Bộ Dữ Liệu Phân Giải Đồng Tham Chiếu Trên Ngôn Ngữ Tiếng Việt

Nguyễn Trọng Mạnh[1,2], **Lê Tuấn Hưng**[1,2], **Trần Quốc Khánh** [1,2], **Nguyễn Gia Tuấn Anh** [1,2]
[1] *University of Information Technology, Ho Chi Minh*
[2] *Faculty of Information Science and Engineering, City, Vietnam*
21520343, 21520250@gm.uit.edu.vn

Tóm Tắt

Bài toán phân giải chuỗi đồng tham chiếu là một bài toán thuộc lĩnh vực xử lý tự nhiên đã được các nhà khoa học nghiên cứu từ rất lâu. Tuy nhiên, tính đến thời điểm hiện tại, các bộ dữ liệu được xây dựng vẫn còn rất hạn chế trong tiếng Việt. Vì thế, trong nghiên cứu này, chúng tôi trình bày một bộ dữ liệu trong vấn đề phân giải đồng tham chiếu với nguồn là các đoạn văn bản ngắn trên Wikipedia, bao gồm 7844 nhãn chỉ mối quan hệ tham chiếu của 10997 nhãn chủ thể được lấy từ 645 đoạn văn khác nhau. Bộ dữ liệu được gán nhãn chỉ tập trung đến các chủ thể liên quan đến con người, cho phép thực hiện đánh giá và nhận định các tác vụ phân giải đồng tham chiếu các chủ thể liên quan đến con người trong ngôn ngữ tự nhiên.

1 Giới Thiệu

Hiện nay, các tác vụ phân giải đồng tham chiếu chủ yếu được đánh giá trên bộ dữ liệu OntoNotes 5.0 Dataset - một bộ dữ liệu lớn được sử dụng rộng rãi trong các tác vụ của lĩnh vực xử lý ngôn ngữ tự nhiên, và tác vụ phân giải đồng tham chiếu cũng nằm trong số đó. Tuy nhiên, bộ dữ liệu từ OntoNotes chỉ phục vụ cho các ngôn ngữ tiếng Anh, tiếng Trung và tiếng Ả Rập. Chỉ có một số lượng ít ỏi dữ liệu phục vụ cho tác vụ này ở ngôn ngữ tiếng Việt của VLSP (Câu lạc bộ Xử lí Ngôn ngữ và Tiếng nói tiếng Việt) thông qua các cuộc thi như VLSP Shared Task 2018 và VLSP Shared Task 2020. Vì thế nguồn cung cấp bộ dữ liệu phục vụ tác vụ phân giải đồng tham chiếu đặc biệt đối với ngôn ngữ tiếng Việt rất khan hiếm. Chính vì thế, chúng tôi giới thiệu trong bài này bộ dữ liệu gồm 645 đoạn văn miêu tả, giới thiệu nhân vật, bộ phim,... được trích xuất từ Wikipedia và được gán nhãn thủ công tập trung vào các đề cập con người sau khi chia nhỏ các bài văn ra thành các đoạn có độ dài từ 16 đến 706 từ.

Mục tiêu của nhóm đó là xây dựng một bộ dữ liệu cho bài toán phân giải đồng tham chiếu trên ngôn ngữ tiếng Việt từ đó giúp xây dựng một mô hình học máy hoặc học sâu có thể có thể giải quyết được các mối quan hệ đồng tham chiếu có ở ngôn ngữ tiếng Việt, bên cạnh đó bộ dữ liệu có thể bổ trợ cho các bài toán khác thuộc lĩnh vực xử lý ngôn ngữ tự nhiên của tiếng Việt, góp phần xây dựng các ứng dụng và công cụ như gợi tóm tắt văn bản, xây dựng hệ thống chatbot, hoặc dịch máy. Qua đó giúp cải thiện khả năng xử lý ngôn ngữ tự nhiên cho tiếng Việt và tăng cường trải nghiệm người dùng trong việc tương tác với các ứng dụng NLP. Chúng tôi còn mong muốn bộ dữ liệu sẽ có thể phát hiện và giải quyết với các thách thức đặc thù của tiếng Việt trong việc phân giải chuỗi đồng tham chiếu. Các thách thức này có thể bao gồm sự đa dạng về cách diễn đạt, sự mờ nhạt về ngữ cảnh, và cấu trúc ngôn ngữ đặc biệt của tiếng Việt.

