

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO ĐỒ ÁN
MÔN NGÔN NGỮ HỌC NGỮ LIỆU

Đề tài:

TÌM HIỂU QUY TRÌNH PHÁT TRIỂN DỮ LIỆU
NHẬN DẠNG THỰC THỂ TÊN RIÊNG TIẾNG VIỆT

Nhóm sinh viên thực hiện: Nhóm 3

- | | |
|---------------------|----------|
| 1. Trần Tuấn Vỹ | 18520406 |
| 2. Huỳnh Trọng Khoa | 18520918 |
| 3. Nguyễn Tấn Phúc | 18521259 |

Giảng viên: TS. Nguyễn Thị Quý

Mã lớp học phần: CS321.M11.KHCL

☪ Tp. Hồ Chí Minh, 12/2021 ☪

LỜI NÓI ĐẦU

Nhận dạng Thực thể Tên riêng thuộc nhánh Xử lý Ngôn ngữ Tự nhiên ngày càng nhận được sự quan tâm, chú ý của cộng đồng Khoa học Máy tính nói chung và cộng đồng Công nghệ Thông tin nói riêng do sự phát triển vượt bậc của Trí tuệ Nhân tạo trong những năm gần đây. Khi thực hiện nghiên cứu và triển khai các ứng dụng trên lĩnh vực Xử lý Ngôn ngữ Tự nhiên, kho dữ liệu đóng vai trò là một trong những thành tố quan trọng nhất khiến quy trình phát triển một kho dữ liệu hợp lý và khoa học cũng trở nên cấp thiết hơn bao giờ hết. Thông qua việc tham gia môn học Ngôn ngữ học Ngữ liệu – CS321.M11.KHCL do cô Nguyễn Thị Quý hướng dẫn, nhóm đã được tiếp thu thêm nhiều kiến thức mới về quy trình phát triển một kho dữ liệu cũng như những vấn đề xoay quanh một cách tường tận và sâu sắc.

“Tìm hiểu quy trình phát triển dữ liệu Nhận dạng Thực thể Tên riêng tiếng Việt” chính là đề tài nhóm chọn cho đồ án môn Ngôn ngữ học Ngữ liệu với mong muốn hiểu được quy trình xây dựng và các kinh nghiệm thực tiễn xoay quanh một kho ngữ liệu. Để hoàn thành tốt đồ án, nhóm đã nghiên cứu các lý thuyết về bài toán Nhận dạng Thực thể Tên riêng cũng như cách thực hiện gán nhãn và lưu trữ dữ liệu sao cho hợp lý.

Để báo cáo môn học được thực hiện thành công tốt đẹp, nhóm xin chân thành cảm ơn Khoa Khoa học Máy tính đã tạo điều kiện, cơ hội để giảng viên và sinh viên tiến hành nghiên cứu, tìm hiểu môn học Ngôn ngữ học Ngữ liệu. Nhóm cũng xin chân thành gửi lời cảm ơn đến cô Nguyễn Thị Quý, giảng viên khoa Khoa học Máy tính đã trực tiếp chỉ dạy, hướng dẫn tận tình trong suốt quá trình học tập, tiếp cận môn học cũng như đề tài “Tìm hiểu quy trình phát triển dữ liệu Nhận dạng Thực thể Tên riêng tiếng Việt”. Trong quá trình tìm hiểu và thực hiện báo cáo sẽ không tránh khỏi sai sót, nhóm mong nhận được những ý kiến đóng góp từ cô để báo cáo được hoàn thiện hơn.

Thủ Đức, tháng 12 năm 2021

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:

Họ và tên	MSSV	Phân công	Đánh giá
Trần Tuấn Vỹ	18520406	<ul style="list-style-type: none"> - Tìm hiểu nội dung bài báo, guideline - Thực hiện quá trình gán nhãn (Annotator 1) - Thảo luận. 	Hoàn thành
Huỳnh Trọng Khoa	18520918	<ul style="list-style-type: none"> - Tìm hiểu nội dung bài báo, guideline. - Xây dựng công cụ gán nhãn bán tự động. - Tổ chức quá trình thực hiện. - Thảo luận 	Hoàn thành
Nguyễn Tấn Phúc	18521259	<ul style="list-style-type: none"> - Tìm hiểu nội dung bài báo, guideline - Thực hiện quá trình gán nhãn (Annotator 2) - Thảo luận. 	Hoàn thành

