

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH

---



ĐỒ ÁN MÔN HỌC  
CS221 – XỬ LÝ NGÔN NGỮ TỰ NHIÊN  
X SENTIMENT ANALYSIS

Giảng viên hướng dẫn : TS. Nguyễn Trọng Chính  
ThS. Đặng Văn Thìn  
ThS. Nguyễn Đức Vũ

Sinh viên thực hiện 1 : Phạm Thanh Lâm  
Mã sinh viên 1 : 21520055  
Sinh viên thực hiện 2 : Trương Quang Nghĩa  
Mã sinh viên 2 : 21522376  
Sinh viên thực hiện 3 : Nguyễn Gia Toàn  
Mã sinh viên 3 : 21521546  
Lớp : CS221.O22

Tp HCM, tháng 6 năm 2024

# BẢNG PHÂN CÔNG THỰC HIỆN ĐỒ ÁN MÔN HỌC

Họ tên SV1: <b>Phạm Thanh Lâm</b> MSSV: <b>21520055</b>	Họ tên SV2: <b>Trương Quang Nghĩa</b> MSSV: <b>21522376</b>
Phương pháp Bert	Phương pháp XGBoost
Làm báo cáo	Làm báo cáo
Làm slide	Làm slide
Cài đặt mô hình	Cài đặt mô hình
Họ tên SV3: <b>Nguyễn Gia Toàn</b> MSSV: <b>21521546</b>	
Phương pháp Logistic Regression	
Làm báo cáo	
Làm slide	
Cài đặt mô hình	
<b>SV thực hiện 1</b> <i>(Ký tên)</i>	<b>SV thực hiện 2</b> <i>(Ký tên)</i>
<b>Phạm Thanh Lâm</b>	<b>Trương Quang Nghĩa</b>
<b>SV thực hiện 3</b> <i>(Ký tên)</i>	
<b>Nguyễn Gia Toàn</b>	

## LỜI CẢM ƠN

Chúng em xin bày tỏ lòng biết ơn chân thành đến thầy Nguyễn Trọng Cảnh, thầy Đặng Văn Thìn và thầy Nguyễn Đức Vũ đã giảng dạy chúng em môn học Xử lý ngôn ngữ tự nhiên. Sự hướng dẫn và kiến thức sâu rộng của thầy đã vô cùng quý giá trong suốt quá trình thực hiện làm đồ án môn học và viết báo cáo về chủ đề "X Sentiment Analysis".

Chúng em cũng xin gửi lời cảm ơn đến các thành viên trong nhóm vì sự tận tâm và tinh thần đồng đội trong việc hoàn thành báo cáo này. Những đóng góp và ý kiến của từng thành viên đã rất cần thiết trong việc thực hiện nghiên cứu phương pháp, phân tích dữ liệu và đưa ra kết luận.

Ngoài ra, chúng em xin cảm ơn những cá nhân đã hỗ trợ, góp ý và giúp đỡ trong quá trình thực hiện công việc. Những ý kiến quý báu và góp ý xây dựng của họ đã góp phần quan trọng vào chất lượng của báo cáo này.

Cuối cùng, chúng em xin ghi nhận các nguồn tài liệu, tham khảo và công cụ đã đóng vai trò quan trọng trong dự án cuối cùng của chúng em. Sự sẵn có của các tài liệu này đã làm phong phú thêm hiểu biết và kiến thức của chúng em về chủ đề và hỗ trợ thực hiện đồ án.

Chúng em xin chân thành cảm ơn.

Sinh viên thực hiện

Trương Quang Nghĩa - Phạm Thanh Lâm

Nguyễn Gia Toàn

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Tp.HCM, ngày 28 tháng 06 năm 2024

GVHD

TS. Nguyễn Trọng Chính

ThS. Đặng Văn Thìn

ThS. Nguyễn Đức Vũ

# MỤC LỤC

Phần 1: GIỚI THIỆU BÀI TOÁN .....	6
Phần 2: BỘ NGỮ LIỆU BÀI TOÁN .....	7
2.1 Thu thập dữ liệu:.....	7
2.2 Các quy tắc chủ giải ngữ liệu .....	7
2.3 Thống kê ngữ liệu .....	8
2.4 Một số mẫu bình luận và nhãn tương ứng của nó trong tập ngữ liệu.....	12
Phần 3: PHƯƠNG PHÁP THỰC HIỆN .....	25
3.1 Phương pháp BERT .....	25
3.1.1 Kiến trúc mô hình Transformer .....	25
3.1.2 Kiến trúc mô hình Bert .....	26
3.1.3 Pretraining task .....	27
3.1.4 Huấn luyện mô hình Bert.....	30
3.2 Fine Tuning .....	35
3.3 Các chỉ số đánh giá .....	38
Phần 4: CÀI ĐẶT VÀ THỬ NGHIỆM .....	40
4.1 Giới thiệu .....	40
4.2 Cài đặt phương pháp của nhóm .....	40
4.2.1 Mô hình BERT .....	40
4.2.2 Kết quả cài đặt .....	42
4.3 Cài đặt phương pháp so sánh.....	43
4.3.1 Mô hình Logistic Regression .....	43
4.3.1.1 Phân tích dữ liệu:.....	43
4.3.1.2 Xây dựng mô hình .....	44

4.3.1.3 Đánh giá và tinh chỉnh.....	45
4.3.2 Mô hình XGBoost.....	46
4.3.2.1 Phân tích dữ liệu .....	46
4.3.2.2 Biểu diễn dữ liệu .....	46
4.3.2.3 Huấn luyện và dự đoán mô hình .....	47
4.3.2.4 Đánh giá và tinh chỉnh.....	47
4.4 Phân tích một vài mâu sai.....	48
Phần 5: KẾT LUẬN .....	51
TÀI LIỆU THAM KHẢO .....	52

## Phần 1:

# GIỚI THIỆU BÀI TOÁN

Phân tích cảm xúc (Sentiment Analysis) là một lĩnh vực quan trọng trong Xử lý Ngôn ngữ Tự nhiên (NLP), được sử dụng để xác định và phân loại cảm xúc từ văn bản. Điều này cho phép chúng ta hiểu rõ hơn về cảm xúc và quan điểm của người dùng trong các bài viết, đặc biệt là trên các nền tảng truyền thông xã hội như Twitter.

- **Bối cảnh và Tầm quan trọng:** Trong thời đại số, mạng xã hội đóng vai trò quan trọng trong việc chia sẻ và truyền tải thông tin. Twitter, với hàng triệu bài đăng hàng ngày, trở thành một nguồn dữ liệu phong phú để phân tích các xu hướng và cảm xúc của công chúng. Phân tích cảm xúc từ các bài đăng này có thể giúp các doanh nghiệp, nhà nghiên cứu và tổ chức hiểu rõ hơn về cảm nhận của công chúng, từ đó hỗ trợ việc ra quyết định và xây dựng chiến lược hiệu quả.
- Mục tiêu của đề tài: "X Sentiment Analysis" tập trung vào việc phân loại cảm xúc của các bài tweet thành ba nhóm chính:
  - Tích cực (Positive)
  - Tiêu cực (Negative)
  - Trung lập (Neutral)

Với lượng dữ liệu khổng lồ được tạo ra mỗi ngày trên Twitter, việc xử lý và phân tích dữ liệu này không chỉ đòi hỏi các phương pháp kỹ thuật hiệu quả mà còn cần hiểu rõ về cách xây dựng và quản lý tập dữ liệu để đảm bảo tính chính xác và độ tin cậy của kết quả phân tích.

Thông qua việc phân loại này, chúng ta có thể hiểu được xu hướng cảm xúc của người dùng đối với các sự kiện, sản phẩm, hoặc chủ đề cụ thể được đề cập trên Twitter.

## Phần 2:

# BỘ NGỮ LIỆU BÀI TOÁN

### 2.1 Thu thập dữ liệu:

Bộ dữ liệu nhóm sử dụng cho đồ án là bộ dữ liệu “Twitter Sentiment Analysis Dataset”. Đây là bộ ngữ liệu bằng tiếng anh, là tập dữ liệu phân tích cảm xúc theo từng thực thể trên Twitter.

Số mẫu của bộ dữ liệu này là 75682, gồm có đặc trưng là twitter id (ID của bài đăng), entity (thực thể được nhắc đến trong bài đăng), sentiment (cảm xúc được phân loại là Positive, Negative, Neutral hoặc Irrelevant), và content (nội dung của bài đăng). Các dòng tweet được chia làm ba nhóm tương ứng với 3 nhãn là: Negative (0), Neutral (1) và Positive (2).

Chọn lọc các dòng tweet có liên quan đến chủ đề nghiên cứu bằng cách sử dụng các từ khóa hoặc hashtag nhất định. Điều này giúp tập trung vào những dòng tweet có khả năng mang thông tin cảm xúc rõ ràng.

### 2.2 Các quy tắc chủ giải ngữ liệu

Quá trình gán nhãn đòi hỏi các quy tắc rõ ràng để thực hiện gán nhãn ngữ liệu đảm bảo tính nhất quán và chính xác. Các quy tắc bao gồm:

Tích cực (Positive): Dòng tweet biểu đạt cảm xúc vui vẻ, hài lòng, lạc quan hoặc tích cực về một sự kiện, sản phẩm hoặc dịch vụ

**Ví dụ:** happy birthday red dead redemption that shit changed my life what a crazy experience.

Tiêu cực (Negative): Dòng tweet biểu đạt cảm xúc buồn bã, giận dữ, thất vọng hoặc tiêu cực về một sự kiện, sản phẩm hoặc dịch vụ.