2 Bộ Dữ Liệu

Nguồn dữ liệu chúng tôi chọn là trang vi.Wikipedia.org dành cho ngôn ngữ tiếng Việt [1], Wikipedia là một trang web thuộc dạng bách khoa toàn thư, được coi là một nguồn thông tin trực tuyến phổ biến và miễn phí. Wikipedia cung cấp thông tin về nhiều chủ đề khác nhau bằng nhiều ngôn ngữ khác nhau. Nó cho phép người dùng đăng nhập và chỉnh sửa, bổ sung nội dung của các bài viết để nâng cao chất lượng và độ phong phú của thông tin. Nhóm tiến hành lựa chọn những văn bản, bài văn, tiêu đề mô tả giới thiệu nhân vật đặc biệt ở các bộ truyện tranh, phim hoạt hình và đến cả phim truyền hình. Điều này giúp chúng tôi xây dựng một bộ dữ liệu phong phú giúp phản ánh được các đặc điểm ngôn ngữ và cấu trúc tiếng Việt thời nay. Bên cạnh đó trang Wikipedia còn đảm bảo được một trong những yêu cầu của một bộ dữ liệu tốt đó chính là độ tin cậy, cũng như rất ít lỗi xuất hiện như lỗi chính tả, lỗi xuống dòng, lỗi dính chữ, lỗi không viết hoa tên nhân vật. Bộ dữ liệu được thu thập

[1] https://vi.wikipedia.org/wiki

bằng các đoạn mã Python (phiên bản 3.10) sử dụng thư viện BeautifulSoup, đây là thư viện cung cấp các công cụ và phương thức hỗ trợ việc trích xuất thông tin từ HTML và XML. Sau khi trích xuất dữ liệu từ Wikipedia là đoạn văn bản về một nhân vật bất kỳ, chúng tôi tiến hành xử lý các ký tự đặc biệt cũng như lọc lại những đoạn có chủ thể nhiều hơn năm bằng thư viện Underthesea, tiếp theo đó chúng tôi lưu trữ chúng thành dạng cấu trúc dữ liệu CSV, điều này giúp chúng tôi tổ chức và lưu trữ dữ liệu một cách có cấu trúc dễ dàng cho bước tiền xử lý tiếp theo. Cấu trúc dữ liệu CSV sẽ gồm cột text chứa văn bản, cột label được bỏ trống, đây là cột sẽ chứa các nhãn sau quá trình gán. Tiếp đó, chúng tôi tiến hành chuyển dữ liệu dạng CSV sang dạng jsonlines, đây là dạng dữ liệu văn bản trong đó mỗi dòng chứa một bản ghi được mã hóa dưới định dạng json. Dữ liệu dạng jsonlines này sẽ được đưa vào công cụ gán nhãn tiến hành cho bước tiếp theo trong quá trình xây dựng bộ dữ liệu.

2.1 Quá trình gán nhãn

Trong bộ dữ liệu này, việc gán nhãn nhằm xác định và đánh dấu các thực thể chỉ người và chỉ ra mối quan hệ đồng tham chiếu giữa các thực thể chỉ người đó.

2.1.1 Thiết kế bộ nhãn

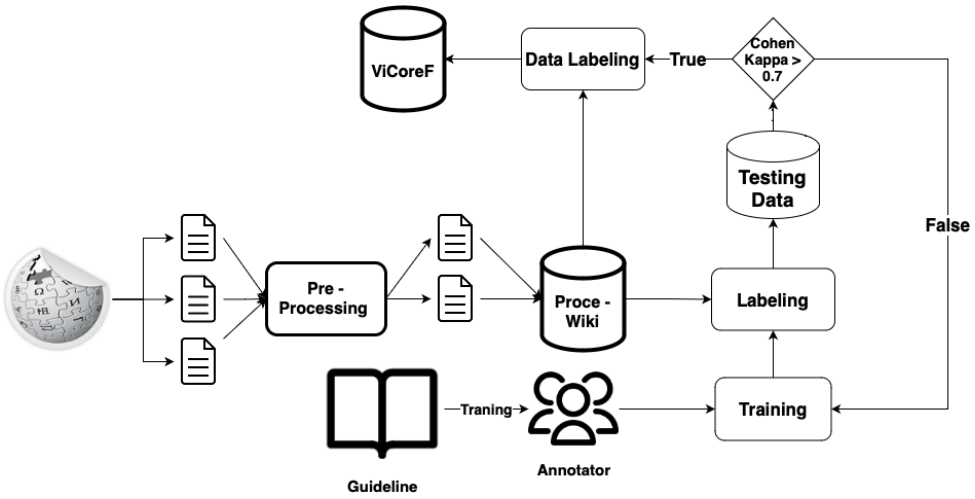
Chúng tôi thiết kế bộ dữ liệu có hai yếu tố là ENTITY (thực thể được nhắc đến trong bài văn) và RELATION (mối quan hệ giữa hai ENTITY). Các bài nghiên cứu như A Dataset of Literary Coreference [1] , An Annotated Dataset of Literary Entities [2], họ đều sử dụng bộ phân loại ACE ENTITY gồm PER (con người), FAC (thiết bị), LOC (địa điểm), GPE (thực thể địa chính trị), VEH (phương tiện giao thông) và ORG (tổ chức). Đối với các loại RELATION, họ sử dụng quan hệ đề cập chung, đề cập cụ thể, đề cập liên từ kết nối, đề cập bổ sung thông tin, đề cập xác định. Khác với bài nghiên cứu trên, chúng tôi chỉ sử dụng loại nhãn ENTITY duy nhất là PER đóng vai trò trong việc xác định tên riêng của người, tên nhân vật, đại từ nhân xưng (tôi, cô ấy,...) và các cụm danh từ đề cập đến con người (một nhà văn, chủ khách sạn,...). Bên cạnh đó, RELATION chỉ chứa nhãn COREF ghi nhận mối quan hệ đồng tham chiếu giữa các ENTITY tạo nên những chuỗi tham chiếu của nhiều ENTITY khác nhau trong bộ dữ liệu.

