

Đại học Quốc Gia Hà Nội  
Trường Đại Học Công Nghệ  
Khoa Công Nghệ Thông tin

\* \* \*



Báo Cáo Khai Phá Dữ Liệu  
(INT3209\_3)

Trích xuất bộ các thành phần so sánh từ nhận xét  
đánh giá sản phẩm của khách hàng

Sinh viên thực hiện

Nguyễn Quý Nghĩa - 21020367  
Cần Minh Nghĩa - 21020082  
Bùi Tuấn Nghĩa - 21020364  
Hoàng Nhật Minh - 21020354

Giảng viên: TS.Lê Hoàng Quỳnh

Hà Nội, ngày 26, tháng 11, năm 2023

**Đóng góp:**

Nguyễn Quý Nghĩa (nhóm trưởng): 30%

Bùi Tuấn Nghĩa: 30%

Cần Minh Nghĩa: 30%

Hoàng Nhật Minh: 10%

**Phân công nhiệm vụ:**

Nguyễn Quý Nghĩa: model

Bùi Tuấn Nghĩa: model

Cần Minh Nghĩa: DA

Hoàng Nhật Minh: DA

Phân công nhiệm vụ trên chỉ là tương đối để chia phần trong lúc làm slide và báo cáo, còn trên thực tế cả 3 thành viên Nghĩa đều cùng làm DA và model để hiểu dữ liệu và mô hình.

<b>Lời mở đầu</b>	<b>2</b>
<b>1. Bài toán</b>	<b>3</b>
1.1. Giới thiệu bài toán	3
1.2. Mô hình bài toán	4
<b>2. Bộ dữ liệu</b>	<b>4</b>
2.1. Cấu trúc dữ liệu đầu vào	4
2.2. Thống kê	5
2.3. Khó khăn	7
2.4. Phương hướng giải quyết	8
<b>3. Phương pháp</b>	<b>8</b>
3.1 Phân loại câu so sánh và trích xuất thành phần so sánh.	10
3.2 Kết hợp, lọc và phân loại bộ các thành phần so sánh	11
<b>4. Đánh giá hiệu năng mô hình</b>	<b>13</b>
4.1 Thang đo	13
4.2 Tối ưu mô hình	13
4.3 Kết quả	14
<b>5. Phân tích mô hình</b>	<b>15</b>
<b>6. Định hướng phát triển</b>	<b>16</b>
<b>7. Tham chiếu</b>	<b>16</b>

# Lời mở đầu

Trích xuất các thành phần so sánh trong câu là một tác vụ quan trọng trong trích xuất ý kiến khách hàng. Hướng đến việc xác định những từ, câu, từ những đánh giá về sản phẩm của người dùng mà có mang hàm ý so sánh các chủ thể với nhau.

Những nghiên cứu trước đây đơn thuần thường trích xuất các thành phần so sánh mà không quan tâm đến việc một câu văn có thể có nhiều các vế so sánh riêng biệt. Do có thể có nhiều vế so sánh trong câu nên các nghiên cứu trước chỉ tập trung vào việc trích xuất các thành phần so sánh mà không nghĩ đến việc kết hợp và kiểm tra lại các bộ đã được kết hợp điều này dẫn đến việc nhiều bộ so sánh bị kết hợp sai, chồng chéo với nhau.

Trong bài báo các nhà nghiên cứu đã xác định và giải quyết những vấn đề còn tồn đọng ở những nghiên cứu trước đó và đã giới thiệu một mô hình Comparative Opinion Quintuple Extraction (COQE) để giải quyết các vấn đề này.

Trong báo cáo này mô hình COQE được sử dụng để huấn luyện trên bộ dữ liệu mới được ban tổ chức cuộc thi VLSP ComOM shared task đưa ra. Nhiệm vụ yêu cầu từ ban tổ chức cuộc thi gần giống với tác vụ trích xuất bộ các thành phần so sánh đã được đề cập ở trên. Mô hình thể hiện khá tốt trên rất nhiều độ đo và được xếp hạng thứ 7 chung cuộc. Cải thiện đáng kể trên nhiều độ đo với các mô hình khác.

# 1. Bài toán

## 1.1. Giới thiệu bài toán

Sự phát triển nhanh chóng của mua sắm trực tuyến và các nền tảng thương mại điện tử đã dẫn đến một cơn bão của đánh giá sản phẩm. Những đánh giá này thường chứa đựng thông tin quý giá về ý kiến của người dùng về các khía cạnh khác nhau của sản phẩm, bao gồm so sánh giữa các thiết bị khác nhau. Hiểu đúng đắn ý kiến so sánh từ những đánh giá sản phẩm là quan trọng đối với cả nhà sản xuất và người tiêu dùng. Nhà sản xuất có thể đạt được cái nhìn sâu sắc về điểm mạnh và điểm yếu của sản phẩm so với đối thủ, trong khi người tiêu dùng có thể đưa ra quyết định mua sắm có thông tin cơ bản hơn dựa trên những cái nhìn so sánh này.

Bài toán đặt ra ở đây là trích xuất đánh giá của khách hàng và xác định ý kiến so sánh của người đánh giá, tức là liệu họ đánh giá sản phẩm là tốt hơn, xấu hơn, hoặc khác biệt so với các sản phẩm khác.

Với bài toán như vậy, yêu cầu được đặt ra là:

- Đầu tiên cần phân loại được câu đánh giá đó là câu so sánh hay không so sánh.
- Với những câu so sánh, cần trích xuất được các đối tượng so sánh, khía cạnh so sánh, từ hoặc cụm từ so sánh, từ đó phân loại câu so sánh đó (so sánh hơn, so sánh bằng,...).

## 1.2. Định nghĩa bài toán

Input output bài toán:

**Đầu vào:** Bộ dữ liệu gồm các câu review bằng tiếng việt về sản phẩm

**Đầu ra:** Mô hình có khả năng phân loại câu so sánh và trích xuất được quintuple từ câu đầu vào

Input output mô hình:

**Đầu vào:** Bộ dữ liệu không gán nhãn về các ý kiến đánh giá bằng tiếng việt

**Đầu ra:** Các câu so sánh và trích xuất các bộ quintuple có trong câu

## 2. Bộ dữ liệu

### 2.1. Cấu trúc dữ liệu đầu vào

Ví dụ về 1 câu dữ liệu đầu vào:

*“Về ngoại hình thì cũng có rất ít sự khác biệt so với thể hệ tiền nhiệm. Về ngoại hình thì cũng có rất ít sự khác biệt so với thể hệ tiền nhiệm.”*

Nhiệm vụ của mô hình là phải trích xuất ra được 1 quintuple:

```
{"subject": [], "object": ["14&&thể", "15&&hệ", "16&&tiền", "17&&nhiệm"],  
"aspect": ["2&&ngoại", "3&&hình"], "predicate": ["7&&rất", "8&&ít", "9&&sự",  
"10&&khác", "11&&biệt"], "label": "EQL"}
```

Ở đây, "subject" và "object" tham chiếu đến các thực thể chủ thể và đối tượng đang được so sánh, "aspect" đề cập đến khía cạnh so sánh (tức là thuộc tính đặc trưng) của các thực thể, "predicate" đề cập đến ý kiến so sánh, đó là biểu đạt ý kiến về sự ưu tiên so sánh giữa hai thực thể, và "label" là ý kiến so sánh, chỉ định liệu "sub" có tệ hơn, bằng, tốt hơn hay khác biệt so với "obj".

