ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

MÔN HỌC: MÁY HỌC

LỚP: CS114.N21.KHCL

Vietnamese Hate and Offensive Detection in Gaming Domain

Sinh viên:
Lê Nguyễn Minh Huy - 20521394
Lê Gia Khang
Võ Huy Hoàng
Phạm Thị Trâm Anh - 21520146

Giảng viên:
Phạm Nguyễn Trường An

THÀNH PHỐ HỒ CHÍ MINH, 15 THÁNG 7 NĂM 2023

1 Tóm tắt đồ án

Trong đồ án lần này, nhóm đã thực hiện một số khảo sát trong bài toán Hate Speech Detection trong tiếng Anh và tiếng Việt, từ các khảo sát đó, nhóm đã có các đóng góp sau:

- Xây dựng bộ dữ liệu cho bài toán Hate Speech Detection trong lĩnh vực game live streaming (với quy trình xây dựng chi tiết được nêu trong mục III).
- Xây dựng quy trình tiền xử lý dữ liệu nghiêm ngặt và hiệu quả để làm sạch các bình luận được thu thập từ các stream (chi tiết ở mục IV). Các kỹ thuật này cải thiện được chất lượng dữ liệu và cải thiện độ chính xác của dự đoán (khi so sánh với quy trình tiền xử lý dữ liệu trong bộ dữ liệu Vi-HSD).
- Thử nghiệm một số mô hình khác nhau thuộc các hướng tiếp cận khác nhau của bài toán HSD trên bộ dữ liệu được xây dựng, kết quả cho thấy phương pháp mới của nhóm đề xuất (kết hợp Transfer Learning và Deep Learning cho kết quả tốt nhất trên bộ dữ liệu mới này). Chi tiết các mô hình cụ thể và cách cài đặt xem ở mục V.

Link github chứa dataset và sourecode: **https://github.com/polieste/CS114.N21.KHCL**

2 Nhận xét giảng viên

- Báo cáo chưa đúng format (ở trang bìa và trang tóm tắt đồ án)
- Cần thực hiện so sánh giữa quy trình tiền xử lý dữ liệu có sẵn và tiền xử lý dữ liệu do nhóm đề xuất

3 Bổ sung của nhóm

- Chỉnh sửa trang bìa báo cáo và thêm trang tóm tắt đồ án
- Thêm phần so sánh giữa 2 quy trình tiền xử lý dữ liệu (chi tiết ở mục VI.B)

Vietnamese Hate and Offensive Detection in the Gaming Domain

Minh-Huy Le-Nguyen, Gia-Khang Le, Tram-Anh Pham-Thi, Huy-Hoang Vo
Trường Đại học Công nghệ Thông tin, Thành phố Hồ Chí Minh, Việt Nam
Đại học Quốc gia Hồ Chí Minh, VNU-HCM
{20521394, 21522189, 21520146, 21522103}@gm.uit.edu.vn

***Tóm tắt nội dung*—Với sự phát triển mạnh mẽ của mạng xã hội hiện nay, đặc biệt là các nền tảng live-streaming game, việc ngăn chặn những nội dung gây thù ghét và xúc phạm để xây dựng một môi trường mạng an toàn, lành mạnh là cần thiết hơn bao giờ hết. Tuy nhiên, các nghiên cứu hiện tại trong lĩnh vực này vẫn đối mặt với một số hạn chế như sau: kỹ thuật tiền xử lý thiếu sót, không sử dụng các mô hình hiệu suất tốt nhất. Bài báo này tập trung vào việc xây dựng một hệ thống thông minh có thể giải quyết những hạn chế này. Đầu tiên, chúng tôi đề xuất một bộ dữ liệu mới tập trung vào các bình luận trong lĩnh vực live-streaming game. Thứ hai, chúng tôi áp dụng bộ dữ liệu trên với các mô hình sota hiện nay cũng như một số mô hình đã được sử dụng trong các bài báo trước đó. Ngoài ra, chúng tôi còn xây dựng một kỹ thuật tiền xử lý hiệu quả để làm sạch các bình luận được thu thập. Kết quả thực nghiệm cho thấy các mô hình Transfer Learning cho kết quả tốt hơn so với các mô hình Machine Learning và Deep Learning, mô hình PhoBERT-CNN kết hợp cho ra kết quả tốt nhất với 90.00 F1-micro.**

***Index Terms*—Hate Speech Detection, Sentiment analysis, Transformer, Streaming Data.**

I. Introduction

DESCRIBE: Sự bùng nổ nhanh chóng của mạng xã hội đã đáng kinh ngạc thay đổi cuộc sống hàng ngày của chúng ta. Trong tình hình đó, an ninh trên không gian mạng là một vấn đề trực tiếp ảnh hưởng đến cuộc sống của người dùng, đặc biệt là đối tượng như trẻ em hoặc những người dễ tổn thương. Theo báo cáo của Mohan et al. [1], môi trường mạng xã hội, nơi có nhiều nội dung độc hại như những bình luận gây thù ghét, tin giả, nội dung vi phạm các tiêu chuẩn cộng đồng, không chỉ ảnh hưởng đến một tỷ lệ lớn người dùng mà còn đối với các người kiểm duyệt. Bài viết gây thù hận thường được mô tả là bất kỳ phát ngôn nào xúc phạm một người hoặc nhóm dựa trên bất kỳ phương diện nào như chủng tộc, màu da, dân tộc, giới tính, tình dục, quốc tịch, tôn giáo hoặc đặc điểm khác. Dưới đây là một số ví dụ về những bình luận gây thù ghét và xúc phạm được đăng trên mạng xã hội tiếng Việt:"cứ phải chửi cho mới chịu im :)))", "lũ bắc kỳ", "hài vcl". Tuy nhiên, việc kiểm duyệt những bình luận gây thù ghét và xúc phạm trên mạng xã hội đối mặt với nhiều thách thức do số lượng và đa dạng của chúng, cả về mức độ và chủ đề. Theo nghiên cứu của Suha Abu et al. [2], có 293.000 bài đăng được đăng mỗi 60 giây trên nền tảng mạng xã hội với hàng tỷ người dùng, Facebook, và có hơn 510.000 bình luận được viết. Hơn nữa, theo trang thống kê uy tín Statista[1], Facebook đã phải loại bỏ hơn 11,3 triệu nội dung vi phạm và gây thù ghét trên toàn cầu trong năm 2018. Năm 2019, YouTube cũng loại bỏ hơn 1.800 triệu bình luận vi phạm các tiêu chuẩn cộng đồng, và số liệu này đã tăng mạnh trên cả hai nền tảng. Trong năm 2020, Facebook phải xóa hơn 81 triệu bài viết gây thù ghét và xúc phạm, tăng gấp bảy lần so với năm 2018. Trong khi YouTube phải loại bỏ hơn 4.800 triệu bình luận vào năm 2020, gấp ba lần con số năm 2019.

Bên cạnh đó, cho dù các công ty công nghệ đã có những nghiên cứu liên quan đến việc nhằm ngăn chặn các bình luận tiêu cực, thù ghét trên mạng xã hội của họ, những nghiên cứu này có thể sử dụng trên đa ngôn ngữ [3]. Nhưng đối với những nội dung thiếu ngữ cảnh, văn hóa chuyên biệt bản địa và chậm cập nhật với sự phát triển liên tục của nội dung gây thù ghét, các hệ thống này gặp khó khăn khá lớn trong việc xử lý dữ liệu từ các ngôn ngữ cụ thể.