**Ví dụ:** What does that say about Microsoft hardware & software security - The Man gets hacked

Trung tính (Neutral): Dòng tweet không biểu đạt rõ ràng cảm xúc tích cực hay tiêu cực, thường là các thông tin hoặc nhận xét trung lập.

**Ví dụ:** dnv the device i'm using is iOS.

### 2.3 Thống kê ngữ liệu

Việc thống kê ngữ liệu giúp hiểu rõ hơn về cấu trúc và phân bố của tập dữ liệu, từ đó chúng ta có thể điều chỉnh cần thiết để cải thiện dữ liệu trước khi áp dụng vào các mô hình học máy. Dưới đây là một số thống kê cơ bản mà nhóm đã thực hiện.

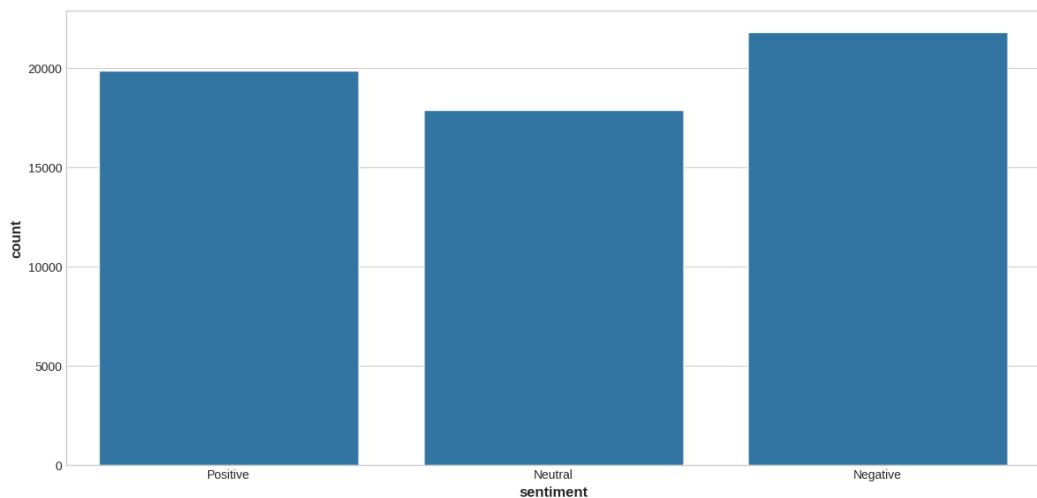
Thực hiện đọc dữ liệu và xác định tổng số dòng tweet có trong tập dữ liệu. Bộ dữ liệu sẽ có 75682 mẫu dữ liệu và gồm 4 đặc trưng là tweetid, entity, sentiment và content.

	tweetid	entity	sentiment	content
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...

Hình 2.1: 5 mẫu dữ liệu đầu tiên trong bộ dữ liệu

Thực hiện tiền xử lý dữ liệu như kiểm tra các mẫu dữ liệu có giá trị null hoặc trùng lặp nhau thì thực hiện loại bỏ các mẫu dữ liệu này. Cùng với đó nhóm thực hiện loại bỏ tất cả các mẫu có nhãn là không liên quan Irrelevant ra khỏi dữ liệu dùng để xây dựng mô hình máy học.

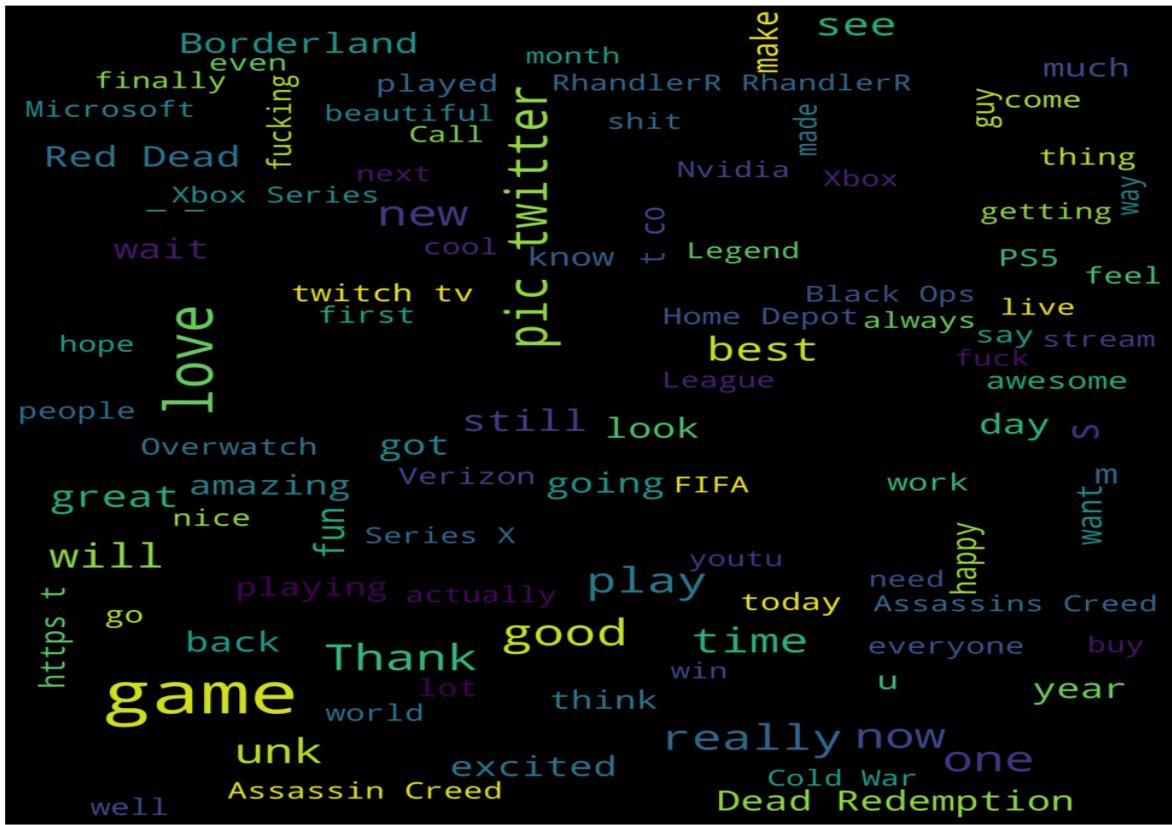
Phân bố của các nhãn cảm xúc lần lượt là 21790 mẫu dữ liệu có nhãn là negative, 19846 nhãn positive và 17879 nhãn trung tính neutral.



Hình 2.2: Biểu đồ thể hiện số lượng mẫu dữ liệu

Độ dài của các dòng tweet mà nhóm sử dụng là từ 1 đến 100 từ trên mỗi tweet.

Trực quan hóa các từ phổ biến nhất được liên kết với một tình cảm cụ thể (ví dụ: tích cực, tiêu cực, trung tính) trong tập dữ liệu văn bản. Bằng cách tạo ra các đám mây từ, ta có thể nhanh chóng hiểu được các chủ đề và thuật ngữ nổi bật dành cho các tình cảm khác nhau.



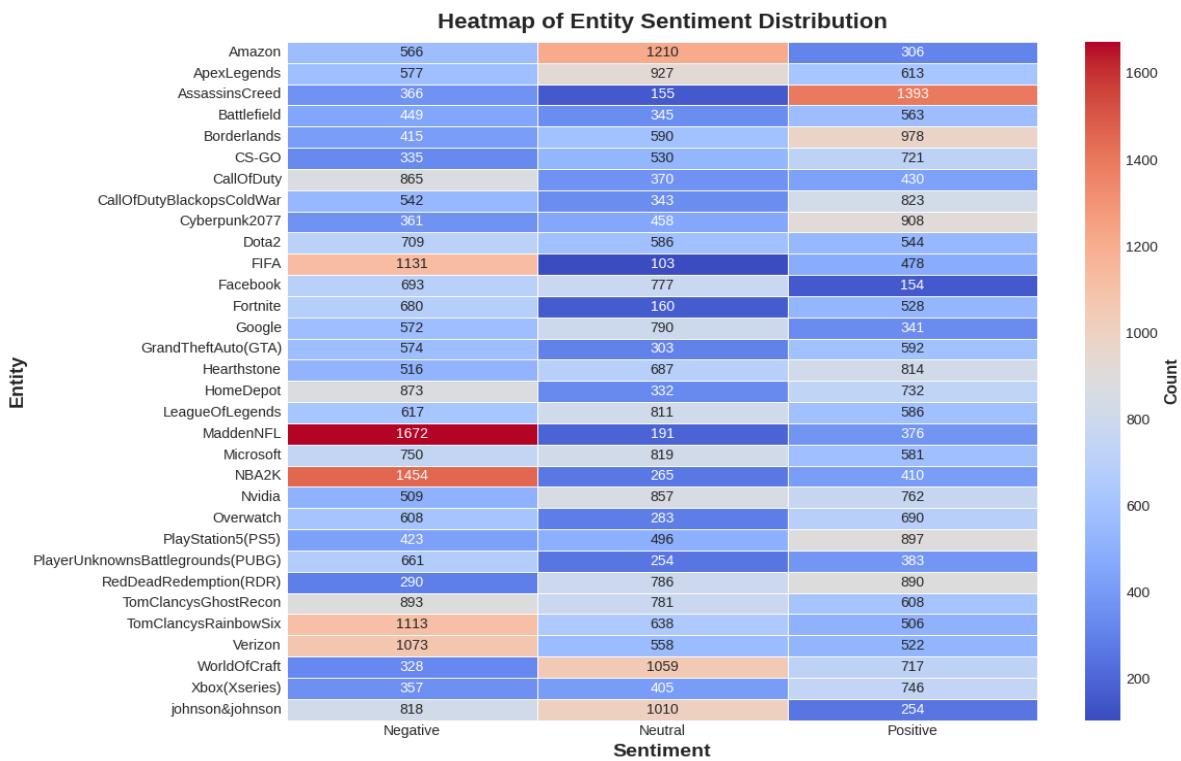
Hình 2.3: Từ phổ biến trong câu tweet positive



Hình 2.4: Từ phổ biến trong câu tweet negative



Hình 2.5: Từ phổ biến trong câu tweet neutral



Hình 2.6: *Bảng thể hiện sự phân bố của các cảm xúc đối với từng thực thể trong tập dữ liệu.*

Phân tích sự phân bố cảm xúc: Heatmap giúp hiển thị sự phân bố của các cảm xúc đối với từng thực thể trong tập dữ liệu. Ta có thể dễ dàng nhìn thấy các mẫu cảm xúc nào thường xuất hiện cùng với các thực thể nào.

