ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐẠI HỌC QUỐC GIA - THÀNH PHỐ HỒ CHÍ MINH

□□ □ □□

ĐỒ ÁN CUỐI KỲ

CS221 - Xử lý ngôn ngữ tự nhiên

GVHD : Nguyễn Trọng Chỉnh

Tên đề tài : **Named Entity Recognition với PhoBERT**

Thành viên nhóm :

Nguyễn Gia Bảo - 22520109

Phạm Nguyên Anh - 22520069

Mục Lục

Chương 1: GIỚI THIỆU BÀI TOÁN.....3
1.1 Đặt vấn đề.....3
1.2 Mục tiêu.....3
Chương 2: BỘ NGỮ LIỆU.....4
2.1. Giới thiệu bộ ngữ liệu và phương pháp xây dựng.....4
2.2 Quy tắc chú thích ngữ liệu.....4
2.3 Thống kê ngữ liệu.....4
2.4 Phân tích ngữ liệu.....7
Chương 3: PHƯƠNG PHÁP SỬ DỤNG.....15
3.1. PhoBERT.....15
3.1.1. PhoBERT-base.....15
3.1.2. Luồng dữ liệu xử lý với PhoBERT-base cho bài toán NER.....16
3.2. Model so sánh.....20
Chương 4 : CÀI ĐẶT VÀ THỬ NGHIỆM.....20
4.1. PhoBERT.....20
4.1.1. Tokenization và Label Alignment.....20
4.1.2. Tải và cấu hình mô hình.....21
4.1.3. Cấu hình các đối số huấn luyện.....21
4.1.4. Thiết lập và huấn luyện Trainer.....21
4.2. Conditional Random Fields - CRF.....22
4.2.1. Chuẩn bị dữ liệu.....22
4.2.2. Trích xuất đặc trưng.....22
4.2.3. Huấn luyện mô hình CRF.....23
4.3. Hidden Markov Model.....24
4.4. Kết quả thử nghiệm.....25
4.4.1. Metrics đánh giá.....25
4.4.2. Phân tích kết quả đạt được.....26
4.5. Xử lí data bị gán nhãn sai.....31
Chương 5 : KẾT LUẬN.....38
Tài liệu tham khảo.....39

Chương 1: GIỚI THIỆU BÀI TOÁN

1.1 Đặt vấn đề

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP), **nhận diện thực thể tên (Named Entity Recognition – NER)** là một trong những bài toán quan trọng, có vai trò nền tảng trong nhiều ứng dụng như: hệ thống hỏi đáp, máy tìm kiếm, phân tích văn bản, và trích xuất thông tin. Bài toán NER yêu cầu xác định các thực thể có tên riêng trong văn bản, chẳng hạn như tên người (PER), tổ chức (ORG), hay địa điểm (LOC).

Đối với tiếng Việt, bài toán NER gặp nhiều thách thức đặc thù do ngôn ngữ không có dấu tách từ rõ ràng, đa dạng biểu đạt ngữ pháp và hiện tượng từ đồng âm – dị nghĩa. Hơn nữa, tài nguyên ngữ liệu có nhãn chất lượng cao còn khan hiếm, gây khó khăn cho việc huấn luyện mô hình hiệu quả.

Truyền thống, bài toán NER được giải quyết bằng các mô hình thống kê như **HMM (Hidden Markov Model)** và **CRF (Conditional Random Fields)**, dựa trên đặc trưng trích xuất thủ công (hand-crafted features). Tuy nhiên, gần đây, các mô hình học sâu như **BERT, XLM-R, PhoBERT** đã chứng minh hiệu quả vượt trội nhờ khả năng học biểu diễn ngữ cảnh sâu sắc.

1.2 Mục tiêu

Đồ án này hướng đến việc nghiên cứu và so sánh hiệu quả giữa các phương pháp cổ điển (HMM, CRF) với mô hình hiện đại PhoBERT trong bài toán NER tiếng Việt. Cụ thể, mục tiêu bao gồm:

- Khảo sát và cài đặt các mô hình cổ điển như HMM và CRF cho bài toán NER, sử dụng đặc trưng từ, kiểu chữ, hậu tố, vị trí,...
- Huấn luyện và đánh giá mô hình PhoBERT trên cùng tập dữ liệu để làm cơ sở so sánh.
- Sử dụng dữ liệu tiếng Việt từ WikiAnn (dữ liệu gán nhãn tự động) làm tập huấn luyện chính.
- Phân tích điểm mạnh – yếu, và điều kiện áp dụng của từng mô hình.
- Đề xuất các hướng cải tiến hoặc kết hợp các phương pháp để tăng hiệu quả NER cho tiếng Việt.

Chương 2: BỘ NGỮ LIỆU

2.1. Giới thiệu bộ ngữ liệu và phương pháp xây dựng

Trong đồ án này, nhóm sử dụng dataset tiếng Việt từ **WikiAnn** trên Hugging Face (link dataset : huggingface.co/datasets/unimelb-nlp/wikiann/). Bộ dữ liệu WikiAnn được xây dựng tự động từ Wikipedia dựa trên các liên kết nội bộ giữa các trang, kết hợp với cấu trúc ngữ nghĩa từ các cơ sở tri thức như DBpedia hoặc YAGO. Mỗi token trong văn bản được gán nhãn theo định dạng **IOB2** với ba loại thực thể chính: PER (tên người), ORG (tổ chức), và LOC (địa điểm).

Dữ liệu bao gồm ba tập con: **train**, **validation**, và **test**, được phân tách sẵn. Các nhãn thực thể được gán một cách bán tự động dựa trên anchor text và liên kết đến các trang Wikipedia tương ứng, nên được coi là nhãn **silver-standard** (không phải gán tay tuyệt đối chính xác như gold-standard, nhưng vẫn có chất lượng đủ tốt cho huấn luyện mô hình học máy).

2.2 Quy tắc chú thích ngữ liệu

Trong đồ án này, nhóm áp dụng quy tắc **gán nhãn thực thể theo chuẩn IOB2 (Inside–Outside–Beginning)** để chú thích ngữ liệu. Mỗi từ (token) trong câu sẽ được gán một nhãn xác định vai trò của nó trong mối liên hệ với một thực thể tên riêng.

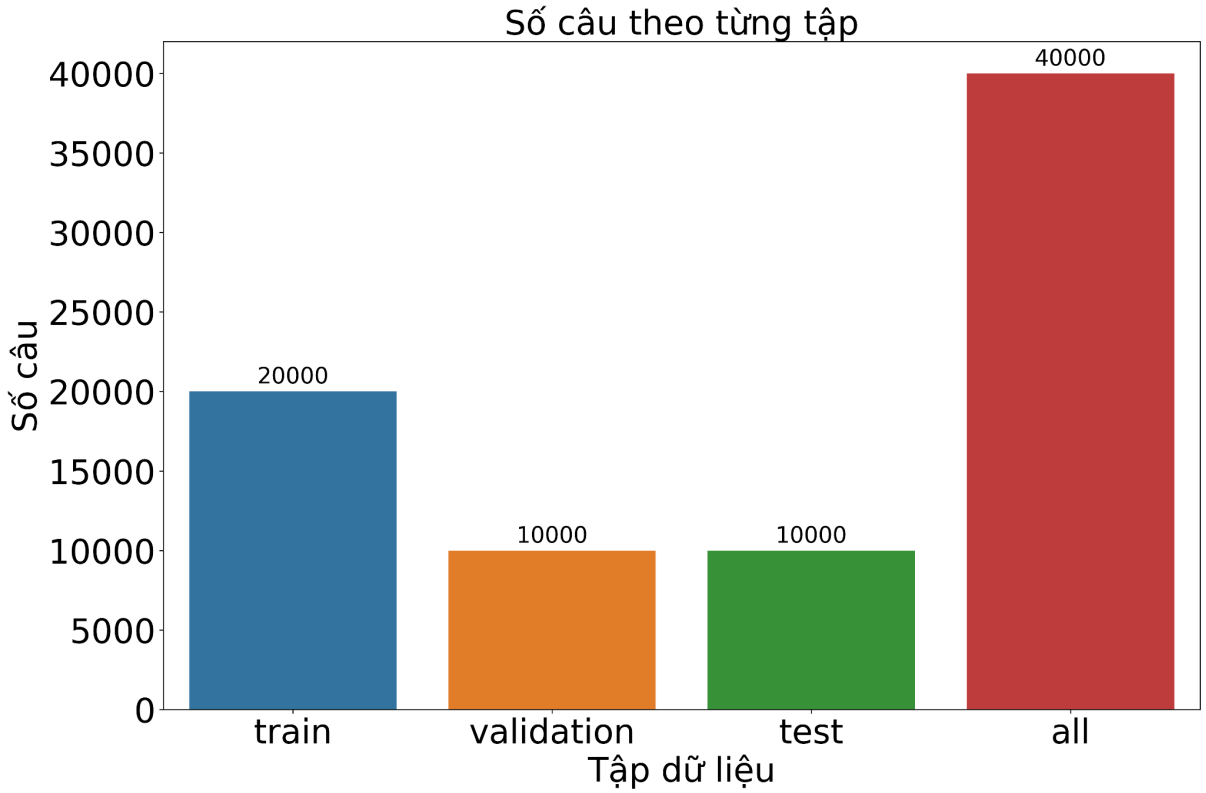
- **B-X** (Begin): đánh dấu token đầu tiên của một thực thể loại X (ví dụ: B-PER cho từ đầu tiên của tên người).
- **I-X** (Inside): đánh dấu các token nằm bên trong thực thể loại X (ví dụ: I-ORG là phần tiếp theo của một tên tổ chức).
- **O** (Outside): đánh dấu các token không thuộc bất kỳ thực thể tên riêng nào.

Ba loại thực thể chính được sử dụng trong chú thích là:

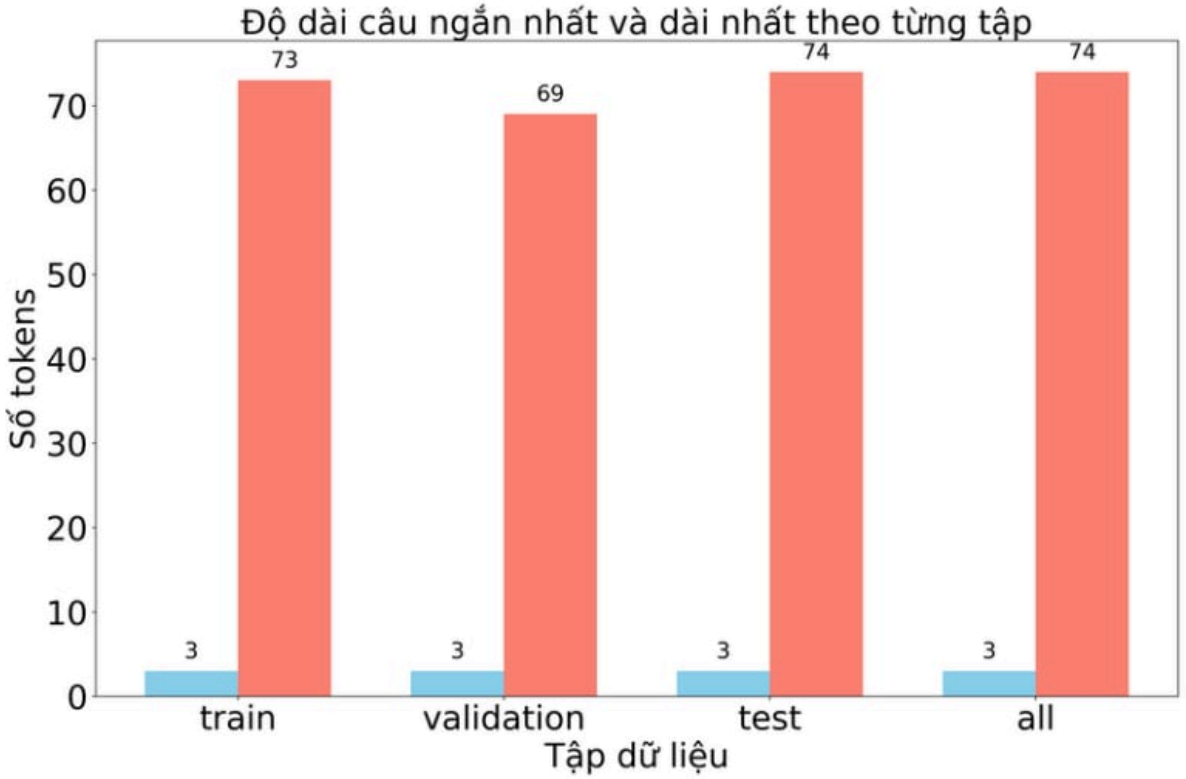
- PER: Tên người (Person)
- ORG: Tên tổ chức (Organization)
- LOC: Tên địa điểm (Location)

2.3 Thống kê ngữ liệu

Để hiểu rõ hơn về tập ngữ liệu WikiAnn tiếng Việt được sử dụng trong bài toán gán nhãn thực thể, nhóm đã tiến hành thống kê các số liệu như : phân bố nhãn, độ dài câu dài nhất (ngắn nhất), tổng số tokens, số lượng câu , số lượng từ vựng (vocab size).