MỤC LỤC

Chương 1: GIỚI THIỆU BÀI TOÁN.....	5
1.1.Nhận dạng thực thể tên riêng.....	5
1.2Bài báo tham khảo	5
1.2.1 Nội dung bài báo	5
1.2.2 Các loại thực thể.....	6
1.2.3 Thông tin về bộ dữ liệu	6
Chương 2: CÔNG CỤ GÁN NHÃN BÁN TỰ ĐỘNG.....	7
2.1Giới thiệu	7
2.2Mô hình.....	8
2.3Giao diện công cụ	10
Chương 3: QUY TRÌNH THỰC HIỆN.....	13
3.1Chuẩn bị dữ liệu.....	13
3.2Huấn luyện Annotators	13
3.3Phương pháp đánh giá.....	14
3.4Kết quả.....	15
Chương 4: KẾT LUẬN.....	18
TÀI LIỆU THAM KHẢO	19

Chương 1: GIỚI THIỆU BÀI TOÁN

1.1 .Nhận dạng thực thể tên riêng

Nhận dạng Thực thể Tên riêng (Named Entity Recognition – NER) là một trong những kỹ thuật then chốt trong lĩnh vực Xử lý ngôn ngữ tự nhiên có nhiệm vụ nhận diện các từ, cụm từ trong văn bản và phân loại chúng vào trong các nhóm đã được định trước [1]. Thêm vào đó, Nhận dạng Thực thể Tên riêng còn là một tác vụ thiết yếu trong quá trình xây dựng cơ sở dữ liệu và là tiền đề cho các bài toán Truy xuất Thông tin, Hệ thống hỏi-đáp, Chatbox, và các bài toán khác.

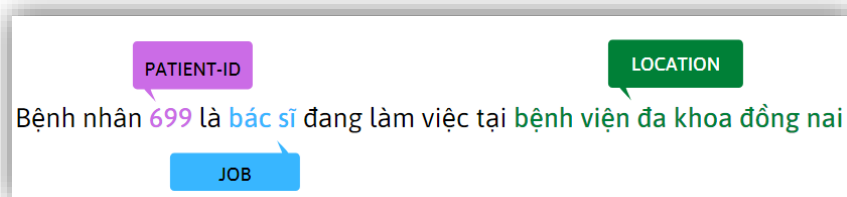


Figure 1. Ảnh minh họa cho NER

1.2 Bài báo tham khảo

Bài báo mà nhóm tham khảo để thực hiện đề tài này có tên “**COVID-19 NAMED ENTITY RECOGNITION FOR VIETNAMESE**” do nhóm tác giả thuộc VinAI Research thực hiện và được đăng ở hội nghị NAACL 2021.



Figure 2. Logo VinAI Research



Figure 3. Logo NAACL

1.2.1 Nội dung bài báo

Trình bày bộ dữ liệu đầu tiên tập trung vào nội dung COVID-19 được gán nhãn thủ công cho tiếng Việt. Cụ thể hơn, tập dữ liệu được gán nhãn cho bài toán Nhận dạng Thực thể Tên riêng (NER) bằng loại thực thể được tái định nghĩa để có thể được sử dụng cho các đại dịch khác trong tương lai. Tập dữ liệu còn có số

lượng thực thể lớn nhất khi so với các tập dữ liệu NER cho tiếng Việt hiện nay [2].

1.2.2 Các loại thực thể

STT	Nhãn thực thể	Định nghĩa
1	PATIENT-ID	Mỗi bệnh nhân COVID-19 ở Việt Nam có một mã định danh “X” (bệnh nhân thứ X)
2	NAME	Tên bệnh nhân hoặc người có liên quan đến bệnh nhân.
3	AGE	Tuổi của bệnh nhân hoặc người có liên quan đến bệnh nhân.
4	GENDER	Giới tính của bệnh nhân hoặc người có liên quan đến bệnh nhân.
5	JOB	Nghề nghiệp của bệnh nhân hoặc người có liên quan đến bệnh nhân.
6	LOCATION	Địa điểm/nơi ở của bệnh nhân hoặc từng đến.
7	ORGANIZATION	Các tổ chức liên quan đến bệnh nhân: Công ty, Tổ chức Chính phủ, ...
8	TRANSPORTATION	Phương tiện giao thông mà bệnh nhân sử dụng (Chỉ gán số hiệu, biển số, ...)
9	DATE	Tất cả các ngày có trong câu.
10	SYMPTOM_AND_DISEASE	Triệu chứng và bệnh mà bệnh nhân gặp phải.

Bảng 1. Các nhãn thực thể

1.2.3 Thông tin về bộ dữ liệu

Bộ dữ liệu mà bài báo cung cấp bao gồm 10,027 câu được crawl từ các trang báo chính thống của Việt Nam như VnExpress, ZingNews, BaoMoi và ThanhNien. Nội dung chủ yếu từ các bài có tiêu đề liên quan đến #COVID hoặc #COVID-19 trong khoảng thời gian từ Tháng 2 năm 2020 đến Tháng 8 năm 2020.

