

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

**NHÓM: Nopen**

**Châu Thiên Long – 21520331**

**Tăng Minh Hiền – 21520229**

**Vũ Anh Đức - 19521384**

**Nguyễn Thái Thành Long - 21520334**



**TÊN ĐỒ ÁN: TOXIC COMMENT DETECTION**  
**ĐỒ ÁN MÔN MÁY HỌC**

**GVHD: ThS. Phạm Nguyễn Trường An**

**Thành phố Hồ Chí Minh, tháng 6 năm 2023**



# MỤC LỤC

DANH MỤC HÌNH ẢNH .....	1
DANH MỤC BẢNG BIỂU .....	2
Chương 1 : CÁC THAY ĐỔI SO VỚI BÁO CÁO ĐÃ NỘP NGÀY 7/7/2023 .....	3
Chương 2 : TÓM TẮT ĐỒ ÁN .....	4
Chương 3 : NỘI DUNG .....	5
1. Tổng quan bài toán.....	5
1.1 Mô tả bài toán: .....	5
1.1.1 Ngữ cảnh bài toán .....	5
1.1.2 Mục tiêu thực hiện đồ án .....	5
1.1.3 Phát biểu bài toán.....	6
1.2 Mô tả dữ liệu phục vụ bài toán .....	6
2. Các nghiên cứu đi trước .....	7
2.1 Bài báo 1: Constructive and Toxic Speech Detection for Open-domain Social Media Comments in Vietnamese [1].....	7
2.2 Bài báo 2: ViHOS: Hate Speech Spans Detection for Vietnamese [2] ..	8
3. Quá trình xây dựng bộ dữ liệu: .....	9
3.1 Mô tả tóm tắt các bước crawl dữ liệu từ mạng xã hội Facebook.....	9
3.2 Gán nhãn dữ liệu:.....	10
3.2.1 Định nghĩa.....	10
3.2.2 Guideline.....	10
3.2.3 Đánh giá mức độ đồng thuận việc gán nhãn giữa các thành viên .....	12
3.2.4 Xử lý dữ liệu trong quá trình gán nhãn.....	16
3.2.5 Phân tích và đánh giá dữ liệu.....	18

4.	Huấn luyện và đánh giá hiệu năng mô hình: .....	22
4.1	Tiền xử lý dữ liệu.....	22
4.1.1	Tổng quan các bước tiền xử lý dữ liệu .....	22
4.1.2	So sánh hiệu năng khi thực hiện tiền xử lý dữ liệu .....	22
4.2	Phân chia bộ dữ liệu.....	24
4.3	Trích xuất đặc trưng cho bộ dữ liệu:.....	24
4.3.1	Count Vectorizer .....	24
4.3.2	TF-IDF Vectorizer .....	25
4.4	Mô hình sử dụng cho bài toán: .....	26
4.4.1	Logistic Regression.....	27
4.4.2	Support Vector Classifier.....	27
4.4.3	Decision Tree .....	27
4.4.4	Random Forest.....	28
4.4.5	Multinomial Naive Bayes .....	28
4.4.6	Transfer Learning – PhoBERT [7] .....	28
5.	Kết quả dự đoán và nhận xét.....	31
5.1	Kết quả dự đoán trên từng mô hình .....	31
5.1.1	Logistic Regression .....	31
5.1.2	Support Vector Classifier .....	31
5.1.3	Decision Tree .....	32
5.1.4	Random Forest .....	32
5.1.5	Multinomial Naive Bayes.....	33
5.1.6	Transfer Learning – PhoBERT.....	34
5.2	So sánh kết quả phân lớp giữa các mô hình .....	34

5.3	Nhận xét – Phân tích lỗi .....	35
5.3.1	Nhận xét.....	36
5.3.2	Phân tích lỗi.....	37
6.	Ứng dụng và hướng phát triển bài toán .....	39
6.1	Ứng dụng .....	39
6.2	Hướng phát triển bài toán .....	41
7.	Kết luận .....	42
Chương 4 : TÀI LIỆU THAM KHẢO.....		43

## DANH MỤC HÌNH ẢNH

Hình 2-1: Kết quả phân lớp của bài báo “Constructive and Toxic Speech Detection for Open-domain Social Media Comments in Vietnamese” .....	8
Hình 2-2: Kết quả phân lớp của bài báo “ViHOS: Hate Speech Spans Detection for Vietnamese” .....	9
Hình 2-3: Guideline gán nhãn dữ liệu của nhóm Nopen .....	11
Hình 2-4: Bảng danh sách các từ chửi rủa tục tĩu và kém văn hóa được nhóm xây dựng dựa trên bộ dữ liệu .....	12
Hình 2-5: Công thức tính hệ số Fleiss Kappa .....	13
Hình 2-6: Cách tính chỉ số Pe .....	14
Hình 2-7: Cách tính chỉ số Po .....	15
Hình 2-8: Đánh giá mức độ đồng thuận bằng thang đo Fleiss Kappa .....	15
Hình 2-9: Chỉ số Fleiss Kappa ở lần điều chỉnh thứ 3 .....	16
Hình 2-10: Minh họa bình luận được loại bỏ, không thêm vào bộ dữ liệu (Hình 1) ..	17
Hình 2-11: Minh họa bình luận được loại bỏ, không thêm vào bộ dữ liệu (Hình 2) ..	17
Hình 2-12: Bộ dữ liệu sau khi xử lý các bình luận không phù hợp mà nhóm đã đề cập .....	18
Hình 2-13: Thống kê số lượng label trên bộ dữ liệu .....	19
Hình 2-14: Độ dài trung bình của các bình luận của từng nhãn: Non-Toxic, Toxic, Very Toxic.....	20
Hình 2-15: Độ dài bình luận xét theo số token trong từng bình luận .....	20
Hình 2-16: Biểu đồ thống kê Top 10 từ có số lượng xuất hiện nhiều nhất trong bộ dữ liệu.....	21
Hình 2-17: Các từ có số lượng xuất hiện nhiều nhất trong bộ dữ liệu.....	21

## DANH MỤC BẢNG BIỂU

Bảng 2-1: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Logistic Regression giữa 2 cách trích xuất đặc trưng .....	31
Bảng 2-2: <i>Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Support Vector Machine giữa 2 cách trích xuất đặc trưng .....</i>	<i>31</i>
Bảng 2-3: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Decision Tree giữa 2 cách trích xuất đặc trưng .....	32
Bảng 2-4: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Random Forest giữa 2 cách trích xuất đặc trưng .....	33
Bảng 2-5: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Multinomial Naive Bayes giữa 2 cách trích xuất đặc trưng.....	33
Bảng 2-6: Bảng kết quả phân lớp của các mô hình mà nhóm đã huấn luyện .....	35

# **Chương 1 : CÁC THAY ĐỔI SO VỚI BÁO CÁO ĐÃ NỘP NGÀY 7/7/2023**

## **1. Xóa các hình ảnh trong phần code**

- Lướt bỏ các hình ảnh về source code trong phần trình bày
- Các source code được push đầy đủ lên github của nhóm
- Trình bày các bước thực hiện không kèm hình về source code

## **2. Cập nhật thêm các thử nghiệm chứng minh hiệu quả của bước tiền xử lý dữ liệu đối với hiệu năng mô hình**

- Nhóm đề xuất 8 thử nghiệm để xem xét hiệu quả của bước tiền xử lý dữ liệu ảnh hưởng thế nào đến bước tiền xử lý dữ liệu.
- Nhận xét của nhóm thấy rằng so với dữ liệu nguyên mẫu ban đầu thì hiệu năng của mô hình là tốt hơn khi thực hiện các bước tiền xử lý dữ liệu mà nhóm đã sử dụng so với khi không thực hiện tiền xử lý dữ liệu.

## **3. Kết luận cho bài báo cáo: Tóm tắt lại đồ án mà nhóm đã thực hiện**



## Chương 2 : TÓM TẮT ĐỒ ÁN

Tên đề tài mà nhóm Nopen thực hiện trình bày trong đồ án môn Máy học (CS114) có tên là “Toxic Comment Detection” - Nhận diện, phân loại các bình luận độc hại.

Bước đầu, nhóm thực hiện thống nhất các định nghĩa cho từng loại bình luận Non-Toxic, Toxic, Very Toxic; xây dựng guideline hướng dẫn chung để các thành viên tham gia gán nhãn trên bộ dữ liệu. Để làm rõ hơn các trường hợp cụ thể, nhóm đã tạo các bảng từ kém văn hóa và từ mang tính tục tĩu, chửi rủa; bổ sung nội dung cho bảng này trong quá trình gán nhãn.

Xây dựng bộ dữ liệu bằng cách crawl các bình luận từ các fanpage trên Facebook với một số chủ đề nhất định như esports, bóng đá, showbiz,.. Lấy khoảng 400 mẫu cho các thành viên thử gán nhãn và tính toán chỉ số đồng thuận, cải thiện các khái niệm và guideline để tăng chất lượng của nhãn. Sau đó phân chia gán nhãn trên bộ dữ liệu gốc với hơn 13 nghìn mẫu. Nhóm đã thực hiện thống kê và phân tích một số đặc điểm của bộ dữ liệu.

Sau đó , thực hiện một số bước tiền xử lý dữ liệu trước khi đưa vào model để train, như lowercase, xóa các dấu câu, dịch nghĩa từ viết tắt, đưa từ về dạng gốc, phân tách từ, loại bỏ stopwords,.. Trong đó ở bước dịch nghĩa các từ viết tắt, nhóm xây dựng một bảng từ điển dựa trên các từ viết tắt thường gặp trong quá trình gán nhãn, tương tự cho các từ sai dạng gốc được viết theo teencode.

Để có được mô hình phân lớp, nhóm đã thực hiện train trên 5 model phân lớp thường được sử dụng như Logistic Regression, Support Vector Classifier, Decision Tree, Random Forest ,Multinomial Naive Bayes. Thực hiện transfer learning PhoBERT để so sánh hiệu quả.

Link notebook:

[https://colab.research.google.com/drive/1lTvZiNAB8t04qvyWWa9awhdkW\\_X8s0qQ?usp=sharing](https://colab.research.google.com/drive/1lTvZiNAB8t04qvyWWa9awhdkW_X8s0qQ?usp=sharing)

Link dataset: [CS114/Project at main · Tlon9/CS114 \(github.com\)](https://github.com/Tlon9/CS114)

## **Chương 3 : NỘI DUNG**

### **1. Tổng quan bài toán**

#### **1.1 Mô tả bài toán:**

##### **1.1.1 Ngữ cảnh bài toán**

Trong thời đại số hóa ngày nay, mạng xã hội và các nền tảng truyền thông trực tuyến đã trở thành nơi giao lưu, chia sẻ thông tin và tương tác với nhau. Mặc dù điều này mang lại nhiều lợi ích nhưng cũng đặt ra những thử thách lớn đối với việc quản lý nội dung trên mạng.

Một trong những vấn đề phổ biến nhất đang có xu hướng phát triển là sự xuất hiện của những bình luận độc hại hay còn gọi là Toxic. Các bình luận này có thể chứa những lời lẽ xúc phạm, phân biệt chủng tộc, kích động bạo lực hay những nội dung không phù hợp khác như thể hiện quan điểm tiêu cực của các nhân,...Tình trạng này không chỉ gây tổn hại cho cộng đồng văn minh trên các nền tảng xã hội mà còn có thể gây ảnh hưởng tiêu cực đến sức khỏe tâm lý của những người bị mắc kẹt trong những cuộc thảo luận này.

Để giải quyết vấn đề này, bài toán phát hiện bình luận độc hại đã được đặt ra. Mục tiêu của bài toán này là xây dựng model có khả năng nhận diện và phân loại các bình luận có tính độc hại. Việc phát hiện bình luận độc hại không chỉ giúp ngăn chặn sự lan truyền của nội dung có hại mà còn tạo ra môi trường an toàn, lành mạnh cho người tham gia mạng xã hội.

Để giới hạn phạm vi của bài toán gần hơn với một bộ phận lớn giới trẻ tham gia trên mạng xã hội hiện nay, báo cáo này được thực hiện dựa trên dữ liệu từ các fanpage Facebook thuộc cộng đồng văn hóa mạng Việt Nam.

##### **1.1.2 Mục tiêu thực hiện đồ án**

Nhóm nhận thấy đã có các nghiên cứu thực hiện việc Xây dựng mô hình Máy học để thực hiện tác vụ “Toxic Speech Detection” và xây dựng ứng dụng Demo đơn

giản để kiểm chứng cho mô hình nhóm đã xây dựng. Từ đó có thể phát triển bài toán của nhóm để áp dụng vào thực tế cho các nền tảng mạng xã hội.

### **1.1.3 Phát biểu bài toán**

- Input: 1 bình luận của người dùng dưới dạng text
- Output: Phân loại bình luận trên vào 1 trong 3 nhãn :
  - + Non-Toxic: không gây hiềm khích, công kích, xúc phạm, miệt thị danh dự nhân phẩm người khác; mang tính đóng góp cho chủ đề được thảo luận.
  - + Toxic:
    - Bình luận có thể chứa các từ ngữ kém văn hóa nhằm mục đích thể hiện quan điểm tiêu cực của cá nhân mà không mang tính đóng góp cho bài viết hay chủ đề được thảo luận.
    - Bình luận dùng các từ ngữ kém văn hóa để chửi, công kích, gây hấn, miệt thị, xúc phạm danh dự, nhân phẩm.
    - + Very-Toxic:
      - Bình luận dùng các từ ngữ chửi rủa, tục tĩu nhằm mục đích đả kích, xúc phạm danh dự, nhân phẩm, miệt thị; đe dọa đến an toàn của người khác, thể hiện quan điểm tiêu cực của cá nhân với thái độ gây hiềm khích, xem thường người khác.
      - Bình luận sử dụng từ ngữ gây thù hận, kích động bạo lực, phân biệt vùng miền, giới tính, tôn giáo, vi phạm pháp luật.