Hình 1. Số lượng câu tổng và theo từng tập



Hình 2. Câu dài nhất và câu ngắn nhất trong các tập dữ liệu

Tập dữ liệu	Tổng số token	Vocab size
train	129439	17881
test	64967	11683
val	64222	11400

Bảng 1. Tổng số token và vocab size theo từng tập dữ liệu

Bảng 1 trình bày các thông tin thống kê cơ bản của tập dữ liệu tiếng Việt từ bộ wikiann, bao gồm ba tập con: train, validation và test. Các chỉ số bao gồm:

- **Tổng số token**: Tổng số từ (token) trong mỗi tập, bao gồm cả dấu câu, như ".", "," , "(" , ")",…

- **Vocab size**: Số lượng từ vựng duy nhất xuất hiện trong mỗi tập, cũng bao gồm cả dấu câu nếu chúng xuất hiện như một token riêng biệt.

Những thống kê này phản ánh quy mô và mức độ phong phú từ vựng của từng tập. Tập train có quy mô lớn nhất và chứa nhiều từ vựng nhất, đảm bảo độ đa dạng khi huấn luyện mô hình. Các tập test và validation có quy mô nhỏ hơn, đóng vai trò đánh giá khả năng tổng quát của mô hình.

Tập dữ liệu	O	B-PER	I-PER	B-ORG	I-ORG	B-LOC	I-LOC
train	47483	7470	14983	7364	27674	7588	16877
test	24295	3884	7787	3704	13562	3717	8018
val	23398	3738	7650	3738	13607	3832	8376

Bảng 2. Phân bố nhãn theo từng tập dữ liệu

Bảng 2 trình bày phân bố số lượng token theo từng nhãn thực thể (entity tag) trong ba tập dữ liệu: **train**, **test**, và **validation** của bộ dữ liệu wikiann tiếng Việt.

- Nhãn **O** (Outside) chiếm tỷ lệ lớn nhất trong cả ba tập, thể hiện các token không nằm trong bất kỳ thực thể nào.
  Nhãn **B-X** đại diện cho token bắt đầu của một thực thể: B-PER (tên người), B-ORG (tổ chức), B-LOC (địa điểm).
- Nhãn **I -X** đại diện cho các token bên trong một thực thể (nối tiếp B-X).

2.4 Phân tích ngữ liệu

Để có cái nhìn tổng quát về bộ ngữ liệu, nhóm tiến hành đánh giá 70 mẫu ngẫu nhiên từ cả 3 tập train (30), validation (15) và test (25). Với mỗi mẫu, chỉ quan tâm đánh giá 3 trường: *tokens, labels* và *span*

TẬP TRAIN				
Index	**tokens**	**labels**	**span**	**Đánh giá**
16558	['đổi', 'Fukui', '(', 'thành', 'phố', ')']	[0, 5, 6, 6, 6, 6]	['LOC: Fukui (thành phố)']	Nhãn của token thứ 3 trở đi bị sai, nó không thuộc cùng một thực thể
788	['-', 'Cá', 'sấu', 'Gena', '"']	[0, 1, 2, 2, 0]	['PER: Cá sấu Gena']	Nhãn đúng, đây là một nhân vật hư cấu trong văn học
11665	['François-René', 'de', 'Chateaubriand']	[1, 2, 2]	['PER: François-René de Chateaubriand']	Nhãn này đúng vì đây là một tên riêng của Nhà văn và cựu Bộ trưởng Bộ châu Âu và Ngoại giao Pháp
1670	['Victoria', 'Azarenka', '"', '(', 'Vòng', 'bốn', ')']	[1, 2, 0, 0, 0, 0, 0]	['PER: Victoria Azarenka']	Nhãn này đúng, vì đây cũng là tên riêng của một vận động viên
4232	['"', '"', '"', '"', '-', 'Petra', 'Susanna', 'Schürmann']	[0, 0, 0, 0, 0, 1, 2, 2]	['PER: Petra Susanna Schürmann']	Nhãn này đúng, vì đây cũng là tên riêng của một người mẫu
2233	['Lục', 'quân', 'Đế', 'quốc', 'Nhật', 'Bản']	[3, 4, 4, 4, 4, 4]	['ORG: Lục quân Đế quốc Nhật Bản']	Mẫu này đơn giản là tên một tổ chức quân sự, phù hợp với nhãn
4064	['đổi', 'Nicôla', 'thành', 'Myra']	[0, 1, 2, 2]	['PER: Nicôla thành Myra']	Đây là tên gọi của một người, nhãn phù hợp
11320	['"', '"', 'Diane', 'Keaton', '"', '"']	[0, 0, 1, 2, 0, 0]	['PER: Diane Keaton']	Đây cũng là tên gọi riêng nên phù hợp với nhãn. Các ký tự thừa cũng được đánh nhãn đúng
11533	['Đại', 'hội', 'đại', 'biểu', 'Nhân', 'dân', 'toàn', 'quốc', '(',	[3, 4, 4, 4, 4, 4, 4, 4, 4, 4,	['ORG: Đại hội đại biểu Nhân	Nhãn của 8 tokens đầu hợp lý, nhưng 4

	'Trung', 'Quốc', ')']	4, 4]	dân toàn quốc (Trung Quốc)']	tokens cuối thì lại không hợp lý, cặp dấu “(“, “)” không nên có nhãn ORG, còn “Trung” và “Quốc” là tên một quốc gia và một vùng lãnh thổ, nên phù hợp với B-LOC hơn
17969	['Ostend', '(', '69,845', ')']	[5, 0, 0, 0]	['LOC: Ostend']	Tên của một thành phố, nhãn phù hợp
6192	["""", 'Bulbophyllum', 'falculicorne', """", 'J.J.Sm', '.']	[0, 5, 6, 0, 0, 0]	['LOC: Bulbophyllum falculicorne']	Đây là tên gọi khoa học của một loài hoa lan, không thuộc 3 nhãn PER, ORG hay LOC, nên gán là O thì phù hợp hơn. “J.J.Sm” cũng là tên riêng của nhà khoa học nên phải gán nhãn I-PER
13854	['Angeles', '(', 'Philippines', ')']	[5, 6, 6, 6]	['LOC: Angeles (Philippines)']	Tên của 1 thành phố ở Philippines nên phù hợp với nhãn. Nhưng cặp ngoặc “()” thì không nên gán nhãn và “Philippines” lại là tên của quốc gia, nên phải là B-LOC
17763	['2004', 'đến', '2007', '-', ""Không", 'tham', 'dự', """]	[3, 0, 3, 0, 0, 0, 0, 0]	['ORG: 2004', 'ORG: 2007']	Mẫu này không xuất hiện tên người hay tổ chức, địa điểm nên phù hợp với hơn với toàn bộ nhãn O
8750	['Joker', '(', 'truyện', 'tranh', ')']	[1, 2, 2, 2, 2]	['PER: Joker (truyện tranh)']	Token “Joker” phù hợp với nhãn, nhưng những tokens sau thì không phải một thực thể chung với “Joker” nên phải là O
824	['Về', 'nhì', 'Euro', '2008']	[0, 0, 3, 4]	['ORG: Euro 2008']	Nhãn phù hợp

4402	["""", """", 'Jessica', 'Lange', """", """"]	[0, 0, 1, 2, 0, 0]	['PER: Jessica Lange']	Nhãn này là đúng, gán Outside cho dấu câu và Jessica Lange là tên diễn viên
18651	["""", 'Aramides', 'mangle', """", """"]	[0, 5, 6, 0, 0]	['LOC: Aramides mangle']	Nhãn đúng dấu câu là outside, nhưng sai vì Aramides mangle là tên loài chim và nhãn đúng là 0 (outside)
2067	['đổi', 'Xã', 'Union', ',', 'Quận', 'Snyder', ',', 'Pennsylvania']	[0, 5, 6, 6, 6, 6, 6, 6]	['LOC: Xã Union , Quận Snyder , Pennsylvania']	Nhãn gán sai cho dấu câu, còn lại nhãn LOC là đúng cho nơi trong tiểu bang của Hoa Kỳ
8358	['Công', 'ty', 'Thông', 'tin', 'di', 'động', 'Việt', 'Nam']	[3, 4, 4, 4, 4, 4, 4, 4]	['ORG: Công ty Thông tin di động Việt Nam']	Nhãn gán đúng cho tên công ty
3863	['Hiệp', 'hội', 'các', 'quốc', 'gia', 'Đông', 'Nam', 'Á']	[3, 4, 4, 4, 4, 4, 4, 4]	['ORG: Hiệp hội các quốc gia Đông Nam Á']	Nhãn gán đúng cho tên hiệp hội
16234	['đổi', 'Sonata', 'số', '16', 'dành', 'cho', 'dương', 'cầm', '(', 'Mozart', ')']	[0, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4]	['ORG: Sonata số 16 dành cho dương cầm (Mozart)']	Nhãn được gán sai, "Sonata số 16 dành cho dương cầm": là tên một tác phẩm âm nhạc (Sonata No.16 for piano) nên cần được gán là 0 (outside), Mozart là tên người nên cần được gán B-PER
14728	['The', 'Edge', '-', 'guitar', ',', 'hát', 'phụ']	[1, 2, 0, 0, 0, 0, 0]	['PER: The Edge']	Nhãn được gán đúng, The Edge – tay guitar, hát phụ của ban nhạc U2 được gán là tên người
15474	['Hydrangea', 'longifolia', """", '.']	[5, 6, 0, 0]	['LOC: Hydrangea longifolia']	Nhãn gán sai, Hydrangea longifolia là tên khoa học của

				một loài thực vật (trong chi Cẩm tú cầu – Hydrangea)
12439	['đổi', 'Philippos', 'II', 'của', 'Macedonia']	[0, 1, 2, 2, 2]	['PER: Philippos II của Macedonia']	Nhãn được gán đúng Philippos II của Macedonia là một cụm để chỉ tên người
6879	['Xã', 'của', 'tỉnh', 'Ardennes']	[5, 6, 6, 6]	['LOC: Xã của tỉnh Ardennes']	Nhãn được gán sai, chỉ Ardennes là tên một tỉnh ở Pháp cần được gán nhãn.
3075	['Lăng', 'mộ', 'của', 'Cyrus', 'Đại', 'đế']	[3, 4, 4, 4, 4, 4]	['ORG: Lăng mộ của Cyrus Đại đế']	Nhãn được gán sai, chỉ có Cyrus Đại đế cần được gán nhãn PER, còn lăng mộ của thì gán là O.
15986	['đổi', 'Huỳnh', 'Hiểu', 'Minh']	[0, 1, 2, 2]	['PER: Huỳnh Hiểu Minh']	Nhãn được gán đúng, Huỳnh Hiểu Minh là tên diễn viên, ca sĩ.
928	['đổi', 'Hội', 'đồng', 'Nhân', 'dân', 'Tối', 'cao', '(', 'Bắc', 'Triều', 'Tiên', ')']	[0, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]	['ORG: Hội đồng Nhân dân Tối cao (Bắc Triều Tiên)']	Nhãn được gán sai, Bắc Triều Tiên là tên nước nên được gán LOC , còn dấu ngoặc thì gán O.
12773	['đổi', 'Nam', 'Tề', 'Vũ', 'Đế']	[0, 1, 2, 2, 2]	['PER: Nam Tề Vũ Đế']	Nhãn được gán đúng, Nam Tề Vũ Đế là tên người.
14180	['Ông', 'là', 'con', 'trai', 'cả', 'của', 'Thủ', 'tướng', 'Nguyễn', 'Tấn', 'Dũng', '.']	[0, 0, 0, 0, 0, 0, 3, 4, 1, 2, 2, 0]	['ORG: Thủ tướng', 'PER: Nguyễn Tấn Dũng']	Nhãn được gán sai cho Thủ tướng, đây là chức vị và cần được gán nhãn O.

