán nhận dạng thực thể định danh (Named Entity Recognition - NER) là bài toán xác định (phát hiện) các biểu diễn trong văn bản và phân lớp chúng vào các kiểu thực thể định danh được định nghĩa trước như Người, Địa danh, Thời gian, Số, giá trị tiền tệ,… Bài toán này có thể bao gồm cả việc nhận dạng các thông tin hay thuộc tính mô tả về thực thể. Ví dụ, trong trường hợp của thực thể tên người, hệ thống NER có thể trích xuất cả các thông tin về Chức danh, Quốc tịch, Địa chỉ, Giới tính,…

Một thực thể định danh là một chuỗi các từ chỉ đến một thực thể trong thế giới thực, ví dụ như “New York”, “Hà Nội”, “Hồ Chí Minh”, “Nông Đức Mạnh” và “UBND Thành phố Hà Nội”. Một thực thể định danh có thể được xếp vào một loại thực thể nào đó, như Người, Địa điểm, Tổ chức, Thời gian,… Như vậy, các thực thể chính là những đối tượng cơ bản nhất trong một văn bản dù ở bất kì ngôn ngữ nào.

Nhận dạng thực thể là một bài toán quan trọng, thường được sử dụng như là một bước tiền xử lý trong các hệ thống trích xuất thông tin hay trích chọn thông tin phức tạp. Có thể kể đến nhận dạng thực thể xuất hiện trong một số ứng dụng như Trích xuất quan hệ, Trích xuất sự kiện, Hệ thống hỏi đáp tự động,…

Trong nghiên cứu này, khóa luận tập trung vào việc nghiên cứu và xây dựng mô hình để đánh giá khả năng nhận dạng thực thể với bốn loại thực thể có tên cụ thể là tên người (PER), tên Tổ chức (ORG), tên địa điểm (LOC) và nhãn MISC.

- Người (PER) bao gồm Tên, tên đệm và họ của một người, tên động vật và các nhân vật hư cấu hoặc các bí danh.

- Tổ chức (ORG) bao gồm các cơ quan chính phủ, các công ty, thương hiệu, tổ chức chính trị, các tạp chí, báo và các tổ chức khác của con người.
- Địa điểm (LOC) bao gồm tên gọi các hành tinh, tên gọi quốc gia, vùng lãnh thổ, tên gọi các thực thể tự nhiên hay tên các địa chỉ, địa điểm.
- Nhãn MISC (tạm dịch là nhập nhằng) dùng để đánh dấu các các trường hợp nhập nhằng giữa tên quốc gia (LOCATION) với các tên có nghĩa thuộc về quốc gia đó. Trong tiếng Anh thì dựa vào hình thức biến hình của từ để xác định (danh từ → tính từ)

Ví dụ: "Chiều ngày 22/9/2017, Tổng cục Du lịch phối hợp với Hiệp hội Lữ hành Nhật Bản và đại diện Vietnam Airlines tại Nhật Bản tiếp tục tổ chức Chương trình phát động thị trường tại Nagoya". Ở đây: "Tổng cục Du lịch", "Hiệp hội Lữ hành Nhật Bản" và "Vietnam Airlines" là ORG, "Nhật Bản" và "Nagoya" là LOC.

1.4. Phát biểu bài toán nhận dạng thực thể trong văn bản tiếng Việt sử dụng mô hình học sâu suốt đời mức ký tự

Đầu vào: Các văn bản tiếng Việt thuộc miền dữ liệu báo chí, bao gồm các lĩnh vực: Văn học, nghệ thuật, giải trí, thể thao, pháp luật,… Trong khoá luận này, tôi sử dụng tập dữ liệu được cung cấp bởi VLSP 2018[1], mô tả chi tiết về tập dữ liệu này sẽ được đề cập đến trong chương 3.

Đầu ra: Một mô hình nhận dạng thực thể. Với mô hình nhận dạng thực thể đầu ra, xây dựng một (mô-đun) chương trình nhận một văn bản và cho ra các thực thể được nhận dạng trong văn bản đó.

Phương pháp:

- Biểu diễn văn bản dựa trên mô hình học sâu suốt đời mức ký tự.
- Thuật toán nhận dạng thực thể**:** sử dụng một (vài) phương pháp học máy theo mô hình máy hữu hạn trạng thái.

[1] http://vlsp.org.vn/vlsp2018

Kết luận chương 1

Chương một đã trình bày một số khái niệm cơ bản, các mô hình và kỹ thuật nổi bật của hai phương pháp học sâu và học suốt đời đồng thời phát biểu được bài toán nhận dạng thực thể trong văn bản tiếng Việt mà khoá luận giải quyết. Chương tiếp theo của khoá luận sẽ trình bày chi tiết về các kỹ thuật tiên tiến giải quyết bài toán.

CHƯƠNG 2: MỘT SỐ MÔ HÌNH HỌC SÂU VÀ HỌC SUỐT ĐỜI TRONG NHẬN DẠNG THỰC THỂ

Chương này trình bày một số mô hình học sâu và học suốt đời có liên quan trực tiếp tới nhận dạng thực thể. Cụ thể, đối với mô hình học sâu cho nhận dạng thực thể, khoá luận sẽ trình bày một mô hình học sâu nổi bật trong tiếng Việt sử dụng mạng bộ nhớ dài ngắn kết hợp với CNN và CRF được giới thiệu bởi tác giả Thai-Hoang Pham và cộng sự [9][11]. Đối với mô hình sử dụng phương pháp học suốt đời, hiện tại chưa có nghiên cứu cụ thể nào cho bài toán NER trong tiếng Việt. Do đó, khoá luận sẽ trình bày một mô hình trích xuất khía cạnh sử dụng phương pháp học suốt đời với CRF được giới thiệu bởi tác giả Lei Shu và cộng sự**Error! Reference source not found.** làm tiền đề để khoá luận xây dựng mô hình học sâu suốt đời cho bài toán NER trong tiếng Việt.

2.1. Mô hình Bi-LTSM-CRF sử dụng đặc trưng mức ký tự của từ

Bài báo trình bày công cụ NNVLP dựa trên mạng nơron cho xử lý ngôn ngữ tự nhiên cơ bản trong tiếng Việt bao gồm gán nhãn từ loại (Part-of-speech – POS), gán nhãn cụm từ (chunking) và nhận dạng thực thể (NER), bộ công cụ đạt kết quả tối ưu nhất về ba nhiệm vụ này.

2.1.1. Trường điều kiện ngẫu nhiên

Trường điều kiện ngẫu nhiên (Conditional Random Field – CRF) được giới thiệu vào những năm 2001 bởi Lafferty và các đồng nghiệp [5]. CRF là một nền tảng để xây dựng mô hình xác suất để phân đoạn và gán nhãn chuỗi. Trường điều kiện ngẫu nhiên dựa trên ý tưởng gốc từ mô hình Markov ẩn (Hidden Markov Model) và được cải thiện để khắc phục các nhược điểm của nó cũng như của mô hình markov entropy cực đại (Maximum Entropy Markov Model, MEMM).

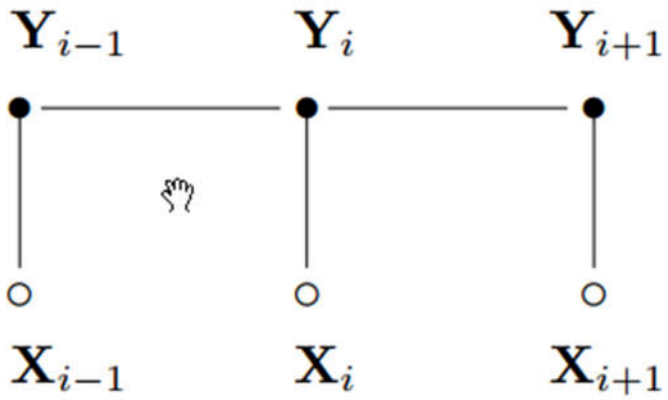
Kí hiệu X là biến ngẫu nhiên nhận giá trị là chuỗi dữ liệu cần phải gán nhãn và Y là biến ngẫu nhiên nhận giá trị là chuỗi nhãn tương ứng. Mỗi thành phần y_i của Y là một biến ngẫu nhiên nhận giá trị trong tập hữu hạn các trạng thái S. Ví dụ, trong bài toán nhận dạng thực thể có tên, X có thể nhận giá trị là các câu trong văn bản, Y là một chuỗi ngẫu nhiên các tên thực thể tương ứng với các câu này và mỗi thành phần y_i của Y có miền giá trị là tập tất cả các nhãn tên thực thể (PER, LOC, ORG, MISC).

Theo Lafferty, CRF được định nghĩa như sau: Cho một đồ thị vô vô hướng không có chu trình $G(V, E)$ sao cho $Y = (Y_v)_{v \in V}$ và Y là tập các đỉnh của G. Ta nói (X, Y) là

một trường ngẫu nhiên có điều kiện khi với điều kiện X, các biến ngẫu nhiên Y_v tuân theo tính chất Marko đối với đồ thị G:

$$p(Y_v|X, Y_w, w \neq v) = p\big(Y_v\big|X, Y_w, w \in N(v)\big)$$

Ở đây, $N(v)$ là tập tất cả các đỉnh kề v. Như vậy, một CRF là một trường ngẫu nhiên phụ thuộc hoàn toàn vào X. Kí hiệu $X = (x_1, x_2, \ldots., x_n), Y = (y_1, y_2, \ldots, y_n)$. Mô hình đồ thị cho CRF có dạng như hình dưới đây:



Hình 2.1: Một mạng CRF đơn giản [5]

2.1.2. Tập đặc trưng sử dụng

Bảng 2.1: Tập đặc trưng cho mỗi từ của mô hình [9]

STT	Đặc trưng cho mỗi từ
1	Đặc trưng ngữ nghĩa
2	Đặc trưng từ loại (POS)
3	Đặc trưng cụm từ (chunking)
4	Đặc trưng mức ký tự

Mô hình của tác giả Thai-Hoang Pham và cộng sự [9] sử dụng 4 đặc trưng cho mỗi từ, bao gồm đặc trưng về ngữ nghĩa, nhãn từ loại (POS), nhãn cụm từ (chunking) và đặc trưng mức ký tự (xem Bảng 2.1).

a) Đặc trưng về ngữ nghĩa

Để trích xuất đặc trưng về ngữ nghĩa của từ, nhóm tác giả sử dụng nhúng từ. Để tạo ma trận nhúng từ cho tiếng Việt, tác giả huấn luyện một mô hình Word2Vec[2] trên tập dữ liệu văn bản gồm 2 triệu bài báo, có dung lượng 7,3 GB, được thu thập từ trang web báo mới Việt Nam[3]. Văn bản ban đầu được chuẩn hoá về các ký tự thường và tất cả các ký tự đặc biệt được loại bỏ. Các ký tự phổ biến như dấu phẩy, dấu chấm phẩy, dấu hai chấm, dấu chấm, dấu phần trăm đều được thay thế bằng mã ký tự đặc biệt và tất cả các chuỗi số được thay thế với ký tự số đặc biệt. Mỗi từ trong ngôn ngữ tiếng Việt có thể bao gồm nhiều âm tiết với khoảng cách ở giữa có thể được coi như là nhiều từ bởi các mô hình không giám sát. Do đó cần phải thay thế các khoảng cách giữa mỗi từ bằng dấu gạch dưới để tạo ra các từ đầy đủ. Đối với các từ không xuất hiện trong tập từ nhúng, mô hình sẽ khởi tạo một vectơ ngẫu nhiên cho từ đó bằng cách lấy mẫu từ khoảng $\left[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}\right]$, trong đó dim là số chiều nhúng.

b) Đặc trưng cú pháp

Đặc trưng về cú pháp của từ (gọi chung cho đặc trưng từ loại và đặc trưng cụm từ) được các tác giả trích xuất từ nhãn từ loại và nhãn cụm từ tương ứng của từ đó. Cụ thể, mỗi nhãn được mã hoá one-hot thông qua một ma trận đơn vị có kích thước bằng số lượng nhãn từ loại hoặc cụm từ tương ứng.

c) Đặc trưng mức ký tự

Các đặc trưng đã kể trên được sử dụng nhằm mục đích nắm bắt thông tin về cú pháp và ngữ nghĩa của từ. Tuy nhiên, trong một số trường hợp, đặc trưng mức từ là chưa đủ. Các thông tin hữu ích về từ có thể là một phần của từ (một hoặc một vài ký tự ở đầu, cuối hoặc giữa của từ). Ví dụ, ký tự "h" trong "10h30" mang ý nghĩa là "giờ". Vì vậy, dựa trên giả thiết rằng ý nghĩa của một từ được tổng hợp từ thông tin của các ký tự, và thông tin của các ký tự trong cùng một từ có liên quan tới nhau, tác giả sử dụng phương pháp trích xuất đặc trưng của từ ở mức ký tự bằng cách kết hợp nhúng ký tự (Character Embedding) và mạng nơron tích chập (CNN).