Vì vậy, nghiên cứu của chúng tôi sẽ đóng góp cho bài toán Hate Speech Detection (HSD) trong tiếng Việt. Chúng tôi đã sử dụng các kỹ thuật tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên để phân loại các bình luận gây thù ghét và xúc phạm trên mạng xã hội, tạo ra một không gian trực tuyến lành mạnh và an toàn hơn. Các đóng góp chính của nghiên cứu của chúng tôi được tóm tắt như sau. Source code và data được lưu trong đường dẫn sau:

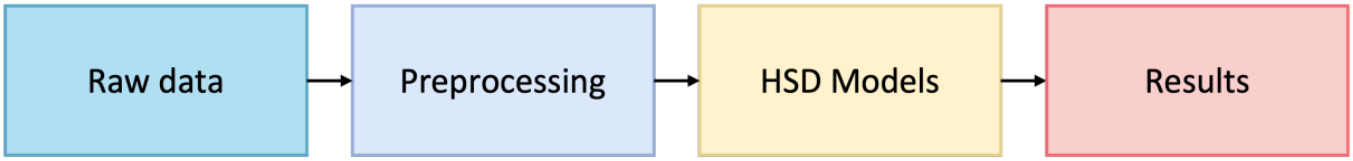
1) Chúng tôi xây dựng một bộ dữ liệu cho HSD trong lĩnh vực game live-streaming (quy trình xây dựng xem chi tiết ở mục III).
2) Chúng tôi thực hiện các kỹ thuật tiền xử lý dữ liệu nghiêm ngặt và hiệu quả để làm sạch các bình luận được thu thập từ các stream (chi tiết các kỹ thuật ở mục IV). Các kỹ thuật này cải thiện chất lượng dữ liệu và cải thiện đáng kể việc trích xuất thông tin trước khi huấn luyện mô hình.
3) Chúng tôi thử nghiệm các phương pháp khác nhau trên bộ dữ liệu được xây dựng: phương pháp machine learning, phương pháp deep learning, phương pháp transfer learning và phương pháp kết hợp giữa trasfer learning và deep learning. Chi tiết các mô hình cụ thể cho từng phương pháp xem ở mục V.

II. Fundamental of Hate Speech Detection on Streaming Data

A. Introduction to hate speech detection task

Hate speech detection và sentiment analysis có sự liên kết chặt chẽ [4], và những bài toán này gần đây đã trở thành các

[1]https://www.statista.com/statistics/1013804/facebook-hate-speech-content-deletion-quarter/



Hình 1: *Quy trình xử lý bài toán HSD.*

chủ đề phổ biến trong Xử lý Ngôn ngữ Tự nhiên. Trong phần này, chúng tôi mô tả về bài toán HSD [5], [6]. Bài toán này nhằm xác định xem một bình luận trên mạng xã hội có phải là THÙ GHÉT (HATE), XÚC PHẠM (OFFENSIVE) hay SẠCH (CLEAN) hay không. Cụ thể, bài toán được mô tả như sau.

Input: Bình luận tiếng Việt trên các trang mạng xã hội.

Output: Dự đoán một trong ba nhãn bằng bộ phân loại.

THÙ GHÉT(HATE) bao gồm việc sử dụng ngôn từ lăng mạ, thường mang mục đích xỉ nhục cá nhân hoặc nhóm và có thể chứa nội dung thù ghét, xúc phạm và gây tổn hại. Một bài đăng hoặc bình luận được xác định là THÙ GHÉT nếu nó:

- Nhắm vào cá nhân hoặc nhóm dựa trên đặc điểm của họ.
- Thể hiện ý định rõ ràng gây hại hoặc lan truyền sự thù ghét.
- Có thể sử dụng từ ngữ xúc phạm hoặc lăng mạ hoặc không.

XÚC PHẠM nhưng không phải THÙ GHÉT(OFFENSIVE) là một bình luận có thể chứa các từ ngữ phản cảm, nhưng không nhắm vào cá nhân cụ thể.

Không phải xúc phạm cũng không phải thù ghét (CLEAN) là một bài viết bình thường, có cuộc trò chuyện, thể hiện cảm xúc một cách bình thường và không chứa ngôn từ thù ghét hoặc phẩm cảm.

B. Existing HSD models

Đã có một số nguyên cứu trên thế giới về bài toán HSD như Waseem et al. [7], Chen et al. [8], và Davidson et al. [9], áp dụng hệ thống HSD nhằm giải quyết vấn đề thực tế về hate speech trên mạng xã hội như Twitter. Tuy nhiên, việc giải quyết bài toán HSD trong tiếng Việt hiện nay vẫn còn khiêm tốn, với chủ yếu các nghiên cứu xoay quanh 2 tập dữ liệu đặc trưng là ViHSD [6] và tập dữ liệu HSD-VLSP [5].

Có nhiều phương pháp để giải quyết bài toán HSD, với các mô hình học máy là phương pháp cơ bản nhất. Mô hình Support Vector Machine và Random Forest được áp dụng trong nghiên cứu của Davidson et al. và Martins et al. [10], kết quả của nghiên cứu này đóng vai trò là nền tảng cho sự phát triển của các phương pháp khác trong tương lai. Trong những năm gần đây, đã xuất hiện các giải pháp SOTA(state-of-the-art) với hiệu suất đáng chú ý, cụ thể là việc phát triển các mô hình đơn ngôn ngữ và đa ngôn ngữ như BERT [11], RoBERTa [12], XLM-R [13], và kết hợp để tạo ra các mô hình cao cấp hơn như BERT-CNN [14], tạo cơ hội cho việc cải thiện hiệu suất cho bài toán HSD.

Lấy cảm hứng từ mô hình BERT, mô hình PhoBERT ra đời và đã được pre-trained trên tiếng Việt, PhoBERT có nhiệm vụ trích xuất các đặc trưng từ câu văn cho đầu vào của mô hình phân loại khác. PhoBERT được giới thiệu bởi Nguyen et al. vào năm 2020 [15]. PhoBERT được huấn luyện trên khoảng 20GB dữ liệu, bao gồm khoảng 1GB từ bộ văn bản Wikipedia tiếng Việt và phần còn lại 19GB từ bộ văn bản tin tức tiếng Việt. Kiến trúc của PhoBERT tương tự mô hình RoBERTa được phát triển bởi Liu et al. tại Facebook (mô hình được tối ưu hóa từ mô hình BERT với một lượng lớn dữ liệu huấn luyện lên đến 160GB và tăng gấp 10 lần so với BERT). Hơn nữa, khi áp dụng cho tiếng Việt, PhoBERT đã được chứng minh có hiệu suất và mang lại kết quả tốt hơn so với mô hình đa ngôn ngữ tốt nhất hiện tại là mô hình XLM-R.

III. Dataset Creation

A. Data Preparation

Chúng tôi thu thập những bình luận của người dùng khi xem live-stream về các trò chơi phổ biến: Liên minh huyền thoại, Dota2, CSGO, Valorant, Liên Quân Mobile,... từ các trang Facebook và video YouTube tiếng Việt khác nhau. Chúng tôi lựa chọn các trang Facebook và kênh YouTube có tỷ lệ tương tác cao và không hạn chế bình luận hoặc ít hạn chế. Sau khi thu thập dữ liệu, chúng tôi loại bỏ các bình luận bị trùng lặp nhiều lần do tính chất của bình luận khi live-stream.

B. Annotation Guidelines

Annotation guideline của chúng tôi được lấy cảm hứng từ bộ dữ liệu Vi-HSD. Bộ dữ liệu của chúng tôi gồm ba nhãn: HATE, OFFENSIVE và CLEAN. Mỗi annotator gán nhãn cho mỗi bình luận trong bộ dữ liệu. Trong bộ dữ liệu, chúng tôi có hai nhãn cho các bình luận mang tính chất thù ghét, và một nhãn chỉ cho các bình luận bình thường. Ý nghĩa chi tiết về ba nhãn và các ví dụ cho mỗi nhãn được mô tả trong Bảng I. Thực tế, nhiều bình luận trong bộ dữ liệu được viết dưới hình thức không chính thức. Bình luận thường chứa viết tắt như ntn (như thế nào), r (rồi) trong **Bình luận 1** và lol (âm đạo) trong **Bình luận 4**, và ngôn ngữ lóng như phế (tệ) trong **Bình luận 2** và lol (âm đạo) trong **Bình luận 4**.