Bảng 3. Bảng đánh giá trên train

TẬP TEST				
Index	**tokens**	**labels**	**span**	**Đánh giá**
9952	["'", "'", 'Gustav', 'Mahler', "'", "'"]	[0, 0, 1, 2, 0, 0]	['PER: Gustav Mahler']	Nhãn được gán đúng, đây là tên nhà soạn

				nhạc.
34	['đổi', 'Đệ', 'nhất', 'Quốc', 'tế']	[0, 3, 4, 4, 4]	['ORG: Đệ nhất Quốc tế']	Nhãn được gán đúng, đây là cách gọi tiếng Việt của Đệ Nhất Quốc Tế Cộng Sản (*First International*), một tổ chức chính trị quốc tế do Karl Marx sáng lập năm 1864.
7297	['đổi', 'Cá', 'vạng', 'mỡ']	[0, 5, 6, 6]	['LOC: Cá vạng mỡ']	Nhãn gán sai, đây là tên của một loại cá, cần được gán nhãn O
4363	['Con', 'đường', 'tơ', 'lụa']	[3, 4, 4, 4]	['ORG: Con đường tơ lụa']	Nhãn được gán sai, đây có thể được xem là tên địa danh trong lịch sử nên có thể gán nhãn LOC
3748	['Longstreet', ',', 'James', '.']	[1, 2, 2, 0]	['PER: Longstreet , James']	Nhãn được gán gần đúng, dấu phẩy ‘,’ cần được gán nhãn O
9685	['Samir', 'Nasri', '(', '2004-2008', ')']	[1, 2, 0, 0, 0]	['PER: Samir Nasri']	Nhãn được gán đúng, Samir Nasri là tên cầu thủ đá banh.
1674	['đổi', 'Ludwig', 'Andreas', 'Feuerbach']	[0, 1, 2, 2]	['PER: Ludwig Andreas Feuerbach']	Nhãn được gán đúng, Ludwig Andreas Feuerbach là tên đầy đủ của một triết gia người Đức
5200	['Tanaka', 'Mayumi', 'vai', 'Monkey', 'D.', 'Luffy']	[1, 2, 0, 1, 2, 2]	['PER: Tanaka Mayumi', 'PER: Monkey D. Luffy']	Nhãn được gán đúng, Tanaka Mayumi là tên diễn viên lồng tiếng, còn Monkey D. Luffy là tên nhân vật.
501	['Gudendorf', '(', '425', ')']	[5, 0, 0, 0]	['LOC: Gudendorf']	Nhãn gán đúng, Gudendorf" là tên một địa danh — cụ thể là một thị trấn nhỏ ở Đức

365	['đổi', 'Phật', 'giáo', 'Nguyên', 'thủy']	[0, 3, 4, 4, 4]	['ORG: Phật giáo Nguyên thủy']	Nhãn đúng vì Phật giáo cũng là một tổ chức có cấp bậc
5893	['Đảng', 'Quốc', 'Đại', 'Ấn', 'Độ']	[3, 4, 4, 4, 4]	['ORG: Đảng Quốc Đại Ấn Độ']	Nhãn đúng, vì Đảng là một tổ chức
5286	['"', '"', 'Fernando', 'Alonso', '"', '"']	[0, 0, 1, 2, 0, 0]	['PER: Fernando Alonso']	Nhãn đúng vì đây là tên của một vận động viên đua xe
1562	['Di', 'sản', 'thế', 'giới']	[3, 4, 4, 4]	['ORG: Di sản thế giới']	Di sản thế giới không phải là một tổ chức, có lẽ nhãn được gán từ categories trên wikipedia của UNESCO. Nhãn phù hợp là O vì đây không phải tên riêng hay tổ chức, vị trí
4520	['Leeteuk', ',', 'Eunhyuk', ',', 'Kyuhyun', ',', 'Yesung']	[1, 0, 1, 0, 1, 0, 1]	['PER: Leeteuk', 'PER: Eunhyuk', 'PER: Kyuhyun', 'PER: Yesung']	Nhãn đúng vì đây là tên các thành viên trong 1 nhóm nhạc Hàn Quốc
5772	['Trị', 'số', 'gần', 'đúng', 'trong', 'hệ', 'thống', 'SI']	[0, 0, 0, 0, 0, 0, 0, 3]	['ORG: SI']	Nhãn không phù hợp với "SI" vì đây là một hệ đo lường chứ không phải một tổ chức
5759	['Việt', 'Nam', 'Quốc', 'dân', 'Đảng']	[3, 4, 4, 4, 4]	['ORG: Việt Nam Quốc dân Đảng']	Nhãn đúng vì Đảng là một tổ chức
8163	['đổi', 'Mậu', 'dịch', 'Nanban']	[0, 3, 4, 4]	['ORG: Mậu dịch Nanban']	Nhãn không hợp lý vì đây là một thời kỳ trong kinh tế Nhật Bản, nên nên gán là O cho toàn bộ tokens
3663	['đổi', 'Thiên', 'hoàng', 'Go-Daigo']	[0, 1, 2, 2]	['PER: Thiên hoàng Go-Daigo']	Nhãn đúng vì đây là tên của một người
9315	['Peru', '(', '1', ')', '.']	[5, 0, 0, 0, 0]	['LOC: Peru']	Nhãn hợp lý vì đây là

				tên của một quốc gia
1210	['*2004', '/', '05', '-', 'Olympique', 'Lyonnais']	[0, 0, 0, 0, 3, 4]	['ORG: Olympique Lyonnais']	Nhãn đúng vì đây là tên một câu lạc bộ bóng đá ở Pháp
9517	['đổi', 'Chỉ', 'thị', 'về', 'hạn', 'chế', 'các', 'chất', 'nguy', 'hiểm']	[0, 3, 4, 4, 4, 4, 4, 4, 4, 4]	['ORG: Chỉ thị về hạn chế các chất nguy hiểm']	Nhãn không hợp lý vì một nhóm chỉ thị không phải là một tổ chức
814	['Liên', 'đoàn', 'Quốc', 'tế', 'về', 'Hoá', 'học', 'Thuần', 'túy', 'và', 'Ứng', 'dụng', '(', 'IUPAC', ')']	[3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0, 3, 0]	['ORG: Liên đoàn Quốc tế về Hoá học Thuần túy và Ứng dụng', 'ORG: IUPAC']	Nhãn hợp lý vì cả 2 thực thể tìm được đều là tên của một tổ chứ, một cái là tên đầy đủ, cái kia là tên viết tắt
4072	['đổi', 'Football', 'League', 'One']	[0, 3, 4, 4]	['ORG: Football League One']	Đúng vì đây là một tổ chứ bóng đá
9925	['Borut', 'Pahor', '(', '2012-nay', ')']	[1, 2, 0, 0, 0]	['PER: Borut Pahor']	Đúng vì đây là tên một người
5477	['Edward', 'IV', 'của', 'Anh']	[1, 2, 2, 2]	['PER: Edward IV của Anh']	Đúng vì đây là tên một người, cụ thể là cựu vua Anh

Bảng 4. Bảng đánh giá trên test

TẬP VALIDATION				
Index	**tokens**	**labels**	**span**	**Đánh giá**
416	['====', 'San', 'José', 'del', 'Valle', '====]	[0, 5, 6, 6, 6, 0]	['LOC: San José del Valle']	Nhãn được gán đúng, "San José del Valle" là tên đầy đủ của một địa danh, cụ thể là một thị trấn tại tỉnh Cádiz, Tây Ban Nha.
8870	['Công', 'chúa', 'Beatrice', 'xứ', 'York']	[1, 2, 2, 2, 2]	['PER: Công chúa Beatrice xứ York']	Nhãn đúng vì đây là biệt danh của một người, giống như "Nicôla thành Myra"
150	['Thiên', 'hoàng', 'Minh', 'Trị']	[1, 2, 2, 2]	['PER: Thiên hoàng Minh Trị']	Nhãn đúng vì đây là tên và chức vị của

				người
6245	['Đế', 'quốc', 'Maratha']	[3, 4, 4]	['ORG: Đế quốc Maratha']	Nhãn được gán đúng, Đế quốc Maratha có thể được gán LOC hoặc ORG
3548	['"', 'Someday', '"', '–', '3:57']	[0, 3, 0, 0, 0]	['ORG: Someday']	Nhãn gán sai, Some day là tên bài hát, cần được gán nhãn O
4853	['Chu', 'Bút', 'Sướng']	[1, 2, 2]	['PER: Chu Bút Sướng']	Nhãn đúng vì đây là tên của một người nổi tiếng
613	['đổi', 'Chu', 'Điệu', 'Vương']	[0, 1, 2, 2]	['PER: Chu Điệu Vương']	Nhãn phù hợp vì đây là tên một người
5023	['đổi', 'British', 'Phonographic', 'Industry']	[0, 3, 4, 4]	['ORG: British Phonographic Industry']	Nhãn đúng vì đây là tên của hiệp hội ghi âm Anh
5647	['Richard', 'Gasquet', '"', '(', 'Vòng', '1', ',', 'thua', 'Tommy', 'Haas', ')']	[1, 2, 0, 0, 0, 0, 0, 0, 1, 2, 0]	['PER: Richard Gasquet', 'PER: Tommy Haas']	Nhãn đúng vì đây là tên của 2 người
4404	['đổi', 'Quận', 'Greene', ',', 'Bắc', 'Carolina']	[0, 5, 6, 6, 6, 6]	['LOC: Quận Greene , Bắc Carolina']	Nhãn đúng vì đây là tên một vị trí địa lý
690	['"', '"', 'Brooklyn', '"', '"']	[0, 0, 5, 0, 0]	['LOC: Brooklyn']	Nhãn đúng vì đây là tên một quận
5310	['Artem', 'Ivanovich', 'Mikoyan']	[1, 2, 2]	['PER: Artem Ivanovich Mikoyan']	Đúng vì đây là tên của một người
9863	['Tổng', 'thống', 'Hoa', 'Kỳ']	[3, 4, 4, 4]	['ORG: Tổng thống Hoa Kỳ']	Nhãn sai vì đây là tên một chức vụ chứ không phải một tổ chứ, nên gán là O cho toàn bộ tokens
3394	['Một', 'tuần', 'sau', 'đó', ',', 'thành', 'viên', 'Kim', 'Kibum', '-', 'người', 'đã', 'tham', 'gia', 'rất', 'ít', 'vào', 'album', 'trước', 'đó', 'Sorry', ',', 'Sorry', '(', 'do',	[0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 4, 3, 0, 0,	['PER: Kim Kibum', 'ORG: Sorry ,', 'ORG: Sorry']	Nhãn về PER đúng, nhưng những nhãn về ORG thì lại sai, do đó là tên của một album, thậm chí còn không

	'bận', 'rộn', 'với', 'việc', 'theo', 'đuổi', 'nghiệp', 'diễn', 'xuất', 'của', 'mình', ')', ',', 'đã', 'phát', 'biểu', 'về', 'khả', 'năng', 'tham', 'gia', 'album', 'lần', 'này', 'trong', 'một', 'buổi', 'phỏng', 'vấn', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]		phải hai thực thể riêng biệt. Nhãn đúng nên là O vì tên album không phù hợp với ORG, LOC hay PER
8619	['Đảng', 'Cộng', 'sản', 'Việt', 'Nam']	[3, 4, 4, 4, 4]	['ORG: Đảng Cộng sản Việt Nam']	Đúng vì Đảng là một tổ chức

Bảng 5. Bảng đánh giá trên validation

Chương 3: PHƯƠNG PHÁP SỬ DỤNG

3.1. PhoBERT

3.1.1. PhoBERT-base

PhoBERT-base là một mô hình ngôn ngữ tiền huấn luyện được thiết kế dành riêng cho tiếng Việt, được phát triển bởi VinAI Research. Mô hình này kế thừa kiến trúc từ RoBERTa – một biến thể cải tiến của BERT – và được huấn luyện trên tập dữ liệu tiếng Việt quy mô lớn sau khi đã qua bước tách từ bằng công cụ RDRSegmenter. PhoBERT giúp cải thiện đáng kể hiệu suất trong các bài toán xử lý ngôn ngữ tự nhiên tiếng Việt như phân loại văn bản, gán nhãn thực thể (NER), phân tích cảm xúc, v.v.

Cấu trúc tổng thể

PhoBERT-base sử dụng kiến trúc Transformer encoder-only, gồm 12 tầng mã hóa (encoder layers). Mỗi tầng bao gồm hai khối chính:

- Multi-head self-attention: học mối quan hệ giữa các token trong chuỗi theo cách hai chiều.
- Feed-forward network (FFN): hai lớp tuyến tính với hàm kích hoạt phi tuyến GELU.

Ngoài ra, mỗi tầng đều có residual connection và layer normalization để tăng độ ổn định và hiệu quả trong huấn luyện.

Lớp Embedding

Tương tự như BERT, PhoBERT có ba loại embedding:

- **Token embedding**: ánh xạ mỗi subword sang vector không gian 768 chiều.
- **Position embedding**: mã hóa vị trí tương đối của token trong câu.

- **Segment embedding**: dù PhoBERT không sử dụng nhiệm vụ Next Sentence Prediction, embedding này vẫn được giữ lại để tương thích với kiến trúc ban đầu.

Tổng embedding đầu vào là tổng ba loại embedding trên và được đưa vào khối encoder.

Bộ mã hóa Transformer

Chuỗi embedding sau khi được tạo sẽ đi qua 12 tầng Transformer encoder. Mỗi tầng có:

- **Multi-head self-attention**: chia không gian biểu diễn thành 12 đầu (head), mỗi head học một khía cạnh ngữ nghĩa khác nhau của token.