### **1.2 Mô tả dữ liệu phục vụ bài toán**

- Dữ liệu cho bài toán “Toxic comment detection” được thu thập từ bình luận của các bài viết trên các fanpage Facebook thuộc các chủ đề gần với giới trẻ và được tranh luận theo nhiều hướng như thể thao điện tử, bóng đá, ngôi sao giải trí, showbiz Việt. Nhóm đã thống nhất lấy các bình luận từ các fanpage gồm GAM Esport LoL, SBTC Esport, LCK Tiếng Việt, Bí mật Showbiz, Theanh28Express, Hóng hót Showbiz Việt, GOAL Vietnam.
- Dữ liệu được thể hiện theo 3 cột: Id, Label, Comment, trong đó Id là số thứ tự của bình luận; Label là nhãn của bình luận trên, mang một trong các giá trị 0, 1, 2 lần

lượt tương ứng Non-Toxic, Toxic, Very-Toxic và Comment là nội dung bình luận của người dùng mạng xã hội.

- Các bộ dữ liệu sẵn có cho bài toán Toxic Comment Detection không đáp ứng được mục tiêu đặt ra phía trên do có nội dung bằng tiếng Anh hoặc lấy từ các nền tảng khác như các trang báo, diễn đàn,... Facebook mang tính riêng biệt hơn khi phần lớn người sử dụng ở độ tuổi trẻ có những xu hướng mới, văn hóa ứng xử hình thành xung quanh các cuộc tranh luận nhiều chiều mà cái tôi luôn được đề cao; do cộng đồng khá lớn và chưa có những quy định xử phạt cụ thể, quyết liệt nên người dùng mạng xã hội Facebook thường xuyên có ngôn từ xúc phạm, thiếu văn hóa để thỏa mãn và thể hiện cái tôi, dần hình thói quen xấu.

## **2. Các nghiên cứu đi trước**

### **2.1 Bài báo 1: Constructive and Toxic Speech Detection for Open-domain Social Media Comments in Vietnamese [1]**

Tác giả bài báo là Luan Thanh Nguyen, Kiet Van Nguyen, Ngan Luu Thuy Nguyen. Đây là mô hình được thực hiện bởi sinh viên và được sự hướng dẫn của thầy cô tại trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh. Bài báo đã thực hiện xây dựng mô hình để thực hiện 2 tác vụ: Nhận diện các bình luận tiêu cực và bình luận mang tính xây dựng trên các nền tảng mạng xã hội Việt Nam. Mô hình đã thực hiện xây dựng mô hình cho bộ dữ liệu UIT-ViCSTD, gồm các bình luận tiếng Việt trên các nền tảng mạng xã hội Việt Nam.

Bài báo đã giới thiệu xây dựng bộ dữ liệu UIT-ViCSTD, với tổng 10000 bình luận được lấy từ báo điện tử VnExpress và được chia thành 10 chủ đề: Giải trí, Giáo dục, Khoa học, Kinh tế, Xe, Pháp luật, Sức khỏe, Thế giới, Thể thao và Tin tức (với mỗi chủ đề gồm có 1000 bình luận) được gán nhãn Non-Toxic, Quite Toxic, Toxic và Very-Toxic với tác vụ phân loại bình luận tiêu cực. Dữ liệu trong bài báo đã thực hiện các bước tiền xử lý (gồm chuẩn hóa từ viết tắt trong câu bằng bộ từ điển các từ viết tắt, xóa bỏ các khoảng trắng, ký tự đặc biệt) và sử dụng mô hình pre-train PhoBERT cho xử lý ngôn ngữ tiếng Việt cho dữ liệu của bài toán để thực hiện biến

đổi dữ liệu dạng Text thành dạng văn bản. Kết quả hiệu năng của các mô hình chạy trên tập dữ liệu của bài báo được biểu diễn ở hình bên dưới.

Table 5: The experimental results of each model for constructive and toxic speech detection.

System	Constructiveness		Toxicity	
	Accuracy	F1-score	Accuracy	F1-score
Logistic Regression	79.91	70.78	90.27	55.35
Random Forest	79.10	73.75	90.03	55.30
SVM	78.00	76.10	90.17	59.06
LSTM + fastText	80.00	76.26	88.90	49.63
LSTM + PhoW2V	78.20	77.42	89.00	49.70
Bi-GRU-LSTM-CNN + fastText	79.90	77.53	89.10	48.88
Bi-GRU-LSTM-CNN + PhoW2V	79.50	77.94	88.90	49.62
<b>Our system</b>	<b>79.40</b>	<b>78.59</b>	<b>88.42</b>	<b>59.40</b>

Hình 3-1: Kết quả phân lớp của bài báo “Constructive and Toxic Speech Detection for Open-domain Social Media Comments in Vietnamese”

## 2.2 Bài báo 2: ViHOS: Hate Speech Spans Detection for Vietnamese [2]

Tác giả của bài báo là Phu Gia Hoang, Canh Duc Luu, Khanh Quoc Tran, Kiet Van Nguyen, Ngan Luu Thuy Nguyen. Đây cũng là bài báo có bản quyền thuộc trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh. Bài báo thực hiện tác vụ phân loại bình luận thù ghét, phản cảm trên bộ dữ liệu ViHOS.

Bộ dữ liệu được sử dụng trong bài báo là ViHOS gồm 11056 bình luận được gán là 1 trong các nhãn là Clean Comment và Hate/Offensive Comment. Bài báo đã sử dụng các mô hình pre-trained gồm XLM-R, PhoBERT kết hợp với mô hình Deep Learning là BiLSTM-CRF. Bên dưới là kết quả hiệu năng mà các mô hình trong bài báo đã được huấn luyện và kiểm thử trên tập dữ liệu.

	BiLSTM-CRF + Pho2W <sub>syllable</sub>	BiLSTM-CRF + Pho2W <sub>word</sub>	XLM-R <sub>Base</sub>	XLM-R <sub>Large</sub>	PhoBERT <sub>Base</sub>	PhoBERT <sub>Large</sub>
Full Data	0.7453	0.7036	0.7467	<b>0.7770</b>	0.7569	0.7716
W/o additional clean comments	0.6241	0.6244	0.6479	0.6756	0.6738	<b>0.6867</b>

Table 4: Experimental results on Full Data versus Without additional clean comments.

Model		Single span			Multiple spans			All spans		
		P	R	F1	P	R	F1	P	R	F1
Syllable	BiLSTM-CRF + Pho2W <sub>syllable</sub>	0.4222	0.5009	0.4329	0.5134	0.5712	0.5068	0.7452	0.7769	0.7453
	XLM-R <sub>Base</sub>	0.7604	0.7653	0.7203	0.7927	0.7574	0.7327	0.7766	0.7574	0.7467
	XLM-R <sub>Large</sub>	<b>0.7577</b>	<b>0.7679</b>	<b>0.7214</b>	<b>0.7829</b>	<b>0.7569</b>	<b>0.7357</b>	<b>0.8071</b>	<b>0.7887</b>	<b>0.7770</b>
Word	BiLSTM-CRF + Pho2W <sub>word</sub>	0.3196	0.4468	0.3594	0.3533	0.5001	0.4013	0.6823	0.7489	0.7036
	PhoBERT <sub>Base</sub>	0.7392	0.7485	0.7016	0.7761	0.7329	0.7092	0.7870	0.7680	0.7569
	PhoBERT <sub>Large</sub>	<b>0.7435</b>	<b>0.7567</b>	<b>0.7067</b>	<b>0.7878</b>	<b>0.7557</b>	<b>0.7321</b>	<b>0.8028</b>	<b>0.7835</b>	<b>0.7716</b>

Table 5: Experimental results on Single span, Multiple spans, and All spans subsets.

Hình 3-2: Kết quả phân lớp của bài báo “ViHOS: Hate Speech Spans Detection for Vietnamese”

### 3. Quá trình xây dựng bộ dữ liệu:

#### 3.1 Mô tả tóm tắt các bước crawl dữ liệu từ mạng xã hội Facebook

Nhóm đã thực hiện lựa chọn các bài đăng từ các trang đã nêu và thực hiện crawl dữ liệu trên từng bài đăng này. Sau đây là các bước để thực hiện crawl dữ liệu từ 1 bài đăng (post) trên mạng xã hội Facebook.

- Bước 1: Cài đặt thư viện Selenium trong python.
- Bước 2: Tiến hành truy cập vào link bài post muốn lấy dữ liệu và đăng nhập bằng tài khoản Facebook.
- Bước 3: Đổi chế độ xem bình luận thành “Tất cả bình luận” có thể hiển thị tất cả bình luận của bài viết, click “Xem thêm” để load lần lượt tất cả bình luận. Thay đổi chế độ lọc bình luận và load thêm bình luận bằng cách truy cập truy element theo XPATH.
- Bước 4: Lưu tất cả bình luận đã load vào file Dataset.csv và đóng trình duyệt.

**Tổng kết sau phần 1** - Sau bước crawl dữ liệu: Nhóm đã thu thập được tổng 13785 mẫu dữ liệu chưa qua xử lý.

### **3.2 Gán nhãn dữ liệu:**

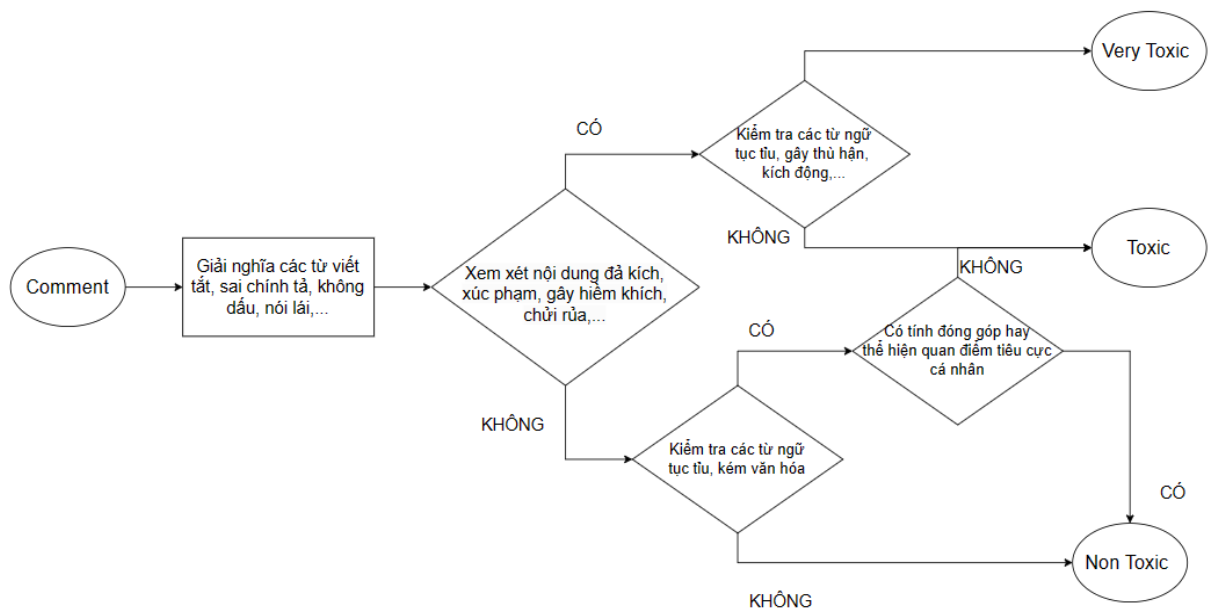
Bốn thành viên trong nhóm Nopen sẽ cùng tham gia việc gán nhãn dữ liệu, để đảm bảo chất lượng của việc gán nhãn, cần xây dựng quy trình như sau:

- Thống nhất định nghĩa cho từng nhãn non-toxic, toxic, very-toxic.
- Xây dựng guideline.
- Gán nhãn thử nghiệm trên 400 mẫu để đo mức độ đồng thuận giữa các thành viên.
- Phân tích điểm khác biệt, điều chỉnh khái niệm và guideline để cải thiện mức độ đồng thuận.
- Chia nhóm để tiến hành gán nhãn trên bộ dữ liệu chính.

#### **3.2.1 Định nghĩa**

- Non-Toxic: không gây hiềm khích, công kích, xúc phạm, miệt thị danh dự nhân phẩm; mang tính đóng góp cho chủ đề được thảo luận.
- Toxic:
  - + Bình luận có thể chứa các từ ngữ kém văn hóa nhằm mục đích thể hiện quan điểm tiêu cực của cá nhân mà không mang tính đóng góp cho bài viết hay chủ đề được thảo luận.
  - + Bình luận dùng các từ ngữ kém văn hóa để chửi, công kích, gây hấn, miệt thị, xúc phạm danh dự, nhân phẩm.
- Very-Toxic:
  - + Bình luận dùng các từ ngữ chửi rủa, tục tĩu nhằm mục đích đả kích, xúc phạm danh dự, nhân phẩm, miệt thị; đe dọa đến an toàn của người khác, thể hiện quan điểm tiêu cực của cá nhân với thái độ gây hiềm khích, xem thường.
  - + Bình luận sử dụng từ ngữ gây thù hận, kích động bạo lực, phân biệt vùng miền, giới tính, tôn giáo, vi phạm pháp luật.

