TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, ĐHQG-HCM

KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐỀ TÀI: NHẬN DIỆN BÌNH LUẬN ĐỘC HẠI TRÊN NỀN TẢNG MẠNG XÃ HỘI

Môn học: CS221.Q13 – Xử Lý Ngôn Ngữ Tự Nhiên

Giảng viên hướng dẫn: TS. Nguyễn Trọng Chỉnh

Thực hiện bởi nhóm Tripl3_Cyber_Kn1ght, bao gồm:

1.	Nguyễn Gia Luân	23520896	Trưởng nhóm
2.	Lương Hoàng Long	23520879	Thành viên
3.	Lê Minh	23520928	Thành viên

Thời gian thực hiện: 01/10/2025 - 16/01/2026

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành nhất đến Thầy Nguyễn Trọng Chỉnh về sự tận tâm và sự hướng dẫn nhiệt tình trong môn học Xử lý ngôn ngữ tự nhiên. Nhờ sự hỗ trợ của Thầy, chúng em đã không chỉ nắm vững lý thuyết về các phương pháp xử lý văn bản và các mô hình ngôn ngữ, mà còn biết cách áp dụng nó vào quá trình thực hiện đồ án thực tế. Qua việc làm đồ án, chúng em cũng đã hiểu rõ hơn về cách thức máy tính phân tích và hiểu ngôn ngữ con người, cũng như là củng cố các kỹ năng mềm, gia tăng khả năng giao tiếp và làm việc trong nhóm.

Sự tận tụy và sự hỗ trợ của Thầy đã tạo nên một môi trường học tập tích cực giúp chúng em được đi sâu tìm hiểu, cập nhật những kiến thức và các thuật toán mới trong lĩnh vực AI. Chúng em cũng xin bày tỏ lòng biết ơn đối với sự động viên và những lời khuyên quý giá từ Thầy trong quá trình học tập và thực hiện đồ án. Thầy đã luôn sẵn lòng lắng nghe và giúp đỡ chúng em vượt qua những khó khăn.

Chúng em xin cam đoan sẽ tiếp tục áp dụng những kiến thức đã học từ Thầy vào quá trình học tập và làm việc sau này. Sự hướng dẫn tận tâm của Thầy sẽ luôn là nguồn động lực cho chúng em trong tương lai. Chúng em xin cảm ơn và kính chúc thầy có nhiều sức khỏe

Nhóm Tripl3_Cyber_Kn1ght

MỤC LỤC

DANH SÁCH HÌNH

DANH SÁCH BẢNG

Chương I. GIỚI THIỆU BÀI TOÁN

1. Đặt vấn đề

Nhận diện bình luận độc hại, hay còn gọi là Toxic Comment Detection hoặc Hate Speech Detection, là một bài toán quan trọng trong lĩnh vực khoa học máy tính và xử lý ngôn ngữ tự nhiên (NLP), tập trung vào việc tự động phát hiện và sàng lọc các nội dung mang tính công kích, thù ghét hoặc khiếm nhã trong văn bản. Mục tiêu chính của việc nhận diện này là phân loại các bình luận của người dùng trên mạng xã hội thành các nhãn cụ thể, điển hình là bình thường (non-toxic) và độc hại (toxic).

Trong thực tế, việc nhận diện bình luận độc hại đóng vai trò sống còn đối với sự phát triển bền vững của các nền tảng trực tuyến. Ví dụ, trên các mạng xã hội, việc ngăn chặn thành công các bình luận bắt nạt (cyberbullying) đồng nghĩa với việc bảo vệ được sức khỏe tinh thần của người dùng và giảm thiểu rủi ro pháp lý cho nhà quản lý, từ đó kiến tạo một môi trường giao tiếp văn minh, an toàn và khuyến khích sự tương tác tích cực từ cộng đồng.

2. Mục tiêu nghiên cứu

Trong phạm vi đề tài này, trọng tâm nghiên cứu của nhóm là ứng dụng mô hình ngôn ngữ tiên tiến BERT (với phiên bản tối ưu cho tiếng Việt là PhoBERT), đồng thời thực hiện đối sánh hiệu quả với thuật toán máy học kinh điển SVM (Support Vector Machine) để giải quyết bài toán nhận diện bình luận độc hại trên dữ liệu tiếng Việt. Quá trình này không chỉ dừng lại ở việc phân loại văn bản đơn thuần, mà còn hướng tới việc trích xuất và nhận biết các sắc thái tiêu cực ẩn trong ngôn ngữ mạng. Mục tiêu cốt lõi là xây dựng nền tảng cho một hệ thống lọc nội dung tự động, giúp phát hiện và ngăn chặn các ý kiến công kích, thù địch nhắm vào cá nhân hoặc tổ chức.

Tuy nhiên, việc xử lý ngôn ngữ tự nhiên trên các nền tảng mạng xã hội luôn đối mặt với những thách thức đáng kể. Rào cản lớn nhất chính là sự nhập nhằng về ngữ nghĩa (polysemy), khi mà cùng một từ ngữ nhưng đặt trong các ngữ cảnh khác nhau sẽ mang sắc thái hoàn toàn trái ngược. Việc dạy cho máy tính phân biệt được đâu là trêu đùa, đâu là xúc phạm thực sự là một bài toán khó.

Ví dụ: Từ “điên” hay “khùng” thường mang nghĩa tiêu cực chỉ trạng thái tâm lý bất ổn hoặc dùng để mắng người khác. Tuy nhiên, trong giao tiếp thân mật của giới trẻ, câu như “Mày điên vừa thôi, cười chết mất” lại mang hàm ý vui vẻ, tích cực. Nếu mô hình không nắm bắt được ngữ cảnh này, rất dễ dẫn đến việc dán nhãn sai (False Positive).

Các từ ngữ tiêu cực trong đồ án của nhóm được sử dụng với mục đích chú giải và phân tích, không nhằm mục đích công kích hay miệt thị đối tượng nào khác.

Chương II. BỘ NGỮ LIỆU

1. Bộ ngữ liệu:

Trong đề tài này, nhóm sử dụng bộ ngữ liệu tarudesu/VOZ-HSD (https://huggingface.co/datasets/tarudesu/VOZ-HSD) được công khai trên nền tảng Hugging Face. Bộ ngữ liệu được lưu dưới dạng csv. Vì bộ ngữ liệu gốc quá lớn để xử lý trên máy tính cá nhân (hơn 10 triệu dòng), bọn em trích ra 10 nghìn dòng để có thể hỗ trợ xử lý tốt hơn. Ví dụ về bộ ngữ liệu như sau:

	texts	labels	probs
1	Thanh niên thời nay toàn game gủng bay lắc ăn chơi, có văn hóa quái đâu mà làm nhà văn hóa thanh n…	1	0.9210277199745178
2	Cán bộ là nhàn mẹ nhất Lương thấp nhưng đúng nghĩa chỉ cần có mặt trên phòng là đủ	0	0.9697265625
3	không bị đắng hả bác, chỉ em với	0	0.9870235919952391
4	Sáng ra đã nhậu, nhất các thím này	0	0.9797027707099916
5	Có xem ko mà phát biểu liều. H2 Saka có tình huống xâm nhập đá hụt trong vòng cấm, MC ngoài tình h…	0	0.9689226746559144
6	Đời tao ghét mấy con chó như thế này. Làm cái đéo gì cũng không suy nghĩ thiệt hơn. Báo gì nữa tôi…	1	0.7090035080909729
7	Ừ , nhưng k mang khẩu trang là phạt	0	0.9925006031990052
8	tụi nó không chịu chạy các fen ạ. Đợi bóng đến chân; bóng không đến thì để tự mất. Đkm coi ức chế vl	1	0.8257996439933777
9	** mẹ lại xứ lờ bắc giang	1	0.9898887276649476
10	trù ếm juv dog die thua trắng	0	0.9303926229476928
11	ai tiện tay vào thả cho nó 1* giúp em ạ	0	0.9917540550231934
12	Lữ bố ổng từ lưu lạc rồi tự gây thế lực 1 phương chứ có dưới trướng ai đâu) Lữ bố toàn chơi với L…	1	0.7989509701728821
13	Cái thằng ngu này, mày phân biệt được dự đoán và kiểm tra ko. Giờ thấy nó còn nhưng mai nó hết mà …	1	0.8942347764968872
14	Tuấn aka xgame anti pep vs barca, fan "chân chính" MC . Tinh iu menh mông fan Liv ghét Mu thế đéo …	1	0.915715217590332
15	ít r t sẽ k thả mặt cười khi nc với nó nhắc nhở nó đi làm ăn chân chính đừng có mà đỏ đen. nhiều t…	0	0.9331860542297364
16	An toàn nhưng làm đéo gì có tuyệt đối như tấm ảnh kia ghi bác ơi	1	0.5618176460266113
17	Giờ bác đang ở nước ngoài à?	0	0.9939821362495422
18	bỏ đi mà làm người. thể loại mà thứ chơi bê đe thì toàn dắm lol thôi	1	0.8446394205093384
19	Người lớn dell bao giờ nc theo cái style như này, chắc chắn của 1 con dzẫm già ế mốc lolz, viết ra…	1	0.9857814908027648
20	Đúng là nó khác, nhưng như tôi đã nói ở post đầu (post mà anh cứ chụp mũ tôi) rằng sẽ k dùng các t…	1	0.6325678825378418
21	Sướng nhất rầu. Qua đó mà chồng đi.t hưởng thụ đi chờ đ' gì nữa	1	0.6734698414802551
22	Bác chuẩn đấy, em làm cty hàn cũng vậy	0	0.9959748387336732
23	đê tơ cho tên thuốc, hình như 12k 1 viên	0	0.9929914474487304
24	Lát lại có mấy thằng HR chó dại vào táp cho xem. . Còn lạ gì cái lũ này.	1	0.9172943234443665
25	Nay mình mới thấy cái bài báo này. Đọc giết thời gian trong lúc chờ khổ dâm đi anh chị em.	0	0.9900522828102112
26	ở q9 toàn rừng rú, vin cũng đéo cứu nổi, đi đêm sang đây sợ thú rừng vs người rừng nhảy ra bóp dái…	1	0.6046333312988281
27	22 - 25 tuổi đi làm ko dư được đồng nào, 25 - 28 dư được 5 tỉ	0	0.993349015712738
28	Hài vl mấy thằng MNSĐ cay cú chửi Martinez, cay lỏ mẹ dái	1	0.7472556829452515

Hình 1. Ví dụ minh họa về bộ ngữ liệu

2. Quy tắc chú thích dữ liệu

2.1. Định nghĩa các nhãn cảm xúc:

Bộ ngữ liệu bao gồm 2 label (0 và 1) đại diện cho 2 trạng thái của bình luận:

- Label 0: Bình luận toxic, thể hiện thái độ công kích cá nhân, sử dụng các từ ngữ tục tĩu, vô văn hóa, thường không có tính xây dựng.
- Label 1: Bình luận bình thường, thể hiện thái độ phản ứng tích cực, mang tính xây dựng, tránh dùng các từ ngữ nhạy cảm, tục tĩu.

2.2. Quy trình chú giải:

Các bình luận được gắn nhãn dựa trên nội dung. Định nghĩa các label thông qua thái độ, tính xây dựng của bình luận và có hay không việc sử dụng các từ ngữ tiêu cực, nhạy cảm, tục tĩu.

Một từ ngữ có thể là tục tĩu, nhạy cảm với bình luận này nhưng lại mang tính xây dựng với bình luận khác, nên khi phân loại cần chú ý ngữ cảnh và nội dung bình luận.

Ví dụ bằng từ "chó" được sử dụng trong hai câu sau với hai nhãn khác nhau:

- Bình luận bình thường: "Con chó của bạn trông đẹp quá"
- Bình luận tiêu cực: "Thằng chó tránh ra cho tao đi"

3. Thống kê ngữ liệu:

Để hiểu rõ hơn, nhóm thực hiện một số thống kê để kiểm tra các tính chất của bộ ngữ liệu (sau khi đã áp dụng underthesea để tách từ):

	Số câu		**Độ dài trung bình**	**Độ dài nhỏ nhất**	**Độ dài lớn nhất**
	0	1			
Dataset	5004	4096	26.94	2	578

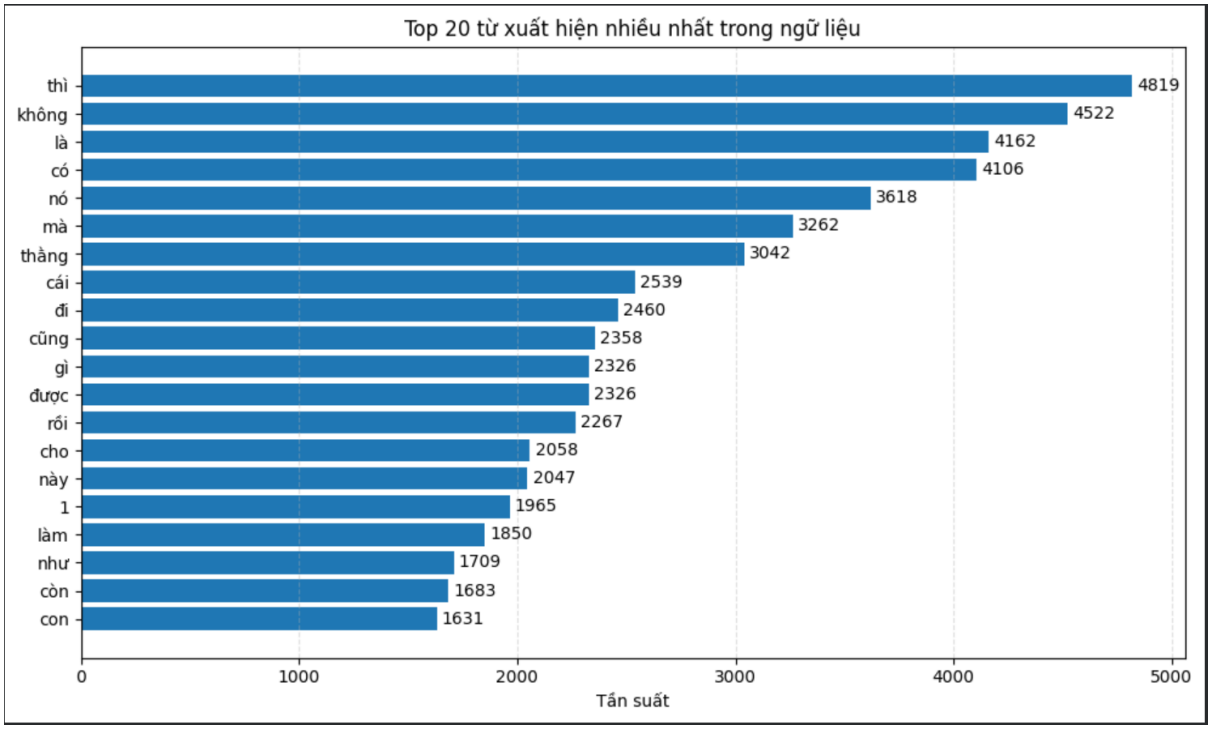
Bảng 1. Thống kê ngữ liệu

Bảng 1 trình bày các thông tin sau:

- Dataset: Bộ ngữ liệu nhóm sử dụng
- Số câu: Số mẫu (comment) thuộc mỗi label (0, 1) trong bộ ngữ liệu, trong đó 0 và 1 là hai label đã giải thích ở trên.
- Độ dài trung bình: Là độ dài trung bình tính theo số từ (không tính dấu câu và emoji) của các mẫu trong bộ ngữ liệu.
- Độ dài nhỏ nhất: Là độ dài nhỏ nhất của các mẫu trong bộ ngữ liệu, tính theo số từ.
- Độ dài lớn nhất: Là độ dài lớn nhất của các mẫu trong bộ ngữ liệu, tính theo số từ.