2.1.2 Quá trình gán nhãn

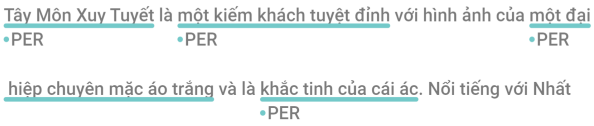
Để gán nhãn cho tập dữ liệu, chúng tôi sử dụng công cụ Doccano - một công cụ đánh dấu văn bản mã nguồn mở cung cấp các tính năng đánh dấu cho việc phân loại văn bản, gán nhãn chuỗi và nhiều tác vụ khác trong lĩnh vực xử lý ngôn ngữ tự nhiên. Nhóm đã xây dựng quy trình gán nhãn như hình 1.

2.2 Phương pháp gán nhãn

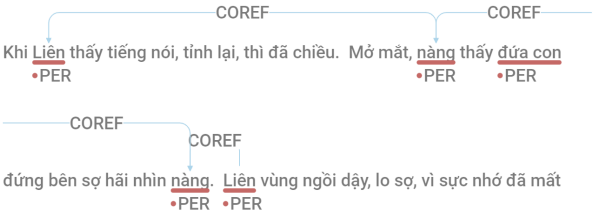
Có hai nhiệm vụ trong quy trình gán nhãn đó là:
Gán nhãn ENYITY - xác định các thực thể chỉ người bằng cách đi gán nhãn PER lên những thực thể ấy.
Xác định RELATION - nối RELATION giữa hai thực thể có mối quan hệ đồng tham chiếu (COREF) với nhau.
Trước tiên, người gán nhãn sẽ thực hiện đánh dấu các thực thể trong đoạn văn. Trong nhiệm vụ này, người gán nhãn sẽ chỉ sử dụng nhãn PER để định vị những từ hoặc cụm từ đề cập người nhưng chỉ dùng cho những từ chỉ số ít (tôi, cô ấy, anh ấy, chị, một người đưa thư,...) về những danh từ chỉ một nhóm người (từ hai người trở lên như họ, hai ông bà, đám trẻ,...) chúng tôi sẽ không áp dụng nhãn PER cho chúng. Bên cạnh các đại từ nhân xưng, tên riêng chỉ người hay nhân vật, chúng tôi còn xác định cho cả cụm danh từ chỉ một cá thể nhất định (*Hình 2*). Độ rộng của thực thể sẽ là độ rộng của cụm từ dài nhất chỉ thực thể đó bao gồm cả những đặc điểm nhận dạng hoặc miêu tả riêng, điều này giúp bao quát cả các trường hợp đặc biệt, khi các cụm từ cùng nói đến hoặc ám chỉ đến thực thể mà chúng tôi đang xem xét trong một chuỗi đồng tham chiếu nhất định.
Tiếp theo, tiến hành gán RELATION. Với chỉ duy nhất nhãn COREF, người gán nhãn sẽ thực hiện nối các ENTITY đã gán nhãn PER trước đó có mối quan hệ đồng tham chiếu với nhau. Điều này có nghĩa là các ENTITY được liên kết với nhau theo một chuỗi liên kết liên tiếp, trong đó một ENTITY trong chuỗi được gán đồng tham chiếu với ENTITY liền trước nó (*Hình 3*).
Trong quá trình gán nhãn, những người gán nhãn sẽ gặp một số trường hợp đặc biệt trong bước gán nhãn RELATION sau. *Đầu tiên* là trường hợp có hai tên riêng khác nhau nhưng cùng chỉ đến một nhân vật. Một đặc trưng của văn bản nguồn ngữ tiếng Việt là khi xét trong ngữ cảnh của một đoạn văn bản, một nhân vật có thể sẽ có hai hay nhiều tên khác nhau, khi đó, người gán nhãn sẽ không cần tạo ra chuỗi COREF mới, mà vẫn tiếp tục kết nối với chuỗi COREF cũ cùng tham chiếu đến nhân vật