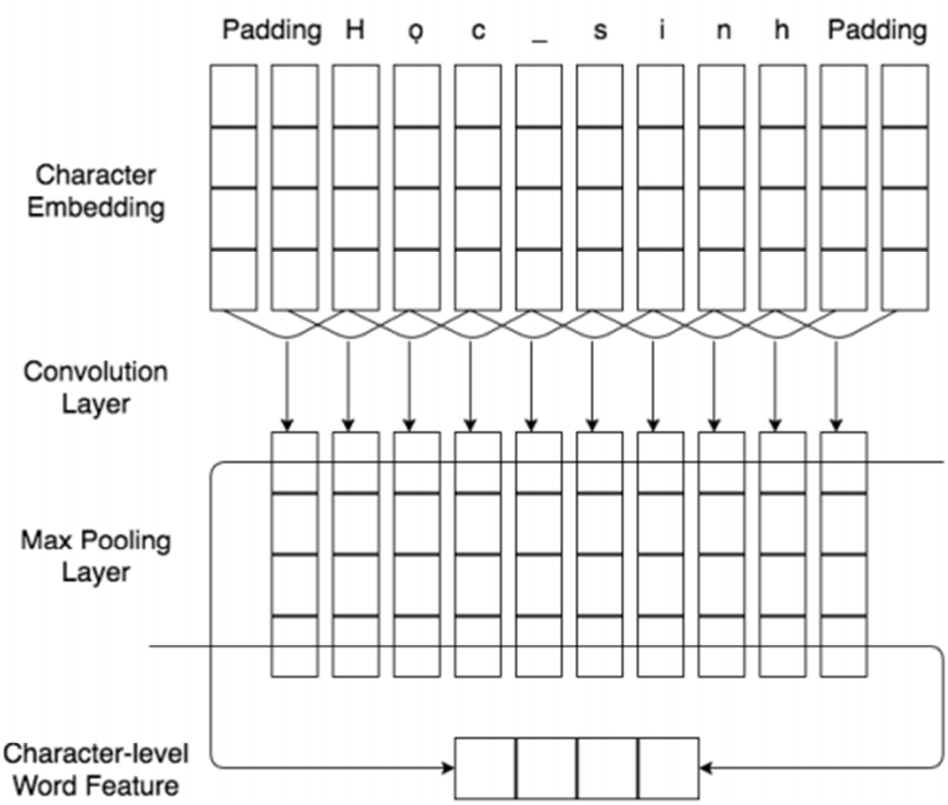
[2] https://code.google.com/archive/p/word2vec/
[3] http://www.baomoi.com

Nhúng ký tự hoạt động như sau. Cho một từ w được cấu tạo từ M ký tự $w = \{c_1, c_2, \dots, c_M\}$. Trước tiên, chúng ta biến đổi mỗi ký tự c_m thành một ký tự nhúng r_m^{chr}. Ký tự nhúng được mã hoá bởi các vectơ cột trong ma trận nhúng $W^{chr} \in \mathbb{R}^{d^{chr} \times |V^{chr}|}$. Cho một ký tự c, dạng nhúng của nó là r^{chr} có công thức

$$r^{chr} = W^{chr} v^c,$$

trong đó v^c là một vectơ có kích thước $|V^{chr}|$, có giá trị 1 tại vị trí (index) c và giá trị 0 tại tất cả các vị trí khác.

Sau khi nhúng ký tự, mỗi từ sẽ được biến đổi thành một ma trận $[r_1^{chr}, r_2^{chr}, \dots, r_M^{chr}]$, trong đó mỗi ký tự của từ đó là một vectơ r_i^{chr}. Ma trận này sẽ được đưa qua một lớp tích chập để nắm bắt đặc trưng mức cụm n ký tự (như đã mô tả tại mục 1.1.4). Sau đó, đầu ra của lớp tích chập sẽ được đưa qua một lớp tổng hợp giá trị cực đại (max-pooling) để nắm bắt đặc trưng nổi bật nhất trên mỗi bộ lọc của lớp tích chập. Cuối cùng ta thu được một vectơ đặc trưng có kích thước bằng với số bộ lọc của lớp tích chập, gọi là vectơ đặc trưng mức ký tự của từ. Hình 2.2 dưới đây mô tả việc sử dụng lớp tích chập để trích xuất đặc trưng mức ký tự của từ.



Hình 2.2: Trích xuất các đặc trưng mức ký tự của từ "Học_sinh" sử dụng CNN [9]

2.1.3. Mô hình Bi-LSTM+CRF sử dụng đặc trưng mức ký tự của từ

Mô hình Bi-LSTM-CRF kết hợp với phương pháp trích xuất đặc trưng mức ký tự của từ được mô tả trong Hình 2.3.

Như đã đề cập, đặc trưng từ mức ký tự sẽ được trích xuất bằng nhúng ký tự kết hợp CNN, giả sử vectơ đặc trưng này có d chiều. Tiếp theo, mô hình sử dụng nhúng từ để trích xuất đặc trưng mức từ, giả sử vectơ này có p chiều. Khi đó vectơ của mỗi nơron tại lớp đầu vào của mô hình Bi-LSTM+CRF sẽ có $(d + p)$ chiều.

Trong mô hình này, nhóm tác giả đã triển khai CRF ở trên lớp Bi-LSTM thay vì sử dụng lớp *softmax* và sử dụng đầu ra của lớp Bi-LSTM như là đầu vào của mô hình. Tham số của CRF là ma trận chuyển đổi A, $A_{i,j}$ đại diện cho điểm chuyển tiếp từ nhãn i tới nhãn j. Điểm của câu đầu vào x và nhãn chuỗi y được tính như sau:

$$S(x, y, \theta \cup A_{i,j}) = \sum_{t=1}^{T} \left(A_{y_{t-1}, y_t} + f_{\theta_{(y_t, t)}} \right)$$

trong đó θ là tham số của Bi-LSTM, f_θ là điểm đầu ra của Bi-LSTM và T là số bước. Sau đó nhãn chuỗi được tính bởi công thức softmax:

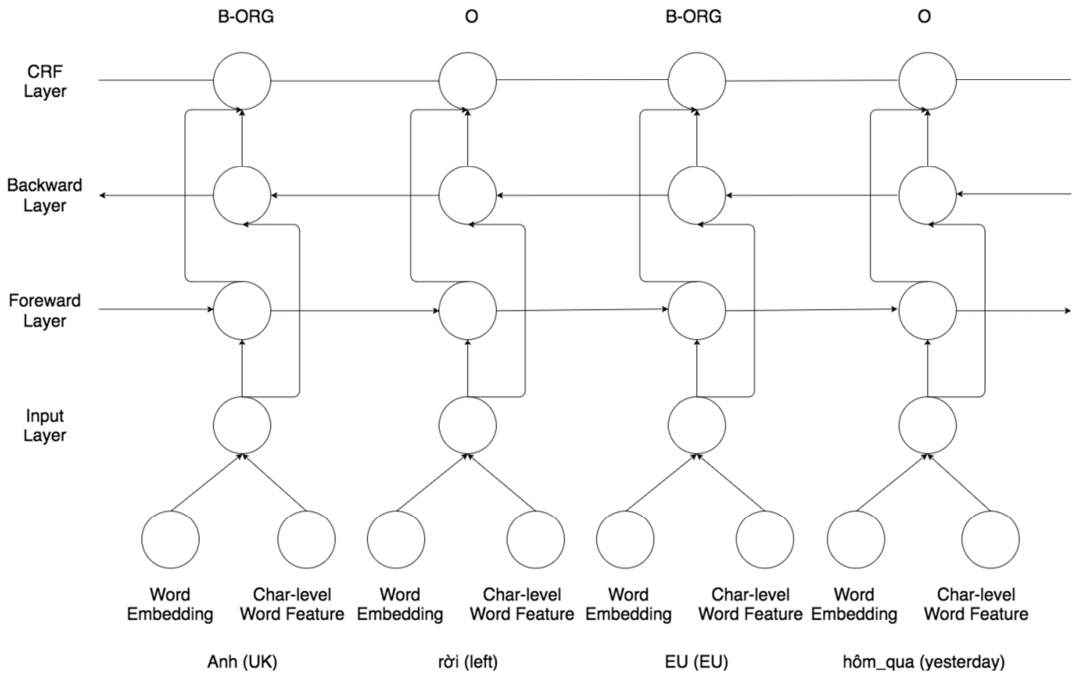
$$p(y|x, A) = \frac{\exp\left(S(x, y, \theta \cup A_{i,j})\right)}{\sum_{y' \in Y} \exp\left(S(x, y', \theta \cup A_{i,j})\right)}$$

trong đó Y là tập tất cả các chuỗi đầu ra có thể có. Trong pha huấn luyện (training), mô hình cực đại hàm Log-likelihood:

$$L = \sum_{i=1}^{N} \log p(y^i | x^i; A)$$

trong đó N là số lượng dữ liệu huấn luyện. Trong giai đoạn suy luận, thuật toán Viterbi được sử dụng để tìm ra chuỗi đầu ra y^* là xác suất cực đại:

$$y^* = \arg\max_{y \in Y} p(y|x; A)$$



Hình 2.3: Kiến trúc mô hình Bi-LSTM+CRF sử dụng đặc trưng mức ký tự của từ [9]

2.2. Mô hình trích xuất khía cạnh suốt đời sử dụng trường điều kiện ngẫu nhiên

2.2.1. Mô tả phương pháp

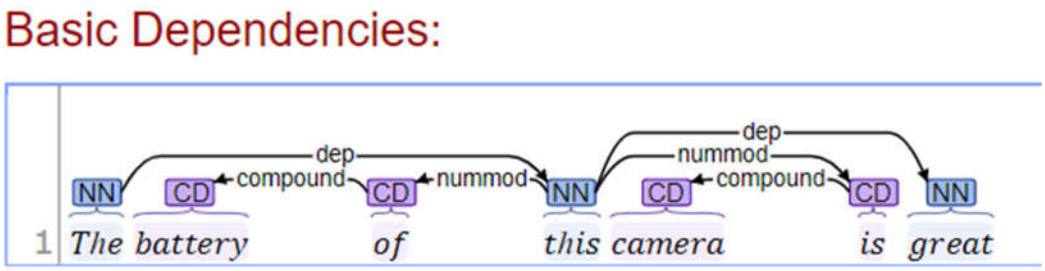
Mô hình trích xuất khía cạnh suốt đời sử dụng trường điều kiện ngẫu nhiên (Lifelong CRF) do tác giả Lei Shu và cộng sự**Error! Reference source not found.** đề xuất. Trích xuất khía cạnh (Aspect Extraction) là một tác vụ quan trọng của khai phá quan điểm [6]. Ví dụ, từ câu *"The battery of camera is good"*, mục tiêu là trích xuất "*battery*" là một đặc trưng của sản phẩm và được gọi là một khía cạnh (aspect).