#### **3.2.2 Guideline**



*Hình 3-3: Guideline gán nhãn dữ liệu của nhóm Nopen*

- Xây dựng bảng danh sách các từ chửi rủa tục tĩu và kém văn hóa dựa trên bộ dữ liệu. Đây là căn cứ để nhóm phân định các bình luận nào thuộc nhãn Toxic hoặc Very Toxic.



BẢNG DANH SÁCH CÁC TỪ CHỮI RỬA, TỤC TỮ

mẹ	Thằng l / vãi l (vl)	Vãi cả lon, hăm lồn
<b>Con chó</b>	đm / đcm	Vãi bùi
Cc / ncc / <u>Coincard</u> / CARD	Cái lùm mía / clm	Vcl / vkl
Não bò	Đéo / Đ / éo / đếch / méo	Cục cức
Ốc / Ốc cut/ óc bò	Con cac / cac / ncc / đặc cầu	Lol * (có thể nhầm thành tên game,...)
DB / đầu buoi / buoi / <u>đbrr</u>	concard	Vãi cứt
Làm gön	Súc vật, súc sinh	Cái nhồn
qq	Sồn lè	Cờ lờ
<u>Xiaolon</u>	Cụ mày	Bán độ / bố đạn
súc sinh / súc vật / sv	nứng	

BẢNG DANH SÁCH CÁC TỪ KÉM VĂN HÓA

Ngu/ <u>ngy</u> / <u>nguy</u> / ngân dù đều	Đần / chú bé đần / chần bé đù	Mất não
điên	khùng	Mất <u>dại</u> , khốn nạn, vô học
Bố mày	Con mẹ / Mẹ mày	Đầu đất
Gà , con gà, bù nhìn	Hôi	Cút, cook (nấu ăn, vua đầu bếp)
thằng	Dog no mom go home / No mom / Dog go home	dốt
Bọn	Rác / rác <u>ruối</u>	Dia / đáí
tạ	Rẻ rách	Thằng trấu
Lòi bản hòng	Ỉa, ẻ, mắc ẻ	Bú, (như) bẹn
Thằng hề	Ăn hại	Nắc
Thg gam con	<u>Thg</u> chột	Thẩm du / thẩm
Tự súc	Hấp diêm	Phim sế, seg / phim con heo / pòn / phim 18+
Đám ăn hại	Ngáo cần	Quốc nhục / nhục nhã
Cha/mẹ/ông nội*	Trứng dái	Phế vật
Đội quần	Thượng đẳng	bần
Giải thể	Lót đường	sủa

Hình 3-4: Bảng danh sách các từ chữi rửa tục tữ và kém văn hóa được nhóm xây dựng dựa trên bộ dữ liệu

### 3.2.3 Đánh giá mức độ đồng thuận việc gán nhãn giữa các thành viên

Vì bộ dữ liệu gồm 10671 mẫu dữ liệu và nhóm đã thực hiện phân công thực hiện gán nhãn cho 4 thành viên trong nhóm nên do đó nhóm đã sử dụng hệ số Fleiss Kappa để đo mức độ đồng thuận giữa các phân loại nhãn trong tiến trình gán nhãn của nhóm. Sau đây là công thức để tính hệ số Fleiss Kappa và thang đo đánh giá độ đồng thuận dựa trên hệ số này.

### Calculate Fleiss Kappa

We can calculate the Fleiss Kappa with this equation:

$$\kappa = \frac{\text{Observed agreement} - p_e}{1 - p_e}$$

Expected agreement if  
random judgment
























*Hình 3-5: Công thức tính hệ số Fleiss Kappa*

- $p_e$  được tính bằng tổng bình phương từng nhãn. Ví dụ:

Patient	Rater 1	Rater 2	Rater 2	😊	😞
1	😞	😞	😞	0	3
2	😞	😊	😞	1	2
3	😊	😊	😊	3	0
4	😊	😞	😞	1	2
5	😞	😞	😞	0	3
6	😊	😊	😞	2	1
7	😞	😞	😊	1	2
				Σ 8	13
				21	
				$\frac{8}{21} = 0.38$	$\frac{13}{21} = 0.62$
				$p_e = \sum p_j^2 = 0.38^2 + 0.62^2 = 0.53$	

Hình 3-6: Cách tính chỉ số Pe

- Po được tính bằng công thức sau:

Patient	Rater 1	Rater 2	Rater 2		
1				0	3
2				1	2
3				3	0
4				1	2
5				0	3
6				2	1
7				1	2

$$p_0 = \frac{1}{N \cdot n \cdot (n-1)} \left( \sum_{i=1}^N \sum_{j=1}^k n_{ij}^2 - \frac{N \cdot n}{7 \cdot 3 = 21} \right)$$

$$\frac{1}{7 \cdot 3 \cdot (3-1)} = 0.024$$

$$0^2 + 3^2 + \dots 1^2 + 2^2 = 47$$

$$p_0 = 0.024 \cdot (47 - 21) = 0.624$$

Hình 3-7: Cách tính chỉ số Po

- Đánh giá mức độ bằng thang đo:

Kappa	Level of Agreement
> 0,8	Almost perfect
> 0,6	Substantial
> 0,4	Moderate
> 0,2	Fair
> 0	Slight
< 0	No agreement

Landis & Koch (1977)

Hình 3-8: Đánh giá mức độ đồng thuận bằng thang đo Fleiss Kappa

Sau 3 lần điều chỉnh, chỉ số Kappa nhóm ghi nhận lần lượt tăng từ 0.29, 0.38, 0.75

STT	Long	Thành Long	Hiển	Đức	Label 0	Label 1	Label 2	Sum						
0	2	2	2	2	0	0	4	1600						
1	0	0	0	0	4	0	0			Label 0	Label 1	Label 2	Sum	
2	0	0	0	0	4	0	0			1112	204	284	1600	
3	0	0	0	0	4	0	0		pj	0.695	0.1275	0.1775	1	
4	0	0	0	0	4	0	0		pe	0.530788				
5	0	0	0	0	4	0	0							
6	1	1	1	1	0	4	0							
7	2	0	2	0	2	0	2			X	Y	p0=X.Y		
8	1	1	1	1	0	4	0			0.000208	4248	0.885		
9	0	0	0	0	4	0	0							
10	0	0	0	0	4	0	0			Fleiss Kappa				
11	0	0	0	0	4	0	0			0.754908				
12	0	0	0	0	4	0	0							
13	0	1	0	1	2	2	0							
14	1	1	1	1	0	4	0							
15	0	0	0	0	4	0	0							
16	0	0	0	0	4	0	0							
17	0	0	0	0	4	0	0							
18	0	0	0	0	4	0	0							
19	0	0	0	0	4	0	0							
20	0	0	0	0	4	0	0							
21	0	0	0	0	4	0	0							
22	0	0	0	0	4	0	0							
23	0	0	0	0	4	0	0							
24	1	0	1	0	2	2	0							
25	0	0	0	0	4	0	0							
26	0	0	0	0	4	0	0							

Hình 3-9: Chỉ số Fleiss Kappa ở lần điều chỉnh thứ 3

Dựa vào Hình 2-8, nhận thấy chỉ số Fleiss Kappa  $> 0.6$  là khá tốt, ở mức Substantial (đáng kể) nên nhóm tiến hành thực hiện gán nhãn trên bộ dữ liệu chính bằng cách phân công công việc như sau:

- Chia 4 thành viên thành 2 nhóm, mỗi nhóm gán nhãn một nửa bộ dữ liệu.
- Các mẫu có nhãn khác nhau sẽ được thảo luận lại và thống nhất.

### 3.2.4 Xử lý dữ liệu trong quá trình gán nhãn

Trong quá trình gán nhãn nhóm phát hiện các bình luận spam, không mang giá trị phân loại như chỉ chứa các icon, dấu câu hay chỉ tag tên người dùng khác.

Id	Label	Comment
5004	*	Ngô Huỳnh Đăng Tuấn
5005	*	Gia Huy
5006	*	Hoàng Minh Khôi
5008	*	Minhphuoc Nguyen Hồ Hiếu
5009	*	Lê Nhựt
5027	*	Chấn Hưng
5069	*	Cường Nguyễn
5150	*	Chánh Lê
7021	*	Plu
7808	*	Vĩ Khang
7904	*	Duy Quang
7905	*	Chau Thanh Hai
7906	*	Quốc Bảo
7907	*	Nguyễn Rin
7908	*	Trần Quốc Tùng
7909	*	Nguyễn Thắng
7911	*	Đạt Phát Trịnh
7913	*	Phát Lê
7915	*	Nguyễn Công Huy
7916	*	Dương Thiện Khải

Hình 3-10: Minh họa bình luận được loại bỏ, không thêm vào bộ dữ liệu (Hình 1)

9048	*	!!!!
9049	*	Phat Man Luu
9050	*	Phat Ly
9051	*	Nguyễn Trọng An
9052	*	Hoàng Nhẫn
9053	*	Thế Bảo
9054	*	Quốc Đạt
9056	*	Gia Bao Pham Hoang
9057	*	Gia Bách
9058	*	## @@@

Hình 3-11: Minh họa bình luận được loại bỏ, không thêm vào bộ dữ liệu (Hình 2)

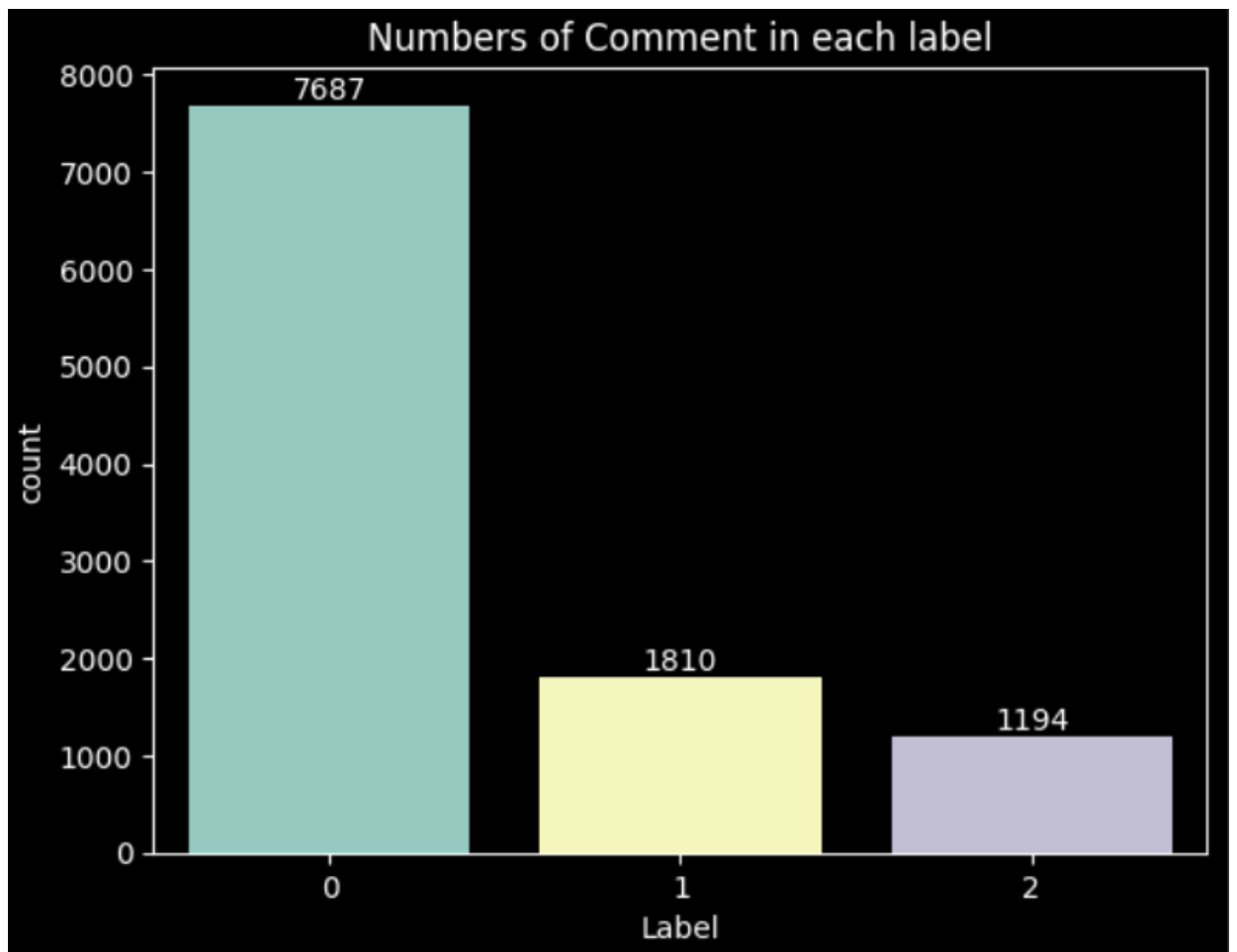
### 3.2.5 Phân tích và đánh giá dữ liệu

Sau khi xử lý các bình luận không phù hợp, bộ dữ liệu còn lại gồm 10691 mẫu, mỗi mẫu có 3 cột thông tin là Id, Label và Comment.