C. Data Creation Process

Quy trình gán nhãn của chúng tôi bao gồm hai giai đoạn chính như trong Hình 2. Giai đoạn đầu tiên là giai đoạn đào tạo, những annotator sẽ được cung cấp hướng dẫn và ví dụ cho mỗi nhãn. Sau đó, chúng tôi tính toán sự động thuận giữa các annotator bằng chỉ số Cohen Kappa (k) [16]. Nếu sự đồng thuận giữa các annotator chưa đạt được mức đủ tốt, chúng tôi sẽ đào tạo lại annotator và cập nhật lại hướng dẫn gán nhãn nếu cần thiết. Sau khi tất cả các annotator đã được đào tạo tốt, chúng tôi tiến hành giai đoạn gán nhãn. Giai đoạn gán

Bảng I: Annotation guidelines cho việc gán nhãn trong bài toán HSD

Label	Discription	Example
CLEAN	Bình luận không có bất kỳ hành vi thù ghét, xúc phạm, quấy rối nào	**Bình luận 1**: ntn r anh em? (Bình luận này hoàn toàn sạch, không có ý xúc phạm hay tấn công bất kỳ ai)
OFFENSIVE	Bình luận có tạo sự khó chịu, thậm chí là thô tục, nhưng không nhắm đến cá nhân hay tổ chức cụ thể nào.	**Bình luận 2**: Phế điên (Bình luận này chứa từ "điên" mang ý xúc phạm, tuy nhiên không nhắm đến một cá nhân hay tổ chức nào). Ngoài ra, "phế" cũng là một từ lóng, nghĩa là "tệ"
HATE	Bình luận chứa nội dung gây khó chịu, quấy rối và nhắm trực tiếp vào một cá nhân, tổ chức cụ thể dựa trên đặc điểm riêng của họ. Có một số trường hợp cụ thể như sau: - Bình luận chứa những từ ngữ xúc phạm và tấn công vào một đối tượng cụ thể như một cá nhân, một cộng đồng, một quốc gia hoặc một tôn giáo. Trường hợp này dễ dàng được xác định là thù ghét. - Bình luận chứa sự phân biệt về chủng tộc, vùng miền, giới tính, quấy rối, tuy nhiên không chứa từ ngữ rõ ràng. - Bình luận chứa sự phân biệt, quấy rối nhưng được thể hiện dưới dạng ẩn dụ. Để xác định bình luận này, annotator cần có kiến thức cụ thể	**Bình luận 3**: Kênh chat trẩu :)) (Bình luận nhắm tới một nhóm người với ý nghĩa xúc phạm) **Bình luận 4**: sp hay thế mà ad thì ngu lol (Bình luận chứa từ tục tĩu, bên cạnh đó còn nhắm tới đối tượng "ad")

nhãn của chúng tôi được lấy cảm hứng từ IEEE[2]. Trong đó, hai annotator gán nhãn toàn bộ bộ dữ liệu. Nếu có bất kỳ nhãn khác nhau nào giữa hai annotator đó, chúng tôi để cho annotator thứ ba gán nhãn những nhãn không đồng thuận đó. Annotator thứ tư sẽ gán nhãn lại nhãn đó nếu cả ba người trước không đồng thuận. Nhãn cuối cùng được xác định bằng phương pháp bỏ phiếu. Bằng cách này, chúng tôi đảm bảo rằng mỗi bình luận được gán bằng một nhãn và tính khách quan cho mỗi bình luận. Do đó, tổng thời gian bỏ ra để gán nhãn là ít hơn so với việc bốn annotator làm cùng một lúc.

D. Dataset Evalutation and Discussion

Chúng tôi ngẫu nhiên lấy 200 bình luận từ tập dữ liệu và giao cho bốn annotator khác nhau, được ký hiệu là A1, A2, A3 và A4, để thực hiện việc gán nhãn. Bảng II cho thấy sự đồng thuận giữa các cặp annotator. Sau đó, chúng tôi tính toán sự đồng thuận trung bình giữa các annotator. Sự đồng nhất giữa các annotator cho toàn bộ tập dữ liệu là k = 0.78. Tập dữ liệu được thu thập từ mạng xã hội nên chúng chứa nhiều từ viết tắt, từ không thông dụng, ngôn ngữ địa phương và ý nghĩa hình tượng. Do đó, điều này gây khó khăn cho annotator. Ví dụ, Bình luận 1 chứa cụm từ "ntn", có nghĩa là "như thế nào", và Bình luận 3 trong Bảng I có từ chỉ đối tượng "ad", có nghĩa là một người chơi trong game. Giả sử hai annotator gán nhãn cho Bình luận 3 và nó chứa từ "ad", annotator thứ nhất đã biết về từ này trước đó, vì vậy annotator gán bình luận này là hate, annotator thứ hai gán nhãn bình luận này là offensive vì annotator không hiểu từ này. Ví dụ tiếp theo là từ "trẩu" trong Bình luận 3 trong Bảng I. Hai annotator gán nhãn cho Bình luận 3, annotator thứ nhất không hiểu ý nghĩa thực sự của cụm từ đó, vì vậy annotator gán bình luận này là clean. Ngược lại, annotator thứ hai biết ý nghĩa thực sự của cụm từ này, vì vậy annotator đánh dấu nó là hate. Mặc dù hướng dẫn đã có định nghĩa rõ ràng về các nhãn CLEAN, OFFENSIVE và HATE, quá trình gán nhãn chủ yếu bị ảnh hưởng bởi kiến thức và chủ quan của annotator. Do đó, việc đào tạo lại annotator và cải thiện hướng dẫn liên tục là cần thiết để nâng cao chất lượng của annotator và sự đồng nhất giữa các annotator.

[2]https://ieeexplore.ieee.org/document/7902269

Bảng II: Confusion matrix khi tính bằng độ đo Cohen Kappa giữa 4 annotator

	A1	A2	A3	A4
A1	-	0.83	0.74	0.76
A2	-	-	0.83	0.79
A3	-	-	-	0.75
A4	-	-	-	-

Bảng III: Một số ví dụ trích ra từ bộ dữ liệu

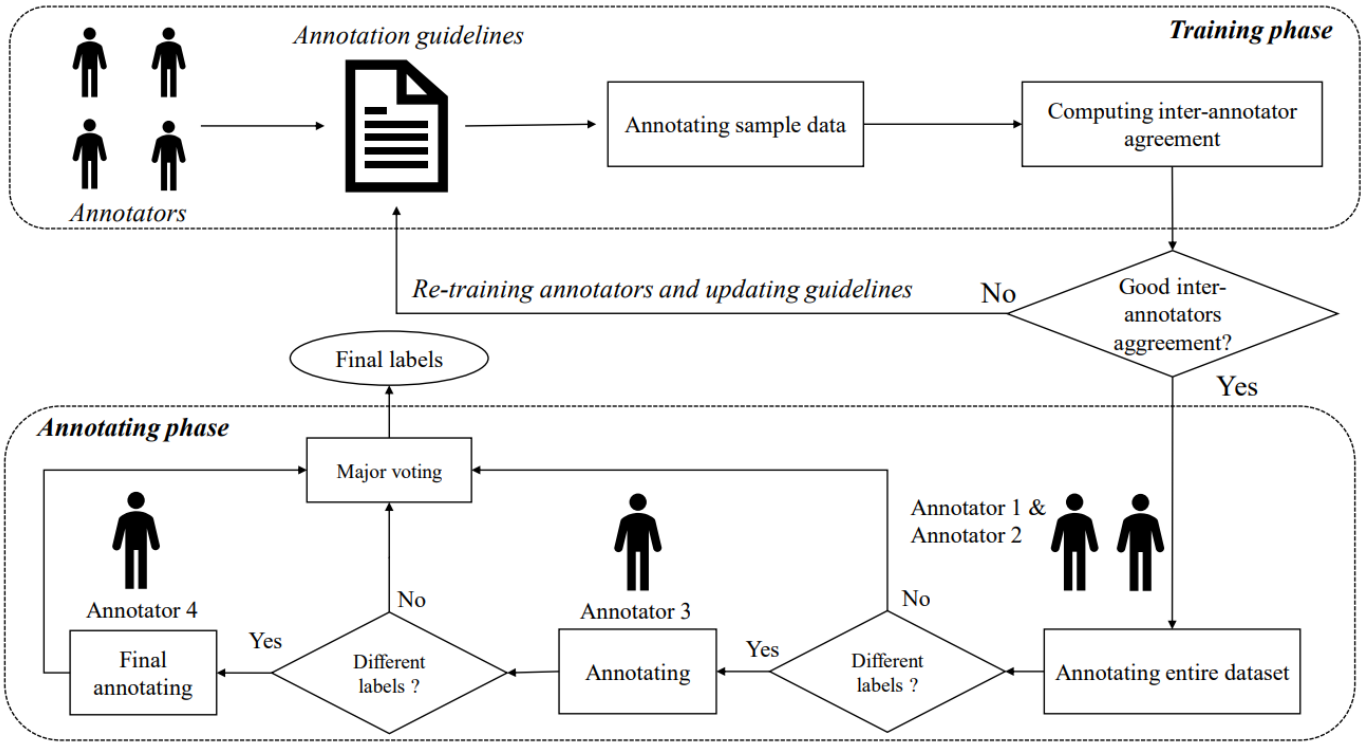
#	Comment	Label
1	đứng đó làm con cặc gì	1
2	mày lên đây làm gì thế ashe	0
3	ad ngu vậy	2
4	con ẳng làm gì	1
5	lo xong cái thân con đi	0
6	tí tuổi đầu đã dám cãi lại bố rồi	0
7	sp hay thế mà ad thì ngu lol	2