Theo tác giả, cũng đã có các nghiên cứu về trích xuất khía cạnh, nhưng hầu hết đầu sử dụng phương pháp học không giám sát. Phương pháp học giám sát cũng đã được sử dụng nhưng đều là bài toán phân lớp hơn là học hay gán nhãn chuỗi như CRF. Bởi vậy, đây là mô hình đầu tiên sử dụng học máy suốt đời (LML) để giúp phương pháp trích xuất có giám sát cải thiện đáng kể hiệu năng của mô hình. Cụ thể, bài báo tập trung vào trích xuất khía cạnh có giám sát và chứng minh rằng nếu hệ thống đã trích xuất khía cạnh từ rất nhiều miền dữ liệu trong quá khứ và giữ lại kết quả giống như là các mẩu tri thức thì Trường điều kiện ngẫu nhiên (CRF) có thể tận dụng tri thức này theo một cách học tập suốt đời để trích xuất trong miền mới hiệu quả hơn CRF truyền thống không sử dụng tri thức trước đó. Bài báo cũng chứng minh việc CRF có thể cải thiện hiệu suất học tập là khả quan vì mặc dù các miền là khác nhau, tuy nhiên có một lượng các khía cạnh chia sẻ giữa các miền. Các khía cạnh chia sẻ có thể không xuất hiện trong dữ liệu huấn luyện nhưng có thể xuất hiện trong dữ liệu kiểm thử.

Cụ thể, phương pháp L-CRF được mô tả như sau: "Một mô hình CRF M được huấn luyện với một tập dữ liệu huấn luyện D. Tại một thời điểm cụ thể, M đã trích xuất khía cạnh từ dữ liệu của n miền trước đó $D_1, D_2, D_3, \dots, D_n$ (chưa gán nhãn) và tập khía cạnh trích xuất được là $A_1, A_2, \dots, A_n$. Bây giờ, hệ thống đối mặt với một miền dữ liệu mới D_{n+1}. M có thể tận dụng các tri thức tiền nghiệm tin cậy trong $A_1, A_2, \dots, A_n$ để trích xuất khía cạnh từ miền D_{n+1} tốt hơn.". Sự cải tiến quan trọng của L-CRF là ngay cả sau khi huấn luyện mô hình sử dụng học có giám sát, mô hình vẫn có thể cải thiện việc trích xuất khía cạnh trong pha đánh giá hoặc trong các ứng dụng.

2.2.2. Tập đặc trưng sử dụng

Mô hình sử dụng tập đặc trưng gồm 7 đặc trưng $\{W, -1W, +1W, P, -1P, +1P, G\}$, trong đó: W là từ hiện tại, P là nhãn từ loại của W, $-1W$ là từ trước đó, $-1P$ là nhãn từ loại của $-1W$, $+1W$ là từ kế tiếp, $+1P$ là nhãn từ loại của $+1W$, G là đặc trưng phụ thuộc chung (General Dependency Feature). Tương tự như mô hình mà khoá luận đã trình bày tại mục 2.1.2, 6 đặc trưng đầu tiên khá đơn giản, do đó khoá luận sẽ không đề cập lại 6 đặc trưng này mà chỉ đề cập tới đặc trưng mang tính chất học suốt đời trong mô hình này là đặc trưng G. Đặc trưng G sử dụng tập các quan hệ phụ thuộc chung (generalized dependency relations và cho phép CRF sử dụng tri thức tiền nghiệm trong quá trình dự đoán để tăng hiệu năng. Mỗi quan hệ phụ thuộc trong tập các quan hệ phụ thuộc chung được gọi là một mẫu phụ thuộc (dependency pattern). Một mẫu phụ thuộc có dạng như sau: $(type, gov, govpos, dep, deppos)$. Trong đó $type$ là loại quan hệ phụ thuộc, gov là từ đang xét, $govpos$ là nhãn từ loại của từ đang xét, dep là từ phụ thuộc và $deppos$ là nhãn từ loại của từ phụ thuộc. Hình 2.4 dưới đây là một ví dụ về mẫu phụ thuộc lấy từ corenlp[4].



Hình 2.4: Ví dụ về một mẫu phụ thuộc cơ bản

Các bước tổng hợp các quan hệ phụ thuộc vào các mẫu phụ thuộc:

- Bước 1: Với mỗi quan hệ phụ thuộc, thay thế từ hiện tại và nhãn POS của nó bằng một ký tự bất kì (ví dụ $*$).
- Bước 2: Thay thế từ ngữ cảnh (context word, không phải từ đang xét) trong mỗi quan hệ phụ thuộc bằng một nhãn tri thức để tạo thành một mẫu phụ thuộc đặc trưng tổng quát hơn. Cho một tập các khía cạnh được đánh dấu trong tập dữ liệu huấn luyện K^t. Nếu từ ngữ cảnh xuất hiện trong tập K^t, ta thay thế từ đó với một nhãn tri thức "A" (aspect) hoặc "O" (other).

[4] http://corenlp.run

Ví dụ: Xét câu *"The battery of this camera is great"*. Giả sử từ hiện tại là *"battery"*, quan hệ phụ thuộc giữa từ *"battery"* và *"camera"* là *(nmod, battery, NN, camera, NN)*.

- Bước 1: Thay thế *"battery"* và nhãn POS của nó bằng ký tự $*$.
- Bước 2: *"camera"* xuất hiện trong K^t, ta thay thế *"camera"* với nhãn "A" .

  Mẫu phụ thuộc trở thành $(nmod,\ *,\ A,\ NN)$.

Tại sao mẫu phụ thuộc có thể cho phép mô hình CRF tận dụng tri thức cũ trong khi xử lý bài toán mới? Câu trả lời chính là nhãn tri thức "A". Giả sử, ta cần giải quyết bài toán mới D_{n+1} sử dụng mô hình CRF đã huấn luyện và chúng ta đã trích xuất khía cạnh từ n miền trước, tập các khía cạnh được giữ lại ương ứng với n bài toán là $A_1, \dots, A_n$. Sau đó, ta có thể khai phá các nhãn tin cậy từ $A_1, \dots, A_n$ và thêm chúng vào cơ sở tri thức. Cơ sở tri thức cho phép nhiều các nhãn tri thức trong mẫu phụ thuộc của tập dữ liệu A_{N+1} do các khía cạnh chia sẻ qua các miền . Càng nhiều đặc trưng mẫu phụ thuộc sẽ cho phép mô hình trích xuất được nhiều khía cạnh hơn trong miền dữ liệu mới D_{N+1}.

2.2.3. Các pha trong mô hình

L-CRF gồm hai pha. Pha huấn luyện và pha trích xuất suốt đời (Lifelong extraction). Pha huấn luyện mô hình CRF sử dụng tập dữ liệu huấn luyện D^t. Cũng giống như các pha huấn luyện mô hình bình thường và đã được đề cập ở tới ở mục 2.1.1.

Trong pha trích xuất suốt đời, Mô hình CRF M đã huấn luyện được sử dụng để trích xuất khía cạnh từ các miền dữ liệu mới (M không thay đổi và dữ liệu chưa gán nhãn). Tất cả kết quả thu được được giữ lại trong kho lưu trữ quá khứ S (trên n miền). Tại một thời điểm cụ thể, giả sử M đã trích xuất trên n miền quá khứ và hiện tại mô hình cần xử lý bài toán mới trên miền dữ liệu $n+1$. L-CRF sử dụng M và nhãn tin cậy (K_{n+1}) khai phá từ S và K^t $(K = K^t \cup K_{n+1})$ để trích xuất từ D_{n+1}. K^t là tập khía cạnh thu được từ tập dữ liệu huấn luyện và luôn tin cậy bởi chúng được gán nhãn thủ công. Không thể sử dụng tất cả các khía cạnh trích xuất từ các bài toán cũ như là dữ liệu tin cậy do khi trích xuất có thể sẽ có lỗi. Tuy nhiên, nếu khía cạnh đó xuất hiện nhiều lần thì khía cạnh đó có khả năng sẽ chính xác hơn. Vì vậy K_{n+1} là các khía cạnh thường xuyên khai phá từ S. Thuật toán trích xuất suốt đời được mô tả như Hình 2.5.

Algorithm 1 Lifelong Extraction of L-CRF

1: $K_p \leftarrow \emptyset$
2: **loop**
3: $F \leftarrow$ FeatureGeneration(D_{n+1}, K)
4: $A_{n+1} \leftarrow$ Apply-CRF-Model(M, F)
5: $S \leftarrow S \cup \{A_{n+1}\}$
6: $K_{n+1} \leftarrow$ Frequent-Aspects-Mining(S, λ)
7: **if** $K_p = K_{n+1}$ **then**
8: **break**
9: **else**
10: $K \leftarrow K^t \cup K_{n+1}$
11: $K_p \leftarrow K_{n+1}$
12: $S \leftarrow S - \{A_{n+1}\}$
13: **end if**
14: **end loop**

Hình 2.5: Thuật toán trích xuất đặc trưng suốt đời (Lifelong extraction) *Error! Reference source not found.*

Thuật toán trên thực hiện trích xuất khía cạnh lặp đi lặp lại trên miền dữ liệu D_{n+1}. Cụ thể các bước trong thuật toán được mô tả như sau:

1) Sinh tập đặc trưng (F) trên miền dữ liệu D_{n+1} (dòng 3) sau đó sử dụng mô hình CRF (M) trên tập F trích xuất khía cạnh thu được tập khía cạnh A_{n+1} (dòng 4).

2) Tập A_{n+1} được thêm vào kho lưu trữ S. Từ S, khai phá ra tập các khía cạnh thường xuyên K_{n+1} sử dụng ngưỡng λ.

3) Nếu tập K_{n+1} giống với tập K_p từ vòng lặp trước, có nghĩa là không có khía cạnh nào được tìm thấy thì vòng lặp sẽ dừng lại. Sử dụng quy trình vòng lặp ở đây vì mỗi lần trích xuất mang lại kết quả mới, có thể sẽ làm tăng kích thước của tập dữ liệu tin cậy trong quá khứ K.Tập K tăng có thể sẽ tạo ra được nhiều mẫu phụ thuộc hơn, từ đó có thể mô hình sẽ trích xuất chính xác hơn.

4) Nếu không, có nghĩa rằng có khía cạnh tin cậy mới được tìm thấy. M có thể trích xuất nhiều hơn trong vòng lặp tiếp theo. Dòng 10 và 11 cập nhật lại hai tập dữ liệu cho vòng lặp sau.