Quan sát bảng 1 ta thấy dữ liệu khá cân bằng khi tỷ lệ 2 label gần như là ngang nhau, vậy nên không cần thiết phải thực hiện các thuật toán cân bằng dữ liệu.

Để hiểu rõ về tần suất xuất hiện các từ trong ngữ liệu, nhóm thực hiện thống kê các từ (sau khi đã thực hiện các bước tách từ qua underthesea, và một số bước tiền xử lý như lọc teencode, …)



Hình 2. Tần suất 20 từ xuất hiện nhiều nhất trong ngữ liệu

4. Kiểm tra chú thích:

Để kiểm tra các mẫu có được chú giải đúng với quy trình đã trình bày không, nhóm thực hiện kiểm tra từ 60 mẫu được trích xuất ngẫu nhiên từ bộ ngữ liệu:

Nội dung	Label	Lý do
ra côn đảo đà lạt hay tây nguyên đi fen	0	Bình thường
dân đen là thành phần nộp thuế 95tr con cừu có bị mất một hai con hay cả trăm con cũng éo ai quan tâm	1	Ngôn từ kích động, phân biệt đối xử (chính trị)
tui chua bao gio an thit dog cay^` vi` dog la` ban than^ nhat^ của con nguoi` quan niem^ europe	**1 -> 0**	Sử dụng từ nhạy cảm nhưng chưa có í định xúc phạm chỉ nêu quan điểm

có nhưng trường hợp này chưa chắc và bản thân những ca ở tàu cũng dek xác định đc là dính từ trc hay trong thời gian cách ly hay ra ngoài mới dính mà bảo ủ bệnh rất dài nếu bảo phát hiện sau 23 ngày đến bm thì ok dựa vào đâu mà lão chung phát biểu sau 23 ngày ủ bệnh và bạn biết quy trình cách ly 14 ngày thế nào ko vào test 1 ra test 1 bao h những người hết thời gian cách ly 14 ngày ra dính mà ko xác định đc nguồn lây hẵng bảo ủ bệnh > 14 ngày nhé	0	Bình thường
thì bán đi hay đéo dám bán tao đọc vị mày luôn bán rồi đéo có xe màđi	1	Tục tĩu (đéo) và thái độ thách thức
do đi làm các bố đầu óc ko tập trung cứ mơ mộng nghĩ về ăn chơi cái khác thì sao có động lực tập trung vô công việc chính	1	Miệt thị, vơ đũa cả nắm (các bố)
ra phố đi bộ thấy em nào dễ thương thì bắt về nuôi thui anh hihi	0	Bình thường
vietsov thôi thím trc h vẫn thế mà	0	Bình thường
moá thằng lautaro nó dặt dẹo vl	**0 -> 1**	sử dụng từ tục tĩu nhưng khó phát hiện "móa"
hix uống phê xong về 2 thằng kia nó bem cho vỡ loz mất thôi	1	Bạo lực và tục tĩu (bem, vỡ loz)
vcl cái nhà trong bài có cả chỗ nấu nướng bếp núc á vậy trong 100 mạng chỉ cần 1 thằng ngu làm cháy phát thì toang cả lũ chứ chạy gì nổi	1	Xúc phạm cá nhân (thằng ngu)
trường cấp 1 là 8h giờ này cha mẹ nó cũng đi làm rồi còn tụi 7h là cấp 2 3 tụi này đa số gần trường tự đi học được làm éo gì mà 5h với	1	Tục tĩu (éo, cc)

5h30 trc học đại học thì 6h dậy ra bắt xe bus đi mất 45 mới đến trường cũng éo bao bao h dậy 5h 5h30 làm cc gì		
thằng obito lúc là tobi ngầu lòi vkl sau tác giả xây dựng theo hướng simp gái thấy đbrr vãi	**0 -> 1**	Chứa từ lóng tục tĩu viết tắt: "đbrr" (đầu b** rẻ rách). Model cần bắt được các từ viết tắt này.
hắn muốn kết thúc câu chuyện tình buồn yêu đơn phương bê đê nhưng càng nhớ lại hắn càng đau đớn quá hắn cắn răng rời bỏ y sau cùng quen vợ hắn và lấy nhau chóng vánh hắn yêu vợ chứ yêu lắm nhưng y sẽ mãi là cái dằm trong tim	0	Bình thường
có 100k mà kì kèo mắc mệt toàn giao lưu với mấy ông keo kiệt tính toán từng đồng thể loại này nếu là tui thì không thèm trả lời luôn 100k của mấy ổng to lớn quá	0	Bình thường
triệu hồi vin nô vào sủa cho anh vượn	1	Hate speech, hạ thấp nhân phẩm (vin nô, sủa)
nói thế thì éo cần họp gì nữa à	1	Tục tĩu (éo)
nên tử hình tội buôn bán người	0	Bình thường
cái thớt này đội tốn 2500đ ùa vào khệnh khạng ra mặt cười vãi chưởng	1	Mỉa mai, hạ thấp uy tín nhóm người (đội 2500đ)
t chứng kiến nhiều cảnh con cháu nhiếc móc đùn đẩy nghĩa vụ khi cha má bệnh già liệt giường rồi a à tham sống thêm chục năm mà nằm 1 chỗ cứt đái đầy đầu thì thà chọn cách mà đi tích cực chủ động khỏi vướng bận chúng nó a nói t xl t cũng chả có nghĩa vụ phải chứng minh cho a làm	1	Tục tĩu và ghê rợn (cứt đái, xl)

gì vì cũng chắc gì a đã sống dc tới lúc t đứt bóng		
bọn trẻ con toàn thế ko hiểu vì sao lúc chưa đi mẫu giáo ở nhà mấy năm chả bao giờ đau ốm gì 1 tiếng ho cũng ko có mà đi lớp là ốm suốt đủ thứ bệnh	**1 -> 0**	Đây là chia sẻ kinh nghiệm nuôi dạy con cái, than thở chuyện con ốm vặt khi đi học, hoàn toàn không có từ ngữ xúc phạm hay tục tĩu.
mô hình đặc thù ở đây là mô hình vi phạm pháp luật ý anh là thế với tôi thì ko có mô hình nào được gọi là đặc thù nếu anh làm ăn đúng luật ko phạm pháp thì đ bao giờ phải mất 1 đồng cho bảo kê hay đen đỏ và tất nhiên anh đừng vi phạm pháp luật thì sẽ được luật pháp bảo vệ còn tham tiền ôm gái tay vịn mại dâm trá hình thì tất nhiên là phải cần rồi có cái mẹ gì là đặc biệt đâu	1	Cáo buộc tiêu cực, giọng điệu gay gắt về tệ nạn
ờ thôi ha mua con xe ô tô có mẹ gì đâu mà vênh với mặt với đời trừ mấy thằng mua xe 4 5 tỉ đổ lên chứ con xe 1 tỉ hơn thì có đéo gì mà vênh mặt nếu có suy nghĩ như thế thì đây không tiếp nữa vì đã trái quan điểm thì ok fine suy nghĩ kiểu ra mẫu mới ng khác càng chê mình càng oai đúng là suy nghĩ của mấy thằng nít ranh kiểu ăn trộm đi tù về là oai ớ	1	Xúc phạm và tục tĩu (đéo, nít ranh, ăn trộm)
đây là điều k ai mong muốn song đừng nói trách nhiệm thuộc về ai mà trách nhiệm thuộc về toàn dân	0	Bình thường
ngày đầu gặp mà tặng cái đéo gì bố thằng simp lõ	1	Xúc phạm, lăng mạ người khác (đéo, simp lõ)
ngay cả khi bạn chọn 1 khu đất hoang j đi nữa thì đó vẫn là nơi công cộng bạn nuôi chó thì tốt nhất cho nó ỉa đái	**1 -> 0**	Do dính các từ nhạy cảm như "ỉa", "chửi", "rọ mõm"

ngay trong nhà bạn có dắt đi dạo thì rọ mõm lại đừng biện minh cho hành động của mình bạn ko cho chó đái ỉa trong nhà bạn là bạn sai hoàn toàn rồi nên cũng đừng chửi ng ta đánh bả chó nhà bạn khi ý thức bạn cũng ko có		
cuối tuần đi làm rảnh rỗi đọc mấy truyện voz cũ thấy hoàn cảnh cảm xúc sao mà quen thuộc không viết ra chuyện đời mình mà lưu lại sau quên đi thì thật tiếc xin phép được xưng là hắn trong câu chuyện của chính mình chuyện thật 100% dự định viết về những kỉ niệm đáng quên chuyện lừa tình gái cùng công ty yêu em đồng nghiệp đã có chồng chưa cưới	0	Bình thường
câm đi thầy văn ít thôi đóng mẹ nó vali vào tối xách đi cho kịp	1	Xúc phạm và thô lỗ (câm, mẹ nó)
bên nhật đi làm 1 giờ là đủ tiền ăn cả ngày rồi ít ăn nhậu đàn đúm thì có dư tiền gửi về thôi	0	Bình thường
éo tin nhật đang kêu gào thiếu lao động nhé	1	Tục tĩu nhẹ (éo)
phấn đào đá đấm tử tế để xem cho nó hấp dẫn nhé	0	Bình thường
khác méo gì ăn học mấy chục năm vẫn thất nghiệp có phải ai ăn học ra cũng có việc làm tử tế đâu so sánh làm gì	1	Tục tĩu nhẹ (méo)
làm tí thống kê vui xem anh em có giống mình ko	0	Bình thường
thế a mua xong rồi a kệ cmn luôn à k gắng làm để thêm mảnh mới à a nhà đất cho dù a ở 30 năm k lên giá thì nó	0	Bình thường

vẫn giá trị cũ còn chung cư a ở 30 năm rồi a bán đc như giá cũ k		
văn tấm gương thằng bạn ông chú anh họ nghe mắc ị vcl	1	Thô tục, phản cảm (mắc ị, vcl)
sử mình do ko đầu tư làm được như bọn tàu để truyền bá thôi chứ đâu có kém lấy 1 sự kiện lịch sử việt nam chiếm đất chămpa hay đập tụi khơ me ra bã mà lên phim thử coi biết bao nhiêu cái hay bao nhiêu mưu mô mưu tính để tiêu diệt lẫn nhau giữa 2 bên	**1 -> 0**	Do dính các từ "bọn", "đập", "tiêu diệt"
tiền vệ có matthaus tiền đạo có gerd muller nữa toàn diện vl	0	Bình thường
ở vn thì xác định 1 khi chạy xe máy tham gia giao thông phải tự mà lo cho thân mình thôi thím ey tụi xe tải chạy mất dạy bỏ mẹ ra đợt mình phượt đà lạt bị tụi xe khách vs xe rau chạy nc chiều lấn hết làn còn ép mình mém rớt khỏi lề đường luôn đây dkm 10 thằng hết 9 thằng súc sinh	1	Xúc phạm nặng nề và tục tĩu (dkm, súc sinh)
đẹp hay xấu thì xem inbox có con nào tự nhắn tin làm quen không là biết liền thôi chứ có cc gì đâu mà lên đây hỏ	**0 -> 1**	Có từ nhạy cảm , chửi thề "cc", giọng điệu toxic
thằng nào gọi tao t chăng cái điện thoại vào mồm ấy dm vin nô dùng như cái ***	1	Công kích và tục tĩu (dm, vin nô)
điều đó chứng tỏ em đã 1 mình chống đỡ cuộc sống này quá lâu so với cái tuổi của em tình trạng của em đang gặp phải giống như tôi của vài năm trước kéo dài tới bây giờ nhưng nhẹ hơn tuy rằng về mức độ thì của tôi ko bằng nhưng vì khác giới và sự chịu đựng của tôi có thể nói là cao nên mọi	0	Bình thường

chuyện vẫn không đi quá xa ít ra thì tôi chưa từng nghĩ đến chết vì tôi chết thì gia đình tôi khổ tôi mang tiếng ích kỉ khi chỉ chết cho mình		
m chửi levi nữa đi gánh xệ dái ra còn đòi hỏi óc chó pique cút giùm	1	Xúc phạm người khác, từ ngữ đả kích (gánh x* d**, óc ch*, …)
sợ bị lộ ra cái ngu dốt của mình à anh	1	Xúc phạm trí tuệ (ngu dốt)
ko đi ăn gửi phong bì 200k	0	Bình thường
sai thế lol nào đc ấy mà 90% sinh viên sai anh lấy số liệu đâu ra đấy	1	Tục tĩu (lol)
thằng quế công công văn vở còn hơn bổn chân nhân nữa cơ mà văn này là văn ngu zồi	1	Xúc phạm, đặt biệt danh xấu (công công, ngu)
chó lao tới là có nguy cơ tao sút hết suy diễn cc đ m thằng chó quyền ngu si mày giỏi thì vác xác con chó mày mà đi kiện	1	Hate speech và tục tĩu cực đoan (cc, đm, chó quyền, ngu si)
dm bắt được thằng gay này	1	Toxic (dm), công kích giới tính của người khác
bọn trẻ trâu manh động lắm may cho fen ko bị mất tai lúc ý	1	Dán nhãn tiêu cực (trẻ trâu)
chuyện này cũng k có gì là lạ kể cả gái xấu nó cũng vậy nhưng mà cái gì cũng có người này người kia đừng có quơ đũa cả nắm nha tml lũ đàn bà là sao hả	1	Phân biệt giới tính (lũ đàn bà) và tục tĩu (tml)
bà con mình sao không lấy tàu fe ra mà đi nhà nước có hỗ trợ mà	0	Bình thường
các bác phải thật bình tĩnh nên nhớ đầu mùa trước chúng ta còn phải ôm cặp đôi lính gác với quân ap ole hiện tại là đ có người thay rồi hắc bạch 1	0	Bình thường

trong 2 thằng nằm là thằng còn lại hiện nguyên hình		
ko thì cứ áp dụng công nghệ mới hơn wifi cho 1 tiếng mỗi bill muốn xài thêm thì mua thêm nước để coi ko có net thì ngồi được bao lâu	0	Bình thường
trong khi đó em vừa bị instant kill moé mấy cái hố ga sắc vcl	0	Bình thường
rút về bank thì phí 60k với tỷ giá chỉ tầm 22k	0	Bình thường
xin like mọi người ơi	0	Bình thường
thật mà cứ thử đi hồi nọ tiếc 1 em người anh vl nó ngon dã man con ngan đm nói chuyện đc 1 tí thì thằng ng yêu nó lại nó vừa đi bắn thuốc lào về đm thằng lol này cũng người anh ngon zai như diễn viên luôn thế là kiếm đc cạ bắn thuốc lào cái đm nó còn nghiện hơn tao giờ nó đi đéo đâu cũng thuốc lào dắt đít	1	Tục tĩu dày đặc (đm, thằng lol)
rồi vào cty tắm à bác	0	Bình thường
vl rô đĩ bay đánh đầu	1	Sử dụng các từ tục tĩu (vl, đ*)
mạ chrome khắp nơi chăm xe bao mệt luôn để nó xước với bẩn là mất chất	0	Bình thường

Bảng 2. Kiểm tra 60 mẫu trích xuất ngẫu nhiên

Qua 60 mẫu trên, nhóm đã thực hiện giải thích và chỉnh sửa những mẫu trích xuất từ bộ ngữ liệu, theo đó:

- Các mẫu có label sử dụng chữ đen, in thường là các mẫu có label đúng, nhóm thực hiện giải thích, chú giải mẫu đó.
- Các mẫu có label sử dụng chữ đỏ, in hoa là các mẫu có label sai, nhóm thực hiện chỉnh sửa, giải thích, chú giải vì sao phải thay đổi label của mẫu đó.

