



CS221

Named Entity Recognition

GVHD: Nguyễn Trọng Chính

Group: Destroyed by NLP

22520109 - Nguyễn Gia Bảo

22520069 - Phạm Nguyên Anh



-
-
- 1. Giới thiệu bài toán**
 - 2. Ngữ liệu**
 - 3. Phương pháp**
 - 4. Cài đặt thực nghiệm**
 - 5. Kết luận**
-
-



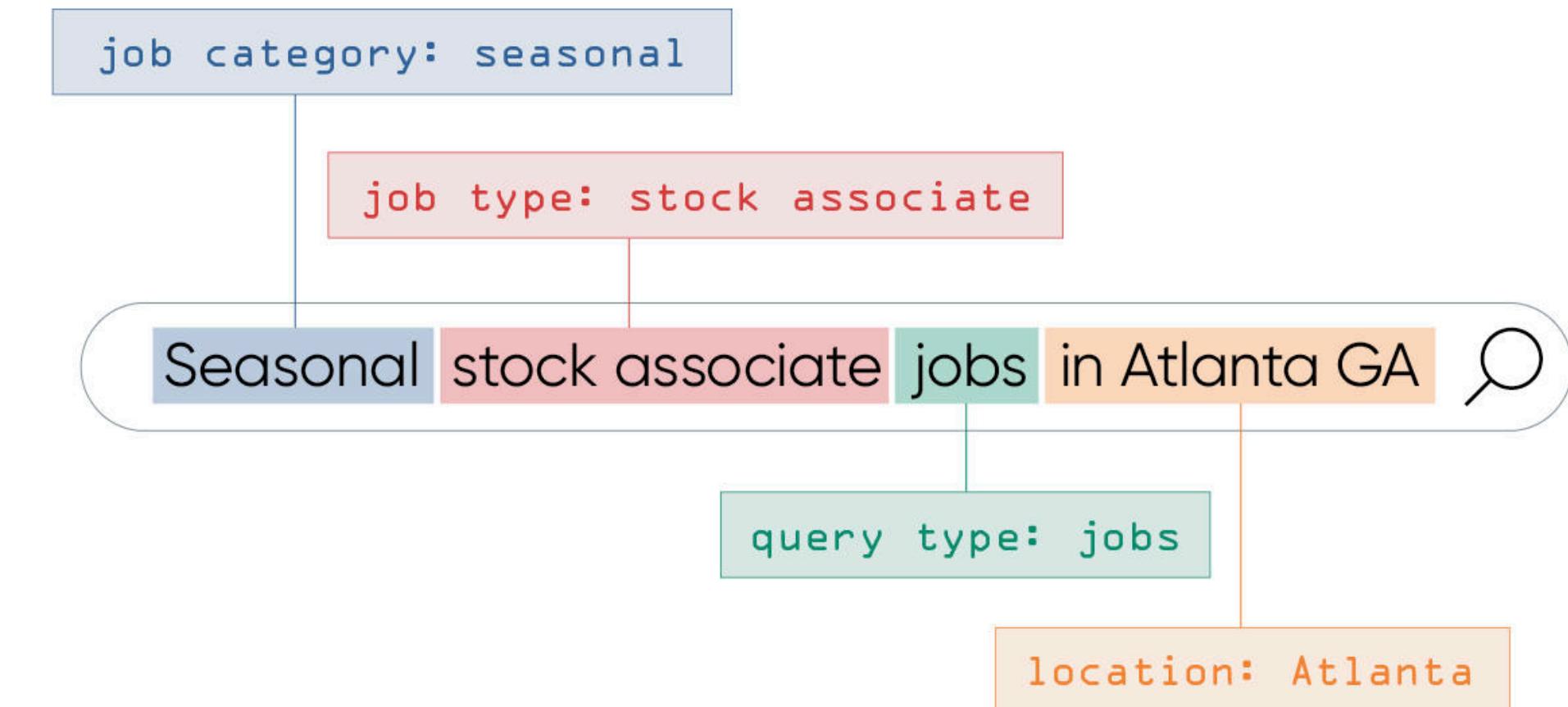
1.Giới thiệu bài toán



1. Giới thiệu

- NER (Named Entity Recognition)

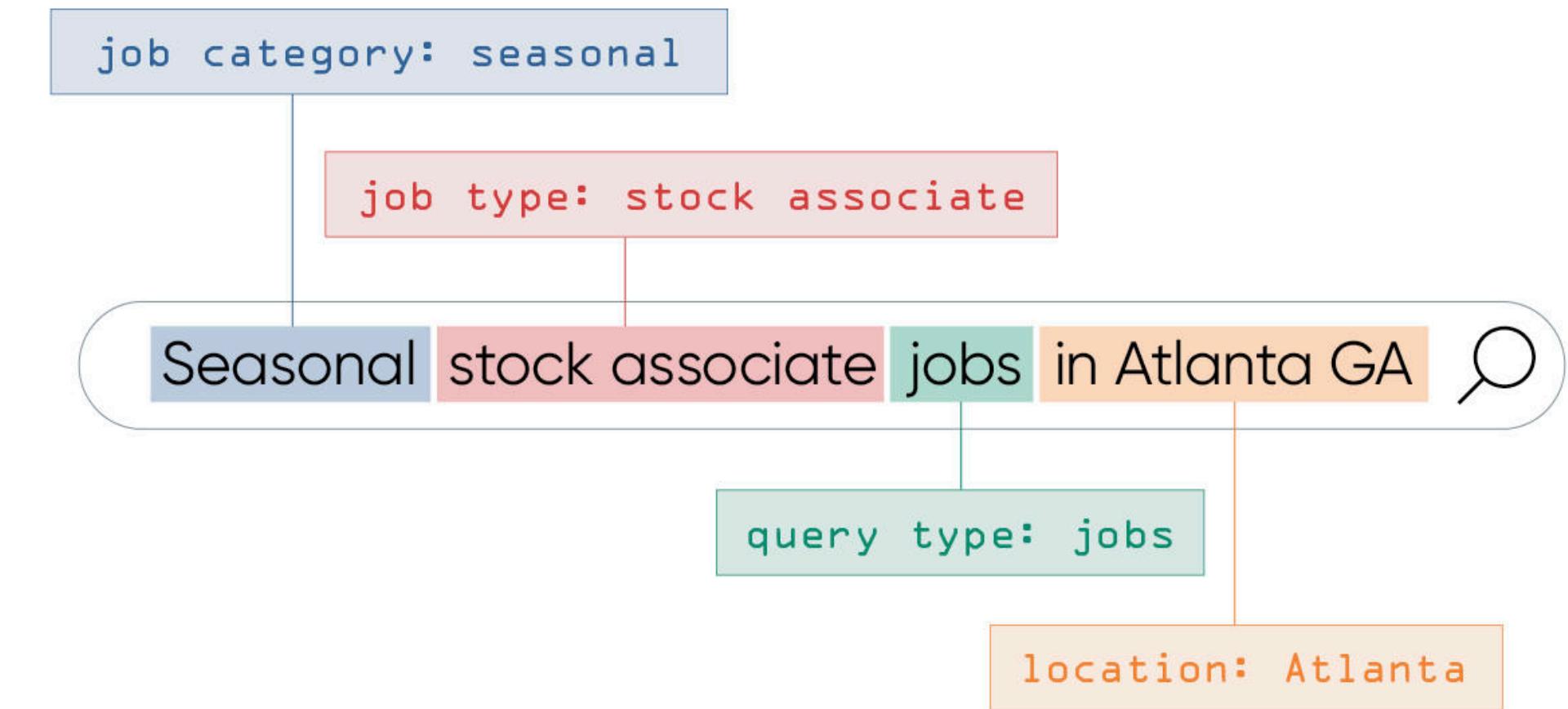
Recognition là một nhánh nhỏ của Xử lý ngôn ngữ tự nhiên, tập trung vào việc phát hiện và phân loại thực thể có tên trong văn bản vào những nhãn được định nghĩa trước.





1. Giới thiệu

- NER được dùng để trích xuất thông tin quan trọng từ văn bản không cấu trúc, giúp ích cho các hệ thống NLP như chatbots, phân tích cảm xúc, ...
- Bài toán mà nhóm thực hiện là huấn luyện một model nhận diện thực thể có tên trong câu văn tiếng Việt.





2.Ngữ liệu

2.1 Giới thiệu ngữ liệu

2.2 Quy tắc chú thích

2.3 Thống kê dữ liệu

2.4 Phân tích ngữ liệu



2.1 Giới thiệu ngữ liệu

- Bộ dữ liệu nhóm sử dụng là một phần từ tập dữ liệu lớn WikiANN (còn có tên khác là PAN-X) - một bộ dữ liệu Named Entity Recognition cho đa ngôn ngữ.
- Nội dung của của ngữ liệu này gồm các bài báo từ Wikipedia được gắn nhãn.
- Nhóm sẽ sử dụng phần dữ liệu chỉ gồm những câu Tiếng Việt.



2.1 Giới thiệu ngữ liệu

- Bộ dữ liệu nhóm sử dụng là một phần từ tập dữ liệu lớn WikiANN (còn có tên khác là PAN-X) - một bộ dữ liệu Named Entity Recognition cho đa ngôn ngữ.
- Nội dung của của ngữ liệu này gồm các bài báo từ Wikipedia được gắn nhãn.
- Nhóm sẽ sử dụng phần dữ liệu chỉ gồm những câu Tiếng Việt.
- Nguồn tải bộ dữ liệu: [HuggingFace](#) 😊



2.1 Cách gắn nhãn dữ liệu

Format nhãn: Bộ dữ liệu được gán nhãn theo chuẩn IOB2

Các loại nhãn chính:

- *PER*: person
- *LOC*: location
- *ORG*: organization



2.1 Cách gắn nhãn dữ liệu

Do được gắn theo chuẩn IOB2 nên các nhãn trên sẽ được tùy chỉnh thêm tiền tố **I** - **O** - **B**, cụ thể:

- **B-X**: Token là từ đầu tiên của một nhãn thuộc loại X
- **I-X**: Token nằm bên trong của nhãn X (nằm sau token **B-X**)
- **O**: Token không thuộc thực thể nào

Trong đó X là một trong các nhãn PER, LOC hoặc ORG. Một thực thể luôn bắt đầu bằng token dạng B-X.