2.3. Nhận xét

Như đã đề cập, mô hình Bi-LSTM-CRF (mục 2.1) đạt kết quả tối ưu nhất trong bài toán NER cho tiếng Việt. Bên cạnh đó, thuật toán học suốt đời trong mô hình trích xuất khía cạnh (mục 2.2) tuy không phải là bài toán NER tiếng Việt nhưng có khả năng biến đổi để phù hợp với với bài toán này. Dựa vào đó, khoá luận sẽ kết hợp hai mô hình trên và xây dựng một mô hình học sâu suốt đời sử dụng kiến trúc của mô hình học sâu Bi-LSTM+CRF và thuật toán trích xuất suốt đời trong mô hình trích xuất khía cạnh cho nhận dạng thực thể trong văn bản tiếng Việt.

Kết luận chương 2

Chương này đã trình bày chi tiết về hai mô hình tiếp cận theo hướng học sâu và học suốt đời. Từ đó, khóa luận đưa ra ý tưởng xây dựng một mô hình học sâu suốt đời để giải quyết bài toán NER dựa trên hai mô hình đã trình bày. Các mô tả chi tiết về kiến trúc và quá trình xây dựng mô hình này sẽ được trình bày trong chương tiếp theo.

CHƯƠNG 3: MÔ HÌNH HỌC SÂU SUỐT ĐỜI MỨC KÝ TỰ CHO NHẬN DẠNG THỰC THỂ TRONG VĂN BẢN TIẾNG VIỆT

Dựa vào các nghiên cứu đã trình bày ở chương 2, trong chương này, khoá luận trình bày mô hình đề xuất sử dụng học sâu suốt đời mức ký tự và áp dụng trong bài toán nhận dạng thực thể.

3.1. Mô tả phương pháp

Học suốt đời là một mục tiêu lâu dài của học máy, trong đó các tác nhân không chỉ học hỏi (và nhớ) một loạt các bài toán đã học theo trình tự mà còn có khả năng chuyển giao kiến thức từ các bài toán trước đó để cải thiện hiệu suất học các bài toán mới. Trong khoá luận này, tôi đề xuất mô hình có thể tận dụng kiến thức đã học được từ các bài toán trước và chuyển giao kiến thức cho bài toán mới.

Phát biểu bài toán: Đề xuất mô hình nhận dạng thực thể trong văn bản tiếng Việt sử dụng học sâu suốt đời mức ký tự.

Khoá luận tìm hiểu và áp dụng phương pháp học sâu kết hợp với học suốt đời cho bài toán nhận dạng thực thể trong văn bản tiếng Việt.

Đầu vào: Tập các văn bản tiếng Việt thu thập từ các trang báo chí điện tử Việt Nam trên các miền như: Văn hoá, Giáo dục, Giải trí, Thể thao, Xã hội, Pháp luật, Thế giới,….

Đầu ra: Mô hình nhận dạng thực thể có tên cho tiếng Việt có khả năng gán nhãn cho các văn bản đầu vào. Sau đây mô hình này được gọi là *Deep LML*

Mô tả phương pháp: Cho K là tập tiền tố tin cậy được trích xuất từ các bài toán trong quá khứ sử dụng mô hình kí hiệu là M, ở đây khoá luận sử dụng kiến trúc mạng bộ nhớ dài ngắn kết hợp với trường điều kiện ngẫu nhiên (Bi-LSTM + CRF) được mô tả ở chương 2. M được huấn luyện sử dụng tập dữ liệu huấn luyện D^t. Ban đầu, tập K chính là tập K^t (tập tất cả các tiền tố tin cậy của tập dữ liệu huấn luyện D^t). Giả sử M xử lý nhiều bài toán hơn và nhiều tiền tố tin cậy được trích xuất, theo đó kích thước tập K cũng sẽ lớn hơn. Khi xử lý bài toán D_{n+1}, tập K cho phép trích xuất đặc trưng tiền tố được nhiều hơn, mô hình M có thể cho kết quả tốt hơn đối với bài toán mới. Mô hình gồm 3 pha chính:

- Huấn luyện mô hình
- Trích xuất đặc trưng suốt đời
- Đánh giá mô hình đề xuất

Pha huấn luyện mô hình gồm 3 bước:

1) *Tiền xử lý dữ liệu*
2) *Trích xuất đặc trưng*
3) *Huấn luyện mô hình mạng nơron Bi-LSTM + CRF*

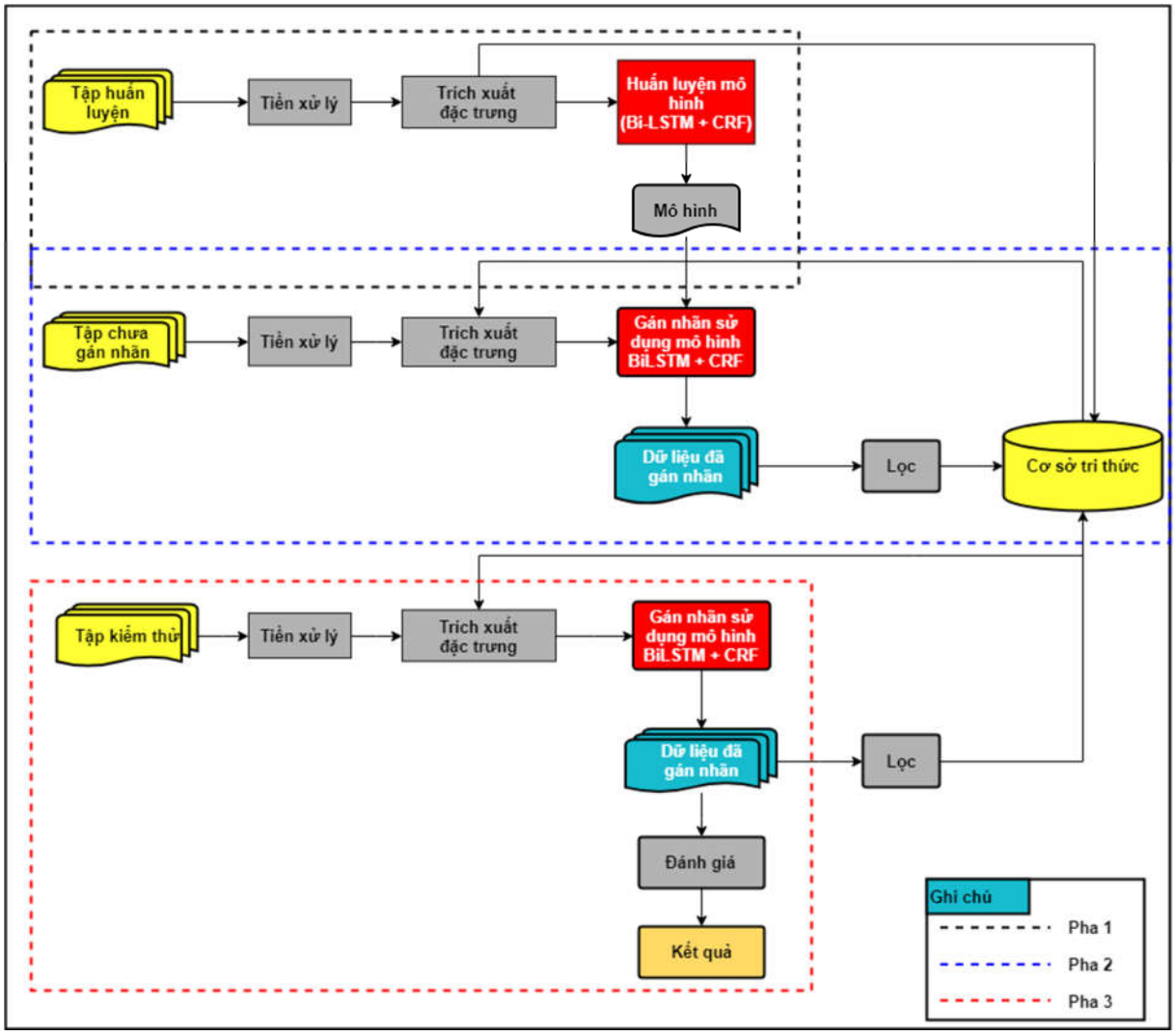
Pha trích xuất đặc trưng suốt đời: Trích xuất các đặc trưng sử dụng dữ liệu chưa gán nhãn được thu thập từ trang báo chí điện tử Dân trí.

Pha đánh giá mô hình đề xuất: Đánh giá mô hình sử dụng tập dữ liệu kiểm thử.

Chi tiết các bước trong các pha sẽ được mô tả chi tiết trong mục tiếp theo.

3.2. Mô hình đề xuất

Các pha của mô hình được mô tả trong hình dưới đây:



Hình 3.1: Mô hình NER sử dụng mạng nơron và phương pháp học suốt đời

3.3. Tập đặc trưng

Mô hình sử dụng tập đặc trưng gồm 4 loại đặc trưng được mô tả trong Bảng 3.1 dưới đây. Tương tự như mô hình Bi-LSTM-CRF đã đề cập tại mục 2.1, khoá luận cũng sử dụng đặc trưng về ngữ nghĩa, từ loại và mức ký tự. Tuy nhiên để có thể áp dụng học suốt đời vào mô hình, ta cần một đặc trưng mà có khả năng mở rộng giống như mô hình trích xuất khía cạnh suốt đời đã đề cập tại mục 2.2. Do đó khoá luận đề xuất một đặc trưng gọi là đặc trưng tiền tố. Cụ thể về cả 4 đặc trưng này sẽ được trình bày chi tiết tại mục 3.5.2.

Bảng 3.1: Tập đặc trưng cho mỗi từ mà mô hình của khoá luận sử dụng

STT	Đặc trưng	Ký hiệu
1	Đặc trưng ngữ nghĩa	w_i
2	Đặc trưng từ loại	w_i^p
3	Đặc trưng mức ký tự	w_i^c
4	Đặc trưng tiền tố	w_i^f

3.4. Cơ sở tri thức

Như đã đề cập, để lưu giữ lại các tri thức đã học được ta cần có một cơ sở tri thức. Khoá luận đề xuất một cơ sở tri thức gồm 3 thành phần chính: kho thông tin quá khứ (S), kho tri thức (K), bộ khai phá tri thức. Kho thông tin quá khứ được sử dụng để lưu lại các kết quả thu được từ các bài toán cũ, cụ thể là các tiền tố của các thực thể. Kho tri thức có tác dụng lưu các tiền tố tin cậy của bài toán đã học. Tập các tiền tố tin cậy được chia ra thành các tập tiền tố con theo nhãn: tiền tố tên người (PER), tiền tố tên địa danh (LOC), tiền tố tên tổ chức (ORG), tiền tố nhập nhằng (MISC). Bộ khai phá tri thức được dùng để khai phá tiền tố tin cậy từ Kho thông tin quá khứ, sau đó lưu vào kho tri thức.