E. Dataset Overview

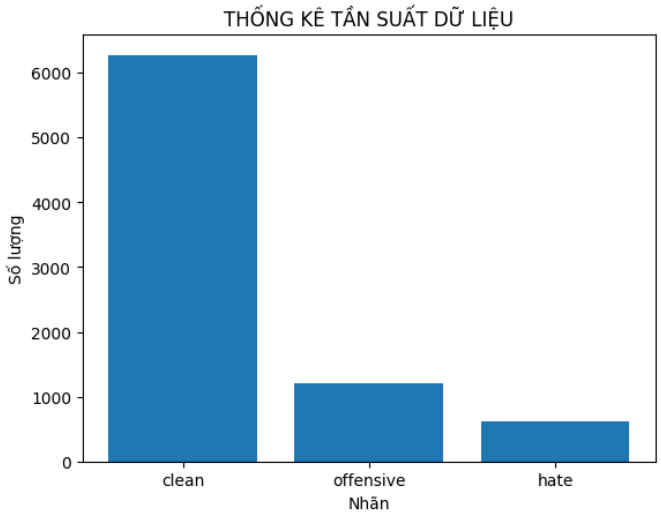
Tập dữ liệu bao gồm 8093 bình luận. Mỗi bình luận được gán nhãn là CLEAN (0), OFFENSIVE (1) và HATE (2) thể hiện trong Hình 3. Bảng III hiển thị một số ví dụ từ tập dữ liệu. Sau đó, chúng tôi chia tập dữ liệu thành ba phần riêng biệt: tập huấn luyện (train), tập phát triển (dev) và tập kiểm tra (test), với tỷ lệ 7-1-2. Phân phối nhãn dữ liệu trên tập huấn luyện, tập phát triển và tập kiểm tra là như nhau và dữ liệu lệch về nhãn CLEAN.

IV. Data Preprocessing

Bởi vì tập dữ liệu được thu thập từ các trang mạng xã hội, nên chúng chứa các bình luận phức tạp và đa dạng. Đặc biệt, có rất nhiều bình luận trong tập dữ liệu chứa các ký tự không chuẩn unicode, teencode, từ viết tắt và từ có ký tự lặp lại. Do đó, chúng tôi tiến hành xây dựng quy trình tiền xử lý dữ liệu



Hình 2: *Quy trình gán nhãn cho dataset.*



Hình 3: *Phân phối các nhãn trong dataset.*

để cải thiện chất lượng của các tập dữ liệu, nhằm trích xuất các đặc trưng có giá trị trước khi sử dụng chúng để huấn luyện các mô hình phân loại. Hình 4 mô tả tổng quan về quy trình tiền xử lý dữ liệu gồm hai phase.

A. Phase 1

Chuyển thành chữ thường: Tất cả các ký tự trong các bình luận trong tập dữ liệu được chuyển thành chữ thường. Chúng tôi thực hiện điều này để tránh việc Python hiểu hai từ nhưng được viết hoa và viết thường là 2 từ khác nhau.

Xóa khoảng trắng dư thừa: Người dùng trên mạng xã hội thường không biết hoặc biết rằng họ gõ nhiều khoảng trắng trong các bình luận của mình. Do đó, chúng tôi quyết định loại bỏ những khoảng trắng dư thừa này.

Xóa đường liên kết: Chúng tôi tin rằng các liên kết website trong bình luận không ảnh hưởng đến cảm xúc của bình luận. Vì vậy, chúng tôi quyết định loại bỏ tất cả chúng.

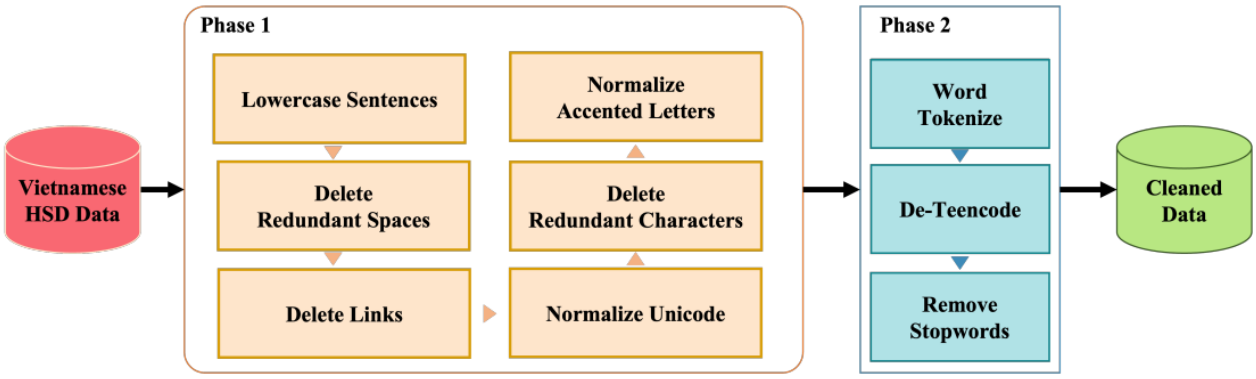
Chuẩn hóa Unicode: Chúng tôi cũng thấy rất nhiều từ tiếng Việt trong tập dữ liệu có cùng nghĩa nhưng Python phát hiện chúng là khác nhau do Unicode khác nhau. Lý do là có nhiều định dạng Chuyển đổi Unicode (UTF) như UTF-8, UTF-16, UTF-32 được sử dụng rộng rãi, nhưng lựa chọn của chúng tôi là chuẩn hóa thành UTF-8.

Xóa các ký tự dư thừa: Chúng tôi loại bỏ các ký tự dư thừa mà người dùng tạo ra cố ý. Lấy ví dụ về quá trình xóa các ký tự dư thừa: từ "vuiiii", có nghĩa là vui, và từ "vllll" là một mã teen xúc phạm trong tiếng Việt nhưng không có ý nghĩa rõ ràng trong tiếng Anh, sau khi loại bỏ các ký tự dư thừa, chúng trở thành "vui" và "vl". Tuy nhiên, các ký tự trùng lặp trong từ "call" sẽ không được loại bỏ vì đây là một từ có nghĩa trong tiếng Anh.

Sau khi thực hiện xong các bước ở Phase 1, output của Phase 1 sẽ được đưa tiếp qua Phase 2.