Bộ dữ liệu sau đó được chia thành các tập training/validation/tests với tỉ lệ 5/2/3 để sử dụng cho quá trình huấn luyện các mô hình máy học.

Nhãn thực thể	Train	Valid.	Test	All
PATIENT-ID	3,240	1,276	2,005	6,521
NAME	349	188	318	855
AGE	682	361	582	1,625
GENDER	542	277	462	1,281
JOB	205	132	173	510
LOCATION	5,398	2,737	4,441	12,576
ORGANIZATION	1,137	551	771	2,459
TRANSPORTATION	226	87	193	3,341
DATE	2,549	1,103	1,654	5,306
SYMPTOM_AND_DISEASE	1,439	766	1,136	506
# Tổng số lượng thực thể	15767	7,478	11,735	34,984
# Tổng số câu	5,027	2,000	3,000	10,027

Bảng 2. Thống kê nhãn thực thể

Cú pháp câu được tổ chức theo định dạng CoNLL (Conference On Natural Language Processing). Mỗi tập dữ liệu chứa các văn bản đã tách từ và gán nhãn. Mỗi từ được đặt trên một dòng riêng biệt và mỗi câu được phân cách nhau bởi một dòng trống. Mỗi dòng bao gồm hai cột, các cột được cách nhau bởi một khoảng trắng.

Mỗi thực thể được biểu diễn dưới dạng cấu trúc BIO [3]: Với mỗi thực thể bao gồm nhiều từ tạo thành (T):

- + Gán nhãn từ bắt đầu của (T) là B-T
- + Các từ tiếp theo là I-T.
- + Không có nhãn: gán 'O'

Word	Anno
Bệnh_nhân	O
669	B-PATIENT-ID
Là	O
Bác_sĩ	B-JOB
Làm	O
Việc	O
Tại	O
Bệnh_viện	B-LOCATION
Đa_Khoa	I-LOCATION
Đồng_Nai	I-LOCATION

Chương 2: CÔNG CỤ GÁN NHÃN BÁN TỰ ĐỘNG

2.1 Giới thiệu

Để hỗ trợ trong việc gán nhãn, nhóm đã xây dựng một công cụ gán nhãn bán tự động đơn giản giúp giảm thời gian thực hiện các thao tác không cần thiết. Đầu tiên là sẽ sử dụng một mô hình máy học và được huấn luyện với dữ liệu đầu vào là dữ liệu đã được gán nhãn của bài báo với số lượng là 1,000 câu (ít hơn so với dữ liệu đã được chia là 5,000), mục đích là để mô hình gán sẵn các nhãn trước sau đó annotator sẽ sửa các nhãn mà cho là gán chưa đúng. Công cụ này được thực hiện trên một file google colab và dữ liệu được lưu trữ trên Google Drive.



Figure 4. Logo Google Colaboratory



Figure 5. Logo Google Drive

Conditional Random Field (CRF) là một mô hình phân lớp sử dụng ngữ cảnh tham gia vào quá trình gán nhãn [4]. Thường được sử dụng trong các bài toán gán nhãn từ loại, nhận diện thực thể tên riêng trong lĩnh vực Xử lý ngôn ngữ tự nhiên. CRF là thuật toán xác suất có điều kiện. Trong CRF, chúng ta cũng xây dựng dự đoán nhãn từ hiện tại theo nhãn của các từ trước đó.

2.2 Mô hình

Để sử dụng mô hình CRF, trong python có hỗ trợ thư viện `sklearn_crfsuite` để sử dụng. Để triển khai công cụ gán nhãn bán tự động, chỉ cần sử dụng thư viện có sẵn và tham số gốc của mô hình.

Code:

```
import sklearn_crfsuite

crf = sklearn_crfsuite.CRF()
```

Dữ liệu mà bài báo cung cấp được đăng công khai trên github nên chỉ cần clone về để lấy dữ liệu.

Code:

```
!git clone https://github.com/VinAIRResearch/PhoNER_COVID19/
```

Chọn bộ train và test để sử dụng cho việc huấn luyện, đo độ chính xác mô hình và sử dụng trong quá trình gán nhãn.

main

PhoNER_COVID19 / data / word /

Go to file

Add file

...

Thinh Truong and Thinh Truong fix annotation errors

c63fa58 on Nov 20

History

..

dev_word.conll

fix annotation errors

last month

test_word.conll

fix annotation errors

last month

train_word.conll

add train/dev/test data on the syllable and word level

13 months ago

Figure 6. Vị trí chứa bộ dữ liệu

Code:

```
dir_train_data = 'PhoNER_COVID19/data/word/train_word.conll'
```

```
dir_test_data = 'PhoNER_COVID19/data/word/test_word.conll'
```

Trước khi đưa dữ liệu vào mô hình, cần phải biến đổi dữ liệu theo format đầu vào để tiến hành huấn luyện và dự đoán.