Vậy chúng ta sẽ có tất cả các nhãn được đánh số lần lượt là:

0 (0), B-PER (1), I-PER (2), B-ORG (3), I-ORG (4), B-LOC (5), I-LOC (6)



2.1 Cách gắn nhãn dữ liệu

Ví dụ:

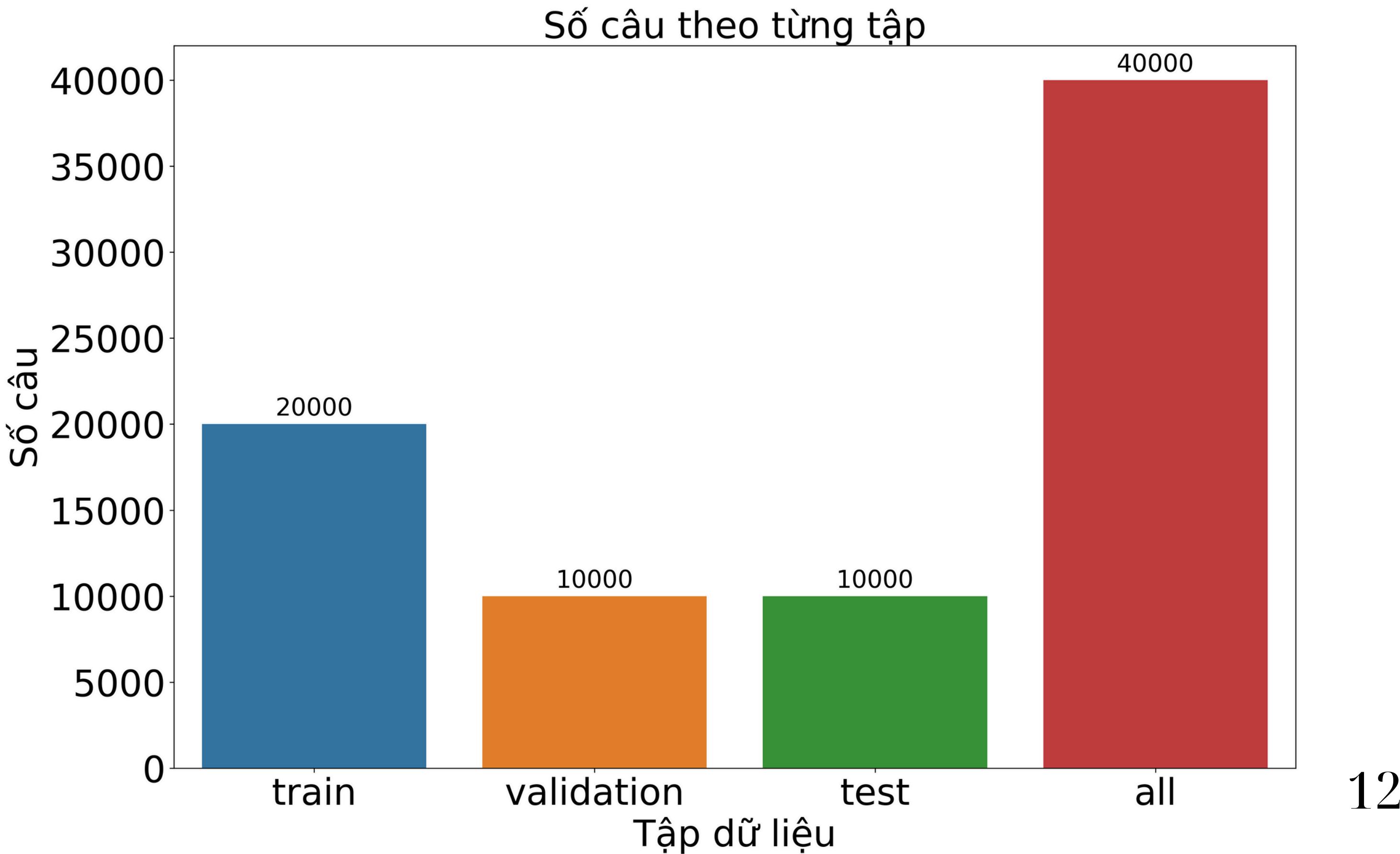
Nguyễn	B-PER
Anh	I-PER
đang	0
học	0
ở	0
UIT	B-ORG





2.3 Thống kê dữ liệu

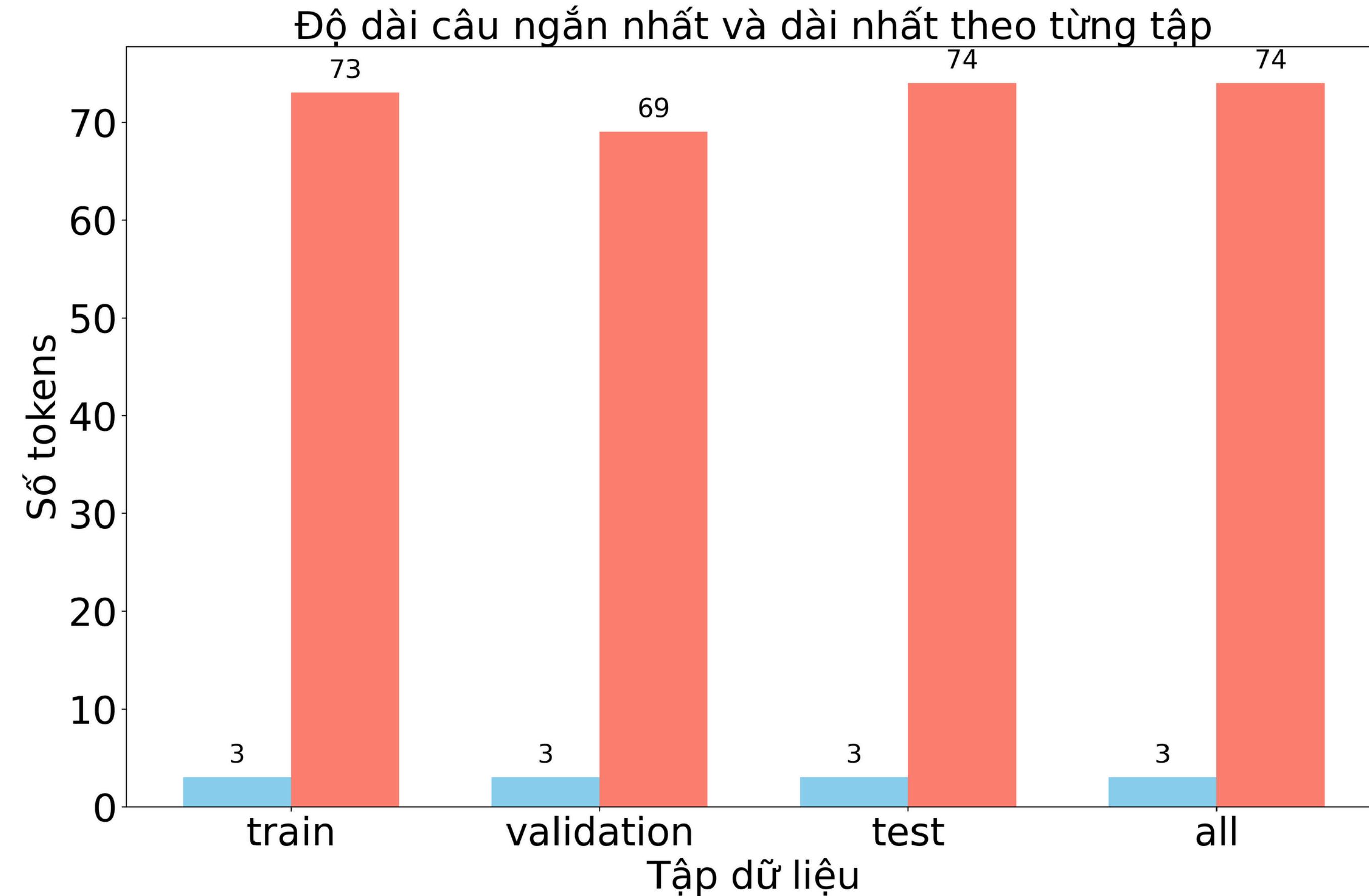
Số câu
trong
từng
tập





2.3 Thống kê dữ liệu

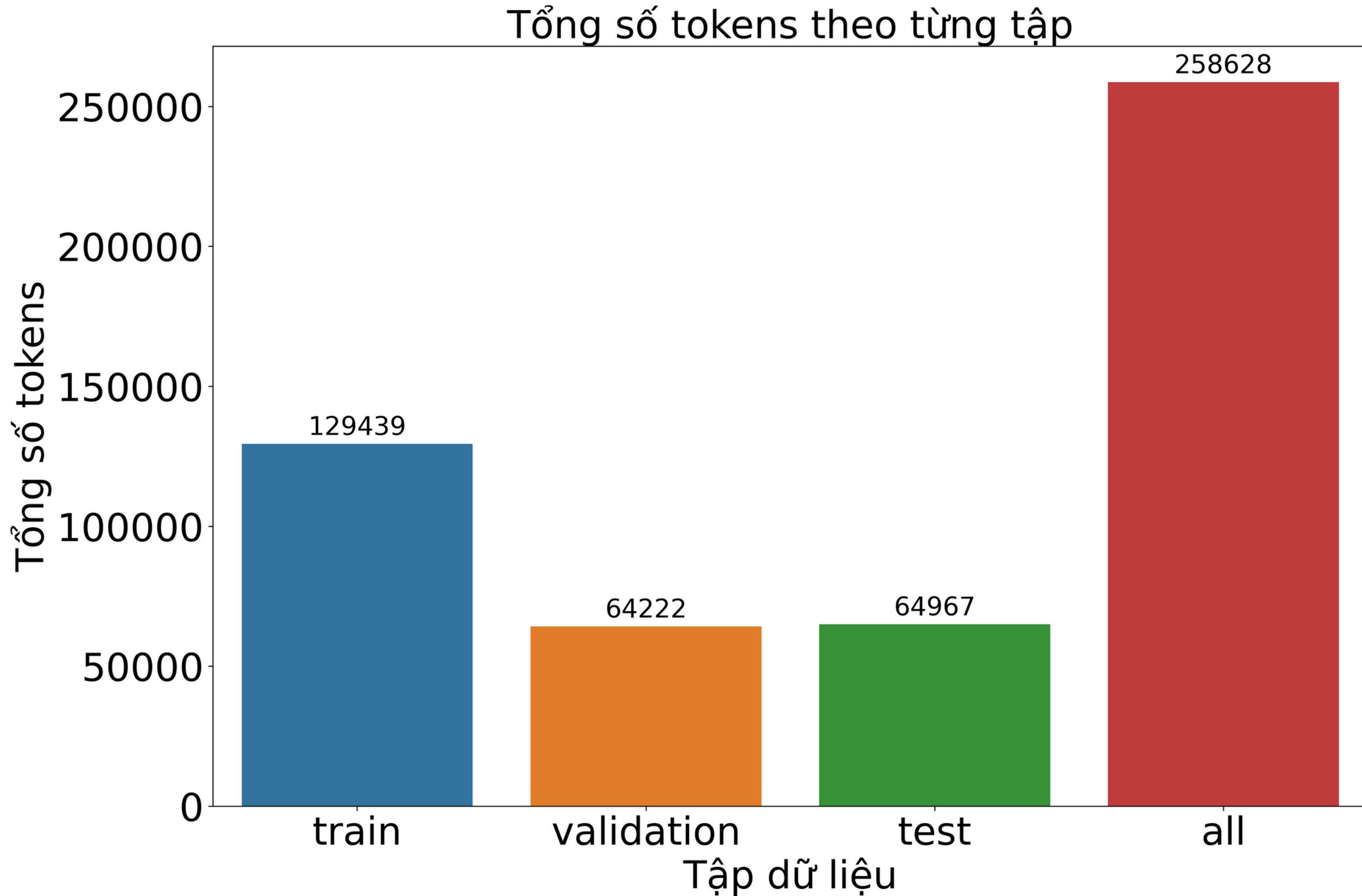
Độ dài
câu
ngắn
nhất và
dài nhất
trong
từng
tập