#### **2.4 Một số mẫu bình luận và nhãn tương ứng của nó trong tập ngũ liệu**

Nhãn	Tweet	Mô tả
Positive	The professional dota 2 scene is fucking exploding and I completely welcome it	Người nói rất hào hứng với sự phát triển của môn thể thao điện tử Dota 2
	Itching to assassinate.	Dù ngũ cảnh không rõ ràng, từ "itching" thường được dùng để diễn đạt sự háo hức hoặc mong đợi.
	yeah, and it's fun	Thể hiện cảm xúc tích cực về một trải nghiệm nào đó. Từ yeah và fun thường được dùng để diễn đạt sự tích cực
	happy birthday red dead redemption that shit changed my life what a crazy experience	Chúc mừng sinh nhật và bày tỏ cảm xúc tích cực về trò chơi Red Dead Redemption và ảnh hưởng của nó đối với cuộc đời

	The things I would do for a @nvidia3090... unspeakable! 	Người nói rất khao khát và mong muốn một card đồ họa NVIDIA 3090, thể hiện sự hào hứng.
	my ass still knee-deep in Assassins Creed Odyssey with no way out anytime soon lmao	Người nói đang cảm thấy vui vẻ khi chơi game Assassins Creed Odyssey, mặc dù đã bị cuốn hút sâu vào trò chơi.
	I like the killstreaks	Người nói thích tính năng killstreak trong một trò chơi. Từ like thể hiện sự thích thú của họ
	I love @Rainbow6Game so much ❤️	Người nói yêu thích trò chơi Rainbow Six: Siege rất nhiều.
	@satyanadella @Microsoft thanks for celebrating #Diversity. We need positive energy these days.	Bày tỏ lòng biết ơn đối với việc ủng hộ đa dạng và mong muốn có thêm năng lượng tích cực.
	this is the absolute FUNNIEST interaction I've ever seen on League of Legends pic.twitter.com/NsWyuMdVrX	Từ funniest thể hiện sự vui vẻ nhất trong ngữ cảnh là chưa từng thấy trong League of Legends

	This looks kinda clean!	Bày tỏ ấn tượng tích cực về điều gì đó
	Great job guys @Peter_shirley and @withyounotsmwhr	Bày tỏ sự đánh giá tích cực đối với công việc của Peter Shirley và người khác được đề cập.
	Interesting... (for real, this is really awesome looking!)	Điển đạt sự kinh ngạc và hứng thú tích cực về điều gì đó có vẻ rất ấn tượng.
	Happy to be back doing what I love :))	Hạnh phúc khi quay lại làm những gì mình yêu thích.
	This #PlayStation5 pre order is an absolute....	Dự đoán rằng việc đặt hàng trước PlayStation 5 là một sự kiện tuyệt vời.
	Nothing like getting two leavers in the game that would have promoted you. I love league of legends 	Mặc dù có điều gì đó không như mong đợi trong trò chơi League of Legends, nhưng vẫn diễn đạt sự yêu thích và tích cực với trò chơi.
	the chopper shot is stunning 	Điển đạt sự ấn tượng với cảnh quay của máy bay trực thăng.

	Hella excited for tonight, as always! 😊	Hào hứng cho đêm nay như mọi khi. Từ tích cực ở đây là excited
	Good fucking day today happy to be back more GTA RP tmr 🤝 ❤️	Từ today happy mô tả rằng hôm nay là 1 ngày tuyệt vời
	Good to know 👏	Biểu thị sự hài lòng và khẳng định tích cực về thông tin đã được chia sẻ.
Negative	Damn	Từ này thường được dùng để biểu lộ sự ngạc nhiên hoặc bất mãn. Ở đây, nó có thể diễn tả sự bất mãn hoặc sự thất vọng đối với một tình huống nào đó.
	Ghost of Tsushima is a better assassins creed game than modern assassins' creed	Người viết cho rằng Ghost of Tsushima cung cấp trải nghiệm tốt hơn so với các phiên bản gần đây của Assassin's Creed. Thể hiện sự không hài lòng trước đó
	Nice bug @Rainbow6Game	Đây là một lời châm biếm về việc phát hiện ra một lỗi (bug) trong trò chơi Rainbow Six Siege. Bằng

		cách đề cập đến tag @Rainbow6Game, người viết có thể muốn gửi thông điệp đến nhà phát triển hoặc cộng đồng để chú ý đến vấn đề này.
	And the US WANTS TO TRUST THIS COMPANY with a VACCINE?	Đây là một câu hỏi gián tiếp, thể hiện sự lo ngại về khả năng của một công ty
	@RockstarGames how the hell is gta online STILL this fucking broken???	Người viết bày tỏ sự bất mãn với việc GTA Online vẫn chưa được sửa chữa hoặc cải thiện, có thể từ khi nó vẫn còn tồn tại trong một trạng thái "đứt đoạn" hoặc bị lỗi.
	and I cant even get a lvl 1 battle pass. life is unfair 😠	Đây là một lời than phiền về việc không thể mua được battle pass ở mức độ 1 của một tựa game. Sử dụng biểu tượng cảm xúc 😠 để thể hiện sự tiếc nuối và bất mãn.
	I can't stand your ass OMG!	một câu tò ra tức giận hoặc khó chịu đói với ai

		đó hoặc một tình huống cụ thể.
@PlayApex I have problems to buy the battlepass with the new patch		Người viết thông báo rằng họ gặp vấn đề khi mua battle pass trong trò chơi Apex Legends sau khi có bản cập nhật mới.
Facebook is a hub of fake information.		một phán đoán tiêu cực về Facebook, cho rằng nó là một nơi chứa đựng thông tin giả mạo và không đáng tin cậy.
Pissing people off in FIFA and on twitter 🤢 🤢 🤢		Người viết tỏ ra bất mãn với việc làm tức giận người khác trong trò chơi FIFA và trên Twitter, sử dụng biểu tượng cảm xúc 🤢 để thể hiện sự tiếc nuối và bất mãn.
dead game 😞		Biểu thị sự thất vọng khi một trò chơi không còn phát triển hoặc thu hút người chơi như trước. Biểu tượng cảm xúc 😞 thể hiện sự buồn bã.

	Delete techies pls fck u @DOTA2	bình luận cay đắng, yêu cầu xóa bỏ Techies, một nhân vật trong Dota 2. Việc gắn thẻ @DOTA2 có thể là để gửi thông điệp tới nhà phát triển hoặc cộng đồng của tựa game.
	damn just want my 100lp back can't have shit in league of legends	Bày tỏ sự thất vọng vì mất điểm Liên Minh Huyền Thoại (League of Legends). "Can't have shit" mang ý nghĩa không thể giữ được điều gì đó quan trọng.
	My experience with ASSASSIN'S CREED: ODYSSEY	Câu chuyện hoặc nhận xét về trải nghiệm chơi Assassin's Creed: Odyssey, có thể là tích cực hoặc tiêu cực.
	2k games is never the same again 💔 🎉	Biểu thị sự tiếc nuối về sự thay đổi không tích cực trong các tựa game của 2K Games. Biểu tượng cảm xúc 💔 🎉 thể hiện sự đau buồn và tiếc nuối.

	@NBA2K if 2K21 anything like 20 again I'm never buying a game from y'all again. Make the game like 2K16	Đe doạ rằng nếu NBA 2K21 tương tự như 2K20, người viết sẽ không mua thêm bất kỳ tựa game nào từ 2K Games nữa. Người viết mong muốn trò chơi giống như 2K16, một phiên bản được cho là tốt hơn.
	So glad I never joined Facebook.	Biểu thị niềm vui vì không tham gia vào Facebook, có thể là do quan điểm về vấn đề về thông tin giả mạo và quyền riêng tư trên nền tảng này.
	NOT FREE TO USE !!!! Recent work #Fortnite pic.twitter.com/P9VMUkt49	một thông điệp hoặc lời bình luận về việc không được phép sử dụng một công việc gần đây liên quan đến Fortnite một cách miễn phí.
	#Pubg is no more available on Android Playstore and ios	Bày tỏ sự bất mãn hoặc thất vọng vì PUBG không còn có sẵn trên Google