Hình 1: Quy trình xây dựng bộ dữ liệu ViCoref (**Vi**et Nam **Coref**erence)



Hình 2: Gán nhãn PER cho tên riêng và cụm danh từ

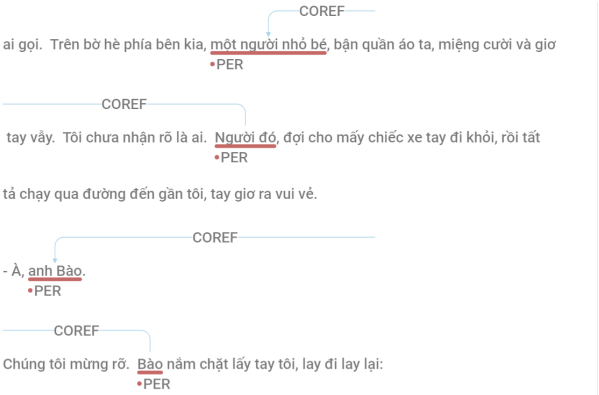


Hình 3: Gán nhãn RELATION cho các ENTITY cùng tham chiếu đến một chủ thể

đó. Trong *Hình 4*, có hai tên gọi khác nhau cùng chỉ đến một nhân vật đó là "Cửu Vĩ"và "Kyubi", hai tên riêng này nằm chung trong một chuỗi COREF. Điều này giúp chúng tôi xây dựng một chuỗi đồng tham chiếu chính xác, thể hiện mối quan hệ giữa các tên gọi và thực thể trong văn bản.

Thứ hai, người gán nhãn sẽ gặp những thực thể

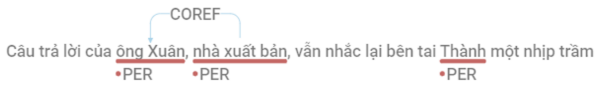
Hình 4: Hai tên riêng khác nhau nhưng cùng nằm trong 1 chuỗi COREF



Hình 5: Những thực thể mơ hồ, chung chung không được đề cập cụ thể

mơ hồ. Một trong những đặc trưng thường xuất hiện trong văn bản tiếng Việt đó là nhân vật ban đầu chỉ được đề cập một cách mơ hồ, chung chung (người đó, một người thấp bé,...) sau đó mới được biết rõ tên riêng. Trong *Hình 5*, xét trên ngữ cảnh chúng ta có thể nhận thấy “một người nhỏ bé”, “người đó”, “anh Bào” cùng hướng đến chung một người, nhưng chúng tôi sẽ cho nó là hai thực thể riêng biệt vì các cụm từ này không phải là độc nhất chỉ đến một nhân vật thật cụ thể như trường hợp đầu tiên. Điều này sẽ tránh gây nhiễu cho bộ dữ liệu khi tiến hành huấn luyện mô hình.

Còn một trường hợp nữa mà chúng tôi gặp trong quá trình gán nhãn. Chúng tôi gọi là đồng tham chiếu chéo khi mà một thực thể được đề cập bằng tên riêng rồi sau đó là một chức vụ, chức danh trong văn bản. Ở ví dụ trong *Hình 6*, "ông Xuân"và "nhà xuất bản"cùng chỉ đến một người. Chúng tôi sẽ gom lại thành một chuỗi COREF đại diện cho cùng



Hình 6: Đồng tham chiếu chéo

một ENTITY tạo tính nhất quán trong dữ liệu.

2.3 Độ đồng thuận

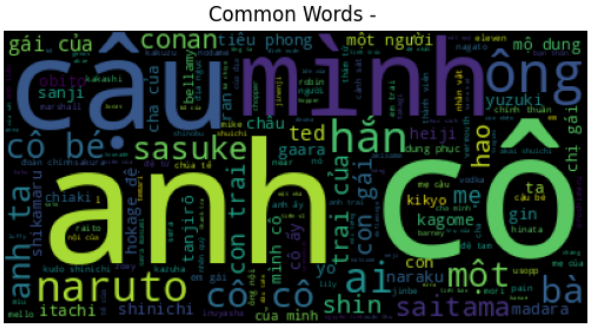
Để đo mức độ đồng thuận trong tập dữ liệu, chúng tôi sử dụng độ đo Cohen – Kappa [3] với công thức $k = \frac{Pr(A)-Pr(e)}{1-Pr(e)}$. Độ đồng thuận được đo giữa hai annotator cho việc xác định các ENTITY có cùng thuộc một chuỗi đồng tham chiếu hay không. Chúng tôi sẽ lấy 10% dữ liệu đã được xác định các thực thể để tiến hành đo thông số này.