Đối với mô hình CRF, cần tách phần dữ liệu từ và nhãn ra thành 2 phần.

Phần từ (X_format) sẽ là một danh sách chứa các danh sách các từ của mỗi câu.

Tương tự với (y_format) sẽ là một danh sách chứa các nhãn của mỗi câu tương ứng.

Code:

```
def prepareData(tagged_sentences):
    X,y=[],[]
    for sentences in tagged_sentences:
        X.append([word for word,tag in sentences])
        y.append([tag for word,tag in sentences])
    return X,y

X_train,y_train = prepareData(Train_data[:1000])
X_test,y_test = prepareData(Test_data)
```

Khi huấn luyện mô hình chỉ cần gọi hàm fit cho mô hình CRF với 2 biến X_train, y_train lần lượt là dữ liệu dùng để huấn luyện theo format đã nói trước đó.

Code:

```
crf = CRF()
crf.fit(X_train,y_train)
y_pred = crf.predict(X_test)
```

Thực hiện đánh giá kết quả mô hình dựa trên hàm “classification_report”:

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
AGE	0.57	0.03	0.06	682
DATE	0.94	0.91	0.92	2547
GENDER	0.68	0.48	0.56	542
JOB	0.46	0.21	0.29	205
LOCATION	0.72	0.64	0.67	5398
NAME	0.46	0.11	0.18	349
ORGANIZATION	0.69	0.54	0.61	1137
PATIENT_ID	0.71	0.81	0.76	3240
SYMPTOM_AND_DISEASE	0.59	0.15	0.23	1439
TRANSPORTATION	0.99	0.63	0.77	226
micro avg	0.75	0.62	0.68	15765
macro avg	0.68	0.45	0.51	15765
weighted avg	0.72	0.62	0.64	15765

Figure 7. Kết quả đánh giá công cụ

Dựa vào kết quả micro-F1-Score, có thể thấy khi sử dụng 1000 câu để làm dữ liệu cho quá trình huấn luyện thì cho ra kết quả phù hợp cho quá trình gán nhãn bán tự động.

2.3 Giao diện công cụ

Để thuận tiện cho việc gán nhãn và tính toán kết quả, nhóm xây dựng một số tính năng như:

- ✓ Chọn câu để gán
- ✓ Chỉnh sửa nhãn
- ✓ Lưu kết quả.

Khi chọn câu để gán nhãn, annotator sẽ chọn câu cần gán theo kế hoạch ở mỗi set. Kết quả trả về sẽ là nội dung câu, danh sách bảng chứa số thứ tự, từ và nhãn được mô hình gán.

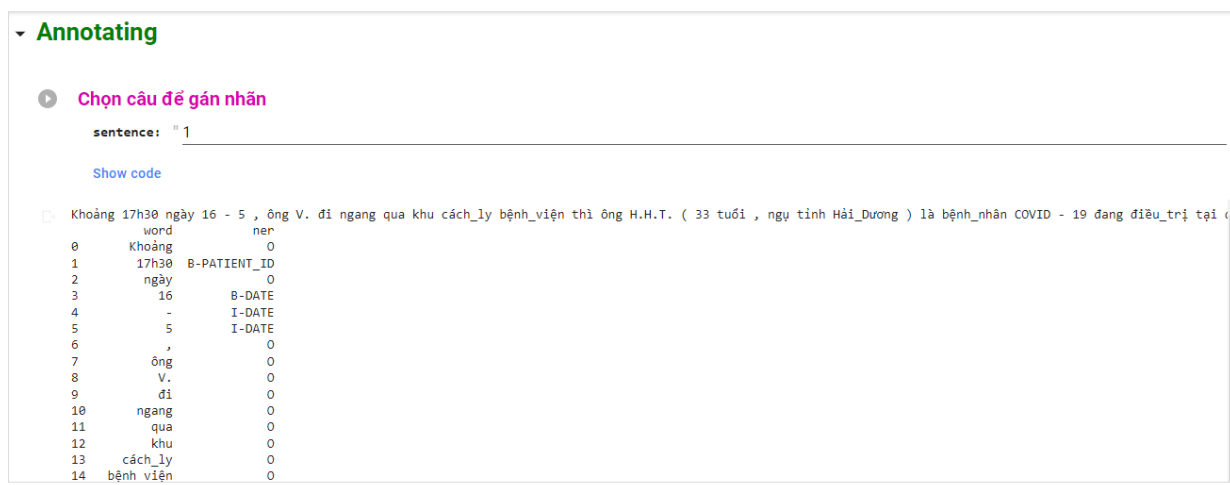


Figure 8. Ảnh minh họa chọn câu gán

Tiếp theo, trong trường hợp nhãn do mô hình gán sai ở vị trí nào thì annotator chỉ cần chọn vị trí và nhãn thích hợp để gán. Mỗi lần thực thi sẽ hiển thị kết quả để theo dõi.