2.3 Thống kê dữ liệu

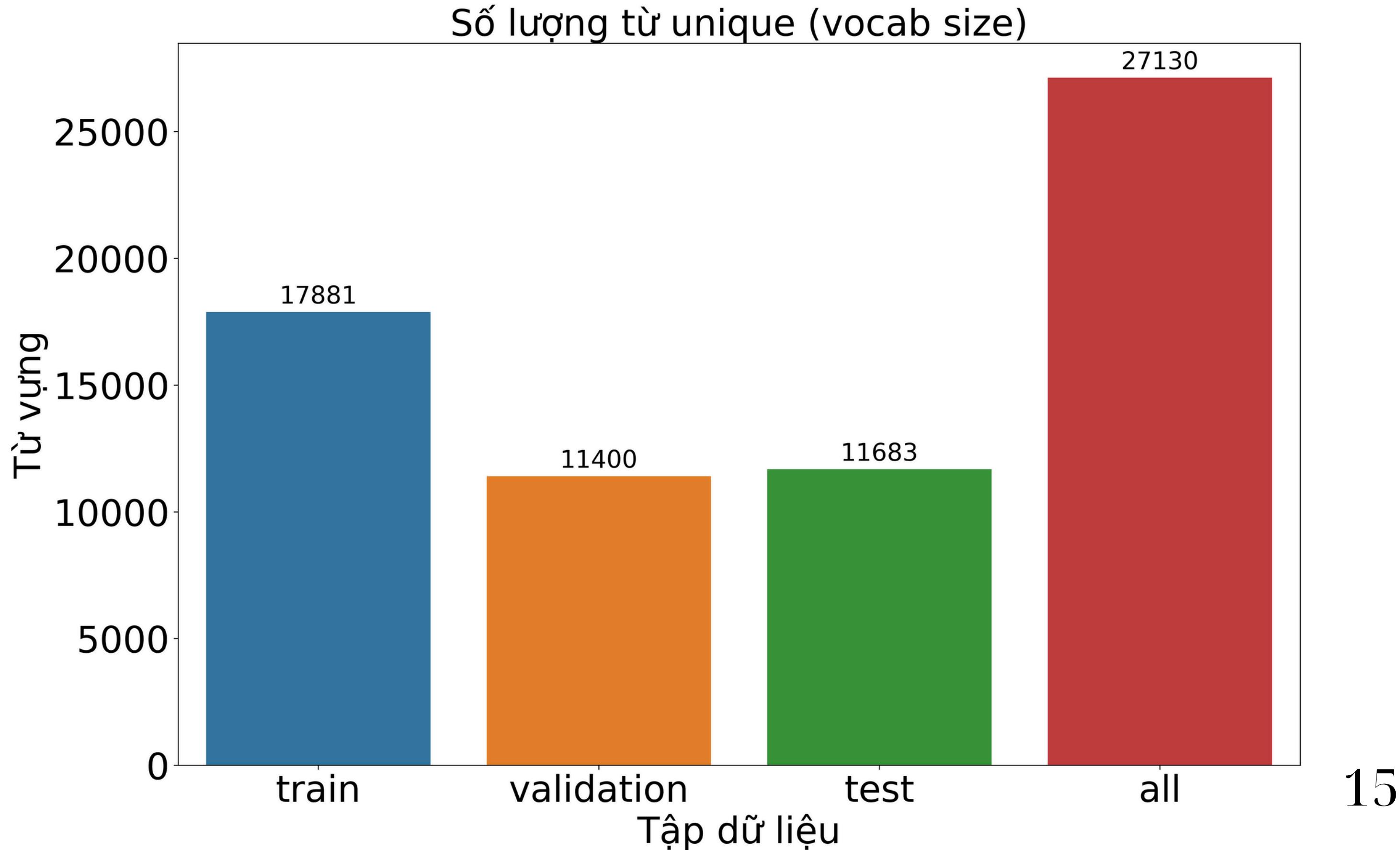
Tổng số
tokens
trong
từng
tập





2.3 Thống kê dữ liệu

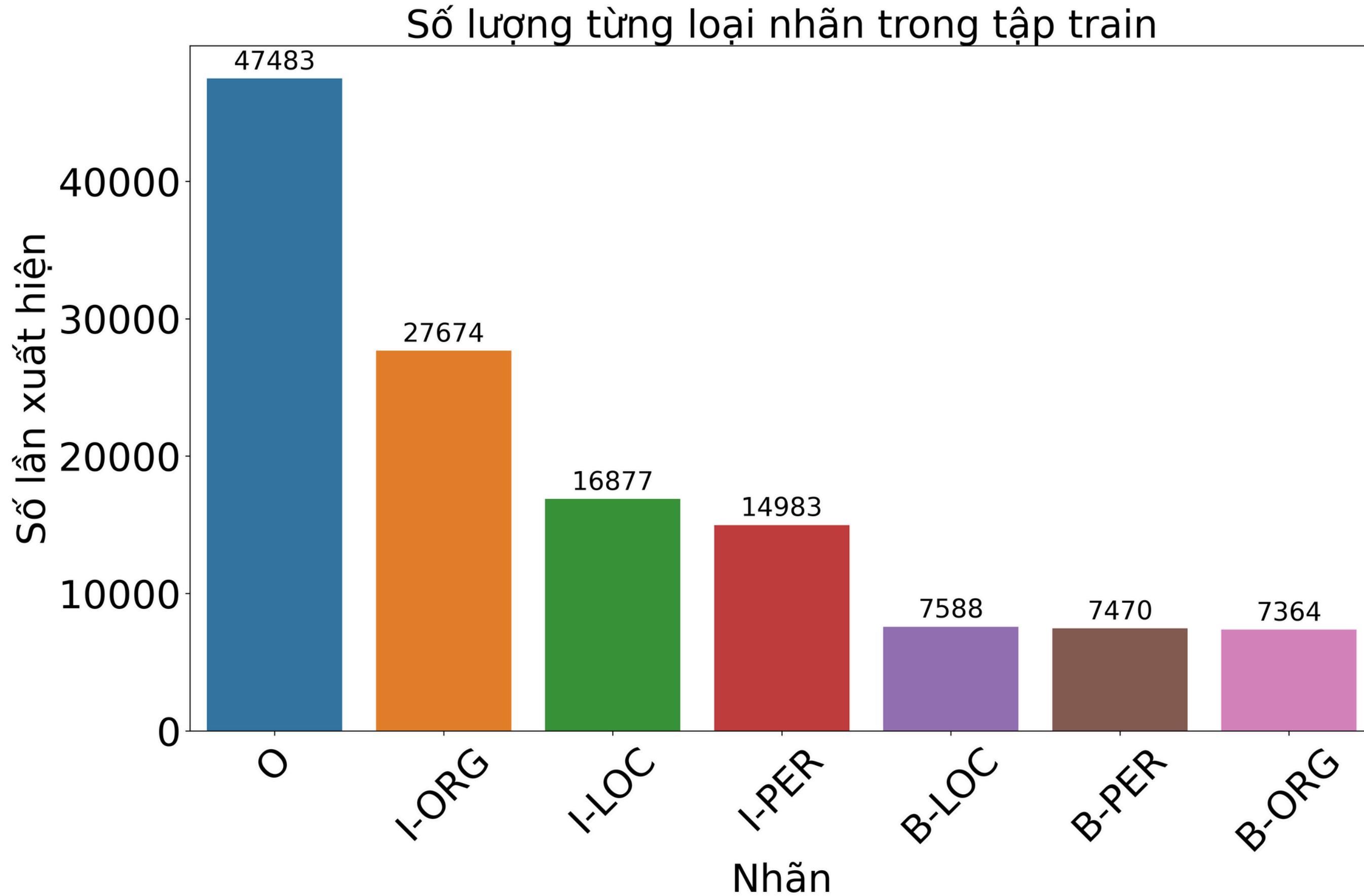
Số lượng token unique (vocab size)