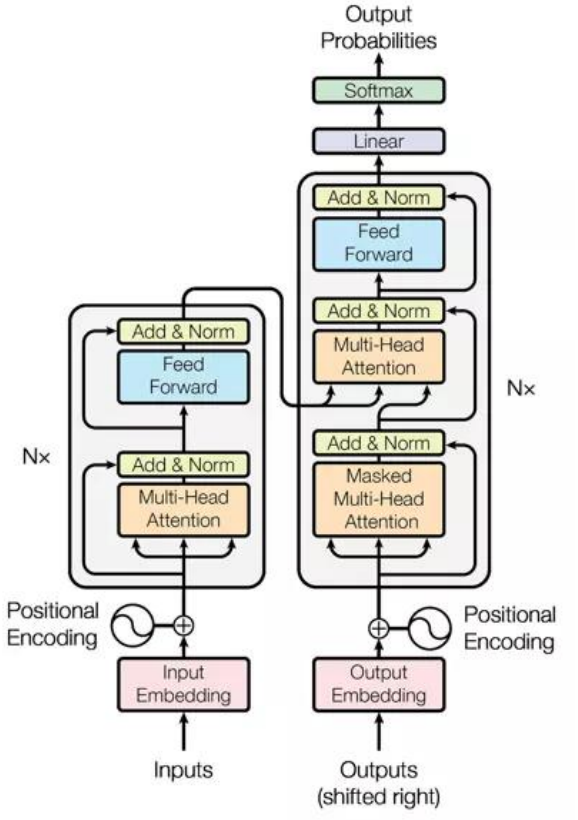
Quan sát bảng, ta thấy tỷ lệ gắn label sai là 7 trên tổng số 60 mẫu, chiếm tỷ lệ 11,67%. Như vậy, ta có thể thấy tuy vẫn còn một số mẫu bị gắn label sai, nhưng đa phần các mẫu được gắn nhãn đúng với quy trình chú thích đã trình bày ở phần trước.

Chương III. PHƯƠNG PHÁP SỬ DỤNG

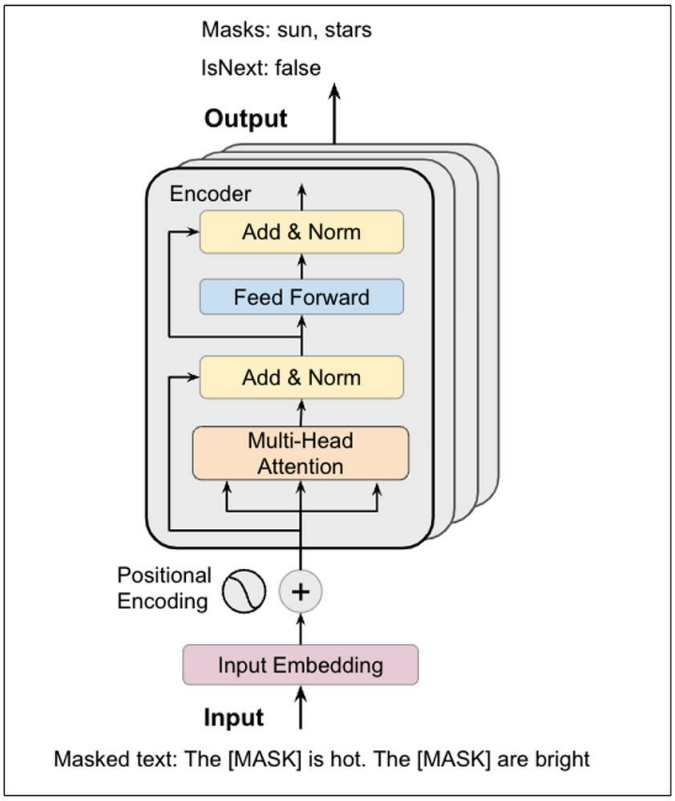
1. Kiến trúc Transformer Encoder - Từ BERT đến RoBERTa và PhoBERT:

BERT (Bidirectional Encoder Representations from Transformers), được Google AI giới thiệu vào năm 2018, là một cột mốc quan trọng trong lĩnh vực Xử lý ngôn ngữ tự nhiên (NLP). Về mặt kiến trúc, BERT không sử dụng toàn bộ mô hình Transformer (vốn bao gồm cả Encoder và Decoder cho các tác vụ dịch máy), mà chỉ khai thác chồng các lớp Transformer Encoder.

Cơ chế hoạt động cốt lõi của BERT dựa trên Self-Attention (Tự chú ý) đa đầu, cho phép mô hình xem xét toàn bộ câu văn cùng một lúc thay vì xử lý tuần tự (trái sang phải hoặc phải sang trái) như các mô hình RNN hay LSTM trước đây. Tính chất này được gọi là Bidirectional (Hai chiều), giúp BERT thấu hiểu sâu sắc ngữ cảnh của từng từ dựa trên mối quan hệ với tất cả các từ khác trong câu. Nhờ đó, BERT tạo ra các biểu diễn ngữ nghĩa phong phú, mang lại hiệu suất vượt trội trên hàng loạt tác vụ như Trả lời câu hỏi (QA), Phân loại văn bản, và Nhận diện thực thể (NER).



Hình 3. Kiến trúc Transformer



Hình 4. Kiến trúc BERT

Sự thành công của BERT đã mở đường cho các biến thể tối ưu hơn như:

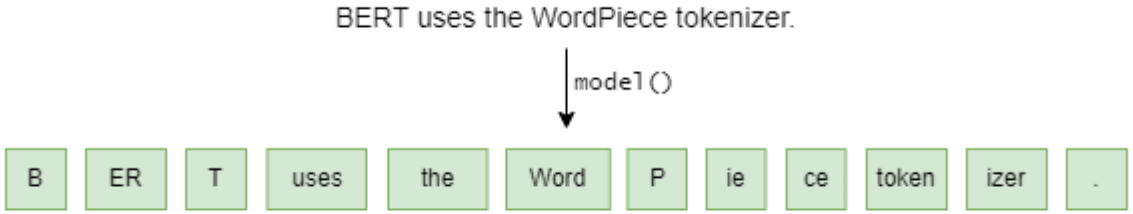
- RoBERTa (Facebook AI): Kế thừa kiến trúc của BERT nhưng tối ưu hóa quy trình huấn luyện (training recipe). RoBERTa sử dụng lượng dữ liệu khổng lồ (~160GB), loại bỏ nhiệm vụ dự đoán câu tiếp theo (NSP) và áp dụng kỹ thuật Dynamic Masking (thay đổi mặt nạ che từ ngẫu nhiên qua mỗi epoch) để tăng khả năng tổng quát hóa.
- PhoBERT (VinAI): Là mô hình ngôn ngữ đơn ngữ (monolingual) dành riêng cho tiếng Việt, được huấn luyện theo phương pháp của RoBERTa. PhoBERT xử lý đặc thù của tiếng Việt (ngôn ngữ đơn lập, nhiều từ ghép) bằng cách yêu cầu tiền xử lý tách từ (Word Segmentation) trước khi đưa vào mô hình. PhoBERT cũng có hai phiên bản tiêu chuẩn là Base (12 blocks) và Large (24 blocks).

1.1. Input Representation (Biểu diễn đầu vào):

Trước khi đi vào các lớp tính toán, dữ liệu văn bản thô cần trải qua quá trình số hóa nghiêm ngặt. Trong BERT và các biến thể, quá trình này bao gồm hai giai đoạn: Tokenization và Embedding.

a. Tokenization (phân tách từ):

Khác với việc tách từ theo dấu cách thông thường, BERT sử dụng kỹ thuật WordPiece Tokenization. Kỹ thuật này chia nhỏ các từ hiếm hoặc từ chưa biết (Out-Of-Vocabulary) thành các đơn vị nhỏ hơn (sub-words). Ví dụ như từ "playing" có thể được tách thành "play" và "##ing".



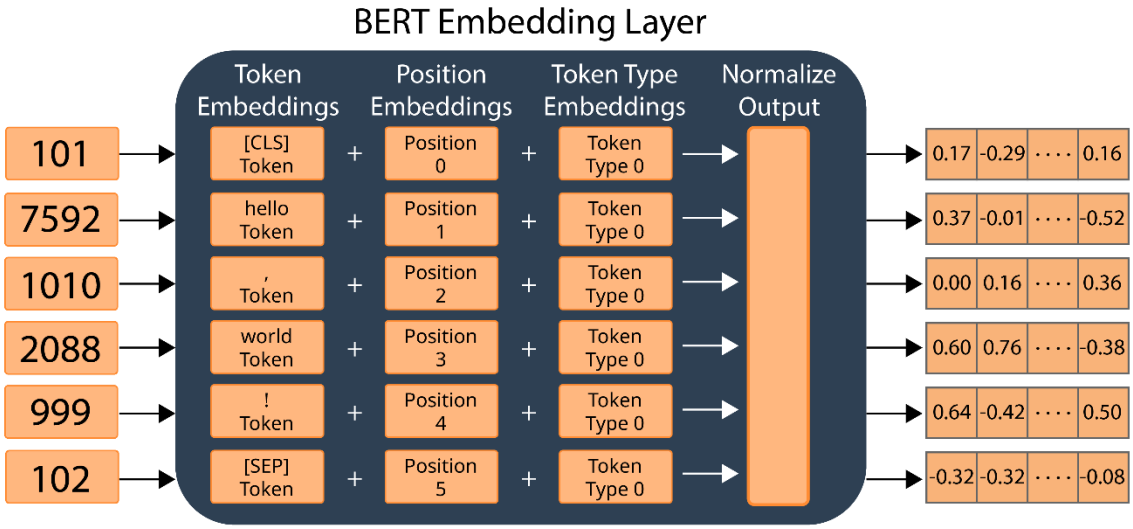
Hình 5. BERT Tokenization

b. Input Embedding:

Sau khi được phân tách thành các token, bước đầu tiên trong kiến trúc mạng nơ-ron là chuyển đổi các token rời rạc này thành các vector liên tục. Mục đích là để ánh xạ dữ liệu vào một không gian vector đa chiều (thường là 768 chiều với bản Base hoặc 1024 chiều với bản Large) nơi các từ có nghĩa tương đồng sẽ nằm gần nhau.

Điểm đặc biệt trong kiến trúc BERT là vector biểu diễn đầu vào cuối cùng không chỉ đơn thuần là vector của từ đó, mà là tổng (element-wise sum) của ba loại embedding thành phần:

- Token Embeddings: Biểu diễn ngữ nghĩa của chính token đó (WordPiece hoặc BPE token). Các giá trị này được khởi tạo ngẫu nhiên và tinh chỉnh liên tục trong quá trình huấn luyện (pre-training).
- Segment Embeddings: Giúp mô hình phân biệt được các câu khác nhau trong cùng một đầu vào (quan trọng cho các tác vụ đầu vào dạng cặp câu như Question Answering).
- Position Embeddings: Cung cấp thông tin về vị trí tuyệt đối của token trong chuỗi. Do cơ chế Self-Attention tính toán song song và không có tính tuần tự, nếu thiếu thành phần này, mô hình sẽ coi câu "Long đánh Luân" giống hệt "Luân đánh Long". Khác với Transformer gốc (dùng hàm Sin/Cos cố định), BERT sử dụng các learnable position embeddings (các vector vị trí được học như một tham số trong quá trình huấn luyện), với độ dài chuỗi tối đa thường là 512 token.



Hình 6. BERT Input Embedding

Các embedding trên sau khi cộng lại sẽ tạo thành một vector tổng hợp chứa đầy đủ thông tin về: nội dung từ, thuộc về câu nào, và nằm ở đâu trong câu, sẵn sàng để đưa vào lớp Encoder đầu tiên.'

1.2. Cơ chế Attention:

Attention là một cơ chế giúp mô hình học cách "chú ý" đến các phần khác nhau của đầu vào khi thực hiện các tác vụ khác nhau. Trong ngữ cảnh của xử lý ngôn ngữ tự nhiên, attention cho phép mô hình tập trung vào các từ liên quan nhất khi mã hóa một câu hoặc đoạn văn bản.

c. Self-Attention (Tự chú ý):

Self-attention là trái tim của kiến trúc Transformer và BERT, cho phép mô hình vượt qua giới hạn của việc xử lý tuần tự (như RNN/LSTM). Cơ chế này giúp mỗi từ trong câu có khả năng "quan sát" và tương tác với tất cả các từ khác trong cùng một câu để xác định ngữ cảnh, bất kể khoảng cách vị trí giữa chúng.

Quá trình tính toán Self-Attention trong BERT có thể được mô hình hóa qua các bước sau:

Bước 1 - Tạo các vector Query, Key, và Value: Mỗi token đầu vào (đã được embedding thành vector X) sẽ được chiếu (project) qua ba ma trận trọng số khả học (learnable weight matrices) W^Q, W^K, W^V để tạo ra ba vector đại diện mới:

- Query (Q): Vector "truy vấn", đại diện cho từ hiện tại đang đi tìm kiếm thông tin.
- Key (K): Vector "chỉ mục", đại diện cho các từ khác trong câu đang được đối chiếu.

- Value (V): Vector "nội dung", chứa thông tin thực sự của từ.

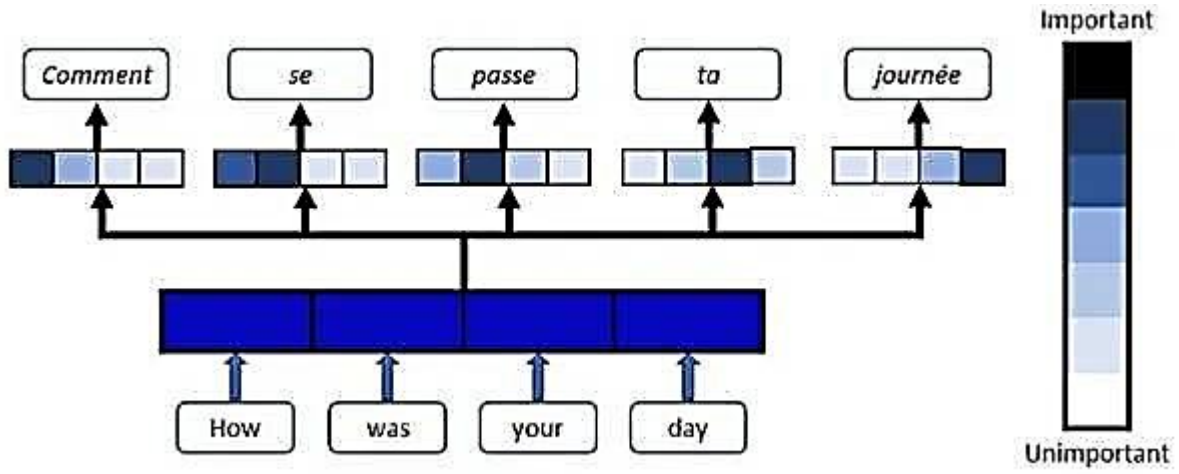
Giả sử X là ma trận đầu vào, ta có:$Q = XW^Q$; $K = XW^K$; $V = XW^V$.