Ghi chú:

- ☒ Index (nhập): Vị trí cần được sửa nhãn theo kết quả trả về ở bước trước.
- ☒ Pre (lựa chọn): Chọn 1 trong 2 nhãn “B-” và “I-”.
- ☒ Ner (lựa chọn): Gồm các nhãn có trong nội dung: Name, Age, Patient-ID, Job, Location, Transportation, Date, Gender, Symptom-and-Disease, Organization.
- ☒ O_label (tích chọn): Nhãn “O”.

Ví dụ: Nếu cần gán nhãn vị trí 14 là “I-NAME” thì chọn các tham số lần lượt là “14”, “I-“, “NAME”. O_label bỏ trống. Còn trường hợp cần gán từ là nhãn “O” thì chỉ cần tích chọn O_label và không cần quan tâm đến các tham số trên.

Điều chỉnh nhãn

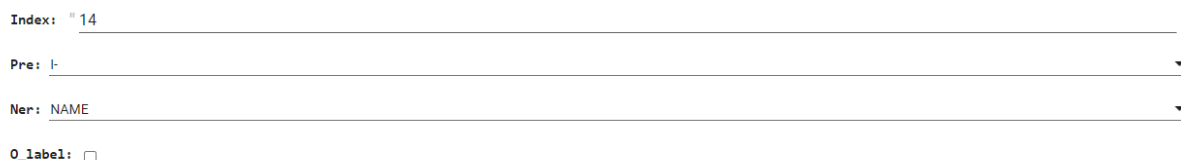


Figure 9. Ảnh minh họa điều chỉnh nhãn

Cuối cùng, annotator sẽ lưu lại kết quả câu vừa gán theo từng vòng vào google drive.

Ghi chú:

- ☒ Round (chọn): Số vòng đang thực hiện gán nhãn (1-10).
- ☒ Annotator_ID: Tùy theo số lượng Annotator có thể tùy chỉnh (1-2). Dùng để phân biệt kết quả khi lưu trữ.



Figure 10. Ảnh minh họa Lưu kết quả

Sau khi nhãn của câu được lưu lại, tệp sẽ có tên theo định dạng:

Annotator_<1 hoặc 2>_sentence_<số thứ tự câu được gán>.txt

My Drive > CS321_PROJECT > SET 1

Name ↑	Owner	Last modified	File size
annotator_1_sentence_1.txt	me	Nov 27, 2021 me	588 bytes
annotator_1_sentence_2.txt	me	Nov 27, 2021 me	675 bytes
annotator_1_sentence_3.txt	me	Nov 27, 2021 me	669 bytes
annotator_1_sentence_4.txt	me	Nov 27, 2021 me	1 KB
annotator_1_sentence_5.txt	me	Nov 27, 2021 me	698 bytes
annotator_2_sentence_1.txt	me	Nov 27, 2021 me	588 bytes

Figure 11. Thư mục chứa các file gán nhãn

Các kết quả gán nhãn mỗi vòng của từng annotator sẽ được lưu lại trong mỗi thư mục tương ứng phục vụ cho việc lưu trữ.

My Drive > CS321_PROJECT

Folders	Name ↑
Data	Report
SET 1	SET 2
SET 3	SET 4
SET 5	SET 6
SET 7	SET 8
SET 9	SET 10
SET 10 (Bán tự động)	Set 10 (Thủ công)

Figure 12. Các thư mục

Để đánh giá công cụ gán nhãn này có tốt hay không thì nhóm thực hiện so sánh quá trình gán nhãn thủ công và bán tự động đó là:

- Sử dụng 5 câu của set 10 để thực hiện gán nhãn bằng 2 phương pháp.
- Đo thời gian thực hiện gán nhãn.
- So sánh thời gian thực hiện trung bình của 2 phương pháp.

Câu	Thủ công		Bán tự động	
	Annotator 1	Annotator 2	Annotator 1	Annotator 2
1	64	191	41	31
2	148	171	36	41
3	81	133	48	46
4	42	173	52	85
5	38	81	31	24
Tổng thời gian (s)	6'13	12'29	3'28	3'47

Bảng 3. Kết quả so sánh giữa hai phương pháp

Kết quả là thời gian gán nhãn giảm 2 lần so với annotator 1 và 4 lần so với annotator 2.

Chương 3: QUY TRÌNH THỰC HIỆN

3.1 Chuẩn bị dữ liệu

Nhóm sử dụng 50 câu trong tập test với 2 quy tắc là:

- ✓ Annotator chỉ biết được nội dung của câu.
- ✓ Nhãn của câu sẽ được dùng làm nhãn gold để đo độ chính xác.