B. Phase 2

Word Tokenize: Câu nhập vào được chia thành các từ hoặc cụm từ có ý nghĩa. Để làm điều này, chúng tôi đã sử dụng word segmenter của VnCoreNLP [17] cho mô hình PhoBERT cũng như cho các mô hình khác. Bởi vì các bình luận trong tập dữ liệu là văn bản thô, việc segment từ là cần thiết để chuẩn bị



Hình 4: *Quy trình tiền xử lý dữ liệu.*

Bảng IV: Một số từ teen code được trích từ bộ dữ liệu.

#	Teencode	True word
1	đc	được
2	vl	vãi lồn
3	j z	gì vậy
4	cmm	con mẹ mày

dữ liệu cho việc huấn luyện các mô hình. Hơn nữa, PhoBERT sử dụng word segmenter VnCoreNLP RDRSegmenter để tiền xử lý dữ liệu huấn luyện trước (bao gồm word segment và câu tiếng Việt), do đó, việc sử dụng cùng bộ word segmentation sẽ dễ dàng hơn.

De-Teencode: Trên mạng xã hội, mọi người thường sử dụng hình thức viết tắt của từ để gõ nhanh hơn. Một số người sử dụng những từ viết tắt này để tránh bị hệ thống phát hiện khi chửi thề. Hơn nữa, những từ viết tắt này cũng có tên của chúng trong tiếng Việt, được gọi là teen code. Vì vậy, để giúp các mô hình của chúng tôi hiểu tốt hơn các câu nhập vào, chúng tôi phải ánh xạ những teen code đó thành các từ gốc của chúng. Hơn nữa, quá trình ánh xạ teen code, chúng tôi gọi là De–teencode, và Bảng IV dưới đây cho thấy một số ví dụ về chúng.

Loại bỏ stopwords: Chúng tôi cũng loại bỏ các stopwords của các bình luận vì chúng không có ý nghĩa. Trong các thí nghiệm của chúng tôi, chúng tôi sử dụng từ điển stopwords tiếng Việt [42] để loại bỏ các từ dừng trong câu.

Trong Phase 2, dữ liệu được segment, de-teencode và loại bỏ stopwords. Các bước được thực hiện theo thứ tự đó vì đầu ra của word segment là một danh sách các từ, cụm từ và ký tự riêng biệt được phân tách bằng khoảng trắng. Sau đó, các ký tự này được kiểm tra xem chúng có phải là teen code và sẽ được de-teencode trong bước tiếp theo. Do đó, việc thực hiện bước de-teencode sau bước phân đoạn từ là một quyết định hợp lý. Cuối cùng, sau bước de-teencode, chúng tôi loại bỏ tất cả stopwords, và lý do chúng tôi loại bỏ stopwords sau bước de-teencode là những teen code này có thể cũng là stopwords.

V. Model

A. Machine Learning Approach(ML)

SVM: SVM [18] (Support Vector Machine) là một phương pháp học có giám sát, có thể sử dụng cho cả việc phân loại và hồi quy. SVM lần đầu tiên được biết đến vào năm 1992 được giới thiệu bởi Boser, Guyon và Vapnik trong COLT-92. SVM là một mô hình phân loại và hồi quy sử dụng lý thuyết học máy để tối đa hóa độ dự đoán chính xác và tránh "over-fit" dữ liệu. SVM hoạt động dựa trên việc tìm ra "hyperplane" trong không gian đa chiều, nơi các điểm dữ liệu được phân chia thành hai lớp khác nhau. SVM trở nên nổi tiếng khi sử dụng "pixel maps" như đầu vào.

Random Forest: Random forest [19] (RF) là một phương pháp học máy được phát triển bởi Leo Breiman và Adele Cutler vào năm 2001. Decision Tree là một mô hình khá nổi tiếng hoạt động trên cả hai lớp bài toán phân loại và học có giám sát dù độ chính xác khá cao những Decision Tree vẫn tồn tại những hạn chế lớn vì thế sự ra đời của Random Forest là sự kết hợp của nhiều Decision Tree sẽ nâng cao độ chính xác, giảm thiểu "over-fit". Random Forests là tập hợp các cây,trong đó việc xây dựng các cây là ngẫu nhiên, sau khi xây dựng một tập hợp gồm các cây, Random Forests thực hiện dự đoán bằng cách lấy trung bình dự đoán của các cây riêng lẻ. Cây ngẫu nhiên thường đưa ra những dự đoán chính xác và ổn định.

XGBoost: XGBoost [20] (Extreme Gradient Boosting) là một thuật toán nổi tiếng được sử dụng rộng rãi trong các cuộc thi và ứng dụng nhiều trong thực tế, được giới thiệu lần đầu bởi Tianqi Chen vào năm 2014. Là một phương pháp boosting mạnh mẽ tập trung vào việc xây dựng các Decision Tree để tìm ra mối quan hệ giữa các đặc trưng và đầu ra dự đoán.

B. Deep Learning Approach(DL)

Text CNN: CNN (Convolutional Neural Network) là một mô hình mạng nơ-ron nhân tạo được đề xuất bởi Yann LeCun, Leon Bottou, Yoshua Bengio và Patrick Haffner vào năm 1998. Mạng nơ-ron tích chập sử dụng các lớp có bộ lọc tích chập được áp dụng cho các đặc trưng cục bộ. Ban đầu được phát minh cho thị giác máy tính, các mô hình CNN sau đó đã được chứng minh là hiệu quả trong xử lý ngôn ngữ tự nhiên (NLP). Vào năm 2014, mô hình Text CNN [21] (Convolutional Neural Network cho xử lý văn bản) được đề xuất bởi Yoon Kim trong bài báo "Convolutional Neural Networks for Sentence Classification". Mô hình này đã trở thành một công cụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và đã được áp dụng rộng rãi trong nhiều tác vụ như phân loại văn bản, phân

loại cảm xúc, dự đoán câu hỏi và nhiều ứng dụng khác liên quan đến xử lý ngôn ngữ.

C. Transfer Learning Approach(TL)

1) XLM-R: XLM-RoBERTa được Conneau xây dựng vào tháng 11/2019 dựa trên cấu trúc transformer và là một biến thể của mô hình RoBERTa. XLM-R sử dụng kỹ thuật đào tọa tự giám sát và được huấn luyện trước cho các tác vụ liên quan tới văn bản trên hơn 100 ngôn ngữ khác nhau. Khối lượng dữ liệu huấn luyện cho mô hình này lên tới 2.5TB, trong đó lượng dữ liệu tiếng Việt được sử dụng là 137GB. Hiện nay, mô hình XLM-R cung cấp hai phiên bản là base và large với số tham số lần lượt là 250 triệu và 560 triệu. Ngoải ra mô hình còn sử dụng cơ chế Masked Language Model - MLM và không có tác vụ dự đoán câu tiếp theo, mô hình này đã được chứng minh là cải thiện hiệu suất hoạt động đáng kể so với các biển thể của BERT.

2) PhoBERT: PhoBERT được biết đến là mô hình ngôn ngữ đầu tiên dành riêng cho tiếng Việt được xây dựng vào năm 2020. Mô hình có cấu trúc, cách tiếp cận tương tự mô hình RoBERTa và được huấn luyện trước trên 20GB dữ liệu tiếng Việt, khoảng 1GB Vietnamese Wikipedia và 19GB còn lại từ Vietnamese news corpus. Do có cấu trúc tương tự BERT, PhoBERT cũng có 2 phiên bản là base và large với số lượng transformers block tương ứng lần lượt là 12 và 24. Ngoài ra, với đặc trưng sử dụng RDRSegmenter của VNCoreNLP để tách từ cho dữ liệu đầu vào trước khi qua lớp BPE encode, các token văn bản đầu vào có sự khác biệt so với các mô hình huấn luyện đa ngôn ngữ khác. Hiện nay, PhoBERT được biết đến là mô hình ngôn ngữ hiện đại nhất dành cho người Việt để xử lý các tác vụ xử lý ngôn ngữ tự nhiên.