	$\mathbf{Y_B}$	$\mathbf{N_B}$
$\mathbf{Y_A}$	838	39
$\mathbf{N_A}$	17	156

Trên đây là bảng ma trận nhầm lẫn giữa hai người gán nhãn trong quá trình gán CORREF cho hai thực thể A và B với Y (Yes) là hai thực thể đang xét có mối quan hệ COREF và ngược lại với N (No). Độ đồng thuận k = 0.813 thể hiện mức độ đồng thuận rất tốt giữa hai người gán nhãn trong việc gán nhãn và đánh giá dữ liệu. Ma trận đánh giá đồng thuận cho thấy sự khớp giữa hai người gán nhãn. Trong ma trận, số liệu 838 đại diện cho số lượng cặp ENTITY được đánh giá đồng thuận là "Yes"cả hai người gán nhãn đồng ý và chỉ có 39 trường hợp người thứ hai không đồng thuận. Tương tự, có 156 trường hợp đồng thuận là "No"và chỉ có 17 trường hợp người thứ nhất không đồng thuận. Sự khớp cao giữa hai người gán nhãn làm tăng giá trị k, cho thấy mức độ đồng thuận rất tốt. Mức độ đồng thuận cao cho thấy quá trình gán nhãn được thực hiện khá nhất quán giữa hai người gán nhãn, điều này rất quan trọng để đảm bảo tính nhất quán và đáng tin cậy của bộ dữ liệu.

2.4 Khảo sát tập dữ liệu

Tập dữ liệu của chúng tôi bao gồm 100 bài báo Wikipedia khác nhau với 10997 nhãn PER chỉ thực thể con người và 7844 nhãn COREF chỉ mối quan hệ đồng tham chiếu giữa các thực thể. Trong các đoạn văn, các thực thể đa số là các tên riêng của các nhân vật, đại tử nhân xưng và đại từ quan hệ (tôi, chàng, nàng, anh, cậu, cô,. . .) như *Hình 7*, cụm danh từ. Mỗi đoạn văn trung bình sẽ có từ 12 đến 15 nhãn COREF giữa các ENTITY.



Hình 7: Đám mây các thực thể trong đoạn văn

2.5 Phương pháp xử lý dữ liệu sau gán nhãn

Sau khi dữ liệu được gán nhãn và lưu dưới dạng jsonlines, nhóm tiến hành chuyển cấu trúc từ dữ liệu jsonlines sang cấu trúc dữ liệu dạng CSV. Cấu trúc dữ liệu CSV mới như bảng 2.5. Trong đó thuộc tính COREF sẽ có giá trị True hoặc False phụ thuộc vào thực thể A và B có cùng chỉ tới cùng một thực thể hay không. Từ 645 đoạn văn, nhóm lấy hai thực thể trong đoạn văn bằng cách lấy tổ hợp chập hai của N thực thể trong đoạn văn đó, nếu trong tệp dữ liệu jsonlines hai thực thể có RELATION là COREF thì thuộc tính COREF trong tệp CSV là True, còn không thì được gán là False.

Thuộc tính	Mô tả
text	Đoạn văn
A	Thực thể A được gán nhãn
A-offset	Vị trí bắt đầu của A
B	Thực thể B được gán nhãn
B-offset	Vị trí bắt đầu của B
COREF	A và B có cùng tham chiếu

Tiếp đó, nhóm thực hiện chuyển chữ hoa thành chữ thường cho thuộc tính **text** nhằm giúp cho những thực thể là tên riêng nhưng vì một số lý do nào đó mà không được viết hoa lên, điều này bảo đảm không có sự sai lệch khi huấn luyện mô hình khi phải phân biệt chữ viết hoa hoặc thường khi mà nó đều cùng là một thực thể.

Cuối cùng, nhóm sử dụng mô hình AutoTokenizer và AutoModel được huấn luyện sẵn từ mô hình PhoBERT [5] nhằm nhúng những thực thể A và B theo ngữ cảnh của đoạn 'text' để trích xuất ra những đặc trưng phục vụ cho việc huấn luyện mô hình ở bước sau. Chuyển đổi dữ liệu sau khi nhúng có cấu trúc như bảng dưới:

Thuộc tính	Mô tả
emb_A	Đặc trưng của thực thể A
emb_B	Đặc trưng của thực thể B
label	A và B có cùng tham chiếu

3 Phương pháp máy học

3.1 Mô hình và các giá trị tham số

Để huấn luyện mô hình cho bộ dữ liệu này, chúng tôi sử dụng mô hình **MLP** (Multi-layer Perceptron) [4] để huấn luyện sau khi thực hiện biểu diễn các đoạn văn bản dạng chữ trong tiếng Việt thành dạng số bằng cách sử dụng bộ AutoTokenizer và AutoModel được huấn luyện sẵn từ mô hình PhoBERT ("vinai/phobert-base"). Trong quá trình huấn luyện, chúng tôi sử dụng bộ tham số như sau:

Tham số	Giá trị
dense_layer_sizes	37
dropout_rate	0.5
learning_rate	0.0001
n_fold	5
batch_size	32
epochs	1000
patience	100
lambd	0.1

Trong đó:

- dense_layer_sizes: kích thước của tầng kết nối đầy đủ (dense layer) trong mô hình, ở đây chúng tôi sử dụng tầng kết nối đầy đủ với kích thước là 37 nơ-ron.