2.3 Thống kê dữ liệu

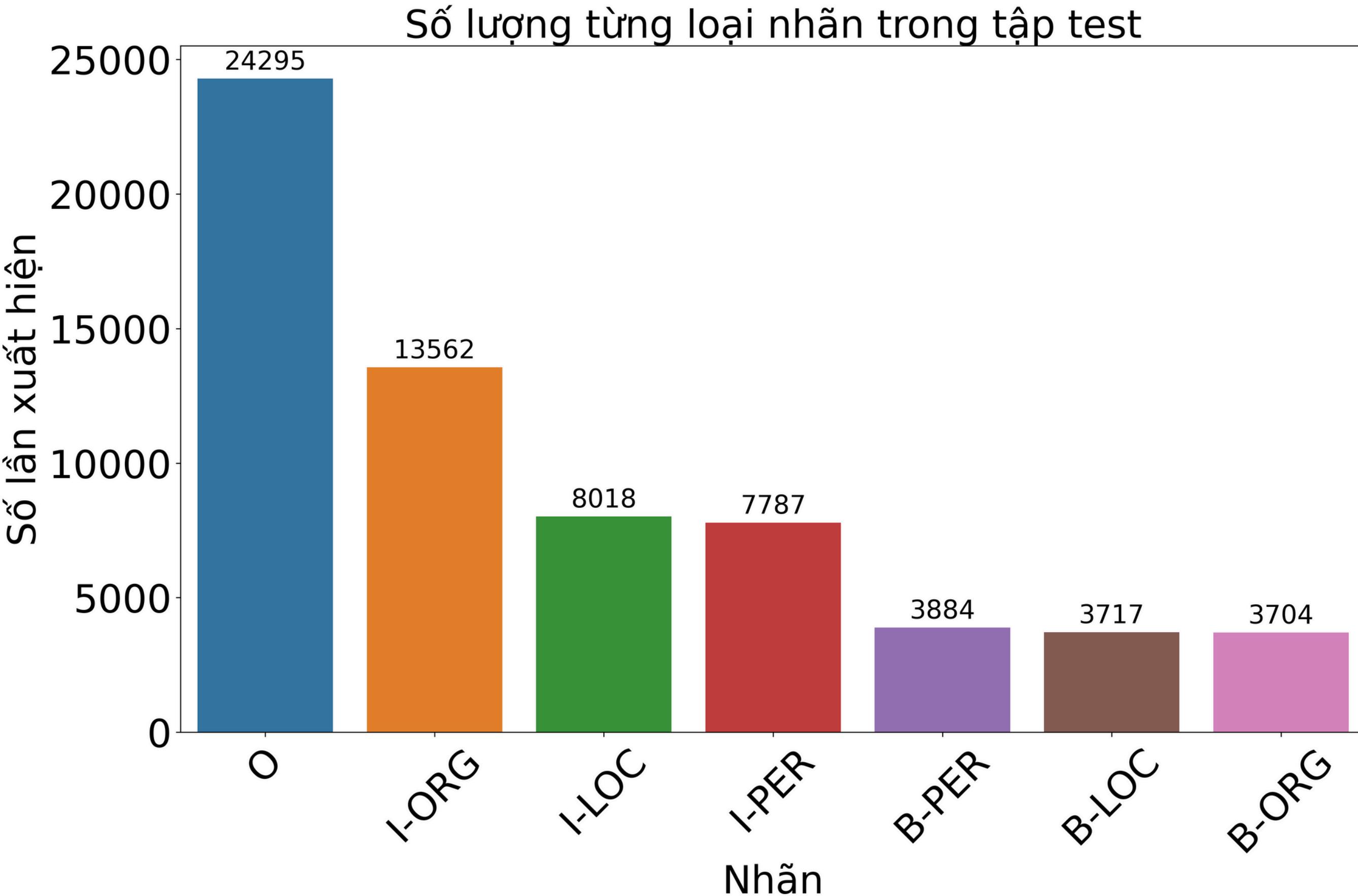
Số lượng từng loại nhãn trong train





2.3 Thống kê dữ liệu

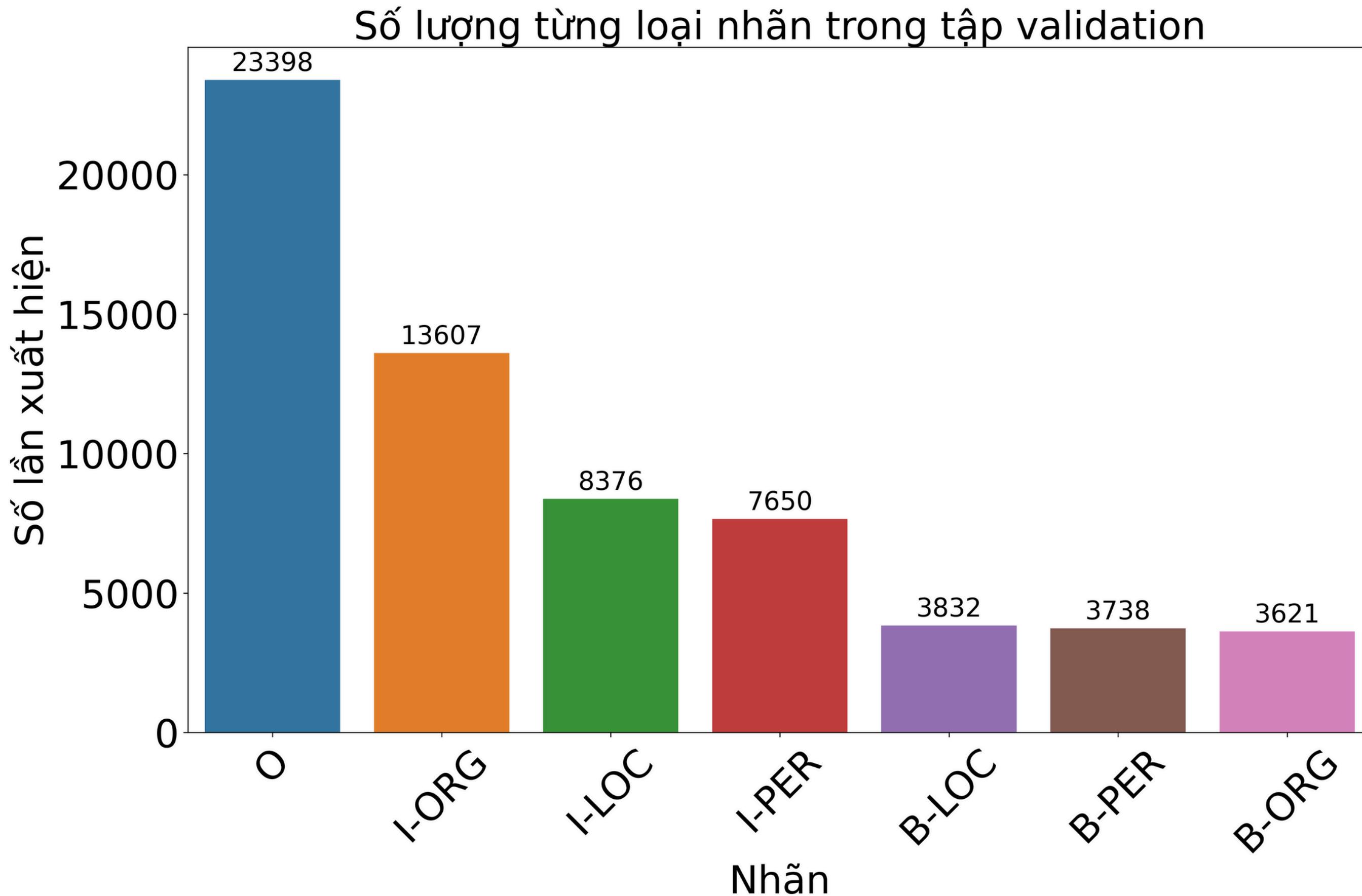
Số lượng từng loại nhãn trong test





2.3 Thống kê dữ liệu

Số lượng từng loại nhãn trong val





2.1 Phân tích ngữ liệu

Mẫu dữ liệu 1:

{

```
'tokens': ['-','Cá','sấu','Gena','"]  
'ner_tags': [0,1,2,2,0]  
'langs': ['vi','vi','vi','vi','vi']  
'spans': ['PER: Cá sấu Gena']  
}
```

- Các token “Cá”, “sấu”, “Gena” là tên một nhân vật, nên được gán là PER, token “Cá” bắt đầu chuỗi danh từ nên được gán 1 (B-PER) , “sấu” và “Gena” đi sau và cùng 1 chủ thể nên được gán 2 (I-PER)
- Các token còn lại thì không thuộc thực thể nào nên đều là 0 (0).



2.1 Phân tích ngữ liệu

Mẫu dữ liệu 2: {

```
'tokens': ['Sân', 'vận', 'động', 'bóng', 'đá', 'Ulsan', 'Munsu']  
'ner_tags': [3, 4, 4, 4, 4, 4, 4]  
'langs': ['vi', 'vi', 'vi', 'vi', 'vi', 'vi', 'vi']  
'spans': ['ORG: Sân vận động bóng đá Ulsan Munsu']  
}
```

- Tất cả các token trong câu đều thuộc cùng một thực thể, nên tất cả đều là ORG
- “Sân” là token bắt đầu nên là 3 (B-ORG)
- Còn lại các token khác, đều là 4 (I-ORG)



2.1 Phân tích ngữ liệu

Có sự sai sót về ngữ nghĩa trong nhãn của mẫu 2:

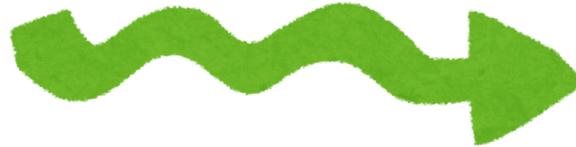


Sân vận động bóng đá Ulsan Munsu



2.1 Phân tích ngữ liệu

Có sự sai sót về ngữ nghĩa trong nhãn của mẫu 2:



Là một sân vận động nên phải dùng nhãn B-LOC và I-LOC chứ không phải B-ORG và I-ORG

Sân vận động bóng đá Ulsan Munsu



2.1 Phân tích ngữ liệu

Cách bộ dữ liệu WikiANN được gán nhãn

Mỗi câu sẽ được thu thập từ các tiêu đề hoặc câu trong wikipedia, có dạng:

[[Anchor link|Span]]

Trong đó *Anchor link* là đường dẫn tới bài viết gốc, và *Span* là nội dung hiển thị sẽ được biến thành các tokens

Nhãn của từng token sẽ được quyết định bằng những **categories** mà bài viết gốc thuộc về

Toàn bộ quá trình trên được thực hiện trong **tiếng Anh**, sau đó ánh xạ sang các ngôn ngữ khác của Wikipedia



2.1 Phân tích ngữ liệu

Cá sấu Gena

Gena the Crocodile

[Article](#) [Talk](#)

From Wikipedia, the free encyclopedia

For the first film of the series, see [Gena the Crocodile \(film\)](#).

Gena the Crocodile (Russian: Крокодил Гена, romanized: *Krokodil Gena*) is a fictional character found in Russian media, particularly animation.

History [edit]

Soviet era: Literature and puppetry [edit]

Gena is known as a friendly crocodile in the series of animation films [Gena the Crocodile](#), [Cheburashka](#) and [Shapoklyak](#) by Roman Kachanov (Soyuzmultfilm studio).

He debuted in the 1966 novel [Gena the Crocodile and His Friends](#) (ru) by [Eduard Uspensky](#).^[1] The crocodile's name is a typical [diminutive](#) of the Russian male name [Gennady](#). Gena and Cheburashka, also a title character in the series, are best friends.

The 50-year-old Gena works in a zoo as an attraction (or, as the original novel's author Uspensky had put it, "Gena the Crocodile worked in a zoo as a crocodile"). In his spare time, he plays the [garmon](#) and likes to sing. His two best-known songs are "Pust' begut neuklyuzhe..." and "Goluboy wagon" ("The Blue Train Car").

Sân vận động bóng đá Ulsan Munsu

Ulsan Munsu Football Stadium

[Article](#) [Talk](#)

From Wikipedia, the free encyclopedia

The **Ulsan Munsu Football Stadium** (Korean: 울산문수축구경기장) is a football stadium in [Ulsan](#), South Korea with a capacity for 37,897 spectators. Since 2001, it has been the home ground of [K League 1](#) team [Ulsan HD](#).

The stadium was built from 18 December 1998 to 28 April 2001 at a total cost of 151.4 billion [won](#) (US\$116.5 million).

International matches [edit]

The venue hosted three matches at the [2002 FIFA World Cup](#).