D. Combine

PhoBERT + Text CNN: Nhiều mô hình kết hợp BERT và CNN đã được sử dụng rộng rãi gần đây để phân loại các văn bản ngắn được thu thập từ mạng xã hội, cụ thể như bài toán phân loại bình luận xúc phạm và thù ghét, đạt được nhiều kết quả triển vọng. Trong bài nghiên cứu của chúng tôi, mô hình kết hợp PhoBERT và CNN đã được sử dụng để đánh giá độ hiệu quả đối với bài toán phân loại bình luận trong tiếng Việt. Nhờ vào cơ chế cộng hưởng của 2 mô hình giúp làm giảm độ lỗi giữa nhãn dự đoán và nhãn thực tế, PhoBERT-CNN đã vượt qua những mô hình khác trong bài toán phân loại bình luận tiếng Việt, đặc biệt trong bộ dataset của chúng tôi. Hình 5 cho thấy tổng quan về hướng tiếp cận của chúng tôi trong bài toán này.

Chúng tôi sử dụng PhoBERT để thực hiện word embedding để trích xuất thông tin từ data. Mô hình PhoBERT được chọn vì vượt trội hơn những mô hình pretrained đơn ngôn ngữ và đa ngôn ngữ trước đây, đạt được SOTA trên những bài toán xử lý ngôn ngữ tự nhiên tiếng Việt, bao gồm phân loại ngôn từ thù ghét. Kiến trúc của mô hình PhoBERT là kiến trúc đa lớp gồm nhiều lớp Bidirectional Transformer encoder. PhoBERT nhận đầu vào là một câu văn được cấu thành từ một chuỗi các từ có ngữ cảnh. Phần biểu diễn đầu vào của mô hình được xây dựng bằng cách tổng hợp các token cùng với segment vector và vị trí tương ứng của từ trong câu. Hình 6 thể hiện trực quan hơn những ý trên.

Lớp Fully-connected ở phần cuối của mô hình PhoBERT được thay thế bằng một kiến trúc mạng CNN. [21] Bởi vì hiện nay CNN đang là mô hình hiệu quả nhất cho bài toán phân loại đoạn văn bản ngắn [22], CNN được dùng thay cho những mạng neural sâu điển hình khác như LSTM, Bi-LSTM hay GRU.

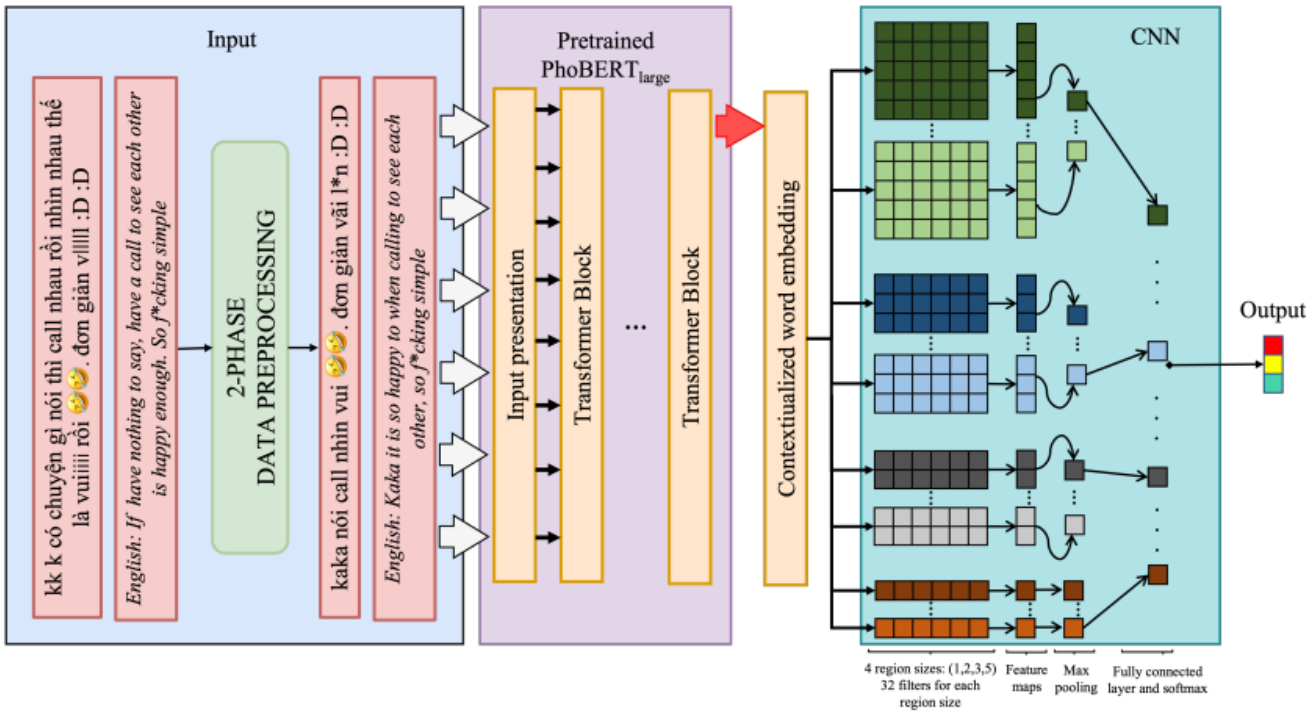
Trong bài nghiên cứu này, chúng tôi xây dựng mạng CNN với cấu trúc như sau:

- **INPUT:** Lớp input được sử dụng để khởi tạo đối tượng đầu vào từ ma trận các vector. Input và output của lớp này có kích thước bằng với số chiều của ma trận vector
- **CONV1D:** Chúng tôi xây dựng bốn lớp Convolution (tích chập) bằng cách sử dụng lớp CONV1D để trích xuất đặc trưng từ ma trận vector. Đối với mỗi lớp CONV1D, chúng tôi sử dụng một filter có kích thước cụ thể, thiết lập giá trị kích thước cho kernel và sử dụng ReLU làm hàm kích hoạt để tăng cường sự hội tụ.
- **POOLING:** Chúng tôi thực hiện Max pooling bằng cách sử dụng hàm Max-Pool1D. Đầu ra của lớp này là một ma trận các đặc trưng đã được giảm kích thước nhưng vẫn giữ lại các đặc trưng được trích xuất từ lớp CONV1D trước đó.
- **DROPOUT:** Trước khi xây dựng lớp Dropout, chúng tôi sử dụng hàm torch.cat() để nối các đặc ma trận đặc trưng mà đã được Pooled trước đó. Sau đó, chúng tôi thiết lập giá trị dropout để loại bỏ ngẫu nhiên các note trong mỗi lớp ẩn (hidden layer).
- **FC:** Trong lớp này, chúng tôi xây dựng một hàm mất mát (loss function) và một hàm tối ưu (optimized function) để kết nối dữ liệu đầu vào từ lớp Dropout tới các lớp phân loại. Chúng tôi sử dụng dụng thuật toán Adam optimization và hàm mất mát Crossentropy (Equation 1 để tối ưu hóa:

$$-\sum_{c=1}^{C} y_{o,c} \log(p_{o,c}) \tag{1}$$

Trong đó, **C** là số lượng các lớp (CLEAN, OFFENSIVE, HATE), **log** là hàm logarit tự nhiên, **y** chỉ số nhị phân (0 hoặc 1) nếu lớp **c** được phân loại chính xác đối với mẫu dữ liệu **o**, **p** là xác suất dự đoán **o** thuộc lớp c

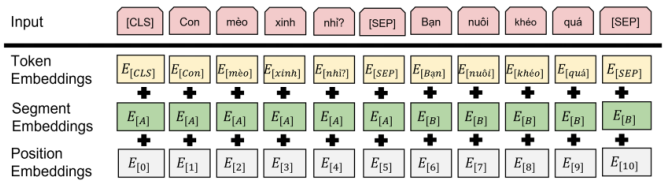
Các kỹ thuật Convolution và Pooling của mạng CNN giúp trích xuất các khái niệm chính và từ khóa của văn bản như các đặc trưng, từ đó cải thiện đáng kể hiệu suất của mô hình phân loại. Tuy nhiên, mạng CNN có một hạn chế tương đối lớn là không phù hợp để xử lý văn bản ở mức độ chuỗi[21], [22]. Để khắc phục hạn chế này, một mô hình ngôn ngữ được pretrained có quy mô lớn cho tiếng Việt như PhoBERT là một sự kết hợp hợp lý, vì PhoBERT có nhiệm vụ trích xuất đặc trưng từ câu văn để làm đầu vào đưa vào mô hình Text-CNN. Sau đó, word embedding có ngữ cảnh của các câu từ PhoBERT được đưa vào Text-CNN để lấy ra các bản đồ đặc trưng. Cuối cùng, nhãn dự đoán được xác định thông qua một lớp Softmax.