Id	Label	Comment
1	0	Slayder, GAM cần anh.
2	0	Dạ mỗi khi Gam cần, em luôn sẵn sàng!!
3	0	sở ty lè k chỉ nhảy qua tường, sở ty lè nhảy phát ra khỏi team luôn
4	0	Mê cái cách anh này nhảy E
5	0	Chúc mừng Gam đã có kỳ chuyển nhượng thành công
6	0	mê cái cách anh ấy nhảy e
7	0	Chuẩn thoát Pressing luônnn
8	0	OK ! Chúc e Sơn thành công trong con đường tiếp theo, giờ thì hóng ad mới thôi.
9	0	V là GAM ko dc đầu tư đi tới cửa sân bay nữa r
10	0	Trận đồ best gánh
11	2	Nhiều thắng cử hankay out hải vì nó dc đón lên là HLV tạm chứ vị trí chính vẫn là analyst, GAM có tuyển HLV thì cũng chả ảnh hưởng gì tới nó
12	0	Tôi đã biết trước trước là MSI xong là sẽ có bài thank you tới từ vị trí Sty1e mà
13	0	nếu đem Shogun về thì đem luôn Takl, 2 người đánh chung với nhau lâu rồi để kết hợp hơn
14	1	thầy sty1e chắc là sang lok để hướng dẫn thắng peyz vs thắng guma cách đánh con zeri chứ 2 thắng này đánh ngu quá
15	0	Dùng Phạm SE với GAM giúp nhau thoát pressing
16	0	nghi thức hiện tề bất đầu
17	1	ôi thật cảm động thế để hoá thân thành fan gam chứ vì khứa này tôi hơi anti gam một chút
18	0	bị kick ác nhượng chỗ cho tướng quân
19	0	t đợi cái bài thank you Kati mãi
20	0	mong mọi điều tốt nhất sẽ tới với sty1e !!
21	0	Style lướt dính quá, lướt phát out team luôn
22	0	Mê cái cách anh style E qua nhảy qua tường thành nhảy ra khỏi team luôn
23	0	Xong chính thức style ko có thêm cái cup nàoooo
24	1	kick là đứg r , sắp CKTG thì tuyển thắng nào đánh adc chiến lên . meta adc mà đánh cứ rên rên ra TG để lết đường trông chán
25	0	Vừa thoát pressing vừa thoát được cục tạ mùa sau làm Gam con thôi bạn Khoa Huỳnh

Hình 3-12: Bộ dữ liệu sau khi xử lý các bình luận không phù hợp mà nhóm đã đề cập

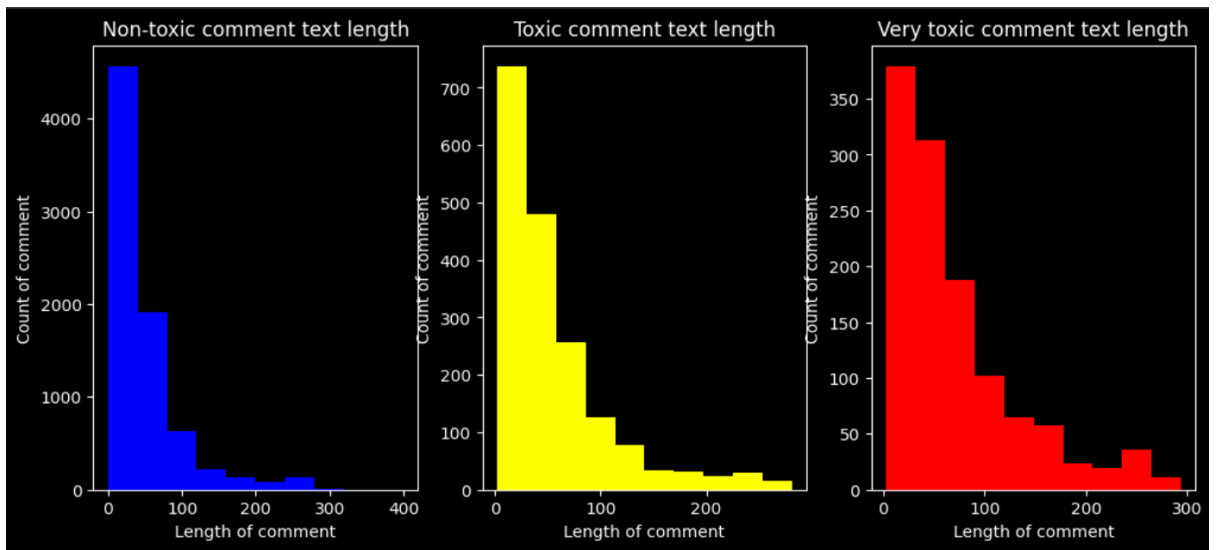
Trong đó có : 7687 mẫu có nhãn non-toxic, 1810 mẫu có nhãn toxic và 1194 mẫu có nhãn very-toxic. Bộ dữ liệu có sự mất cân bằng nhiều giữa các nhãn, cụ thể nghiêng hẳn về non-toxic. Điều này cũng phần nào hợp lý nếu xét tỉ lệ này trên thực tế, trung bình cứ 10 bình luận sẽ có 1 bình luận very-toxic, 2 bình luận toxic và 7 bình luận non-toxic.



*Hình 3-13: Thống kê số lượng label trên bộ dữ liệu*

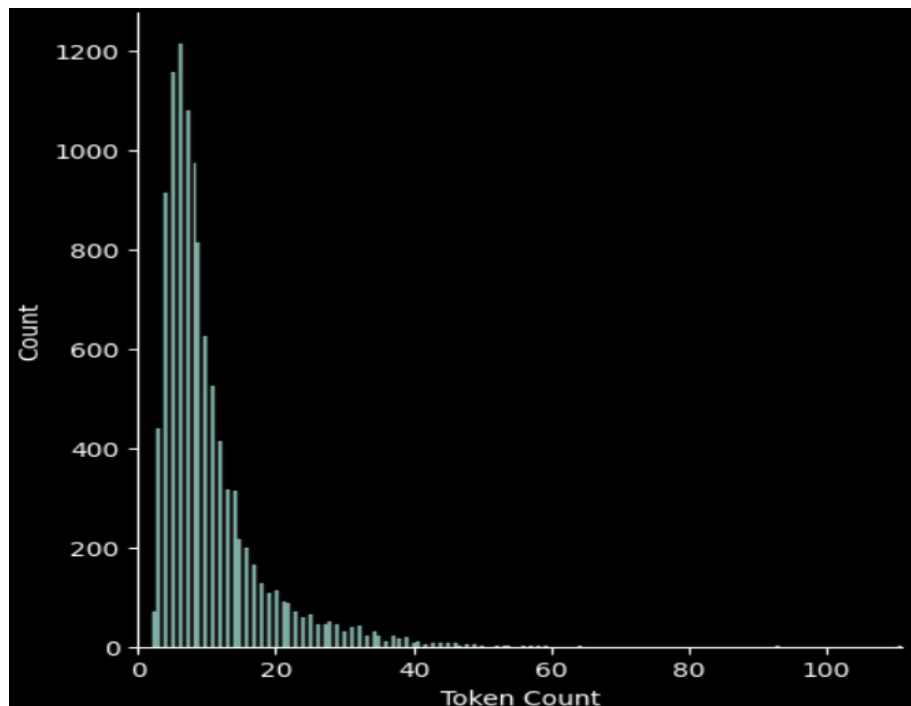
Độ dài trung bình của bình luận là 35 kí tự, các bình luận very-toxic và toxic có độ dài với xu hướng dài hơn các bình luận Non-Toxic.





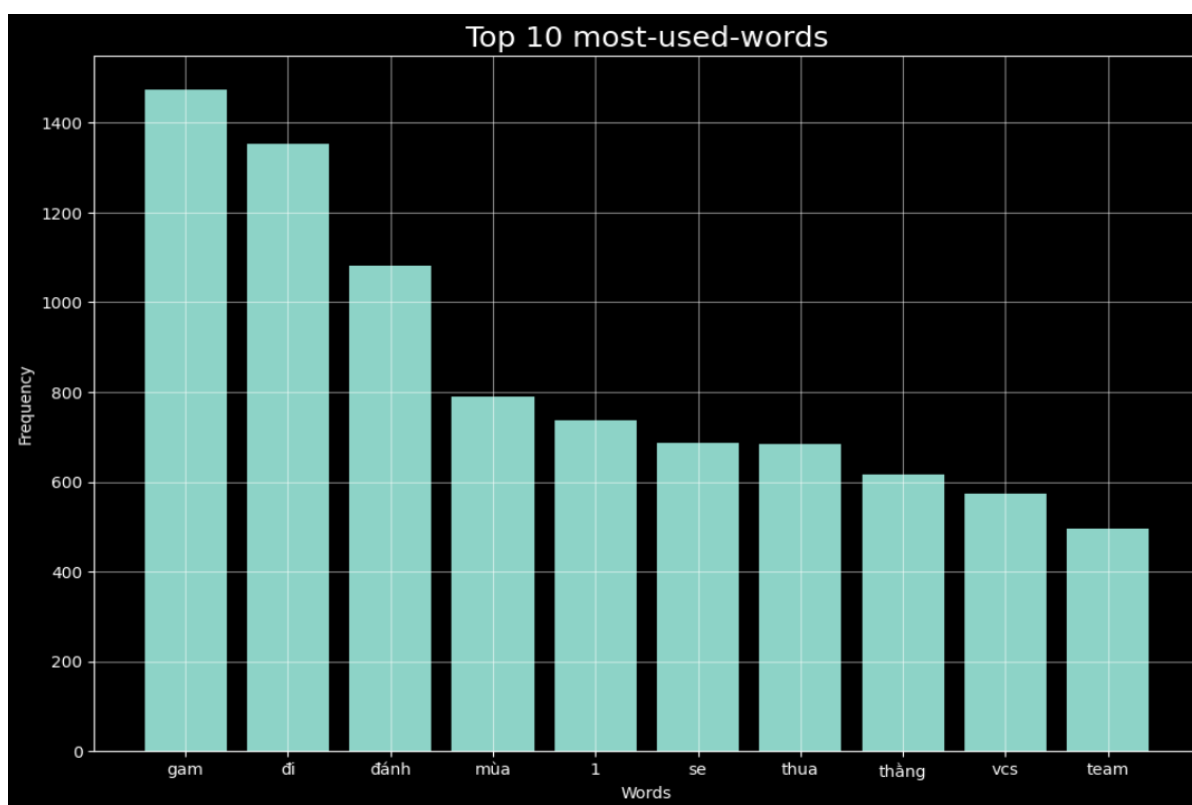
Hình 3-14: Độ dài trung bình của các bình luận của từng nhãn: Non-Toxic, Toxic, Very Toxic

Từ hình 2-14, ta có thể thấy độ dài bình luận trong bộ dữ liệu được nhóm thập xét theo số từ chủ yếu dao động từ 10-50 từ.

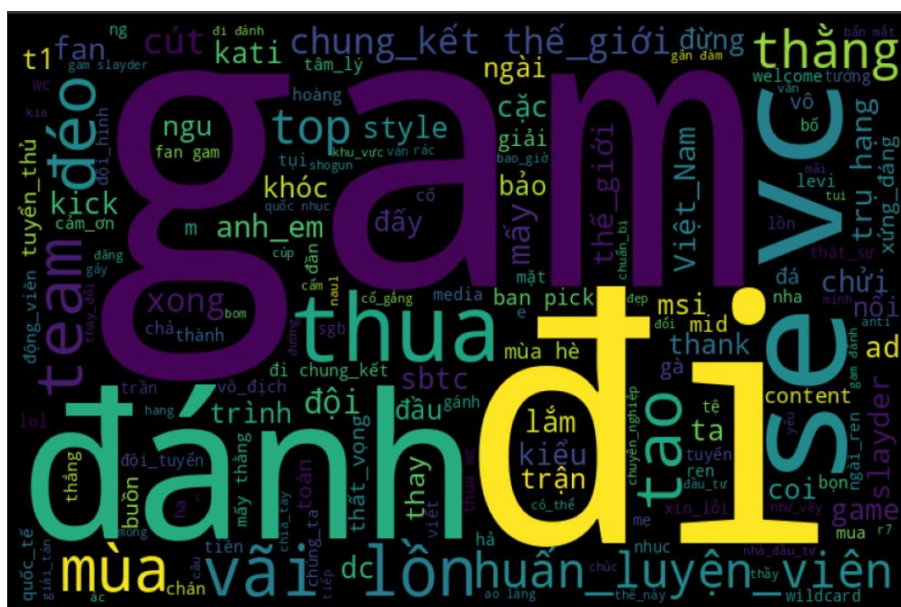


Hình 3-15: Độ dài bình luận xét theo số token trong từng bình luận

Các từ sử dụng nhiều nhất được thể hiện qua biểu đồ và word map bên dưới.



Hình 3-16: Biểu đồ thống kê Top 10 từ có số lượng xuất hiện nhiều nhất trong bộ dữ liệu



Hình 3-17: Các từ có số lượng xuất hiện nhiều nhất trong bộ dữ liệu

#### 4. Huấn luyện và đánh giá hiệu năng mô hình:

#### 4.1 Tiền xử lý dữ liệu

### 4.1.1 Tổng quan các bước tiền xử lý dữ liệu

Với bộ dữ liệu sau khi thực hiện bước gán nhãn, nhóm đã đề xuất các bước tiền xử lý như sau để thực hiện xử lý cho dữ liệu dạng Text. Nhóm cũng đã có tiến hành các thử nghiệm để kiểm chứng tính hiệu quả của các bước tiền xử lý và cho ra các mô hình đều hoạt động tốt hơn khi thực hiện các bước xử lý này. Sau đây là tóm tắt các bước tiền xử lý bao gồm:

- Bước 1: Chuyển các từ về dạng lowercase và xóa bỏ các dấu câu, danh sách gồm các dấu: ("\' , \' . , \', \': , \'\', \'! , \'~ , \'? , \'] , \'\*')
- Bước 2: Thực hiện chuyển các từ viết tắt thành từ đầy đủ về mặt ngữ pháp Tiếng Việt, danh sách các từ này được nhóm xây dựng khi xem xét trên toàn bộ tập dữ liệu. [1]
- Bước 3: Thực hiện Word Segmentation: Áp dụng mô hình Word Segmentation VNCORENLP, Bộ Toolkit xử lý ngôn ngữ tự nhiên cho ngôn ngữ tiếng Việt) [3] [4] [5] để tách token trong từng điểm dữ liệu. Thực hiện phân tách từ theo ngữ nghĩa của Tiếng Việt, các từ phải đi với nhau thì mới mang đúng ý nghĩa sẽ được gom nhóm như từ "học sinh", "bằng chứng". Thay vì tách riêng 2 từ là "học" và "sinh" thì sẽ được phân tách thành 1 từ là "học\_sinh".
- Bước 4: Thực hiện xóa các stopwords tiếng Việt, nhóm đã sử dụng danh sách các stopwords bằng bộ Vietnamese Stopword [6]. Nhóm đã tham khảo và tải về bộ Vietnamese stopword để sử dụng cho mô hình này.

