ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

PHẠM THỊ THU TRANG

NHẬN DẠNG THỰC THỂ ĐỊNH DANH TỪ VĂN BẢN NGẮN TIẾNG VIỆT VÀ ĐÁNH GIÁ THỰC NGHIỆM

LUẬN VĂN THẠC SĨ HỆ THỐNG THÔNG TIN

Hà Nội - 2018

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

PHẠM THỊ THU TRANG

NHẬN DẠNG THỰC THỂ ĐỊNH DANH TỪ VĂN BẢN NGẮN TIẾNG VIỆT VÀ ĐÁNH GIÁ THỰC NGHIỆM

Ngành: Công nghệ thông tin

Chuyên ngành: Hệ thống thông tin

Mã số: 60480104

LUẬN VĂN THẠC SĨ HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS Hà Quang Thụy

Hà Nội – 2018

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc nhất tới thầy giáo PGS.TS Hà Quang Thụy đã tận tình giúp đỡ, chỉ bảo và hướng dẫn em trong suốt quá trình thực hiện luận văn này.

Em xin bày tỏ lời cảm ơn trân thành đến những thầy cô nhiệt tình và tâm huyết đã giảng dạy em trong suốt hai năm qua, giúp em trang bị những kiến thức cơ bản nhất để có thể vững bước trong tương lai.

Em muốn gửi lời cảm ơn tới các anh chị và các bạn trong phòng thí nghiệm Khoa học dữ liệu và Công nghệ Tri thức đã chia sẻ cho em nhiều kiến thức bổ ích cũng như giúp đỡ em những lúc khó khăn khi thực hiện khóa luận này.

Lời cuối cùng, em muốn gửi lời cảm ơn đến cha mẹ và các chị những người luôn tin tưởng và ủng hộ em trên con đường em đã chọn, cũng như luôn che chở và giúp đỡ em để em có thể vượt qua những khó khăn trong cuộc sống.

Hà Nội, ngày 16 tháng 11 năm 2018

Học viên

Phạm Thị Thu Trang

LỜI CAM ĐOAN

Em xin cam đoan nhận dạng thực thể định danh từ văn bản ngắn tiếng Việt và đánh giá thực nghiệm được trình bày trong luận văn này là do em thực hiện dưới sự hướng dẫn của PGS.TS Hà Quang Thụy.

Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong khóa luận. Trong khóa luận, không có việc sao chép tài liệu, công trình nghiên cứu của người khác mà không chỉ rõ về tài liệu tham khảo.

Hà Nội, ngày 16 tháng 11 năm 2018

Học viên

Phạm Thị Thu Trang

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG BIỂU

Mở đầu

Nhận dạng thực thể định danh là một cầu nối quan trọng trong việc kết nối dữ liệu có cấu trúc và dữ liệu phi cấu trúc. Nó cũng có rất nhiều ứng dụng như: xây dựng máy tìm kiếm thực thể, tóm tắt văn bản, tự động đánh chỉ số cho các sách, bước tiền xử lí làm đơn giản hóa các bài toán dịch máy,… Bên cạnh đó, việc bùng nổ của các mạng xã hội như Facebook, Twitter,.. và các hệ thống hỏi đáp đã mang lại một lượng thông tin khổng lồ. Đặc điểm của các dữ liệu đó thường là các văn bản ngắn, từ ngữ được sử dụng thường là văn nói và liên quan đến nhiều miền dữ liệu khác nhau. Chính đặc điểm này đã mang lại nhiều khó khăn khi thực hiện bài toán nhận dạng thực thể định danh.

Khi gặp phải một vấn đề mới, chúng ta thường giải quyết nó dựa vào những tri thức, kinh nghiệm có trước. Ví dụ như: khi giải một bài toán ta thường liên hệ để đưa chúng về các dạng bài trước đây đã làm hoặc tìm sự tương đồng giữa chúng. Việc áp dụng những tri thức này thường làm tăng tốc độ cũng như chất lượng của việc học. Nhận xét này không chỉ liên quan đến việc học của con người mà còn liên quan đến học máy. Việc học trong một nhiệm vụ mới được cải thiện bằng việc sử tri thức đã được lưu lại từ những nhiệm vụ học trước đó. Nói cách khác là ta sử dụng những tri thức đã có nhằm nâng cao hiệu quả của việc học cho nhiệm vụ mới.

Ý thức được tầm quan trọng của bài toán nhận dạng thực thể cũng như ý nghĩa của học suốt đời, em đã chọn đề tài nhận dạng thực thể định danh từ văn bản ngắn tiếng Việt và đánh giá thực nghiệm. Đối với luận văn này, em sẽ tìm hiểu áp dụng thực nghiệm nhận dạng thực thể trong văn bản ngắn Tiếng Việt với mô hình CRFs áp dụng học suốt đời. Cụ thể, em sẽ tiến hành nghiên cứu áp dụng các tri thức được lưu lại từ việc học trong các miền trong quá khứ nhằm nâng cao hiệu suất của bài toán nhận dạng thực thể định danh trong nhiệm vụ học hiện tại.

Luận văn được tổ chức thành 4 chương như sau:

- *Chương 1* giới thiệu tổng quan về bài toán nhận dạng thực thể trong văn bản Tiếng Việt, những khó khăn gặp phải khi thực hiện bài toán này cho văn bản ngắn Tiếng Việt và những nghiên cứu có liên quan áp dụng cho Tiếng Anh, Tiếng Việt.

- *Chương 2* định nghĩa học suốt đời, kiến trúc mô hình học suốt đời, các đặc điểm của học suốt đời và phương pháp áp dụng học suốt đời vào mô hình trường ngẫu nhiên có điều kiện.
- *Chương 3* trình bày thuật toán L-CRFs nhằm tăng hiệu quả của mô hình trường ngẫu nhiên có điều kiện áp dụng cho bài toán nhận dạng thực thể định danh trong văn bản ngắn Tiếng Việt.
- *Chương 4* trình bày đánh giá thực nghiệm trong hai trường hợp: trong cùng một miền dữ liệu, đánh giá chéo miền không áp dụng học suốt đời và áp dụng học suốt đời với các kịch bản dữ liệu huấn luyện khác nhau.

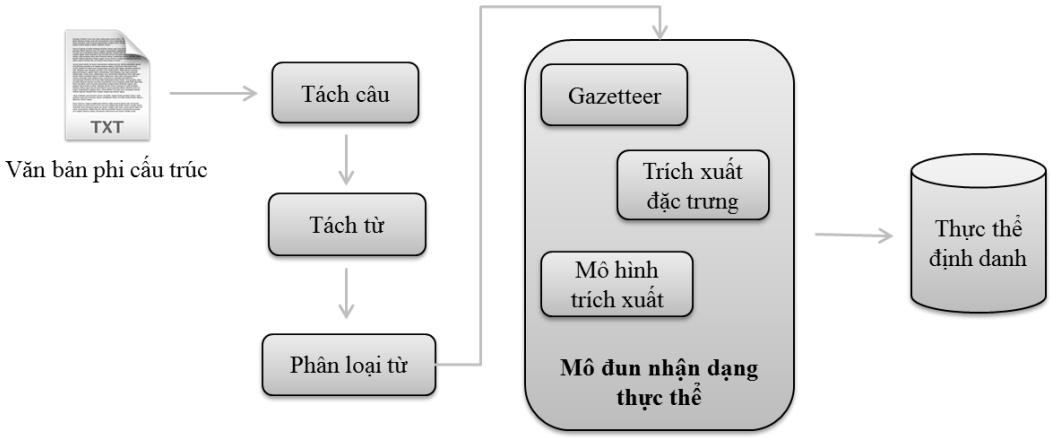
Chương 1. Bài toán nhận dạng thực thể cho văn bản ngắn Tiếng Việt

Đề tài chính của luận văn là nhận dạng thực thể định danh trong văn bản ngắn Tiếng Việt. Chương này sẽ giới thiệu về bài toán nhận dạng thực thể trong văn bản Tiếng Việt cùng những khó khăn gặp phải khi thực hiện bài toán này đối với văn bản ngắn.

1.1 Bài toán nhận dạng thực thể

1.1.1 Bài toán

Khác với việc đọc toàn bộ văn bản, các hệ thống trích chọn thông tin chỉ nhận biết các thông tin đáng quan tâm. Có nhiều mức độ trích chọn thông tin từ văn bản: trích chọn các thực thể, trích chọn mối quan hệ giữa các thực thể, xác định đồng tham chiếu… Vậy để trích chọn các thực thể hay mối quan hệ giữa chúng, ta phải nhận dạng được các thực thể. Nói cách khác, bài toán nhận dạng thực thể là bài toán đơn giản nhất trong các bài toán trích chọn thông tin, tuy vậy nó lại là bước cơ bản nhất để giải quyết các bài toán phức tạp hơn trong lĩnh vực này. Bài toán nhận dạng thực thể thường được chia thành hai quy trình liên tiếp: Nhận dạng thực thể và phân loại thực thể[2]. "Nhận dạng thực thể" là quá trình tìm kiếm các đối tượng được đề cập tới trong văn bản trong khi "Phân loại thực thể là việc gán nhãn cho các đối tượng đó. Một kiến trúc tiêu biểu mô tả cho quy trình nhận dạng thực thể được trình bày trong Hình 1.1:



Hình 1.1 Quy trình nhận dạng thực thể định danh[2]

Quy trình bao gồm:

- Tách câu: Trong qui trình này, văn bản phi cấu trúc được tách thành các câu riêng biệt
- Tách từ: Các câu được tách thành các từ, chữ số và dấu câu.
- Phân loại từ: Các từ sẽ được phân loại thành danh từ, động từ, tính từ …
- Mô đun nhận dạng thực thể bao gồm ba thành phần:
    - ***Từ điển định danh***: Bao gồm danh sách các tên đã được phân thành các loại thực thể. Trong lịch sử, thuật ngữ gazetteer được dùng để đề cập đến danh sách các địa danh địa lý và các thông tin liên quan; ở đây thuật ngữ này được áp dụng rộng rãi hơn cho danh sách tên của bất kỳ lớp nào.
    - ***Trích xuất đặc trưng***: Trích xuất các đặc trưng có ý nghĩa để làm đầu vào cho mô hình trích xuất.
    - ***Mô hình trích xuất***: Thành phần quan trọng nhất dùng để phân loại các thực thể dựa vào các đặc trưng được trích xuất.

Với mục tiêu của bài toán nhận diện thực thể là trích chọn ra những thực thể trong các văn bản, ta có thể xem xét bài toán nhận dạng thực thể như là một trường hợp cụ thể của bài toán gán nhãn cho dữ liệu dạng chuỗi. Ta có thể trình bày bài toán như sau[20]:

Đầu vào:

- O (o_1, o_2,…, o_T) : chuỗi dữ liệu quan sát, với o_i là các từ
- S (s_1, s_2,…, s_T) : chuỗi các trạng thái tương đương với chuỗi các nhãn cần gán cho dữ liệu.

Đầu ra: Các câu đã được gán nhãn (chuỗi các nhãn s_i cho từng câu)

Đối với bài toán nhận dạng thực thể trong văn bản Tiếng Việt, có một số loại thực thể thông dụng thường được tập trung nghiên cứu như: tên người, tên tổ chức…[9]. Các nhãn tương ứng với các loại thực thể được cho trong Bảng 1:

STT	Tên nhãn	Ý nghĩa
1	PER	Tên người
2	ORG	Tên tổ chức

3	LOC	Tên địa danh
4	NUM	Số
5	PCT	Phần trăm
6	CUR	Tiền tệ
7	TIME	Ngày tháng, thời gian
8	MISC	Những loại thực thể khác ngoài 7 loại trên
9	O	Không phải thực thể

Bảng 1.1 Danh sách các loại thực thể

Trong phạm vi nghiên cứu, luận văn chỉ tập trung vào 3 loại thực thể: tên người, tên tổ chức và tên địa danh.