Date	Team 1	Result	Team 2	Round
1 June 2002	Uruguay	1–2	Denmark	Group A
3 June 2002	Brazil	2–1	Turkey	Group C
21 June 2002	Germany	1–0	United States	Quarter-finals



2.1 Phân tích ngữ liệu

Cá sấu Gena

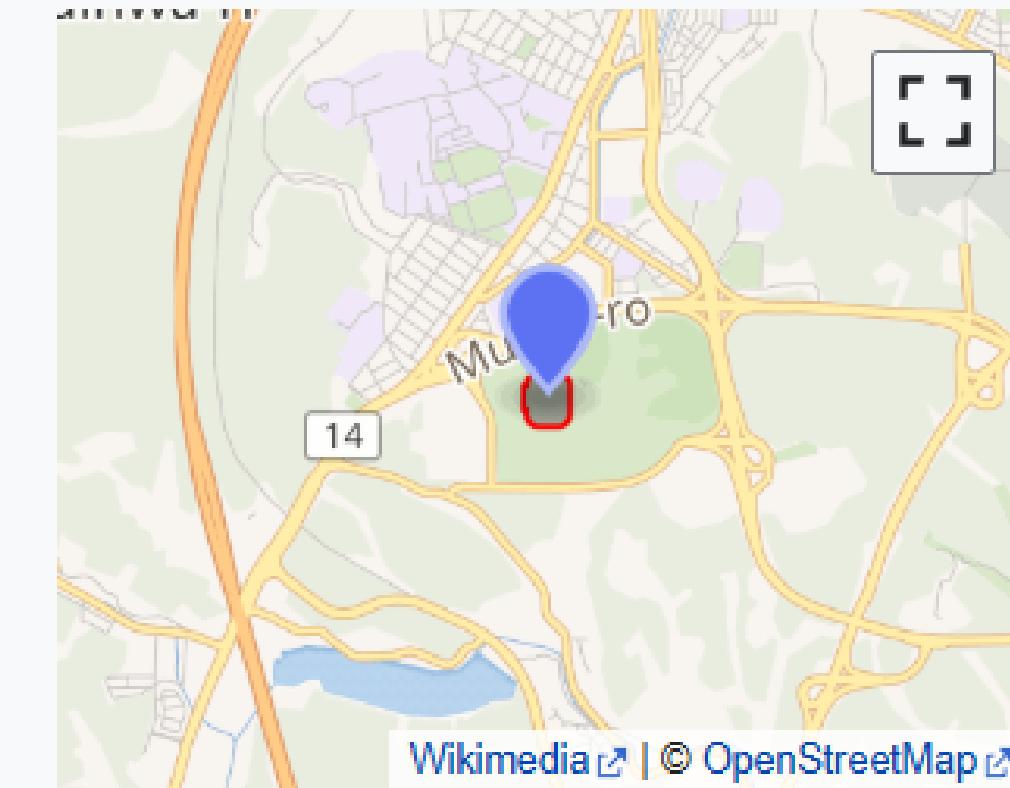


USSR stamp depicting Gena and Cheburashka, 1988.

Created by	Eduard Uspensky
Designed by	Eduard Uspensky Roman Kachanov & his companions Makoto Nakamura Dmitry Dyachenko & his equipes

Sân vận động bóng đá Ulsan Munsu

Ulsan Munsu Football Stadium in 2012



Wikimedia | © OpenStreetMap

Location	San 5-1, Ok-dong, Nam-gu, Ulsan , South Korea
Coordinates	35°32'07"N 129°15'34"E
Owner	Ulsan Metropolitan City Hall
Operator	Ulsan Facilities Corporation
Capacity	37,897 ^[1]
Surface	Grass



2.1 Phân tích ngữ liệu

Cá sấu Gena

Categories: [Literary characters introduced in 1966](#) | [Characters in children's literature](#) | [Fictional crocodilians](#) | [Soviet animation](#)
| [Cheburashka](#) | [Anthropomorphic crocodilians](#) | [Male characters in literature](#) | [Short stories about talking animals](#)
[Male characters in animation](#) | [Animated characters introduced in 1969](#) | [Animated films about talking animals](#)

Sân vận động bóng đá Ulsan Munsu

Categories: [Ulsan HD FC](#) | [Venues of the 2002 Asian Games](#) | [Football venues in South Korea](#) | [Sports venues in Ulsan](#)
| [Sports venues completed in 2001](#) | [2001 establishments in South Korea](#) | [K League 1 stadiums](#)



2.1 Phân tích ngữ liệu

Cá sấu Gena

Categories [Literary characters introduced in 1966](#) [Characters in children's literature](#) | [Fictional crocodilians](#) | [Soviet animation](#)
[Cheburashka](#) | [Anthropomorphic crocodilians](#) | [Male characters in literature](#) | [Short stories about talking animals](#)
[Male characters in animation](#) | [Animated characters introduced in 1969](#) | [Animated films about talking animals](#)

Sân vận động bóng đá Ulsan Munsu

Categories [Ulsan HD FC](#) | [Venues of the 2002 Asian Games](#) | [Football venues in South Korea](#) | [Sports venues in Ulsan](#)
[Sports venues completed in 2001](#) | [2001 establishments in South Korea](#) | [K League 1 stadiums](#)



3. Phương pháp

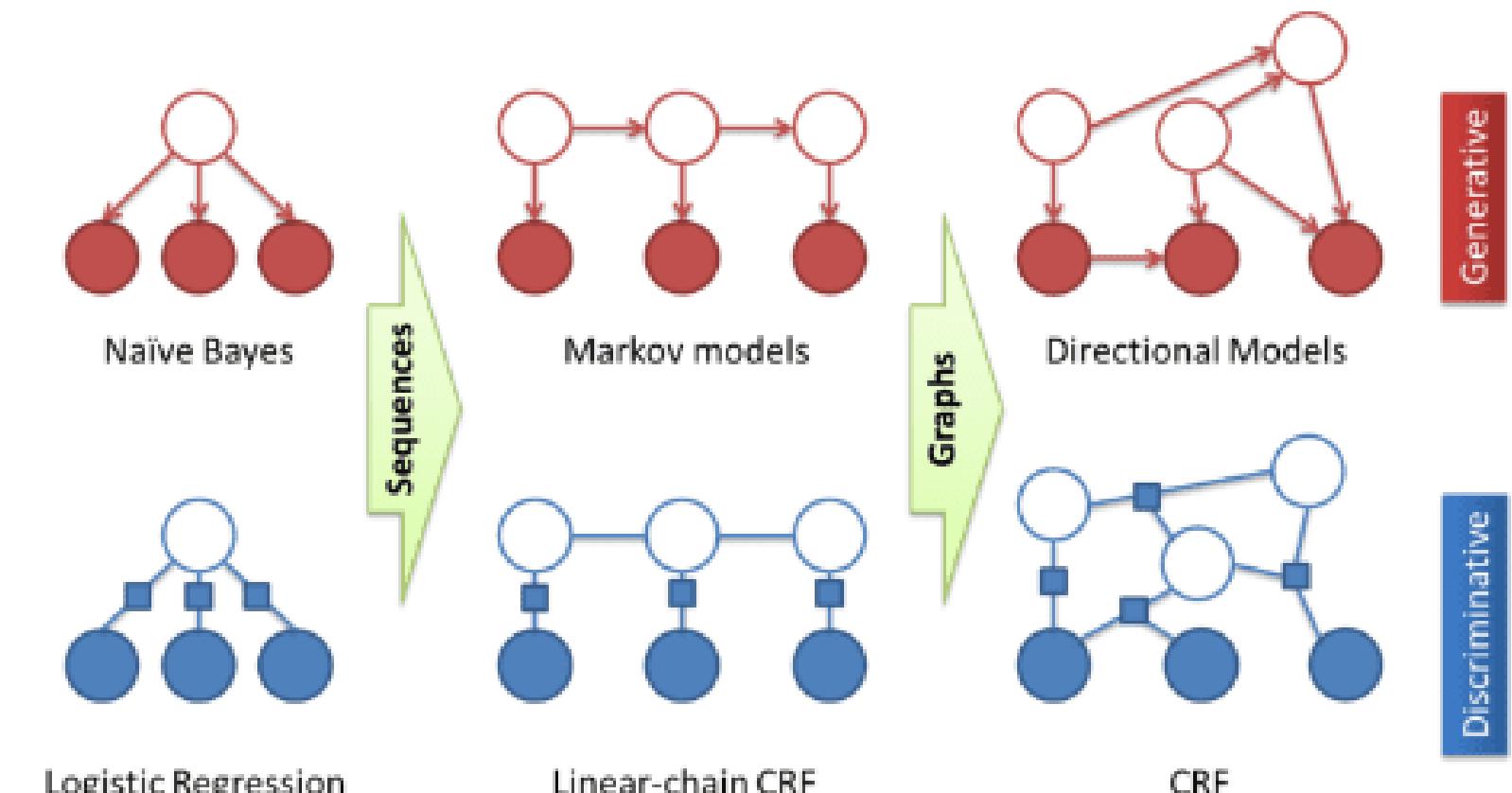




3. Phương pháp

Conditional Random Fields - CRF:

- 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.
- Trực tiếp mô hình hóa xác suất có điều kiện $P(\text{Chuỗi nhãn} | \text{Chuỗi quan sát})$.
- Mục tiêu: Dự đoán chuỗi nhãn tốt nhất cho một chuỗi đầu vào.



Adapted from C. Sutton, A. McCallum, "An Introduction to Conditional Random Fields", ArXiv, November 2010



3. Phương pháp

Kiến trúc và Hoạt động

- Dựa trên đồ thị không hướng: Biểu diễn các phụ thuộc giữa các nhãn liền kề.
- Hàm đặc trưng (Feature Functions):
 - Là cốt lõi của CRF.
 - Có thể là bất kỳ hàm nào ánh xạ từ trạng thái (hoặc cặp trạng thái) và ngữ cảnh đầu vào đến một giá trị thực.
 - Rất linh hoạt, có thể kết hợp nhiều loại đặc trưng (từ, tiền tố, hậu tố, chữ hoa, vị trí, từ điển, POS tag, v.v.).
 - Ví dụ: "từ hiện tại là 'Hà Nội' và nhãn hiện tại là 'B-LOC'".
- Trọng số (Weights): Mỗi hàm đặc trưng có một trọng số được học trong quá trình huấn luyện, thể hiện mức độ quan trọng của đặc trưng đó.
- Chuẩn hóa toàn cục: Đảm bảo tổng xác suất của tất cả các chuỗi nhãn có thể bằng 1, giúp mô hình mạnh mẽ hơn.