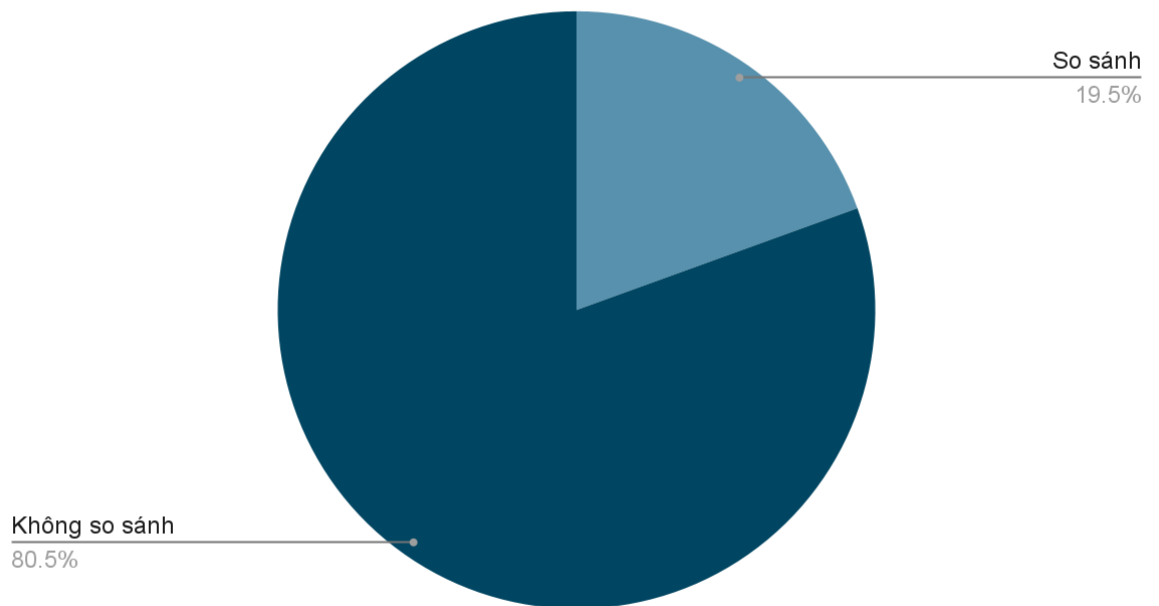
### 2.2. Thống kê

- Tỷ lệ câu so sánh và không so sánh

	Train	Dev
So sánh	812	349
Không so sánh	3359	1382

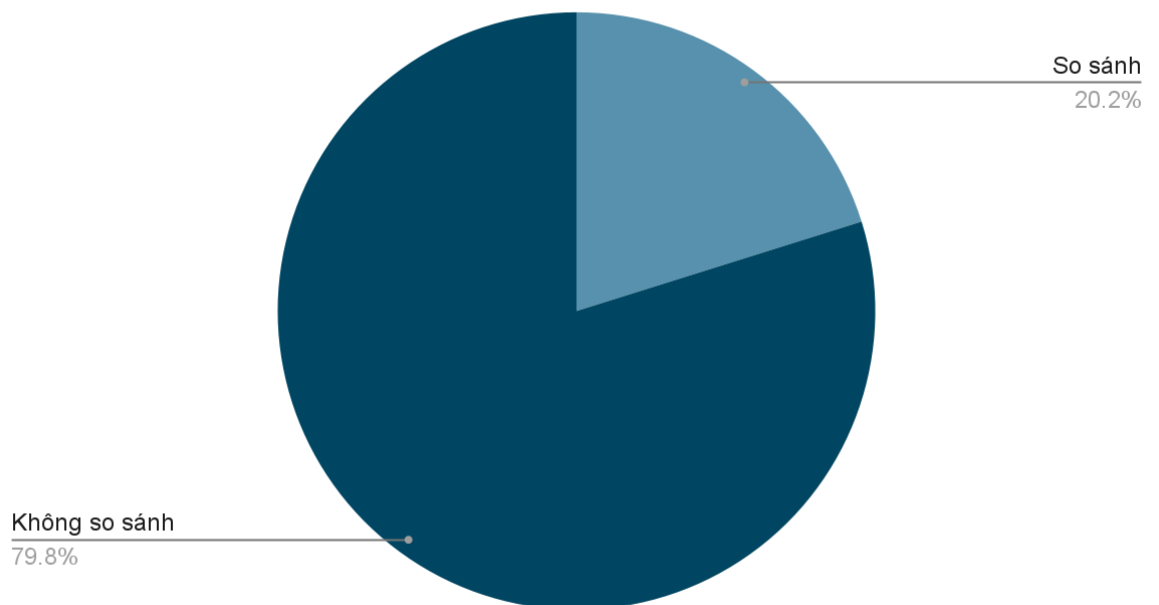
Bảng 1: Thống kê số câu so sánh và không so sánh