- **Feed-forward network**: mở rộng chiều lên 3072, sau đó giảm lại về 768 để giữ nguyên kích thước.

Tokenizer và tiền xử lý

Khác với các mô hình BERT đa ngôn ngữ, PhoBERT yêu cầu **tách từ tiếng Việt trước khi token hóa**. Cụ thể:

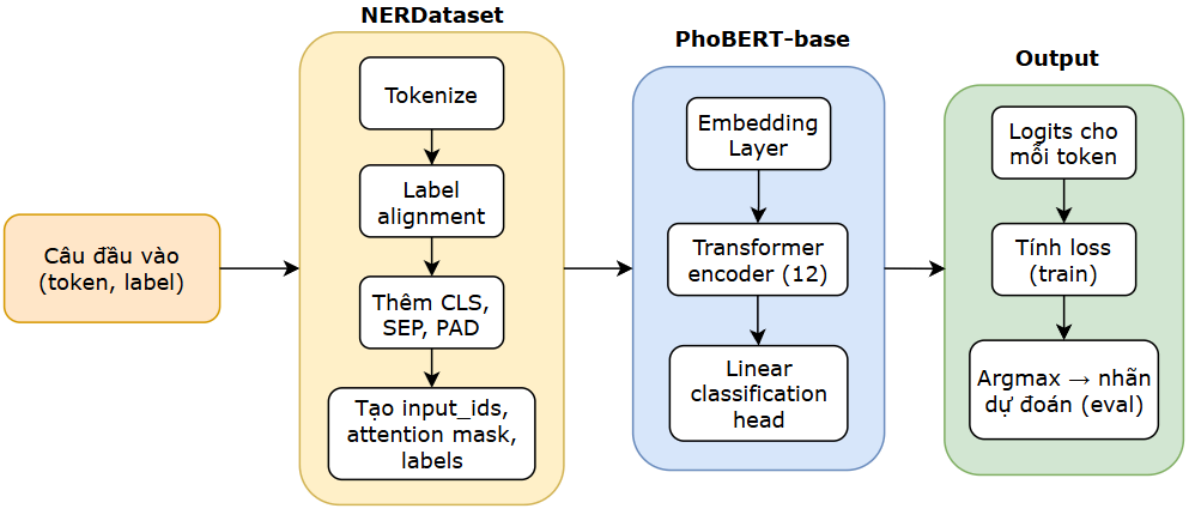
- Văn bản đầu vào được tách từ bằng **RDRSegmenter**.
- Sau đó, các từ đã tách sẽ được xử lý bởi **SentencePiece tokenizer** để phân thành các subword phù hợp với từ điển học của PhoBERT.

Đầu ra của mô hình PhoBERT-base

Tương ứng với mỗi token đầu vào, PhoBERT trả về một vector ẩn có chiều dài 768. Trong các bài toán như NER (nhận diện thực thể), đầu ra này sẽ được đưa qua một lớp tuyến tính để dự đoán nhãn tương ứng cho từng token.

3.1.2. Luồng dữ liệu xử lý với PhoBERT-base cho bài toán NER

Để áp dụng PhoBERT-base cho bài toán gán nhãn thực thể (NER), dữ liệu đầu vào phải được xử lý theo một chu trình cụ thể nhằm tương thích với kiến trúc của mô hình. Dưới đây là trình tự các bước xử lý dữ liệu từ đầu vào đến đầu ra của mô hình.



Dữ liệu ban đầu bao gồm:

- Một danh sách các câu, trong đó mỗi câu là một danh sách các từ (token).
- Một danh sách tương ứng các nhãn (label) theo định dạng IOB2 cho mỗi từ.

Ví dụ: "*Thành phố Fukui của Nhật Bản*" sẽ được đưa vào với dạng *["Thành", "phố", "Fukui", "của", "Nhật", "Bản"]*

Tạo tập dữ liệu huấn luyện (NERDataset)

Lớp NERDataset có nhiệm vụ chuyển đổi dữ liệu thô thành các tensor đầu vào cho mô hình. Các bước xử lý chính bao gồm:

Token hóa subword (SentencePiece tokenizer)

- Mỗi từ sẽ được token hóa thành 1 hoặc nhiều subword.
- PhoBERT sử dụng một từ điển subword cố định đã được huấn luyện sẵn. Với các từ không phổ biến (hoặc viết hoa, viết sai chính tả, hoặc ở vị trí không quen), tokenizer sẽ không nhận dạng được từ đầy đủ và tự động chia nhỏ từ đó thành nhiều subword token (thường có @@ ở cuối để biểu thị rằng đây không phải là token kết thúc từ).
    - Ví dụ : ['Thành', 'phố', 'Fuk@@', 'ui', 'của', 'Nhật', 'Bản']. Ở đây tokenizer không nhận dạng được "Fukui" nên đã chia thành "Fuk@@" và "ui".

Căn chỉnh nhãn (label alignment)

- Với mỗi từ ban đầu, nhãn của từ đầu tiên được giữ nguyên.

- Các subword còn lại được gán nhãn -100 để **bỏ qua trong quá trình tính toán hàm mất mát**.
    - Ví dụ : [5, 6, 6, -100, 0, 5, 6, 0]. Đây là nhãn cho câu trên sau khi label alignment. -100 là giá trị được gán cho 'ui' sau 'Fuk@@'

Thêm các token đặc biệt

- <s> được thêm vào đầu chuỗi, </s> vào cuối.
- Các token đặc biệt này được gán nhãn -100.
    - Ví dụ : ['<s>', 'Thành', 'phố', 'Fuk@@', 'ui', 'của', 'Nhật', 'Bản', '.', '</s>']. Các token đặc biệt đã được thêm vào đầu và cuối câu.

Padding

- Nếu chuỗi dài hơn max_len, nó được cắt bớt nhưng sẽ không xảy ra vì max_len = 128 và câu dài nhất chỉ có 74 tokens.
- Nếu ngắn hơn, nó được đệm bằng [PAD], nhãn của [PAD] là -100.
    - Ví dụ : [-100, 5, 6, 6, -100, 0, 5, 6, 0, -100].
    - Rất nhiều nhãn -100 cho [PAD].
- Cập nhật *input_ids, attention_mask, label_ids*: trong đó, *input_ids* là id của subword hoặc token được tạo ra từ tokenizer, *attention_mask* là một danh sách nhị phân (chỉ gồm 1 và 0), cho biết token nào cần được mô hình chú ý đến trong quá trình xử lý, *label_ids* là nhãn của từng token, mang giá trị -100 nếu bỏ qua.

Mỗi mẫu sẽ trả về :

"input_ids" : ID của token/subword

"attention_mask" : 1 nếu là token thật, 0 nếu là PAD

"labels" : nhãn của từng token, -100 nếu bỏ qua

Dữ liệu sau tiền xử lý được đưa vào mô hình phobert-base, với các bước bên trong như sau:

Embedding Layer

- Nhận input_ids và ánh xạ mỗi token thành vector ẩn 768 chiều.
- Thêm position embedding và segment embedding.

Transformer Encoder

- Chuỗi embedding được truyền qua 12 tầng Transformer.
- Mỗi tầng gồm:
    - Multi-head self-attention (12 heads)
    - Feed-forward network (FFN)
    - LayerNorm + residual connection

Token Classification Head

- Đầu ra từ encoder có shape: *[batch_size, seq_len, hidden_size]*.

  Với ví dụ trên, đầu ra từ encoder sẽ có shape là: [1, 10, 768]

  Với :

  batch_size = 1 : số câu đầu vào, ở đây do dự đoán từng câu nên là 1.

  seq_len = 10 : Câu sau khi token hóa có **10 tokens**, bao gồm token thật (subword) và token đặc biệt như <s>, </s>

  hidden_size = 768 : Mỗi token được biểu diễn bằng 1 vector có 768 chiều (PhoBERT-base)

- Một lớp tuyến tính (linear layer) ánh xạ mỗi vector ẩn sang không gian nhãn:

  logits = Linear(hidden_size, num_labels)

  → *shape: [batch_size, max_len, num_labels]*

  Với ví dụ trên, logists có shape là: [1, 128, 7]

  Với :

  batch_size = 1 : số câu đầu vào, ở đây do dự đoán từng câu nên là 1.

  max_len = 128 : được định nghĩa từ trước, cho biết đây là độ dài max của câu.

num_labels = 7 : bài toán cần dự đoán 7 nhãn

Hàm mất mát (Loss) và Dự đoán

Tính loss

- Mô hình sử dụng CrossEntropyLoss với *ignore_index* = -100.
- Chỉ các token gốc (không phải subword hoặc token đặc biệt) mới được tính loss.

Dự đoán

- Dự đoán nhãn bằng cách chọn nhãn có xác suất cao nhất tại mỗi vị trí:

preds = torch.argmax(logits, dim=-1)

- Sau đó, bỏ các vị trí có nhãn -100 và so sánh với nhãn thật để tính F1, Precision, Recall.

<s> : [2.7531, -0.9471, -1.7047, 0.0955, -2.0954, 4.1578, -0.0298],
Thành : [0.9630, -0.7678, -1.6563, 0.6971, -1.9340, 6.5291, -1.3280],
phố : [0.6471, -2.2295, -1.4591, -2.2075, -0.7278, 0.5898, 5.7021],
Fuk@@ : [-1.0981, -1.6403, -1.8808, -1.1438, -0.8789, 1.8642, 5.7527],
ui : [-1.1864, -1.6130, -1.6270, -1.5515, -0.3068, 0.0313, 6.4669],
của : [7.0193, -2.0248, -1.1557, -2.0222, -0.9679, -1.3171, 0.0096],
Nhật : [-1.0145, -1.1972, -1.4499, 0.5740, -1.3551, 6.7935, -0.0149],
Bản : [-1.1984, -1.7551, -1.4588, -1.6797, -0.3015, 0.1471, 6.4970],
. : [7.4004, -1.7814, -1.2032, -1.7057, -1.1786, -1.1262, -0.7275],
</s>: [2.7531, -0.9471, -1.7047, 0.0955, -2.0954, 4.1578, -0.0298],]

Đây là 10 hàng đầu tiên của logits, tương ứng với các token trên. Hai token <s> và </s> cũng như các giá trị PAD có label_ids là -100 sẽ bị loại bỏ, và giữ lại những token/subword của câu gốc. Mỗi token sẽ có một list gồm 7 giá trị tương ứng với khả năng thuộc về 7 nhãn của đề bài, theo thứ tự lần lượt là: O, B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC. Sau khi áp dụng hàm argmax, sẽ lấy ra vị trí nhãn có giá trị cao nhất, từ đó có thể quyết định nhãn của token.
Với token "Fuk@@", có giá trị tại index số 6 (giá trị đầu có index là 0) là lớn nhất, vậy nên nó sẽ mang nhãn là I-LOC.

3.2. Model so sánh

Nhóm sẽ so sánh kết quả với 2 mô hình cổ điển là CRF và HMM.

Conditional Random Fields - CRF là là một mô hình đồ thị xác suất phân biệt, không hướng, được sử dụng để dự đoán chuỗi các nhãn đầu ra. Điểm khác biệt cốt lõi so với HMM là CRF trực tiếp mô hình hóa phân phối xác suất có điều kiện P(Y|X), trong đó Y là chuỗi các nhãn (ví dụ: các nhãn NER) và X là chuỗi các quan sát (ví dụ: các từ trong câu). Thay vì mô hình hóa cách các quan sát được sinh ra từ các trạng thái (như HMM), CRF tập trung vào việc mô hình hóa mối quan hệ giữa các quan sát và các nhãn dựa trên các hàm đặc trưng linh hoạt.

Hidden Markov Model - HMM là một mô hình thống kê xác suất, thường được sử dụng cho các bài toán nhận dạng mẫu theo trình tự và chuỗi thời gian. Trong NER, HMM có thể được áp dụng để mô hình hóa trình tự các nhãn thực thể (ví dụ: B-PER, I-PER, B-LOC, I-LOC, O) dựa trên chuỗi các từ trong câu.

Chương 4 : CÀI ĐẶT VÀ THỬ NGHIỆM

4.1. PhoBERT

4.1.1. Tokenization và Label Alignment

Mô hình PhoBERT sử dụng một loại tokenization riêng (WordPiece tokenization), thường chia một từ thành nhiều "subword token". Do đó, việc căn chỉnh các nhãn NER từ các từ ban đầu sang các subword token là rất quan trọng. Chúng em sử dụng AutoTokenizer để tải bộ tokenizer tương ứng với mô hình PhoBERT đã được huấn luyện trước. Bộ tokenizer này sẽ chuyển đổi văn bản đầu vào thành các input_ids và attention_mask mà mô hình PhoBERT yêu cầu.