3.Phương pháp

Ưu điểm trong NER

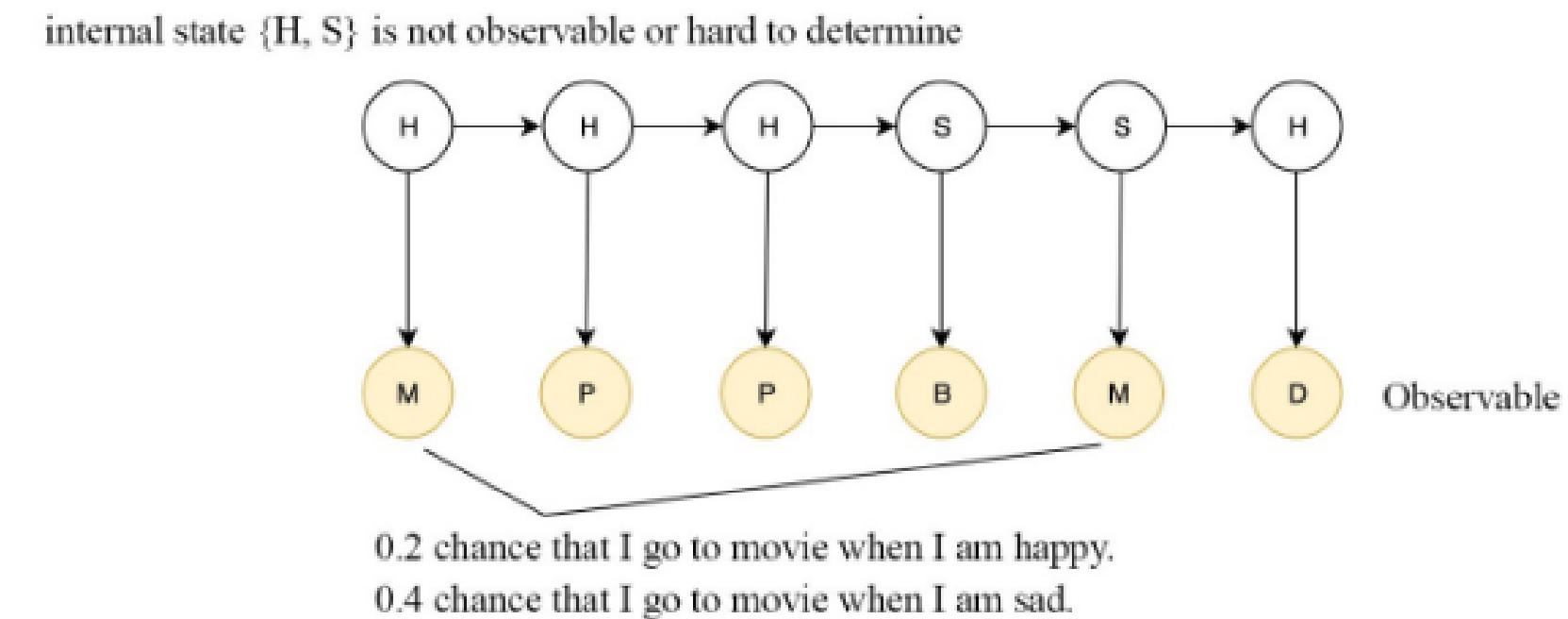
- Tích hợp đặc trưng phong phú: Khả năng sử dụng đa dạng các đặc trưng, kể cả những đặc trưng chồng chéo hoặc phụ thuộc ngữ cảnh rộng.
- Hiệu suất cao: Thường đạt kết quả tốt hơn các mô hình sinh ra trong các bài toán gắn nhãn trình tự do khả năng học phân biệt và chuẩn hóa toàn cục.
- Xử lý ngữ cảnh: Hiểu rõ hơn mối quan hệ giữa các từ và nhãn dựa trên ngữ cảnh toàn diện của câu.



3.Phương pháp

Hidden Markov Model (HMM) :

- Mô hình thống kê xác suất: Sử dụng cho chuỗi các sự kiện quan sát được dựa trên các trạng thái ẩn không thể quan sát trực tiếp.
- Mục tiêu: Học mối quan hệ giữa trạng thái ẩn và quan sát, và các chuyển đổi giữa các trạng thái ẩn.
- Cốt lõi : Giả định Markov bậc nhất - trạng thái hiện tại chỉ phụ thuộc vào trạng thái ngay trước đó.





3. Phương pháp

- 1. Tập hợp trạng thái ẩn (S): Các trạng thái nội tại không quan sát được (ví dụ: các nhãn NER như B-PER, I-LOC).
- 2. Tập hợp quan sát (O): Các sự kiện có thể quan sát được (ví dụ: các từ trong câu).
- 3. Ma trận xác suất chuyển trạng thái (A): - Xác suất chuyển từ trạng thái s_i sang s_j .
- 4. Ma trận xác suất phát xạ (B): - Xác suất quan sát o_k được phát ra khi ở trạng thái s_j .
- 5. Xác suất trạng thái ban đầu (π): - Xác suất bắt đầu tại mỗi trạng thái.



3.Phương pháp

Ưu và nhược điểm (so với CRF)

Ưu điểm: Đơn giản, dễ hiểu, hiệu quả cho các bài toán với giả định Markov đơn giản.

Nhược điểm:

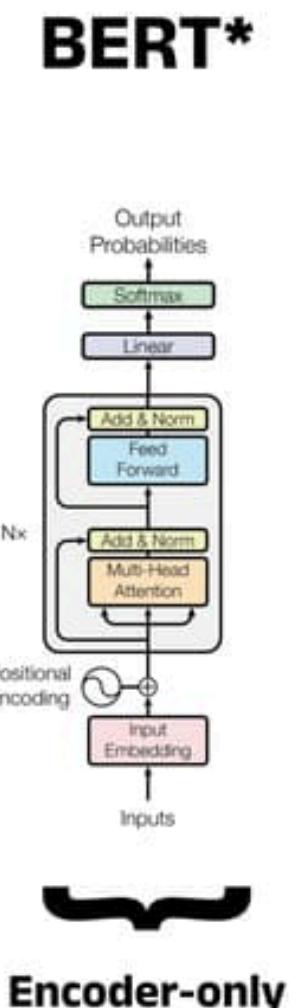
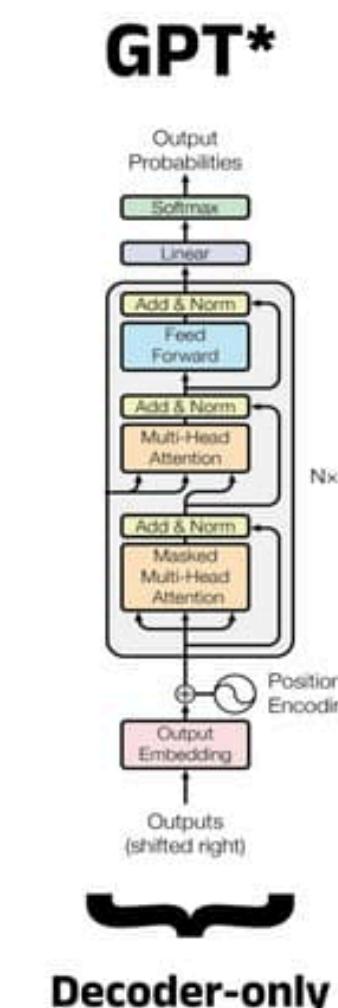
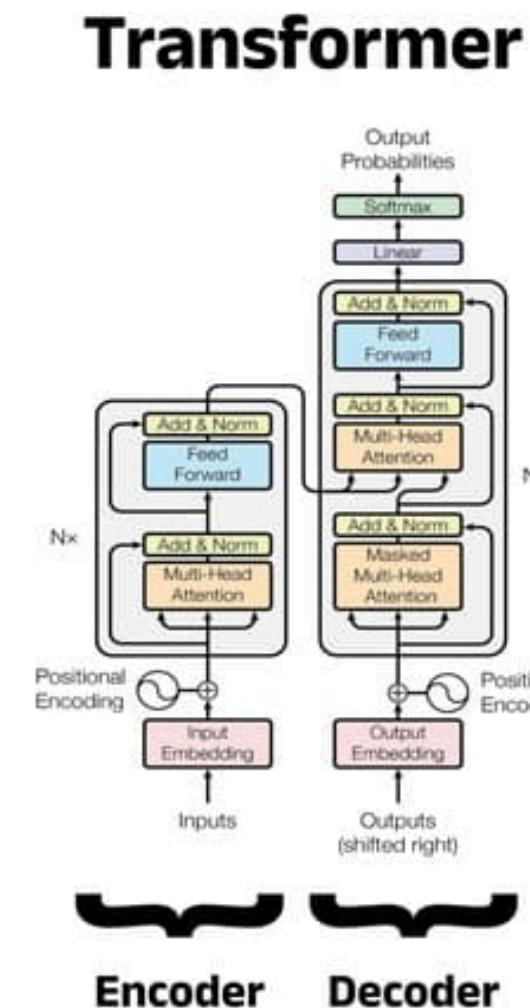
- Mô hình sinh ra: Mô hình hóa $P(O, S)$, không phải $P(S|O)$ trực tiếp.
- Chuẩn hóa cục bộ: Dễ gặp "Label Bias Problem".
- Khó tích hợp các đặc trưng phức tạp, chồng chéo hoặc phụ thuộc ngữ cảnh rộng.



3.Phương pháp

PhoBERT :

- PhoBERT là mô hình BERT được huấn luyện đặc biệt cho tiếng Việt.
- Phát triển bởi: VinAI Research (Viện Nghiên cứu Trí tuệ nhân tạo VinAI).
- Mục tiêu: Cải thiện hiệu suất các tác vụ Xử lý ngôn ngữ tự nhiên (NLP) trên tiếng Việt.



*Illustrative example, exact model architecture may vary slightly



3.Phương pháp

- Kiến trúc cơ sở: Dựa trên kiến trúc RoBERTa (một biến thể được tối ưu của BERT).
- Nhiệm vụ huấn luyện trước: Chỉ sử dụng nhiệm vụ Masked Language Model (MLM) để học biểu diễn ngôn ngữ sâu sắc, bỏ qua nhiệm vụ Next Sentence Prediction (NSP) của BERT gốc.
- Dữ liệu huấn luyện: Được huấn luyện trên một lượng lớn dữ liệu văn bản tiếng Việt.
- Tiền xử lý dữ liệu: Sử dụng RDRSegmenter của VnCoreNLP để tách từ cho dữ liệu đầu vào trước khi huấn luyện. Khuyến nghị sử dụng cùng bộ tách từ này cho các ứng dụng dựa trên PhoBERT.