## Points scored



Biểu đồ 1: Tỷ lệ các câu so sánh và không so sánh trong tập train

## Points scored



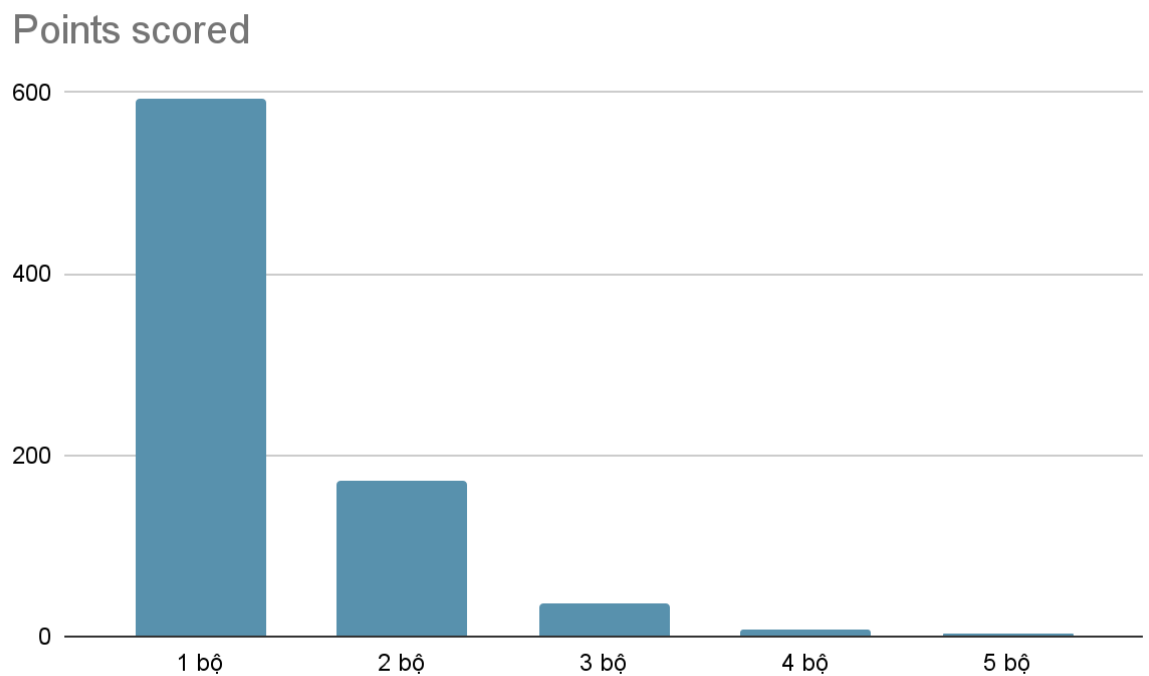
Biểu đồ 2: Tỷ lệ các câu so sánh và không so sánh trong tập dev

Nhận xét: Số lượng câu không so sánh khá lớn, gấp ~ 4 lần số câu so sánh

- Số lượng bộ 5 trong câu

	Train	Dev
1 bộ	594	248
2 bộ	172	82
3 bộ	36	17
4 bộ	7	2
5 bộ	3	0

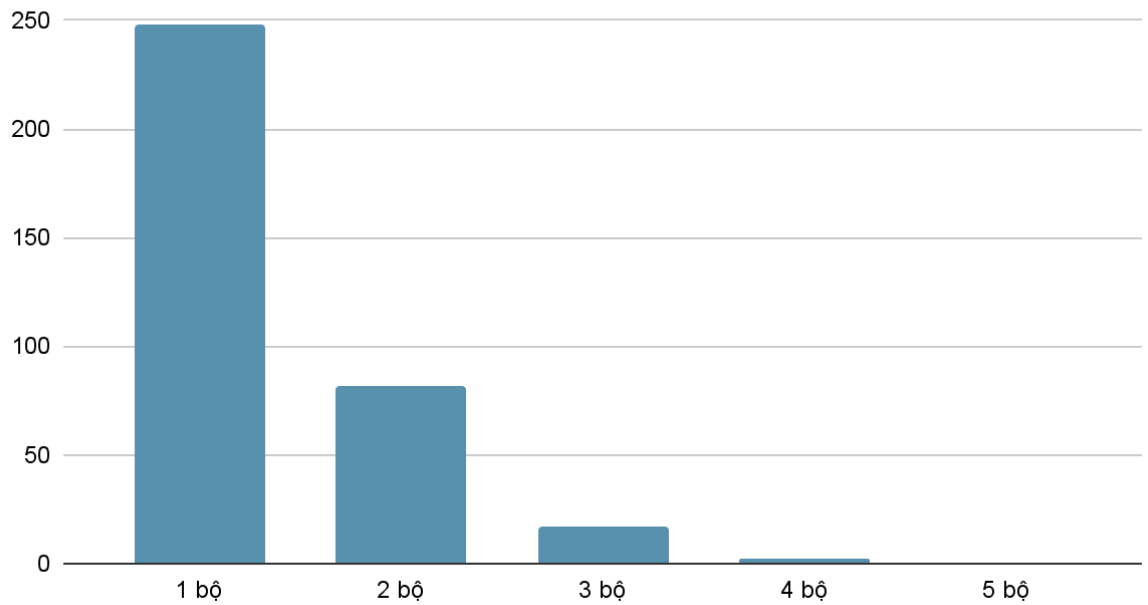
Bảng 2: Thống kê số lượng bộ 5 trong câu



Biểu đồ 3: Số lượng các bộ 5 trong câu trong tập train



Points scored



Biểu đồ 4: Số lượng các bộ 5 trong câu trong tập dev

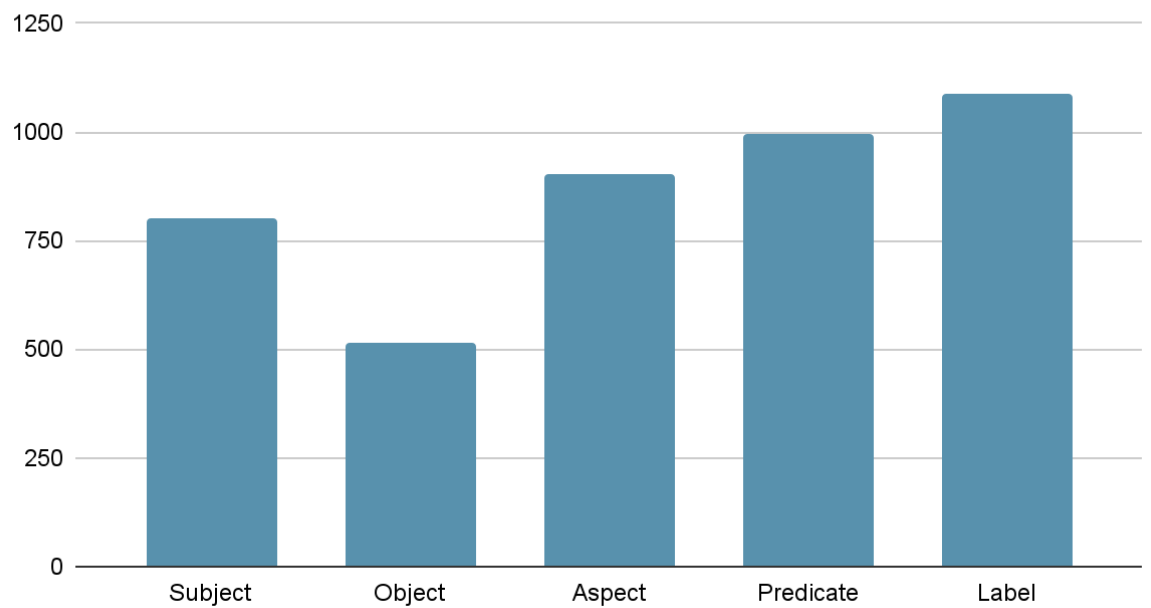
Nhận xét: Đa số các câu chỉ có 1 đến 2 bộ 5

- Số lượng các thành phần trong bộ 5

	Train	Dev
subject	803	353
object	516	263
aspect	901	385
predicate	996	442
label	1089	471