3.5. Pha 1 – Huấn luyện mô hình

3.5.1. Tiền xử lý dữ liệu

Tập dữ liệu huấn luyện trong các kĩ thuật học máy luôn đòi hỏi phải được tiền xử lý trước khi huấn luyện. Dữ liệu sẽ được đưa qua công cụ VnCoreNLP [5]để tiến hành tách từ, gán nhãn từ loại. Đây là công cụ được tác giả Nguyễn Quốc Đạt phát triển nhằm mục

[5] https://github.com/vncorenlp/VnCoreNLP

đích hỗ trợ các bài toán xử lý ngôn ngữ tự nhiên trong tiếng Việt [7]. Sau đó, dữ liệu được định dạng theo chuẩn ConLL, chia thành 4 cột, các cột cách nhau bởi một khoảng trắng. Mỗi từ (token) được đặt trên một dòng và các câu tách nhau bởi một dòng trắng. Bốn cột trong dữ liệu theo thứ tự sẽ là: từ, nhãn từ loại (POS tag), nhãn thực thể, nhãn thực thể lồng nhau. Tuy nhiên, phạm vi của khoá luận không đề cập tới nhãn thực thể lồng nên cột thứ 4 sẽ không được sử dụng. Nhãn cú pháp và nhãn thực thể có dạng I-TYPE. Nếu hai cụm từ đồng loại đặt liền nhau thì từ đầu tiên của cụm thứ hai sẽ có nhãn là B-TYPE để chỉ ra nó là từ bắt đầu của một cụm mới. Từ có nhãn O là từ không thuộc về cụm nào cả. Có tất cả 9 nhãn: B-PER và I-PER sử dụng cho tên người, B-ORG và I-ORG sử dụng cho thực thể tổ chức, B-LOC và I-LOC sử dụng cho thực thể địa điểm, B-MISC và I-MISC sử dụng cho nhãn nhập nhằng và O sử dụng cho các thành phần còn lại.

3.5.2. Trích xuất đặc trưng

Giả sử câu đầu vào là một chuỗi các từ $S = [w_1, w_2, \dots, w_n]$ có độ dài n. Tại bước này, đặc trưng của mỗi từ w_i sẽ được trích xuất và biểu diễn dưới dạng một vectơ x_i.

a) Đặc trưng về ngữ nghĩa và từ loại

Giống như mô hình đã trình bày tại mục 2.1, khoá luận sử dụng nhúng từ và nhãn từ loại để trích xuất đặc trưng về ngữ nghĩa và cú pháp của từ trong câu.

Trong khoá luận này, tôi sử dụng một mô hình Word2Vec[6] đã được huấn luyện trước trên kho dữ liệu gồm 74 triệu bản ghi của Wikipedia. Ma trận nhúng từ của mô hình này có kích thước 10.087 từ, mỗi từ là một vectơ 100 chiều. Khoá luận cũng khởi tạo ngẫu nhiên một vectơ 100 chiều để đại diện cho các từ có trong dữ liệu nhưng không có trong ma trận nhúng. Mỗi từ w_i trong câu sẽ được tìm kiếm trong ma trận nhúng để thu được vectơ đặc trưng về ngữ nghĩa w_i^w.

Đối với nhãn từ loại, khoá luận sử dụng phương pháp one-hot để mã hoá nhãn. Một tập m nhãn từ loại sẽ được mã hoá thông qua ma trận đơn vị (identity matrix) có kích thước $m * m$ (chỉ các số trên đường chéo của ma trận có giá trị 1, các vị trí còn lại có giá trị 0). Vectơ đặc trưng từ loại của từ w_i^p sẽ là dòng thứ j của ma trận, trong đó $j \in [0; m-1]$ là vị trí (index) của nhãn từ loại của từ đó trong danh sách các từ loại.

[6] https://github.com/Kyubyong/wordvectors

b) Đặc trưng của từ mức ký tự

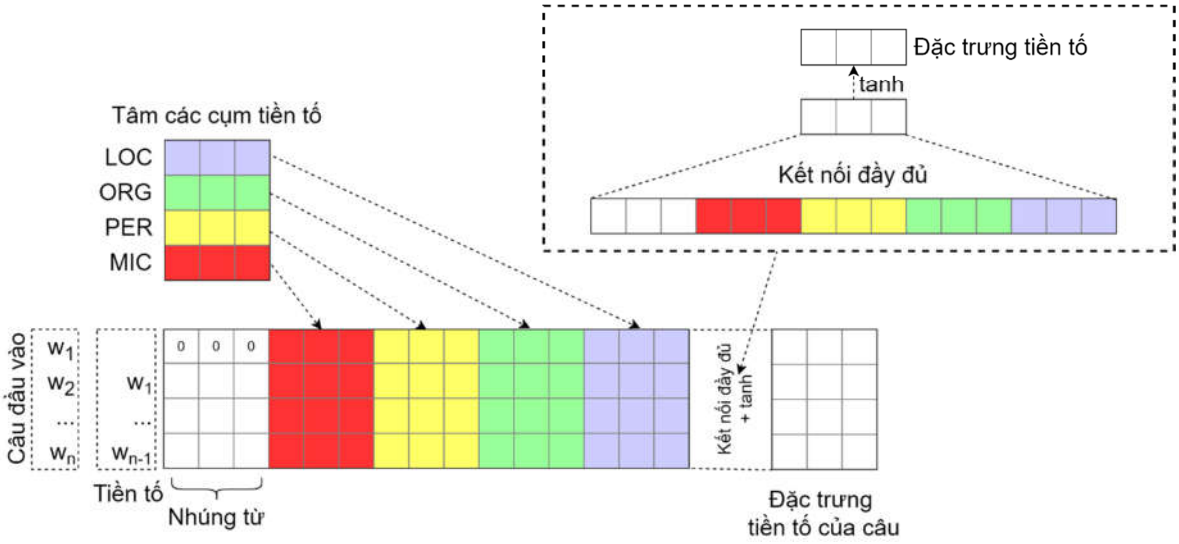
Giống như mô hình đã trình bày tại mục 2.1, khoá luận sử dụng CNN để trích xuất đặc trưng mức ký tự của từ. Ma trận nhúng ký tự sẽ được khởi tạo ngẫu nhiên sau đó cho mô hình tự học. Các ký tự của từ sẽ được mã hoá dựa trên ma trận này, sau đó được đưa qua mạng CNN để thu được vectơ đặc trưng mức ký tự w_i^c.

c) Đặc trưng tiền tố

Thông tin về tiền tố có vai trò quan trọng trong việc xác định một thực thể có tên trong câu, ví dụ từ "công ty" thường đứng trước tên một tổ chức (nhãn ORG) hay đứng trước các thực thể tên người (nhãn PER) thường là "ông", "bà",… Thông qua việc sử dụng một danh sách tiền tố cho các nhãn, ta có thể trích xuất đặc trưng cho mỗi từ bằng cách tính độ tương quan giữa từ đó với danh sách các tiền tố. Đồng thời, sử dụng danh sách tiền tố còn có khả năng mở rộng – một tính chất quan trọng để mô hình có thể áp dụng học suốt đời. Do đó, khoá luận đề xuất trích xuất đặc trưng về tiền tố của từ.

Một tập các tiền tố được khoá luận xác định trước bằng cách lấy tất cả tiền tố của các từ có nhãn NER khác O trong tập dữ liệu huấn luyện. Tập tiền tố được chia ra thành các tập con dựa trên nhãn NER của từ đứng sau nó và tất cả tập tiền tố này được lưu vào trong cơ sở tri thức. Sau đó, tôi sử dụng nhúng từ như mô tả ở phần a) để mã hoá các từ này, rồi tính vectơ trung bình trên mỗi tập tiền tố con (gọi là tâm cụm tiền tố).

Các vectơ tâm các cụm tiền tố sẽ được nối với vectơ nhúng từ của tiền tố của mỗi từ (xem Hình 3.2). Sau đó vectơ này được đưa qua một lớp kết nối đầy đủ (fully connected) với hàm kích hoạt $tanh$ để thu được vectơ đặc trưng tiền tố w_i^f có kích thước d_f.



Hình 3.2: Biểu diễn đặc trưng tiền tố

d) Biểu diễn vectơ đặc trưng của từ

Cuối cùng, tất cả các đặc trưng trích xuất được sẽ được nối lại thành một vectơ duy nhất cho mỗi từ:

$$x_i = w_i^w \oplus w_i^p \oplus w_i^c \oplus w_i^f$$

3.5.3. Huấn luyện mô hình - mạng nơron Bi-LSTM + CRF

Sau bước tiền xử lý dữ liệu, hệ thống trích xuất và biểu diễn dữ liệu huấn luyện dưới dạng các vectơ đặc trưng của từ trong câu (như đã mô tả trong mục 3.4.2).

Tập huấn luyện sẽ được chia thành các lô (batch) nhỏ để xử lý theo lô. Mô hình sẽ tính giá trị mất mát (loss) và các độ đo (sẽ trình bày ở pha đánh giá mô hình) theo lô, cuối mỗi lô mô hình sẽ cập nhật lại trọng số một lần. Để cập nhật lại các trọng số của mô hình sau mỗi lô, khoá luận sử dụng hàm tối ưu hoá Adaptive Moment Estimation (Adam). Gọi các trọng số của mô hình tại thời điểm lô thứ t là $w^{(t)}$ và mất mát (loss) tại thời điểm đó là $L^{(t)}$. Thuật toán lan truyền ngược sử dụng hàm tối ưu hoá Adam như sau:

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)}$$

$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1-\beta_2)\left(\nabla_w L^{(t)}\right)^2$$

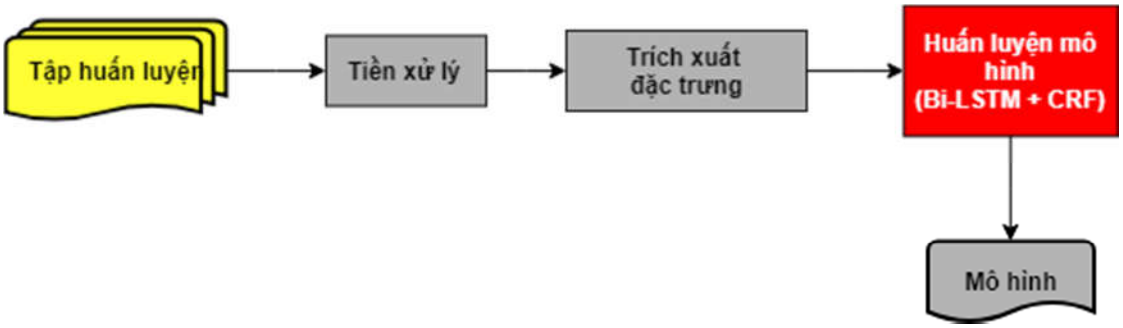
$$\hat{m}_w = \frac{m_w^{(t+1)}}{1-\beta_1^t}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-\beta_2^t}$$

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon}$$

trong đó, ϵ là một số rất nhỏ để tránh trường hợp chia cho 0, β_1 và β_2 là các tham số về độ giảm và mô men của độ dốc, η là tốc độ học của Adam.

Huấn luyện mạng nơron BiLSTM +CRF với tập dữ liệu huấn luyện sẽ thu được mô hình dùng để trích xuất và suy luận trong pha tiếp theo.