3.2 Huấn luyện Annotators

Quá trình huấn luyện annotator sẽ được thực hiện trong 10 vòng gồm các bước:

- ☒ Đọc guideline
- ☒ Thực hiện gán nhãn bằng tool (2 annotator)
- ☒ Đo độ chính xác, đồng thuận
- ☒ Thảo luận

Bước 1: Đọc guideline

- ✓ Cần hiểu rõ cách gán từng loại nhãn trong từng ngữ cảnh cụ thể.
- ✓ Ghi chú các nội dung chưa rõ và thảo luận chung với nhóm.

Bước 2: Thực hiện gán nhãn

- ✓ 2 Annotators bắt đầu gán nhãn 5 câu ở mỗi vòng.
- ✓ Sử dụng công cụ đã xây dựng để thực hiện gán nhãn.

Bước 3: Đo độ chính xác, độ đồng thuận

- ✓ Dựa vào kết quả gán nhãn của 2 Annotators, tiến hành đo độ chính xác, độ đồng thuận.
- ✓ Khi độ chính xác, độ đồng thuận đều đạt trên 90% thì sẽ kết thúc quá trình huấn luyện.

Lưu ý: Trong nội dung môn học, số lượng câu sử dụng không được nhiều nên sẽ có các vòng đạt trên 90% trước vòng thứ 10 vì tùy vào chất lượng

của dữ liệu sử dụng trong quá trình gán nhãn (các câu ít nhãn thực thể, ít nhập nhằng về ngữ nghĩa).

Bước 4: Thảo luận

- ✓ So sánh nhãn gold với nhãn gán của từng annotator.
- ✓ Tìm hiểu lý do gán sai nhãn của từng annotator.
- ✓ Đưa ra hướng giải quyết cho từng trường hợp cụ thể.

3.3 Phương pháp đánh giá

Để đo độ chính xác và độ đồng thuận, nhóm sử dụng độ đo **F1-SCORE** để đánh giá.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Trong đó:

- **Accuracy** (Độ chính xác) :tính trên mỗi annotator, thông qua nhãn gold và nhãn của từng annotator gán.
- **Agreement** (Độ đồng thuận): được tính thông qua nhãn do 2 annotator gán.

Code:

```
from seqeval.metrics import f1_score  
a = f1_score(y_gold,y_annotator_1)  
b = f1_score(y_gold,y_annotator_2)  
c = f1_score(y_annotator_1,y_annotator_2)
```

Trong file colab (F1-SCORE) đã xây dựng sẵn công thức tính toán độ đo này, chỉ cần bổ sung thêm một file kết quả gán nhãn vào mỗi folder Set tương ứng của mỗi vòng để có thể tính toán.

[16] **Đánh giá**

round: 2

[Show code](#)

	Metric	Annotator 1	Annotator 2	Average
0	Accuracy	97.1429	97.1429	97.142857
1	Agreement	-	-	100.000000

Figure 13. Ảnh minh họa kết quả đo

Sau mỗi tính toán kết quả, có thể hiển thị kết quả gán nhãn của từng câu, phục vụ cho việc bàn luận và biết được nhãn nào gán đúng, gán sai.

[19] **Hiển thị kết quả**

sentence_: 3

[Show code](#)

	word	ner_gold	ner_a1	ner_a2
0	Tuy_nhiên	0	0	0
1	,	0	0	0
2	bệnh_nhân	0	0	0
3	chỉ	0	0	0
4	được	0	0	0
5	phép	0	0	0
6	duy_trì	0	0	0
7	tư_thể	0	0	0
8	này	0	0	0
9	trong	0	0	0
10	vòng	0	0	0
11	16	0	0	0
12	tiếng	0	0	0

Figure 14. Ảnh minh họa hiển thị kết quả

3.4 Kết quả

Vòng	Độ chính xác		Độ đồng thuận
	Annotator 1	Annotator 2	
1	93.48%	91.3%	97.87%
2	97.14%	97.14%	100%
3	87.5%	91.43%	85.71%
4	100%	100%	100%
5	84.85%	94.12%	91.42%
6	85.71%	96.55%	88.88%
7	90.47%	100%	90.47%
8	93.1%	91.52%	91.22%
9	91.42%	100%	91.42%
10	91.43%	91.43%	100%

Bảng 4. Kết quả đo độ chính xác, độ đồng thuận

Trong 10 vòng thì cơ bản từ vòng 7 trở đi độ chính xác và độ đồng thuận của cả hai đã đạt trên 90%. Do số lượng câu dùng ít nên số lượng lỗi sai ở các vòng đầu sẽ không đánh giá được toàn diện, đặc biệt là ở vòng 4 đạt 100% vì các câu không có sự nhập nhằng về nghĩa, các lỗi sai ban đầu cũng được khắc phục ở các vòng sau. Kết quả cuối cùng vòng 10 với độ chính xác là 91.43% và độ đồng thuận đạt 100%.