Bước 2 - Tính toán Scaled Dot-Product Attention: Mức độ quan trọng (attention score) giữa các từ được tính bằng tích vô hướng giữa Query và Key.

- Tính điểm (Score): Nhân ma trận Q với chuyển vị của K. Kết quả càng lớn thể hiện sự tương đồng hoặc liên quan cao giữa hai từ.
- Chuẩn hóa (Scale): Chia kết quả cho căn bậc hai chiều dài vector key. Bước này cực kỳ quan trọng để ngăn chặn hiện tượng bùng nổ giá trị (vanishing gradients) khi đi qua hàm Softmax.
- Softmax: Áp dụng hàm Softmax để chuyển đổi điểm số thành xác suất (tổng bằng 1).
- Tổng hợp (Weighted Sum): Nhân trọng số xác suất vừa tìm được với vector Value (V).

Công thức tổng quát cho cơ chế này là:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Hình 7. Cơ chế Self Attention

Kết quả đầu ra là một vector tổng hợp, chứa thông tin của từ hiện tại được "làm giàu" thêm bởi ngữ cảnh của các từ liên quan nhất trong câu.

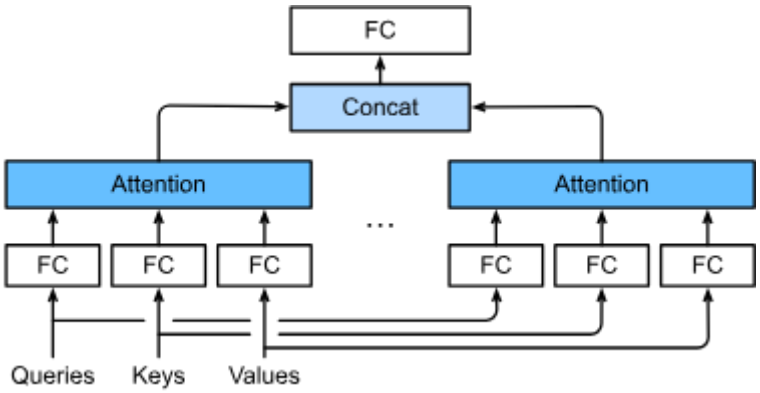
d. Multi-Head Attention (Đa đầu chú ý):

Thay vì chỉ sử dụng một bộ trọng số chú ý đơn lẻ, BERT sử dụng cơ chế Multi-Head Attention. Ý tưởng là chia nhỏ vector embedding thành h phần (heads) và chạy cơ chế self-attention song song trên mỗi phần đó.

Mục đích là để Cho phép mô hình tập trung vào các không gian biểu diễn khác nhau tại các vị trí khác nhau. Ví dụ: một đầu (head) có thể tập trung vào quan hệ ngữ pháp (chủ ngữ - động từ), trong khi đầu khác tập trung vào quan hệ ngữ nghĩa (từ đồng nghĩa).

Đầu ra của các head (head_1, head_2, ..., head_h) được nối lại (concatenate) và đưa qua một lớp tuyến tính cuối cùng (W^o) để khôi phục kích thước ban đầu.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$



Hình 8. Multi-Head Attention

e. Position-wise Feed-Forward Networks & Residual Connections:

Một khối Encoder (Encoder Block) trong BERT không chỉ dừng lại ở Multi-Head Attention. Để mạng nơ-ron có khả năng học sâu và hội tụ tốt, cấu trúc mỗi block bao gồm hai thành phần thiết yếu khác (phần này bổ sung chi tiết kỹ thuật quan trọng):

- Add & Norm (Kết nối tắt và Chuẩn hóa lớp): Xung quanh mỗi lớp con (Multi-Head Attention và Feed-Forward), BERT áp dụng cơ chế kết nối tắt (Residual Connection) theo sau là chuẩn hóa lớp (Layer Normalization).

  $$Output = LayerNorm(x + Sublayer(x))$$

  Cơ chế này giúp luồng gradient truyền đi mượt mà hơn trong mạng sâu 12-24 lớp, giải quyết vấn đề biến mất đạo hàm (vanishing gradient).
- Position-wise Feed-Forward Networks (FFN): Sau lớp Attention, dữ liệu đi qua một mạng nơ-ron truyền thẳng (Feed-Forward) được áp dụng riêng biệt cho từng vị trí token. Cấu trúc gồm hai lớp tuyến tính với một hàm kích hoạt phi tuyến ở giữa. Trong Transformer gốc, hàm kích hoạt là ReLU. BERT và RoBERTa sử dụng hàm kích hoạt GELU (Gaussian Error Linear Unit). GELU mượt hơn ReLU, cho phép xác suất nhỏ đi qua đối với các giá trị âm, giúp mô hình ngôn ngữ hoạt động hiệu quả hơn.

$$FFN(x) = GELU(xW_1 + b_1)W_2 + b_2$$

2. Quy trình Pre-training của BERT:

Khác với các phương pháp học giám sát truyền thống yêu cầu dữ liệu được gán nhãn thủ công, BERT được huấn luyện theo cơ chế tự giám sát (self-supervised learning) trên một lượng lớn văn bản không gán nhãn. Để đạt được sự hiểu biết sâu sắc về ngôn ngữ, BERT được đào tạo đồng thời trên hai tác vụ chính: Masked Language Model (MLM) và Next Sentence Prediction (NSP).

2.1. Task 1: Masked Language Model (MLM)

Masked Language Model (MLM), hay còn gọi là nhiệm vụ điền từ vào chỗ trống (Cloze task), là sự đổi mới quan trọng nhất của BERT.

Các mô hình ngôn ngữ thông thường (như GPT đời đầu) thường quét văn bản từ trái sang phải. Nếu áp dụng cơ chế hai chiều (bidirectional) một cách ngây thơ, từ hiện tại sẽ "nhìn thấy" chính nó thông qua chuỗi phía sau, dẫn đến việc mô hình sao chép kết quả thay vì thực sự học ngữ cảnh.

Để huấn luyện biểu diễn hai chiều sâu (deep bidirectional representation) mà không bị lộ thông tin (data leakage), BERT che đi một tỷ lệ phần trăm các token đầu vào một cách ngẫu nhiên và yêu cầu mô hình dự đoán các token bị che đó dựa trên ngữ cảnh xung quanh.

Cơ chế hoạt động: Trong quá trình huấn luyện, bộ tạo dữ liệu sẽ chọn ngẫu nhiên 15% số lượng token trong chuỗi đầu vào để thực hiện che giấu. Các vector ẩn (hidden vectors) tại lớp cuối cùng tương ứng với các token bị che sẽ được đưa qua một lớp Softmax để dự đoán xác suất trên toàn bộ từ điển. Hàm mất mát (Loss function) chỉ tính toán dựa trên việc dự đoán các token bị che này, bỏ qua các token không bị che.

Chiến lược Masking (Quy tắc 80-10-10): Việc luôn thay thế từ bằng token [MASK] sẽ gây ra sự sai lệch (mismatch) giữa giai đoạn Pre-training và Fine-tuning, vì trong thực tế (khi Fine-tuning), token [MASK] không bao giờ xuất hiện. Để giảm thiểu vấn đề này, trong 15% số từ được chọn để dự đoán, BERT áp dụng chiến lược thay thế hỗn hợp như sau:

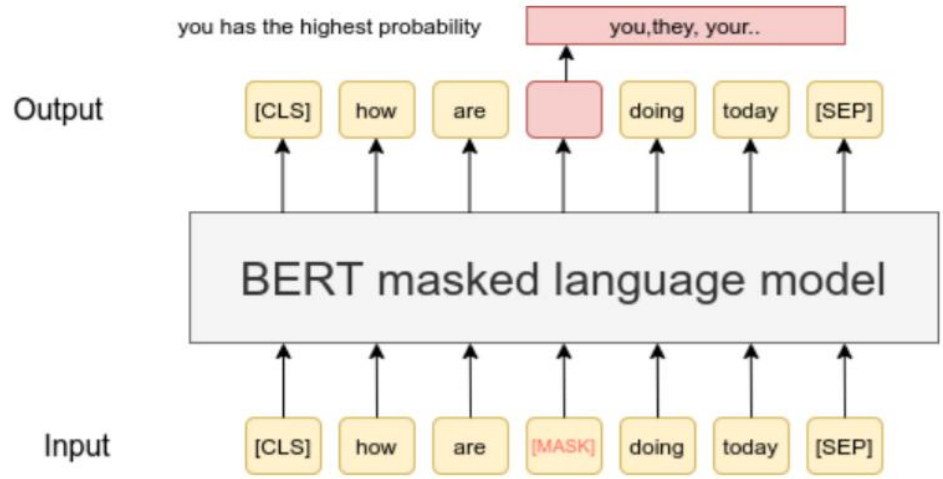
- 80% trường hợp: Thay thế bằng token [MASK]. Ví dụ: "Messi là cầu thủ xuất sắc nhất thế giới." sẽ thay bằng "Messi là [MASK] xuất sắc nhất thế giới". Mục đích là bắt buộc mô hình dựa vào ngữ cảnh để đoán từ.
- 10% trường hợp: Thay thế bằng một từ ngẫu nhiên bất kỳ. Ví dụ: “Messi là cầu thủ xuất sắc nhất thế giới.” sẽ thay bằng “Messi là [ba lô] xuất sắc nhất thế giới”. Mục đích là buộc mô hình phải kiểm tra ngữ cảnh xem từ hiện tại

có hợp lý hay không, giúp mô hình học được các đặc trưng ngữ nghĩa sâu hơn thay vì chỉ học cách đoán [MASK].

- 10% trường hợp: Giữ nguyên từ gốc. Ví dụ: “Messi là cầu thủ xuất sắc nhất thế giới.” sẽ thay bằng “Messi là [cầu thủ] xuất sắc nhất thế giới”. Mục đích là giúp mô hình thiên về việc giữ lại biểu diễn đúng của từ thực tế thay vì luôn cố gắng sửa đổi nó, đồng thời thu hẹp khoảng cách với giai đoạn Fine-tuning.

Mở rộng nâng cao: Đây là điểm khác biệt kỹ thuật quan trọng giữa BERT và các thế hệ sau (như RoBERTa):

- Static Masking (BERT): Quá trình tạo mặt nạ (masking) được thực hiện một lần duy nhất trong giai đoạn tiền xử lý dữ liệu. Điều này có nghĩa là trong suốt quá trình huấn luyện (dù chạy bao nhiêu epochs), một câu cụ thể sẽ luôn bị che ở cùng một vị trí cố định.
- Dynamic Masking (RoBERTa): Khắc phục hạn chế trên bằng cách sinh ra mặt nạ ngẫu nhiên ngay tại thời điểm đưa dữ liệu vào mô hình (on-the-fly). Mỗi lần mô hình gặp lại câu " Messi là cầu thủ xuất sắc nhất thế giới.", nó có thể bị che ở một vị trí khác (lần 1 che "Messi", lần 2 che "thế giới"). Điều này giúp tăng cường tính đa dạng của dữ liệu và cải thiện khả năng tổng quát hóa của mô hình.



Hình 9. Task 1 - Masked Language Model

2.2. Task 2: Next Sentence Prediction (NSP)

Trong khi Masked Language Model (MLM) giúp BERT hiểu sâu sắc ngữ cảnh nội bộ của một câu (intra-sentence context), thì nhiều tác vụ NLP quan trọng khác như Hỏi đáp (Question Answering - QA) hay Suy luận ngôn ngữ tự nhiên (Natural Language Inference - NLI) lại đòi hỏi mô hình phải nắm bắt được mối quan hệ logic giữa hai câu văn bản (inter-sentence relationship).

Để giải quyết vấn đề này, BERT được huấn luyện thêm một tác vụ nhị phân đơn giản: Cho trước một cặp câu (A, B), mô hình phải xác định xem câu B có phải là câu tiếp theo thực sự của câu A trong văn bản gốc hay không.

Các Token đặc biệt và Cấu trúc đầu vào: Để thực hiện NSP, kiến trúc đầu vào của BERT sử dụng các token đặc biệt với vai trò cụ thể:

- Token [CLS] (Classification Token): Luôn được chèn vào vị trí đầu tiên của mọi chuỗi đầu vào. Mặc dù [CLS] không mang ý nghĩa từ vựng cụ thể, nhưng sau khi đi qua các lớp Encoder, vector đầu ra tương ứng tại vị trí này được coi là đại diện tổng hợp cho toàn bộ cặp câu. Trong tác vụ NSP, vector C này sẽ được đưa qua một lớp mạng nơ-ron (Classification Layer) rồi dùng hàm Softmax để dự đoán xác suất nhị phân: IsNext hoặc NotNext.
- Token [SEP] (Separator Token): Là ký tự phân cách dùng để đánh dấu điểm kết thúc của một câu. Trong một cặp input (A, B), token này xuất hiện ở cuối câu A và cuối câu B để ngăn cách rạch ròi hai thành phần.

Ngoài ra, để mô hình phân biệt rõ hơn, ta áp dụng Segment Embeddings.

Quy trình tạo dữ liệu: Dữ liệu huấn luyện NSP được sinh ra tự động từ corpus văn bản với tỷ lệ cân bằng 50/50:

- 50% Positive (IsNext): Câu B thực sự là câu tiếp theo của câu A trong văn bản. Nhãn mục tiêu là IsNext.
- 50% Negative (NotNext): Câu B là một câu được lấy ngẫu nhiên từ một văn bản khác. Nhãn mục tiêu là NotNext.

Mặc dù Google AI thiết kế NSP với mục đích giúp BERT hiểu quan hệ câu, nhưng các nghiên cứu sau này đã chỉ ra một thực tế thú vị:

- Tác vụ quá đơn giản: Việc phân biệt một câu ngẫu nhiên (NotNext) thường quá dễ dàng đối với mô hình, vì sự khác biệt về chủ đề (topic shift) là rất rõ ràng. Do đó, mô hình không thực sự học được khả năng suy luận logic sâu sắc.
- Hiệu quả thực tế: RoBERTa đã chứng minh rằng việc loại bỏ NSP và thay vào đó huấn luyện với các chuỗi văn bản liền mạch dài hơn (FULL-SENTENCES) giúp mô hình học tốt hơn và đạt hiệu suất cao hơn trên các bảng xếp hạng. Đây là lý do các mô hình hiện đại sau BERT thường lược bỏ tác vụ này.

3. BERT Fine-tuning (Tinh chỉnh mô hình):

Fine-tuning là bước chuyển giao tri thức (Transfer Learning), nơi mô hình BERT đã được huấn luyện trước (Pre-trained) trên tập dữ liệu khổng lồ được điều chỉnh

để giải quyết một bài toán cụ thể. Thay vì khởi tạo trọng số ngẫu nhiên, ta khởi tạo mô hình bằng các tham số đã học được từ quá trình Pre-training.

Ưu điểm cốt lõi của phương pháp này là mô hình hội tụ rất nhanh và đạt hiệu suất cao (State-of-the-art) ngay cả khi tập dữ liệu của nhiệm vụ cụ thể khá nhỏ. Trong quá trình Fine-tuning, toàn bộ các tham số của mô hình (bao gồm cả các lớp Encoder bên dưới) đều được cập nhật đạo hàm (end-to-end update), chứ không chỉ riêng lớp đầu ra.

3.1. Task 1: Single Sentence Classification (Phân loại văn bản)

Đây là tác vụ phổ biến nhất, ứng dụng trong việc phân loại thư rác (Spam detection), phân tích cảm xúc, hoặc phát hiện mã độc trong URL (đối với lĩnh vực An toàn thông tin).