1.1.2 Khó khăn của bài toán nhận dạng thực thể trong văn bản ngắn Tiếng Việt

Bên cạnh việc thiếu dữ liệu huấn luyện, bài toán nhận dạng thực thể trong văn bản Tiếng Việt còn gặp khá nhiều khó khăn do một số đặc điểm của Tiếng Việt[3].

- ***Tách từ :*** đây là bước tiền xử lý quan trọng trước khi hệ thống xác định được các thực thể. Hệ thống nhận diện được thực thể đúng với điều kiện cần là bước tách từ chính xác. Đơn vị cấu tạo cơ bản của Tiếng Việt là các "tiếng" tuy nhiên không phải "tiếng" nào cũng có nghĩa mà nó chỉ có nghĩa khi được ghép với một "tiếng" khác để tạo nên một từ có nghĩa. Ví dụ từ "âm ỉ" là một tính từ chỉ sự ngấm ngầm, không dữ dội nhưng lại kéo dài, tuy nhiên khi tách riêng ra thì từ "ỉ" là một từ không có nghĩa. Hay nói cách khác, hai từ cách nhau bởi một dấu cách chưa chắc đã là hai từ khác nhau mà là hai tiếng của một từ ghép. Do đó, công việc tách từ không đơn giản như tiếng Anh là chỉ dùng dấu cách để phân chia, mà phụ thuộc vào ngữ nghĩa, ngữ cảnh của câu
- ***Từ mượn:*** Hơn 50% Tiếng Việt bắt nguồn từ tiếng Trung Quốc gọi là từ Hán Việt. Tuy nhiên đây không phải là từ mượn mà là những từ được từ kế thừa. Hầu hết các từ mượn là có nguồn gốc từ Pháp. Ví dụ từ cinéma (Pháp) → xinê hoặc xi-nê. Hay

từ White House → Bạch_Ốc(Hán Việt), Nhà_trắng, chỉ những ngôi nhà có màu trắng, trong khi Nhà Trắng là chỉ nơi ở chính thức là làm việc của Tổng thống Mĩ.

- ***Định dạng*** của từ Tiếng Việt khác biệt so với trong Tiếng Anh. Ví dụ như những danh từ số nhiều trong Tiếng Anh được cấu thành từ những từ nguyên thể được thêm "s" hoặc "es" (apples, books). Trong khi để chỉ danh từ số nhiều trong Tiếng Việt thì được hình thành bằng việc thêm vào các từ như "các", "nhiều",…
- ***Từ đồng âm khác nghĩa*** (Ví dụ: "cuốc" và "quốc") và có những từ khác âm cùng nghĩa(Ví dụ: "tía", "ba", "cha"… cùng có nghĩa là bố).

Bên cạnh đó, ta cần xem xét những thách thức khi áp dụng bài toán cho văn bản ngắn. Văn bản ngắn đề cập đến ở đây có thể là các tweet, bài đăng trên facebook, đoạn trích tìm kiếm, đánh giá sản phẩm… Điểm khác biệt lớn nhất của các văn bản này với các văn bản truyền thống là về độ dài của văn bản [3] . Các văn bản ngắn thường có xu hướng mơ hồ và không đủ thông tin ngữ cảnh, một văn bản ngắn thường không có đủ nội dung hoặc các từ cụ thể trong khi một từ có thể được lặp đi lặp lại rất nhiều lần. Điều này gây khó khăn trong việc trích xuất các đặc trưng để làm đầu vào cho việc nhận dạng thực thể.

Chính bởi những đặc điểm đã khiến cho việc nhận dạng thực thể trong văn bản ngắn Tiếng Việt gặp nhiều khó khăn hơn trong việc áp dụng trong Tiếng Anh và trong các văn bản truyền thống.

Như vậy, ta cần một mô hình học có thể khắc phục được các thách thức về ngữ cảnh cũng như nội dung khi nhận dạng thực thể cho văn bản ngắn Tiếng Việt.

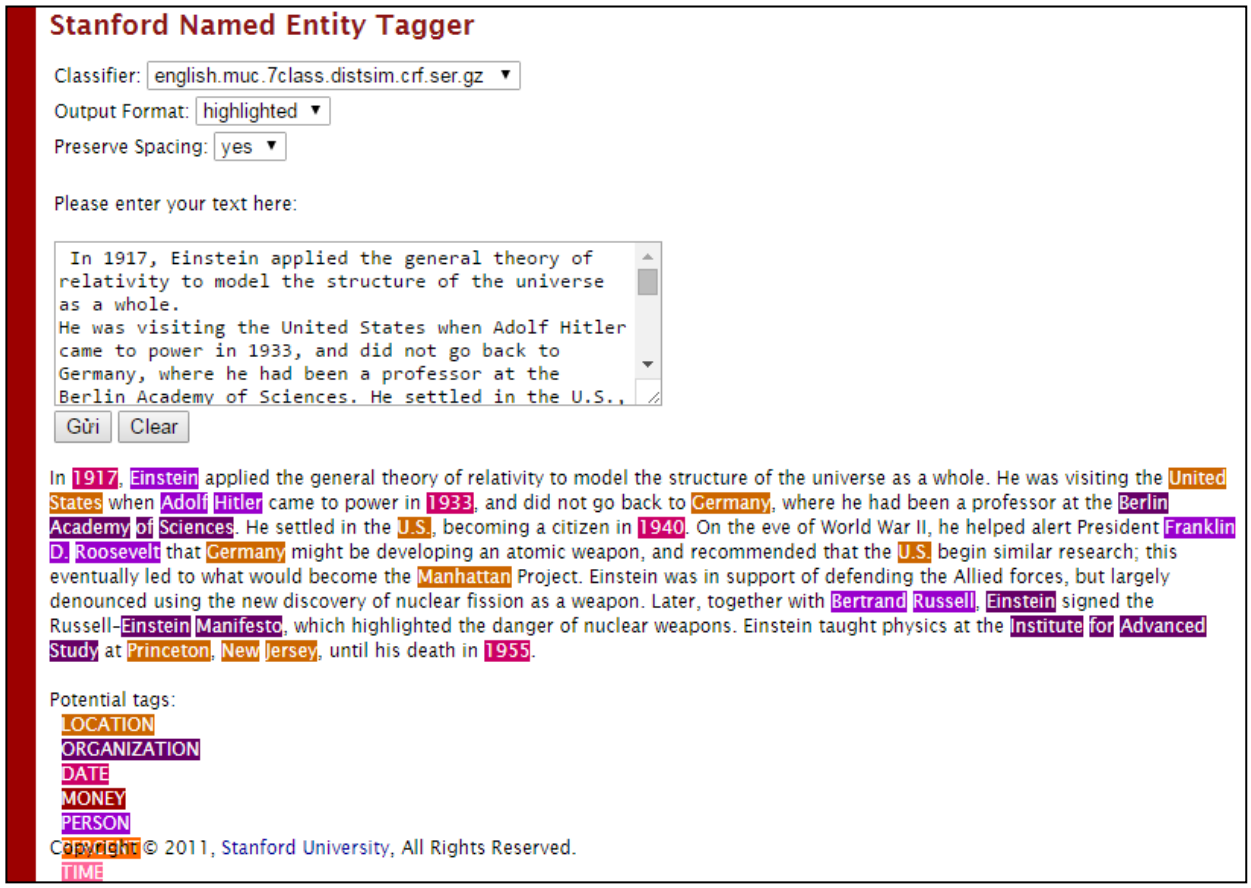
1.2 Các nghiên cứu có liên quan

1.2.1 Các nghiên cứu về nhận dạng thực thể trong Tiếng Anh

Bài toán nhận diện thực thể nhận được nhiều sự quan tâm của các nhà nghiên cứu trên toàn thế giới trong nhiều năm qua, bao gồm bài toán chung và các bài toán riêng trên từng miền ngôn ngữ. Trong thời kỳ ban đầu xuất hiện bài toán, các nghiên cứu tập trung xây dựng các hệ thống luật thủ công. Có đến năm trên tám hệ thống được giới thiệu tại MUC-7 (Seventh Message Understanding Conference, 1997) được xây dựng dựa trên luật. Một số nghiên cứu tiêu biểu là hệ thống Proteus của đại học New York [23A] hay các nghiên cứu trong các ngôn ngữ khác như nghiên cứu của E.Ferreira và cộng sự [6] trong tiếng Bồ Đào Nha, D.Farmakiotou và cộng sự [5] trong tiếng Hy Lạp.

Tuy nhiên trong thời gian gần đây, các nghiên cứu tập trung sang hướng áp dụng các phương pháp học máy. Trong đó, các kỹ thuật nổi bật hiện nay để giải quyết bài toán nhận diện thực thể là học có giám sát, bao gồm các phương pháp như sử dụng các mô hình Markov ẩn (HMMs) như nghiên cứu của Zhou và cộng sự [22], các mô hình Maximum Entropy (MEMMs) với nghiên cứu của McCallum và cộng sự [12], sử dụng máy vector hỗ trợ (SVM) hay tiêu biểu là mô hình các trường điều kiện ngẫu nhiên (CRFs) trong đó có nghiên cứu của McCallum và cộng sự [13].

Đã có rất nhiều hệ thống nhận dạng thực thể được xây dựng, ví dụ như hệ thống nhận dạng thực thể online được xây dựng bởi đại học Stanford, chúng ta có thể tìm hiểu tại địa chỉ http://nlp.stanford.edu:8080/ner . Một ví dụ được thực hiện có kết quả như sau:



Hình 1.2 Ví dụ về hệ thống nhận dạng thực thể Tiếng Anh

1.2.2 Các nghiên cứu về nhận dạng thực thể trong Tiếng Việt

Tương tự các nghiên cứu trên thế giới, các nghiên cứu về bài toán nhận diện thực thể trong tiếng Việt cũng sử dụng hai hướng tiếp cận là sử dụng luật và áp dụng các phương pháp học máy. Bên cạnh một số nghiên cứu sử dụng luật, hầu hết các nghiên cứu tập trung vào các phương pháp học máy, trong đó chủ yếu dựa trên học có giám sát và học bán giám sát. Các nghiên cứu nổi bật gần đây sử dụng học có giám sát thường áp dụng mô hình CRFs. Nổi bật như nghiên cứu của tác giả Nguyễn Cẩm Tú và cộng sự (năm 2005)[20] về bài toán nhận diện thực thể thực nghiệm trên tám kiểu thực thể cơ bản sử dụng CRFs và đạt được kết quả cao trong miền dữ liệu tiếng Việt (độ chính xác đạt 83,69%, độ hồi tưởng đạt 87,41% và độ đo F1 đạt 85,51%). Hệ thống cho kết quả với một ví dụ như sau:

Cao Xumin , Chủ tịch **Phòng Thương mại Xuất Nhập** khẩu thực phẩm của **Trung Quốc** , cho rằng , cách xem xét của **DOC** khi đem so sánh giá tôm của **Trung Quốc** với giá tôm của **Ấn Độ** là vi phạm luật thương mại .

Để đảm bảo lợi ích của **Nhà nước** và doanh nghiệp, sau thời điểm bàn giao tài sản , **VMS** mới có thể tiến hành kiểm kê và thuê tổ chức tư vấn xác định giá trị doanh nghiệp .

Hiệp hội chất lượng **Thượng Hải** đã phỏng vấn **2.714** khách hàng ở **29** siêu thị quanh thành phố trong tháng qua.

Thủ tướng **Trung Quốc Ôn Gia Bảo** vừa cho biết , **năm nay** nước này sẽ giảm tốc độ tăng trưởng kinh tế xuống còn **8%** so với con số **9,4%** trong **năm 2004** nhằm đạt được sự phát triển ổn định hơn .