Quá trình căn chỉnh nhãn được thực hiện bởi lớp NERDataset tùy chỉnh. Lớp này nhận vào danh sách các từ gốc và nhãn NER tương ứng của chúng. Đầu tiên, nó token hóa các từ này bằng tokenizer của PhoBERT, có tính đến việc cắt bớt (truncation) và tự động thêm padding để xử lý các chuỗi có độ dài khác nhau trong một batch. Sau đó, nó tạo ra một danh sách các label_ids mới cho các subword token. Các token đặc biệt của PhoBERT (như <s>, </s>, [PAD]) và các subword token **không phải là subword token đầu tiên của một từ gốc** sẽ được gán nhãn -100. Nhãn -100 được thư viện transformers sử dụng để bỏ qua các token này trong tính toán hàm mất mát. Đối với **subword token đầu tiên của mỗi từ gốc**, nhãn NER gốc của từ đó sẽ được gán.

4.1.2. Tải và cấu hình mô hình

Chúng em tải một mô hình phoBERT-base đã được huấn luyện trước và cấu hình nó cho nhiệm vụ phân loại token (Token Classification), đây chính là NER. Tham số num_labels được đặt bằng tổng số lượng nhãn NER mà chúng em đang dự đoán, để tầng đầu ra của mô hình phù hợp với bài toán của chúng em (cụ thể là 7 nhãn).

4.1.3. Cấu hình các đối số huấn luyện

Đối tượng TrainingArguments được sử dụng để định nghĩa tất cả các siêu tham số và cấu hình cho quá trình huấn luyện mô hình. Chúng em thiết lập output_dir là nơi các checkpoint của mô hình, file log và kết quả đầu ra khác sẽ được lưu.. Kích thước batch cho huấn luyện và đánh giá (per_device_train_batch_size, per_device_eval_batch_size) đều được đặt là 16. Mô hình sẽ được huấn luyện trong 5 num_train_epochs (số vòng lặp qua toàn bộ tập dữ liệu huấn luyện).

4.1.4. Thiết lập và huấn luyện Trainer

Cuối cùng, chúng em sử dụng lớp Trainer từ thư viện transformers, một lớp tiện ích mạnh mẽ giúp đơn giản hóa toàn bộ quá trình huấn luyện. Chúng em truyền vào mô hình phoBERT đã tải, các đối số huấn luyện đã cấu hình, tập dữ liệu huấn luyện và đánh giá đã được xử lý, bộ tokenizer. Bằng cách gọi trainer.train(), quá trình fine-tuning mô hình phoBERT trên tập dữ liệu đã chuẩn bị sẽ bắt đầu. Trong quá trình này, các trọng số của mô hình phoBERT sẽ được điều chỉnh để tối ưu hóa hiệu suất trên nhiệm vụ NER.

4.2. Conditional Random Fields - CRF

4.2.1. Chuẩn bị dữ liệu

Dữ liệu đầu vào cho mô hình CRF cần được chuyển đổi sang định dạng phù hợp. Mỗi "câu" (sentence) được biểu diễn dưới dạng một danh sách các cặp (từ, nhãn), ví dụ: [("Obama", "B-PER"), ("đã", "O"), ("đến", "O"), ("Hà Nội", "B-LOC")]. Để chuẩn bị dữ liệu cho huấn luyện, chúng em sử dụng hai hàm chính:

- **sent2features(sent)**: Hàm này nhận vào một câu (dưới dạng danh sách các cặp từ-nhãn) và trả về một danh sách các bộ đặc trưng cho mỗi từ trong câu. Mỗi bộ đặc trưng là một từ điển chứa các thông tin liên quan đến từ đó và ngữ cảnh xung quanh nó. Đây là đầu vào X cho mô hình CRF.
- **sent2labels(sent)**: Hàm này nhận vào một câu và trả về một danh sách các nhãn thực thể tương ứng với từng từ trong câu. Đây là đầu vào y cho mô hình CRF.

Dữ liệu được chia thành ba tập:

- X_train, y_train: Dữ liệu dùng để huấn luyện mô hình.
- X_val, y_val: Dữ liệu dùng để kiểm tra và tinh chỉnh mô hình trong quá trình phát triển (tập validation).
- X_test, y_test: Dữ liệu dùng để đánh giá hiệu suất cuối cùng của mô hình sau khi đã hoàn tất huấn luyện và tinh chỉnh.

4.2.2. Trích xuất đặc trưng

Hàm word2features(sent, i) là trái tim của quá trình trích xuất đặc trưng. Hàm này được thiết kế để tạo ra một tập hợp các đặc trưng cho từ thứ i trong câu sent. Các đặc trưng này cung cấp thông tin cần thiết để mô hình CRF học cách nhận diện các thực thể có tên:

- **Đặc trưng cơ bản (feats):**
    - 'bias': 1.0: Một hằng số bias giúp mô hình học một ngưỡng cơ bản.
    - 'word.lower': word.lower(): Dạng chữ thường của từ. Đặc trưng này giúp mô hình nhận diện các từ không phụ thuộc vào cách viết hoa ban đầu (ví dụ: "Apple" và "apple" có thể được coi là cùng một từ trong ngữ cảnh nhất định).
    - 'word[-3:]': word[-3:]: Ba ký tự cuối cùng của từ (hậu tố). Hậu tố có thể là một đặc trưng hữu ích để nhận dạng loại thực thể (ví dụ: các hậu tố thường xuất hiện trong tên người hoặc tổ chức).
    - 'word.isupper': word.isupper(): Giá trị Boolean cho biết liệu từ có được viết hoàn toàn bằng chữ hoa hay không. Đặc trưng này quan trọng vì các từ viết hoa toàn bộ thường là viết tắt của tổ chức (ví dụ: "UNESCO").
    - 'word.istitle': word.istitle(): Giá trị Boolean cho biết liệu từ có được viết theo dạng tiêu đề (chữ cái đầu tiên viết hoa, các chữ còn lại viết thường) hay không. Đây là một đặc trưng mạnh mẽ cho các danh từ riêng.
    - 'word.isdigit': word.isdigit(): Giá trị Boolean cho biết liệu từ có phải là một chữ số hay không. Hữu ích cho việc nhận dạng các thực thể liên quan đến số, ngày tháng.
- **Đặc trưng ngữ cảnh (từ liền kề):**
    - **Từ trước đó (if i > 0):**
        - '-1:word.lower': Dạng chữ thường của từ liền trước.
        - '-1:word.istitle': Dạng tiêu đề của từ liền trước.
        - Các đặc trưng này giúp mô hình hiểu ngữ cảnh cục bộ, ví dụ, nếu từ trước đó là "Ông", thì từ hiện tại có thể là tên người.
    - **Đánh dấu đầu câu (else cho if i > 0):**
        - 'BOS': True: Đặc trưng này được thêm vào nếu từ là từ đầu tiên của câu (Beginning Of Sentence). Nó cung cấp một tín hiệu mạnh mẽ cho mô hình về vị trí của từ trong chuỗi.
    - **Từ kế tiếp (if i < len(sent)-1):**
        - '+1:word.lower': Dạng chữ thường của từ liền sau.
        - '+1:word.istitle': Dạng tiêu đề của từ liền sau.
        - Tương tự như từ liền trước, các đặc trưng này giúp mô hình nắm bắt ngữ cảnh từ phía sau.
    - **Đánh dấu cuối câu (else cho if i < len(sent)-1):**

- 'EOS': True: Đặc trưng này được thêm vào nếu từ là từ cuối cùng của câu (End Of Sentence).

Việc kết hợp các đặc trưng về bản thân từ, hình thái học, và ngữ cảnh cục bộ cho phép CRF xây dựng một bức tranh toàn diện về vai trò của mỗi từ trong việc hình thành một thực thể có tên.

4.2.3. Huấn luyện mô hình CRF

Sau khi dữ liệu được chuẩn bị và đặc trưng được trích xuất, mô hình CRF được khởi tạo và huấn luyện.

Khởi tạo mô hình:

```
crf = CRF(algorithm='lbfgs',max_iterations=100,all_possible_transitions=True)
```

- algorithm='lbfgs': Chỉ định thuật toán tối ưu hóa "Limited-memory Broyden–Fletcher–Goldfarb–Shanno" (L-BFGS). Đây là một thuật toán phổ biến và hiệu quả để huấn luyện các mô hình CRF, được biết đến với khả năng xử lý các hàm mục tiêu phức tạp trên không gian đặc trưng lớn.
- max_iterations=100: Đặt số lần lặp tối đa cho thuật toán tối ưu hóa. Mô hình sẽ dừng huấn luyện nếu đạt đến số lần lặp này hoặc nếu hội tụ trước đó.
- all_possible_transitions=True: Tham số này cho phép mô hình học các xác suất chuyển đổi giữa tất cả các cặp nhãn có thể có. Điều này giúp mô hình linh hoạt hơn trong việc biểu diễn các chuỗi nhãn phức tạp và các phụ thuộc giữa các nhãn liền kề.

4.3. Hidden Markov Model

Đối với Mô hình Markov Ẩn (HMM), quá trình chuẩn bị dữ liệu đầu vào train_sents tương tự như đối với CRF, tức là một danh sách các câu, mỗi câu là một danh sách các cặp (từ, nhãn). Tuy nhiên, HMM không yêu cầu bước trích xuất đặc trưng tường minh như CRF, vì HMM học trực tiếp các xác suất chuyển trạng thái và xác suất phát xạ từ các cặp từ-nhãn này.

Huấn luyện mô hình:

```
tagger = HiddenMarkovModelTrainer().train_supervised(
    train_sents, estimator=lambda fd, bins: LidstoneProbDist(fd, 0.1, bins)
)
```

- *HiddenMarkovModelTrainer().train_supervised()*: Đây là phương thức được sử dụng để huấn luyện HMM một cách có giám sát. Nó học các tham số của mô

hình (xác suất chuyển trạng thái và xác suất phát xạ) trực tiếp từ dữ liệu huấn luyện đã được gán nhãn.

- *train_sents*: Dữ liệu huấn luyện được cung cấp cho mô hình, dưới dạng các chuỗi từ đã được gán nhãn.
- *estimator=lambda fd, bins: LidstoneProbDist(fd, 0.1, bins)*: Tham số này chỉ định phương pháp ước lượng xác suất được sử dụng. Trong trường hợp này, LidstoneProbDist được sử dụng, đây là một kỹ thuật làm mịn (smoothing) **Lidstone** với tham số **gamma là 0.1**. Kỹ thuật làm mịn này rất quan trọng để tránh xác suất bằng 0 cho các sự kiện không được nhìn thấy trong dữ liệu huấn luyện, giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới.
- Quá trình huấn luyện này sẽ tính toán các ma trận xác suất chuyển trạng thái (từ nhãn này sang nhãn khác) và xác suất phát xạ (từ một nhãn tạo ra một từ cụ thể) dựa trên tần suất xuất hiện trong train_sents.

4.4. Kết quả thử nghiệm

4.4.1. Metrics đánh giá

Trong các bài toán phân loại, đặc biệt là phân loại trên dữ liệu mất cân bằng lớp như NER (nơi nhãn "O" - Outside thường chiếm đa số), việc chỉ sử dụng **Accuracy** có thể không đủ để phản ánh đúng hiệu suất. Do đó, chúng em cũng sử dụng **F1-score**, một chỉ số cân bằng hơn và confusion matrix.

Để hiểu rõ hơn về các chỉ số, chúng em cần định nghĩa các khái niệm cơ bản:

- **True Positive (TP)**: Số lượng các trường hợp mô hình dự đoán là tích cực và thực tế cũng là tích cực.
- **True Negative (TN)**: Số lượng các trường hợp mô hình dự đoán là tiêu cực và thực tế cũng là tiêu cực .
- **False Positive (FP)**: Số lượng các trường hợp mô hình dự đoán là tích cực nhưng thực tế là tiêu cực.
- **False Negative (FN)**: Số lượng các trường hợp mô hình dự đoán là tiêu cực nhưng thực tế là tích cực.

Accuracy là chỉ số đơn giản nhất, đo lường tỷ lệ các dự đoán đúng trên tổng số các dự đoán.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

- **Ý nghĩa**: Accuracy cho biết tổng thể mô hình đã dự đoán đúng bao nhiêu phần trăm.
- **Hạn chế trong NER**: Mặc dù dễ hiểu, Accuracy có thể gây hiểu lầm trong các bài toán NER. Do nhãn "O" (Outside) thường chiếm phần lớn số lượng token, một mô hình có thể đạt Accuracy cao chỉ bằng cách dự đoán hầu hết các token là "O", trong khi bỏ lỡ nhiều thực thể quan trọng. Do đó, Accuracy không phải là chỉ số lý tưởng duy nhất cho NER khi dữ liệu bị mất cân bằng lớp.

F1-score là một chỉ số cân bằng, đặc biệt hữu ích khi có sự mất cân bằng giữa các lớp hoặc khi cả Precision và Recall đều quan trọng.