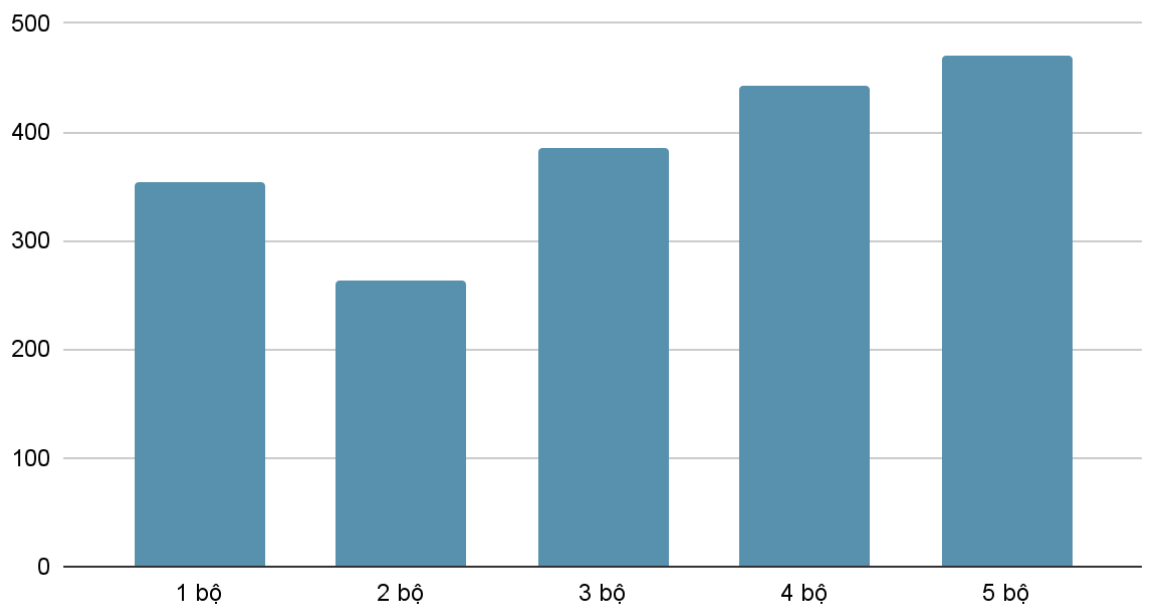
Bảng 3: Số lượng các thành phần trong bộ 5