Hồi **tháng 12 năm ngoái** , Tổng thống **Mỹ George Bush** , người tháo ngòi cuộc chiến tranh thép với **EU** và một số nước châu **Á** , cũng đã phải dỡ bỏ thuế suất cao sau nhiều lần **WTO** đưa ra lời cảnh cáo .

Hình 1.3 Ví dụ về hệ thống nhận dạng thực thể Tiếng Việt

Tổng kết chương 1

Chương này giới thiệu bài toán nhận dạng thực thể áp dụng trong văn bản Tiếng Việt và những nghiên cứu đã được thực hiện cho bài toán nhận dạng thực thể cho Tiếng Anh, Tiếng Việt và các nghiên cứu áp dụng cho văn bản ngắn

Chương 2. Học suốt đời và mô hình trường ngẫu nhiên có điều kiện

Chương này luận văn sẽ trình bày chi tiết về việc sử dụng mô hình trường ngẫu nhiên để giải quyết bài toán nhận dạng thực thể trong văn bản ngắn Tiếng Việt. Bên cạnh đó, luận văn cũng sẽ trình bày về học suốt đời, phương pháp áp dụng mô hình học suốt đời kết hợp với mô hình trường ngẫu nhiên có điều kiện nhằm nâng cao hiệu suất của việc học cũng như giải quyết những thách thức mà văn bản ngắn Tiếng Việt mang lại.

2.1 Mô hình trường ngẫu nhiên có điều kiện áp dụng cho bài toán nhận dạng thực thể

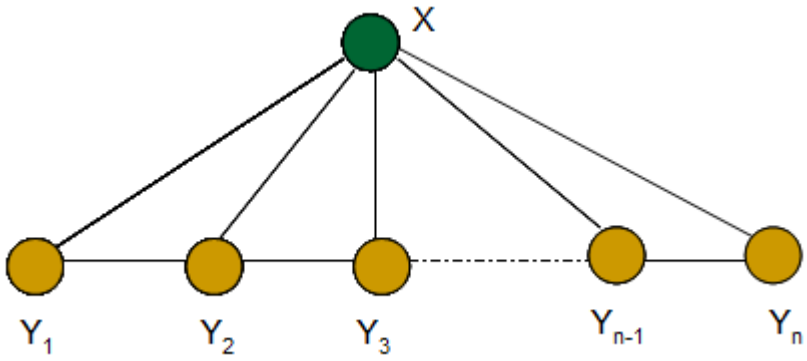
2.1.1 Khái niệm mô hình trường ngẫu nhiên có điều kiện

Có rất nhiều hướng tiếp cận nhằm giải quyết bài toán nhận dạng thực thể như phương pháp thủ công, các phương pháp học máy như mô hình Markov ẩn(HMM)[12] và mô hình Markov cực đại hóa Entropy(MEMM)[12]. Các hướng tiếp cận thủ công có nhược điểm là tốn kém về mặt thời gian, công sức và không khả chuyển. Các phương pháp học máy như HMM hay MEMM tuy có thể khắc phục được nhược điểm của phương pháp tiếp cận thủ công nhưng lại gặp phải một số vấn đề do đặc thù của mỗi mô hình.

Mô hình trường ngẫu nhiên có điều kiện (Conditional Random Fields, CRFs) là mô hình dựa trên xác suất điều kiện được đề xuất bởi J.Laffety và các cộng sự (năm 2001)[11] chúng có thể tích hợp được các thuộc tính đa dạng của chuỗi dữ liệu quan sát nhằm hỗ trợ cho quá trình phân lớp. Tuy nhiên CRFs là các mô hình đồ thị vô hướng. Điều này cho cho phép CRFs có thể định nghĩa phân phối xác suất cho toàn bộ chuỗi trạng thái với điều kiện biết chuỗi quan sát cho trước. Ta có một số qui ước kí hiệu như sau[11]:

- *X, Y, Z,...* kí hiệu các biến ngẫu nhiên
- ***x ,y ,f ,g ,...*** kí hiệu các vector như vector biểu diễn chuỗi các dữ liệu quan sát, vector biểu diễn chuỗi các nhãn.
- x_i, y_i**...** kí hiệu một thành phần trong một vector.
- *x,y,...* kí hiệu các giá trị đơn như một dữ liệu quan sát hay một trạng thái
- *S*: Tập hữu hạn các trạng thái của một mô hình CRFs.

Với $X = (X_1,X_2...X_n)$: biến ngẫu nhiên nhận các giá trị là chuỗi cần phải gán nhãn, $Y=(Y_1,Y_2,...,Y_n)$ là biến ngẫu nhiên nhận giá trị là chuỗi nhãn tương ứng. Ta có đồ thị sau[11]:



Hình 2.1 Đồ thị biểu diễn mô hình CRFs

Đồ thị vô hướng không có chu trình $G=(V,E)$. Các đỉnh V biểu diễn các thành phần của biến ngẫu nhiên Y sao cho tồn tại ánh xạ một-một giữa một đỉnh và một thành phần của Y_v của Y. Ta có $(Y|X)$ là một trường ngẫu nhiên điều kiện(CRFs) với điều kiện X, các biến ngẫu nhiên Y_v tuân theo tính chất Markov đối với đồ thị G[20]:

$$p_x(\mathrm{x}\mid \mathrm{y}) = \frac{1}{Z(\mathrm{x})}\exp\left[\sum_{t=1}^{T}\sum \lambda_k f_k(\mathrm{y}_{t-1},\mathrm{y}_t,\mathrm{x},t)\right]$$

Trong đó ta có:

- $Z(o)$ là thừa số chuẩn hóa, đảm bảo tổng các xác suất luôn bằng 1.
- λ_k là trọng số chỉ mức độ biểu đạt thông tin của thuộc tính f_k, chúng ta chỉ lựa chọn những dữ liệu có ý nghĩa trong văn bản.
- f_k là thuộc tính của chuỗi dữ liệu quan sát, có 2 loại thuộc tính như sau:
    - Thuộc tính chuyển hay còn gọi là Label-Label (LL) (ứng với một cạnh của đồ thị trong hình1) có công thức như sau[16]:

$$f_{ij}^{LL}(y_{t-1},y_t) = 1\{y_t = i\}\,1\{y_{t-1} = j\}, \forall\, i\,, j\ \in Y$$

    - Thuộc tính trạng thái hay còn gọi là Label-Word(ứng với một đỉnh của đồ thị trong hình 1) có công thức như sau[16]:

$$f_{iv}^{LW}(y_t, x_t) = 1\{y_t = i\}\ 1\{x_t = v\}, \forall\ i \in Y\ , \forall v\ \in V$$

Trong đó V là tập từ vựng, thuộc tính trên trả về giá trị bằng 1 khi từ thứ t là v và nhãn của từ thứ t là i - nhãn được gán cho từ v. x_t là từ hiện tại và được biểu diễn bằng một vec tơ đa chiều. Mỗi chiều của vec tơ là một thuộc tính của x_t.

Theo như nghiên cứu của Jakob và Gurevych [9], một từ sẽ được biểu diễn bởi một tập đặc trưng như sau:

$$\{W, -1W, +1W, P, -1P, +1P, G\}$$

Trong đó:

- W là từ đang xét, P là từ loại của nó
- -1W là từ liền trước và -1P là từ loại của nó
- +1W là từ liền sau và +1P là từ loại của nó
- G là thuộc tính phụ thuộc tổng quát

Ta có hai loại thuộc tính LW: Label-dimension và Label-G.

Label-dimension cho 6 thuộc tính đầu tiên và được định nghĩa như sau:

$$f_{iv^d}^{Ld}(y_t, x_t) = 1\{y_t = i\}\ 1\{x_l^d = v^d\}, \forall\ i \in Y\ , \forall v^d\ \in V^d$$

V^d là tập các giá trị quan sát được trong thuộc tính $d\ \in \{W, -1W, +1W, P, -1P, +1P\}$. Thuộc tính trên trả lại giá trị bằng 1 nếu thuộc tính d của x_t bằng với các giá trị của v^d và nhãn của từ thứ t bằng i.

Em sẽ trình bày thuộc tính Lable-G ở phần sau, đây là một thuộc tính quan trọng cho việc áp dụng học suốt đời cho mô hình CRFs (L-CRFs).

2.1.2 Ước lượng tham số cho mô hình

Mô hình CRFs hoạt động theo nguyên lý cực khả năng (likelihood):

Nguyên lý cực đại likelihood: "các tham số tốt nhất của mô hình là các tham số làm cực đại hàm likelihood"

Việc huấn luyện mô hình CRFs được thực hiện bằng việc xác định: $\theta(\lambda_1, \lambda_2, ..., \lambda_n)$ là các tham số của mô hình bằng việc cực đại hóa logarit của hàm likelihood của tập huấn luyện $D= (x_k,l_k)\ k = 1...$N[9]:

$$\ell = \sum_{j=1}^{N} \log(p_\theta(l^{(j)}, \mathrm{x}^{(j)})) - \sum_k \frac{\lambda_k^2}{2\sigma^2}$$

Các tham số cực đại hàm likelihood đảm bảo rằng dữ liệu mà chúng ta quan sát được trong tập huấn luyện sẽ nhận được xác suất cao trong mô hình. Nói cách khác, các tham số làm cực đại hàm likelihood sẽ làm phân phối trong mô hình gần nhất với phân phối thực nghiệm trong tập huấn luyện.

2.1.3 Tìm chuỗi nhãn phù hợp nhất

Thuật toán Viterbi được sử dụng để tìm chuỗi y^* mô tả tốt nhất cho chuỗi dữ liệu quan sát x:

$$y^* = \arg\max_{y^*} P(y|x).$$

Ta có: $\partial_t(y_t)$: xác suất của chuỗi trạng thái có độ dài t kết thúc bởi trạng thái s_t với chuỗi quan sát là o. Với $\partial_0(y_t)$ là xác suất tại điểm bắt đầu của mỗi trạng thái y[18].

$$\partial_t(y_t) = max_{y_j}\{\partial_t(y_j) exp(\sum \lambda_k f_k(y_j, y_t, \mathbf{x}, t))\}$$

Bằng cách tính như trên ta sẽ dừng thuật toán khi $t = T-1$,và $p^* = argmax(\partial_t(s_t))$. Từ đó ta có thể quay lại và tìm được chuỗi s^* tương ứng.

2.2 Thuộc tính phụ thuộc tổng quát (G)

Thuộc tính G sử dụng các mối quan hệ phụ thuộc tổng quát, chúng ta sẽ tìm hiểu tại sao thuộc tính này có thể cho phép L-CRFs sử dụng các kiến thức trong quá khứ tại thời điểm kiểm tra để làm tăng độ chính xác. Giá trị của thuộc tính này được thể hiện thông qua một mẫu phụ thuộc (dependency pattern), được khởi tạo từ các mối quan hệ phụ thuộc.

Thuộc tính phụ thuộc tổng quát (G) của x_t là một tập các giá trị V^G. Mỗi thuộc tính v^G là một mẫu phụ thuộc. Label-G được định nghĩa như sau[16]:

$$f_{iv^G}^{LG}(y_t, x_t) = 1\{y_t = i\}\, 1\{x_l^G = v^G\}, \forall\, i \in Y\,, \forall v^G \in V^G$$

Hàm trên sẽ trả lại giá trị bằng 1 nếu thuộc tính phụ thuộc của biến x_t bằng với mẫu v^G và y_t có nhãn là i.

Các mối quan hệ phụ thuộc đã được thể hiện rằng rất hữu ích trong các ứng dụng phân tích ngữ nghĩa[9]. Một mối quan hệ phụ thuộc được định nghĩa như sau:

(type, gov, govpos, dep,deppos)

Trong đó:

- type: Loại quan hệ
- gov: governor word, govpos là từ loại của nó
- dep: từ phụ thuộc, deppos là từ loại của nó

Từ thứ t có thể là governor word hoặc từ phụ thuộc trong một mối quan hệ phụ thuộc.