		Play Store và App Store của iOS nữa
	now i'm just offended	Biểu thị sự bức bối hoặc tức giận vì một tình huống cụ thể hoặc hành động của ai đó.
Neutral	RED DEAD REDEMPTION 2	Chỉ đơn giản là đề cập đến tựa game Red Dead Redemption 2, không kèm với ý kiến hoặc cảm xúc đặc biệt.
	Team JerseyBoys is now also represented on Twitch.	Thông tin về việc đội JerseyBoys đã có mặt trên Twitch, không chứa ý kiến hoặc cảm xúc đặc biệt.
	 Red Dead Redemption 2 pic.twitter.com/2XB1cpjLxL	Chỉ đơn giản là một bức ảnh từ Red Dead Redemption 2, không kèm với ý kiến hoặc cảm xúc đặc biệt.
	@Respawn <sup>[PDI]</sup> <sup>[LR]</sup> @PlayApex <sup>[PDI]</sup> #singing	Đưa ra hashtag #singing kèm theo thẻ @Respawn và @PlayApex, có thể là để nhắc đến các lần hát

		hoặc sự kiện trong trò chơi.
	Playing fifa with my girl. She got her first goal against me, and someone won't shut up	Mô tả một trải nghiệm chơi game FIFA, có sự tham gia của bạn gái và sự hào hứng khi cô ấy ghi được bàn thắng đầu tiên.
	Finnish CS Player Jampi Sues Valve Over Alleged VAC Ban » TalkEsport,bit.ly/3byCjrT	Đưa ra thông tin về việc cầu thủ CS người Phần Lan Jampi kiện Valve về cáo buộc bị cấm VAC (Valve Anti-Cheat), không có ý kiến hoặc cảm xúc đặc biệt.
	Early adopters get arsed once again.	Mô tả một vấn đề hoặc cảm xúc về việc sớm áp dụng công nghệ hoặc sản phẩm.
	Stream on Borderlands 3 tonight at 8pm!	Thông báo về việc stream trò chơi Borderlands 3 vào tối nay lúc 8 giờ, không chứa ý kiến hoặc cảm xúc đặc biệt.
	I am dota 2 dota 2 i like happy - Casey	Một câu thoại hoặc trích dẫn liên quan đến trò chơi Dota 2, với một sự tham

		chiếu nhẹ nhàng đến tình trạng tâm trạng của nhân vật.
	People who killed Michael or Trevor at the end of gta probably doing hooligan shenanigans right about now	Một lời nhận xét hoặc phán đoán về hành động của những người chơi GTA V đã lựa chọn giết Michael hoặc Trevor trong phần kết của trò chơi.
	die with honor stfu and press ur bkb	Câu này không chứa đựng những từ ngữ nặng nề, xúc phạm hoặc có tính chất tấn công mạnh mẽ. Nó mang tính chất lời khuyên hoặc hướng dẫn trong bối cảnh chơi game
	Seems like #Playstation has the marketing deal for #CallOfDutyBlackOpsColdWar	Đưa ra nhận định rằng PlayStation có thỏa thuận tiếp thị cho Call of Duty: Black Ops Cold War, không có ý kiến tích cực hay tiêu cực.
	Johnson&Johnson to stop selling baby powder in US.	Thông tin về quyết định của Johnson & Johnson

		ngừng bán bột talc cho trẻ sơ sinh tại Hoa Kỳ, không phân biệt tính cực hay tiêu cực.
	This could go very well...or horribly wrong	Một lời nhận xét về một tình huống không chắc chắn, người viết đang đánh giá sự tiềm năng của một việc gì đó có thể thành công hoặc thất bại.
	The Corruption and Knocking up of Overwatch Babes is a serious Mood atm... pic.twitter.com/QNy4kmgBLV	Đây là một bình luận châm biếm hoặc góc nhìn về các vấn đề nổi lên trong cộng đồng Overwatch, có thể liên quan đến các sự kiện hoặc nhận định chung về trò chơi.
	Solo Q and this freak is spinning as fast as he can to lower the FPS.	Mô tả một tình huống trong game khi đối thủ đang xoay nhanh nhất có thể để giảm FPS (khung hình mỗi giây), có thể là để gây khó chịu cho người chơi khác.

	Great play dude , what a good optic for the mk2 Carbine too 	Lời khen về một pha chơi hay trong game, đồng thời đánh giá cao lựu chọn mục tiêu (optic) cho súng mk2 Carbine.
	"#gtc20 - nice, motivational, and very accessible Nvidia/AI product fair + related tech talks"	Đánh giá tích cực về sự kiện GTC (GPU Technology Conference) 2020 của Nvidia, mô tả nó là một sự kiện thú vị, truyền cảm hứng và dễ tiếp cận, với các hội thảo về công nghệ AI.
	Umm @PlayApex when I died it said Bug This pic.twitter.com/bzMHzbadOF	Người viết đang báo cáo về một lỗi (bug) họ gặp phải khi chơi Apex Legends, và họ đính kèm một hình ảnh để minh họa vấn đề này.
	CSGO WIngman (Im Silver dont bully) twitch.tv/lprezh	Người viết đang chia sẻ thông tin về việc chơi game CSGO trong chế độ Wingman trên kênh Twitch của họ, cũng nhắc nhở rằng họ là Silver rank và mong không bị bắt nạt.

## Phần 3: PHƯƠNG PHÁP THỰC HIỆN

### 3.1 Phương pháp BERT

#### 3.1.1 Kiến trúc mô hình Transformer

Transformer là một kiến trúc mô hình học sâu dựa trên cơ chế self-attention, cho phép mô hình này hiểu được mối quan hệ giữa các từ trong một câu mà không cần đến kiến trúc tuần tự truyền thống như RNN (Recurrent Neural Networks) hay LSTM (Long Short-Term Memory). Transformer có khả năng xử lý toàn bộ câu cùng một lúc, điều này giúp tăng tốc độ huấn luyện và cải thiện hiệu quả xử lý.

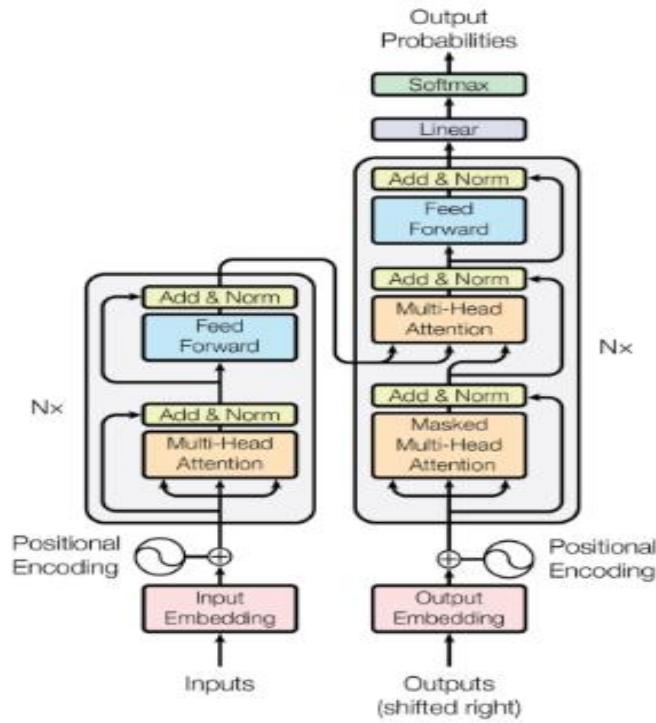
Cốt lõi của transformer là attention mechanism (cơ chế tập trung), giúp mô hình tập trung vào các phần quan trọng của văn bản để đưa ra dự đoán chính xác hơn.

Transformer được cấu trúc thành hai phần chính là encoder và decoder.

Encoder: Encoder xử lý dữ liệu đầu vào và nén dữ liệu vào vùng nhớ hoặc context mà Decoder có thể sử dụng sau đó.

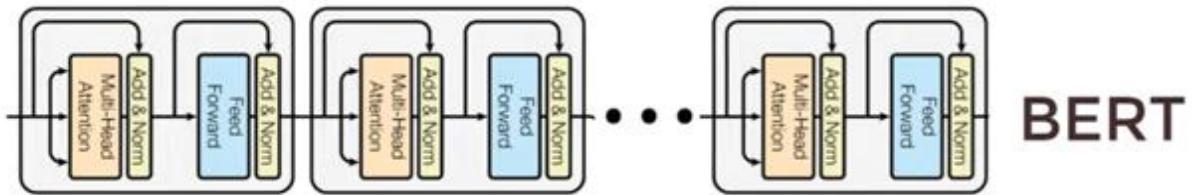
Decoder: Decoder nhận đầu vào từ đầu ra của Encoder (gọi là "Encoded input") kết hợp với một chuỗi đầu vào khác (gọi là "Target") để tạo ra chuỗi đầu ra cuối cùng.

Mỗi encoder và decoder đều bao gồm nhiều lớp, mỗi lớp chứa các thành phần self-attention và feed-forward neural networks.