Points scored



Biểu đồ 5: Số lượng các thành phần trong bộ 5 trong tập train

Points scored



Biểu đồ 6: Số lượng các thành phần trong bộ 5 trong tập dev

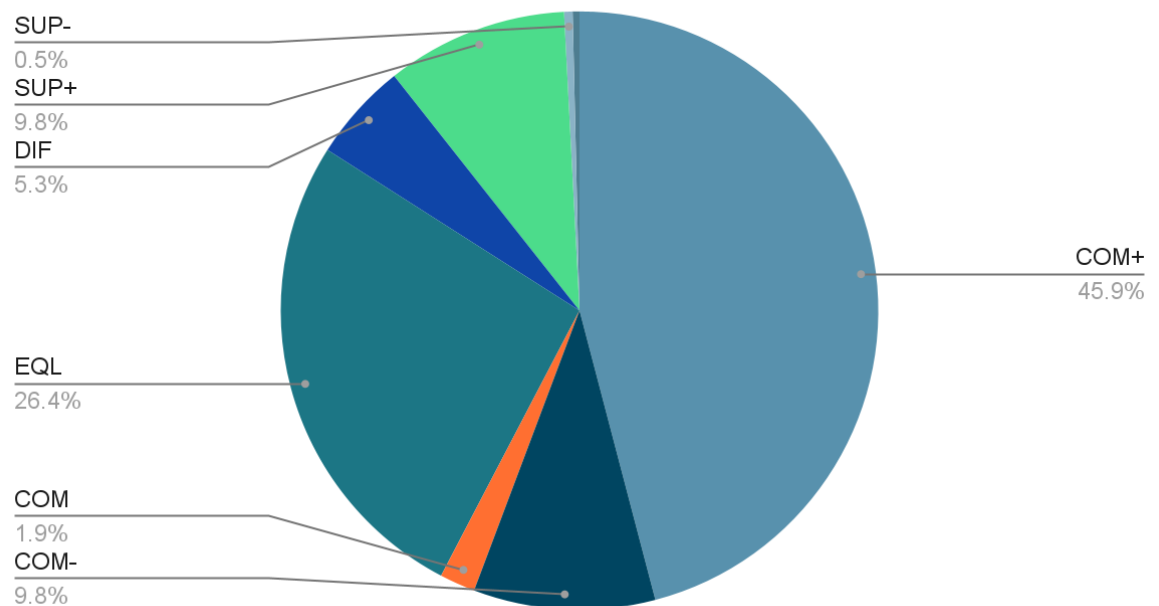
- Tỷ lệ các nhãn

	Train	Dev
--	-------	-----

COM+	500	203
COM-	107	51
COM	21	18
EQL	287	133
DIF	58	33
SUP+	107	28
SUP-	5	1
SUP	4	4

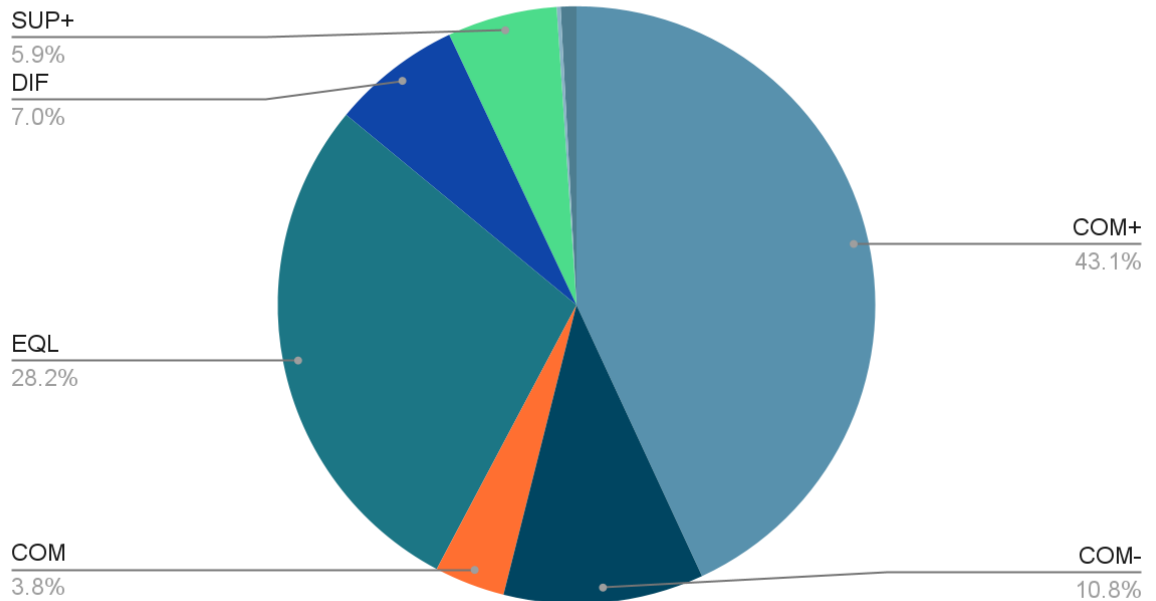
Bảng 4: Thống kê số lượng các nhãn

### Points scored



Biểu đồ 7: Tỷ lệ giữa các nhãn trong tập train

## Points scored



Biểu đồ 8: Tỷ lệ giữa các nhãn trong tập dev

### - Quintuple khuyết thành phần

Tập/Thi ếu	Subject	Object	Aspect	Predicat e	Label	Ít nhất 1
Train	128	468	154	0	0	605
Dev	53	154	68	0	0	223

Có tới 55% câu trong tập train và 47% câu trong tập dev quintuple không đủ thành phần.

Ví dụ:

*“Đi kèm hệ thống camera sánh ngang và thậm chí đánh bại Galaxy trong một số trường hợp.”*

Quintuple của câu đó là:

`{"subject": [], "object": ["13&&Galaxy"], "aspect": ["3&&hệ", "4&&thống", "5&&camera"], "predicate": ["6&&sánh", "7&&ngang"], "label": "EQL"}`

Ở đây câu đánh giá của khách hàng không có phần subject.

### - Quintuple có subject và object trùng nhau

129 các quintuple trong tập train (chiếm tỉ lệ 12%) và 63 các quintuple trong tập dev (chiếm tỉ lệ 13%) có subject trùng với object

Ví dụ:

*“So với iPhone 13, thì rất khó để nhận ra sự khác biệt giữa 2 thế hệ này qua vẻ bề ngoài, thậm chí ngay cả những phần mềm bên trong cũng không có quá nhiều sự khác biệt.”*

Quintuple của câu đó là:

```
{ "subject": ["16&&2", "17&&thế", "18&&hệ", "19&&này"], "object":  
["16&&2", "17&&thế", "18&&hệ", "19&&này"], "aspect": ["21&&về",  
"22&&bề", "23&&ngoài"], "predicate": ["8&&khó", "9&&để", "10&&nhận",  
"11&&ra", "12&&sự", "13&&khác", "14&&biệt"], "label": "EQL" }
```

Ở trong câu trên subject và object là trùng nhau, đều là “2 thế hệ này”.

## 2.3. Khó khăn

- Khó khăn đầu tiên với bộ dữ liệu trên là việc mất cân bằng dữ liệu. Số câu không so sánh nhiều hơn đáng kể số câu so sánh, nhãn COM+ và nhãn EQL- chiếm đa số so với các nhãn khác (cụ thể số nhãn trên chiếm 55,73% trên tổng số nhãn). Nếu có sự mất cân bằng giữa câu so sánh và câu không so sánh, mô hình có thể trở nên chệch về một trong hai hướng. Điều này có thể làm cho mô hình ít nhạy bén hơn đối với lớp thiểu số, làm giảm khả năng phát hiện và đánh giá đúng các câu so sánh. Mặt khác, trong trường hợp sự mất cân bằng giữa các nhãn trong các câu so sánh, mô hình có thể trở nên quá tập trung vào dự đoán nhãn phổ biến hơn. Điều này có thể làm giảm khả năng phân loại đúng đối với các trường hợp ít phổ biến.
- Khó khăn thứ hai là dữ liệu chứa nhiều câu không hoàn chỉnh làm nhiễu: những câu chứa các thành phần alt, des, title, link ảnh,... Tỷ lệ là câu so sánh của các câu này là rất thấp so với các câu khác trong tập dữ liệu. Với 1000 câu bắt đầu bằng alt, title, des, chỉ có khoảng 20 câu là câu so sánh, trong khi với tập dữ liệu 1000 câu phải đc 200 câu so sánh. Điều này có thể gây nhiễu cho mô hình.
- Khó khăn thứ ba là vấn đề có nhiều quintuple trong câu, cấu trúc quintuple khuyết, trùng thành phần có thể gây khó cho mô hình. Nếu trong câu có nhiều quintuple các mô hình trước đưa ra chỉ nhằm trích xuất các thành phần so sánh mà ko nghĩ tới việc một câu có nhiều về so sánh riêng biệt nên các mô hình này thường thất bại trong việc kết

hợp và cấu tạo nên các bộ so sánh. Từ đó dẫn đến kết quả kém khi xác định ý kiến so sánh của người dùng hay comparative preference.

## 2.4. Phương hướng giải quyết

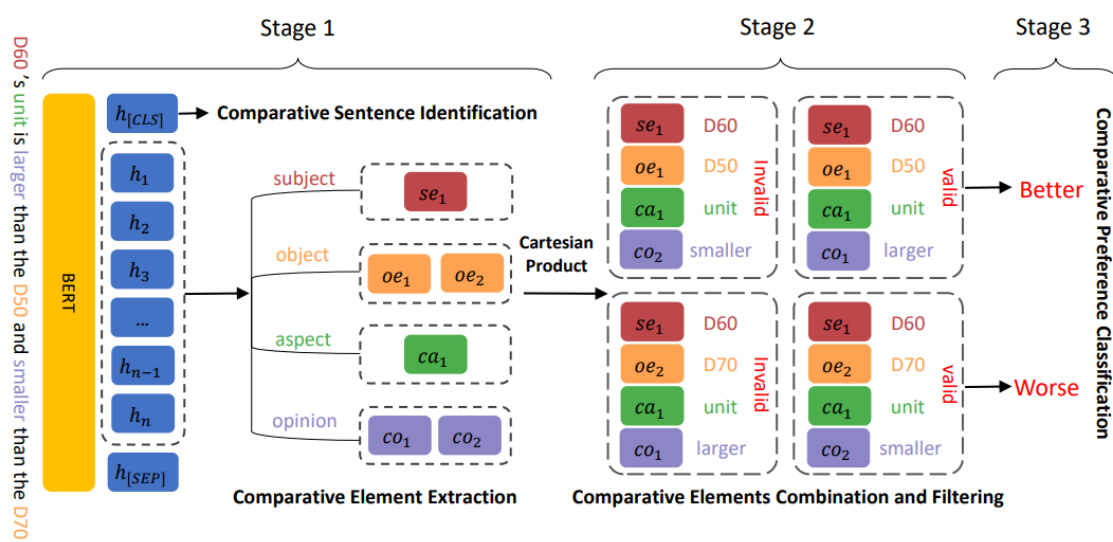
- Với việc mất cân bằng dữ liệu, cần thực hiện undersampling, tức là giảm số câu không so sánh về gần số câu so sánh, thực hiện training với các tỉ lệ 2:1, 1:1 hoặc giữ nguyên để lựa chọn ra tỉ lệ đưa ra kết quả tốt nhất
- Với những câu không hoàn chỉnh, có thể loại bỏ do số lượng ít, không ảnh hưởng quá nhiều đến mô hình.
- Về vấn đề các câu có nhiều quintuple hay khuyết quintuple, cần thiết kế 1 model phù hợp để có thể xử lý được các trường hợp này.