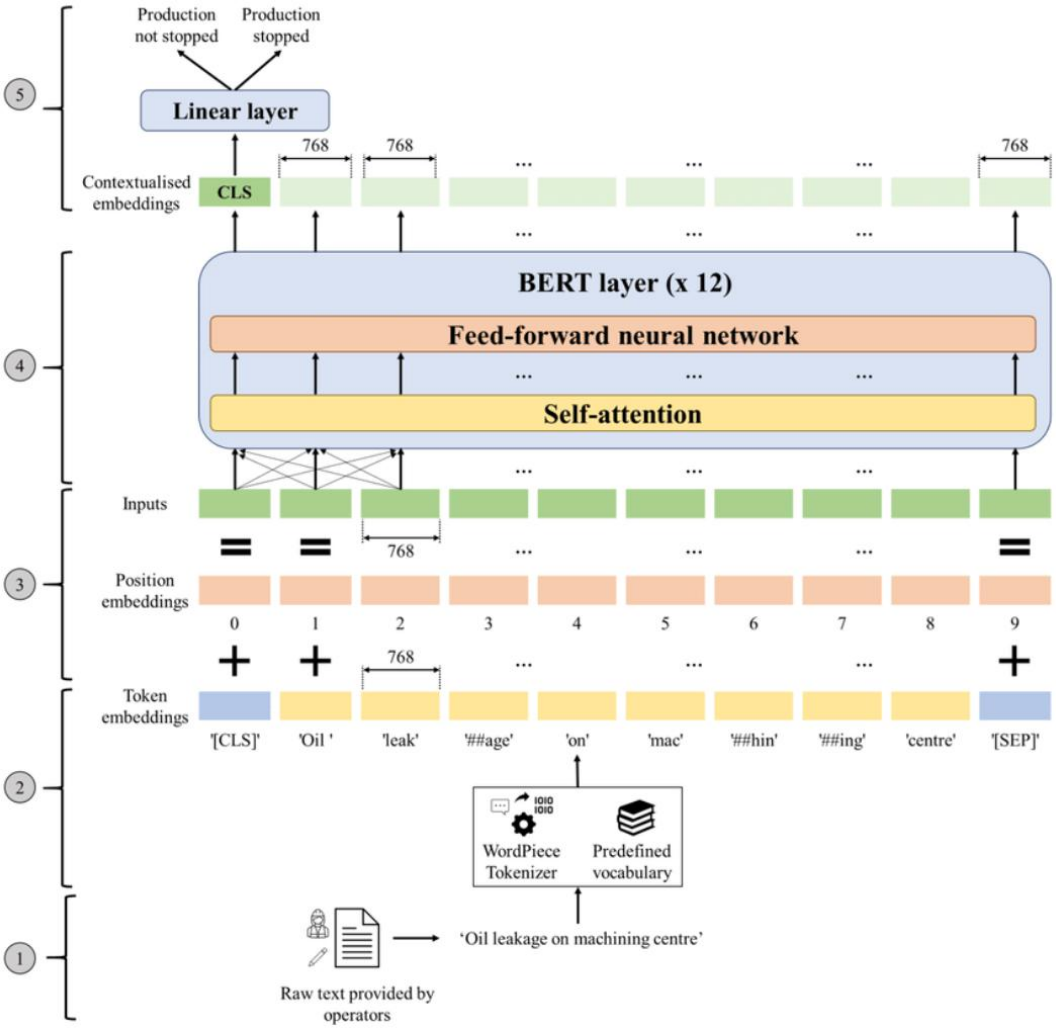
Để phân loại một đoạn văn bản, BERT tận dụng token đặc biệt [CLS]:

- Input: Thêm [CLS] vào đầu câu: [CLS] Nội dung văn bản [SEP].
- Encoding: Chuỗi đi qua 12 (hoặc 24) lớp Encoder của BERT.
- Representation: Ta lấy vector đầu ra tại vị trí đầu tiên (tương ứng với token [CLS]). Ký hiệu là C. Vector này được coi là đặc trưng tổng quát nhất của toàn bộ câu.

Vector C được đưa qua một lớp kết nối đầy đủ (Fully-connected layer) đơn giản, theo sau là hàm Softmax để tính xác suất cho từng nhãn. Giả sử bài toán có K nhãn, ta có ma trận trọng số phân loại W. Xác suất của nhãn P được tính như sau:

$$P = \text{softmax}(CW^T)$$

Mô hình được huấn luyện để tối thiểu hóa hàm mất mát Cross-Entropy Loss giữa nhãn dự đoán và nhãn thực tế. Trước khi vào lớp Linear, vector C thường đi qua một lớp Dropout để giảm thiểu hiện tượng Overfitting khi tập dữ liệu fine-tune quá nhỏ.



Hình 10. Mô phỏng quá trình Fine-tune BERT cho một tác vụ phân loại văn bản

3.2. Task 2: Question Answering (Hỏi đáp):

Trong các bài toán hỏi đáp trích xuất (Extractive QA), điển hình như trên bộ dữ liệu SQuAD (Stanford Question Answering Dataset), mục tiêu của mô hình là tìm ra một đoạn văn bản (span) trong đoạn ngữ cảnh (context) trả lời chính xác cho câu hỏi đưa ra.

Để giải quyết bài toán này, quá trình Fine-tuning BERT cần giải quyết hai thách thức kỹ thuật chính:

- **Biểu diễn ngữ cảnh**: Làm thế nào để mô hình phân biệt rõ ràng đâu là "Câu hỏi" và đâu là "Đoạn văn tham chiếu" trong cùng một chuỗi đầu vào.
- **Dự đoán vị trí**: Làm thế nào để mô hình chỉ ra chính xác vị trí bắt đầu (Start Span) và kết thúc (End Span) của câu trả lời.

Khác với tác vụ phân loại văn bản chỉ dùng một câu, tác vụ QA yêu cầu đầu vào là một cặp chuỗi (Question, Paragraph). BERT xử lý vấn đề này bằng cách đóng gói chúng thành một chuỗi duy nhất với các token đặc biệt:

- Token [CLS]: Đặt ở đầu chuỗi. Trong SQuAD 2.0, token này còn có vai trò đặc biệt là đại diện cho câu trả lời "không có đáp án" (nếu câu hỏi không thể trả lời dựa trên đoạn văn).
- Token [SEP]: Đặt ở giữa để ngăn cách Câu hỏi và Đoạn văn, và một token ở cuối cùng.
- Segment Embeddings: Đây là yếu tố quyết định. Toàn bộ token thuộc Câu hỏi được gán Segment A (nhúng A), và toàn bộ token thuộc Đoạn văn được gán Segment B (nhúng B).

Thay vì sử dụng một bộ giải mã (Decoder) phức tạp để sinh ra câu trả lời, BERT sử dụng cơ chế dự đoán vị trí dựa trên các vector đầu ra của Encoder. Chúng ta đưa thêm hai vector trọng số mới vào mô hình để huấn luyện:

- Vector trọng số Start (S)
- Vector trọng số End (E)

Quá trình tính toán diễn ra như sau:

- Mã hóa (Encoding): Toàn bộ chuỗi đầu vào đi qua các lớp Encoder của BERT. Giả sử T_i là vector đầu ra tại lớp cuối cùng tương ứng với token thứ i trong đoạn văn ngữ cảnh.
- Dự đoán vị trí Bắt đầu (Start Span): Mô hình tính tích vô hướng giữa vector trọng số S và vector T_i của từng từ, sau đó áp dụng hàm Softmax để tính xác suất từ thứ i là từ bắt đầu của câu trả lời:

$$P_{start}(i) = \text{softmax}(S \cdot T_i) = \frac{e^{S \cdot T_i}}{\sum_k e^{S \cdot T_k}}$$

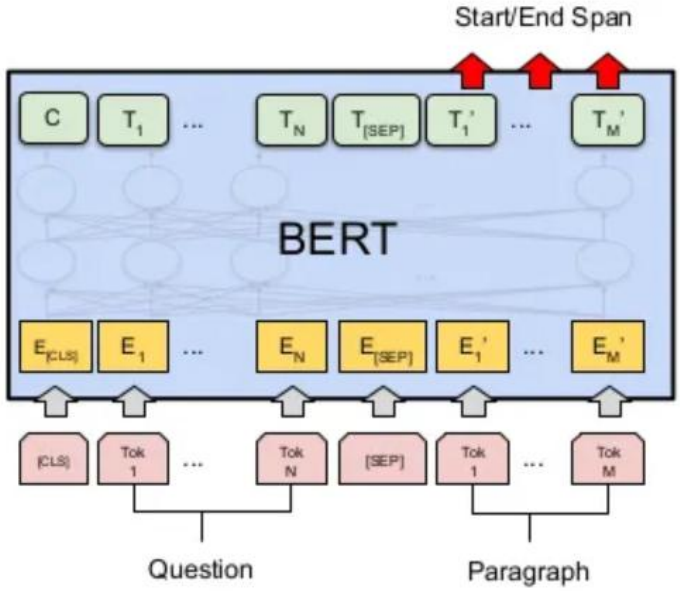
- Dự đoán vị trí Kết thúc (End Span): Tương tự, xác suất từ thứ i là từ kết thúc của câu trả lời được tính bằng:

$$P_{end}(i) = \text{softmax}(E \cdot T_i) = \frac{e^{E \cdot T_i}}{\sum_k e^{E \cdot T_k}}$$

Trong quá trình suy luận (Inference), câu trả lời được chọn là đoạn văn bản nằm từ chỉ số i đến j sao cho thỏa mãn hai điều kiện:

- i <= j (Vị trí kết thúc phải nằm sau hoặc trùng với vị trí bắt đầu).
- Tổng điểm số xác suất là lớn nhất.

Trong trường hợp mô hình được huấn luyện trên SQuAD 2.0 (có các câu hỏi không thể trả lời), nếu điểm số của vị trí [CLS] lớn hơn điểm số của span tốt nhất tìm được, mô hình sẽ trả về kết quả là "Không có câu trả lời" (No Answer).



Hình 11. Mô phỏng quá trình Fine-tune BERT cho một tác vụ hỏi đáp

4. RoBERTa và PhoBERT:

Sau sự ra đời của BERT, cộng đồng nghiên cứu đã nỗ lực cải tiến mô hình này để đạt hiệu suất cao hơn. Hai trong số những biến thể quan trọng và liên quan trực tiếp đến đề tài này là RoBERTa (phiên bản tối ưu hóa của BERT) và PhoBERT (phiên bản dành riêng cho tiếng Việt).

4.1. RoBERTa (A Robustly Optimized BERT Pretraining Approach):

RoBERTa, được phát triển bởi Facebook AI (2019), về cơ bản vẫn giữ nguyên kiến trúc Transformer Encoder như BERT. Tuy nhiên, các tác giả lập luận rằng BERT gốc đã bị "huấn luyện chưa đủ mức" (undertrained). Do đó, RoBERTa tập trung vào việc tối ưu hóa quy trình huấn luyện (training recipe) để khai thác tối đa tiềm năng của kiến trúc này.

Những cải tiến cốt lõi của RoBERTa so với BERT bao gồm:

- Dữ liệu huấn luyện khổng lồ: RoBERTa được huấn luyện trên 160GB văn bản (lớn hơn gấp 10 lần so với 16GB của BERT), bao gồm BookCorpus, Wikipedia tiếng Anh, CC-News, OpenWebText và Stories. Điều này giúp mô hình học được các mẫu ngôn ngữ đa dạng và phức tạp hơn.

- Dynamic Masking (Masking động): Sinh ra mặt nạ ngẫu nhiên mỗi khi chuỗi dữ liệu được đưa vào mô hình. Dữ liệu huấn luyện được nhân bản và mask 10 lần với các chiến lược khác nhau. Điều này giúp mô hình không "học vẹt" vị trí bị che mà thực sự hiểu ngữ cảnh biến thiên.
- Loại bỏ Next Sentence Prediction (NSP): Các thực nghiệm của RoBERTa chỉ ra rằng nhiệm vụ dự đoán câu tiếp theo không đóng góp nhiều vào hiệu suất, thậm chí gây nhiễu cho các tác vụ hạ nguồn. RoBERTa loại bỏ NSP và thay vào đó huấn luyện với các chuỗi văn bản liền mạch dài hơn (lên đến 512 token), giúp mô hình nắm bắt sự phụ thuộc xa tốt hơn.
- Batch size lớn: Thay vì batch size 256 như BERT, RoBERTa sử dụng batch size lên đến 8.000 chuỗi. Việc này giúp gradient ổn định hơn và dễ dàng song song hóa trên các hệ thống phân tán quy mô lớn.

4.2. PhoBERT (Mô hình ngôn ngữ tiền huấn luyện cho tiếng Việt):

PhoBERT là mô hình ngôn ngữ đơn ngữ (monolingual) đầu tiên đạt chuẩn State-of-the-art cho tiếng Việt, được phát triển bởi VinAI Research. PhoBERT kế thừa hoàn toàn kiến trúc và phương pháp huấn luyện tối ưu của RoBERTa, nhưng được điều chỉnh để phù hợp với đặc thù ngôn ngữ tiếng Việt.

Dữ liệu huấn luyện:

- Phiên bản đầu: Huấn luyện trên 20GB dữ liệu (1GB Wikipedia + 19GB Báo chí).
- PhoBERT Base v2: Được bổ sung thêm 120GB dữ liệu văn bản từ bộ dữ liệu OSCAR-2301, nâng cao đáng kể khả năng bao quát ngôn ngữ.

Xử lý đầu vào (Input Processing): Đây là điểm khác biệt quan trọng nhất. Tiếng Việt là ngôn ngữ đơn lập nhưng đơn vị ngữ nghĩa thường là từ ghép (2-3 âm tiết). PhoBERT bắt buộc sử dụng RDRSegmenter (từ bộ công cụ VnCoreNLP) để tách từ và nối các âm tiết bằng dấu gạch dưới (ví dụ: nhân_viên, điều_hoà) trước khi đưa vào mã hóa BPE.

4.3. Minh họa quy trình Fine-tuning PhoBERT cho bài toán phân loại:

Để làm rõ cách dữ liệu đi qua mô hình, nhóm lấy ví dụ cụ thể với một câu phản hồi của người dùng về một ứng dụng ngân hàng (Mobile Banking). Giả sử câu đầu vào là: “App ngân hàng này lỗi liên tục, bảo mật kém khiến tôi rất lo lắng khi giao dịch.”. Mô hình sẽ xử lý tuần tự qua các bước sau:

- Bước 1 - Tiền xử lý và Tách từ (Tokenization): Khác với tiếng Anh tách từ bằng khoảng trắng, PhoBERT yêu cầu các từ ghép tiếng Việt phải được nối lại để đảm bảo tính toàn vẹn về ngữ nghĩa. Câu văn trên sau khi đi qua bộ

tách từ RDRSegmenter (VnCoreNLP) sẽ trở thành: “App ngân_hàng này lỗi liên_tục , bảo_mật kém khiến tôi rất lo_lắng khi giao_dịch .”. Sau đó, chuỗi này được chuyển đổi thành các định dạng tensor mà mô hình hiểu được.