- dropout_rate: kỹ thuật chính regularization trong mạng nơ-ron. Giá trị này xác định tỷ lệ nơ-ron sẽ bị bỏ qua ngẫu nhiên trong quá trình huấn luyện để tránh overfitting. Trong trường hợp này, 50% các nơ-ron sẽ bị bỏ qua ngẫu nhiên trong quá trình huấn luyện.

- n_fold: số lượng fold (phân chia) dùng trong quá trình cross-validation. Cross-validation là một phương pháp để đánh giá hiệu suất của mô hình bằng cách chia dữ liệu thành các tập train và validation và thực hiện huấn luyện và kiểm tra trên các tập này. Số lượng fold là 5, nghĩa là dữ liệu sẽ được chia thành 5 fold.

- batch_size: số lượng mẫu được sử dụng trong mỗi lần cập nhật trọng số trong quá trình huấn luyện. Giá trị này là 32, tức là mỗi lần cập nhật trọng số, mô hình sẽ sử dụng 32 mẫu.

- epochs: số lần toàn bộ dữ liệu được sử dụng trong quá trình huấn luyện. Mô hình sẽ được huấn luyện qua 1000 epochs, tức là dữ liệu sẽ được lặp lại qua mạng nơ-ron 1000 lần để cập nhật trọng số.

- patience: số lượng epochs mà mô hình có thể không có sự cải thiện trong hiệu suất trước khi dừng huấn luyện sớm. Trong trường hợp này, nếu không có cải thiện trong 100 epochs liên tiếp, quá trình huấn luyện sẽ dừng lại.

- lambd: một tham số sử dụng trong regularization, như là L1 regularization hoặc L2 regularization (chúng tôi sử dụng L2), để kiểm soát overfitting. Trong trường hợp này, giá trị lambda là 0.1, cho biết mức độ ưu tiên của regularization.

3.2 Các độ đo đánh giá

Để đánh giá và kiểm tra hiệu suất của mô hình trên bộ dữ liệu, chúng tôi sử dụng hàm log_loss có sẵn trong thư viện sklearn. Đây là hàm mất mát thường được sử dụng trong bài toán phân loại nhị phân và phân loại đa lớp. Nó được sử dụng để đo lường sự khác biệt giữa các dự đoán và nhãn thực tế của mô hình. Hàm log_loss tính toán mất mát bằng cách đo lường sự không chính xác của dự đoán so với nhãn thực tế bằng logarit tự nhiên của xác suất dự đoán. Nếu dự đoán đúng, mất mát sẽ gần như bằng 0. Ngược lại, nếu dự đoán sai, mất mát sẽ tăng lên vô hạn.
Công thức:

$$log\_loss = -(ylog(p) + (1-y)log(1-p))$$

Trong đó:
- y: nhãn thực tế (mang 2 giá trị là 0 hoặc 1).
- p: xác suất dự đoán của mô hình cho lớp dương.
Bên cạnh đó, các độ đo như precision, recall, f1-score cũng được chúng tôi sử dụng để đánh giá hiệu suất của mô hình khi dự đoán các nhãn trong bộ dữ liệu.

- **Precision**: đo lường khả năng của mô hình phân loại nhị phân để dự đoán chính xác các mẫu thuộc vào lớp positive (1) so với tổng số các mẫu được dự đoán là positive. Biểu thị khả năng của mô hình dự đoán chính xác các mẫu thuộc lớp dương.
Công thức: Precision = TP / (TP + FP)

- **Recall**: đo lường tỷ lệ các mẫu thuộc lớp dương mà mô hình dự đoán chính xác so với tổng số mẫu thuộc lớp dương, biểu thị khả năng của mô hình trong phát hiện các mẫu thuộc lớp dương.
Công thức: Recall = TP / (TP + FN)

- **F1-score**: là một độ đo kết hợp giữa Precision và Recall trong bài toán phân loại. Nó giúp

đánh giá hiệu suất tổng thể của mô hình phân loại dựa trên cả khả năng dự đoán chính xác các mẫu thuộc lớp dương và khả năng phát hiện tất cả các mẫu thuộc lớp dương.
Công thức: F1-score = 2 * (Precision * Recall) / (Precision + Recall)

Trong đó:

- TP (True Positive): Số lượng các mẫu thuộc lớp dương được mô hình dự đoán chính xác là thuộc lớp dương.
- FP (False Positive): Số lượng các mẫu thuộc lớp âm bị mô hình dự đoán là thuộc lớp dương.
- FN (False Negative): Số lượng các mẫu thuộc lớp dương bị mô hình dự đoán là thuộc lớp âm.