Ta có một số loại quan hệ phụ thuộc như sau[4]

- nsubj (nominal subject) là một cụm danh từ được dùng làm chủ ngữ của một mệnh đề, từ chủ đề (governor word) không phải lúc nào cũng là động từ khi từ đó là một động từ phổ biến hoặc bổ sung cho một động từ phổ biến
  Ví dụ: "Việt Nam đánh bại Mỹ" => nsubj(đánh bại, Việt Nam)
- det(determiner) là mối quan hệ giữa đầu của 1 cụm danh từ và từ xác định của nó
  Ví dụ: "Điện thoại này rất đẹp" => det(Điện thoại, này)
- cop(copula): là mối quan hệ giữa hai động từ hoặc động từ và tính từ
  Ví dụ: "Nam là học sinh giỏi" => cop(giỏi,là)
- num(number): là mối quan hệ giữa số từ và danh từ, bổ nghĩa cho danh từ
  Ví dụ: "Nhà có 3 cửa sổ" => num(nhà,3)
- cc(coordination): là mối quan hệ giữa một phần tử của 1 liên kết và từ nối của nó
  Ví dụ: "Nam học giỏi và thông minh" => cc(giỏi,và)
- nmod(nominal modifiers): được sử dụng cho các biến tố của danh từ hoặc bổ ngữ của danh từ
  Ví dụ: "Quận Cầu Giấy của Hà Nội" => nmod(Cầu Giấy, Hà Nội)
  Có thể tham khảo thêm nhiều loại quan hệ tại:
  https://nlp.stanford.edu/software/dependencies_manual.pdf

2.3 Định nghĩa học suốt đời

Học máy suốt đời (LML) hoặc học suốt đời (LL) đã được đề xuất vào năm 1995 bởi Thrun và Mitchell [17, 18]. Thrun đã phát biểu rằng các mối quan tâm khoa học phát sinh trong học tập suốt đời là việc sử dụng lại, trình bày và chuyển giao kiến thức về miền [14]. Trong những năm gần đây của cuộc cách mạng công nghiệp thứ tư, học máy suốt đời trở thành một mô hình học máy nổi lên nhờ vào khả năng sử dụng kiến thức từ các nhiệm vụ trong quá khứ cho nhiệm vụ hiện tại. Kể từ khi khái niệm học suốt đời được đề xuất, nó đã được nghiên cứu trong bốn lĩnh lực chính: Học giám sát suốt đời, học không giám sát suốt đời, học bán giám sát suốt đời và học tăng cường suốt đời.

Định nghĩa ban đầu của LML [18] được phát biểu như sau: Cho một hệ thống đã thực hiện N bài toán. Khi gặp bài toán thứ N+1, nó sử dụng tri thức thu được từ N bài toán để trợ giúp bài toán N+1. Zhiyuan Chen và Bing Liu đã mở rộng định nghĩa này bằng cách bổ sung thêm một cơ sở tri thức (Knowledge base: KB) hiện để nhấn mạnh tầm quan trọng của việc tích lũy tri thức và chuyển đổi các tri tức mức độ cao hơn được thêm vào từ tri thức thu được trong quá trình học trước đó.

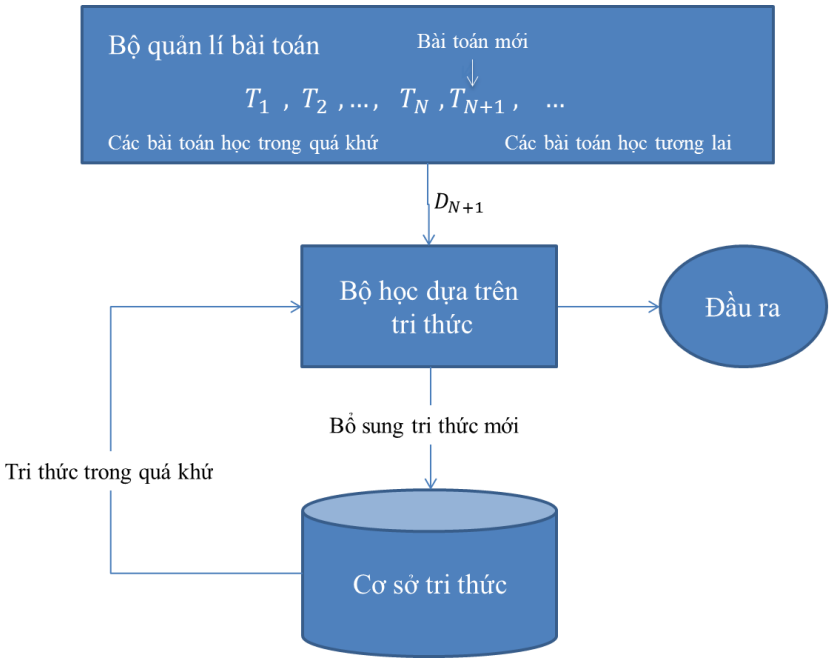
Định nghĩa (Học máy suốt đời (Lifelong Machine Learning: LML)) [21] : Học máy suốt đời là một quá trình học liên tục. Tại thời điểm bất kỳ, bộ học đã thực hiện một chuỗi N bài toán học, $T_1, T_2, \dots, T_N$. Các bài toán này, còn được gọi là các bài toán trước (previous tasks) có các tập dữ liệu tương ứng là $D_1, D_2, \dots, D_N$. 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 gặp bài toán thứ N+1, T_{N+1} (được gọi là bài toán mới hoặc bài toán hiện tại) với dữ liệu D_{N+1} bộ học có thể tận dụng tri thức quá khứ trong cơ sở tri thức (KB) để giúp học bài toán T_{N+1} . Lưu ý rằng bài toán có thể được cung cấp hoặc phát hiện bởi chính hệ thống. Mục tiêu của LML thường là tối ưu hóa hiệu năng của bài toán mới 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. 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 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 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.

Nhóm tác giả đã đưa ra một số nhận xét (được xếp theo ưu tiên) nhằm làm rõ hơn các nội dung của định nghĩa như sau[21]:

1. Định nghĩa cho thấy LML có ba đặc điểm chính: (1) học liên tục, (2) tích lũy và duy trì tri thức trong cơ sở tri thức (KB), (3) khả năng sử dụng tri thức quá khứ để giúp việc học tương lai.
2. Do các bài toán không cùng một miền, không có định nghĩa thống nhất về miền (domain) trong tài liệu có khả năng áp dụng cho tất cả lĩnh vực. Trong hầu hết các trường hợp, thuật ngữ được sử dụng một cách "gần gũi" (không chính thống) để chỉ một cài đặt có không gian đặc trưng cố định, có thể có nhiều bài toán khác nhau cùng loại hoặc các loại khác nhau (ví dụ, trích xuất thông tin, liên kết thực thể).
3. Việc chuyển sang bài toán mới có thể xảy ra đột ngột hoặc từng bước, các bài toán và dữ liệu của chúng không cần phải được cung cấp bởi một số hệ thống bên ngoài hoặc người sử dụng. Lý tưởng nhất là bộ học suốt đời sẽ tìm ra các bài toán học và huấn luyện dữ liệu của nó trong quá trình tương tác với môi trường bằng cách thực hiện học tự khuyến khích.
4. Các bài báo hiện tại chỉ dùng một hoặc hai kiểu đặc trưng của tri thức phù hợp với kỹ thuật được đề xuất. Bài toán biểu diễn tri thức vẫn là một chủ đề nghiên cứu tích cực. Định nghĩa cũng không chỉ rõ cách duy trì và cập nhật cơ sở tri thức như thế nào. Đối với một ứng dụng cụ thể, người ta có thể thiết kế một KB dựa trên nhu cầu ứng dụng.
5. Định nghĩa cho thấy LML có thể yêu cầu một cách tiếp cận hệ thống (systems approach) kết hợp nhiều thuật toán học và các sơ đồ biểu diễn tri thức khác nhau. Không có khả năng một thuật toán học duy nhất có thể đạt được mục tiêu của LML.
6. Hiện nay không có hệ thống LML chung nào có thể áp dụng LML trong mọi miền ứng dụng với mọi loại bài toán có thể xảy ra. Trên thực tế chúng ta còn ở rất xa với điều đó. Đó là, không giống như nhiều thuật toán học máy như SVM và học sâu, có thể được áp dụng cho bất kỳ bài toán học nào miễn là dữ liệu được biểu diễn theo một định dạng cụ thể. Các thuật toán LML hiện nay vẫn còn khá riêng biệt đối với một số loại bài toán và dữ liệu.

2.4 Kiến trúc hệ thống học suốt đời

Từ định nghĩa và các nhận xét ở trên, chúng ta có thể phác thảo một quá trình tổng quát và một kiến trúc hệ thống của LML như Hình 2.1:



Hình 2.2 Kiến trúc hệ thống học suốt đời

Kiến trúc tổng quát này chỉ có mục đích minh hoạ. Không phải tất cả hệ thống hiện tại đều sử dụng tất cả các thành phần hoặc tiểu phần trong kiên trúc này. Trên thực tế, hầu hết các hệ thống hiện tại đơn giản hơn nhiều. Kiến trúc bao gồm các thành phần chính như sau:

1. **Cơ sở tri thức** (Knowledge Base: KB): Nó chủ yếu chứa tri thức đã học được từ các bài toán trước. KB gồm một số thành phần con như sau:

a) *Kho thông tin quá khứ* (Past Information Store: PIS): lưu thông tin kết quả từ việc học trong quá khứ, bao gồm: các mô hình kết quả, mẫu hoặc các dạng kết quả khác. 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, (3) mô hình hoặc các mẫu cuối cùng 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.

Đối với một hệ thống cụ thể, người dùng phải quyết định những gì cần giữ lại để trợ giúp việc học trong tương lai.

b) *Bộ khai phá siêu tri thức* (Meta-Knowledge Miner: MKM). Nó thực hiện việc khai phá các siêu tri thức trong kho thông tin quá khứ và trong kho siêu tri thức (xem bên dưới). Chúng tôi gọi đây là siêu khai phá (meta-mining) vì nó khai phá tri thức mức cao từ tri thức đã lưu trữ. Tri thức kết quả được lưu trong kho siêu tri thức (Meta-Knowledge Store). Tại đây nhiều thuật toán khai phá có thể sử dụng để tạo ra các kiểu kết quả khác nhau.

c) *Kho siêu tri thức* (Meta-Knowledge Store: MKS): Lưu các tri thức được khai phá hoặc củng cố từ kho thông tin quá khứ (PIS) và từ chính kho siêu tri thức (MKS). Một vài sơ đồ biểu diễn tri thức phù hợp thực sự cần thiết đối với mỗi ứng dụng.

d) *Bộ suy luận tri thức* (Knowledge Reasoner: KR): Nó thực hiện suy luận dựa trên tri thức trong MKB và PIS để tạo ra nhiều tri thức. Hầu hết các hệ thống hiện tại không có thành phần con này. Tuy nhiên, với sự tiến bộ của LML, thành phần này trở nên ngày càng quan trọng.

Như đã nêu ở trên, hiện nay nghiên cứu về LML còn rất mới, chưa có hệ thống nào có tất cả các thành phần con này.