Hình 3.1 Kiến trúc mô hình Transformer

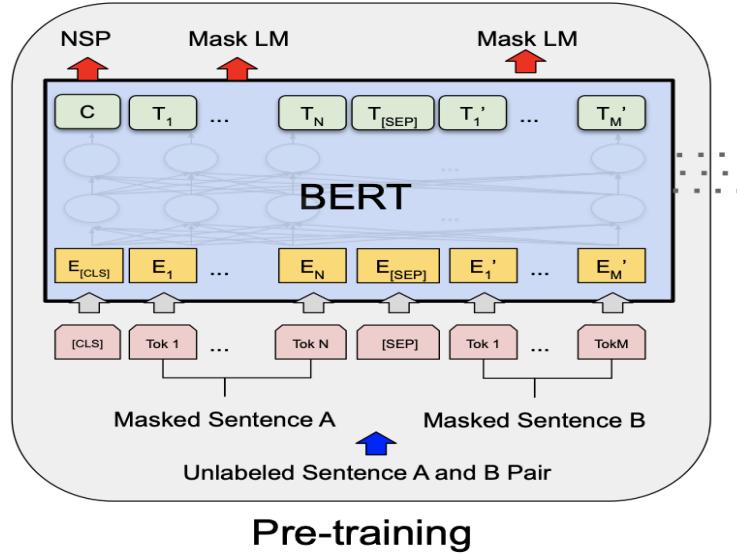
### 3.1.2 Kiến trúc mô hình Bert



Hình 3.3 Kiến trúc mô hình Bert

Các khối encoder tạo ra các embedding cho tất cả các từ một cách đồng thời trong câu. Những embedding này là những vector chứa đựng ý nghĩa của các từ trong câu. Khối encoder học ngôn ngữ là gì, học cú pháp của ngôn ngữ và quan trọng nhất là sẽ học được ngữ cảnh của ngôn ngữ. Vì vậy nếu xếp chồng các khối encoder này với nhau chúng ta sẽ có được mô hình BERT (Bidirectional Encoder Representation from Transformer )

### 3.1.3 Pretraining task



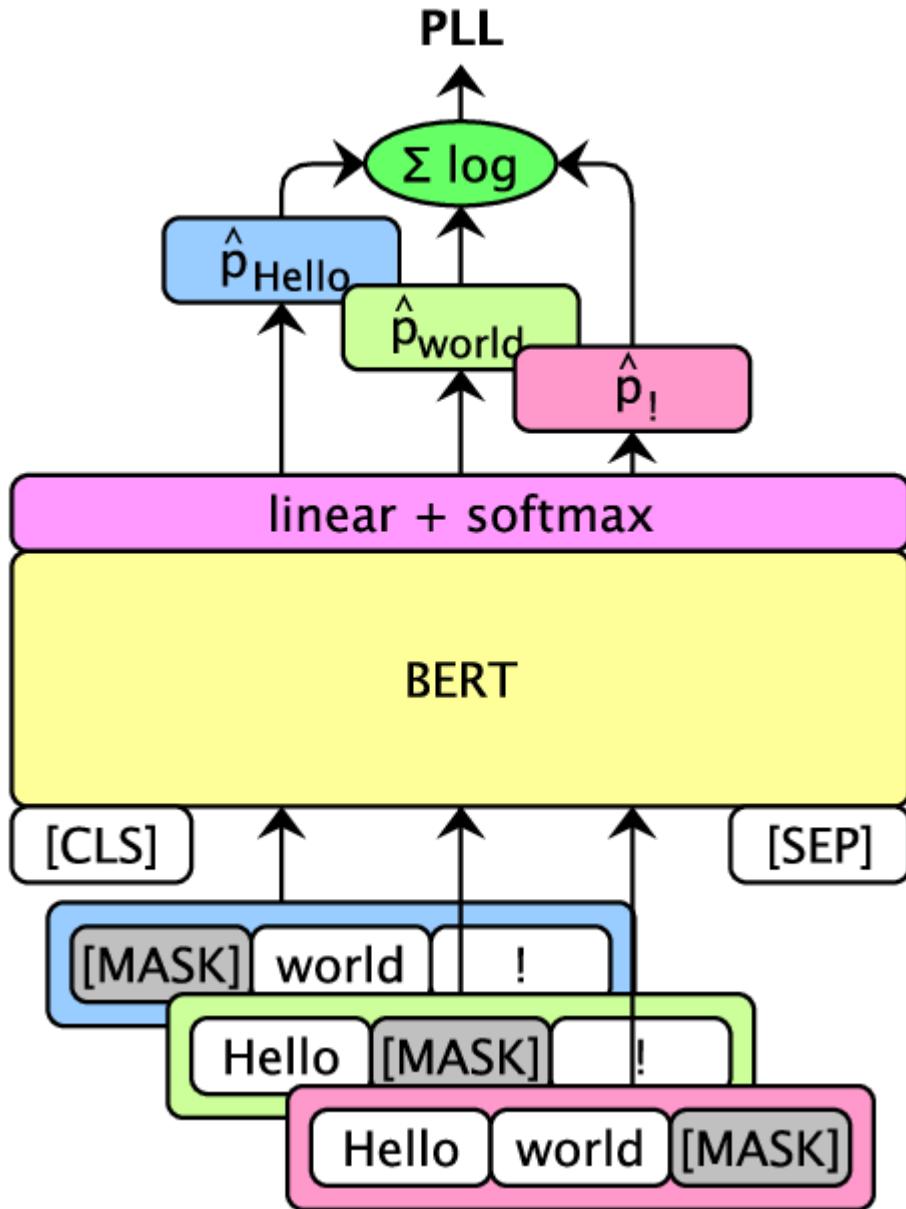
Hình 3.2: Pre-training

#### Masked LM(MLM)

Masked LM giúp cho BERT hiểu được mối quan hệ giữa các từ trong câu và học được biểu diễn phản ánh ngữ nghĩa của từ dựa trên ngữ cảnh.

Masking tokens: Ngẫu nhiên chọn một số token trong câu để thay thế bằng token [MASK].

- o 80% tokens được chọn sẽ bị thay thế bởi [MASK].
- o 10% tokens được chọn sẽ bị thay thế bằng một token ngẫu nhiên từ từ điển.
- o 10% còn lại sẽ được giữ nguyên.



Hình 3.4: Masked Language Model

Dự đoán tokens: BERT sử dụng các token còn lại để dự đoán giá trị của các token [MASK].

- Đầu ra của mô hình là xác suất của từng từ trong từ điển để điền vào vị trí của các token [MASK].

Lợi ích: MLM giúp BERT học được biểu diễn ngữ nghĩa của các từ dựa trên ngữ cảnh toàn bộ câu, bao gồm cả ngữ cảnh từ trái sang phải và từ phải sang trái (bidirectional context).

## Next Sentence Prediction (NSP)

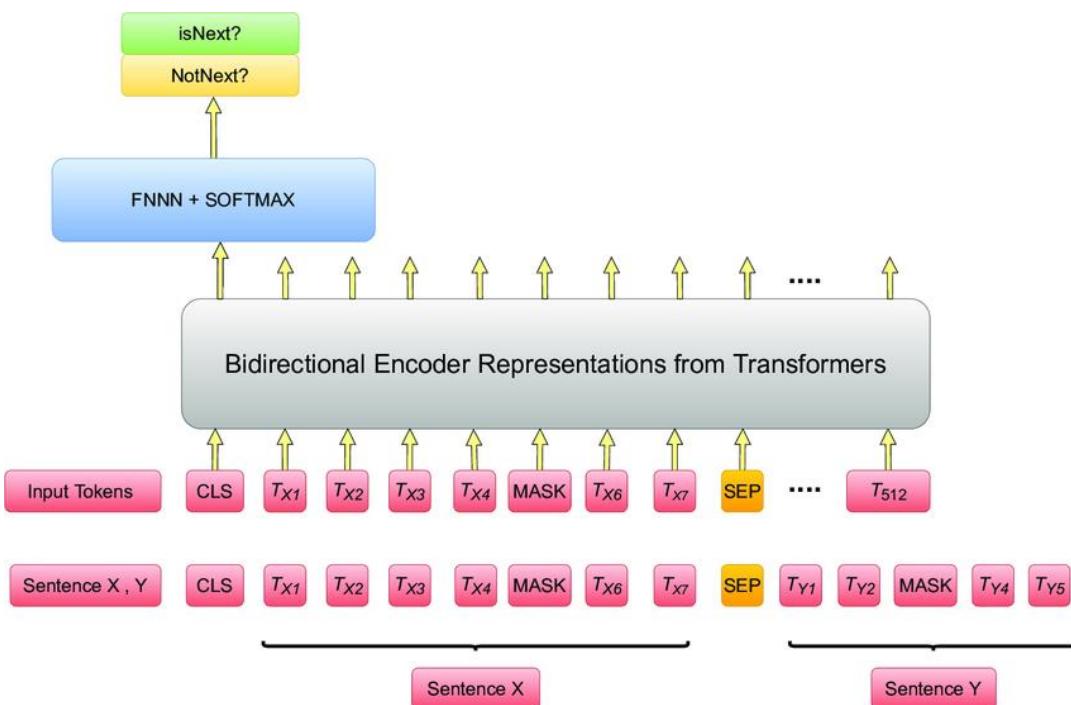
Mục đích: NSP giúp BERT hiểu mối quan hệ giữa hai câu và cải thiện khả năng xử lý các tác vụ yêu cầu sự hiểu biết liên quan giữa các câu.

**Chọn cặp câu:** Với mỗi mẫu huấn luyện, chọn ngẫu nhiên một cặp câu X và câu Y.

**Dự đoán câu tiếp theo:** BERT dự đoán xem câu Y có phải là câu tiếp theo của câu X hay không.

50% trường hợp, câu Y là câu tiếp theo của câu X.

50% trường hợp còn lại, câu Y là một câu ngẫu nhiên khác trong tập dữ liệu.



Hình 3.5: Next Sentence Prediction

Đào tạo: Sử dụng hàm mất mát để tối ưu hóa dự đoán của BERT cho việc dự đoán câu tiếp theo.