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

- **Ý nghĩa**: F1-score cung cấp một thước đo tổng thể về hiệu suất của mô hình, xem xét cả khả năng chính xác (ít lỗi FP) và khả năng bao phủ (ít lỗi FN). Một F1-score cao cho thấy mô hình vừa có khả năng dự đoán đúng các thực thể, vừa có khả năng tìm ra hầu hết các thực thể trong dữ liệu.
- **Tầm quan trọng trong NER**: Đối với NER, F1-score thường được ưu tiên hơn Accuracy. Nó giúp đánh giá hiệu quả của mô hình trong việc xác định các thực thể có tên một cách chính xác mà không bị đánh lừa bởi số lượng lớn các nhãn "O".

Confusion matrix là một bảng tóm tắt hiệu suất của một thuật toán phân loại trên một tập dữ liệu thử nghiệm. Nó cho phép phân tích chi tiết hơn các loại lỗi mà mô hình mắc phải.

- **Cấu trúc**:
    - Ma trận có các hàng đại diện cho các lớp thực tế (Actual Classes).
    - Các cột đại diện cho các lớp được mô hình dự đoán (Predicted Classes).
    - Đối với bài toán phân loại nhị phân, ma trận có dạng 2x2. Đối với phân loại đa lớp (như NER với nhiều loại thực thể), nó sẽ là NxN, trong đó N là số lượng các lớp.
- **Giải thích các ô**:
    - Ô trên đường chéo chính (từ trên cùng bên trái đến dưới cùng bên phải) thể hiện số lượng các dự đoán đúng:
        - Ô (Actual: Lớp A, Predicted: Lớp A) là số lượng dự đoán đúng cho Lớp A (True Positive cho Lớp A khi xem xét Lớp A là tích cực).
    - Các ô ngoài đường chéo chính thể hiện số lượng các dự đoán sai (lỗi):

- Ô (Actual: Lớp A, Predicted: Lớp B) là số lượng các trường hợp thực tế là Lớp A nhưng mô hình dự đoán nhầm thành Lớp B. Đây là các lỗi False Negative cho Lớp A và False Positive cho Lớp B.

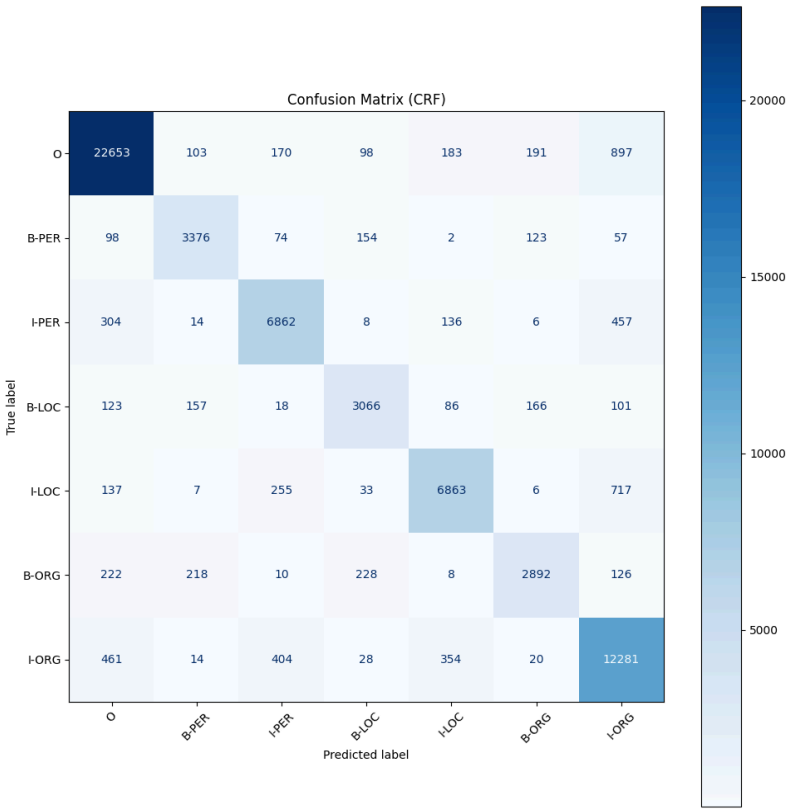
Ý nghĩa: Ma trận nhầm lẫn cung cấp cái nhìn trực quan và chi tiết về các loại lỗi. Từ ma trận này, chúng ta có thể dễ dàng tính toán Precision và F1-score cho từng lớp, giúp xác định lớp nào mô hình hoạt động tốt và lớp nào cần cải thiện. Điều này đặc biệt hữu ích trong NER để phân tích các lỗi chuyển đổi giữa các loại thực thể hoặc giữa thực thể và nhãn "O".

4.4.2. Phân tích kết quả đạt được

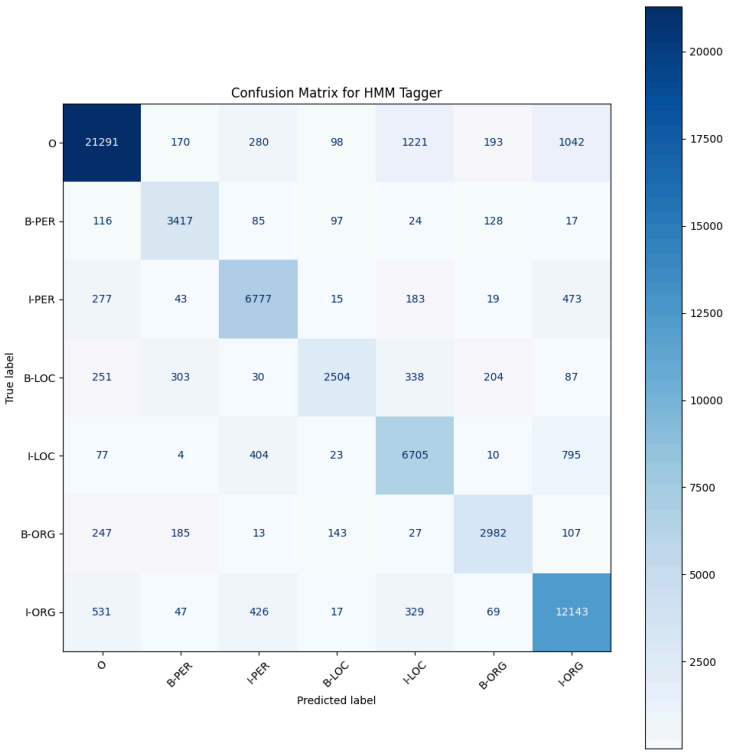
Kết quả sau khi train và được test trên tập test của 3 phương pháp CRF, HMM và PhoBERT

	Accuracy	F1-score
CRF	0.89	0.87
HMM	0.86	0.83
PhoBERT	**0.96**	**0.91**

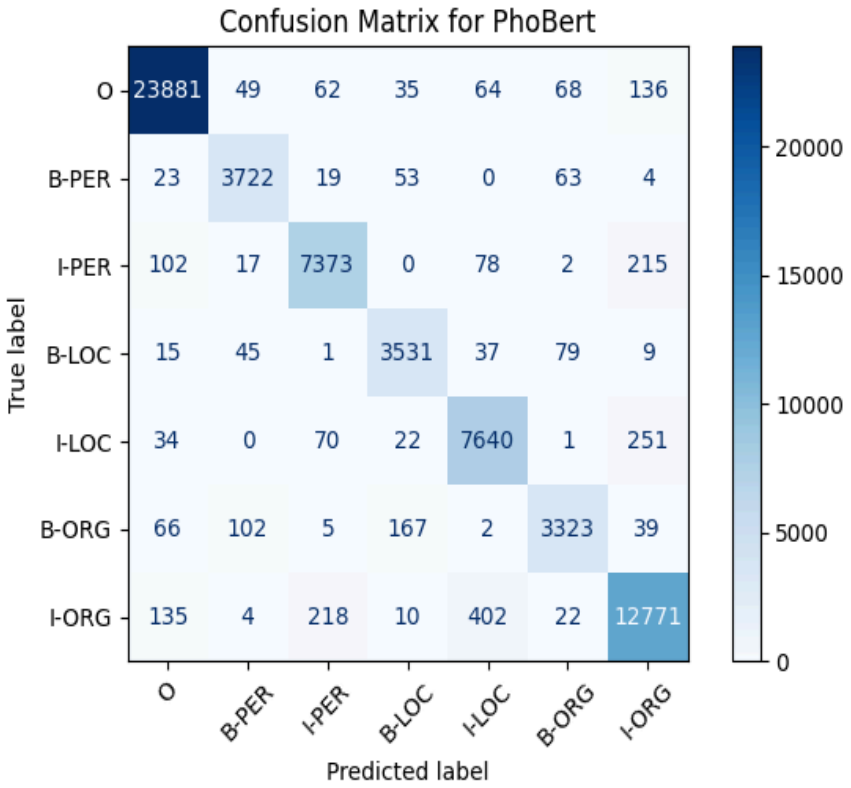
Bảng 6. Kết quả accuracy và f1-score cho 3 model



Hình 3. Confusion matrix (CRF)



Hình 4. Confusion matrix (HMM)



Hình 5. Confusion matrix (PhoBERT)

Bảng phân tích một số mẫu mà mô hình **PhoBERT** dự đoán sai:

Tokens	**Nhãn của tập test**	**Mô hình dự đoán**	**Phân tích**
["""", """", 'Rolling', 'Stone', """", """", """"]	O, O, **B-ORG, I-ORG**, O, O, O	O, O, **B-PER, I-PER,** O, O, O	Trường hợp này "Rolling Stone" là một ban nhạc, nên nhãn của tập test là đúng, nhưng mô hình dự đoán sai. Nguyên do vì mẫu này chỉ có tên của ban nhạc, không bao gồm ngữ cảnh nên mô hình khó nhận biết được đây là người hay tổ chức, thêm vào đó PhoBERT được tối ưu trên tiếng Việt, trong khi đây là một tên trong Tiếng Anh. Nếu chưa thấy qua thì con người cũng khó đoán được.
['Lời', 'phát', 'biểu', '``', 'Tôi', """]	B-ORG, I-ORG, I-ORG, I-ORG, I-ORG, I-ORG	**O, O, O, O, O, O**	Mô hình dự đoán đúng với thực tế, vì đây chỉ là một khái niệm trong giao tiếp. Trường hợp này sai là do nhãn của tập test
['Vườn', 'quốc', 'gia', 'Quần', 'đảo',	B-ORG, I-ORG, I-ORG,	**B-LOC, I-LOC, I-LOC, I-LOC,**	Mô hình dự đoán đúng với thực tế, vì đây chỉ là một địa điểm trên bản đồ.

'Haparanda']	I-ORG, I-ORG, I-ORG	**I-LOC, I-LOC**	Trường hợp này sai là do nhãn của tập test
['Vũ', 'Thị', 'Hương', """, '(', 'điền', 'kinh', ')', '(', '100m', 'nữ', ')', """]	B-PER, I-PER, I-PER, O, O, B-ORG, I-ORG, O, O, O, O, O, O	B-PER, I-PER, I-PER, O, O, **O, O**, O, O, O, O, O, O	Nhãn của tập test đúng khi xác định tên người, nhưng lại sai khi xác định cụm "điền kinh" là một tổ chức, trong khi mô hình không mắc lỗi này. Chứng tỏ việc học từ các mẫu được gán nhãn đúng của mô hình khá tốt.
['===', 'Các', 'vua', 'của', 'Śailendra', '===']	O, O, O, O, B-ORG, O	O, O, O, O, **B-LOC**, O	Mô hình coi đây là một địa điểm, trong khi nhãn của tập test coi đó là một tổ chức. Thực tế, 'Śailendra' là tên một triều đại trong lịch sử của Indonesia, nên sẽ phù hợp hơn với nhãn B-ORG. Ngữ cảnh của câu này dễ khiến mô hình lầm tưởng 'Śailendra' là tên một quốc gia hay vùng lãnh thổ nên đã gán là B-LOC.
Phó Tổng lý (Phó Thủ tướng)	B-ORG, I-ORG, I-ORG, O, O, O, O, O	B-ORG, I-ORG, I-ORG, O, **B-ORG, I-PER, I-ORG, I-PER**	Đây là tên một chức vụ trong chính phủ Trung Quốc. Mặc dù 3 tokens đầu mô hình dự đoán đúng với nhãn, nhưng lại sai với thực tế vì tên chức vụ không phải là tên tổ chức. Những token cuối model bị hỗn loạn, những nhãn I-PER lại đan xen vào cụm B-ORG và I-ORG, điều này có thể do dữ liệu train bị sai, khiến mô hình học chưa được tốt cấu trúc của các nhãn theo chuẩn IOB2
['Trình', 'quản', 'lý', 'tải', 'xuống']	B-PER, I-PER, I-PER, I-PER, I-PER	**B-ORG, I-ORG, I-ORG, I-ORG, I-ORG**	Trường hợp này model khác hoàn toàn so với nhãn, nhưng thực tế thì cả 2 đều sai, do đây là tên chung của một nhóm công cụ, chứ không phải tên riêng của người hay tổ chức.
['Lê', 'dương', 'Pháp']	B-ORG, I-ORG, I-ORG	**B-LOC, I-LOC, I-LOC**	Mẫu này là tên của một tổ chức quân đội, nên nhãn của tập test là đúng. Nhưng model lại dự đoán là địa điểm. Có thể do từ cuối là tên một đất nước, trong khi từ "Lê" ở đầu cũng giống như một tên riêng của địa điểm, khiến model nghĩ rằng đây là một địa danh giống như "Thủ Đức Việt Nam", nên coi cả cụm là một LOC
[""", """, 'River', 'Plate', """, """]	O, O, B-ORG, I-ORG, O, O	O, O, **B-PER, I-PER**, O, O	Với mẫu này, đây là một câu lạc bộ bóng đá, nên nhãn của test là đúng.