2. **Bộ học dựa trên tri thức** (Knowledge-Based Learner: KBL): Đối với LML, bộ học cần có tri thức tiên nghiệm để học. Chúng tôi gọi bộ học như vậy là bộ học dựa trên tri thức, nó có khả năng tận dụng tri thức trong KB để học bài toán mới. Bộ học dựa trên tri thức có thể có hai thành phần con:

(1) Bộ khai phá tri thức bài toán (Task knowledge miner: TKM), sử dụng tri thức thô hoặc thông tin trong KB để khai phá hoặc xác định những tri thức phù hợp với bài toán hiện thời. Đây là điều cần thiết vì trong một số trường hợp, bộ học dựa trên tri thức không thể sử dụng trực tiếp tri thức thô trong KB mà cần tri thức đặc tả bài toán và tri thức tổng quát hơn được khai phá từ KB.

(2) Bộ học có thể sử dụng tri thức được khai phá vào việc học.

3. **Đầu ra** (Output): Đây là kết quả học cho 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, một chính sách trong học tăng cường, v.v.

4. **Bộ quản lý bài toán** (Task Manager: TM): Nó nhận và quản lý các bài toán đến hệ thống và xử lý sự thay đổi bài toán và giới thiệu bài toán học mới cho bộ học dựa trên tri thức theo cách suốt đời.

Quá trình học suốt đời: Một quá trình học suốt đời điển hình bắt đầu với Bộ quản lý bài toán chỉ định một bài toán mới cho KBL. Sau đó KBL làm việc với sự trợ giúp của tri thức quá khứ trong KB để tạo ra kết quả (ví dụ như một mô hình) cho người dùng và cũng gửi tới KB các thông tin hoặc tri thức cần giữ lại để sử dụng trong tương lai.

Đối với LML, 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. Nhóm tác giả nêu bật hai thách thức tiềm ẩn nhưng cơ bản của LML dựa trên những kinh nghiệm của chúng tôi ở một số dự án. Chúng tôi sẽ mô tả cách nghiên cứu hiện tại đối phó với những thách thức này trong suốt cuốn sách này.

1. 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.
2. 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. Một lần nữa cho thấy, con người khá giỏi nhận ra ngữ cảnh thích hợp với một mẩu tri thức.

2.5 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 thực nghiệm một thuật toán LML trong nghiên cứu hiện nay thường được thực hiện bằng cách sử dụng các bước sau đây:

1. *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). Rõ ràng, có thể thực nghiệm với nhiều biến thể hoặc phiên bản của thuật toán (ví dụ: sử dụng các kiểu tri thức khác nhau hoặc tri thức được sử dụng ít hay nhiều).

2. *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).
3. *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.
4. *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.

Một số chú ý bổ sung trong thực hiện đánh giá thực nghiệm LML:

1. *Một lượng lớn các bài toán:* Để đánh giá thuật toán LML cần một lượng lớn các bài toán và tập dữ liệu. Điều này thực sự cần thiết do tri thức thu được từ một vài bài toán có thể không cải tiến việc học của bài toán mới vì tri thức thu được từ mỗi bài toán này có thể chỉ cung cấp một lượng rất nhỏ tri thức có ích đối với bài toán mới (trừ khi tất cả các bài toán rất giống nhau) và dữ liệu của bài toán mới thường khá nhỏ.
2. *Trình tự bài toán*: Thứ tự thực hiện các bài toán cần học có thể có ý nghĩa nhất định nào đó, nghĩa là thứ tự thực hiện các bài toán khác nhau có thể tạo ra các kết quả khác nhau. Nguyên nhân là các thuật toán LML điển hình không đảm bảo các giải pháp tối ưu cho tất cả các bài toán trước đó. Để xem xét hiệu quả của thứ tự thực hiện các bài toán trong thực nghiệm, người ta có thể thử ngẫu nhiên thứ tự một số bài toán và tạo ra các kết quả cho từng trình tự đó. Sau đó, tổng hợp các kết quả cho các mục đích so sánh. Các bài báo hiện nay chủ yếu chỉ sử dụng một trình tự ngẫu nhiên trong các thực nghiệm của họ.
3. *Tiến hành thực nghiệm*: Vì nhiều bài toán trước đó hướng tới việc tạo ra nhiều tri thức, nhiều tri thức hơn có thể làm cho thuật toán LML tạo ra các kết quả tốt hơn cho bài toán mới. Điều này cho thấy rằng mong muốn thuật toán chạy trên bài toán mới khi số lượng các bài toán trước tăng lên.

2.6 Học giám sát suốt đời

Dựa trên định nghĩa chung của LML ở phần 2.1. Ta có định nghĩa Học giám sát suốt đời như sau:

Định nghĩa [21]: 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 T_1, T_2,… T_N, và giữ lại tri thức đã học được trong cơ sở tri thức (KB). Khi một bài toán mới 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 D_{N+1} của T_{N+1}. Sau khi học T_{N+1}, KB cũng được cập nhật các tri thức đã học được từ T_{N+1}.

Học giám sát suốt đời bắt đầu từ bài báo của Thrun [14] với đề xuất một vài phương pháp LML ban đầu trong ngữ cảnh học theo ghi nhớ (memory-based learning) và mạng nơ-ron. Cách tiếp cận mạng nơ-ron đã được Silver và cộng sự cải tiến năm 2015[15]. 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 [7], một hình thức đặc biệt của LML được gọi là học tích luỹ được đề xuất. Tương tự như các công trình trên, mỗi bài toán mới được trình bày với một lớp dữ liệu mới cần phải học được. Tuy nhiên, không giống như các công trình trên, hệ thống chỉ duy trì một mô hình phân lớp đa lớp duy nhất ở mọi thời điểm. Khi một lớp mới xảy đến, mô hình được cập nhật để phân lớp tất cả các lớp quá khứ và lớp mới. Vì vậy hình thức học này có tên gọi là học tích lũy. . Nhóm tác giả Fei và cộng sự [7] cũng đề xuất một phương pháp học dựa trên không gian tương tự để phát hiện các lớp mới chưa được nhìn thấy trong quá trình huấn luyện. Ruvolo và Eaton đề xuất thuật toán ELLA cải tiến phương pháp học đa nhiệm GO-MTL [10] để làm cho nó trở thành một phương pháp Học suốt đời. Chen và cộng sự [14] đề xuất thêm một kỹ thuật trong ngữ cảnh phân lớp Naïve Bayesian.

2.7 Áp dụng học suốt đời vào mô hình trường ngẫu nhiên có điều kiện

Như đã được trình bày ở phần trên, chúng ta không thể thay đổi mô hình khi nó đã được xây dựng và áp dụng đối với học máy giám sát. Vậy làm cách nào để chúng ta có thể tăng hiệu suất của mô hình mà không phải thay đổi mô hình sẵn có?

Ý tưởng chính của phương pháp này là chúng ta sẽ tập trung khai phá các mối quan hệ phụ thuộc hay các mẫu phụ thuộc trong quá trình áp dụng mô hình CRFs cho một miền mới. Thực thể sẽ được gán một nhãn tri thức là "A" để đánh dấu là một thực thể tiềm năng nếu như thực thể ở cùng mẫu quan hệ phụ thuộc với nó xuất hiện trong cơ sở tri thức và được gán nhãn là "O" cho trường hợp ngược lại. Thuộc tính phụ thuộc tổng quát (Label-G) là một trong hai loại thuộc trính trạng thái (Lable-World) được sử dụng trong mô hình CRFs, giá trị của thuộc tính này được khởi tạo từ các mối quan hệ phụ thuộc. Như vậy, các mối quan hệ phụ thuộc với các nhãn tri thức "A" hoặc "O" chính là cầu nối giữa mô hình và dữ liệu, giúp tăng hiệu suất của viêc nhận dạng thực thể mà không cần phải thay đổi mô hình sẵn có.

Chi tiết về phương pháp áp dụng học suốt đời vào mô hình CRFs cũng như cách xây dựng các mẫu phụ thuộc dựa vào cơ sở tri thức sẽ được trình bày ở chương 3.

Tổng kết chương 2

Chương này đã giới thiệu khái niệm mô hình trường ngẫu nhiên có điều kiện, ước lượng tham số cho mô hình cũng như bài toán gán nhãn cho dữ liệu dạng chuỗi. Bên cạnh đó, chương này cũng những kiến thức cơ bản nhất về học suốt đời bao gồm: định nghĩa về học suốt đời, kiến trúc của mô hình học suốt đời và những chi tiết các thành phần của kiến trúc, cách đánh giá bài toán áp dụng mô hình học suốt đời, trình bày tổng quát về ý tưởng nhằm áp dụng học suốt đời để nâng cao hiệu quả của mô hình mà không cần phải thay đổi mô hình sẵn có. Chương sau luận văn sẽ trình bày chi tiết về vấn đề áp dụng học suốt đời vào bài toán nhận dạng thực thể trong văn bản Tiếng Việt.

Chương 3. Mô hình học suốt đời áp dụng vào bài toán nhận dạng thực thể

Chương này luận văn sẽ giới thiệu về việc áp dụng mô hình học suốt đời áp dụng vào bài toán nhận dạng thực thể, cụ thể là áp dụng kết hợp với mô hình CRF. Nội dung của chương sẽ nhấn mạnh về vấn đề kết quả của CRFs sẽ được cải thiện bằng cách sử dụng các tri thức trước đó từ các kết quả nhận được khi áp dụng cho các miền khác. Trước hết luận văn trình bày về mẫu phụ thuộc – "chìa khóa" cho việc nâng cao hiệu quả của mô hình CRFs áp dụng học chuyển đổi.

3.1 Mẫu phụ thuộc

Chúng ta khởi tạo các mối quan hệ phụ thuộc sử dụng các bước dưới đây:

1. Với mỗi quan hệ phụ thuộc, thay từ hiện tại (governor word hoặc từ phụ thuộc) và từ loại của nó bằng một ký tự đại diện khi chúng ta đã có thuộc tính từ (W) và từ loại (P) như thuộc tính Label-dimension đã trình bày ở phần trước.
2. Thay thế từ ngữ cảnh (context word) – từ khác với từ thứ t trong mỗi mối quan hệ phụ thuộc bằng một nhãn tri thức để tạo thành một mẫu quan hệ tổng quát. Ta có tập các khía cạnh được chú thích trong dữ liệu huấn luyện là K^t, nếu từ ngữ cảnh xuất hiện trong K^t, chúng ta sẽ thay nó bằng một nhãn "A" (aspect) và "O" (other) cho trường hợp ngược lại.

   Ví dụ: Chúng ta đang làm việc với câu sau:

   "Chất lượng camera của Samsung ngày càng tuyệt vời"

   Giả sử "Samsung" đã được trích xuất trong các lần học trước đó và lưu vào K^t. Ta có mẫu phụ thuộc tổng quát của camera như sau:

(nmod,camera,NN,Samsung,NN) → (nmod,*,Samsung,NN) → (nmod,*,A,NN)

Sau đây, luận văn sẽ trình bày thêm về vấn đề tại sao mẫu phụ thuộc lại có thể làm tăng tính chính xác của mô hình CRFs bằng việc sử dụng các kiến thức trong quá khứ. Điều then chốt ở đây là nhãn tri thức "A" được dùng để đánh dấu một thực thể là tiềm năng. Quay lại bài toán trích xuất thực thể định danh từ 1 miền mới D_{N+1} sử dụng mô hình M đã được huấn luyện trước đó, chúng ta đã thực hiện nhận dạng thực thể từ nhiều miền trước $D_1, D_2, \dots, D_N$ và lưu lại tập các thực thể trích xuất được $A_1, A_2, \dots, A_N$. Sau đó chúng ta có thể khai thác các thực thể tin cậy và thêm chúng vào K^t, cho phép có nhiều nhãn kiến thức cho các mẫu phụ thuộc của dữ liệu mới A_{N+1} do có sự chia sẻ các thực thể giữa các miền. Điều này làm phong phú thêm các thuộc tính mẫu phụ thuộc, cho phép trích xuất được nhiều thực thể hơn từ miền D_{N+1}.