Lợi ích: NSP giúp BERT hiểu được mối quan hệ giữa các câu và cải thiện khả năng của mô hình trong các tác vụ như hỏi đáp (question answering) hay phân tích cảm xúc (sentiment analysis) dựa trên ngữ cảnh của đoạn văn

Hai tokens đặc biệt là [CLS] (class) và [SEP] (separator). Mục đích ở đây là kiểm tra xem câu Y có phải là câu tiếp theo của câu X hay không. Đầu tiên chúng ta lấy tokens [CLS] để biểu diễn và giải quyết nhiệm vụ nhị phân. Ta đã có các câu tự nhiên với X, Y là một cặp câu và nhãn (1/0) đại diện cho nhãn ‘isNext’. Trong quá trình huấn luyện, chúng ta truyền vào inputs X, Y và một nhãn, nhiệm vụ của mô hình là tối đa hóa xác suất của nhãn (1/0)

### 3.1.4 Huấn luyện mô hình Bert

#### Input và quá trình embedding

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	# #ing	[SEP]
Token Embeddings	$E_{[CLS]}$	$E_{my}$	$E_{dog}$	$E_{is}$	$E_{cute}$	$E_{[SEP]}$	$E_{he}$	$E_{likes}$	$E_{play}$	$E_{##ing}$	$E_{[SEP]}$
Segment Embeddings	$E_A$	$E_A$	$E_A$	$E_A$	$E_A$	$E_A$	$E_B$	$E_B$	$E_B$	$E_B$	$E_B$
Position Embeddings	$E_0$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$	$E_8$	$E_9$	$E_{10}$

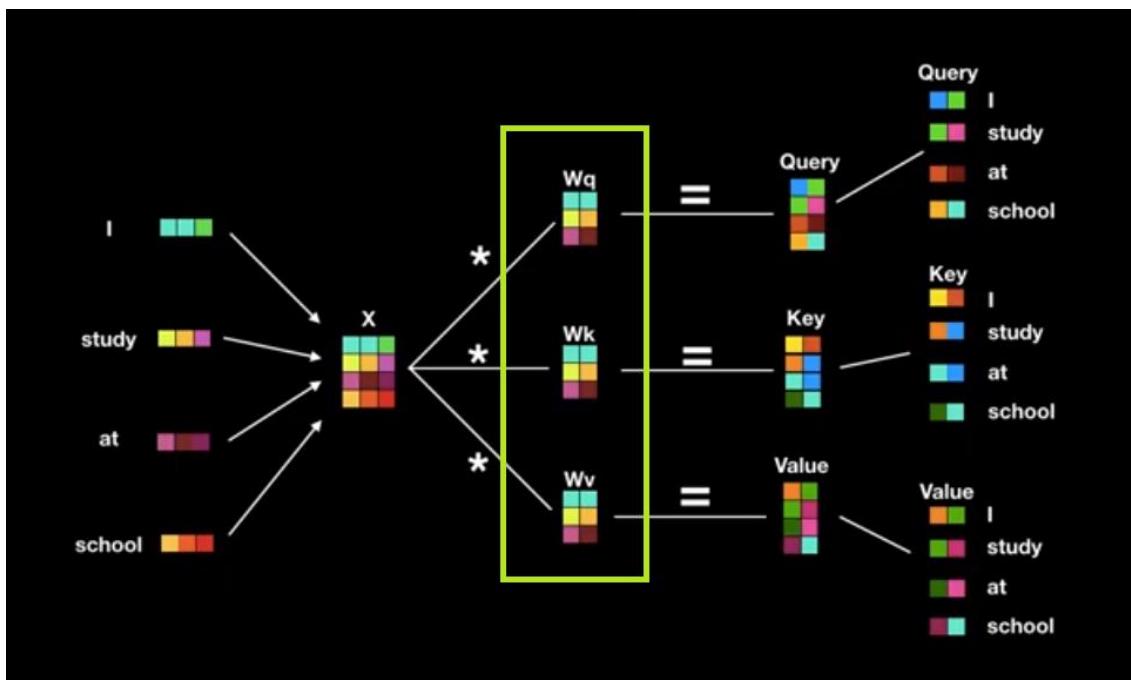
Hình 3.6: Input mô hình và quá trình embedding

- Token Embedding: Một token [CLS] được thêm vào chuỗi input ở vị trí đầu tiên của chuỗi và token [SEP] được thêm vào cuối mỗi câu.
- Segment Embedding: Một đánh dấu xác định câu A hay câu B được thêm vào mỗi token. Việc này cho phép khôi encoder phân biệt giữa các câu trong chuỗi input.

- Position Embedding: Một embedding vị trí được thêm vào mỗi token để chỉ ra vị trí của nó trong chuỗi.

### Cơ chế self attention

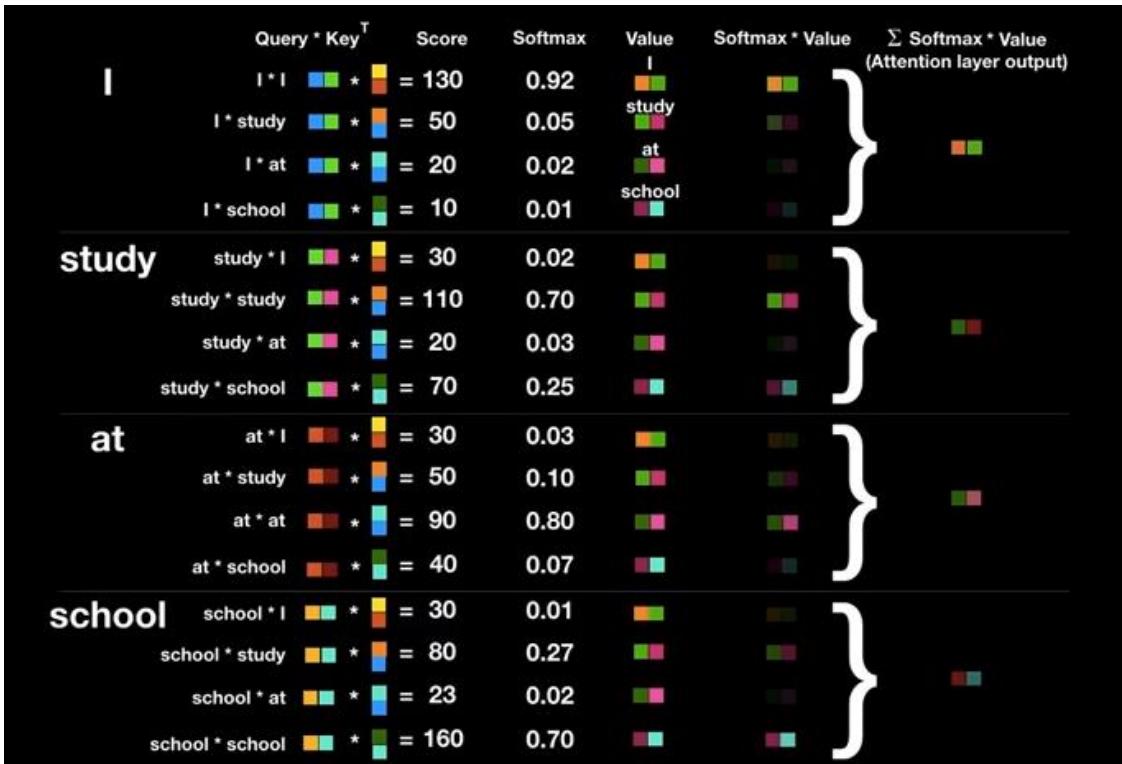
Scale dot product attention: là một cơ chế self-attention khi mỗi từ có thể điều chỉnh trọng số của nó cho các từ khác trong câu sao cho từ ở vị trí càng gần nó nhất thì trọng số càng lớn và càng xa thì càng nhỏ dần. Sau bước nhúng từ (đi qua embedding layer) ta có đầu vào của encoder và decoder là ma trận  $X$  kích thước  $m \times n$ ,  $m, n$  lần lượt là độ dài câu và số chiều của một vector nhúng từ (768).



Hình 3.7: Quá trình thực hiện self attention

khung màu vàng là 3 ma trận  $W_q$ ,  $W_k$ ,  $W_v$  chính là những hệ số mà model cần huấn luyện. Sau khi nhân các ma trận này với ma trận đầu vào  $X$  ta thu được ma trận  $Q$ ,  $K$ ,  $V$  (tương ứng với trong hình là ma trận Query, Key và Value). Ma trận Query và Key có tác dụng tính toán ra phân phối score cho các cặp từ (giải thích ở hình 6). Ma trận Value sẽ dựa trên phân phối score để tính ra véc tơ phân phối xác suất output. Như vậy mỗi một từ sẽ được gán bởi 3 vector query, key và value là các dòng của  $Q$ ,  $K$ ,  $V$ .

Để tính ra score cho mỗi cặp từ trong câu, chúng ta sẽ sử dụng scale dot product giữa các query với key để tìm ra mối liên hệ trong trọng số của các cặp từ. Sau đó dùng softmax để chuẩn hóa và đưa về xác xuất mà độ lớn sẽ đại diện cho mức độ attention của từ query với key. Trọng số càng lớn càng chứng tỏ từ  $w_i$  trả về một sự chú ý lớn hơn đối với từ  $w_j$ . Sau đó chúng ta nhân hàm softmax với các vector giá trị của từ hay còn gọi là value vector để tìm ra vector đại diện (attention vector) sau khi đã học trên toàn bộ câu input.



Hình 3.8: Kết quả tính toán attention cho câu “I study at school”

Đầu vào để tính attention sẽ bao gồm ma trận  $Q$  (mỗi dòng của nó là một vector query đại diện cho các từ input), ma trận  $K$  (tương tự như ma trận  $Q$ , mỗi dòng là vector key đại diện cho các từ input). Hai ma trận  $Q, K$  được sử dụng để tính attention mà các từ trong câu trả về cho 1 từ cụ thể trong câu. attention vector sẽ được tính dựa trên trung bình có trọng số của các vector value trong ma trận  $V$  với trọng số attention (được tính từ  $Q, K$ ).