3.Phương pháp

Ưu điểm nổi bật

- Hiệu suất vượt trội: PhoBERT đạt được hiệu suất tiên tiến (State-of-the-Art) trên nhiều tác vụ NLP tiếng Việt như:
- Nhận dạng thực thể có tên (Named-entity recognition - NER).
- Gắn thẻ từ loại (Part-of-speech tagging).
- Phân tích cú pháp phụ thuộc (Dependency parsing).
- Suy luận ngôn ngữ tự nhiên (Natural language inference).
- Hiểu biết ngôn ngữ sâu sắc: Do được huấn luyện trên dữ liệu tiếng Việt quy mô lớn, PhoBERT có khả năng nắm bắt cấu trúc ngữ pháp và ngữ cảnh đặc thù của tiếng Việt tốt hơn các mô hình đa ngôn ngữ.



4. Cài đặt thực nghiệm



4. Cài đặt thực nghiệm

Chuẩn bị dữ liệu

- Định dạng chung: Dữ liệu đầu vào cho cả ba mô hình đều được chuẩn hóa thành danh sách các câu, mỗi câu là một chuỗi các cặp (từ, nhãn).
- Phân chia tập dữ liệu: Dữ liệu được chia thành ba tập:
 - Tập huấn luyện (Training Set): Dùng để huấn luyện mô hình.
 - Tập kiểm định (Validation Set): Dùng để tinh chỉnh siêu tham số và theo dõi quá trình huấn luyện.
 - Tập kiểm tra (Test Set): Dùng để đánh giá hiệu suất cuối cùng của mô hình.



4. Cài đặt thực nghiệm

Xử lý dữ liệu đặc thù cho từng mô hình:

- CRF: Dữ liệu thô được chuyển đổi thành các tập đặc trưng (X) và nhãn (y) bằng các hàm sent2features và sent2labels.



4. Cài đặt thực nghiệm

Xử lý dữ liệu đặc thù cho từng mô hình:

- HMM: Dữ liệu huấn luyện (train_sents) được sử dụng trực tiếp ở định dạng (từ, nhãn) cho quá trình huấn luyện có giám sát.



4. Cài đặt thực nghiệm

Xử lý dữ liệu đặc thù cho từng mô hình:

- PhoBert:
 - Tokenization: chia mỗi từ trong câu thành một hoặc nhiều subword token.
 - Căn chỉnh nhãn (Label Alignment)
 - Padding và Truncation: Chuỗi token được điều chỉnh về độ dài tối đa (max_len). Nếu ngắn hơn max_len, nó được đệm bằng token [PAD] (tokenizer.pad_token_id). Nếu dài hơn, nó bị cắt bớt. Nhãn cho các token [PAD] cũng là -100.
 - Tạo Attention Mask



4. Cài đặt thực nghiệm

CRF:

- Thư viện sử dụng: `sklearn-crfsuite`.
- Trích xuất đặc trưng: Hàm `word2features` được sử dụng để tạo các bộ đặc trưng cho mỗi từ, bao gồm:
 - Đặc trưng hình thái: dạng chữ thường, ba ký tự cuối, viết hoa toàn bộ, dạng tiêu đề, là chữ số.
 - Đặc trưng ngữ cảnh: dạng chữ thường và dạng tiêu đề của từ liền trước và liền sau.
 - Đặc trưng vị trí: đánh dấu đầu câu (BOS) và cuối câu (EOS).
- Cấu hình mô hình: Khởi tạo CRF với thuật toán tối ưu hóa **`lbfgs`**, số lần lặp tối đa **`max_iterations=100`**, và cho phép tất cả các chuyển đổi trạng thái có thể có (**`all_possible_transitions=True`**).



4. Cài đặt thực nghiệm

HMM:

- Thư viện sử dụng: NLTK's HiddenMarkovModelTrainer.
- Không có trích xuất đặc trưng tường minh: HMM học trực tiếp các xác suất chuyển trạng thái và phát xạ từ các cặp (từ, nhãn) trong dữ liệu.
- Cấu hình và Huấn luyện: Mô hình được huấn luyện có giám sát bằng HiddenMarkovModelTrainer().train_supervised(). Một **estimator** sử dụng **LidstoneProbDist** với **gamma=0.1** được áp dụng để làm mịn xác suất, tránh trường hợp xác suất bằng 0 cho các sự kiện không thấy trong dữ liệu huấn luyện



4. Cài đặt thực nghiệm

PhoBERT

- Thư viện sử dụng: transformers của Hugging Face và torch (PyTorch).
- Tokenizer: AutoTokenizer.from_pretrained("vinai/phobert-base")
- Mô hình: AutoModelForTokenClassification.from_pretrained("vinai/phobert-base") - một phiên bản PhoBERT đã được huấn luyện trước.
- Cấu hình huấn luyện: Sử dụng TrainingArguments để định nghĩa các siêu tham số huấn luyện.



4. Cài đặt thực nghiệm

Metrics đánh giá : Sử dụng **Accuracy** và **F1-score, confusion matrix**

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

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

Kết quả sẽ được đánh giá trên tập test (**WikiANN Tiếng Việt**)

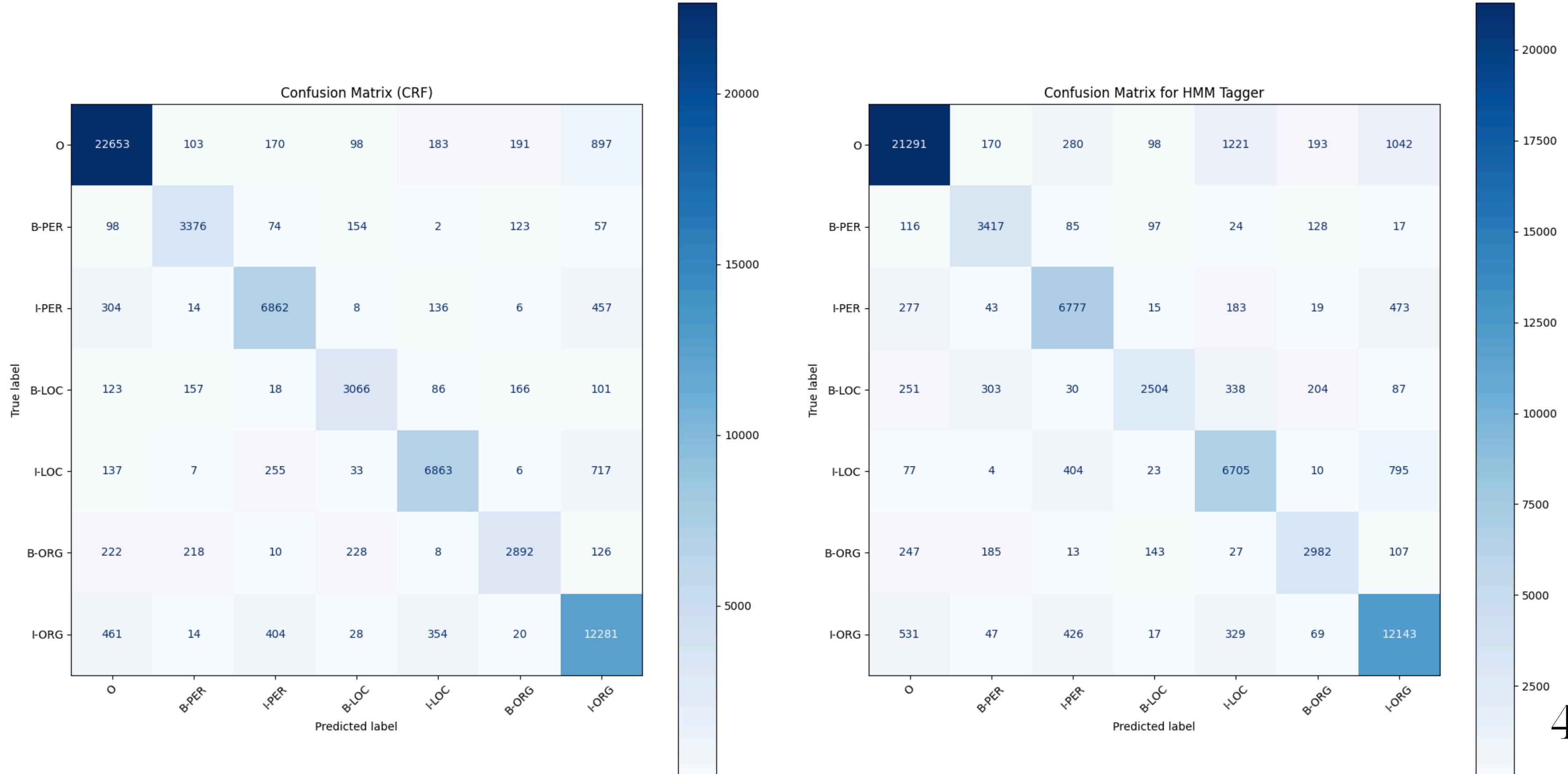


4. Cài đặt thực nghiệm

Kết quả trên tập test	Accuracy	F1-score
CRF	0.89	0.87
HMM	0.86	0.83
PhoBERT	0.96	0.91