3.2 Thuật toán L-CRF

Các mẫu phụ thuộc cho thuộc tính phụ thuộc tổng quát không sử dụng bất kỳ từ thực tế nào và chúng cũng có thể sử dụng kiến thức trước, chúng khá mạnh để nhận dạng thực thể chéo miền (miền thử nghiệm không được sử dụng trong đào tạo).

Gọi K là tập các khía cạnh tin cậy được khai thác từ các thực thể được trích xuất trong bộ dữ liệu của các miền trước sử dụng mô hình CRFs (M). Lưu ý rằng chúng ta giả sử rằng M đã được huấn luyện sử dụng dữ liệu đã được gán nhãn D_t. Ban đầu, K được gán bằng K^t (tập hợp của tất cả các khía cạnh trong dữ liệu huấn luyện D_t). Càng thêm nhiều miền áp dụng mô hình M chúng ta càng có thêm nhiều dữ liệu và K ngày càng lớn. Tuy nhiên chúng ta không lấy tất cả các khía cạnh trích xuất được mà chỉ lấy những khía cạnh đáng tin cậy. Khía cạnh đáng tin cậy thỏa mãn 2 tiêu chí:

- Xuất hiện trong nhiều miền
- Tần suất xuất hiện trong một miền lớn hơn 1 ngưỡng nhất định.

Khi cần thực hiện nhận dạng thực thể trên một miền mới D_{N+1}, K cho phép thuộc tính phụ thuộc tổng quát tạo thêm nhiều mẫu tổng quát liên quan đến các khía cạnh do có thêm nhãn tri thức 'A' như đã được giải thích trong phần trước. Do đó, CRFs có nhiều thuộc tính hơn để tạo ra kết quả tốt hơn.

L-CRFs thực hiện trong hai pha: pha huấn luyện và pha học suốt đời. Pha huấn luyện huấn luyện một mô hình CRFs M sử dụng dữ liệu huấn luyện D_t như việc huấn luyện các mô hình CRFs truyền thống khác. Trong pha học suốt đời, M được sử dụng để nhận dạng thực thể từ các miền mới(M không được thay đổi và dữ liệu của miền mới là

không được gán nhãn). Tất cả các kết quả được lưu lại vào S. Tại một thời điểm nhất định, giả sử rang M đã được áp dụng cho N miền trước đây và giờ cần thực hiện trên miền N+1. L-CRFs sử dụng M và các khía cạnh tin cậy (kí hiệu là K_{N+1},) để trích xuất từ D_{N+1}. Lưu ý rằng các khía cạnh K^t từ dữ liệu huấn luyện được coi là luôn đáng tin cậy vì chúng được gắn nhãn thủ công, do đó một tập hợp con K. Chúng ta không thể sử dụng tất cả các khía cạnh được trích xuất từ các miền trước đây như các khía cạnh đáng tin cậy do nhiều lỗi trích xuất. Nhưng những khía cạnh đó xuất hiện trong nhiều miền trước đây có nhiều khả năng là chính xác hơn như đã được trình bày ở phần trước. Vì vậy, K chứa những khía cạnh thường xuyên trong S. Pha học suốt đời được thể hiện qua thuật toán dưới đây[16]:

Đầu vào:

- Dữ liệu D_{N+1}: $O\ (o_1, o_2, \dots, o_n)$ chuỗi dữ liệu quan sát, o_i là các từ
- $L(l_1, l_2, \dots, l_n)$ chuỗi các nhãn cần gán cho dữ liệu
- Mô hình M đã được huấn luyện và áp dụng tại N miền trong quá khứ
- $S(s_1, s_2, \dots, s_m)$ tập kết quả của N miền trong quá khứ

1. $K_p \leftarrow \emptyset$
2. Loop
3. $F \leftarrow FeatureGeneration(D_{N+1}, K)$
4. $A_{n+1} \leftarrow ApplyCRFModel(M, F)$
5. $S \leftarrow S \cup \{A_{n+1}\}$
6. $K_{n+1} \leftarrow FrequentAspectsMining(S, \lambda)$
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. $S \leftarrow S - \{A_{n+1}\}$
13. end if
14. end loop

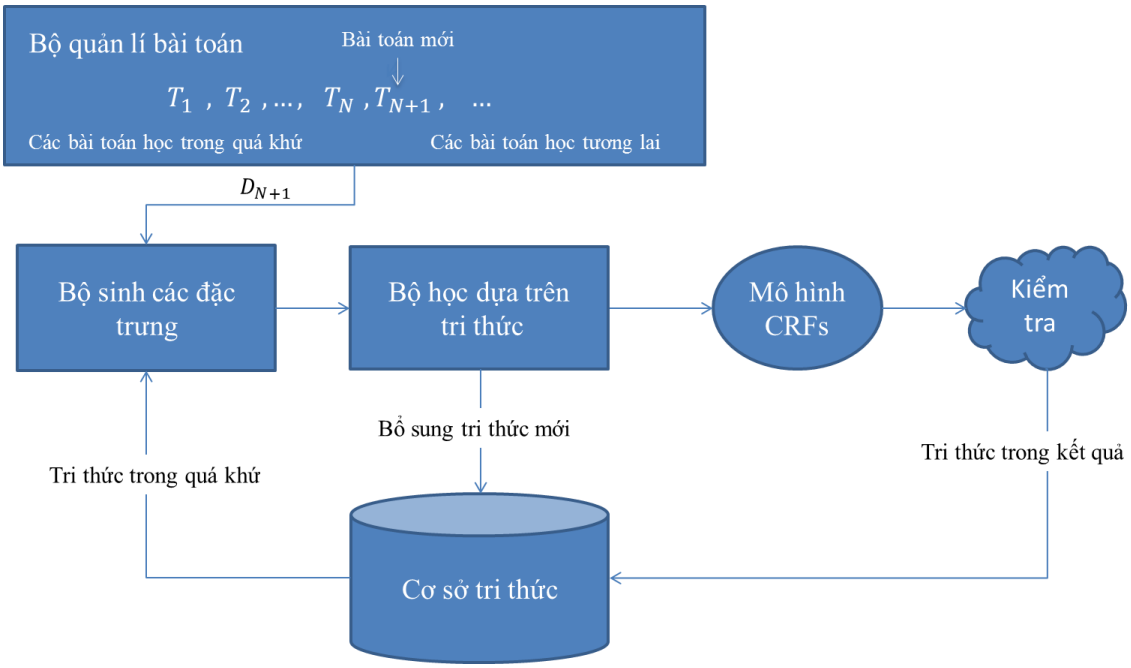
Đầu ra:

- Các câu đã được gán nhãn
- Cơ sở tri thức S đã được bổ sung kết quả từ miền D_{N+1}

Pha học suốt đời: thuật toán trên thực hiện trên tập dữ liệu của D_{N+1}lặp đi lặp lại

1. Thực hiện khởi tạo các thuộc tính (F) trên dữ liệu của D_{N+1} (dòng 3) và áp dụng mô hình CRFs M (dòng 4) trên F để trích xuất ra một tập các thực thể A_{N+1}
2. A_{N+1} được thêm vào S (lưu các thực thể đã được khai thác trong quá khứ). Từ S, chúng ta khai thác một loạt các khía cạnh thường xuyên K_{n+1}. Ngưỡng tần số là λ.
3. Nếu K_{n+1} giống với K_p ở lần lặp trước, thuật toán sẽ được dừng vì không tìm thấy các thực thể mới. Chúng ta lặp đi lặp lại quy trình này vì mỗi lần trích xuất mang lại kết quả mới, có thể làm tăng kích thước của K, các khía cạnh đáng tin cậy trong quá khứ hoặc kiến thức trong quá khứ. K tăng có thể tạo ra các mẫu phụ thuộc nhiều hơn, có thể cho phép nhiều thực thể hơn.
4. Ngược lại: một số khía cạnh đáng tin cậy bổ sung được tìm thấy. M có thể trích xuất các khía cạnh bổ sung trong lần lặp tiếp theo. Các dòng 10 và 11 cập nhật hai tập cho lần lặp tiếp theo.

Mô hình của hệ thống NER trong văn bản Tiếng Việt áp dụng học suốt đời được thể hiện như hình 3.2 dưới đây:



Hình 3.1 Mô hình hệ thống NER trong văn bản Tiếng Việt áp dụng học suốt đời

Các thành phần chính của mô hình:

- **Bộ quản lý bài toán**: Quản lý các bài toán đã được thực hiện hay N miền đã được áp dụng mô hình M vào để nhận dạng thực thể, cung cấp dữ liệu cho bộ sinh các đặc trưng khi áp dụng mô hình M cho miền mới N+1
- **Cơ sở tri thức**: Chứa các thực thể đã nhận dạng được khi áp dụng mô hình trên N miền trong quá khứ
- **Bộ sinh các đặc trưng**: nhiệm vụ chính của bộ này là trích xuất ra các mẫu quan hệ từ dữ liệu của miền thứ N+1 kết hợp với dữ liệu trong cơ sở tri thức với nhãn tri thức "A" hoặc "O". Đầu ra của bộ này sẽ là đầu vào của bộ học dựa trên tri thức, đây chính là chìa khóa giúp tăng hiệu quả của mô hình khi áp dụng cho một miền dữ liệu mới.
- **Bộ học dựa trên tri thức**: Sử dụng các mẫu quan hệ có được từ bộ sinh các đặc trưng để nhận dạng thực thể cho một miền mới sử dụng mô hình CRFs.
- **Mô hình CRFs**: Mô hình đã được huấn luyện và áp dụng trên N miền.

Tổng kết chương 3

Chương 3 đã trình bày phương pháp nhận dạng thực thể trong văn bản Tiếng Việt áp dụng học suốt đời. Đồng thời, chương này cũng trình bày chi tiết về thuật toán để tăng cường sử dụng các kiến thức đã được học trong quá khứ nhằm tăng hiệu quả của việc học tại miền hiện tại.

Chương 4. Thực nghiệm và kết quả

Như đã trình bày ở phần trên, luận văn sẽ tiến hành thực nghiệm đánh giá phương pháp nhận dạng thực thể trong văn bản ngắn Tiếng Việt áp dụng học suốt đời và so sánh với phương pháp truyền thốn. Chương này sẽ mô tả chi tiết về quá trình tiến hành thực nghiệm cũng như kết quả thực nghiệm

4.1 Môi trường và các công cụ sử dụng

4.1.1 Cấu hình phần cứng

Thành phần	Chỉ số
CPU	Intel(R) Core(TM) i5-4210U CPU @ 2.40 GHz
RAM	8.00 GB (7.87 GB usable)
Operating System	Windows 7 Ultimate SP1 64-bit
HDD	500 GB

Bảng 4.1 Môi trường thực nghiệm

4.1.2 Các phần mềm và thư viện

Các phần mềm sử dụng

STT	Tên phần mềm	Nguồn
1	Eclipse Oxygen.2 Release (4.7.2)	http://www.eclipse.org/downloads

Bảng 4.2 Các phần mềm sử dụng

Các thư viện sử dụng

STT	Tên thư viện	Nguồn
1	JvnTexpro.jar	http://jvntextpro.sourceforge.net/
2	stanford-ner.jar	https://nlp.stanford.edu/software/CRF-NER.shtml

3	dependensee-3.7.0.jar	https://nlp.stanford.edu/software/lex-parser.shtml

Bảng 4.0.3 Các thư viện sử dụng

4.2 Dữ liệu thực nghiệm

Dữ liệu bao gồm 6 miền với 675 câu, chi tiết được thể hiện ở bảng dưới đây:

Miền	Số câu
Pháp luật	144 câu
Kinh tế	124 câu
Công nghệ thông tin	147 câu
Giáo dục	80 câu
Xã hội	98 câu
Thể thao	82 câu

Bảng 4.4 Dữ liệu thực nghiệm

Dữ liệu đã được tiền xử lý (tách câu, tách từ, gán nhãn từ loại và gán nhãn thực thể) mỗi từ được biểu diễn trên 1 dòng và hai câu được cách nhau bằng một dòng trống.