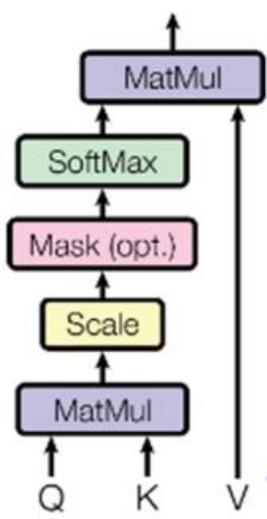
Phương trình Attention như sau:

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

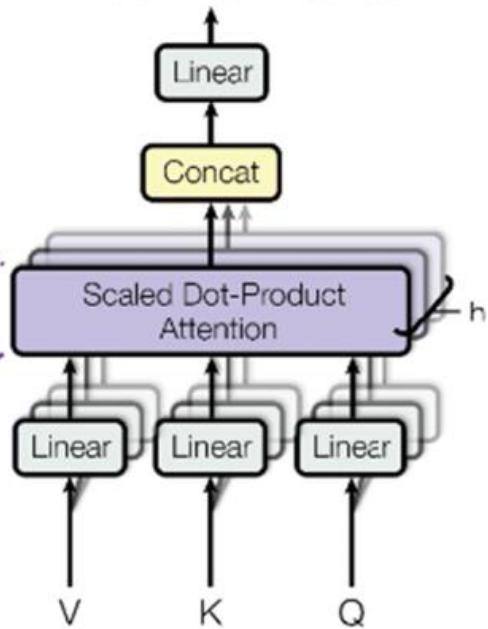
Chia cho  $d_k$  là số dimension của vector key nhằm mục đích tránh tràn luồng nếu số mõi là quá lớn.

### Multi head Attention

#### Scaled Dot-Product Attention



#### Multi-Head Attention



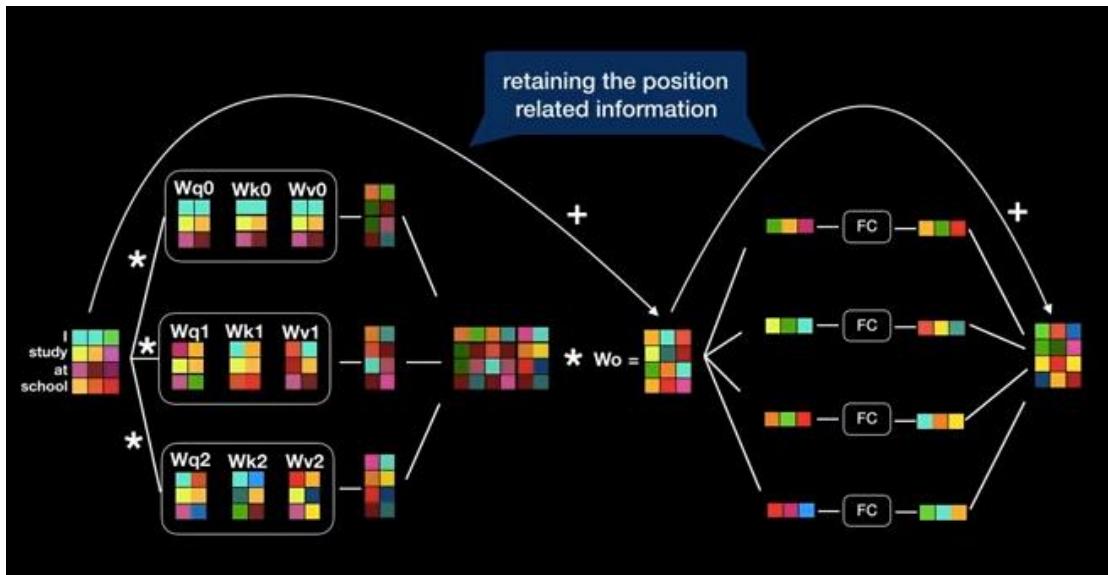
Hình 3.9: Multi-Head attention

Sau quá trình Scale dot production chúng ta sẽ thu được 1 ma trận attention. Các tham số mà model cần tinh chỉnh chính là các ma trận  $Wq$ ,  $Wk$ ,  $Wv$ . Mỗi quá trình như vậy được gọi là 1 head của attention. Khi lặp lại quá trình này nhiều lần ta sẽ thu được quá trình Multi-head Attention.

Sau khi thu được 3 matrix attention ở đầu ra chúng ta sẽ concatenate các matrix này theo các cột để thu được ma trận tổng hợp multi-head matrix có chiều cao trùng với chiều cao của ma trận input.

$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{concatenate}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \mathbf{W}_0$  Ở đây  
 $\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$

Để trả về output có cùng kích thước với ma trận input chúng ta chỉ cần nhân với ma trận  $W_0$  chiều rộng bằng với chiều rộng của ma trận input.



Hình 3.10: Sơ đồ của một block layer áp dụng multi-head attention layer

Như vậy kết thúc quá trình trên là chúng ta đã hoàn thành sub-layer thứ nhất của Transformer là multi-head Attention layer. Ở sub-layer thứ 2 chúng ta sẽ đi qua các kết nối fully connected và trả ra kết quả ở đầu ra có shape trùng với input. Mục đích là để chúng ta có thể lặp lại các block này Nx lần.

### Feed forward network

Triển khai mạng Nơ-ron lan truyền Xuôi (FFN) trong mô hình Bert với 2 lớp tuyến tính hoặc là hai lớp dense. Lớp dense đầu tiên có kích thước là ( $d_{\text{model}}$ ,  $d_{\text{ffn}}$ ), và lớp thứ hai có kích thước là ( $d_{\text{ffn}}$ ,  $d_{\text{model}}$ ).

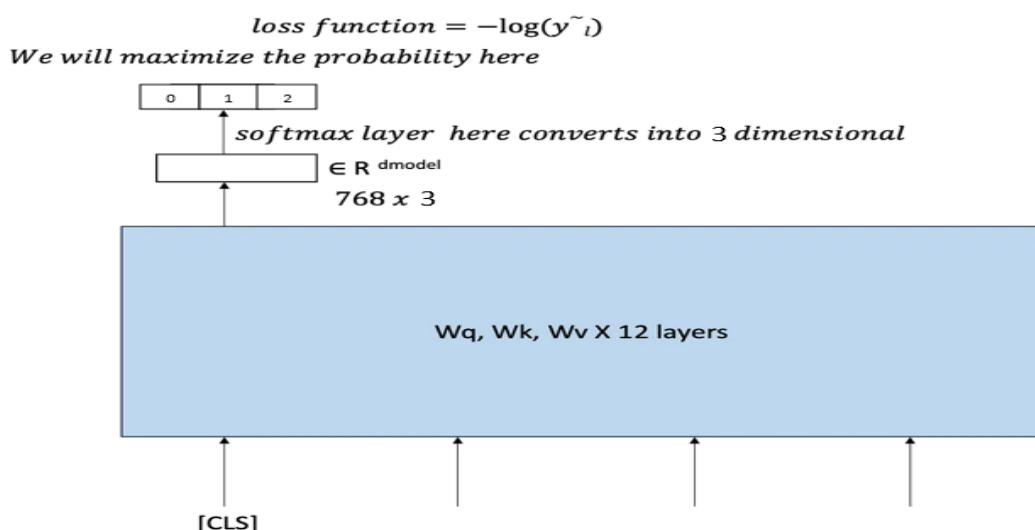
Công thức toán học cho mạng nơ-ron truyền thẳng trong một lớp Transformer:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

### 3.2 Fine Tuning

Fine tuning được thực hiện cho những nhiệm vụ khác nhau phụ thuộc vào việc thay đổi các inputs hoặc outputs thích hợp. Để huấn luyện mô hình cho từng nhiệm vụ cụ thể, chúng ta có thể thêm một lớp output bổ sung cho mô hình Bert và fine tune mô hình cho tất cả các tham số từ đầu đến cuối. Chỉ cần học một số lượng tham số tối thiểu từ đầu, quy trình huấn luyện sẽ nhanh chóng, giảm chi phí và sử dụng tài nguyên hiệu quả hơn.

Token đầu tiên của mỗi chuỗi là token phân loại đặc biệt ([CLS]). Không giống như vectơ trạng thái ẩn tương ứng với token biểu diễn từ thông thường, trạng thái ẩn tương ứng với token đặc biệt này được chỉ định bởi các tác giả của BERT là đại diện của toàn bộ câu được sử dụng cho các nhiệm vụ phân loại. Như vậy, khi chúng ta cung cấp một câu đầu vào cho mô hình trong quá trình huấn luyện, đầu ra là vectơ trạng thái ẩn độ dài 768 tương ứng với token này. Lớp bổ sung được thêm ở trên bao gồm các nơ ron tuyến tính chưa được huấn luyện có kích thước [hidden\_state, number\_of\_labels], hay [768,3], có nghĩa là đầu ra của BERT kết hợp với lớp phân loại của chúng ta là một vectơ gồm ba số đại diện cho xác suất để làm cơ sở phân loại.



Hình 3.11: Fine tuning mô hình BERT cho sentiment analysis

Nhóm sẽ lan truyền ngược từ toàn bộ mạng, ở bước này chúng ta có thể huấn luyện các tham số mới được giới thiệu ( $768 \times 3$ ) và cũng như học toàn bộ trọng số  $W_q$ ,  $W_k$ ,  $W_v$ . Điều này có nghĩa là mọi thứ trong mạng bắt đầu chuyên sâu cho nhiệm vụ Sentiment Analysis, nghĩa là nó sẽ có gắng điều chỉnh các tham số để thực hiện tốt trên nhiệm vụ này.

Để sử dụng BERT để diễn giải câu "Now i'm just offended", chúng ta có thể sử dụng mô hình BERT đã được huấn luyện trước (pre-trained BERT) hoặc fine-tuning BERT cho các tác vụ cụ thể. Trong trường hợp này, ta giả sử sử dụng pre-trained BERT để phân tích ngữ nghĩa của câu.

**Tokenization:** Đầu tiên, câu "Now i'm just offended" sẽ được phân thành các token riêng biệt. BERT sử dụng một tokenizer để chuyển đổi câu thành dạng mà mô hình có thể hiểu.