- Bước 2 - Trích xuất đặc trưng (Feature Extraction): Dữ liệu được đưa vào kiến trúc PhoBERT. Tại đây, thông tin đi qua 12 lớp Encoder với cơ chế Self-Attention. Đầu ra quan trọng nhất mà nhóm sử dụng là vector pooler_output. Đây là vector đại diện cho token đặc biệt [CLS] (token đầu tiên của chuỗi). Trong kiến trúc BERT/PhoBERT, sau khi đi qua toàn bộ các lớp mạng, vector [CLS] này đã học được ngữ cảnh tổng quát của toàn bộ câu văn, chứa đựng thông tin về ngữ nghĩa ("lỗi", "bảo mật kém") và thái độ của người nói ("lo lắng").
- Bước 3 - Phân loại (Classification Head): Vector [CLS] (768 chiều) được đưa vào một mạng nơ-ron kết nối đầy đủ (Fully Connected Layer) mà nhóm đã ghép thêm vào sau PhoBERT. Lớp này thực hiện phép nhân ma trận để nén vector từ kích thước 768 xuống kích thước 3 (tương ứng với 3 nhãn: Tiêu cực, Trung tính, Tích cực). Kết quả đầu ra là một tensor logits, ví dụ: [4.5, -1.2, -3.0].
- Bước 4 - Dự đoán kết quả: Mô hình áp dụng hàm Softmax (hoặc so sánh trực tiếp) để tìm ra giá trị lớn nhất trong tensor kết quả. Trong ví dụ này, giá trị lớn nhất nằm ở chỉ số (index) 0 (tương ứng với nhãn Negative/Tiêu cực). Kết luận: Mô hình dự đoán câu phản hồi trên mang sắc thái Tiêu cực.

Chương IV. TRIỂN KHAI

Để giải quyết bài toán nhận diện bình luận độc hại, nhóm tiến hành cài đặt và thử nghiệm trên hai hướng tiếp cận: phương pháp học máy truyền thống với SVM (Support Vector Machine) đóng vai trò là mô hình cơ sở (baseline) để so sánh, và phương pháp học sâu hiện đại sử dụng kỹ thuật Fine-tuning mô hình ngôn ngữ tiền huấn luyện PhoBERT.

1. Mô hình cơ sở: Support Vector Machine (SVM):

1.1. Tiền xử lý dữ liệu:

Tiền xử lý dữ liệu là bước nền tảng quyết định "thành bại" của các mô hình học máy truyền thống như SVM. Khác với con người có thể hiểu ngữ cảnh linh hoạt, máy tính chỉ làm việc trên các con số và quy tắc cứng nhắc. Do đó, việc

chuẩn hóa dữ liệu đầu vào giúp mô hình tập trung vào các đặc trưng quan trọng nhất thay vì bị nhiễu bởi các yếu tố vô nghĩa.

Thông qua quá trình Khám phá dữ liệu (EDA), nhóm nhận thấy bộ dữ liệu bình luận độc hại thu thập từ mạng xã hội có đặc điểm rất khác so với văn bản báo chí hay Wikipedia. Dữ liệu này chứa nhiều "rác" (như ký tự đặc biệt, emoji vô nghĩa), sử dụng ngôn ngữ tự do, sai chính tả và đặc biệt là hiện tượng sử dụng teencode/viết tắt tràn lan nhằm mục đích lách luật kiểm duyệt hoặc để gõ nhanh. Nếu không xử lý kỹ, khi vector hóa (TF-IDF), không gian đặc trưng sẽ bị phình to với hàng nghìn từ vô nghĩa, làm giảm hiệu suất của thuật toán SVM.

a. Làm sạch dữ liệu và xử lý teencode, viết tắt:

Trong môi trường mạng xã hội, "Teencode" hay viết tắt là một "đặc sản". Đặc biệt đối với các bình luận mang tính đả kích, tiêu cực (toxic), người dùng thường có xu hướng viết tắt các từ chửi thề hoặc cố tình viết sai chính tả để tránh bị các bộ lọc từ khóa tự động phát hiện. Việc này tạo ra một thách thức lớn cho mô hình SVM: Máy tính sẽ hiểu "vcl", "vl", "vãi l**" là 3 từ hoàn toàn khác nhau, dù về mặt ngữ nghĩa chúng đều biểu thị cùng một mức độ cảm xúc tiêu cực. Điều này làm phân tán trọng số của từ khóa, khiến mô hình khó học được mẫu (pattern) của sự độc hại.

Ví dụ thực tế trong bộ dữ liệu:

Câu gốc: "m nch ngu vl ra, t ko thèm nghe !!!"

Trong câu trên xuất hiện các nhiễu:

- Viết tắt: "m" (mày), "nch" (nói chuyện), "t" (tao), "ko" (không).
- Teencode/Slang: "vl" (chửi thề).
- Ký tự đặc biệt: "!!!" (không mang ý nghĩa phân loại trong ngữ cảnh này).

Để giải quyết vấn đề này, nhóm thực hiện quy trình 2 bước:

- Làm sạch cơ bản: Đưa toàn bộ văn bản về chữ thường (lowercase) để đồng nhất. Loại bỏ các ký tự đặc biệt (dấu câu dư thừa, URL, thẻ HTML) bằng biểu thức chính quy (Regex).
- Chuẩn hóa Teencode: Nhóm xây dựng một bộ từ điển teencode.txt chứa các cặp từ khóa (key-value), trong đó key là từ viết tắt thường gặp trong tập dữ liệu và value là từ tiếng Việt chuẩn tương ứng. Từ điển này đóng vai trò như một bộ "từ điển sống" giúp máy tính hiểu được ngôn ngữ mạng.

```
Số mục teencode: 408
[('ctrai', 'con trai'),
 ('khôg', 'không'),
 ('bme', 'bố mẹ'),
 ('cta', 'chúng ta'),
 ('mih', 'mình'),
 ('mqh', 'mối quan hệ'),
 ('cgai', 'con gái'),
 ('nhữg', 'những'),
 ('mng', 'mọi người'),
 ('svtn', 'sinh viên tình nguyện')]
```

Hình 12. Một số ví dụ về từ điển teencode nhóm tự xây dựng

Kết quả sau xử lý sau làm sạch & map teencode: "mày nói chuyện ngu v** l** ra tao không thèm nghe"

Việc mapping này giúp gom nhóm các biến thể từ vựng về một dạng chuẩn duy nhất, giúp thuật toán TF-IDF tính toán tần suất từ chính xác hơn, từ đó nâng cao độ chính xác khi phân loại.

b. Phân đoạn, tách từ (Word Segmentation) với Underthesea:

Tiếng Việt là ngôn ngữ đơn lập, nhưng đơn vị có nghĩa nhỏ nhất để cấu thành câu thường là từ ghép (2-3 âm tiết) chứ không phải từ đơn. Ví dụ: từ "mất dạy" mang nghĩa thô tục. Nếu tách theo khoảng trắng thông thường (như tiếng Anh), ta sẽ có 2 từ: "mất" (động từ: không còn) và "dạy" (động từ: giáo dục). Khi đứng riêng lẻ, hai từ này hoàn toàn không mang sắc thái độc hại, nhưng khi đi cùng nhau, chúng là một đặc trưng quan trọng của lớp Toxic.

Nếu không tách từ đúng, mô hình SVM sẽ bị mất đi ngữ cảnh quan trọng, dẫn đến việc dự đoán sai lệch (Underfitting).

Để giải quyết vấn đề này, nhóm sử dụng thư viện Underthesea - một bộ công cụ xử lý ngôn ngữ tự nhiên tiếng Việt mạnh mẽ (tương tự VnCoreNLP nhưng nhẹ và dễ tích hợp hơn trong Python). Underthesea sử dụng các mô hình thống kê (CRF) để dự đoán ranh giới giữa các từ dựa trên ngữ cảnh câu.

Hình 13. Underthesea

Cụ thể, các từ ghép sau khi được nhận diện sẽ được nối với nhau bằng dấu gạch dưới _. Điều này biến cụm từ ghép thành một token duy nhất (một đơn vị từ vựng liền mạch) trong quá trình vector hóa.

Ví dụ đầu vào "thanh niên thời nay toàn game gủng bay lắc" sau khi xử lý tách từ sẽ thành "thanh_niên thời nay toàn game gủng bay_lắc"

Ở ví dụ trên, "thanh_niên" và "bay_lắc" được gom lại. Đặc biệt từ "bay_lắc" là một từ lóng chỉ tệ nạn xã hội, là dấu hiệu nhận biết mạnh mẽ của các bình luận tiêu cực/tệ nạn. Việc tách từ chính xác giúp mô hình nắm bắt được "tín hiệu" này.

c. Loại bỏ Stopword:

Stopword là những từ hay gặp nhưng ít giá trị phân loại thông tin. Do đó, trong xử lý ngôn ngữ tự nhiên (NLP) thường loại bỏ stopword trước khi thực hiện các tác vụ như TF-IDF, phân loại,… Nhóm đã xây dựng một danh sách stopword thường dùng trong ngữ cảnh tiếng Việt.

```
stopwords = {
    "có", "rất", "tôi", "ở", "của", "là", "với", "cho", "được",
    "thì", "đã", "trong", "sẽ", "này", "đến",
    "và", "nhưng", "hay", "cũng", "lại", "đang", "đi", "gì",
    "nữa", "nên", "như", "khi", "này", "kia"
}
```

Hình 14. Danh sách stopword mà nhóm đã xây dựng

Mục đích của loại bỏ stopword là loại bỏ các từ không có giá trị phân tích trong văn bản, giúp giảm độ phức tạp của mô hình, tăng tốc độ xử lý và cải thiện hiệu suất của mô hình, giúp mô hình tập trung vào các từ mang ý nghĩa quan trọng.

d. Vector hóa dữ liệu bằng TF-IDF:

Sau khi văn bản đã được làm sạch và tách từ, chúng ta có một tập hợp các từ chuẩn. Tuy nhiên, SVM là một thuật toán toán học, nó không thể tính toán trực tiếp trên chuỗi ký tự. Chúng ta cần chuyển đổi văn bản sang dạng vector số học.

Thay vì sử dụng phương pháp đếm từ đơn giản (CountVectorizer) - vốn chỉ quan tâm từ đó xuất hiện bao nhiêu lần mà không quan tâm độ quan trọng, nhóm quyết định sử dụng TF-IDF (Term Frequency – Inverse Document Frequency). Trong các bình luận, có những từ xuất hiện rất nhiều nhưng không mang ý nghĩa phân loại (như "cái", "con", "thì", "là"...). Nếu chỉ đếm số lượng, các từ này sẽ lấn át các từ khóa quan trọng. TF-IDF giải quyết vấn đề này bằng cách:

- TF (Tần suất): Từ nào xuất hiện nhiều trong câu hiện tại. Quan trọng với câu đó.
- IDF (Nghịch đảo tần suất): Từ nào xuất hiện tràn lan ở khắp mọi câu trong bộ dữ liệu. Giảm trọng số của từ đó xuống.

$$w_{x,y} = tf_{x,y} \times log(\frac{N}{df_x})$$

Text1: Basic Linux Commands for Data Science
Text2: Essential DVC Commands for Data Science

	basic	commands	data	dvc	essential	for	linux	science
Text 1	0.5	0.35	0.35	0.0	0.0	0.35	0.5	0.35
Text 2	0.0	0.35	0.35	0.5	0.5	0.35	0.0	0.35

Hình 15. Phương pháp TF-IDF

Kết quả là những từ khóa mang tính "độc hại" đặc trưng (như các từ chửi thề, từ lóng xúc phạm) thường ít xuất hiện trong văn phong thông thường nhưng lại xuất hiện tập trung trong các câu toxic, sẽ có điểm số TF-IDF rất cao. Điều này giúp siêu phẳng (hyperplane) của SVM dễ dàng phân tách hai lớp dữ liệu hơn.

1.2. Cài đặt mô hình SVM:

Sau khi dữ liệu văn bản đã được chuyển đổi thành các vector số học thông qua TF-IDF, nhóm tiến hành xây dựng mô hình phân loại. Nhóm lựa chọn Support Vector Machine (SVM) làm mô hình cơ sở (baseline) để so sánh. SVM là một thuật toán học máy giám sát mạnh mẽ, đặc biệt hiệu quả trong các bài toán phân loại văn bản có số chiều lớn (high-dimensional space) như bài toán này (nơi số lượng đặc trưng từ vựng có thể lên tới hàng nghìn).

Mục tiêu cốt lõi của SVM là tìm ra một siêu phẳng (hyperplane) tối ưu trong không gian n-chiều để phân tách hai lớp dữ liệu: Độc hại (Toxic) và Bình thường (Non-toxic).

Tại sao SVM phù hợp với TF-IDF? Dữ liệu sau khi qua TF-IDF là một ma trận thưa (sparse matrix). Trong không gian đặc trưng này, các điểm dữ liệu văn bản thường có xu hướng phân tách tuyến tính khá tốt. SVM hoạt động bằng cách tối đa hóa "lề" (margin) giữa các điểm dữ liệu gần nhất của hai lớp (gọi là support vectors), giúp mô hình có khả năng tổng quát hóa tốt, hạn chế Overfitting tốt hơn so với các thuật toán như Naive Bayes hay Decision Tree.

Hiệu suất của SVM phụ thuộc rất lớn vào việc lựa chọn tham số. Thay vì chọn ngẫu nhiên, nhóm tập trung phân tích và tinh chỉnh hai tham số quan trọng nhất:

- Kernel (Hàm nhân): Đây là "trái tim" của SVM, quyết định cách mô hình biến đổi không gian dữ liệu.
    - Linear Kernel (Tuyến tính): Đây là lựa chọn ưu tiên cho bài toán phân loại văn bản. Vì số chiều của TF-IDF rất lớn (hàng nghìn từ), dữ liệu thường đã phân tách tuyến tính sẵn trong không gian cao chiều này. Kernel tuyến tính tính toán nhanh và ít bị Overfitting.
    - RBF Kernel (Radial Basis Function): Sử dụng để ánh xạ dữ liệu sang không gian vô hạn chiều nhằm giải quyết các ranh giới phân lớp phi tuyến tính phức tạp. Tuy nhiên, nó tốn kém tài nguyên tính toán hơn nhiều.
- Tham số C (Regularization Parameter): Tham số kiểm soát sự đánh đổi giữa độ phẳng của đường biên và việc phân loại đúng các điểm dữ liệu huấn luyện.
    - C nhỏ (ví dụ 0.1, 1): Chấp nhận một số điểm bị phân loại sai (margin mềm) để có đường biên giới đơn giản hơn. Giúp tránh Overfitting khi dữ liệu nhiễu.
    - C lớn (ví dụ 10, 100): Cố gắng phân loại đúng tuyệt đối mọi điểm dữ liệu huấn luyện (margin cứng). Dễ dẫn đến Overfitting (học vẹt).

Để tìm ra bộ tham số (Kernel, C) tối ưu nhất mà không dựa vào cảm tính, nhóm sử dụng kỹ thuật Grid Search kết hợp với Kiểm định chéo (Cross-Validation). Phương pháp này đảm bảo kết quả đánh giá là khách quan, không phụ thuộc vào việc "ăn may" do cách chia dữ liệu train/test, giúp nhóm chọn được bộ tham số có tính ổn định cao nhất.

- Cấu hình lưới tham số: Nhóm thiết lập một không gian tìm kiếm như sau:

```
param_grid = {
    "kernel": ["linear", "rbf"],
    "C": [0.5, 1.0, 1.2, 2.0]
    # nếu muốn có gamma cho RBF:
    # "gamma": ["scale", 0.1, 0.01]
}
```

Hình 16. Cấu hình lưới tham số

- K-Fold Cross-Validation (K=5): Dữ liệu huấn luyện được chia ngẫu nhiên thành 5 phần (folds). Quá trình huấn luyện diễn ra theo quy trình lặp: 1. Mô hình được train trên 4 phần (80%) và validate trên 1 phần còn lại (20%). 2. Quá trình này lặp lại 5 lần, mỗi lần đổi phần validate khác nhau. 3. Kết quả cuối cùng là trung bình cộng độ chính xác của 5 lần chạy.