➤ **Phân tích một số kết quả trong quá trình thực hiện gán nhãn**

🚦 **Vòng 1:**

“Tính đến ngày 30 - 7 , Việt_Nam có thêm một_số ca bệnh nặng đang điều_trị tại các bệnh_viện Đà_Nẵng , đặc biệt là các bệnh_nhân tiên_lượng rất nặng: bệnh_nhân 416, 418, 428, 431, 436, 437, 438...”

Ở trường hợp này, hai annotator đều gán nhãn đúng các thực thể:

DATE: 30-7

LOCATION: Việt Nam

PATIENT-ID: 416, 418, ...

Tuy nhiên có một trường hợp gán nhãn sai là “bệnh_viện Đà_Nẵng”, theo nhãn gold thì chỉ gán “Đà_Nẵng” là **LOCATION** vì trước cụm từ trên có từ “các” – nơi chốn không rõ ràng. Kết quả gán nhãn của cả hai annotator ở trường hợp này là “bệnh_viện Đà_Nẵng” với **LOCATION**. Với trường hợp này thì cả 2 annotator sẽ đọc lại guideline phần **LOCATION** để rõ hơn cách gán phần này.

🚦 **Vòng 2:**

“Tuy_nhiên , bệnh_nhân chỉ được phép duy_trì tư_thể này trong vòng 16 tiếng mỗi ngày , nếu không sẽ bị loét điễm tì”

Ở trường hợp này không chứa thực thể nào, tuy nhiên cả hai annotator đều gán cụm từ “loét điễm tì” là **SYMPTOM-AND-DISEASE**. Nguyên nhân là do không hiểu nghĩa

CS321.M11.KHCL – NGÔN NGỮ HỌC NGỮ LIỆU

của từ (do bệnh này không liên quan đến triệu chứng, di chứng hay hậu di chứng liên quan đến COVID-19). Để tránh lặp lại lỗi này thì chỉ cần nhắc nhở hai annotator cần hiểu được ý nghĩa của các từ chưa rõ trong câu.

Vòng 3:

“Lần thứ 4 hiến máu tại bệnh_viện , **nữ hộ_sinh** Nguyễn_Việt_Dung , khoa Phụ ngoại , cho biết lần này có khác_biệt khi chị và các đồng_nghiep đều phải đeo khẩu_trang đi hiến máu .”

Ở trường hợp này, vì người có tên “**Nguyễn_Việt_Dung**” trong câu không có liên quan đến bệnh nhân COVID-19 hay lịch trình di chuyển của bệnh nhân nên sẽ không gán nhãn Nghề nghiệp (JOB) cho từ bỏ trợ trước đó. Tuy nhiên cả hai annotator đều gán sai với hai cách khác nhau:

- Annotator 1 (nữ hộ_sinh, B-GENDER B-JOB)
- Annotator 2 (nữ hộ_sinh, B-JOB I-JOB)

Nguyên nhân là cả hai chưa đọc kỹ toàn bộ câu để loại bỏ các trường hợp theo yêu cầu chung của guideline (Chủ thể không liên quan đến bệnh nhân COVID-19). Giải quyết việc này thì cả 2 annotator cần chú ý đọc cả câu để nắm rõ ngữ cảnh.

Vòng 8:

“**UBND TP Biên_Hòa , Đồng_Nai** dỡ bỏ cách_ly tuyến đường Hồ_Văn_Đại , nơi cư_trú của hai bệnh_nhân 595 và 669 sau 14 ngày phong_toả .

Ở trường hợp này, “**UBND TP Biên_Hòa**” sẽ được gán là **ORGANIZATION**, và “**Đồng_Nai**” sẽ được gán là **LOCATION**. Cả hai annotator đều gán nguyên cụm từ trên là một thực thể **ORGANIZATION**. Nguyên nhân: “TP Biên Hòa” và “Đồng Nai” là tên của 2 địa phương cụ thể cùng bổ nghĩa cho UBND nhưng các trường hợp có trong guidelines có điểm chung là chỉ gán tổ chức đi kèm với tên của 1 địa phương cụ thể. Để giải quyết vấn đề này, nhóm đã thảo luận và ghi chú thêm vào guidelines rằng “*khi có nhiều hơn 1 tên địa phương cùng bổ nghĩa cho cơ quan ở cấp độ địa phương thì gán cơ quan và tên địa phương đứng cạnh là một thực thể ORGANIZATION hoàn chỉnh*”.