Hình 3.3: Pha 1 - Huấn luyện mô hình

3.6. Pha 2 – Trích xuất đặc trưng suốt đời

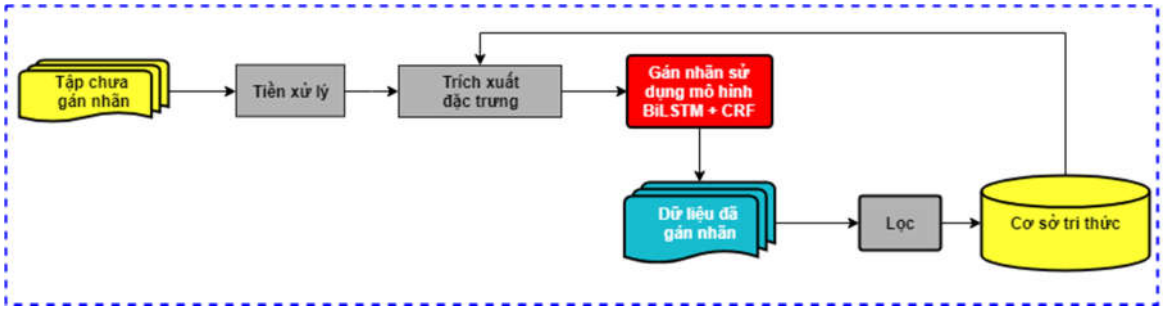
Như đã trình bày ở mục 2.2, mô hình đề xuất của khoá luận sẽ sử dụng thuật toán trích xuất đặc trưng suốt đời trong mô hình trích xuất khía cạnh để trích xuất đặc trưng suốt đời, tuy nhiên thay vì sử dụng đặc trưng phụ thuộc chung thì mô hình sử dụng đặc trưng tiền tố. Trong pha này, mô hình M đã huấn luyện ở pha trước được sử dụng để trích xuất các tiền tố từ các bài toán mới (mô hình M không thay đổi và dữ liệu bài toán mới chưa gán nhãn). Tất cả các kết quả từ miền mới được lưu lại trong kho thông tin quá khứ S. Sau khi M đã trích xuất tiền tố với n miền, và khi đối mặt với bài toán $(n+1)$, sử dụng mô hình M và các tiền tố tin cậy (kí hiệu là K_{n+1}) khai phá từ S và K^t ($K = K^t \cup K_{n+1}$). Các tiền tố trong K_t từ tập dữ liệu là tin cậy vì chúng được gán nhãn thủ công. Không thể sử dụng tất cả các tiền tố thu được từ các bài toán cũ như là dữ liệu tin cậy do khi mô hình gán nhãn có thể sẽ có lỗi. Tuy nhiên, nếu tiền tố đó xuất hiện nhiều lần thì tiền tố đó có khả năng sẽ chính xác hơn.

Thuật toán trích xuất đặc trưng suốt đời:

1 $K_p \leftarrow \emptyset$
2 **loop:**
3 $F \leftarrow FeatureGeneration\ (D_{n+1}, K)$
4 $E_{n+1} \leftarrow Apply - Model(M, F)$
5 $S \leftarrow S \cup \{E_{n+1}\}$
6 $K_{n+1} \leftarrow Frequent - prefixs - Mining(S, \lambda)$
7 **if** $K_p = K_{n+1}$ **then:**
8 **break**
9 **else:**
10 $K \leftarrow K^t \cup K_{n+1}$
11 $K_p \leftarrow K_{n+1}$
12 $S \leftarrow S - \{E_{n+1}\}$
13 **end if**
14 **end loop**

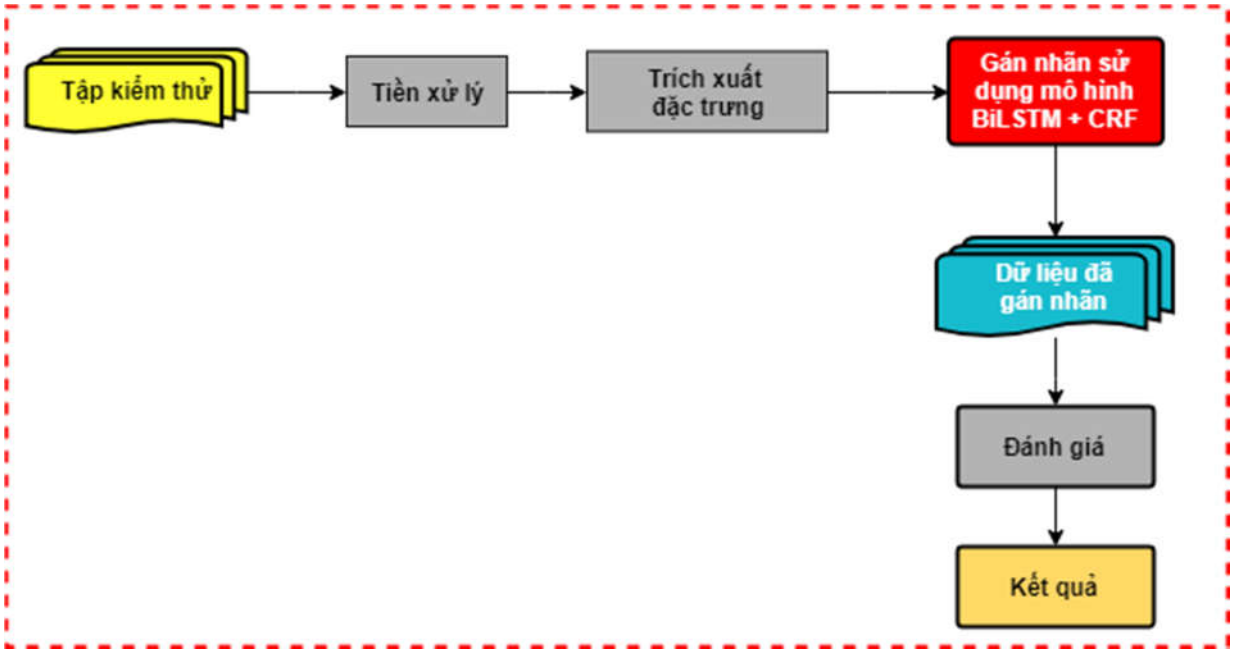
Giải thích các bước trong thuật toán:

1. Sinh đặc trưng F trên tập dữ liệu D_{n+1} và áp dụng vào mô hình M để sinh ra tập các thực thể E_{n+1} (dòng 3)
2. E_{n+1} (kết quả thu được khi sử dụng mô hình M) được thêm vào tập S - kho thông tin quá khứ. Từ S, khai phá ra các tiền tố thường xuyên K_{n+1} sử dụng ngưỡng λ.
3. Nếu tập K_{n+1} giống với tập K_p từ vòng lặp trước, có nghĩa là không có tiền tố nào được tìm thấy thì vòng lặp sẽ dừng lại.
4. Nếu không, có nghĩa rằng có các tiền tố tin cậy mới được tìm thấy. M có thể gán nhãn chính xác hơn trong vòng lặp tiếp theo. Dòng 10 và 11 cập nhật lại hai tập dữ liệu cho vòng lặp sau.



Hình 3.4: Pha 2 - Trích xuất đặc trưng suốt đời

3.7. Pha 3 – Đánh giá mô hình



Hình 3.5: Pha 3 - Đánh giá mô hình

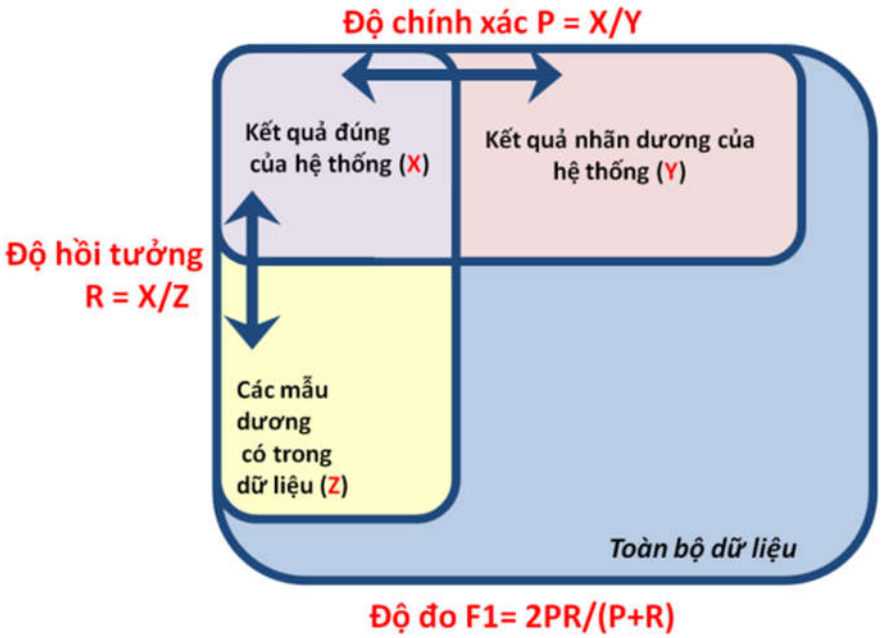
Trong pha này, dữ liệu kiểm thử được đưa qua mô hình nhận dạng thực thể (Bi-LSTM + CRF) và sử dụng cơ sở tri thức có được từ các bài toán trước để trích xuất đặc trưng. Trước khi đưa qua bước nhận dạng, các dữ liệu cũng được tiền xử lý bằng công cụ VnCoreNLP để tách từ, gán nhãn từ loại, sau đó được trích xuất các đặc trưng tương tự như các pha trên nhưng sử dụng thêm cơ sở tri thức để trích xuất thêm đặc trưng tiền tố. Kết quả đầu ra của pha này sẽ là các câu đã được gán các nhãn thực thể như mô tả ban đầu. Sau khi nhận được kết quả là dữ liệu đã gán nhãn, mô hình đánh giá độ chính xác bằng các độ đo được mô tả ở mục 3.7.1.

3.7.1. Độ đo đánh giá

Để đánh giá hiệu năng của mô hình trên tập dữ liệu đã chuẩn bị, khoá luận sử dụng độ chính xác (Precision - P), độ hồi tưởng (Recall - R) và độ đo F_1 được mô tả như sau:

- Độ chính xác (P) được tính bằng phần trăm các kết quả đúng trong tổng số nhãn dương của hệ thống.
- Độ hồi tưởng (R) là phần trăm các các trường hợp được gán nhãn đúng trong tất cả các mẫu dương hiện có trong dữ liệu.
- Độ đo F_1 là trung bình nhân của độ chính xác và độ hồi tưởng.

Hình dưới đây mô tả một cách trực quan các độ đo này.



Hình 3.6: Mô tả các độ đo chính xác, độ hồi tưởng và độ đo F_1

3.7.2. Phương pháp đánh giá

Để so sánh, khoá luận sử dụng cùng dữ liệu huấn luyện và dữ liệu kiểm thử trên cùng các mô hình. Các miền dữ liệu được kết hợp lại với nhau cho pha huấn luyện và tiến hành kiểm thử theo 2 cách cùng miền (in-domain) và khác miền (cross-domain). Giả sử rằng pha trích xuất đặc trưng suốt đời đã được thực hiện trên 16 miền dữ liệu chưa gán nhãn với ngưỡng λ.

- ***Khác miền*** (cross-domain): Kết hợp 9 miền dữ liệu cho pha huấn luyện và kiểm thử trên 10 miền khác nhau (không sử dụng trong pha huấn luyện). Thu được

10 kết quả. Phương pháp đánh giá này mong muốn có được mô hình huấn luyện để sử dụng hiệu quả trong các miền khác nhau, từ đó có thể tiết kiệm được công sức gán nhãn thủ công

- ***Cùng miền*** (in-domain): Huấn luyện và kiểm thử trên 9 miền giống nhau. Thu được 10 kết quả.

Ngoài ra, để đánh giá được sự cải tiến của mô hình khi giải quyết bài toán mới sau khi đã tận dụng các tri thức học được từ các bài toán cũ, khoá luận so sánh mô hình *Deep LML* với 3 mô hình không sử dụng tri thức của bài toán cũ:

1) Mô hình CRF-suite[7]: Mô hình này chỉ sử dụng tập đặc trưng gồm 2 đặc trưng $F = (w_i, w_i^p)$
2) Mô hình Bi-LSTM+CRF+prefix: Mô hình này sử dụng tập đặc trưng $F = (w_i, w_i^p, w_i^c, w_i^f)$ giống với mô hình *Deep LML* nhưng không sử dụng tri thức tiền nghiệm.