## 3. Phương pháp

Do yêu cầu đầu ra của bài toán gồm nhiều các yêu cầu nên ta sẽ xử lý bằng cách giải quyết từng yêu cầu theo từng bước nhỏ:

- Bước thứ nhất là xác định câu đó có phải là một câu so sánh hay không, nếu là một câu so sánh ta sẽ trích xuất các thành phần so sánh trong câu đánh giá sản phẩm này tương ứng với bộ 4 thành phần trong câu so sánh là subject, object, aspect, opinion (sub, obj, ca, co)
- Bước thứ hai bao gồm 2 giai đoạn. Mô hình sẽ từ các tập thành phần so sánh đã trích xuất được kết hợp thành một bộ 4 (sub, obj, ca, co). Sau đó bộ bốn này sẽ được xác định có phải một bộ hợp lệ hay không và xác định cp hay comparative preference của người dùng.

Kiến trúc mô hình được lấy từ một bài báo [\[1\] Z Liu et al.](#) được minh họa như Hình 1.



Hình 1: Kiến trúc mô hình COQE [1] (comparative opinion quintuple extraction)

Ví dụ với một câu đầu vào của bộ dữ liệu như: "iPhone 14 được ra mắt vào 09/2022, được đánh giá là chỉ nâng cấp nhẹ về phần cứng cũng như phần mềm so với thế hệ trước"

Trong bước thứ nhất ta cần đi xác định đây có phải một câu so sánh hay không và trích xuất được 4 tập tương ứng:

- $S_{sub} = \{ \text{"iPhone 14"} \}$
- $S_{obj} = \{ \text{"thế hệ trước"} \}$
- $S_{ca} = \{ \text{"phần cứng"}, \text{"phần mềm"} \}$
- $S_{co} = \{ \text{"nâng cấp nhẹ"} \}$

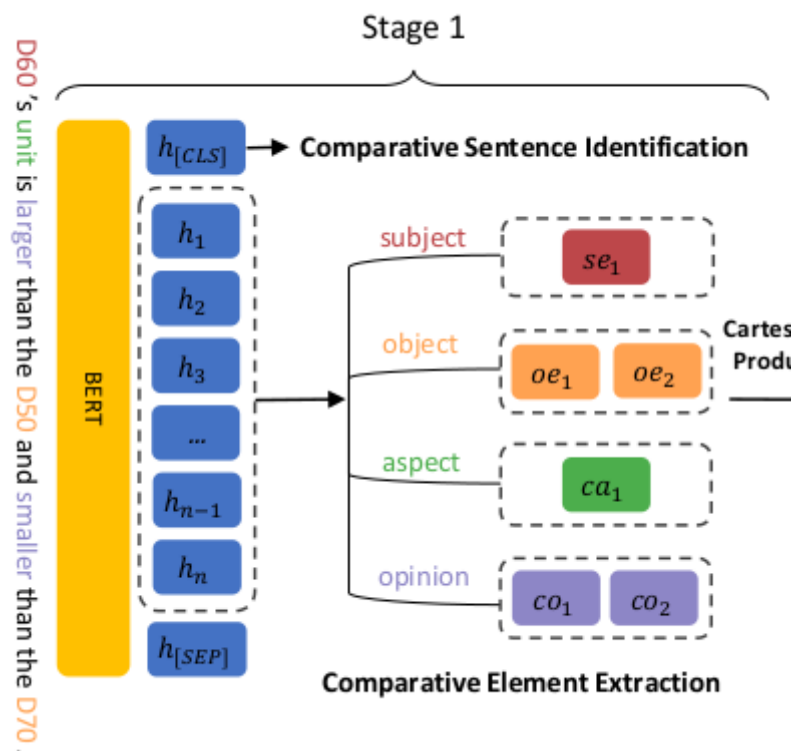
Đến bước thứ hai ta kết hợp các thành phần ở mỗi tập sử dụng tích Cartesian để tạo nên một bộ bốn. Sau đó một bộ phân lớp sẽ được huấn luyện để xem những bộ bốn được kết hợp có hợp lệ hay không. Tiếp theo mô hình sẽ dựa vào các bộ bốn hợp lệ để gán nhãn cho cp (comparative preference).

### 3.1 Phân loại câu so sánh và trích xuất thành phần so sánh.

Trong bước này mô hình sẽ sử dụng một mô hình được tiền huấn luyện trên dữ liệu tiếng Việt là [PhoBERT](#) để qua đó phân loại được các câu so sánh hoặc không so sánh.

Trong mô hình nguyên bản của tác giả do huấn luyện trên dữ liệu tiếng Anh nên sẽ sử dụng mô hình tiền huấn luyện trên bộ dữ liệu tiếng Anh nhưng với dữ liệu là tiếng Việt, [BERT](#) không thể hiểu được ngữ nghĩa của tiếng Việt nên mô hình phải thay thế BERT bởi PhoBERT.

PhoBERT là một mô hình xử lý ngôn ngữ tự nhiên dựa trên BERT được huấn luyện trên dữ liệu tiếng Việt phần nào hiểu được dữ liệu tiếng Việt. Ở đây để tokenize mô hình cũng sử dụng tokenizer được cung cấp bởi PhoBERT.



Với một câu đầu vào sẽ được tokenize bởi PhoBERT:  $X = [x_1, x_2, \dots, x_n]$ . Ta sẽ chèn hai token đặc biệt là  $\langle s \rangle$  và  $\langle /s \rangle$ . Token  $\langle s \rangle$  được chèn vào đầu câu làm nhiệm vụ phân loại câu so sánh hay không so sánh, token  $\langle /s \rangle$  được chèn vào cuối câu nhằm nhiệm vụ thông báo một câu đã kết thúc hay chưa. Đầu ra cuối cùng sau đi qua PhoBERT sẽ được biểu diễn như:

$$h = [h_{\langle s \rangle}, h_1, h_2, \dots, h_n, h_{\langle /s \rangle}]$$



Với mỗi biểu diễn đầu ra như trên ta sẽ truyền biểu diễn của token làm nhiệm vụ phân loại hay  $h_{<s>}$  vào một tầng softmax để xác định câu X có phải câu so sánh không. Đầu ra của lớp softmax này sẽ là:

$$y^c = \text{softmax}(W^c h_{<s>} + b^c)$$

Với  $W^c$  và  $b^c$  lần lượt là ma trận trọng số và bias  $y^c$  sẽ xác định câu X là so sánh hay không tương ứng với câu so sánh là nhãn {1} không so sánh là nhãn {0}

Để trích xuất các thành phần so sánh trong câu mô hình sử dụng 4 mô đun CRF riêng biệt tương ứng với việc trích xuất ra 4 tập.