			Mô hình bị sai do mẫu chỉ gồm tên riêng, không có ngữ cảnh cụ thể, nên dễ bị nhầm lẫn.

Xét riêng hiệu năng của model PhoBERT từ bảng trên, ta có thể thấy đa phần những lần bị đánh giá là sai đều do mô hình dự đoán đúng với thực tế trong khi nhãn của tập test lại sai. Điều đó không phải lỗi của mô hình, mà còn chứng tỏ khả năng học tốt từ các mẫu được gán nhãn đúng trong tập train của PhoBERT. Bên cạnh đó, cũng có một số trường hợp mà nhãn của tập test đúng với thực tế, thì dữ liệu thường mơ hồ về ngữ cảnh, khiến model bị nhầm lẫn. Những mẫu như vậy mang tính thử thách cao, đến cả người nếu chưa từng nghe qua thì cũng không phân biệt được, nên không thể trách được model. Hoặc các trường hợp hiếm hơn, khi mà cả nhãn và dự đoán đều sai với thực tế, phản ánh rằng mô hình còn có những sai sót, không thể đúng hoàn toàn 100%.

So sánh với 2 mô hình còn lại về kết quả đánh giá trên tập test, có thể thấy rõ ràng rằng:

- **Mô hình PhoBERT** đạt hiệu suất cao nhất với Accuracy 0.96 và F1-score 0.91, chứng tỏ khả năng vượt trội trong việc nhận dạng thực thể.
- **Mô hình CRF** thể hiện hiệu suất tốt hơn đáng kể so với HMM, với Accuracy 0.89 và F1-score 0.87.
- **Mô hình HMM** có hiệu suất thấp nhất trong ba mô hình, với Accuracy 0.86 và F1-score 0.83.

Sự chênh lệch về F1-score đặc biệt quan trọng trong bài toán NER do tính chất mất cân bằng lớp (nhãn 'O' - Outside thường chiếm số lượng lớn), phản ánh khả năng nhận diện các thực thể có tên một cách chính xác và đầy đủ.

Khi xét sâu hơn về kết quả đánh giá trên các loại thực thể khác nhau, ta nhận thấy mô hình PhoBERT thể hiện hiệu suất vượt trội và nhất quán trên tất cả các loại thực thể (PER, ORG, LOC). Cụ thể, PhoBERT đạt F1-score rất cao cho PER và LOC, cho thấy khả năng tìm ra hầu hết các thực thể này mà không bỏ sót. Đối với ORG, PhoBERT cũng duy trì F1-score mạnh mẽ. Điều này là nhờ khả năng hiểu biết ngữ cảnh toàn diện và sâu sắc của PhoBERT, được thừa hưởng từ kiến trúc Transformer và quá trình huấn luyện trước trên lượng dữ liệu tiếng Việt khổng lồ.

Trong khi đó, **CRF** mặc dù tốt hơn HMM, nhưng vẫn có những hạn chế nhất định. CRF cho F1-score tốt trên các thực thể như PER, nhưng lại cho thấy **thấp hơn ở các nhãn bắt đầu thực thể như B-ORG và B-LOC**, cho thấy mô hình này có xu hướng bỏ sót một số thực thể tại điểm khởi đầu.

Mặt khác, **HMM** thể hiện hiệu suất thấp nhất. Mô hình này gặp khó khăn trong việc nhận diện chính xác điểm khởi đầu của các thực thể địa điểm. Sự hạn chế của HMM xuất phát từ giả định Markov bậc nhất và việc chuẩn hóa cục bộ, khiến nó không thể nắm bắt được các phụ thuộc phức tạp và ngữ cảnh rộng như CRF hay phoBERT.

Nhìn chung, sự chênh lệch rõ rệt về hiệu suất giữa các mô hình phản ánh khả năng xử lý ngữ nghĩa và ngữ cảnh của chúng. Các mô hình học sâu như PhoBERT, với kiến trúc phức tạp và quá trình huấn luyện trước trên lượng dữ liệu lớn, đã chứng minh được tính hiệu quả vượt trội trong việc giải quyết bài toán NER cho tiếng Việt so với các phương pháp học máy truyền thống.

4.5. Xử lí data bị gán nhãn sai

Như chúng em đã trình bày ở phần 2, có một số mẫu data bị sai, và khi phân tích, nhóm thấy thực tế dữ liệu có một số lượng không nhỏ lỗi gán nhãn.

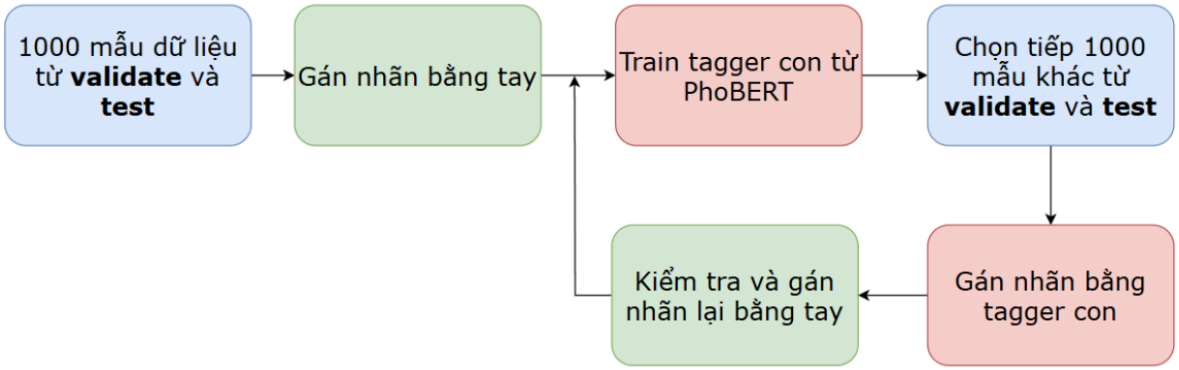
Điều này làm model học sai và có kết quả không tốt khi thực nghiệm với những câu văn ở ngoài dữ liệu mặc dù điểm số của model rất cao trên tập test.

```
Tổng        B-ORG
Bí          I-ORG
thư         I-ORG
Tô          I-ORG
Lâm         I-ORG
đã          O
có          O
bài         O
phát        O
biểu        O
quan        O
trọng       O
.           O
====================
Con         B-ORG
đường       I-ORG
tơ          I-ORG
lụa         I-ORG
====================
```

Hình 4: Minh họa 2 ví dụ mà mô hình dự đoán sai

Hướng xử lý và Phương pháp tinh chỉnh dữ liệu/mô hình

Để nâng cao hiệu suất của hệ thống Nhận dạng thực thể có tên (NER), nhóm đã tập trung vào hai hướng xử lý chính: cải thiện chất lượng dữ liệu thông qua gán nhãn thủ công và tiếp tục tinh chỉnh (fine-tune) mô hình ngôn ngữ PhoBERT đã được huấn luyện trước đó.



Gán nhãn lại dữ liệu bằng tay

Nhận thấy tầm quan trọng của chất lượng dữ liệu đối với hiệu suất của các mô hình học máy, đặc biệt là trong các tác vụ ngữ nghĩa phức tạp như NER, nhóm đã thực hiện một quá trình gán nhãn lại dữ liệu bằng tay. Quá trình này bao gồm việc các thành viên trong nhóm trực tiếp rà soát, kiểm tra và điều chỉnh các nhãn thực thể đã có, hoặc gán nhãn mới cho các đoạn văn bản chưa được gán nhãn một cách chính xác.

Mục tiêu của việc gán nhãn thủ công là:

- **Nâng cao độ chính xác của nhãn**: Khắc phục các lỗi hoặc sự không nhất quán trong bộ dữ liệu gốc, vốn có thể ảnh hưởng tiêu cực đến quá trình huấn luyện mô hình.
- **Xử lý các trường hợp phức tạp và mơ hồ**: Ngôn ngữ tự nhiên thường chứa đựng nhiều sự mơ hồ và các trường hợp đặc biệt mà chỉ con người mới có thể đưa ra quyết định gán nhãn chính xác.
- **Thích nghi với miền dữ liệu cụ thể**: Đảm bảo rằng các nhãn phản ánh đúng ngữ nghĩa của thực thể trong ngữ cảnh miền dữ liệu mà nhóm đang làm việc, nếu có.

Tính đến thời điểm hiện tại, nhóm đã hoàn thành việc gán nhãn lại cho khoảng **2000 mẫu dữ liệu**. Số lượng mẫu này được thu thập từ hai nguồn chính:

- 1000 mẫu từ tập validation
- 1000 mẫu từ tập test

Quá trình gán nhãn thủ công, mặc dù tốn thời gian và nguồn lực, nhưng là một khoản đầu tư quan trọng để đảm bảo tính toàn vẹn và độ tin cậy của bộ dữ liệu huấn luyện, từ đó cải thiện đáng kể chất lượng đầu ra của mô hình.

Tinh chỉnh (Fine-tune) mô hình PhoBERT đã được huấn luyện trước đó

Song song với việc cải thiện chất lượng dữ liệu, nhóm đã tiếp tục tinh chỉnh (fine-tune) mô hình PhoBERT.

Quá trình tinh chỉnh được thực hiện bằng cách huấn luyện mô hình PhoBERT trên tập dữ liệu đã được gán nhãn lại bằng tay. Mục tiêu là để mô hình có thể học và thích nghi với các đặc điểm cụ thể của dữ liệu đã được làm sạch và mở rộng, từ đó cải thiện khả năng nhận diện các thực thể có tên một cách chính xác và đầy đủ hơn. Các tham số huấn luyện (như tốc độ học, kích thước batch, số epoch) được điều chỉnh để tối ưu hóa quá trình học của mô hình trên bộ dữ liệu mới này.

Đánh giá

Do đã học trên dữ liệu mới được lấy từ tập test và tập validate, nên việc đánh giá điểm accuracy hay F1 của mô hình finetuned trên hai tập dữ liệu này là vô nghĩa. Vậy nên nhóm sẽ đánh giá chung thông qua một số câu văn bên ngoài thay vì những câu trong test hay validate. Kết quả của 2 mô hình sẽ được so sanh trong bảng dưới đây. Để thuận tiện trong việc so sánh, mỗi mẫu sẽ được biểu diễn dưới dạng các từ riêng biệt, tương ứng với 2 hàng nhãn, hàng trên là của PhoBERT, còn hàng dưới là của mô hình Finetuned.