Sau khi Grid Search tìm ra bộ tham số tốt nhất (ví dụ: Kernel='linear', C=1), nhóm sử dụng bộ tham số này để huấn luyện lại mô hình trên toàn bộ tập train và đánh giá trên tập test độc lập (Test Set) mà mô hình chưa từng nhìn thấy. Việc cài đặt được thực hiện hoàn toàn trên thư viện Scikit-learn, tận dụng các class SVC, GridSearchCV và classification_report để đảm bảo quy trình chuẩn công nghiệp.

2. Mô hình chính: Fine-tune PhoBERT:

2.1. Tiền xử lý dữ liệu:

Đối với mô hình học sâu PhoBERT, việc chuẩn bị dữ liệu đầu vào đòi hỏi sự chính xác cao hơn nhiều so với các mô hình học máy truyền thống. Trước tiên, nhóm cũng thực hiện hai bước cơ bản là chuẩn hóa văn bản (xử lý teencode, viết tắt) và phân đoạn từ tiếng Việt. Tuy nhiên, thay vì sử dụng phương pháp tách từ thông thường, nhóm sử dụng thư viện Underthesea để nối các từ ghép (do yêu cầu đặc thù của mô hình PhoBERT cần nhận diện các đơn vị ngữ nghĩa tiếng Việt trọn vẹn).

Sau khi dữ liệu văn bản đã được làm sạch và chuẩn hóa, nhóm tiến hành chia tập dữ liệu thành các tập huấn luyện (train), kiểm định (validation) và kiểm thử (test) với tỷ lệ phù hợp (60:20:20) để đảm bảo mô hình được đánh giá khách quan.

Quá trình quan trọng tiếp theo là Tokenization (Mã hóa văn bản). Nhóm sử dụng công cụ AutoTokenizer được cung cấp bởi thư viện transformers của Hugging Face, tải về bộ từ điển gốc của mô hình vinai/phobert-base-v2. Bộ tokenizer này đóng vai trò "cầu nối" giữa ngôn ngữ tự nhiên và ma trận số học, thực hiện các nhiệm vụ cụ thể sau:

- Phân token (Tokenization): Chia văn bản đầu vào thành các đơn vị nhỏ hơn (sub-words) dựa trên thuật toán Byte-Pair Encoding (BPE). Điều này cho phép mô hình xử lý tốt các từ hiếm hoặc từ không có trong từ điển (OOV) bằng cách tách chúng thành các âm tiết cơ sở. Đồng thời, tokenizer tự động chèn các special tokens theo kiến trúc của RoBERTa: thêm token <s> (tương đương [CLS] - đại diện cho phân loại) vào đầu chuỗi và </s> (tương đương [SEP] - phân tách) vào cuối chuỗi.
- Chuẩn hóa độ dài (Padding & Truncation): Các mạng nơ-ron yêu cầu đầu vào dạng tensor có kích thước cố định. Tokenizer sẽ cắt ngắn các câu quá dài vượt ngưỡng cho phép hoặc thêm các token đệm <pad> (có ID là 1) vào các câu ngắn để đảm bảo đồng nhất kích thước.
- Tạo Attention Mask: Sinh ra một vector mặt nạ gồm các giá trị 1 và 0. Giá trị 1 báo hiệu cho mô hình biết đây là từ ngữ thực tế cần xử lý, còn giá trị 0

đại diện cho phần đệm (padding) cần được bỏ qua, giúp tiết kiệm tài nguyên tính toán.

- Ánh xạ Token sang ID: Chuyển đổi từng token văn bản thành một số nguyên duy nhất (Input IDs) dựa trên bộ từ điển gồm 64.000 từ vựng của PhoBERT.

Cụ thể trong quá trình cài đặt, nhóm sử dụng AutoTokenizer để mã hóa dữ liệu với các tham số cấu hình chi tiết như sau:

- padding="max_length": Tham số này chỉ định rằng tất cả các chuỗi đầu vào sẽ được điền thêm các token đệm <pad> để đạt đến độ dài cố định.
- truncation=True: Kích hoạt chế độ cắt ngắn dữ liệu. Nếu một bình luận dài hơn giới hạn cho phép, phần đuôi sẽ bị cắt bỏ để tránh lỗi bộ nhớ.
- max_length=128: Xác định độ dài tối đa cho chuỗi đầu vào (sequence length). Nhóm chọn giá trị 128 dựa trên khảo sát độ dài trung bình của các bình luận, đảm bảo giữ lại đủ thông tin ngữ nghĩa mà không gây lãng phí tài nguyên tính toán.
- return_attention_mask=True: Yêu cầu tokenizer trả về vector mặt nạ chú ý, giúp mô hình phân biệt được nội dung chính và phần đệm.
- return_tensors='pt': Trả về kết quả dưới dạng PyTorch Tensors, định dạng dữ liệu bắt buộc để đưa vào tính toán trên GPU.

```python
def encode_batch(text_list):
    return tokenizer(
        text_list,
        padding="max_length",        # thêm [PAD] để đủ độ dài
        truncation=True,             # cắt nếu dài quá
        max_length=MAX_LEN,
        add_special_tokens=True,     # thêm [CLS], [SEP]
        return_attention_mask=True,
        return_tensors="pt"          # trả về tensor PyTorch
    )
```

Hình 17. Cấu hình sử dụng AutoTokenizer

Sau khi quá trình tokenization hoàn tất, dữ liệu dạng Tensor được đóng gói vào DataLoader với các thiết lập tối ưu:

- batch_size=16: Việc chia dữ liệu thành các lô nhỏ giúp giảm tải áp lực lên bộ nhớ VRAM của GPU và tăng tốc độ hội tụ của mô hình.
- shuffle=True (đối với tập Train): Việc xáo trộn dữ liệu ngẫu nhiên sau mỗi epoch là cực kỳ quan trọng. Nó giúp phá vỡ sự phụ thuộc vào thứ tự xuất hiện của dữ liệu, ngăn chặn mô hình "học vẹt" theo trình tự và cải thiện khả năng tổng quát hóa.

```
train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
val_loader   = DataLoader(val_dataset, batch_size=16, shuffle=False)
test_loader  = DataLoader(test_dataset, batch_size=16, shuffle=False)
```

Hình 18. Thiết lập đóng gói vào DataLoader

2.2. Cài đặt mô hình:

Sau khi hoàn tất tiền xử lý và đóng gói dữ liệu vào các DataLoader, nhóm tiến hành xây dựng kiến trúc mạng nơ-ron và thiết lập quy trình huấn luyện Fine-tuning.

Thay vì sử dụng nguyên bản mô hình PhoBERT chỉ để trích xuất đặc trưng, nhóm xây dựng một lớp mô hình tùy chỉnh có tên PhoBERT_Classifier kế thừa từ nn.Module của PyTorch. Kiến trúc này được thiết kế end-to-end (từ đầu đến cuối) bao gồm hai khối chính:

- Khối Backbone (Trích xuất đặc trưng): Sử dụng mô hình vinai/phobert-base-v2 làm nền tảng. Dữ liệu đi qua 12 lớp Transformer Encoder để tạo ra các vector ngữ cảnh. Nhóm trích xuất vector đặc trưng tại vị trí đầu tiên (tương ứng với token <s>/[CLS]), có kích thước 768 chiều.
- Khối Classifier (Phân loại): Ghép nối trực tiếp vector đặc trưng 768 chiều vào một lớp Tuyến tính (nn.Linear) để nén xuống không gian 2 chiều (tương ứng với 2 nhãn Toxic/Non-toxic).

```
class PhoBERT_Classifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.phobert = phobert            # backbone
        hidden_size = 768                 # phobert-base-v2 hidden dim
        self.fc = nn.Linear(hidden_size, num_classes)
```

Hình 19. Kiến trúc mô hình mà nhóm xây dựng

Quá trình huấn luyện được thực hiện trên môi trường Google Colab sử dụng GPU Tesla T4 để tăng tốc độ tính toán. Các siêu tham số (Hyperparameters) được thiết lập như sau:

- Hàm mất mát (Loss Function): CrossEntropyLoss. Đây là lựa chọn tiêu chuẩn giúp tối ưu hóa độ chính xác phân loại.
- Thuật toán tối ưu (Optimizer): Nhóm sử dụng Adam (torch.optim.Adam).
- Adam là thuật toán tối ưu hóa thích nghi phổ biến, giúp mô hình hội tụ nhanh bằng cách điều chỉnh tốc độ học cho từng tham số riêng biệt. Việc chọn Learning rate nhỏ (lr = 2e-5) là cực kỳ quan trọng khi Fine-tuning để tránh phá vỡ các trọng số đã được tiền huấn luyện kỹ lưỡng của PhoBERT.
- Batch Size: 16

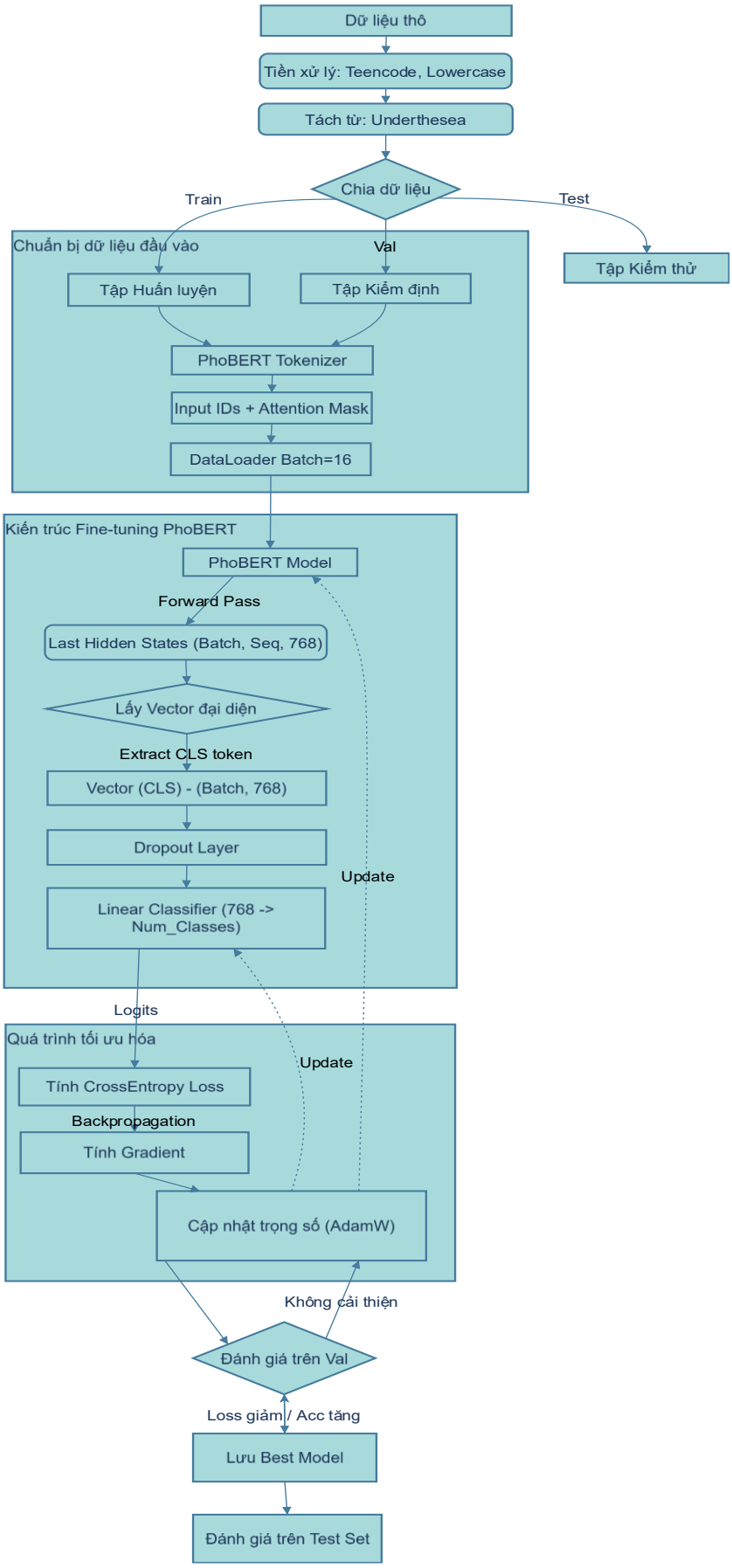
- Epochs: 3

Nhóm thiết lập vòng lặp huấn luyện tiêu chuẩn:

- Training: Tại mỗi epoch, mô hình tính toán loss và cập nhật trọng số trên tập Train.
- Validation: Ngay sau mỗi epoch, mô hình được đánh giá trên tập Validation.

```
Epoch 1/3 | Train loss: 0.3009  acc: 0.8753 | Val loss: 0.2414  acc: 0.9020
Epoch 2/3 | Train loss: 0.1737  acc: 0.9363 | Val loss: 0.2329  acc: 0.9035
Epoch 3/3 | Train loss: 0.1135  acc: 0.9628 | Val loss: 0.2691  acc: 0.9040
```

Hình 20. Huấn luyện mô hình Fine-tune PhoBERT



Hình 21. Mô hình hệ thống

Chương V. ĐÁNH GIÁ

1. Phân tích kết quả đạt được:

Để đánh giá hiệu năng của các mô hình, nhóm sử dụng tập kiểm thử (Test Set) gồm 2.000 mẫu (tương ứng 20% dữ liệu) hoàn toàn độc lập, chưa từng xuất hiện trong quá trình huấn luyện. Các chỉ số được sử dụng để đánh giá bao gồm: Accuracy (Độ chính xác), Precision (Độ chính xác dự báo), Recall (Độ phủ), F1-Score và Confusion Matrix.

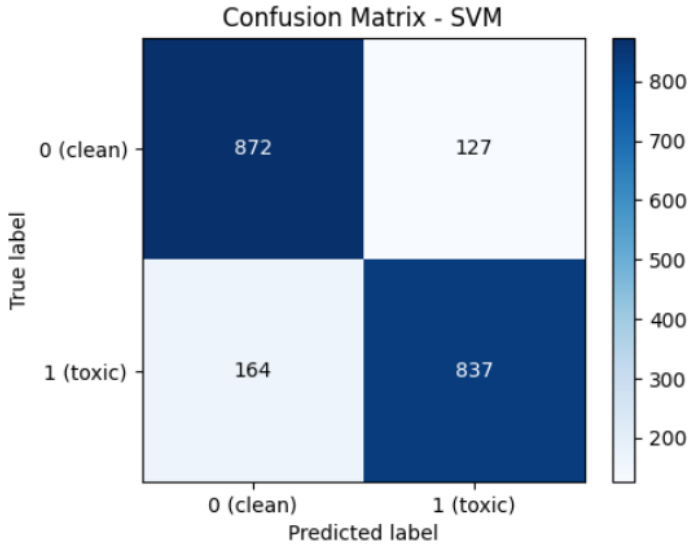
1.1. Kết quả mô hình SVM (Baseline):

Mô hình SVM với bộ tham số tối ưu (Kernel='linear', C=1.0) kết hợp trích chọn đặc trưng TF-IDF cho kết quả như sau:

- Accuracy: 85.45%

Class	Precision	Recall	F1-Score
0 (non-toxic)	0.8417	0.8729	0.8570
1 (toxic)	0.8683	0.8362	0.8519
Trung bình	0.8550	0.8545	0.8545

Bảng 3. Bảng báo cáo chi tiết kết quả mô hình SVM



Hình 22. Confusion Matrix của mô hình SVM

Mô hình SVM đạt kết quả khá tốt với độ chính xác hơn 85%. Tuy nhiên, chỉ số Recall của lớp Toxic chỉ đạt 83.62%. Điều này có nghĩa là mô hình bỏ lọt khá nhiều bình luận độc hại (164 mẫu bị gán nhãn sai là an toàn). Nguyên nhân chủ yếu là do phương pháp TF-IDF chỉ dựa vào tần suất từ khóa mà không hiểu được ngữ cảnh hoặc các cấu trúc câu phức tạp (ví dụ: các câu chửi thề ẩn ý, không dùng từ tục tĩu trực tiếp).