Mô hình sử dụng 4 mô đun CRF (Conditional Random Fields) riêng biệt để trích xuất các thành phần tránh trường hợp các thành phần bị trùng nhau. Ví dụ như trong câu sau: “Mặc dù cả hai thiết bị đều sở hữu viên pin có dung lượng tương đương, nhưng tuổi thọ pin có chênh lệch khá lớn”. Ở ví dụ này sub và obj đều được xác định là “cả hai” khi đó nếu sử dụng một mô đun CRF thì có thể sẽ không trích xuất được sub hoặc obj.

Từ đầu ra của PhoBERT ta sẽ truyền biểu diễn từ vào 4 mô đun CRF riêng biệt để trích xuất ra thành 4 phần so sánh. CRF sẽ gán nhãn các từ dựa trên 5 tag Begin-Middle-End-Single-Outside (BMESO). Để đánh dấu từ nào sẽ là từ bắt đầu của thành phần, từ nào sẽ là từ ở giữa hay kết thúc, còn từ nào không nằm trong thành phần so sánh đang được trích xuất hoặc các từ là thành phần đang được trích xuất mà đứng đơn lẻ. Hàm mất mát cho bước thứ nhất được tính toán như sau:

$$L = \lambda_c L_{csi} + \lambda_e \sum L_{cee}$$

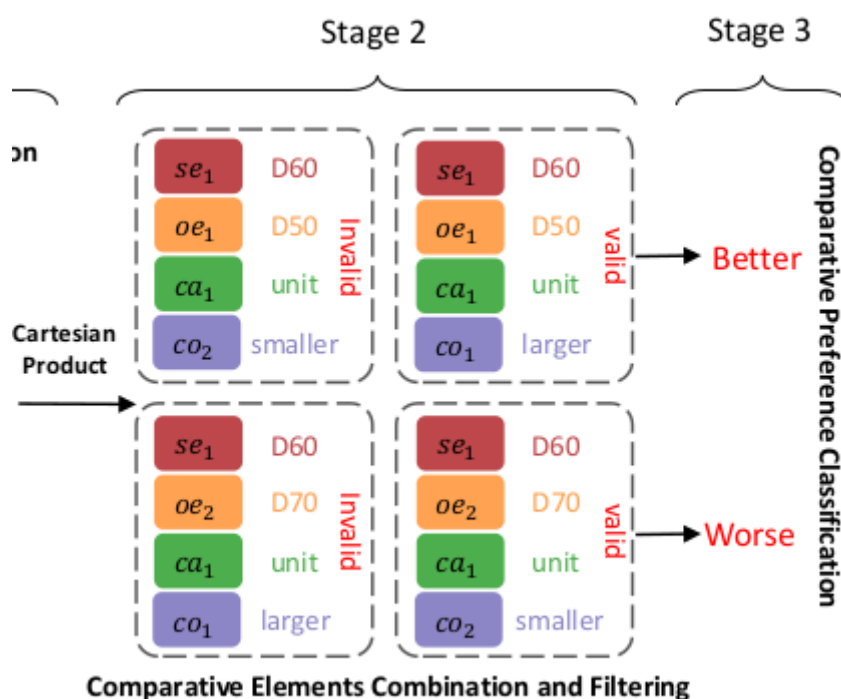
Với  $\lambda_c$ ,  $\lambda_e$  là các trọng số và  $L_{csi}$  là hàm mất mát cho bước phân loại câu so sánh.  $L_{cee}$  là hàm mất mát cho từng thành phần trích xuất.

### 3.2 Kết hợp, lọc và phân loại bộ các thành phần so sánh

Với mỗi tập các thành phần so sánh đã được trích xuất ở trên ta sẽ kết hợp chúng để tạo nên một bộ bốn (sub, obj, ca, co).

Ví dụ:  $S_{\text{sub}} = \{ \text{sub}_1, \text{sub}_2 \}$ ,  $S_{\text{obj}} = \{ \text{obj}_1 \}$ ,  $S_{\text{ca}} = \{ \text{ca}_1 \}$ ,  $S_{\text{co}} = \{ \text{co}_1, \text{co}_2 \}$   
thì ta sẽ có các bộ bốn là:

- $\{ \text{sub}_1, \text{obj}_1, \text{ca}_1, \text{co}_1 \}$
- $\{ \text{sub}_1, \text{obj}_1, \text{ca}_1, \text{co}_2 \}$
- $\{ \text{sub}_2, \text{obj}_1, \text{ca}_1, \text{co}_1 \}$
- $\{ \text{sub}_2, \text{obj}_1, \text{ca}_1, \text{co}_2 \}$



Mỗi bộ bốn các thành phần so sánh đã kết hợp mô hình sẽ phải lọc đi những bộ bốn không hợp lệ.

Với mỗi bộ bốn mô hình đi kết hợp biểu diễn của chúng ở các bước trước đó, ta sẽ nối biểu diễn của chúng ở bước phân loại câu so sánh hay không so sánh và biểu diễn khi đã được trích xuất thành thành phần so sánh.

$$r^e = [ \text{avg}( h^e_{[\text{start}:\text{end}]} ) ; \text{avg}( h_{[\text{start}:\text{end}]} ) ]$$

Ở đây start và end kí hiệu cho số thứ tự của từ trong câu văn, avg biểu diễn cho việc lấy trung bình các biểu diễn. Với mỗi biểu diễn của 4 thành phần ta sẽ có biểu diễn của mỗi bộ 4 như sau:

$$r = [ r_{\text{sub}} ; r_{\text{obj}} ; r_{\text{ca}} ; r_{\text{co}} ]$$

Cuối cùng biểu diễn này sẽ được đi qua một lớp softmax để lọc đi những bộ không hợp lệ:

$$y^{\text{quad}} = \text{softmax}(W^q r + b^q)$$

Với  $y^{\text{quad}} \in \{0, 1\}$  với 0 là bộ không hợp lệ và ngược lại. Sau đó với những bộ 4 hợp lệ mô hình sẽ truyền biểu diễn của mỗi bộ 4 vào một lớp softmax riêng biệt để xác định cp (comparative opinion) của chúng:

$$y^s = \text{softmax}(W^s r + b^s)$$

Với nhãn của  $y^s$  sẽ là: [ COM , COM+ , COM- , SUP , SUP+ , SUP- , EQL , DIF ]. Hàm mất mát được tác giả sử dụng là Cross-entropy.