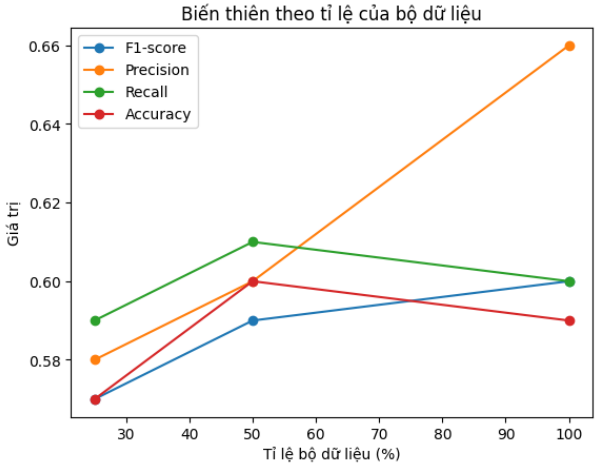
Sau quá trình train dữ liệu và Cross-validation với 5 folds dữ liệu, giá trị trung bình CV mean score đạt 0.6266 và std (độ lệch chuẩn) đạt 0.0042 cho thấy độ biến động khá thấp tuy nhiên hiệu suất đạt được không được cao. Kết quả của 5 lần cross-validation trên từng fold lần lượt đạt 0.6287, 0.6225, 0.6411, 0.6318 và 0.6289. Kết quả của mô hình khi được kiểm tra trên dữ liệu kiểm tra độc lập. Test score đo lường hiệu suất của mô hình trên dữ liệu mới mà nó chưa được huấn luyện trước đó và đạt test score (log_loss) là 0.6364.
Kết quả cuối cùng sau khi huấn luyện mô hình được đo bằng các độ đo như bảng 1. Kết quả mô hình cả bốn thang đo nằm ở mức khá là khoảng 0.5954, điều này chứng tỏ bộ dữ liệu vẫn còn rất phức tạp và cần xây dựng một mô hình máy học mạnh mẽ hơn. Điều này cũng chính là cơ hội là thách thức cho những bài nghiên cứu tiếp sau của bộ dữ liệu của nhóm xây dựng.

Mô hình	Acc	F1-score	Precision	Recall
MLP model	0.59	0.60	0.66	0.60

Bảng 1: Bảng kết quả thực nghiệm

4 Phân tích lỗi của bộ dữ liệu

Khi so sánh với những bộ dữ liệu nổi tiếng và phổ biến như bộ OntoNotes Release 5.0 (Ralph Weischedel và đồng tác giả khác, 2013) gồm tổng cộng 1.300.000 từ của các ngôn ngữ như Anh, Trung Quốc, Ả Rập, bộ An Annotated Dataset of Coreference in English Literature [1] gồm 210.532



Hình 8: Kết quả mô hình tương ứng với tỉ lệ bộ dữ liệu

từ của 100 đoạn văn tiếng Anh khác nhau, bộ chúng tôi vẫn còn nhiều thiếu xót trước hết là độ đa dạng các thực thể và các loại nhãn mối liên hệ giữa các thực thể trong khi bộ dữ liệu chúng tôi chỉ sử dụng duy nhất một nhãn là PER trong tác vụ nhận diện thực thể và một nhãn COREF cho tác vụ xác định mối liên hệ giữa các thực thể. Đối với việc chỉ tập trung vào các chủ thể là con người trong đoạn văn. Và bên cạnh đó, đặc trưng của nguồn dữ liệu nhóm sử dụng là Wikipedia không có xuất hiện nhiều đoạn hội thoại giữa các nhân vật làm cho tính đa dạng của bộ dữ liệu không được cao, lần việc nhãn PER được gán cho cả danh từ riêng, danh từ trừu tượng, đại từ nhân xưng, đại từ sở hữu hoặc là cụm danh từ không phân rõ ra các trường hợp từ đó ảnh hưởng trực tiếp đến việc phan giải đồng tham chiếu của mô hình.
Về kích thước bộ dữ liệu, nhóm đã tiến hành huấn luyện mô hình MLP trên các tập dữ liệu có kích thước khác nhau lần lượt là 25 %, 50 % và 100 % của bộ dữ liệu ban đầu như mô tả ở hình 8, nhóm nhận thấy được ở độ đo F1-Score đang dần bão hòa nghĩa là dù tăng kích thước bộ dữ liệu lên thì thống số F1-Score của mô hình MLP trên bộ dữ liệu cũng không tăng được đáng kể.