4.3 Mô tả thực nghiệm

Thực nghiệm được tiến hành theo 4 bước sau đây:

- Bước 1: Thu thập dữ liệu từ một số hệ thống hỏi đáp (diễn đàn tin học, mục hỏi đáp của trang luật Dương gia …), phân tích và tiền xử lí dữ liệu (loại bỏ từ dừng, từ xuất hiện quá nhiều hoặc quá ít).
- Bước 2: Sử dụng công cụ Jvn Textpro để tách từ và gán nhãn từ loại và gán nhãn thực thể. Sau đó tiến hành kiểm tra và gán lại nhãn thủ công cho những trường hợp sai nhằm tăng độ chính xác khi huấn luyện mô hình.
- Bước 3: Sử dụng bộ công cụ Standford CoreNLP tiến hành trích xuất các quan hệ phụ thuộc và huấn luyện mô hình CRFs dựa trên dữ liệu đã được gán nhãn và các thuộc tính trích xuất được(tương ứng với pha huấn luyện mô hình như đã trình bày ở trên)
- Bước 4: Áp dụng mô hình học suốt đời và tiến hành đánh giá thực nghiệm trên miền D_i với các kịch bản sau:

- Đánh giá nội miền: Thực hiện thực nghiệm trên 6 miền và chia dữ liệu của các miền thành 2 phần: 50% dữ liệu huấn luyện và 50% dữ liệu kiểm tra.
- Đánh giá chéo miền: Thực hiện đánh giá chéo miền với 3 kịch bản sau đây:
  - Dữ liệu kiểm tra là D_i và dữ liệu huấn luyện là dữ liệu của các miền còn lại (khác D_i)
  - Dữ liệu kiểm tra là 1/2 D_i, tập dữ liệu huấn luyện gồm hai thành phần:
    - Thành phần dữ liệu từ các miền khác D_i
    - Dữ liệu từ D_i với số lượng tăng dần: 1/6 D_i , 1/4 D_i và 1/2 D_i
  - Dữ liệu kiểm tra là 1/2 D_i , dữ liệu huấn luyện là dữ liệu từ miền gần với D_i dựa theo độ đo được trình bày dưới đây.

4.4 Đánh giá

Như đã trình bày ở trên là luận văn sẽ sử dụng ba độ đo để đánh giá thực nghiệm. Mục đích của việc sử dụng ba độ đo này là giúp chúng ta có thể ước lượng được tính đáng tin cậy của mô hình nhận dạng thực thể trong văn bản ngắn Tiếng Việt áp dụng mô hình học suốt đời. Sau đây luận văn sẽ trình bày chi tiết về ba độ đo trên.

Ta có ma trận nhầm lẫn được trình bày như bảng dưới đây[1]:

		Lớp dự đoán	
		Lớp = P	Lớp = N
Lớp thực sự	Lớp = P	TP	FN
	Lớp = N	FP	TN

Bảng 4.5 Ma trận nhầm lẫn

Bảng trên thể hiện ma trận nhầm lẫn cho một phân lớp nhị phân. Tuy bài toán nhận dạng thực thể là phân lớp đa nhãn, nhưng ta vẫn có thể áp dụng bằng cách coi việc phân lớp cho mỗi nhãn là một phân lớp nhị phân để đánh giá hay nói cách khác ta có thể giải thích các giá trị bằng cách như sau:

- TP là số ví dụ có nhãn là l và được gán đúng nhãn là *l* (T).

- TN là số ví dụ có nhãn khác *l* và được gán nhãn khác *l* (T)
- FP là số ví dụ có nhãn khác *l* những lại được gán nhãn *l* (F)
- FN là số ví dụ có nhãn *l* nhưng lại được gán nhãn khác *l* (F)

Ba độ đo trên được tính theo công thức sau[2]:

- Độ đo hồi tưởng: $\pi = \frac{TP}{TP+FN}$
- Độ đo chính xác: $\rho = \frac{TP}{TP+FP}$
- Độ đo f_1: $f_1 = \frac{2\pi\rho}{\pi+\rho}$

4.5 Kết quả thực nghiệm

4.5.1 Kết quả đánh giá nội miền

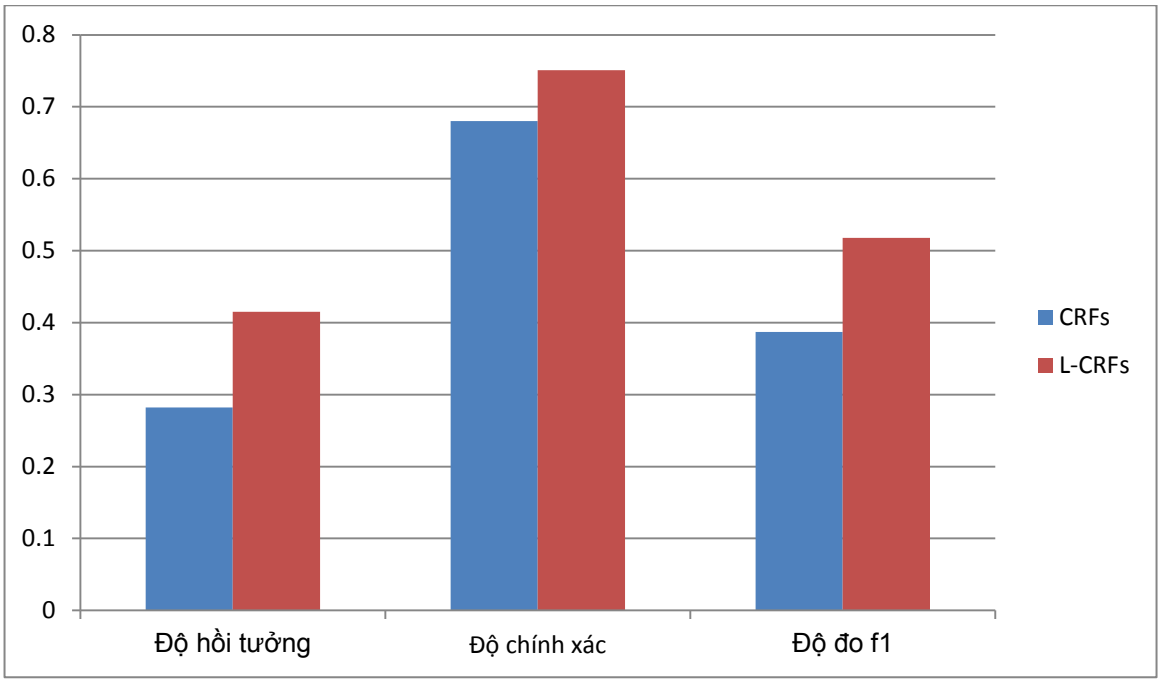
Kết quả thực nghiệm là kết quả trung bình của 3 loại thực thể: tên người, tên địa danh và tên tổ chức.

Kết quả đánh giá nội miền được trình bày trong bảng sau:

Miền	CRFs			L-CRFs		
	Độ hồi tưởng	Độ chính xác	Độ đo f1	Độ hồi tưởng	Độ chính xác	Độ đo f1
CNTT	0.427	0.898	0.579	0.51	0.849	0.637
KT	0.2	0.95	0.332	0.33	0.9	0.483
PL	0.248	0.666	0.362	0.304	0.622	0.409
XH	0.149	0.5	0.229	0.434	0.566	0.491
TT	0.364	0.582	0.448	0.419	0.555	0.478
GD	0.306	0.482	0.374	0.492	0.799	0.609
TB	0.282	0.68	0.387	0.415	0.715	0.518

Bảng 4.6 Kết quả thực nghiệm đánh giá nội miền

Để có thể so sánh và đánh giá được kết quả chính xác và dễ dàng hơn, luận văn sẽ thể hiện kết quả trung bình của 3 độ đo với hai phương pháp tiếp cận dưới dạng biểu đồ như sau :



Hình 4.1 Kết quả thực nghiệm đánh giá nội miền

L-CRFs cho kết quả tốt hơn với phương pháp CRFs truyền thống, cụ thể là độ đo f1 cao hơn 0.131. Như vậy có thể nhận ra rằng, các tri thức đã được học từ các miền trong quá khứ có ảnh hưởng đáng kể tới kết quả học ở miền hiện tại.

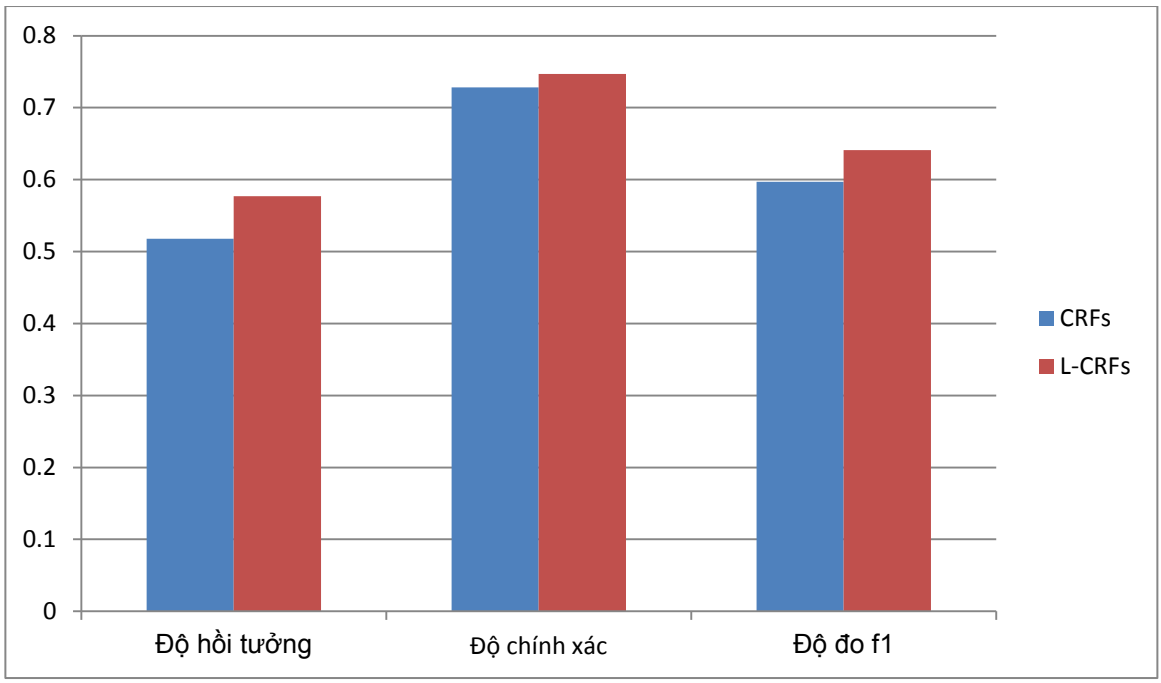
4.5.2 Kết quả đánh giá chéo miền

Miền	CRFs			L-CRFs		
	Độ hồi tưởng	Độ chính xác	Độ đo f1	Độ hồi tưởng	Độ chính xác	Độ đo f1
CNTT	0.512	0.801	0.624	0.532	0.787	0.635
KT	0.618	0.756	0.68	0.655	0.795	0.718
PL	0.266	0.642	0.376	0.286	0.655	0.398
XH	0.62	0.669	0.644	0.806	0.757	0.781

TT	0.522	0.647	0.578	0.555	0.65	0.599
GD	0.568	0.857	0.683	0.626	0.84	0.717
TB	0.518	0.728	0.597	0.577	0.747	0.641

Bảng 4.7 Kết quả thực nghiệm đánh giá chéo miền

Để có thể so sánh và đánh giá được kết quả chính xác và dễ dàng hơn, luận văn sẽ thể hiện kết quả trung bình của 3 độ đo với hai phương pháp tiếp cận dưới dạng biểu đồ như sau :



Hình 4.2 Kết quả thực nghiệm đánh giá chéo miền

Trong kịch bản thực nghiệm này, kết quả của L-CRFs vẫn cao hơn của CRFs truyền thống, tuy nhiêu cao hơn không đáng kể (f1 tăng 0.044). Chúng ta có thể dễ dàng lí giải cho hiện tượng này. Với việc dữ liệu huấn luyện là kết hợp của tất cả các miền, như vậy tập dữ liệu huấn luyện là khá đa dạng, dẫn đến kết quả nhận được là khả quan hơn so với trường hợp đánh giá nội miền. Bên cạnh đó, dữ liệu của các miền khác đã được sử dụng trong quá trình huấn luyện nên tác dụng L-CRFs là không đáng kể.