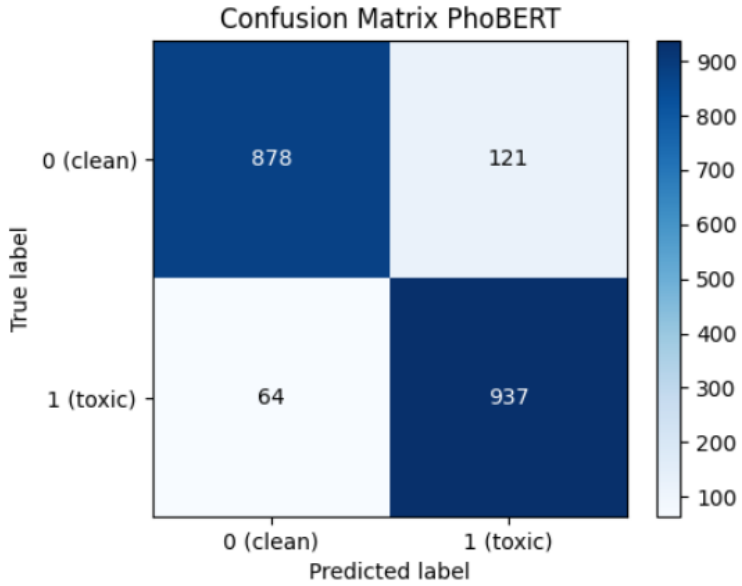
1.2. Kết quả mô hình PhoBERT (Fine-tunning):

Sau 3 epochs huấn luyện, mô hình PhoBERT cho thấy sự cải thiện vượt bậc:

- Accuracy: 90.75% (Tăng ~5.3% so với SVM)

Class	Precision	Recall	F1-Score
0 (non-toxic)	0.9321	0.8789	0.9047
1 (toxic)	0.8856	0.9361	0.9102
Trung bình	0.9088	0.9075	0.9074

Bảng 4. Bảng báo cáo kết quả chi tiết mô hình Fine-tune PhoBERT



Hình 23. Confusion Matrix của mô hình Fune-tune PhoBERT

PhoBERT thể hiện sự vượt trội, đặc biệt ở chỉ số Recall lớp Toxic đạt tới 93.61%. Số lượng bình luận độc hại bị bỏ lọt (False Negative) giảm mạnh từ 164 xuống còn 64 trường hợp. Điều này chứng tỏ mô hình đã "hiểu" sâu hơn về ngữ nghĩa, phát hiện được các mẫu câu độc hại tinh vi mà mô hình truyền thống bỏ qua.

1.3. So sánh và phân tích chuyên sâu:

Để làm rõ hiệu quả của việc áp dụng mô hình ngôn ngữ tiền huấn luyện, nhóm thực hiện so sánh đối chứng giữa hai phương pháp:

- Về khả năng phát hiện độc hại (Recall - Toxic Class): Đây là chỉ số quan trọng nhất trong bài toán. SVM gặp hạn chế khi gặp các từ mới (OOV) hoặc các câu không chứa từ khóa chửi thề rõ ràng. SVM coi các từ là độc lập, không nắm bắt được thứ tự từ. Còn PhoBERT, nhờ cơ chế Self-Attention, mô hình nắm bắt được ngữ cảnh toàn cục. Ví dụ, câu "Mày khôn thế này bao giờ mới chết" chứa toàn từ tích cực ("khôn"), nhưng PhoBERT hiểu

được sắc thái mỉa mai trong ngữ cảnh đó để gán nhãn Toxic, trong khi SVM dễ bị nhầm là Non-toxic.

- Về F1-Score: PhoBERT đạt F1-Score 0.91 so với 0.85 của SVM. Sự cân bằng giữa Precision và Recall cho thấy PhoBERT không chỉ "bắt nhầm còn hơn bỏ sót" mà thực sự phân loại chính xác và đáng tin cậy hơn.
- Phân tích lỗi (Error Analysis): Mặc dù đạt 90.75%, PhoBERT vẫn sai ở 64 mẫu Toxic (False Negative) và 121 mẫu Non-toxic (False Positive). Qua phân tích thủ công, nhóm nhận thấy các nguyên nhân chính:
    - Từ lóng/Teencode mới: Một số từ lóng quá mới hoặc quá hiếm chưa được xử lý triệt để trong bước tiền xử lý.
    - Sự mơ hồ (Ambiguity): Các câu bình luận tranh luận gay gắt nhưng không dùng từ ngữ xúc phạm trực tiếp (ranh giới giữa "tranh luận" và "công kích" rất mong manh).
    - Dữ liệu gán nhãn: Tồn tại một số nhãn gán sai (Label noise) trong tập dữ liệu gốc, khiến mô hình bị học sai.

Kết quả thực nghiệm khẳng định giả thuyết ban đầu của nhóm: Việc sử dụng mô hình ngôn ngữ lớn (LLM) như PhoBERT kết hợp Fine-tuning mang lại hiệu quả vượt trội so với các phương pháp học máy truyền thống trong bài toán xử lý ngôn ngữ tự nhiên tiếng Việt, đặc biệt là với dữ liệu phức tạp như bình luận mạng xã hội.

2. Phân tích các trường hợp sai:

Bình luận	Dự đoán	Thực tế	Phân tích và nhận xét
đm thằng đầu bồui lớp trưởng đại học tốt nghiệp 3 8 xong kiếm học bổng đi du học châu âu thạc sĩ xong về làm cđb gì không biết nhưng mà 27 tuổi mua nhà mua santafe 29 tuổi đổi santafe sang glc300 giờ có thêm 2 con nhà mặt phố đéo ở đến mang cho thuê	0	1	Câu văn bắt đầu bằng những từ chửi thề rất nặng ("đm", "đầu bồui"). Tuy nhiên, phần lớn nội dung phía sau (chiếm 90% độ dài) lại mô tả về sự thành công, giàu có ("tốt nghiệp", "học bổng", "mua nhà", "xe Santafe", "nhà mặt phố"). Có thể do cơ chế Self-Attention của PhoBERT bị "phân tán" bởi quá nhiều từ khóa mang ngữ nghĩa tích cực/mô tả sự việc ở phần sau, làm "pha loãng" (dilute) tín hiệu độc hại ở đầu câu. Mô hình hiểu nhầm đây là một câu kể chuyện hoặc ngưỡng mộ thay vì chửi bới.

hài hước mấy người như friend còn không biết gì về tâm lý tư tưởng mà cứ truyền bá những cái lý thuyết giáo điều đơn cử như cái chân đau friend còn không giải quyết được chuyện đau khổ của nó mà còn đòi giải quyết được sự đau khổ của thất tình	1	0	Giọng văn mang tính tranh luận gay gắt, chỉ trích đối phương ("không biết gì", "giáo điều"). Tuy nhiên, nó không dùng từ tục tĩu. Mô hình PhoBERT rất nhạy cảm với các cấu trúc câu mang tính công kích cá nhân hoặc giọng điệu mỉa mai ("hài hước mấy người"). Mô hình đã đánh đồng giữa Tranh luận gay gắt (Aggressive Debate) và Độc hại (Toxic).
vòng này 2pillz mất tích lun mai vòng trước chơi với tamke đã phế flop dập mu rồi vòng này lại tamke thì như cc lun có 2pillz mà éo ai biết tận dụng	0	1	Câu này sử dụng từ viết tắt "cc" (cụm từ tục tĩu phổ biến) và "éo". Ngữ cảnh là bình luận về một chương trình giải trí (Rap/Game). Có thể do ngữ cảnh là nhận xét về chuyên môn/giải trí khiến mô hình coi đây là lời chê bai bình thường (Negative sentiment) chứ chưa đủ ngưỡng để coi là độc hại/tấn công (Toxic). Đây là ranh giới mờ nhạt giữa "chê bai gay gắt" và "xúc phạm".
biết lên voz là hơn được khối người rồi nói chứ có đi học không vậy có bằng cấp gì không	0	1	Đây là một câu hỏi mang tính khinh miệt ("có đi học không vậy"). Cụm từ "có đi học không" hay "có não không" là những mẫu câu (pattern) rất phổ biến trong các bình luận xúc phạm trí tuệ. Mô hình đã học được pattern này và đánh nhãn Toxic là điều dễ hiểu. Ở đây có sự mơ hồ trong gán nhãn (Label Ambiguity), vì câu này thực sự là Toxic.
người có gần tỷ mua con xe cũng được gọi là gà à	0	1	Câu này chứa từ "gà" (nghĩa bóng là kém cỏi/ngu ngốc). Tuy nhiên, đây là câu hỏi tu từ dùng để phản biện/bảo vệ người mua xe ("Có ai gọi người giàu thế là gà đâu?"). Mô hình bắt được từ khóa tiêu cực "gà" nhưng chưa hiểu trọn

			vẹn cấu trúc câu hỏi tu từ/phủ định (Rhetorical Question). Nó nghĩ rằng câu này đang chửi người mua xe là "gà".

Bảng 5. Phân tích một số trường hợp sai

Từ việc phân tích các mẫu sai trên, nhóm rút ra 3 vấn đề cốt lõi mà mô hình hiện tại đang gặp phải:

- Vấn đề về ngữ cảnh dài và hỗn hợp (Context Dilution): Khi một câu chứa từ độc hại ngắn nhưng đi kèm với một đoạn văn dài mang ngữ nghĩa trung tính hoặc tích cực, mô hình dễ bị "xao nhãng" và bỏ qua tính độc hại.
- Sự nhập nhằng giữa "Tiêu cực" và "Độc hại": Mô hình đôi khi quá nhạy cảm, đánh đồng những lời chỉ trích, tranh luận gay gắt hoặc câu hỏi mỉa mai là hành vi độc hại, dẫn đến tỷ lệ báo động giả (False Positive) cao ở các câu không chứa từ tục tĩu.
- Thách thức của tiếng lóng và từ đa nghĩa: Các từ lóng viết tắt mới lạ hoặc từ đa nghĩa ("gà") khi đặt trong các cấu trúc câu phức tạp (phủ định, nghi vấn) vẫn là thách thức lớn đối với khả năng hiểu ngữ nghĩa của mô hình.

Chương VI. KẾT LUẬN

Trong khuôn khổ đồ án này, nhóm đã tập trung nghiên cứu và giải quyết bài toán nhận diện bình luận độc hại trên mạng xã hội, sử dụng kỹ thuật Fine-tuning trên mô hình ngôn ngữ tiền huấn luyện PhoBERT làm phương pháp chủ đạo và thực hiện so sánh đối chứng với mô hình học máy truyền thống SVM. Từ kết quả thực nghiệm thu được, có thể khẳng định rằng PhoBERT đã thể hiện sự ưu việt vượt trội so với SVM về mọi chỉ số đánh giá. Cụ thể, với độ chính xác tổng thể đạt 90.75% và đặc biệt là chỉ số Recall cho lớp độc hại lên tới 93.61%, PhoBERT đã khắc phục được nhược điểm chí mạng của SVM (chỉ đạt 85.45% Accuracy và 83.62% Recall). Sự chênh lệch đáng kể này xuất phát từ bản chất kiến trúc của hai mô hình: trong khi SVM kết hợp với TF-IDF chỉ thuần túy hoạt động dựa trên thống kê tần suất từ vựng (bag-of-words) và thường thất bại trước các câu mang hàm ý mỉa mai, ẩn ý hoặc cấu trúc phức tạp, thì PhoBERT – với cơ chế Self-Attention và tri thức tiền huấn luyện khổng lồ – lại có khả năng thấu hiểu sâu sắc ngữ cảnh hai chiều và nắm bắt được các đặc trưng ngữ nghĩa tinh vi của tiếng Việt.

Bên cạnh việc tối ưu hóa thuật toán, nhóm cũng nhận thức rõ vai trò then chốt của quá trình tiền xử lý dữ liệu, đặc biệt đối với dữ liệu văn bản trên mạng xã hội vốn chứa nhiều nhiễu và biến thể. Việc áp dụng linh hoạt các kỹ thuật như chuẩn hóa teencode, xử lý từ lóng và đặc biệt là phân đoạn từ chuyên biệt bằng thư viện Underthesea đã giúp chuẩn hóa dữ liệu đầu vào, đóng góp trực tiếp vào việc nâng cao hiệu suất phân loại cho cả hai mô hình. Tuy nhiên, nhóm cũng nhìn nhận rằng việc triển khai PhoBERT đòi hỏi tài nguyên tính toán lớn hơn và thời gian huấn luyện lâu hơn so với sự đơn giản, gọn nhẹ của SVM.

Tổng kết lại, qua quá trình thực hiện đề tài, nhóm không chỉ nắm vững quy trình xây dựng một hệ thống xử lý ngôn ngữ tự nhiên từ khâu thu thập, làm sạch đến huấn luyện mô hình, mà còn chứng minh được tiềm năng to lớn của việc ứng dụng các mô hình ngôn ngữ lớn (LLM) vào bài toán an toàn thông tin. Đây sẽ là nền tảng vững chắc để nhóm tiếp tục phát triển các hướng tiếp cận mới, như mở rộng bộ từ điển lóng, tối ưu hóa thời gian suy luận để xây dựng các ứng dụng cảnh báo thời gian thực, cũng như tiếp cận các bài toán phân tích cảm xúc đa chiều phức tạp hơn trong tương lai.

PHỤ LỤC

- Bộ ngữ liệu tarudesu/VOZ-HSD: https://huggingface.co/datasets/tarudesu/VOZ-HSD
- Thư viện tách từ và xử lý Tiếng Việt underthesea: https://github.com/undertheseanlp/underthesea
- Thư viện hỗ trợ vector hóa TF-IDF và thuật toán SVM Scikit-learn: https://scikit-learn.org/
- Mô hình huấn luyện PhoBERT base v2: https://huggingface.co/vinai/phobert-base-v2
- Toàn bộ mã nguồn huấn luyện lưu tại Google Colab Notebook: https://colab.research.google.com/drive/11eJaHQnFlTwIeEIiwS9jhoAUQeSxWVBh?usp=drive_link

TÀI LIỆU THAM KHẢO

[1] Viện FMIT – Từ điển quản trị chuẩn mực quốc tế. “stopword removal là gì”. https://fmit.vn/en/glossary/stopword-removal-la-gi

[2] Nguyen, D. Q., & Nguyen, A. T. (2020). PhoBERT: Pre-trained language models for Vietnamese. Findings of the Association for Computational Linguistics: EMNLP 2020, 1037-1042.

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in neural information processing systems, 30.

[5] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.