#### 4.1.2 So sánh hiệu năng khi thực hiện tiền xử lý dữ liệu

Nhóm xin trình bày các kết quả về hiệu năng của mô hình với bộ dữ liệu khi chưa qua bước tiền xử lý và qua các bước tiền xử lý như sau (với thứ tự các bước như trên). Tổng quan, nhóm đã thực hiện 8 lần thử nghiệm để xem xét độ hiệu quả của các bước tiền xử lý trên. Để đánh giá, nhóm sẽ thực hiện huấn luyện mô hình Máy học (gồm Support Vector Classifier, Logistic Regression với dữ liệu được trích

xuất đặc trưng bằng Count Vectorizer và TF-IDF và mô hình Transfer Learning) và sau đó dựa vào accuracy của các mô hình mà đánh giá được bước tiền xử lý. Thứ tự lần lượt các lần thử nghiệm gồm:

- Không thực hiện tiền xử lý dữ liệu
- Chỉ thực hiện xóa stopwords cho từng mẫu dữ liệu (Bước 4)
- Chỉ thực hiện Word Segmentation (Bước 3)
- Chỉ thực hiện bước chuyển đổi các từ viết tắt thành từ đầy đủ (Bước 2)
- Chỉ thực hiện chuyển về lowercase và xóa dấu câu (Bước 1)
- Kết hợp bước 1 và bước 2
- Kết hợp bước 1, bước 2 và bước 3
- Thực hiện cả 4 bước

*Bảng 3-1: Bảng so sánh kết quả khi thực hiện các bước tiền xử lý dữ liệu*

STT	B1	B2	B3	B4	SVC		Logistic Regression		Transfer Learning
					Count	TF-IDF	Count	TF-IDF	
1					0.8214	0.8391	0.8546	0.8433	0.7923
2				X	0.8168	0.8332	0.8492	0.8309	0.8104
3			X		0.8158	0.8377	0.8500	0.8340	0.8058
4		X			0.8215	0.8349	0.8523	0.8211	0.7998
5	X				0.8214	0.8391	0.8546	0.8433	0.8120
6	X	X			0.8386	0.8550	0.8587	0.8465	0.8255
7	X	X	X		0.8344	0.8512	0.8550	0.8489	0.8428
8	X	X	X	X	0.8419	0.8519	0.8675	0.8561	0.8673

Trong đó:

- B1: Bước 1, B2: Bước 2, B3: Bước 3, B4: Bước 4
- X: Đánh dấu bước tiền xử lý được thực hiện tại từng lần thử nghiệm

#### **Nhận xét:**

- Khi thực hiện riêng rẽ các bước tiền xử lý dữ liệu như từ thử nghiệm 2 đến 5, có thể thấy hiệu năng của mô hình nhìn chung có xu hướng thấp hơn so với khi chưa thực hiện tiền xử lý dữ liệu (Bước 1). Ngoại lệ với Transfer Learning, thì hiệu

năng lại tăng qua tại 1 bước trích xuất đặc trưng so với thử nghiệm không tiền xử lý dữ liệu.

- Khi kết hợp các bước tiền xử lý dữ liệu như ở thử nghiệm 6, 7 và 8. Có thể thấy hiệu năng của mô hình đã có sự cải thiện hơn so với các thử nghiệm ở phía trên. Trong đó, với thử nghiệm 8, các chỉ số hiệu năng của các mô hình là tốt nhất, cải thiện nhẹ từ 1.5 đến 2% ở 2 mô hình SVC và Logistic và 7% ở mô hình Transfer Learning khi so sánh với khi chưa thực hiện tiền xử lý dữ liệu. Cho thấy, các bước tiền xử lý được nhóm áp dụng có kết quả tốt trên bộ dữ liệu.

#### **4.2 Phân chia bộ dữ liệu**

- Bộ dữ liệu của nhóm có tổng cộng gồm 10691 điểm dữ liệu, với mô hình Transfer Learning, nhóm thực hiện phân chia tập dữ liệu thành 3 tập Train, Validation, Test với tỉ lệ 7:1:2.

- Đối với các mô hình Máy học mà nhóm sẽ sử dụng bên dưới, nhóm sẽ thực hiện chia bộ dữ liệu thành 2 tập Train và Test với tỉ lệ 8:2.

#### **4.3 Trích xuất đặc trưng cho bộ dữ liệu:**

Vì đặc điểm của dữ liệu là dạng văn bản, để chuyển thành các ma trận số để huấn luyện các mô hình dự đoán, nhóm đã sử dụng 2 phương pháp trích xuất đặc trưng cho dữ liệu văn bản là Count Vectorizer và TF-IDF Vectorizer. Trong các mô hình Máy học, nhóm sẽ lần lượt sử dụng các cách trích xuất đặc trưng này để có sự so sánh về hiệu năng giữa các mô hình này và mô hình Transfer Learning - PhoBERT

##### **4.3.1 Count Vectorizer**

- Với đầu vào là các điểm dữ liệu văn bản, Count Vectorizer sẽ tạo bộ từ điển các từ đã xuất hiện trong toàn bộ các điểm dữ liệu đó. Sau đó tại mỗi dòng dữ liệu, phương pháp này sẽ thực hiện chuyển thành các ma trận có số chiều là số từ trong từ điển, và giá trị tại từng điểm trong vector là tần suất xuất hiện của từ trong điểm dữ liệu đó.

- Nhóm sử dụng Count Vectorizer từ thư viện sklearn của Python và thực hiện fit trên tập Train và transform trên tập Train và Test.

### 4.3.2 TF-IDF Vectorizer

- Thuật ngữ TF-IDF dùng để con số đại diện mức độ quan trọng của một từ đối với một đoạn văn bản trong kho chứa các văn bản.

- Cách tính: TF-IDF được tính bằng tích của 2 độ đo

+ Term Frequency of a word in document (TF weight): Có nhiều cách để tính độ đo này. Thông thường, cách đơn giản nhất là sẽ tính số lần xuất hiện của một từ trong văn bản. Sau đó, chuẩn hóa tần số này bằng độ dài của văn bản hoặc tần số xuất hiện của từ xuất hiện nhiều nhất trong văn bản hoặc tổng tần số của các từ xuất hiện trong văn bản. Bên dưới là 1 số cách tính Term Frequency theo wikipedia.

**Variants of term frequency (tf) weight**

weighting scheme	tf weight
binary	0, 1
raw count	$f_{t,d}$
term frequency	$f_{t,d} / \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Hình 3-18 Công thức tính TF weight (Nguồn: [Wikipedia](https://en.wikipedia.org/wiki/Tf-idf))

+ Inverse Document Frequency (IDF weight): để tính thông số này, người ta thực hiện tính tổng số văn bản chia cho số văn bản chứa từ đó và scale bằng logarit. Thông số này có nghĩa là một từ có tần suất xuất hiện thường xuyên hay thưa thớt trong tập hợp các văn bản. Nếu tính được độ đo này càng gần 0 thì từ này xuất hiện

thường xuyên trong tập văn bản này còn ngược lại khi càng gần 1, thì từ này ít xuất hiện trong tập văn bản. Bên dưới là các cách tính IDF weight theo wikipedia.

<b>Variants of inverse document frequency (idf) weight</b>	
<b>weighting scheme</b>	<b>idf weight (<math>n_t =  \{d \in D : t \in d\} </math>)</b>
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log \left( \frac{N}{1 + n_t} \right) + 1$
inverse document frequency max	$\log \left( \frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t} \right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

Hình 3-19: Công thức tính IDF weight (Nguồn: [Wikipedia](#))

- + Cuối cùng, cách tính để tính TF-IDF một từ: ta thực hiện lấy tích của TF weight và IDF weight. Kết quả của độ đo này càng lớn, từ đó càng phù hợp với đoạn văn bản đang xét.

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

Hình 3-20: Công thức tính TFIDF weight cho 1 từ (Nguồn: [Wikipedia](#))

- Nhóm sử dụng TF-IDF Vectorizer từ thư viện sklearn của Python. Thực hiện fit trên tập Train và transform trên tập Train và Test.

#### 4.4 Mô hình sử dụng cho bài toán:

Ở đề án này, nhóm thực hiện xây dựng mô hình như sau:

- Xây dựng mô hình Transfer Learning dùng model pre-trained cho ngôn ngữ tiếng Việt PhoBERT
- Xây dựng 5 mô hình Máy học gồm: Logistic Regression, Support Vector Classifier, Decision Tree, Random Forest và Multinomial Naive Bayes. Các mô hình này sẽ dùng lần lượt các cách trích xuất đặc trưng (Feature Extraction) gồm: Count Vectorizer và TF-IDF Vectorizer đã giới thiệu ở trên.
- Sau đây là một số tóm tắt về cách hoạt động của các mô hình mà nhóm sử dụng

#### **4.4.1 Logistic Regression**

Mô hình Logistic Regression là mô hình Máy học thuộc mô hình Máy học có giám sát, phân lớp sử dụng hàm Logistic để tính toán xác suất mà một điểm dữ liệu sẽ thuộc một lớp. Đối với bài toán phân lớp nhị phân, thì với xác suất tìm được, dựa vào Decision Boundary thì sẽ thực hiện phân lớp cho điểm dữ liệu. Đối với bài toán đa lớp, ý tưởng hoạt động là sẽ chia thành  $n$  bài toán nhị phân ( $n$  là số nhãn), sau đó tìm xác suất của điểm dữ liệu trong từng bài toán nhị phân và chọn xác suất lớn nhất trong các bài toán nhị phân và chọn nhãn đó trong bài toán nhị phân cho dự đoán của điểm dữ liệu.

#### **4.4.2 Support Vector Classifier**

Support Vector Classifier là mô hình Máy học thuộc nhóm mô hình Máy học có giám sát, có khả năng phân lớp tốt đối với các bài toán phân lớp tuyến tính hoặc phân lớp phi tuyến tính và có thể dùng cho bài toán hồi quy. Mô hình tìm một siêu phẳng (hyperplane) để thực hiện phân loại giữa các lớp. Đối với các bài toán phân lớp phi tuyến tính, SVC hỗ trợ việc chuyển đổi số lượng chiều ban đầu của thuộc tính (feature) của dữ liệu sang không gian có số lượng chiều nhiều hơn thông qua điều chỉnh kernel để tìm được hyperplane phù hợp để phân lớp.

#### **4.4.3 Decision Tree**



Mô hình Decision Tree là mô hình Máy học Máy học thuộc nhóm mô hình học có giám sát. Ý tưởng của mô hình thực hiện xây dựng cây quyết định dựa vào các thuộc tính mà dữ liệu có được. Từng node không phải node lá trên cây là các quyết định sẽ phân loại dữ liệu vào nhóm nào. Để lựa chọn thuộc tính phân loại cho các node này, mô hình sẽ dùng thuật toán ID3 hoặc CART. Node lá của cây sẽ là chính là nhãn được gán của dữ liệu.

#### **4.4.4 Random Forest**

Random Forest là mô hình Máy học thuộc nhóm mô hình Máy học học có giám sát. Mô hình sử dụng một tập hợp các Decision Tree. Mô hình này sẽ thực hiện huấn luyện dựa trên phương pháp bagging (hoặc là Bootstrap Aggregation), một số thuộc tính được lựa chọn ngẫu nhiên từ tập dữ liệu và sẽ thực hiện huấn luyện trên một “cây” trong “rừng”. Và một số lượng ngẫu nhiên dữ liệu sẽ được lựa chọn để huấn luyện cho từng “cây” trong khu “rừng”. Mỗi cây sẽ có kết quả dự đoán và thông qua biểu quyết số đông hoặc lấy giá trị trung bình của các kết quả dự đoán để đưa ra kết quả cuối cùng của khu “rừng”. Điều này sẽ làm giảm khả năng overfitting của mô hình với các bộ dữ liệu khác nhau.

#### **4.4.5 Multinomial Naive Bayes**

Multinomial Naive Bayes là dùng thuật toán được sử dụng phổ biến trong Xử lý ngôn ngữ tự nhiên (NLP). Đây là thuật toán thuộc nhóm kỹ thuật phân loại theo xác suất dựa trên định lý Bayes.

#### **4.4.6 Transfer Learning – PhoBERT [7]**

Transfer Learning model đang ngày càng thu hút rất nhiều sự quan tâm của các nhà nghiên cứu trong lĩnh vực NLP bởi hiệu năng ấn tượng mà mô hình này đem lại [8]. Một trong những SOTA của Language Model cho tiếng Việt hiện nay là PhoBERT, là mô hình với hướng tiếp cận dựa RoBERTa, là mô hình pre-train cho ngôn ngữ được tối ưu bước huấn luyện mô hình với hiệu suất tốt hơn trên bộ dữ liệu nhỏ hơn từ mô hình BERT.