4.5.3 Kết quả đánh giá chéo miền có dữ liệu của miền đích

Bảng dưới đây thể hiện kết quả thực nghiệm với dữ liệu của miền đích trong tập huấn luyện tăng dần sử dụng độ đo F1 được thực hiện với CRFs và L-CRFs:

Miền	CRFs			L-CRFs		
	1/2	1/4	1/6	1/2	1/4	1/6
CNTT	0.67	0.646	0.631	0.672	0.647	0.635
KT	0.731	0.725	0.7227	0.7492	0.7329	0.728
PL	0.433	0.405	0.394	0.458	0.434	0.422
XH	0.774	0.758	0.744	0.792	0.764	0.748
TT	0.608	0.590	0.582	0.659	0.63	0.624
GD	0.738	0.723	0.719	0.741	0.735	0.73

Bảng 4.8 Kết quả thực nghiệm đánh giá chéo miền có dữ liệu miền đích

Trong quá trình học, vai trò dữ liệu của miền đích trong tập huấn luyện là vô cùng quan trọng. Qua kịch bản thực nghiệm này, ta có thể dễ dàng nhận thấy nếu dữ liệu của miền đích trong tập huấn luyện càng nhiều thì kết quả nhận được có độ chính xác càng cao. Việc áp dụng hóc suốt đời thông qua thuật toán L-CRFs vẫn cho kết quả tốt hơn mặc dù không đáng kể.

4.5.4 Kết quả đánh giá chéo miền chỉ lấy dữ liệu miền gần

Để kiểm tra các miền có "gần" nhau hay không, luận văn thực hiện đánh giá mức độ tương đồng giữa hai miền trên mức độ từ vựng, với công thức như sau[8]:

$$\frac{|V_i \cap V_j|}{|V_i|} + \frac{|V_i \cap V_j|}{|V_j|}$$

Trong đó: $\boldsymbol{V_i}$ là tập từ vựng thuộc miền $\boldsymbol{D_i}$ và $\boldsymbol{V_j}$ là tập từ vựng thuộc miền $\boldsymbol{D_i}$

$\frac{|V_i \cap V_j|}{|V_i|}$ cho biết mức độ của V_j trong V_i .

Ta có bảng kết quả như sau:

	CNTT	KT	XH	PL	TT	GD
CNTT	-	0.672	**0.933**	0.616	0.8	0.721
KT	0.672	-	**0.764**	0.696	0.665	0.659
XH	**0.933**	0.764	-	0.645	0.928	0.548
PL	0.616	**0.696**	0.645	-	0.645	0.675
TT	0.8	0.665	**0.928**	0.645	-	0.631
GD	**0.721**	0.659	0.548	0.675	0.631	-

Bảng 4.9 Kết quả đo độ "gần" giữa các miền mức từ vựng

Từ kết quả trên, luận văn đã thực hiện thực nghiệm đánh giá với phương pháp L-CRFs và có kết quả như sau:

Miền	L-CRFs			
	Độ chính xác	Độ hồi tưởng	Độ đo F1	Miền "gần"
CNTT	0.5197	0.7913	0.6273	XH
KT	0.7014	0.7183	0.7097	XH
PL	0.337	0.669	0.448	KT
XH	0.765	0.733	0.749	CNTT
TT	0.5427	0.6609	0.596	XH
GD	0.5057	0.7113	0.5911	CNTT

Bảng 4.10 Kết quả thực nghiệm chỉ sử dụng dữ liệu từ miền "gần"

Nhận xét:

Kết quả thực nghiệm đã chứng minh tính khả thi và ưu điểm khi áp dụng phương pháp học suốt đời cho bài toán nhận dạng thực thể định danh trong văn bản Tiếng Việt. Bên cạnh đó kết quả thực nghiệm cũng làm bật lên được những khó khăn của việc nhận dạng thực thể định danh trong văn bản ngắn tiếng Việt. Cụ thể như sau:

- Khi ta thực hiện thực nghiệm trên cùng một miền, không gian đặc trưng cũng như phân bố của dữ liệu huấn luyện và kiểm tra là như nhau. Tuy nhiên do đặc điểm của văn bản ngắn nên kết quả nhận được là không khả quan, chỉ đạt được độ đo f1 là 0.387 . Khi áp dụng học suốt đời, ta nhận được kết quả f1 là 0.518 tăng 0.131 so với phương pháp truyền thống.
- Trong thực nghiệm đánh giá chéo miền, mặc dù không gian đặc trưng là như nhau nhưng phân bố dữ liệu ở các miền khác nhau, vì vậy kết quả của CRFs trong trường hợp này chỉ đạt f1 = 0.597. L-CRFs cho kết quả là f1 = 0.641 nhờ tận dụng được các dữ liệu đã học trong quả khứ. Tuy nhiên trong trường hợp này kết quả chỉ tăng 0.044 so với phương pháp truyền thống, bởi trong tập dữ liệu huấn luyện đã được kết hợp với dữ liệu của các miền khác nên việc tận dụng tri thức của các miền đó đem lại hiệu quả không đáng kể.
- Một câu hỏi đặt ra là sự có mặt của dữ liệu ở miền đích ở tập dữ liệu huấn luyện ảnh hưởng nhiều hay ít tới kết quả của thực nghiệm? Để trả lời cho những câu hỏi trên, luận văn đã tiến hành thực nghiệm trường hợp thứ ba. Như kết quả thực nghiệm ta có thể dễ dàng nhận thấy càng nhiều dữ liệu miền đích trong tập huấn luyện thì cho kết quả càng cao.
- Trong thực nghiệm thứ 4, ta chỉ sử dụng tri thức có được từ miền "gần" với miền đang xét, kết quả nhận được là khá tốt so với việc sử dụng tri thức từ tất cả các miền. Tuy nhiên thời gian chạy trong trường hợp này thấp hơn rất nhiều bởi ta chỉ cần xem xét dữ liệu nhỏ hơn nhiều.

Kết luận

Luận văn đã đạt được:

- Tìm hiểu bài toán nhận dạng thực thể trong văn bản Tiếng Việt và cách tiếp cận bằng phương pháp học máy sử dụng mô hình trường ngẫu nhiên(Conditional Random Fields)
- Tìm hiểu những kiến thức cơ bản về học suốt đời (định nghĩa, phân loại, cách đánh giá…) cùng những áp dụng của học suốt đời.
- Tìm hiểu việc áp dụng học suốt đời cho mô hình CRFs nhằm cải tiến phương pháp nhận dạng thực thể trong văn bản ngắn để khắc phục những khó khăn gặp phải do đặc điểm của văn bản ngắn.

Những đóng góp chính của luận văn:

- Xây dựng mô hình CRFs để nhận dạng thực thể trong văn bản Tiếng Việt áp dụng học suốt đời.
- Tiến hành đánh giá thực nghiệm để so sánh giữa nhiều trường hợp, từ đó chứng minh được áp dụng học suốt đời có thể làm tăng hiệu suất của việc học cũng như chỉ ra vai trò quan trọng cảu dữ liệu có được thông qua các bài toán học trong quá khứ cho việc nhận dạng thực thể định danh ở bài toán học hiện tại.

Tài liệu tham khảo

Tiếng Việt

1. Thụy, H. Q., Hiếu, P. X., & Sơn, Đ. Nguyễn Trí Thành, Nguyễn Thu Trang, Nguyễn Cẩm Tú (2009). Giáo trình Khai phá dữ liệu Web.

Tiếng Anh

2. Abdallah, Z. S., Carman, M., & Haffari, G. (2017). Multi-domain evaluation framework for named entity recognition tools. *Computer Speech & Language*, *43*, 34-55.

3. Chen, M., Jin, X., & Shen, D. (2011, July). Short text classification improved by learning multi-granularity topics. In *IJCAI* (pp. 1776-1781).

4. De Marneffe, M. C., & Manning, C. D. (2008). *Stanford typed dependencies manual* (pp. 338-345). Technical report, Stanford University.

5. Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000, September). Rule-based named entity recognition for Greek financial texts. In Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000) (pp. 75-78).

6. Ferreira, E., Balsa, J., & Branco, A. (2007). Combining rule-based and statistical methods for named entity recognition in Portuguese. In Actas da 5a Workshop em Tecnologias da Informaçao e da Linguagem Humana.

7. Fei, G., Wang, S., & Liu, B. (2016, August). Learning cumulatively to become more knowledgeable. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1565-1574). ACM.

8. Ha, Q. T., Pham, T. N., Nguyen, V. Q., Nguyen, T. C., Vuong, T. H., Tran, M. T., & Nguyen, T. T. (2018, March). A New Lifelong Topic Modeling Method and Its Application to Vietnamese Text Multi-label Classification. In Asian Conference on Intelligent Information and Database Systems (pp. 200-210). Springer, Cham.

9. Jakob, N., & Gurevych, I. (2010, October). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1035-1045). Association for Computational Linguistics.

10. Kumar, A., & Daume III, H. (2012). Learning task grouping and overlap in multi-task learning. arXiv preprint arXiv:1206.6417.

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

12. McCallum, A., Freitag, D., & Pereira, F. C. (2000, June). Maximum Entropy Markov Models for Information Extraction and Segmentation. In ICML (Vol. 17, pp. 591-598).

13. McCallum, A., & Li, W. (2003, May). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 188- 191). Association for Computational Linguistics.

14. Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., ... & Krishnamurthy, J. (2018). Never-ending learning. Communications of the ACM, 61(5), 103-115.

15. Silver, D. L., Mason, G., & Eljabu, L. (2015, June). Consolidation Using Sweep Task Rehearsal: Overcoming the Stability-Plasticity Problem. In Canadian Conference on Artificial Intelligence (pp. 307-322). Springer, Cham.

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

17. Thrun, S., Mitchell, T.M.: Lifelong robot learning. Robot. Auton. Syst. 15(1–2), 25–46(1995)

18. Thrun, S.: Explanation-Based Neural Network Learning: A Lifelong Learning Approach.Springer, US (1996).

19. Tran, Q. T., Pham, T. T., Ngo, Q. H., Dinh, D., & Collier, N. (2007). Named entity recognition in Vietnamese documents. Progress in Informatics Journal,5, 14-17.

20. Tu, N. C., Oanh, T. T., Hieu, P. X., & Thuy, H. Q. (2005). Named entity recognition in vietnamese free-text and web documents using conditional random fields. In The 8th Conference on Some selection problems of Information Technology and Telecommunication.

21. Zhiyuan Chen and Bing Liu. Lifelong Machine Learning. Morgan & Claypool Publishers, November 2016.

22. Zhou, G., & Su, J. (2002, July). Named entity recognition using an HMM-based chunk tagger. In proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 473-480). Association for Computational Linguistics.

Trang web

22. http://cs.nyu.edu/cs/projects/proteus