Một số trường hợp cần chú ý khác:

Bốn người “ngoài **VN0054**” gồm N.T.T, nữ 24 tuổi từ Anh về nước ngày 9-3 trên máy bay thuê riêng, người nhà và lái xe của bệnh nhân số 17, nam bệnh nhân 27 tuổi đi từ Hàn Quốc về trên chuyến bay VJ981 ngày 4-3.”

“VN0054” là số hiệu chuyến bay nhưng đi kèm với từ phủ định “ngoài” nên trong câu này, “VN0054” không phải là phương tiện di chuyển của bệnh nhân vì vậy không gán nhãn “VN0054” là **TRANSPORTATION**.

Trước đó, vào sáng 19-9, **Bệnh viện Phổi Đà Nẵng** đã chuyển hai bệnh nhân COVID-19 đã được công bố khỏi bệnh viện trước đó nhiều ngày là bệnh nhân số 416 và BN 478 về **Bệnh viện Đà Nẵng** để tiếp tục điều trị bệnh nền.

"Bệnh viện Phổi Đà Nẵng" đóng vai trò là chủ ngữ của câu, thực hiện hành động là chuyển hai bệnh nhân nên gán nhãn **ORGANIZATION**. "Bệnh viện Đà Nẵng" là nơi tiếp nhận bệnh nhân, được dùng như một địa điểm nên gán nhãn **LOCATION**.

Lãnh đạo huyện Chương Mỹ cho biết trên địa bàn huyện có một người tiếp xúc với bệnh nhân thứ 17, đó là chị D.T.T, đã tiếp xúc với bệnh nhân ngày 2 và ngày 4-3.

"Lãnh đạo huyện Chương Mỹ" chưa được định danh rõ ràng nên chỉ gán "huyện Chương Mỹ" là **LOCATION**.

Chương 4: KẾT LUẬN

Để có thể hoàn thành các nội dung của đề án, nhóm đã tìm hiểu về bài toán Nhận dạng Thực thể Tên riêng để tiến hành nghiên cứu bài báo “**COVID-19 NAMED ENTITY RECOGNITION FOR VIETNAMESE**” nhằm tìm hiểu về kho ngữ liệu, guidelines, mô hình máy học và các công cụ cần thiết để hoàn thành mục tiêu đề án.

Giai đoạn tìm hiểu bài toán, các thành viên rút ra và tổng hợp các kiến thức nền tảng về bài toán Nhận dạng Thực thể Tên riêng cùng với thông tin sau khi nghiên cứu bài báo, guidelines và kho ngữ liệu cũng như các công cụ hỗ trợ có sẵn. Ở công đoạn kế tiếp, nhóm đã tìm hiểu và tiến hành huấn luyện công cụ gán nhãn bán tự động để phục vụ cho việc huấn luyện annotator tiếp sau đó. Đến với công đoạn cuối cùng, sau khi chuẩn bị dữ liệu thì các annotator sẽ tiến hành gán nhãn thủ công, bán tự động rồi thực hiện so khớp dựa trên độ chính xác và độ đồng thuận để thảo luận để đưa ra đánh giá khách quan về các trường hợp gán nhãn dữ liệu.

Thông qua quá trình hiện thực đề án, nhóm rút ra được các kinh nghiệm làm việc thực tiễn về quy trình phát triển một kho ngữ liệu hoàn chỉnh cho bài toán Nhận dạng Thực thể Tên riêng xét trong bối cảnh đại dịch COVID-19. Từ đó, nghiệm thu các phương pháp đánh giá công cụ gán nhãn, annotator và kho ngữ liệu bằng việc thử nghiệm, khảo sát và rút ra kết luận về quá trình nghiên cứu cá nhân và tiến hành thảo luận nhóm định kỳ trên toàn bộ quá trình.

TÀI LIỆU THAM KHẢO

- [1] Dong Gao, Lanfei Peng, and Yujie Bai, “HAZOP Text Named Entity Recognition using CNN-BiLSTM-CRF Model,” *2020 Chinese Automation Congress (CAC)*, pp. 6159–6164, Nov. 2020, doi: 10.1109/CAC51589.2020.9327702.
- [2] Thinh Hung Truong, Mai Hoang Dao, and Dat Quoc Nguyen, “COVID-19 Named Entity Recognition for Vietnamese,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, Jun. 2021, pp. 2146–2153. doi: 10.18653/v1/2021.naacl-main.173.
- [3] Erik F. Tjong Kim Sang and Jorn Veenstra, “Representing Text Chunks,” *arXiv:cs/9907006*, Jul. 1999, Accessed: Dec. 29, 2021. [Online]. Available: <http://arxiv.org/abs/cs/9907006>
- [4] John Lafferty, Andrew McCallum, and Fernando C N Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” p. 10.