## 4. Đánh giá hiệu năng mô hình

### 4.1 Thang đo

Để đánh giá bộ những thành phần so sánh đã được trích xuất bởi mô hình (CEE-Comparative Element Extraction), cuộc thi VLSP ComOM thách thức những đội thi tham dự trên nhiều các độ đo để xem xét hiệu suất của mô hình như Precision, Recall và F1 cho mỗi thành phần trích xuất được. Hơn nữa để đánh giá xem các bộ bốn (sub, obj, ca, co) hay bộ năm (sub, obj, ca, co, cp) được trích xuất Micro-Average và Macro-average sẽ được áp dụng:

- CEE - Comparative Element Extraction: Kết hợp đánh giá riêng lẻ trên từng thành phần và cả năm thành phần.
- T4 - tuple of four: đánh giá bộ bốn được trích xuất (sub , obj , co , ca ).
- T5 - tuple of five: đánh giá bộ năm được trích xuất (sub , obj , ca , co , cp ).

### 4.2 Tối ưu mô hình

Mô hình được huấn luyện và tinh chỉnh trên nhiều cài đặt khác nhau nhằm tối ưu hóa kết quả đạt được. Learning-rate được đặt là  $2e-5$  và  $5e-4$  lần lượt cho 2 bước như trên. Thuật toán Adam được sử dụng để tối ưu hóa mô hình.

Khi thực hiện đánh giá trên tập dữ liệu dev nhìn vào đầu ra của mô hình có thể thấy được rất nhiều các bộ bốn được trích xuất thiếu hay không hợp lệ, điều này làm cho hiệu quả mô hình bị giảm đi một cách đáng kể.

Để khắc phục điểm yếu này mô hình cải tiến thêm một bước để kiểm tra xem các thành phần của bộ bốn có đủ cả 4 thành phần hay không. Điều này làm tăng một phần kết quả của mô hình.

### 4.3 Kết quả

Cuộc thi VLSP ComOM xếp hạng theo kết quả của E-T5-MACRO-F1 mô hình phần nào đó thể hiện khá tốt trên bộ dữ liệu private test của ban tổ chức.

Team	E-T5-MACRO-F1
thindang	0.2373
pthutrang513	0.2300
thanhlt998	0.2131
duyvu1110	0.1119
ComOM_RTX5000	0.0997
kien-vu-uet	0.0975
nnn (our team)	0.0956
...	...

Kết quả trên E-T5-MACRO-F1 của đội thi xếp hạng thứ 7 trên private test. chỉ thua một chút so với các đội top 5 và 6 xếp trên.

Kết quả chi tiết từng thang đo đánh giá cho mô hình theo thang đo F1

Độ đo	Kết quả
E-T5-MACRO-F1	0.0956
E-T4-MACRO-P	0.1061
E-T5-MACRO-R	0.0909
E-T5-MICRO-F1	0.1868
E-T4-MICRO-P	0.1898

E-T5-MICRO-R	0.1839
E-T4-F1	0.2315
E-T4-P	0.2352
E-T4-R	0.2280
E-CEE-MACRO-F1	0.6010
E-CEE-MACRO-P	0.6379
E-CEE-MACRO-R	0.5702
E-CEE-MICRO-F1	0.5903
E-CEE-MICRO-P	0.6311
E-CEE-MICRO-R	0.5544

Điểm số trên các thang đo E-CEE khá cao nhưng khi kết hợp thành các bộ bốn thì điểm đánh giá trên các thang đo sụt giảm một cách đáng kể.

Điều này cho thấy khi kết hợp từ các thành phần đã được trích xuất thành các bộ 4 mô hình vẫn còn một lỗ hổng lớn cho thấy mô hình không thể kết hợp các bộ bốn một cách hợp lý.

## 5. Phân tích mô hình

Nhìn vào bảng kết quả như đã đề cập ở trên bước kết hợp ra các bộ bốn của mô hình vẫn còn kém hiệu quả dù trích xuất các thành phần so sánh trong quintuple cũng khá đầy đủ nhưng khi kết hợp và lọc đi những bộ bốn không hợp lệ, các bộ bốn này vẫn không đủ tốt do việc xét tất cả các bộ bốn có thể hình thành mô hình có thể đã không lọc tốt các bộ bốn không chính xác.

Ngay từ khi trích xuất ra các thành phần so sánh mô hình vẫn còn thua kém rất nhiều so với các đội thi khác mặc dù đã sử dụng một mô hình hiệu quả trong tiếng Việt như PhoBERT. Điều này có thể do các siêu tham số của mô hình chưa được tối ưu hiệu quả hay việc chạy quá nhiều epoch khiến mô hình có thể bị overfitting mặc dù đã có valid trên tập dev.

## 6. Định hướng phát triển

Do mô hình vẫn còn rất đơn giản nên ta có thể tiếp tục phát triển rất nhiều thứ như những điểm yếu đã phân tích ở trên. Ví dụ như bước kết hợp bằng tích hay lọc đi những bộ bốn không phù hợp cũng được thực tế chỉ ra là vẫn còn nhiều hạn chế. Một số hướng phát triển để khắc phục điểm yếu này có thể làm cách nào đó sắp xếp lại hay cấu trúc lại các bộ bốn từ các bộ thành phần đã được trích chọn.

Có thể tiếp tục tinh chỉnh mô hình bằng cách thử nghiệm với các cài đặt khác nhau xem bộ tham số nào là phù hợp nhất so với dữ liệu.

Tiếp đến là vấn đề dữ liệu như thống kê ở trên dữ liệu đang rất mất cân bằng. Số lượng câu so sánh so với câu không so sánh hay các số lượng các nhãn trong cp cũng có khoảng cách khá lớn điều này dẫn đến việc mô hình thể hiện chưa đủ tốt khi dự đoán các bộ bốn có nhãn như SUP-.

Một số phương pháp xử lý để dữ liệu huấn luyện trở nên tốt hơn như oversampling hay undersampling nhằm mục đích giảm sự mất cân bằng dữ liệu cũng có thể được áp dụng.

PhoBERT có thể không hiểu được một số từ trong dữ liệu, ta có thể sử dụng một phương pháp tối ưu hơn nhằm giúp cải thiện vấn đề này.

Một số kĩ thuật hay phương pháp đang được ưa chuộng hiện nay là ứng dụng graph vào giải quyết các bài toán xử lý ngôn ngữ tự nhiên như [UniCOQE](#) cũng đạt được kết quả rất tốt.

## 7. Tham chiếu

[1] <https://aclanthology.org/2020.findings-emnlp.92>

[2] <https://aclanthology.org/2021.emnlp-main.322.pdf>

[3] <https://ieeexplore.ieee.org/document/10191436/>

[4] <https://arxiv.org/abs/1810.04805>