5 Hướng phát triển bộ dữ liệu

Sau quá trình gán nhãn và huấn luyện mô hình trên bộ dữ liệu nhóm nhận thấy được bộ dữ liệu có nhiều tiềm năng phát triển tiếp lần lượt là, phân giải nhãn PER thành các từ loại riêng biệt chỉ người từ phân rõ danh từ riêng, dang từ trừu tượng,... thành các nhãn riêng biệt nhằm tránh sự trùng lặp khi gán nhãn quan hệ tham chiếu COREF, bên cạnh đó bộ

dữ liệu có thể mở rộng ra khi có thể xác định các mối quan hệ đồng tham chiếu cho chủ thể là các sự vật, hiện trượng nhằm tăng độ đa dạng của dữ liệu, phù hợp với thực tế bài toán. Về mối quan hệ đồng tham chiếu giữa các thực thể nhóm chỉ áp dụng mối qua hệ tham chiếu đơn nhất nghĩa là một từ trong câu chỉ tham chiếu tới một chủ thể khác nên trong tương lại bộ dữ liệu có thể áp dụng thêm mối quan hệ tham chiếu bộ phận là một từ trong câu tham chiếu đến hai hoặc nhiều hơn hai chủ thể có trong đoạn. Gắn kết thông tin thêm bên ngoài mối quan hệ tham chiếu thông thường. Bên cạnh việc gán nhãn đồng tham chiếu giữa các thực thể, cần xem xét việc gắn kết thông tin khác như giới tính, tuổi, quốc tịch, vai trò trong câu chuyện và các thuộc tính khác của các thực thể. Một hướng khác mà nhóm có thể phát triển là đầu ra của bài toán, thay vì chỉ phân lớp nhị phân giống như đầu ra của các mô hình đánh giá ở trên, chúng ta có thể tạo ra chuỗi các thực thể giống như thư viện neuralcoref của SpaCy - hiện tại mới chỉ được xây dựng cho tiếng Anh và tiếng Trung Quốc.

6 Kết luận

Trong bài nghiên cứu này, chúng tôi giới thiệu bộ dữ liệu đã được gắn nhãn cho tác vụ phân giải đồng tham chiếu trong các đoạn văn về miêu tả nhân vật, bộ phim trên ngôn ngữ tiếng Việt được trích xuất từ Wikipedia, với gần 11000 nhãn chỉ thực thể trong 645 đoạn văn khác nhau. Bộ dữ liệu được tạo ra nhằm phục vụ các mục đích đánh giá hiệu suất và so sánh các mô hình phân giải đồng tham chiếu. Ngày nay, khi mà các bài nghiên cứu, các mô hình tận dụng các phương thức tính toán để khám phá sự tồn tại của các thực thể trong văn bản tiếng Việt, việc có một bộ dữ liệu chất lượng và đáng tin cậy được dùng để làm thang đo đánh giá các mô hình là vô cùng quan trọng. Bộ dữ liệu của chúng tôi vẫn còn nhiều thiếu sót, chúng tôi hy vọng bộ dữ liệu của chúng tôi sẽ giúp ích trong việc đánh giá các mô hình trong quá trình phân giải đồng tham chiếu trong ngôn ngữ tiếng Việt cũng như góp phần hỗ trợ cho các bài toán về lĩnh vực xử lý ngôn ngữ tự nhiên trên tiếng Việt.

7 Tài liệu tham khảo

Tài liệu

[1] David Bamman, Olivia Lewke **and** Anya Mansoor. “An Annotated Dataset of Coreference in English Literature”. English. **in***Proceedings of the Twelfth Language Resources and Evaluation Conference*: Marseille, France: European Language Resources Association, **may** 2020, **pages** 44–54. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.6.

[2] David Bamman, Sejal Popat **and** Sheng Shen. “An annotated dataset of literary entities”. **in***Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*: Minneapolis, Minnesota: Association for Computational Linguistics, **june** 2019, **pages** 2138–2144. DOI: 10.18653/v1/N19-1220. URL: https://aclanthology.org/N19-1220.

[3] J. Cohen. “A Coefficient of Agreement for Nominal Scales”. **in***Educational and Psychological Measurement*: 20.1 (1960), **page** 37.

[4] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[5] Dat Quoc Nguyen **and** Anh Tuan Nguyen. “PhoBERT: Pre-trained language models for Vietnamese”. **in***Findings of the Association for Computational Linguistics: EMNLP 2020*: 2020, **pages** 1037–1042.

8 Phân công công việc

STT	Công việc	Người thực hiện
1	Nghiên cứu đề tài	Mạnh, Hưng
2	Thu thập dữ liệu và thiết kế nhãn	Mạnh, Hưng
3	Phân tích và tiền xử lý dữ liệu	Mạnh, Hưng
4	Gán nhãn dữ liệu	Mạnh, Hưng
5	Lựa chọn và xây dựng mô hình	Mạnh, Hưng
6	Thực nghiệm và hiệu chỉnh mô hình	Mạnh, Hưng
7	Viết báo cáo và thuyết trình	Mạnh, Hưng

Bảng 2: Bảng phân công công việc