Tokens	Nhãn	Chú thích
['Thành', 'phố', 'Hồ', 'Chí', 'Minh', 'là', 'trung', 'tâm', 'kinh', 'tế', 'lớn', 'nhất', 'Việt', 'Nam', '.']	[5, 6, 6, 6, 6, 0, 0, 6, 6, 6, 0, 0, 5, 6, 0]	PhoBERT dự đoán sai cho 3 tokens
	[5, 6, 6, 6, 6, 0, 0, **0, 0, 0**, 0, 0, 5, 6, 0]	Mô hình Finetuned không bị nhầm
['Tiến', 'sĩ', 'Lê', 'Thị', 'Thu', 'Hà', 'đang', 'nghiên', 'cứu', 'về', 'trí', 'tuệ', 'nhân', 'tạo', '.']	[0, 0, 1, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]	Đúng
	[0, 0, 1, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]	Đúng
['Tập', 'đoàn', 'FPT', 'Software', 'là', 'một', 'trong', 'những', 'công', 'ty', 'công', 'nghệ', 'hàng', 'đầu', 'Việt', 'Nam', '.']	[0, 0, 3, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 6, 0]	Không coi "Tập đoàn" là tên của một tổ chức
	[**3, 4**, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 6, 0]	Finetuned coi "Tập đoàn" là tên của tổ

		chức.
['Vịnh', 'Hạ', 'Long', 'được', 'UNESCO', 'công', 'nhận', 'là', 'Di', 'sản', 'Thiên', 'nhiên', 'Thế', 'giới', '.']	[5, 6, 6, 0, 3, 0, 0, 0, 3, 4, 4, 4, 4, 4, 0]	PhoBERT bị nhầm một giải thưởng thành một tổ chức
	[5, 6, 6, 0, 3, 0, 0, 0, **0, 0, 0, 0, 0, 0, 0**]	Finetuned đã đúng, không còn bị nhầm
['Công', 'ty', 'cổ', 'phần', 'Sữa', 'Việt', 'Nam', 'Vinamilk', 'là', 'nhà', 'sản', 'xuất', 'sữa', 'hàng', 'đầu', 'Việt', 'Nam', '.']	[0, 0, 4, 4, 4, 4, 4, 3, 0, 0, 0, 0, 0, 0, 0, 5, 6, 0]	Xác định được tổ chức nhưng lại không có token B-ORG ở đầu, mà lại ở cuối cụm
	[**3, 4**, 4, 4, 4, 4, 4, **4**, 0, 0, 0, 0, 0, 0, 0, 0, 5, 6, 0]	Đã xác định đúng tổ chức và đặt đúng token B-ORG ở đầu
['Sông', 'Hồng', 'chảy', 'qua', 'thủ', 'đô', 'Hà', 'Nội', 'của', 'Việt', 'Nam', '.']	[5, 6, 0, 0, 0, 0, 5, 6, 0, 5, 6, 0]	Đúng
	[5, 6, 0, 0, 0, 0, 5, 6, 0, 5, 6, 0]	Đúng
['Ông', 'Nguyễn', 'Thanh', 'Long', 'từng', 'là', 'Bộ', 'trưởng', 'Bộ', 'Y', 'tế', '.']	[0, 1, 2, 2, 0, 0, 0, 0, 3, 4, 4, 0]	Đúng
	[0, 1, 2, 2, 0, 0, 0, 0, 3, 4, 4, 0]	Đúng
['Cầu', 'Cần', 'Thơ', 'nối', 'liền', 'hai', 'bờ', 'sông', 'Hậu', 'thuộc', 'thành', 'phố', 'Cần', 'Thơ', '.']	[3, 4, 4, 0, 0, 0, 0, 6, 6, 0, 5, 6, 6, 6, 0]	PhoBERT nhầm LOC thành ORG cho "Cầu Cần Thơ" và không có token B-LOC cho cụm "sông Hậu"
	[**5, 6, 6**, 0, 0, 0, 0, **6, 6**, 0, 5, 6, 6, 6, 0]	Mặc dù không nhầm LOC thành ORG cho "Cầu Cần Thơ" nhưng vẫn chưa có token B-LOC cho "sông Hậu"
['Tổng', 'Bí', 'thư', 'Tô', 'Lâm', 'đã', 'có', 'bài', 'phát', 'biểu', 'quan', 'trọng', '.']	[3, 4, 4, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0]	Nhầm tên người thành tên tổ chức
	[**1, 2, 2, 2, 2,** 0, 0, 0, 0, 0, 0, 0, 0]	Đúng
['Liên', 'Hợp', 'Quốc', 'có', 'trụ', 'sở', 'chính', 'tại', 'New',	[3, 4, 4, 0, 0, 0, 0, 0, 5, 6, 0, 5, 6, 0]	Đúng

'York', ',', 'Hoa', 'Kỳ', '.']	[3, **6**, 4, 0, 0, 0, 0, 0, 5, 6, 0, 5, 6, 0]	Mô hình Finetuned dự đoán sai cho token "Hợp"
['đổi', 'Fukui', '(', 'thành', 'phố', ')']	[0, 5, 6, 6, 6, 6]	Là một mẫu trong tập train, PhoBERT dự đoán sai đúng như được học từ tập train
	[0, 5, **0, 0, 0, 0**]	Finetuned đã phân biệt đúng, mặc dù không có mẫu này trong tập dữ liệu dùng để finetune
['Con', 'đường', 'tơ', 'lụa']	[3, 4, 4, 4]	Là một mẫu trong tập test, PhoBERT dự đoán sai nhưng đúng với nhãn từ tập test cũ
	[**5, 6, 6, 6**]	Finetuned đã phân biệt đúng
['đổi', 'Chỉ', 'thị', 'về', 'hạn', 'chế', 'các', 'chất', 'nguy', 'hiểm']	[0, 3, 4, 4, 4, 4, 4, 4, 4, 4]	Là một mẫu trong tập test, PhoBERT dự đoán sai nhưng đúng với nhãn từ tập test cũ
	[0, **0, 0, 0, 0, 0, 0, 0, 0**]	Finetuned đã phân biệt đúng
['Tổng', 'thống', 'Hoa', 'Kỳ']	[3, 4, 4, 4]	Là một mẫu trong tập validate, PhoBERT dự đoán sai nhưng đúng với nhãn từ tập validatecũ
	[**0, 0, 5, 6**]	Finetuned đã phân biệt đúng

So sánh bằng nhãn:

```
Tokens:    Thành  phố    Hồ     Chí    Minh   là     trung  tâm    kinh   tế     lớn    nhất   Việt   Nam    .
PhoBERT:   B-LOC  I-LOC  I-LOC  I-LOC  I-LOC  O      O      I-LOC  I-LOC  I-LOC  O      O      B-LOC  I-LOC  O
Finetuned: B-LOC  I-LOC  I-LOC  I-LOC  I-LOC  O      O      O      O      O      O      O      B-LOC  I-LOC  O
==========================================================================================================
Tokens:    Tiến   sĩ     Lê     Thị    Thu    Hà     đang   nghiên cứu    về     trí    tuệ    nhân   tạo    .
PhoBERT:   O      O      B-PER  I-PER  I-PER  I-PER  O      O      O      O      O      O      O      O      O
Finetuned: O      O      B-PER  I-PER  I-PER  I-PER  O      O      O      O      O      O      O      O      O
==========================================================================================================
Tokens:    Tập    đoàn   FPT    Software là     một    trong  những  công   ty     công    nghệ   hàng   đầu    Việt   Nam    .
PhoBERT:   O      O      B-ORG  I-ORG    O      O      O      O      O      O      O       O      O      O      B-LOC  I-LOC  O
Finetuned: B-ORG  I-ORG  I-ORG  I-ORG    O      O      O      O      O      O      O       O      O      O      B-LOC  I-LOC  O
==========================================================================================================
Tokens:    Vịnh   Hạ     Long   được   UNESCO công   nhận   là     Di     sản    Thiên  nhiên  Thế    giới   .
PhoBERT:   B-LOC  I-LOC  I-LOC  O      B-ORG  O      O      O      B-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  O
Finetuned: B-LOC  I-LOC  I-LOC  O      B-ORG  O      O      O      O      O      O      O      O      O      O
==========================================================================================================
Tokens:    Công   ty     cổ     phần   Sữa    Việt   Nam    Vinamilk là     nhà    sản    xuất   sữa    hàng   đầu    Việt   Nam    .
PhoBERT:   O      O      I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  B-ORG  O      O      O      O      O      O      O      B-LOC  I-LOC  O
Finetuned: B-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  O      O      O      O      O      O      O      B-LOC  I-LOC  O
==========================================================================================================
Tokens:    Sông   Hồng   chảy   qua    thủ    đô     Hà     Nội    của    Việt   Nam    .
PhoBERT:   B-LOC  I-LOC  O      O      O      O      B-LOC  I-LOC  O      B-LOC  I-LOC  O
Finetuned: B-LOC  I-LOC  O      O      O      O      B-LOC  I-LOC  O      B-LOC  I-LOC  O
==========================================================================================================
Tokens:    Ông    Nguyễn Thanh  Long   từng   là     Bộ     trưởng Bộ     Y      tế     .
PhoBERT:   O      B-PER  I-PER  I-PER  O      O      O      O      B-ORG  I-ORG  I-ORG  O
Finetuned: O      B-PER  I-PER  I-PER  O      O      O      O      B-ORG  I-ORG  I-ORG  O
==========================================================================================================
Tokens:    Cầu    Cần    Thơ    nối    liền   hai    bờ     sông   Hậu    thuộc  thành  phố   Cần    Thơ    .
PhoBERT:   B-ORG  I-ORG  I-ORG  O      O      O      O      I-LOC  I-LOC  O      B-LOC  I-LOC  I-LOC  I-LOC  O
Finetuned: B-LOC  I-LOC  I-LOC  O      O      O      O      I-LOC  I-LOC  O      B-LOC  I-LOC  I-LOC  I-LOC  O
==========================================================================================================
Tokens:    Tổng   Bí     thư    Tô     Lâm    đã     có     bài    phát   biểu   quan   trọng  .
PhoBERT:   B-ORG  I-ORG  I-ORG  I-ORG  I-ORG  O      O      O      O      O      O      O      O
Finetuned: B-PER  I-PER  I-PER  I-PER  I-PER  O      O      O      O      O      O      O      O
==========================================================================================================
Tokens:    Liên   Hợp    Quốc   có     trụ    sở     chính  tại    New    York   ,      Hoa    Kỳ     .
PhoBERT:   B-ORG  I-ORG  I-ORG  O      O      O      O      O      B-LOC  I-LOC  O      B-LOC  I-LOC  O
Finetuned: B-ORG  I-LOC  I-ORG  O      O      O      O      O      B-LOC  I-LOC  O      B-LOC  I-LOC  O
```

```
Tokens:    đổi    Fukui  (      thành  phố   )
PhoBERT:   O      B-LOC  I-LOC  I-LOC  I-LOC  I-LOC
Finetuned: O      B-LOC  O      O      O      O
==========================================================================================
Tokens:    Con    đường  tơ     lụa
PhoBERT:   B-ORG  I-ORG  I-ORG  I-ORG
Finetuned: B-LOC  I-LOC  I-LOC  I-LOC
==========================================================================================
Tokens:    đổi    Chỉ    thị    về     hạn    chế    các    chất   nguy   hiểm
PhoBERT:   O      B-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG  I-ORG
Finetuned: O      O      O      O      O      O      O      O      O      O
==========================================================================================
Tokens:    Tổng   thống  Hoa    Kỳ
PhoBERT:   B-ORG  I-ORG  I-ORG  I-ORG
Finetuned: O      O      B-LOC  I-LOC
==========================================================================================
```

Nhận xét:

- Mặc dù vẫn còn một số điểm chưa hoàn hảo, **mô hình đã tinh chỉnh (finetuned model)** của nhóm đã cho thấy sự cải thiện đáng kể. Cụ thể, mô hình này không chỉ **giữ vững độ chính xác** trong phần lớn các trường hợp mà mô hình **PhoBERT gốc (được huấn luyện trên tập train 20000 mẫu)** đã dự đoán đúng, mà còn **khắc phục được các lỗi** ở những mẫu mà PhoBERT gốc còn gặp khó khăn.

- Điều này là một minh chứng rõ ràng cho hiệu quả của việc fine-tuning. Kết quả đạt được cho thấy hướng đi này là đúng đắn và có tiềm năng lớn. Để nâng cao hơn nữa hiệu suất của mô hình, việc tăng cường lượng dữ liệu dùng cho quá trình fine-tuning là một bước đi đầy hứa hẹn, hứa hẹn sẽ mang lại những cải tiến vượt trội hơn nữa.

Chương 5 : KẾT LUẬN

Nghiên cứu này đã thành công trong việc triển khai và đánh giá ba phương pháp tiếp cận khác nhau để giải quyết bài toán Nhận dạng thực thể có tên (NER) cho văn bản tiếng Việt: HMM, CRF và PhoBERT..

Chúng em đã chuẩn bị dữ liệu một cách cẩn thận, bao gồm các bước xử lý riêng biệt phù hợp với yêu cầu của từng mô hình, từ việc trích xuất đặc trưng thủ công cho CRF đến tokenization và căn chỉnh nhãn phức tạp cho PhoBERT. Các mô hình đã được huấn luyện trên cùng một tập dữ liệu chuẩn hóa và được đánh giá bằng các chỉ số quan trọng như Độ chính xác (Accuracy) và F1-score, cùng với phân tích thông qua Ma trận nhầm lẫn (Confusion Matrix).

Kết quả thực nghiệm đã chứng minh rằng các mô hình học sâu, đặc biệt là PhoBERT, thường vượt trội hơn đáng kể so với các mô hình truyền thống (HMM, CRF) trong các tác vụ NLP phức tạp như NER. Sự vượt trội này đến từ khả năng của PhoBERT trong việc nắm bắt ngữ cảnh sâu và hai chiều của ngôn ngữ thông qua cơ chế tự chú ý và quá trình huấn luyện trước trên một lượng lớn dữ liệu. Các mô hình truyền thống, mặc dù có thể cung cấp hiệu suất cơ bản tốt, lại bị giới hạn bởi khả năng tích hợp đặc trưng thủ công và giả định về độc lập của dữ liệu.

Tài liệu tham khảo

- TechTarget: What Is Named Entity Recognition (NER) ?
- Datascience.stackexchange: Difference between IOB and IOB2 format.
- Cross-lingual Name Tagging and Linking for 282 Languages
- Slides môn học