Hình 5: *Mô hình tổng quan của PhoBERT-CNN.*

Bảng V: Kết quả khi thực nghiệm trên bộ dữ liệu của chúng tôi, kết hợp với một số pre-trained word embedding.

Approach	Model	Word Embedding	F1-micro	F1-macro
ML	SVM	TF-IDF	89.31	71.48
	RF	TF-IDF	89.00	67.46
	XGBoost	TF-IDF	88.88	69.02
DL	Text CNN	fasttest	89.07	68.56
	Text CNN	phow2v (syllables)	89.25	69.69
	Text CNN	phow2v (words)	89.38	69.2
TL	PhoBERT	PhoBERT	89.50	73.81
	XLM-R	XLM-R	**90.00**	74,21
Combine	Text CNN	PhoBERT	**90.00**	**74.38**



Hình 6: *Quy trình biểu diễn đầu vào của mô hình PhoBERT.*

VI. Experiments

A. Setup Experiments

Chúng tôi sử dụng bộ dữ liệu đã qua 2 Phase tiền xử lý như đã nêu ở phần trước. Tuy nhiên có một số điểm khác biệt giữa các mô hình ở bước word embedding. Cụ thể, với hướng tiếp cận sử dụng machine learning, chúng tôi sử dụng kỹ thuật tf-idf vectorize cho bước này. Mặt khác, với mô hình Text CNN, chúng tôi sử dụng 2 pre-trained word embedding nổi tiếng cho tiếng việt là fasttext và phow2v. Còn đối với các mô hình transfer learning, chúng có khả năng tự vector hóa câu để biến thành đặc trưng.

- **fasttest:** một pre-trained đa ngôn ngữ vector từ, được giới thiệu bởi Grave et al. [23]
- **phow2v:** là một pre-trained w2v dành riêng cho tiếng việt, được giới thiệu bởi Nguyen et al. [24]

ML approach: Với các mô hình SVM, Random Forest, XGBoost chúng tôi đã sử dụng kỹ thuật tf-idf vectorize để chuẩn hóa các dữ liệu phù hợp với mô hình.

Với SVM, tham số gamma xác định mức độ ảnh hưởng của 1 điểm dữ liệu đến với các điểm dữ liệu khác, ở đây được đặt là 0.0001 mô hình được đào tạo với một đường ranh giới phân loại tương đối đơn giản, tham số C được đặt là 1000 để đánh giá tác động của các điểm dữ liệu bị phân loại sai đến quá trình học của mô hình và kernel được sử dụng là "rbf" phân loại dữ liệu chính xác hơn bằng cách chuyển đổi dữ liệu vào không gian cao hơn. Với Random Forest, chúng tôi đã cài đặt số lượng Decision Tree n-estimators=108 để xây dựng mô hình Random Forest, và độ sâu tối đa của Decision Tree max-depth = 40.

Với XGBoost chúng tôi cài đặt số lượng Decision Tree n-

Bảng VI: Kết quả thực nghiệm trên bộ dữ liệu của chúng tôi, cùng với quy trình tiền xử lý dữ liệu của Vi-HSD

Approach	Model	Word Embedding	F1-micro	F1-macro
ML	SVM	TF-IDF	88.38	68.75
	RF	TF-IDF	88.75	67.55
	XGBoost	TF-IDF	88.26	67.54
DL	Text CNN	Fasttext	88.94	68
	Text CNN	phow2v (syllables)	89.19	69.55
	Text CNN	phow2v (words)	88.7	67.99
TL	PhoBERT	PhoBERT	89.69	74
	XLM-R	XML-R	89.49	74.27
Combine	Text CNN	PhoBERT	88.63	72.74

estimators = 100 để ổn định độ chính xác và tốc độ của mô hình và random-state = 50 để kết quả đánh giá có thể được so sánh và phân tích một cách chính xác

DL approach: Với mô hình học sâu, chúng tôi lựa chọn Text-CNN cùng ba bộ pretrained Word Embedding: fasttest, phow2v(syllables) và phow2v(words) với Embedding size là 300. Chúng tôi thực nghiệm với 40 epochs, batch-size là 256, độ dài câu tương ứng là 100 và hệ số dropout 0.5. Mô hình của chúng tôi sử dụng Lớp Convolution 2D với 32 bộ lọc và kích thước 2, 3, 5 tương ứng. Trong quá trình huấn luyện, hàm số tối ưu Adam với hệ số học bằng 1e-4 được áp dụng và đưa ra kết quả trong Bảng V.

TL approach: Chúng tôi tiến hành cài đặt các thí nghiệm liên quan đến transfer learning sử dụng thư viện simpletransformer. Chúng tôi tách từ bằng VNCoreNLP, các tham số được sử dụng cho cả XLM-R base là hàm tối ưu Adam, hệ số học bằng 0.001, batch-size là 64 và được huấn luyện trên 15 epochs do giới hạn về tài nguyên. Với PhoBERT base là hàm tối ưu Adam, hệ số học bằng 1e-3, batch-size là 64 và được huấn luyện trên 20 epochs. Với PhoBERT + CNN, chúng tôi huấn luyện mô hình trên 20 epochs, với hàm tối ưu Adam, hệ số học là 1e-5, epsilon là 1e-8, batch-size là 64.

B. Experiments Results

Bảng V cho thấy kết quả khi thực nghiệm các mô hình trên bộ dữ liệu của chúng tôi. Kết quả được tính trên micro-F1 score và macro-F1 score. Bảng trên cũng cho thấy rằng mô hình XGBoost + TF-IDF đạt kết quả thấp nhất với 88.88 trên F1-micro và 69.02 trên F1-macro, mô hình RF có kết quả tương tự với F1-micro là 89.00 nhưng có F1-macro thấp hơn với 67.46, SVM là mô hình mang lại hiệu quả tốt nhất trong nhóm này với 89.31 F1-micro và 71.48 F1-macro. Với nhóm sử dụng hướng tiếp cận DL, hướng tiếp cận này cho kết quả tốt hơn so với hướng tiếp cận sử dụng ML khi kết quả của cả 3 mô hình đều tốt hơn nhóm sử dụng ML (trừ SVM). Tương tự với phương pháp tiếp cận TL khi 2 mô hình trong hướng này cho kết quả cao hơn so với 2 hướng kia. Đặc biệt, mô hình đa ngôn ngữ XLM-R đạt tới 90.00 trên F1-micro, đây cũng là con số tốt nhất trong bảng. Khi kết hợp Text CNN cùng với đặc trưng được trích xuất từ PhoBERT, kết quả là đáng kinh ngạc khi đạt 90.00 ở F1-micro và 74.38 ở F1-macro. Việc đứng đầu ở cả 2 độ đo đã giúp mô hình này trở thành mô hình tốt nhất trong phần thực nghiệm.

Với việc các mô hình được pretrained có kết quả tốt hơn so với các mô hình chưa pretrained, điều này cho thấy dữ liệu huấn luyện là một vấn đề quan trọng cần được chú ý trong bài toán này nói chung là lĩnh vực xử lý ngôn ngữ tự nhiên nói riêng.