Kết luận chương 3

Trong chương này, khoá luận đã trình bày tư tưởng chính của phương pháp đề xuất cho bài toán nhận dạng thực thể trong văn bản tiếng Việt sử dụng mô hình học sâu suốt đời mức ký tự. Khoá luận đã giới thiệu chi tiết các pha cũng như các bước của phương pháp đề xuất. Trong chương tiếp theo, khoá luận tiến hành thực nghiệm trên các phương pháp đã xây dựng và đánh giá kết quả đạt được của mô hình đề xuất.

[7] https://sklearn-crfsuite.readthedocs.io/en/latest/

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1. Giới thiệu chung

Trong chương 3 đã giới thiệu mô hình nhận dạng thực thể sử dụng mạng nơron kết hợp với phương pháp học suốt đời và các pha thực hiện cũng như từng bước tiến hành mỗi pha. Trong chương này, một thực nghiệm xây dựng mô hình nhận dạng thực thể đề xuất ở chương 3 được tiến hành nhằm làm rõ các bước thực hiện các pha như đã giới thiệu. Mô hình được thực hiện trên tập dữ liệu VLSP2018 trên các miền thuộc lĩnh vực báo chí điện tử.

4.2. Môi trường và các công cụ sử dụng thực nghiệm

4.2.1. Cấu hình phần cứng

Để huấn luyện và đánh giá mô hình, khoá luận sử dụng máy chủ ảo của Google Cloud Platform[8] với cấu hình phần cứng như sau:

Bảng 4.1: Cấu hình phần cứng

Thành phần	Cấu hình
CPU	4 vCPUs, Intel Skylake
RAM	3.6 GB
OS	Ubuntu 16.04
SSD	10 GB

[8] https://cloud.google.com/

4.2.2. Các phần mềm sử dụng

Bảng 4.2: Các phần mềm sử dụng

STT	Tên phần mềm	Tác giả	Chức năng	Nguồn
1	Pycharm		Môi trường phát triển	https://www.jetbrains.com/pycharm
2	Anaconda 5.1 64-bit with Python 3.6		Ngôn ngữ phát triển và môi trường ảo	https://www.anaconda.com/download/
3	VnCoreNLP	Nguyễn Quốc Đạt	Tách từ, gán nhãn từ loại	https://github.com/vncorenlp/VnCoreNLP
4	Pre-trained word2vec	Kyubyong	Bộ nhúng từ được huấn luyện sẵn	https://github.com/Kyubyong/wordvectors
5	Numpy 1.14.0		Thư viện Python để tính toán trên các ma trận	http://www.numpy.org
6	Keras 2.1.5		Thư viện Python để thiết kế mạng nơron	https://github.com/keras-team/keras

4.3. Dữ liệu

Khoá luận sử dụng 2 loại dữ liệu. Dữ liệu VLSP2018 và dữ liệu chưa gán nhãn được thu thập từ trang báo điện tử Dân trí[9].

Bộ dữ liệu học được cung cấp bởi cuộc thi xử lý ngôn ngữ và giọng nói tiếng Việt 2018 (Vietnamese Language and Speech Processing 2018 – VLSP 2018), là các bài viết đăng trên các phương tiện truyền thông và mạng xã hội, không phải dữ liệu nhân tạo (do người làm dữ liệu tự sinh ra). Trong đó, bốn loại thực thể có tên được xác định tương thích với các loại thực thể mô tả trong nhiệm vụ cộng đồng (CoNLL Shared Task 2003[10]) thuộc hội nghị Conference on Natural Language Learning (CoNLL) năm 2002 và 2003 là Tên địa danh (LOC), tên người (PER), tên tổ chức (ORG) và nhãn MISC. Một thực thể có thể chứa thực thể khác nhúng trong đó. Ví dụ "Uỷ ban nhân dân Thành phố Hà Nội" là tên tổ chức, trong đó có chứa tên địa danh "thành phố Hà Nội". Dữ liệu được định dạng

[9] http://dantri.com.vn/
[10] http://www.clips.uantwerpen.be/conll2002/ner/

với chuẩn MUC6[11]. Cụ thể, dữ liệu huấn luyện là văn bản thô có bổ sung thêm các thẻ đánh dấu các thực thể, ví dụ với câu *"Nhóm Da LAB gồm 3 thành viên Mpakk, Thỏ và JGKid."* thì dữ liệu được định dạng như sau:

Nhóm <ENAMEX TYPE="ORGANIZATION">Da LAB</ENAMEX> gồm 3 thành viên <ENAMEXTYPE="PERSON">Mpakk </ENAMEX>,<ENAMEX TYPE="PERSON"> Thỏ </ENAMEX> và <ENAMEX TYPE="PERSON">JGKid </ENAMEX>.

Tập dữ liệu được chia thành 3 phần Train, Dev và Test với 10 tập dữ liệu nhỏ hơn theo các lĩnh vực. Bảng 4.3 thống kê số lượng thực thể chia theo từng miền của tập dữ liệu VLSP2018. Theo thống kê, tập dữ liệu học xấp xỉ 16.300 câu và 374.000 cụm từ. Số lượng thực thể địa danh nhỏ hơn so với hai thực thể còn lại, tuy nhiên không nhỏ hơn quá nhiều nên điều đó không ảnh hưởng lớn tới sự cân bằng của dữ liệu. Tuy nhiên, tập dữ liệu còn chứa thực thể lồng. Bên cạnh 3 thực thể chính PER, ORG, LOC, thực thể lồng là thực thể mà có một thực thể khác nằm trong nó. Ví dụ "[*Ủy ban nhân dân*]^{ORG} [*thành phố Hà Nội*]^{LOC}" là thực thể tổ chức (ORG) lồng thực thể địa điểm (LOC), "[*Đại học*]^{ORG} [*Tôn Đức Thắng*]^{PER}" là thực thể tổ chức (ORG) lồng thực thể tên người (PER). Đó chính là nguyên nhân gây ra nhầm lẫn trong quá trình nhận dạng thực thể. Hình 4.1 chỉ ra một ví dụ về thực thể lồng trong tập dữ liệu VLSP 2018.

"Uỷ ban Mặt trận Tổ Quốc tỉnh Hải Dương đã tổ chức phát động quyên góp ủng hộ đồng bào miền Trung khắc phục thiệt hại do bão số 10 gây ra."

<ENAMEX TYPE="ORGANIZATION"> Ủy ban Mặt trận Tổ quốc <ENAMEX TYPE="LOCATION"> tỉnh Hải Dương </ENAMEX></ENAMEX> đã tổ chức phát động quyên góp ủng hộ đồng bào <ENAMEX TYPE="LOCATION"> miền Trung </ENAMEX> khắc phục thiệt hại do bão số 10 gây ra.

Hình 4.1: Ví dụ về thực thể lồng

Bảng 4.4 so sánh lượng từ vựng giao nhau giữa các miền (Xác suất một từ là thực thể trong lớp X của miền A cũng là thực thể trong lớp X của miền B) trong tập VLSP2018. Ta có thể thấy, số lượng thực thể giao nhau giữa các miền khá thấp (cao nhất là 15%). Do đó khi thực hiện kiểm thử chéo các miền, sẽ thấy được ý nghĩa của việc tận dụng kiến thức từ các bài toán cũ cho bài toán mới.

[11] https://cs.nyu.edu/cs/faculty/grishman/muc6.html

Bộ dữ liệu thứ 2 là tập dữ liệu chưa gán nhãn được thu thập từ 1600 bài báo thuộc 16 lĩnh vực trên trang báo điện tử Dân trí. Thống kê số lượng câu và cụm từ được thể hiện chi tiết trong Bảng 4.5.

Bảng 4.3: Số lượng thực thể chia theo từng miền của tập dữ liệu VLSP 2018

Tập	Miền	Cụm từ	Số câu	PER	ORG	LOC	MISC
Train	Đời sống	37513	1756	138	427	59	8
	Giải trí	33496	1642	226	1078	165	97
	Giáo dục	40634	1715	412	603	442	81
	KH-CN	36981	1495	417	200	497	142
	Kinh tế	49873	1918	671	409	1028	56
	Pháp luật	31847	1385	590	1066	466	13
	Thế giới	31986	1422	1830	597	592	42
	Thể thao	27147	1340	447	1074	873	147
	Văn hoá	49051	2194	1507	481	216	199
	Xã hội	36417	1449	881	382	713	16
Dev	Đời sống	16098	778	35	66	27	7
	Giải trí	9821	461	79	319	48	57
	Giáo dục	14867	600	146	180	157	6
	KH-CN	8930	412	159	76	94	14
	Kinh tế	13170	508	202	105	377	16
	Pháp luật	12237	502	176	436	243	3
	Thế giới	11853	503	680	109	263	13
	Thể thao	9860	576	147	423	345	23
	Văn hoá	17136	741	415	242	99	39
	Xã hội	11286	447	297	153	221	5
Test	Đời sống	17117	798	36	115	38	7
	Giải trí	21685	1011	136	771	164	58
	Giáo dục	10382	446	30	80	55	6

	KH-CN	7048	312	64	76	29	4
	Kinh tế	42698	1656	418	293	414	24
	Pháp luật	14499	498	160	341	172	3
	Thế giới	8978	416	305	253	76	24
	Thể thao	19253	878	122	800	586	33
	Văn hoá	19881	872	495	402	63	78
	Xã hội	20454	706	117	234	290	3
Tổng		682198	29437	11338	11791	8812	1224

Bảng 4.4: So sánh số thực thể giao nhau giữa các miền trong tập dữ liệu VLSP2018

Miền	**Đời sống**	**Giải trí**	**Giáo dục**	**KH-CN**	**Kinh tế**	**Pháp luật**	**Thế giới**	**Thể thao**	**Văn hoá**	**Xã hội**
Đời sống	1	0.14	0.13	0.1	0.14	0.14	0.09	0.11	**0.15**	0.13
Giải trí	0.05	1	0.05	0.05	0.07	0.04	0.04	0.06	0.08	0.05
Giáo dục	0.06	0.07	1	0.06	0.12	0.11	0.05	0.06	0.12	0.12
KH-CN	0.06	0.08	0.07	1	0.13	0.06	0.12	0.08	0.12	0.1
Kinh tế	0.04	0.06	0.08	0.07	1	0.08	0.06	0.05	0.09	0.11
Pháp luật	0.05	0.04	0.08	0.04	0.09	1	0.04	0.03	0.08	0.1
Thế giới	0.03	0.05	0.04	0.08	0.08	0.04	1	0.05	0.09	0.07
Thể thao	0.03	0.05	0.04	0.04	0.05	0.03	0.04	1	0.05	0.04
Văn hoá	0.03	0.05	0.06	0.05	0.07	0.06	0.05	0.04	1	0.08
Xã hội	0.04	0.05	0.08	0.05	0.11	0.08	0.05	0.04	0.1	1

Bảng 4.5: Thống kê số lượng thực thể theo từng miền của tập dữ liệu Dân trí

Miền	Cụm từ	Số câu
Chuyện lạ	350450	15169
Giaỉ trí	271666	11086
Giáo dục	680809	24331
Kinh doanh	483219	15795
Nhịp sống trẻ	309252	10910
Ô tô xe máy	480321	14680
Pháp luật	462295	16003
Sức khoẻ	475327	17885
Sức mạnh	427959	14340
Sự kiện	404959	14480
Tấm lòng nhân ái	180972	6746
Thế giới	401711	14664
Thể thao	402051	17215
Tình yêu giới tính	514916	23872
Văn hoá	433822	15947
Xã hội	402472	13463
Tổng	6682201	246586

4.4. Cài đặt tham số

Như đã đề cập ở mục 3.5.2, khoá luận sử dụng nhúng từ và số chiều đặc trưng tiền tố là 100. Do tập dữ liệu tương đối lớn, nên khoá luận chọn kích thước lô bằng 20. Đối với số đơn vị trong lớp LSTM, do hạn chế về phần cứng nên khoá luận giảm xuống còn 100. Đối với các tham số của hàm tối ưu hoá Adam, khoá luận để mặc định theo thư viện mà khoá luận sử dụng. Các tham số còn lại (xem Bảng 4.6), khoá luận sử dụng giống với mô hình của nhóm tác giả Thai-Hoang Pham[10].