Sử dụng PhoBERT-base-v2 với 135M tham số, nhóm đã tiến hành tham khảo từ Source đã được xây dựng [9] và thực hiện quá trình transfer learning cho bài toán được đặt ra trong đó các bước xử lý cho mô hình này được nhóm xử lý bao gồm:

- Import model và tokenizer của PhoBert.
- Đầu vào cho tokenizer là chuỗi các từ đã được phân tách. Do đơn vị từ trong tiếng Việt không chỉ phân biệt bởi khoảng trắng mà có thể được ghép từ nhiều từ đơn, các từ đơn thuộc cùng một từ sẽ được nối với nhau bởi dấu “\_” (ví dụ: riêng tư). Bộ phân tác từ được phoBert khuyến cáo sử dụng cho đầu vào của tokenizer là `py_vncorenlp.word_segment()`
- Import thư viện `torch.nn`, `torch.optim`, `torch.utils.data` để transfer.
- Viết hàm chuyển đổi dataset thành vector với `input_ids`, `attention_mask` và `target`. Sử dụng `tokenizer.encode_plus` để encode từ text sang vector. `Max_len = 30` do giới hạn của lượng từ phân bố nhiều nhất như đã phân tích ở trên thuộc phạm vi <30 từ.
- Viết lớp thực hiện phân loại nhãn bằng model Phobert, thực hiện dropout và thêm lớp Linear có `n_class` để chuyển model về phân loại theo số nhãn mong muốn (trường hợp này `n_class = 3`)
- Xuất thử model kiểm tra để thấy đầu ra đã chuyển từ 768 thuộc tính trở về 3 thuộc tính ứng với bài toán phân lớp

```

1 model = CommentClassifier(n_classes = 3)
2 model

SentimentClassifier(
  (bert): RobertaModel(
    (embeddings): RobertaEmbeddings(
      (word_embeddings): Embedding(64001, 768, padding_idx=1)
      (position_embeddings): Embedding(258, 768, padding_idx=1)
      (token_type_embeddings): Embedding(1, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): RobertaEncoder(
      (layer): ModuleList(
        (0-11): 12 x RobertaLayer(
          (attention): RobertaAttention(
            (self): RobertaSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): RobertaSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): RobertaIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): RobertaOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
      (pooler): RobertaPooler(
        (dense): Linear(in_features=768, out_features=768, bias=True)
        (activation): Tanh()
      )
    )
    (drop): Dropout(p=0.3, inplace=False)
    (fc): Linear(in_features=768, out_features=3, bias=True)
  )
)

```

*Hình 3-21: Model vừa được nhóm xây dựng*

- Viết hàm train model cho bộ dữ liệu, mỗi vòng lặp tính hàm loss bằng CrossEntropy, thực hiện backward và cập nhật tham số cho model.
- Viết hàm đánh giá model sau mỗi epoch train, giá trị dự đoán được lấy từ xác suất cao nhất trong 3 nhãn, tính accuracy dựa theo số mẫu dự đoán đúng trên tổng số mẫu.
- Viết hàm chia tập train, val, test bằng CommentDataset, tỉ lệ 7-1-2.

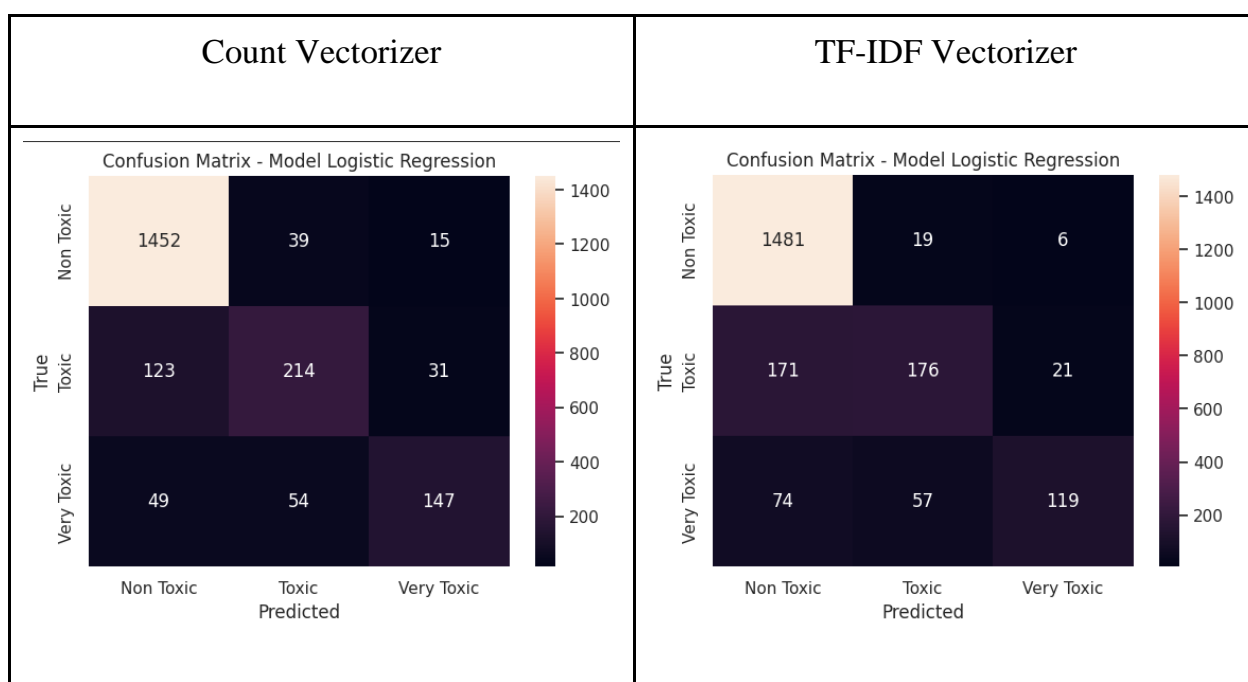
- Tiến hành train qua X epoch, batchsize = 32, tại mỗi epoch cập nhật và lưu lại model có kết quả tốt nhất.

## 5. Kết quả dự đoán và nhận xét

### 5.1 Kết quả dự đoán trên từng mô hình

#### 5.1.1 Logistic Regression

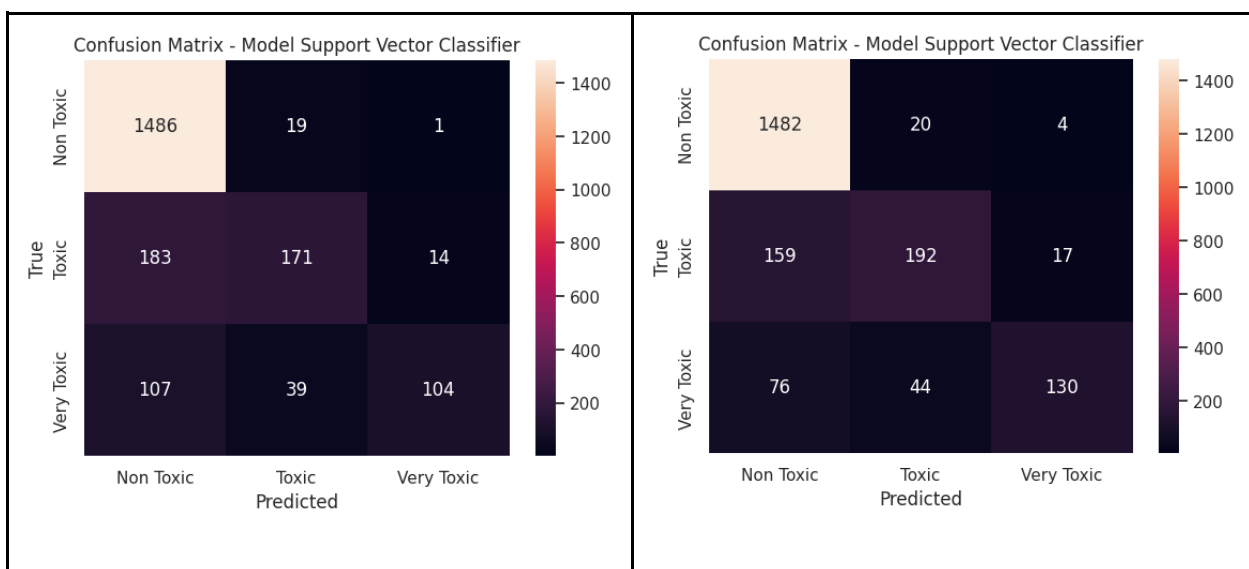
*Bảng 3-2: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Logistic Regression giữa 2 cách trích xuất đặc trưng*



#### 5.1.2 Support Vector Classifier

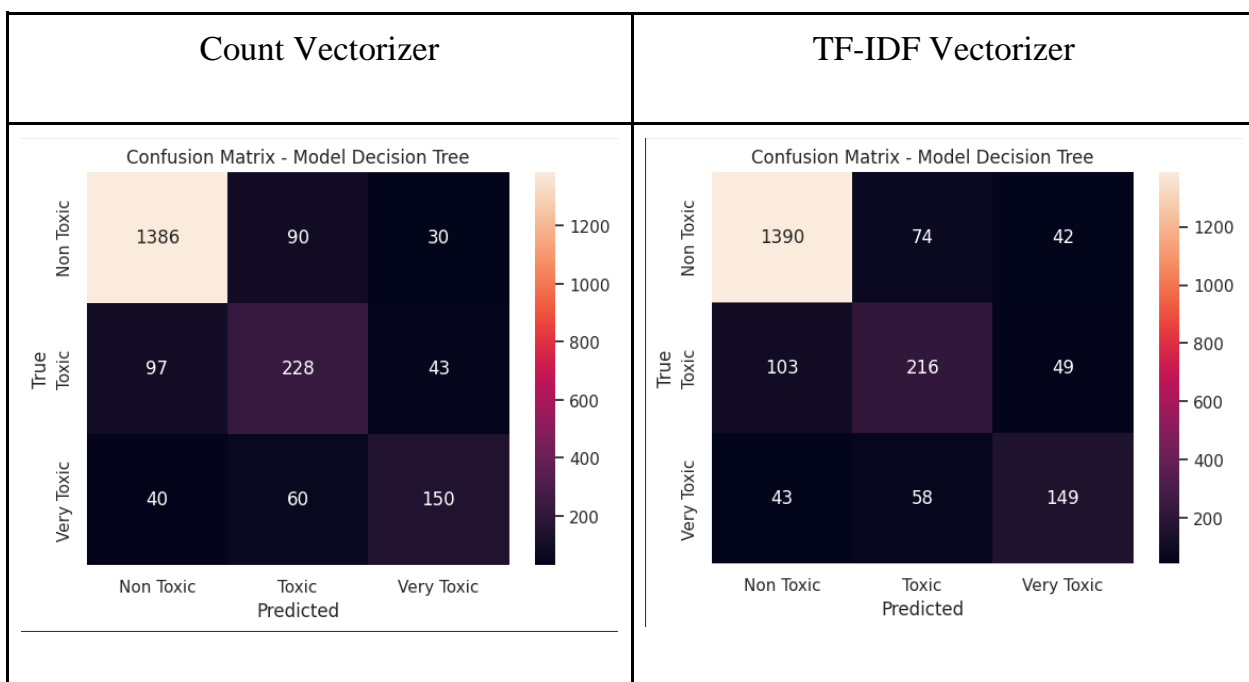
*Bảng 3-3: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Support Vector Machine giữa 2 cách trích xuất đặc trưng*

Count Vectorizer	TF-IDF Vectorizer
------------------	-------------------



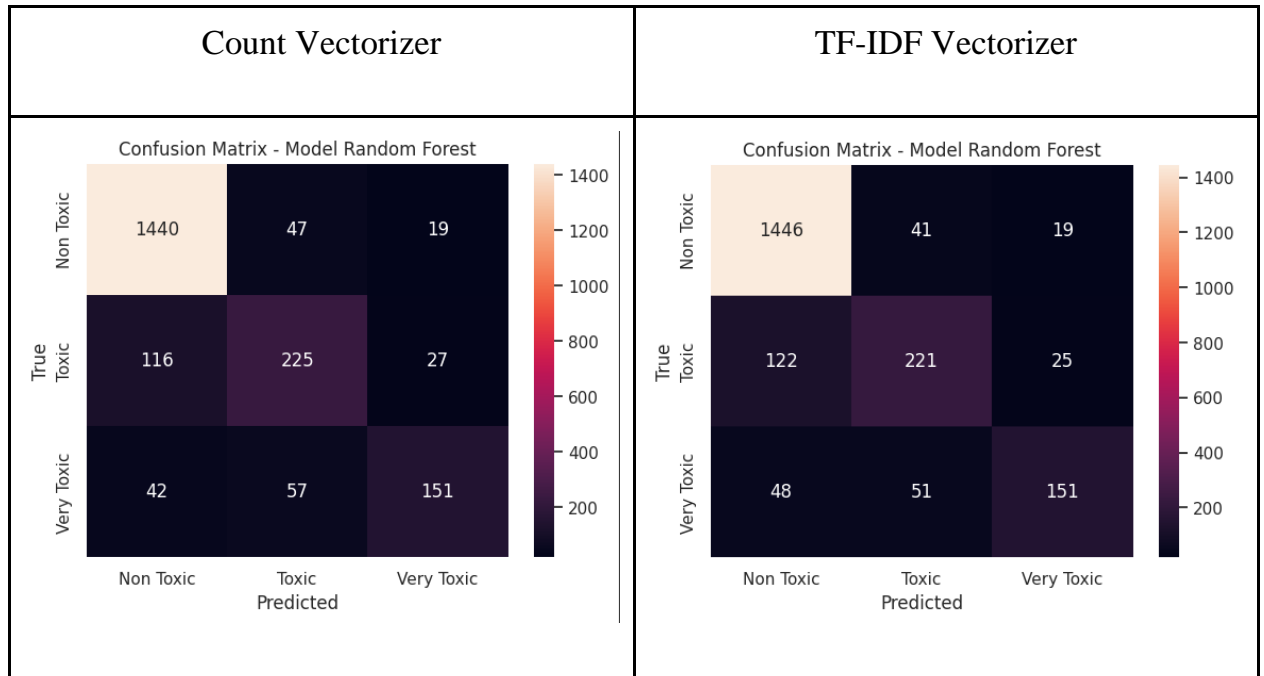
### 5.1.3 Decision Tree

*Bảng 3-4: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Decision Tree giữa 2 cách trích xuất đặc trưng*