Ngoài ra, chúng tôi còn thực hiện so sánh kết quả không sử dụng quy trình tiền xử lý dữ liệu của chúng tôi đề xuất, mà thay vào đó là quy trình tiền xử lý dữ liệu của [6]. Kết quả thu được ở Bảng VI. Kết quả này chỉ ra rằng quy trình tiền xử lý dữ liệu của chúng tôi có làm ảnh hưởng tới kết quả dự đoán ra được, cụ thể là tăng nhẹ từ 1-2 điểm trên cả F1-micro và F1-macro. Điều này cũng chứng minh rằng, với mỗi bộ dữ liệu khác nhau thì cần một quy trình xử lý riêng để tối ưu được kết quả.

C. Error Analysis

Kết quả từ Bảng V cho thấy rằng tuy rằng các mô hình pre-trained có kết quả tốt hơn, nhưng chưa vượt trội so với các mô hình khác, điều này có thể bị gây ra bởi việc mất cân bằng trong tập dữ liệu. Hơn 70% bình luận là nhãn CLEAN, dẫn đến vấn đề các mô hình có thể bị bias bởi nhãn này. Vấn đề này có thể giải quyết bằng cách sử dụng các kỹ thuật tăng cường data.

Hình 7: *Confusion matrix của mô hình XLM-R.*

Ngoài ra còn một số lỗi khác làm cho mô hình dự đoán sai 7. Ví dụ như khi dự đoán những từ tiếng lóng, cụ thể là bình luận "mao phắc", "phắc" ở đây ám chỉ từ "fuck" trong tiếng anh, thì ý nghĩa của câu này sẽ được gán nhãn OFFENSIVE, nhưng mô hình không thể hiểu được từ "phắc" nên đã gán nhãn CLEAN. Một ví dụ khác là trong bình luận có chứa "bá

đần", "bá đần" là từ lóng ám chỉ "bá điền" là một nhân vật có thật, khi đặt trong ngữ cảnh chuyên biệt thì có thể dễ dàng hiểu được bình luận này thuộc nhãn HATE, nhưng khi tách riêng nó ra thì bình luận này trông giống nhãn OFFENSIVE hơn. Thêm một lỗi sai hay mắc phải nữa là giữa các bình luận có nội dung gần tương tự nhau, ví dụ như "sặc" và "cặc", "sặc" là một từ ám chỉ cảm xúc bất ngờ theo hướng tiên cực nhưng không mang nội dung thù ghét hay xúc phạm gì, được gán nhãn CLEAN, còn "cặc" ám chỉ bộ phận sinh dục của nam giới, thường được dùng để chửi nhau, được gán nhãn "OFFENSIVE" nếu đứng riêng lẻ, nhưng hai từ này có cách viết tương tự nhau nên nhãn của 2 từ này bị mô hình hiểu sai và gán nhầm nhãn. Thêm vào đó, các kí hiệu đặc biệt vẫn chưa được chuẩn hóa như ":)))))))))", các kí hiệu này cần được chuẩn hóa về ":)"

Nhìn chung, các mẫu dự đoán sai và hầu hết các bình luận trong bộ dữ liệu đều được viết với các từ không chính thống, viết tắt, ngôn ngữ địa phương, không trong hoàn cảnh cụ thể và teencode. Do đó, trong quá trình tiền xử lý, chúng ta cần xử lý những đặc điểm này để cải thiện hiệu suất của các mô hình phân loại. Đối với việc viết tắt và ngôn ngữ địa phương, chúng ta có thể xây dựng một từ điển tiếng Việt về ngôn ngữ địa phương để thay thế các từ ngữ địa phương và một từ điển viết tắt tiếng Việt để thay thế các từ viết tắt trong các bình luận. Ngoài ra, đối với những từ không xuất hiện trong phương tiện truyền thông chính thống, chúng ta có thể cố gắng chuẩn hóa chúng thành các từ thông thường. Ngoài ra, biểu tượng emoji cũng là một đặc điểm cho biểu thị tính chất tích cực hoặc tiêu cực của một bình luận, điều này có thể hỗ trợ trong việc phát hiện nội dung thù ghét và xúc phạm trong các bình luận.

Không chỉ tồn tại lỗi sai của mô hình, mà đội ngữ gán nhãn cũng có một số lỗi sai khi gán nhãn vì thiếu hiểu biết về ngữ cảnh cũng như lĩnh vực chuyên biệt. Ví dụ như bình luận "bl líu lưỡi vãi lồn =))" thì từ "bl" mang ý nghĩa "bình luận viên", ám chỉ đến một cá nhân cụ thể, bình luận này nên được gán nhãn HATE, nhưng vì thiếu hiểu biết nên annotator đã gán nhãn OFFENSIVE.

VII. Kết luận

Chúng tôi đã xây dựng được một bộ dữ liệu về nhận diện ngôn từ thù ghét trong tiếng việt ở lĩnh vực gaming. Ngoài ra chúng tôi cũng đã thử nghiệm bộ dữ liệu này trên các mô hình thuộc các hướng tiếp cận khác nhau. Kết quả thu được cho thấy các mô hình hoạt động khá tốt trên bộ dữ liệu. Tuy nhiên vì kích thước bộ dữ liệu còn chưa đủ lớn và còn một số thiếu sót ở bước tiền xử lý cũng như là tăng cường dữ liệu nên vẫn có một số lỗi sai cố hữu. Trong tương lai nhóm sẽ cải thiện phương pháp thực nghiệm bằng cách cải thiện quy trình tiền xử lý cũng như tăng cường thêm data.

Tài liệu

[1] S. Mohan, A. Guha, M. Harris, F. Popowich, A. Schuster, and C. Priebe, "The impact of toxic language on the health of reddit communities," in *Advances in Artificial Intelligence - 30th Canadian Conference on Artificial Intelligence, Canadian AI 2017*, vol. 10233, 2017, pp. 51–56.

[2] S. Abu-Ghazaleh, Y. Hassona, and S. Hattar, "Dental trauma in social media-analysis of facebook content and public engagement," *Dental Traumatology: Official Publication of International Association for Dental Traumatology*, pp. 394–400, 2018.

[3] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, 2017, pp. 467–472.

[4] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, SocialNLP@EACL 2017*, 2017, pp. 1–10.

[5] X. Vu, T. Vu, M. Tran, T. Le-Cong, and H. T. M. Nguyen, "HSD shared task in VLSP campaign 2019: Hate speech detection for social good," *CoRR*, 2020.

[6] S. T. Luu, K. V. Nguyen, and N. L. Nguyen, "A large-scale dataset for hate speech detection on vietnamese social media texts," in *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices - 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021*, 2021, pp. 415–426.

[7] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 88–93.

[8] J. Chen, S. Yan, and K. Wong, "Verbal aggression detection on twitter comments: Convolutional neural network for short-text sentiment analysis," *Neural Comput. Appl.*, pp. 10 809–10 818, 2020.

[9] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, May 15-18, 2017*, 2017, pp. 512–515.

[10] R. Martins, M. Gomes, J. J. Almeida, P. Novais, and P. R. Henriques, "Hate speech classification in social media using emotional analysis," in *7th Brazilian Conference on Intelligent Systems, BRACIS 2018*, 2018, pp. 61–66.

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, 2019, pp. 4171–4186.

[12] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, 2019.

[13] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 2020, pp. 8440–8451.

[14] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at semeval-2020 task 12: BERT-CNN for offensive speech identification in social media," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING 2020*, 2020, pp. 2054–2059.

[15] D. Q. Nguyen and A. T. Nguyen, "Phobert: Pre-trained language models for vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, 2020, pp. 1037–1042.

[16] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 1960.

[17] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese natural language processing toolkit," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, Jun. 2018.

[18] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*, ACM, 1992, pp. 144–152.

[19] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[21] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 2014, pp. 1746–1751.

[22] C. He, S. Chen, S. Huang, J. Zhang, and X. Song, "Using convolutional neural network with BERT for intent determination," in *International Conference on Asian Language Processing, IALP 2019*, 2019, pp. 65–70.

[23] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018*, 2018.

[24] A. T. Nguyen, M. H. Dao, and D. Q. Nguyen, "A pilot study of text-to-sql semantic parsing for vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, ser. Findings of ACL, vol. EMNLP 2020, 2020, pp. 4079–4085.