Bảng 4.6: Danh sách các tham số của mô hình

Tham số		**Giá trị**
Số chiều nhúng từ		100
Số chiều nhúng ký tự		30
Số chiều đặc trưng tiền tố		100
Số bộ lọc CNN		30
Kích thước cửa sổ tích chập		3
Số đơn vị trong 2 lớp LSTM		100
Dropout		0.5
Kích thước lô		20
Adam	Tốc độ học	0.01
	β_1	0,9
	β_2	0,999
	ϵ	10^{-7}
Ngưỡng λ		2

4.5. Kết quả thực nghiệm và nhận xét

Bảng 4.7: Kết quả thực nghiệm theo Cross-domain và In-Domain

Cross-domain										
Training	**Testing**	**CRF**			**Bi-LSTM+CRF**			**Deep LML**		
		$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
– Đời sống	Đời sống	75.71	67.05	70.36	65.31	65.98	65.64	68.37	71.28	69.79
– Giải trí	Giải trí	64.00	53.96	55.73	68.38	68.56	68.47	69.86	68.46	69.15
– Giáo dục	Giáo dục	70.83	63.42	66.27	81.29	72.77	76.8	81.29	73.16	77.01
– KH-CN	KH-CN	60.18	62.8	57.89	66.47	53.74	59.43	67.63	53.92	60.00
– Kinh tế	Kinh tế	74.38	64.53	67.12	70.23	69.81	70.02	67.89	73.24	70.46
– Pháp luật	Pháp luật	83.47	75.78	78.92	84.91	86.84	85.86	85.65	86.68	**86.16**
– Thế giới	Thế giới	50.00	56.55	59.08	63.22	63.22	63.22	67.48	66.77	67.12
– Thể thao	Thể thao	62.62	37.47	42.54	40.88	51.85	45.72	39.45	56.88	46.59
– Văn hoá	Văn hoá	62.53	49.55	53.17	68.24	65.74	66.96	62.27	64.01	63.36
– Xã hội	Xã hội	82.38	68.57	74.23	76.24	78.31	77.26	77.8	76.84	77.31
Average		68.61	59.97	**62.53**	68.52	67.68	**67.94**	68.77	69.12	**68.70**
In - domain										
– Đời sống	– Đời sống	70.64	63.88	66.34	65.69	70.95	68.22	69.34	70.36	69.84
– Giải trí	– Giải trí	71.46	63.84	66.79	70.24	68.93	69.58	68.22	68.17	68.19
– Giáo dục	– Giáo dục	69.82	66.36	67.23	69.29	70.31	69.8	66.78	69.97	69.34
– KH-CN	– KH-CN	70.74	64.58	66.7	69.76	71.55	70.64	69.25	70.82	70.03
– Kinh tế	– Kinh tế	70.78	61.75	65.02	69.26	71.33	70.28	66.88	69.29	68.06
– Pháp luật	– Pháp luật	69.06	61.21	64.3	65.34	69.19	67.21	67.3	69.89	68.57
– Thế giới	– Thế giới	72.03	63.61	66.84	66.85	71.09	68.9	67.94	71.21	69.54
– Thể thao	– Thể thao	72.18	66.15	68.56	70.64	72.81	71.71	73.00	70.67	**71.82**
– Văn hoá	– Văn hoá	73.70	66.17	69.07	68.96	71.93	70.42	70.19	71.55	70.86
– Xã hội	– Xã hội	69.34	62.19	64.73	66.84	70.48	68.61	68.65	70.75	69.68
Average		70.98	63.97	**66.56**	68.29	70.86	**69.54**	68.76	70.27	**69.59**

Bảng 4.7 thể hiện kết quả thực nghiệm chi tiết của khoá luận, trong đó ký hiệu $-X$ có nghĩa là tất cả các miền ngoại trừ X. Ta thấy rằng đối với cả hai phương pháp thực nghiệm, kết quả của mô hình học sâu tốt hơn kết quả của mô hình học không sâu thông thường và mô hình học sâu suốt đời cho kết quả tốt hơn cả 2 mô hình còn lại.

Cross-Domain: Mỗi miền $-X$ ở trong cột Training có nghĩa rằng X không được sử dụng trong pha huấn luyện. Mỗi miền X trong cột Testing có nghĩa rằng X được sử dụng trong pha đánh giá. Nhìn vào bảng kết quả, ta có thể thấy Deep LML cho kết quả tốt hơn CRF và Bi-LSTM+CRF với độ đo F_1 đạt 68.7%

In-Domain: Mỗi miền $-X$ ở trong cột Training và cột Testing có nghĩa rằng tất cả 9 miền còn lại ngoại trừ X được sử dụng để huấn luyện và đánh giá. Chúng ta lại thấy rằng Deep LML cho kết quả tốt hơn CRF và Bi-LSTM+CRF với độ đo F_1 đạt 69.59%, tuy nhiên % cao hơn không đáng kể. Nhưng kết quả này là hợp lý bởi hầu hầu hết các dữ liệu nhãn có trong pha huấn luyện cũng sẽ xuất hiện trong pha đánh giá.

Kết luận chương 4

Trong chương này, khoá luận đã mô tả về tập dữ liệu và các tham số mà mô hình sử dụng. Bên cạnh đó, khoá luận cũng đã tiến hành thực nghiệm và thu được các kết quả khả quan ban đầu. Qua đây ta thấy phương pháp đề xuất trong khoá luận có thể áp dụng được trong thực tiễn với nhiều miền ứng dụng khác nhau

KẾT LUẬN

Khoá luận đã tiếp cận được những phương pháp học sâu và học suốt đời trong bài toán nhận dạng thực thể được nghiên cứu và công bố trên thế giới. Dựa vào đó, khoá luận đã tiến hành phân tích và xây dựng mô hình học sâu suốt đời mức ký tự cho nhận dạng thực thể trong văn bản tiếng Việt.

Kết quả đạt được của khoá luận:

- Khảo sát, tìm hiểu về phương pháp học sâu và học sâu suốt đời cũng như các mô hình nổi bật về nhận dạng thực thể. Từ đó, khoá luận đưa ra phương pháp tiến cận dựa trên nghiên cứu của tác giả Lei Shu**Error! Reference source not found.** và Thai-Hoang Pham [9] để xây dựng một mô hình học sâu suốt đời mức ký tự cho bài toán nhận dạng thực thể trong văn bản tiếng Việt.
- Khoá luận đã thừa kế thuật toán từ nghiên cứu của tác giả Lei Shu**Error! Reference source not found.** và đưa ra một thuật toán học suốt đời cụ thể áp dụng cho bài toán nhận dạng thực thể trong tiếng Việt đồng thời xây dựng một mô hình học sâu suốt đời sử dụng mạng nơron dài ngắn hạn hai chiều (Bi-LSTM) kết hợp với trường điều kiện ngẫu nhiên (CRF) cho phép mô hình giữ lại tri thức thu được từ các bài cũ và tận dụng tri thức đó để cải thiện tốc độ học và hiệu suất giải quyết bài toán mới.
- Khoá luận tiến hành thực nghiệm trên tập dữ liệu VLSP2018 và xây dựng tập dữ liệu thu thập từ trang báo chí điện tử Dân trí để trích xuất đặc trưng suốt đời. Qua thực nghiệm đã thu được kết quả ban đầu khá khả quan với độ đo F1 trung bình đạt % với cross-domain và % với in-domain. Kết quả cho thấy phương pháp học sâu suốt đời có thể cải thiện được hiệu suất của mô hình dựa trên các tri thức tiền nghiệm.

Hạn chế: Do hạn chế về thời gian và kiến thức của cá nhân, khoá luận vẫn tồn tại một số hạn chế như sau: Thứ nhất, khoá luận mới chỉ tập trung vào xây dựng mô hình chứ chưa xây dựng thành một hệ thống có ứng dụng cụ thể và trực quan. Thứ hai, tập dữ liệu của khoá luận chưa được làm mịn. Thứ ba, do sử dụng các công cụ tách từ và gán nhãn từ loại bên ngoài nên vẫn tồn tại những từ được tách và gán nhãn không đúng, phần nào ảnh hưởng tới hiệu suất học của mô hình. Cuối cùng, mô hình chưa tận dụng triệt để

được tri thức từ các bài toán cũ và chuyển giao để học bài toán mới dẫn tới kết quả của mô hình đề xuất chưa cao hơn nhiều so với mô hình cơ sở.

Hướng phát triển trong tương lai: Trong thời gian tới, khoá luận sẽ cố gắng tinh chỉnh dữ liệu để cải thiện hiệu suất học của mô hình, đồng thời xây dựng một hệ thống nhận dạng thực thể trực quan hơn. Theo hướng tiếp cận hiện tại, mô hình cơ sở (Bi-LSTM+CRF) không hề thay đổi cấu trúc bên trong, vì vậy trong tương lai khoá luận dự kiến sẽ thay đổi mô hình cơ sở để có thể tận dụng tri thức tiền nghiệm cho bài toán mới tốt hơn.

Bên cạnh kết quả đã đạt được, khoá luận còn nhiều thiếu sót và hạn chế, tôi rất mong nhận được sự đóng góp ý kiến của thầy cô và bạn bè.

TÀI LIỆU THAM KHẢO

[1] Caruana, R. Multitask learning(1998). In *Learning to learn*. Springer, Boston, MA. pp. 95-133.

[2] Chen, Z., & Liu, B. (2016). Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *10*(3), 1-145..

[3] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493-2537.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[5] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data..

[6] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*(1), 1-167.

[7] Nguyen, D. Q., Vu, T., Nguyen, D. Q., Dras, M., & Johnson, M. (2017). From Word Segmentation to POS Tagging for Vietnamese. *arXiv preprint arXiv:1711.04951*..

[8] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2018). Continual Lifelong Learning with Neural Networks: A Review. *arXiv preprint arXiv:1802.07569*..

[9] Parisi, G. I., Tani, J., Weber, C., & Wermter, S. (2017). Lifelong learning of human actions with deep neural network self-organization. *Neural Networks*, *96*, 137-149.

[10] Pham, T. H., Pham, X. K., Nguyen, T. A., & Le-Hong, P. (2017). NNVLP: A Neural Network-Based Vietnamese Language Processing Toolkit. *arXiv preprint arXiv:1708.07241*.

[11] Pham, T. H., & Le-Hong, P. (2017). End-to-end Recurrent Neural Network Models for Vietnamese Named Entity Recognition: Word-level vs. Character-level. *arXiv preprint arXiv:1705.04044*..

[12] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... & Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

[13] Shu, L., Xu, H., & Liu, B. (2017). Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*.

[14] Shu, L., Xu, H., & Liu, B. (2017). Lifelong learning crf for supervised aspect extraction. *arXiv preprint arXiv:1705.00251*.

[15] Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[16] Thrun, S. (1996). Is learning the n-th thing any easier than learning the first?. In *Advances in neural information processing systems* (pp. 640-646).