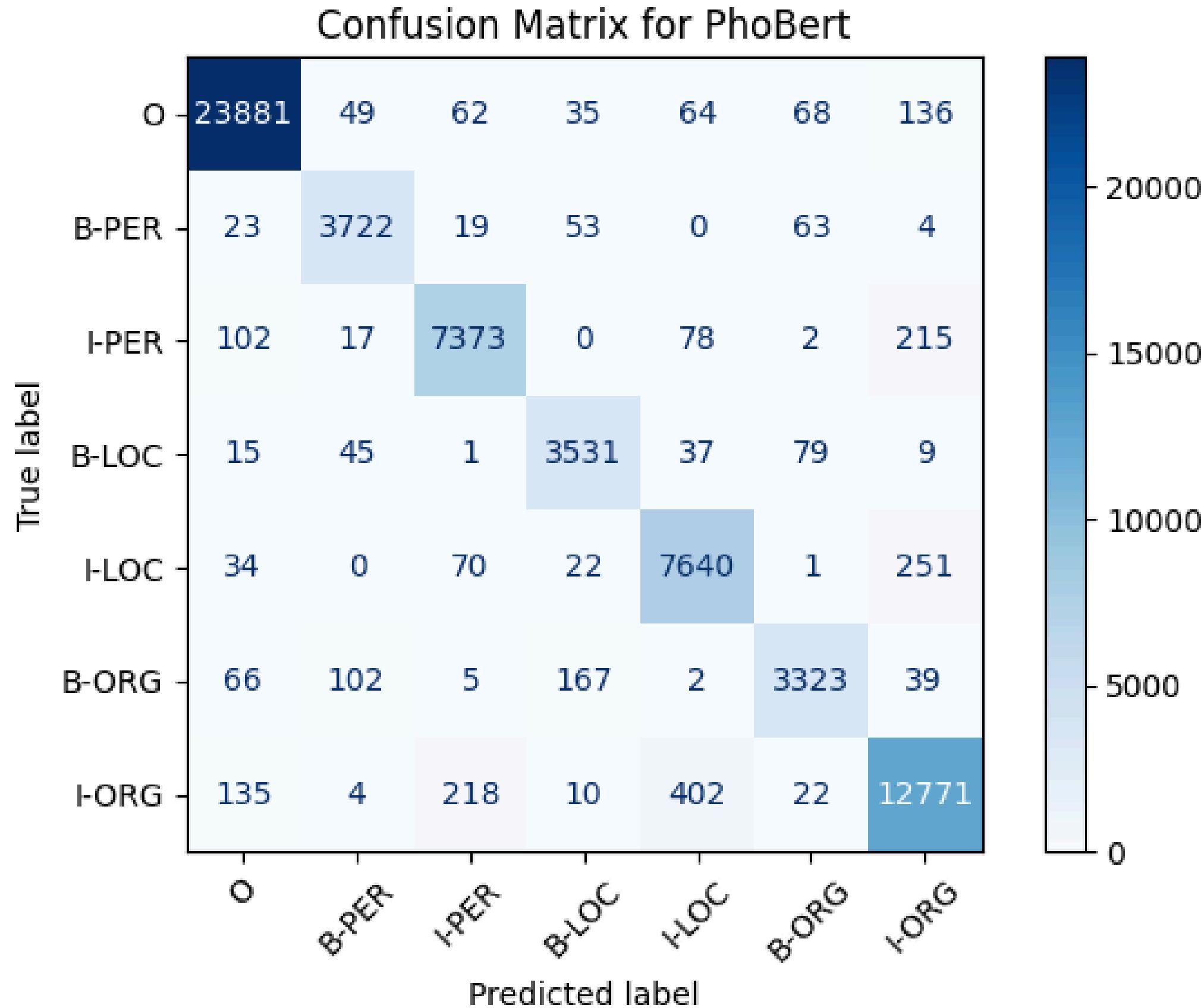
4. Cài đặt thực nghiệm





4. Cài đặt thực nghiệm

Confusion Matrix
của PhoBERT





4.2 Finetune PhoBERT model

Xử lý thêm khi data bị gán nhãn sai

- Như đã nói 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.



4.2 Finetune PhoBERT model

Xử lý thêm khi data bị gán nhãn sai

- Như đã nói 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.

Con	-> B-ORG
đường	-> I-ORG
tơ	-> I-ORG
lụa	-> I-ORG

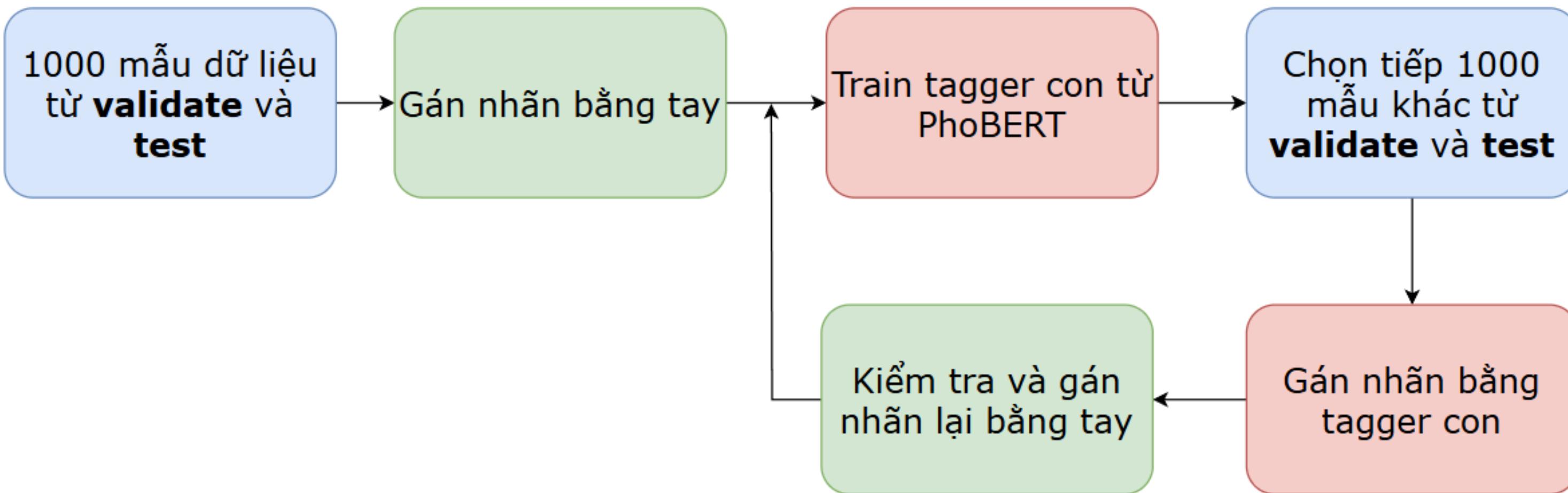
Công	-> 0
ty	-> 0
cổ	-> I-ORG
phần	-> I-ORG
Sữa	-> I-ORG
Việt	-> I-ORG
Nam	-> I-ORG
Vinamilk	-> B-ORG
là	-> 0
nhà	-> 0
sản	-> 0
xuất	-> 0
sữa	-> 0
hàng	-> 0
đầu	-> 0
Việt	-> B-LOC
Nam	-> I-LOC
.	-> 0



4.2 Finetune PhoBERT model

Hướng xử lý:

- Nhóm đã thực hiện gán nhãn lại dữ liệu bằng tay, sau đó tiếp tục finetune model PhoBERT đã được huấn luyện trước đó.
- Tính đến hiện tại, nhóm đã gán nhãn lại khoảng **2000** mẫu dữ liệu, lấy từ tập **validation (~1000)** và tập **test (~1000)**.





4.2 Finetune PhoBERT model

Hướng xử lý:

- Sau khi có dữ liệu mới, nhóm thực hiện finetune model PhoBERT chính (đã được train trên 20000 mẫu của tập train)
- Do đã được finetune với những dữ liệu khác, nên việc đánh giá mô hình mới này trên tập test là vô nghĩa, nên nhóm không đánh giá trên tập test, mà thực hiện so sánh với mô hình PhoBERT cũ trên một số câu văn bên ngoài.



4.2 Finetune PhoBERT model

Tokens	Finetuned	PhoBERT	Tokens	Finetuned	PhoBERT
Ông	-> 0	-> 0	Sông	-> B-LOC	-> B-LOC
Nguyễn	-> B-PER	-> B-PER	Hồng	-> I-LOC	-> I-LOC
Thanh	-> I-PER	-> I-PER	chảy	-> 0	-> 0
Long	-> I-PER	-> I-PER	qua	-> 0	-> 0
từng	-> 0	-> 0	thủ	-> 0	-> 0
là	-> 0	-> 0	đô	-> 0	-> 0
Bộ	-> 0	-> 0	Hà	-> B-LOC	-> B-LOC
trưởng	-> 0	-> 0	Nội	-> I-LOC	-> I-LOC
Bộ	-> B-ORG	-> B-ORG	của	-> 0	-> 0
Y	-> I-ORG	-> I-ORG	Việt	-> B-LOC	-> B-LOC
tế	-> I-ORG	-> I-ORG	Nam	-> I-LOC	-> I-LOC
.	-> 0	-> 0	.	-> 0	-> 0

Mô hình finetuned vẫn hoạt động tốt với một số mẫu mà PhoBERT vẫn đúng



4.2 Finetune PhoBERT model

Tokens	Finetuned	PhoBERT
Con	-> B-LOC -> B-ORG	
đường	-> I-LOC -> I-ORG	
tơ	-> I-LOC -> I-ORG	
lụa	-> I-LOC -> I-ORG	

Mô hình Finetuned có sự cải thiện

Công	-> B-ORG	-> 0
ty	-> I-ORG	-> 0
cổ	-> I-ORG	-> I-ORG
phần	-> I-ORG	-> I-ORG
Sữa	-> I-ORG	-> I-ORG
Việt	-> I-ORG	-> I-ORG
Nam	-> I-ORG	-> I-ORG
Vinamilk	-> I-ORG	-> B-ORG
là	-> 0	-> 0
nha	-> 0	-> 0
sản	-> 0	-> 0
xuất	-> 0	-> 0
sữa	-> 0	-> 0
hàng	-> 0	-> 0
đầu	-> 0	-> 0
Việt	-> B-LOC	-> B-LOC
Nam	-> I-LOC	-> I-LOC
.	-> 0	-> 0



4.2 Finetune PhoBERT model

Thành → B-LOC → B-LOC
phố → I-LOC → I-LOC
Hồ → I-LOC → I-LOC
Chi → I-LOC → I-LOC
Minh → I-LOC → I-LOC
là → 0 → 0
trung → 0 → 0
tâm → 0 → I-LOC
kinh → 0 → I-LOC
tế → 0 → I-LOC
lớn → 0 → 0
nhất → 0 → 0
Việt → B-LOC → B-LOC
Nam → I-LOC → I-LOC
. → 0 → 0

Tổng → B-PER → B-ORG
Bí → I-PER → I-ORG
thư → I-PER → I-ORG
Tô → I-PER → I-ORG
Lâm → I-PER → I-ORG
đã → 0 → 0
có → 0 → 0
bài → 0 → 0
phát → 0 → 0
biểu → 0 → 0
quan → 0 → 0
trọng → 0 → 0
. → 0 → 0

Mô hình Finetuned có sự cải thiện



4.2 Finetune PhoBERT model

Tokens	Finetuned	PhoBERT	
Cầu	-> B-LOC	-> B-ORG	
Cần	-> I-LOC	-> I-ORG	
Thơ	-> I-LOC	-> I-ORG	
nối	-> 0	-> 0	
liền	-> 0	-> 0	
hai	-> 0	-> 0	
bờ	-> 0	-> 0	
sông	-> I-LOC	-> I-LOC	
Hậu	-> I-LOC	-> I-LOC	
thuộc	-> 0	-> 0	
thành	-> B-LOC	-> B-LOC	
phố	-> I-LOC	-> I-LOC	
Cần	-> I-LOC	-> I-LOC	
Thơ	-> I-LOC	-> I-LOC	
.	-> 0	-> 0	

**Tuy nhiên vẫn
có một số câu
model finetuned
chưa thực sự
tốt**



4.2 Finetune PhoBERT model

Nhận xét:

- Mặc dù đã cải thiện đáng kể so với mô hình gốc, nhưng mô hình vẫn còn những câu sai sót.
- Nếu gán thêm dữ liệu tay, finetune tiếp thì mô hình có thể sẽ cải thiện thêm



5. Kết luận



THE END!!!

Cảm ơn thầy và các bạn đã
lắng nghe





References



- 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
- ChatGPT