### 5.1.4 Random Forest

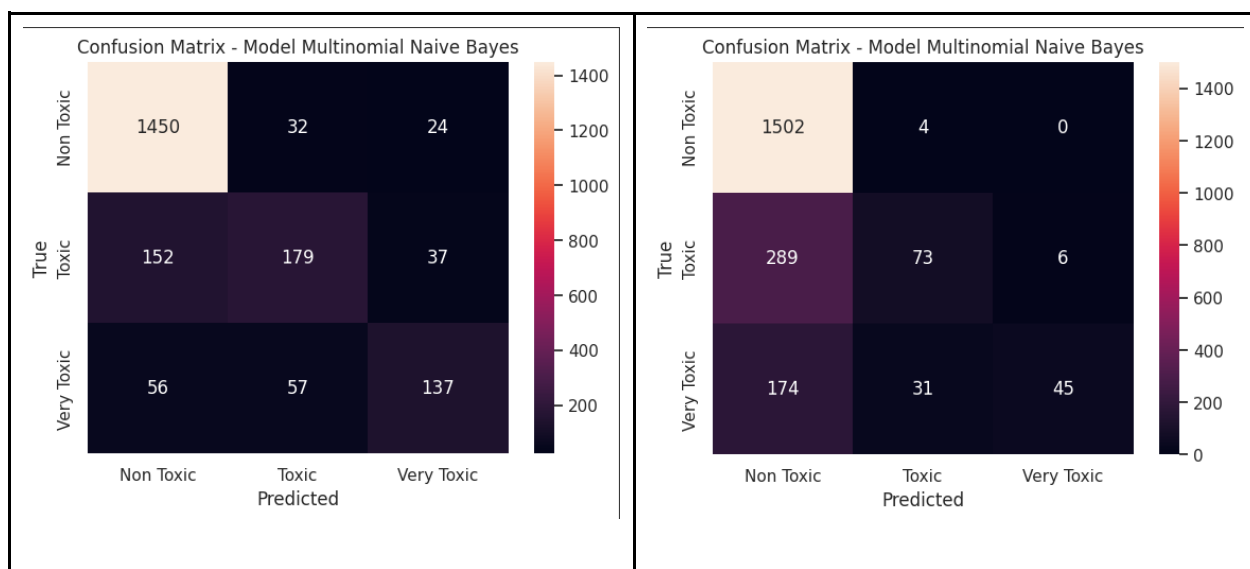
*Bảng 3-5: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Random Forest giữa 2 cách trích xuất đặc trưng*



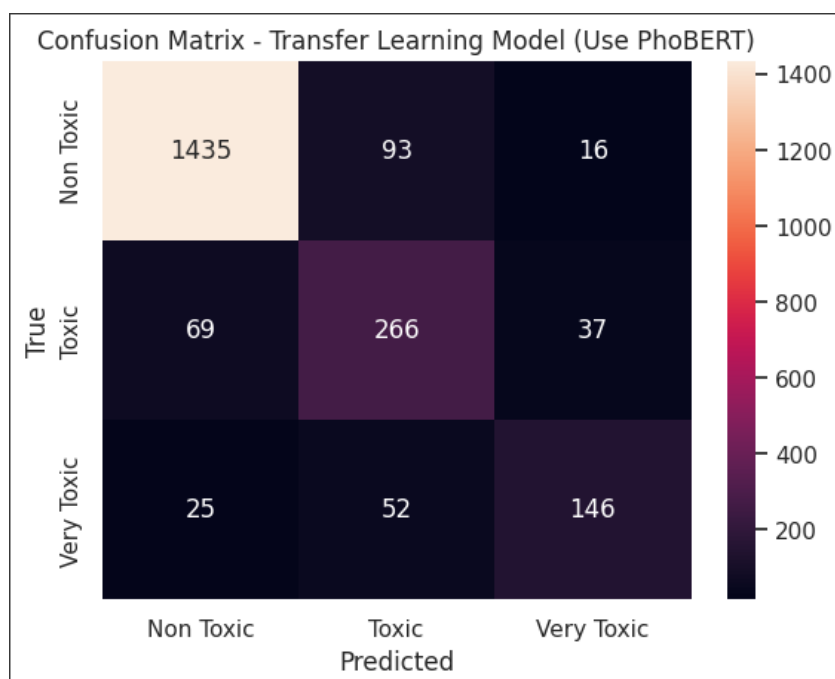
### 5.1.5 Multinomial Naive Bayes

*Bảng 3-6: Bảng kết quả phân lớp (biểu diễn bằng Confusion Matrix) của mô hình Multinomial Naive Bayes giữa 2 cách trích xuất đặc trưng*

Count Vectorizer	TF-IDF Vectorizer
------------------	-------------------



### 5.1.6 Transfer Learning – PhoBERT



Hình 3-22: Confusion Matrix của mô hình Transfer Learning – PhoBERT

## 5.2 So sánh kết quả phân lớp giữa các mô hình

*Bảng 3-7: Bảng kết quả phân lớp của các mô hình mà nhóm đã huấn luyện*

Classification Model	Classification Report							
	Accuracy		Macro Precision		Macro Recall		Macro F1	
Type of Feature Extraction	Count	TF-IDF	Count	TF-IDF	Count	TF-IDF	Count	TF-IDF
Logistic Regression	85.4	83.6	78.4	79.1	71.1	64.6	74.2	69.5
Support Vector Classifier	82.9	84.9	81.9	82.5	62.2	67.5	68.0	72.8
Decision Tree	82.9	82.6	72.5	71.5	70.9	70.2	71.6	70.8
Random Forest	85.5	85.6	78.7	79.2	71.9	72.2	74.7	75.1
Multinomial Naive Bayes	83.1	76.3	74.4	77.4	66.6	45.9	69.7	49.0
Transfer Learning use PhoBERT	87		79.6		79.2		79.4	

\*Các kết quả độ đo dùng đơn vị %, được làm tròn đến 1 chữ số thập phân

### 5.3 Nhận xét – Phân tích lỗi



### 5.3.1 Nhận xét

- Nhìn chung, các mô hình có độ chính xác trên độ đo accuracy là trên 75%. Mô hình Transfer Learning dùng PhoBERT có kết quả cao nhất 87%. Mô hình Random Forest khi dùng 2 trích xuất đặc trưng đều cho kết quả tốt ở độ đo này chỉ sau mô hình Transfer Learning. Accuracy thấp nhất ở mô hình Multinomial Naive Bayes, khi kết quả chỉ đạt 76.3% khi dùng TF-IDF Vectorizer.

- Xét trên độ đo Precision, các kết quả so với accuracy là thấp hơn ở các mô hình (trừ Multinomial Naive Bayes), cao nhất ở độ đo này là mô hình Support Vector Classifier, khi kết quả khi dùng 2 phương pháp trích xuất đặc trưng đều đạt trên 80% (lần lượt là 81.9% và 82.5%). Thấp nhất là mô hình Decision Tree khi dùng 2 cách trích xuất đặc trưng có kết quả là 72.5% và 71.5%.

- Xét trên độ đo Recall, độ đo này có kết quả thấp hơn khá nhiều so với accuracy và so với precision. Mô hình Transfer Learning cho kết quả Recall tốt nhất. Đối với mô hình Máy học còn lại, Các kết quả chỉ đạt từ 46% đến 72%. Đặc biệt, mô hình Multinomial Naive Bayes khi dùng TF-IDF Vectorizer cho kết quả đạt 45.9%, là kết quả thấp nhất trong các mô hình.

- Xét trên F1, mô hình Transfer Learning cho kết quả tốt nhất. Tuy nhiên, với các mô hình còn lại, kết quả chỉ từ ~49% đến 75%, trong đó thấp nhất, giống với Recall, là mô hình Multinomial Naive Bayes với kết quả là 49.0%. Mô hình Random Forest cho kết quả khá tốt khi dùng cả 2 cách trích xuất đặc trưng, lần lượt đạt 74.7% và 75.1%.

- So sánh về hiệu năng của các mô hình khi dùng 2 cách trích xuất đặc trưng, nhóm nhận thấy không có sự vượt trội về mặt kết quả phân lớp ở các mô hình giữa 2 cách trích xuất đặc trưng này. Tuy nhiên, mô hình Multinomial Naive Bayes không phù hợp khi dùng TF-IDF Vectorizer khi kết quả ở các độ đo đều là khá thấp so với các mô hình khác, đặc biệt là Recall và F1 của mô hình chỉ ở mức dưới 50%. Dễ hiểu vì mô hình này phân loại văn bản tốt với cách thức trích xuất đặc trưng Count Vectorizer. Xét các mô hình còn lại, nhóm nhận thấy:

+ Mô hình Logistic Regression: Count Vectorizer cho kết quả tốt hơn ở accuracy và ở 2 độ đo Recall và F1 thì cho kết quả lớn hơn và có độ chênh lệch so với TF-IDF Vectorizer.

+ Mô hình Support Vector Machine: TF-IDF Vectorizer cho hiệu năng cao hơn với mô hình này so với Count Vectorizer ở tất cả các độ đo

+ Mô hình Decision Tree: Count Vectorizer cho hiệu năng cao hơn ở mô hình này nhưng không quá nhiều so với khi dùng TF-IDF Vectorizer (chỉ từ 0.3 đến 1%)

+ Mô hình Random Forest: TF-IDF Vectorizer cho hiệu năng cao hơn Count Vectorizer nhưng không nhiều, các độ đo chênh lệch dưới 1%.

- Xét trên điểm số F1 của các mô hình, nhóm nhận thấy mô hình Transfer Learning có kết quả phân lớp tốt nhất cho bài toán “Toxic Comment Detection” với bộ dữ liệu nhóm đã thu thập. Kế đến là mô hình Random Forest, có thể dùng cả 2 cách trích xuất đặc trưng cho mô hình này. Và mô hình Logistic Regression khi dùng Count Vectorizer có kết quả F1 rất tốt, chỉ xếp sau 2 mô hình trên. Bên cạnh đó, mô hình Multinomial Naive Bayes có kết quả thấp nhất với bài toán này.

- Về thời gian huấn luyện, nhóm nhận thấy 5 mô hình Máy học có thời gian huấn luyện và thực hiện dự đoán trên tập kiểm thử khá nhanh (theo tính toán, chỉ tầm gần 2 phút cho 5 mô hình). Với ưu điểm thể hiện kết quả phân lớp tốt nhất khi xét trên các độ đo, nhưng mô hình Transfer Learning - PhoBERT có nhược điểm lớn là thời gian train rất lâu (khoảng 1 giờ cho 1 epoch) nhưng chưa đem lại hiệu quả quá cao so với các mô hình thông thường, cần cân nhắc sử dụng.

### **5.3.2 Phân tích lỗi**

index	Sentence	Predict	True
0	thằng ad mid đi thể_giới đánh ngu thể_này 2 con_mẹ ngu xong kêu tâm_lý	1	2
1	nhường slot đi chung_kết thể_giới se đi tao lạy chúng_mày đấy bọn mày đi chung_kết thể_giới cười mặt	0	1
2	kick kati đi	0	1
3	đm mày nát rồicook	1	2
4	kick	0	1
5	cà_rốt lớn d	2	1
6	hoàng lực hoàng lâmncct	0	2
7	động_viên cac ngu dở cú	1	2
8	đánh sĩ_nhục vãi chương cầm liss đánh đéo 1 tí áp_lực toàn bản_thân	0	2
9	đái chân vui	0	1
10	cánh bot gam thời_kì hậu noway bắt_ồn ad gam pha xử_lý đần thể_này hi_vọng bđ có_thể đem hãm_mộ chất_lượng số_lượng thể_này msi đơ	1	2
11	địt cu vãn chia_tay eq âm mẹ nhắm ngửa_mắt	1	2
12	tao hà hể gam thất_bại tao cực hà_hê cười một_số bộ_phận fan vcs fan gam nói_riêng giải tao nhận_định vcs trình na pcs 1 lũ chúng_mày comment kêu thằng nhục chửi_bới ranh_giới nhục ...	1	2
13	thối_nát lãnh_đạo media	0	1
14	vcs xuống_cấp nhà_tài_trợ đéo dám tiền một_số đội tiền duy_trì giải_tán đến_nơi tàn lol việt	2	0
15	mày tao ghéc thằng đầu vàng đấy	0	1
16	chuyên đạo_ly sống cực	2	1
17	hàn_quốc đánh_đấm thể_này đội thay máu con_mẹ đấy	2	1
18	hồng_hoa đái nước_mắt may	0	1
19	gà vãi lớn	1	2
20	team đéo tương_lai	0	2
21	quần què	2	0
22	hiệp lý hên đần	1	0
23	mùa lớn	1	2
24	đánh_đấm lolthấy giải ao làng vcs đội đánh oai tướng thể_giới tép lột đường	0	2

Hình 3-23: Minh họa các trường hợp mô hình Random Forest dự đoán sai

- Việc giải nghĩa các từ viết tắt hay không đưa các từ về đúng nghĩa gốc do có quá nhiều biến thể của từ để từ các lý do: sai chính tả, teencode, không dấu,... dẫn đến một số trường hợp nhầm lẫn như:

+ Các từ tục tĩu, kém không được hiểu đúng trở thành từ bình thường (vd: coincard, buoi)

+ Các từ ngữ bình thường do viết tắt hoặc không dấu bị nhầm thành từ tục tĩu, kém văn hóa (c,l,xl,...)

- Các từ tục tĩu, kém văn hóa bị chơi chữ, nói lái thành các từ bình thường, trường hợp này chưa đủ nhiều và có quá nhiều biến thể để nhận dạng (gândam,...)

- Bộ dữ liệu bị lệch về non-toxic nên các trường hợp sử dụng từ kém văn hóa để công kích hay chỉ thể hiện quan điểm tiêu cực cá nhân bị nhầm lẫn với trường hợp có tính đóng góp.

- Việc xác định chính xác thái độ công kích hay không trong quá trình gán nhãn chưa thật sự tốt làm cho bộ dữ liệu vẫn còn trường hợp nhầm lẫn gây nhiễu cho model.

- Với mô hình được sử dụng, nhóm chưa thực hiện tối ưu các tham số mô hình cho bài toán này. Đây là đặc điểm thiếu sót lớn mà nhóm chưa có đủ thời gian để tạo các bộ tham số để thực hiện tinh chỉnh cho từng mô hình nhằm tìm ra bộ tham số tốt nhất cho từng mô hình.

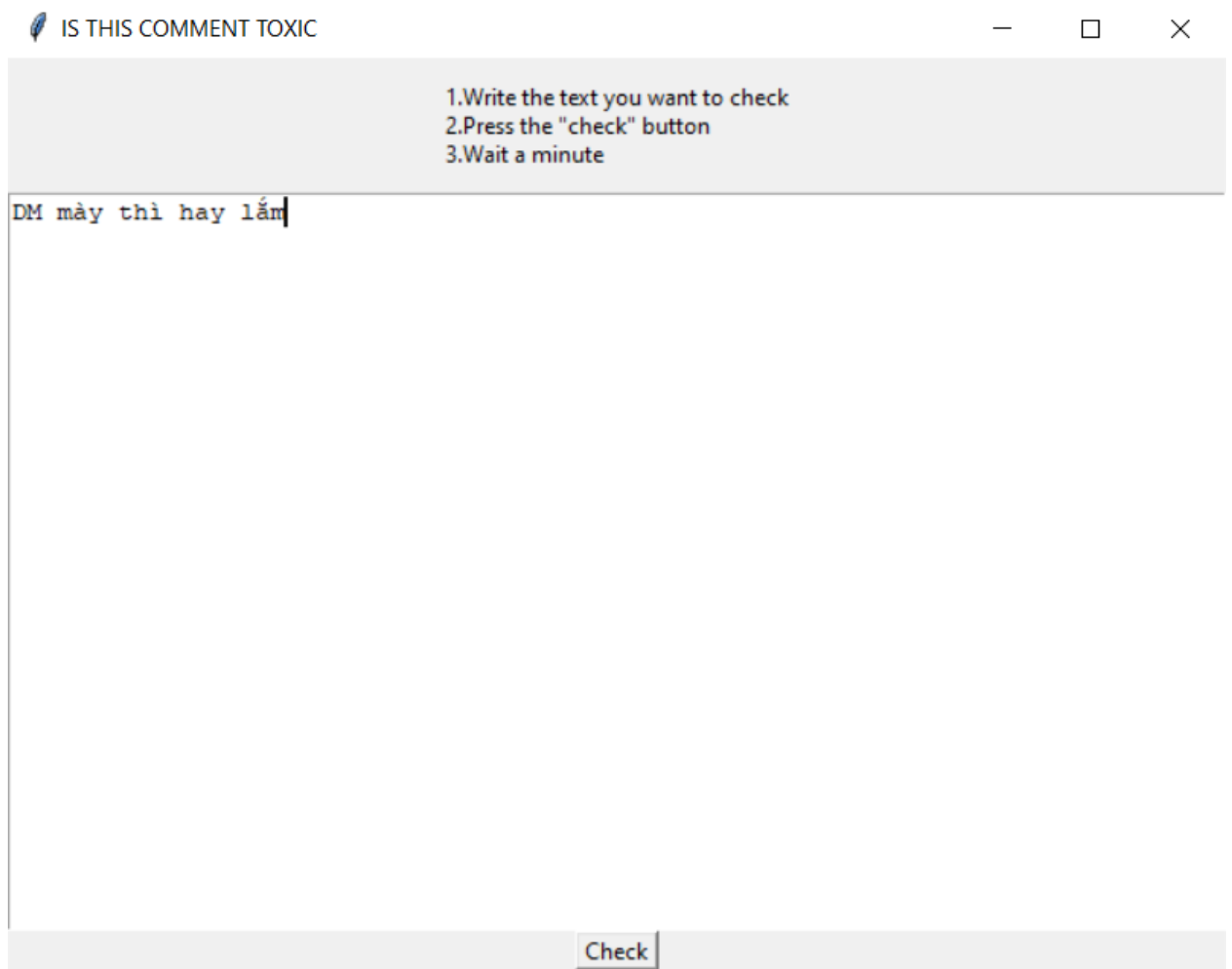
## **6. Ứng dụng và hướng phát triển bài toán**

### **6.1 Ứng dụng**

Tóm tắt các bước để xây dựng ứng dụng demo cho mô hình Transfer Learning được nhóm xây dựng

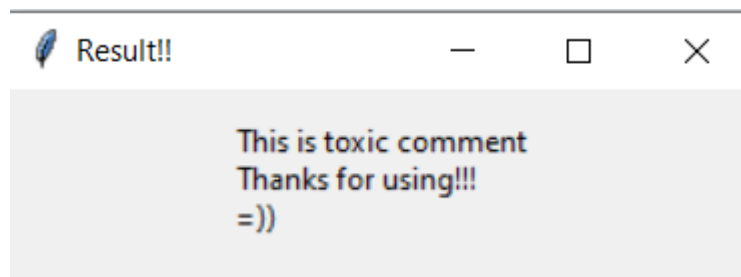
- Sau khi chạy xong các model, nhóm sẽ lưu các model lại thành file .pth và sử dụng thư viện torch để xây dựng 1 chương trình kiểm tra đoạn bình luận được truyền vào là toxic hay không

- Tạo phobert và class cần để tạo model.
- Tiến hành load model từ file phobert.pth
- Thêm những hàm làm sạch dữ liệu: Loại bỏ dấu câu, Loại bỏ stopwords
- Sử dụng tkinter để tạo 1 giao diện người dùng đơn giản, tạo ra được 1 vùng trống để người dùng đưa đoạn văn bản cần được kiểm tra vào để kiểm tra. Giao diện có được sẽ có dạng như sau:



*Hình 3-24: Giao diện của ứng dụng Demo nhóm xây dựng*

- Tạo ra 1 phím bấm Check trên giao diện, để khi người dùng nhấn Check sẽ chạy hàm `CreateResultWindow` để dự đoán. Sau khi ấn check thì kết quả sẽ là:



*Hình 3-25: Màn hình kết quả sau khi nhập 1 comment của mô hình Transfer Learning*

Tuy nhiên, không phải lúc nào chương trình cũng chạy ra được kết quả chính xác. Điều này là do model không thể nào đạt được 100%, điều này vẫn có thể chấp nhận được vì phần lớn kết quả vẫn chính xác (>90%).

## **6.2 Hướng phát triển bài toán**

Từ kết quả phân lớp của bài toán “Toxic Comment Detection” nhóm đã thu được, nhóm cũng đề ra một số hướng phát triển cho bài toán của nhóm như sau:

- + Thứ nhất, xây dựng chi tiết hơn về định nghĩa và guideline cho các loại bình luận, tiến hành tập gán nhãn nhiều lần để tăng độ đồng thuận của nhãn.
- + Thứ hai, mở rộng chủ đề trên mạng xã hội cũng như tăng cường bộ dữ liệu, giảm sự mất cân bằng bằng cách tìm kiếm nhiều bình luận toxic từ các bài viết có chủ đề kích war, gây hấn.
- + Thứ ba, mở rộng từ điển xử lý các từ ngữ viết tắt, teencode, nói lái, chơi chữ,... để giải nghĩa bình luận chính xác hơn. Bên cạnh đó, nhóm cũng thấy rằng có thể thực hiện thêm bước tiền xử lý dữ liệu như phân loại các icon cảm xúc, giảm bớt việc xuất hiện tên riêng do tag tên trên mạng xã hội trong các bình luận ...
- + Thứ tư, tăng cường thời gian train cho PhoBERT và thực hiện train thay vì trên nền tảng online nhóm đã sử dụng là Google Collab thì thực hiện train offline trên phần cứng máy tính.

+ Thứ năm, thực hiện tinh chỉnh tham số cho các mô hình đã thực hiện để cải thiện hiệu năng cho các mô hình đã sử dụng. Bên cạnh đó, tiến hành nghiên cứu thêm các giải pháp, các mô hình mới hơn để thực hiện phân lớp cho bài toán này.

+ Thứ sáu, cải tiến phần xây dựng ứng dụng gồm cải tiến phần giao diện người dùng, tạo chương trình có thể thực thi trên web hoặc máy tính để mở rộng tính thực tiễn của bài toán. Từ đó, đi xa hơn có thể áp dụng cho các nền tảng mạng xã hội.

## **7. Kết luận**

Trong đồ án “Toxic Comment Detection” lần này, nhóm Nopen đã được thực hiện xây dựng bộ dữ liệu cho việc phân loại tính tiêu cực trong các bình luận trên các bài đăng tiếng Việt trên mạng xã hội Facebook do nhận thấy rằng đã có nhiều bộ dữ liệu tiếng Anh đã dùng để phục vụ cho bài toán này [8], và nhóm tham khảo được hai bài báo đã thực hiện tác vụ này cho tiếng Việt [1] [2], là các nghiên cứu công khai hiếm hoi mà nhóm thấy đã xây dựng để phục vụ cho bài toán này. Nhóm đã đề xuất các định nghĩa cho khái niệm Non-Toxic, Toxic, Very-Toxic của bình luận và xét tỉ lệ đồng thuận giữa các thành viên trong nhóm để cải thiện định nghĩa, guideline để gán nhãn cho bộ dữ liệu. Để thực hiện phân loại, nhóm đã thực hiện các bước tiền xử lý gồm 4 bước cơ bản đã tham khảo từ 2 bài báo đã nêu và đã tham khảo từ bài tập quá trình thực hiện trước đó. Nhóm cũng đã tiến hành các thử nghiệm và cho thấy hiệu năng của chương trình đã được cải thiện với các bước tiền xử lý này. Sau đó, nhóm đã sử dụng 6 mô hình để phân loại các bình luận và rút ra được kết quả mô hình Transfer Learning cho kết quả phân lớp tốt nhất ở bài toán này. Cuối cùng, nhóm đã thực hiện đánh giá và phân tích lỗi của các dự đoán sai này. Từ đó, có thể đưa ra các hướng phát triển mới hơn cho bài toán này.

## Chương 4 : TÀI LIỆU THAM KHẢO

- [1] N. T. Luan, N. V. Kiet và N. L. T. Ngan, “Constructive and Toxic Speech Detection for Open-domain Social Media Comments in Vietnamese,” 2021.
- [2] H. G. Phu, L. C. Canh, T. Q. Khanh, N. V. Kiet và N. L.-T. Ngan, “ViHOS: Hate Speech Spans Detection for Vietnamese,” trong *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023.
- [3] V. Thanh, N. Q. Dat, N. Q. Dai, D. Mark và J. Mark, “VnCoreNLP: A Vietnamese Natural Language Processing Toolkit,” trong *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, NAACL 2018*, pages 56-60, 2018.
- [4] D. Q. N. T. V. M. D. a. M. J. Dat Quoc Nguyen, “A Fast and Accurate Vietnamese Word Segmenter,” trong *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*, pages 2582-2587, 2018.
- [5] T. V. D. Q. N. M. D. a. M. J. Dat Quoc Nguyen, “From Word Segmentation to POS Tagging for Vietnamese,” trong *Proceedings of the 15th Annual Workshop of the Australasian Language Technology Association, ALTA 2017*, pages 108-113, 2017.
- [6] V.-D. Le, “vietnamese-stopwords,” 2015. [Trực tuyến]. Available: <https://github.com/stopwords/vietnamese-stopwords>.
- [7] D. Q. N. a. A. T. Nguyen, “PhoBERT: Pre-trained language models for Vietnamese,” trong *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, 2020, pp. 1037-1042.
- [8] J. R. a. R. Krestel, “Toxic Comment Detection in Online Discussions”.
- [9] T. M. TIẾN, “[PhoBERT] Classification for Vietnamese Text,” 07 April 2022. [Trực tuyến]. Available: <https://www.kaggle.com/code/trnmtin/phobert-classification-for-vietnamese-text>. [Đã truy cập 20 June 2023].