- Ví dụ: "Now i'm just offended" có thể được chuyển thành ["now", "i", "", "m", "just", "offended"]

**Input Representation:** BERT yêu cầu đầu vào là một chuỗi token và một chuỗi attention mask (đánh dấu vị trí các từ thực sự có nghĩa trong câu). Đối với câu này, ta có thể có một vector input như sau (đã tokenized và sử dụng attention mask):

- Input: "[CLS]", "now", "i", "", "m", "just", "offended", "[SEP]"
- Attention Mask: [1, 1, 1, 1, 1, 1, 1, 1] (1 cho các từ thực tế và 0 cho các từ padding nếu có)

Trong đó:

- "[CLS]" là token đặc biệt cho mô hình BERT để biểu diễn đầu câu.

- "[SEP]" là token đặc biệt để phân tách giữa hai câu trong các tác vụ dựa trên câu.

### **Chuyển đổi các token thành ID và tạo Mask:**

- Sử dụng bộ từ điển của BERT để chuyển các token thành các số nguyên tương ứng: [101, 2085, 1045, 1005, 1049, 2074, 7506, 102]
- Tạo một mask để chỉ định các từ thật và các từ đệm (padding) khi có: [1, 1, 1, 1, 1, 1, 1]

**Đưa vào BERT và dự đoán:** Sau khi chuẩn bị đầu vào, ta đưa nó vào mô hình BERT. BERT sẽ trả về một biểu diễn vector cho mỗi token đầu vào sau khi đã qua các lớp Transformer Encoder. Lấy embedding của token [CLS], vector đặc trưng tương ứng với token [CLS] thường được sử dụng để biểu diễn toàn bộ câu.

**Đưa embedding qua một lớp phân loại (classification layer):** Lớp phân loại sẽ dựa vào embedding để dự đoán sentiment của câu tweet.

Lớp phân loại có thể là một lớp dense (fully connected) với hàm kích hoạt softmax để phân loại các cảm xúc như positive, negative, neutral.

**Interpretation (Điễn giải):** Để diễn giải câu "Now I'm just offended" sử dụng BERT, chúng ta có thể quan sát các giá trị vector biểu diễn từng token. Các mô hình sử dụng BERT thường sẽ cho phép trích xuất các biểu diễn vector của các token để phân tích ý nghĩa của câu.

- Ví dụ: BERT có thể biểu diễn "now", "i", "", "m", "just", "offended" như các vector trong không gian nghĩa ngữ, và với việc sử dụng self-attention, mô hình có thể nhận ra rằng "offended" là từ mang nghĩa tiêu cực trong câu này.

Đầu ra của việc sử dụng BERT trong mô hình sẽ là dự đoán phân loại Negative vì mô hình đã được fine-tuning cho một tác vụ sentiment analysis

### 3.3 Các chỉ số đánh giá

**Accuracy** là một đơn vị đo tỷ lệ phần trăm của dự đoán đúng trên tổng số dự đoán. Nó được tính bằng công thức sau:

$$\text{accuracy} = \frac{\text{tổng số dự đoán đúng}}{\text{tất cả các dự đoán}}$$

**F1-Score macro** tính toán điểm F1 cho từng lớp rồi lấy giá trị trung bình cộng của chúng. Công thức tính F1-score macro như sau:

$$\text{Precision}_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N \text{Precision}_i$$

$$\text{Recall}_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N \text{Recall}_i$$

$$\text{F1}_{\text{macro}} = \frac{2 \times \text{Precision}_{\text{macro}} \times \text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{Recall}_{\text{macro}}}$$

Trong đó:  $\text{Precision}_i$  và  $\text{Recall}_i$  là precision và recall của lớp thứ  $i$ ,  $N$  là tổng số lớp. F1-score macro thường được sử dụng khi chúng ta muốn đánh giá hiệu suất trung bình qua tất cả các lớp, không quan trọng đến sự chênh lệch về kích thước của các lớp. Nó có thể là một lựa chọn tốt khi các lớp có kích thước khác nhau và chúng ta muốn trọng số mỗi lớp theo cách như nhau trong việc tính toán tổng thể.

**Confusion matrix:** Đây là một ma trận ánh xạ các dự đoán đầu ra với các dự đoán thực sự. Các thành phần chính của ma trận nhằm lẩn bao gồm:

True Positives (TP): Số lượng trường hợp mô hình dự đoán đúng lớp thực tế.

True Negatives (TN): Số lượng trường hợp mô hình dự đoán đúng không thuộc lớp thực tế.

False Positives (FP): Số lượng trường hợp mô hình dự đoán thuộc lớp nhưng thực tế không thuộc lớp đó (lỗi dự đoán giả mạo).

**False Negatives (FN):** Số lượng trường hợp mô hình dự đoán không thuộc lớp nhưng thực tế thuộc lớp đó (lỗi bỏ sót).

**Classification Report:** Classification Report là một công cụ quan trọng trong việc đánh giá hiệu suất của mô hình phân loại, như trong trường hợp fine-tuning mô hình BERT cho các tác vụ phân loại câu. Báo cáo này cung cấp một cái nhìn tổng quan về khả năng phân loại của mô hình trên từng lớp nhãn và các độ đo chính xác, nhạy cảm, F1-score và hỗ trợ (support) của từng lớp

# Phần 4:

## CÀI ĐẶT VÀ THỬ NGHIỆM

### 4.1 Giới thiệu

Trong chương này nhóm sẽ mô tả chi tiết quá trình cài đặt và thử nghiệm phương pháp mà nhóm đã nghiên cứu và phát triển, nhóm sẽ cài đặt và thử nghiệm hai phương pháp khác để so sánh với kết quả cài đặt của nhóm. Phương pháp chính của nhóm được xây dựng dựa trên mô hình BERT (Bidirectional Encoder Representations from Transformers) để giải quyết bài toán. Hai phương pháp nhóm dùng để so sánh là Logistic Regression và XGBoost.

### 4.2 Cài đặt phương pháp của nhóm

#### 4.2.1 Mô hình BERT

Quá trình fine-tuning của BERT cho các nhiệm vụ phân loại câu:

##### Bước 1: Chuẩn bị dữ liệu

Dữ liệu đầu vào: Chuẩn bị dữ liệu huấn luyện và kiểm tra, trong đó mỗi câu đã được gán nhãn vào một trong K nhãn lớp khác nhau (ví dụ: tích cực, tiêu cực, trung tính).

Tokenization và mã hóa: Sử dụng tokenizer của BERT để chuyển đổi các câu thành dạng token và thêm các token đặc biệt như [CLS] ở đầu câu và [SEP] để phân tách câu.

##### Bước 2: Xây dựng mô hình

Load mô hình BERT: Sử dụng mô hình BERT đã được huấn luyện sẵn từ thư viện transformers. Mô hình BERT sẽ có nhiều lớp Transformer encoder, với lớp cuối cùng làm nhiệm vụ trích xuất biểu diễn ngữ cảnh tổng quát của câu từ token [CLS].

Thêm lớp phân loại: Thay đổi lớp đầu ra của BERT để phù hợp với số lớp phân loại K. Thêm một lớp Dense với K đầu ra và sử dụng hàm kích hoạt softmax để tính toán xác suất của từng lớp.

### Bước 3: Huấn luyện mô hình

Định nghĩa hàm loss và optimizer: Định nghĩa hàm loss như categorical cross-entropy và optimizer như Adam optimizer để huấn luyện mô hình.

Mô hình BERT-base-multilingual-cased được train với 4 epochs, kích thước batch là 32, độ dài lớn nhất của chuỗi là 150, learning-rate là 1e-4.

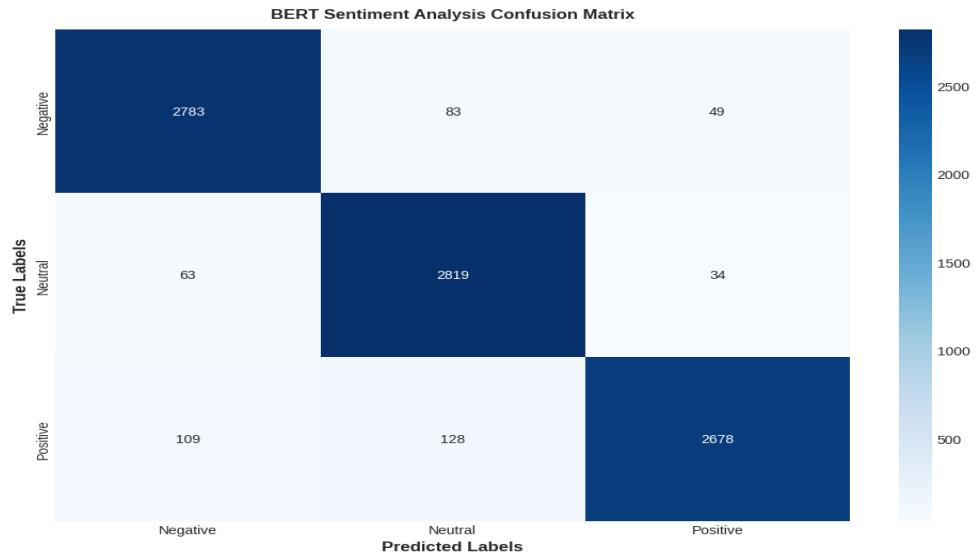
Fine-tuning: Huấn luyện mô hình trên dữ liệu huấn luyện đã chuẩn bị. Trong mỗi epoch, mô hình sẽ nhận đầu vào là các câu đã được mã hóa và dự đoán xác suất của từng lớp.

### Bước 4: Đánh giá và tinh chỉnh

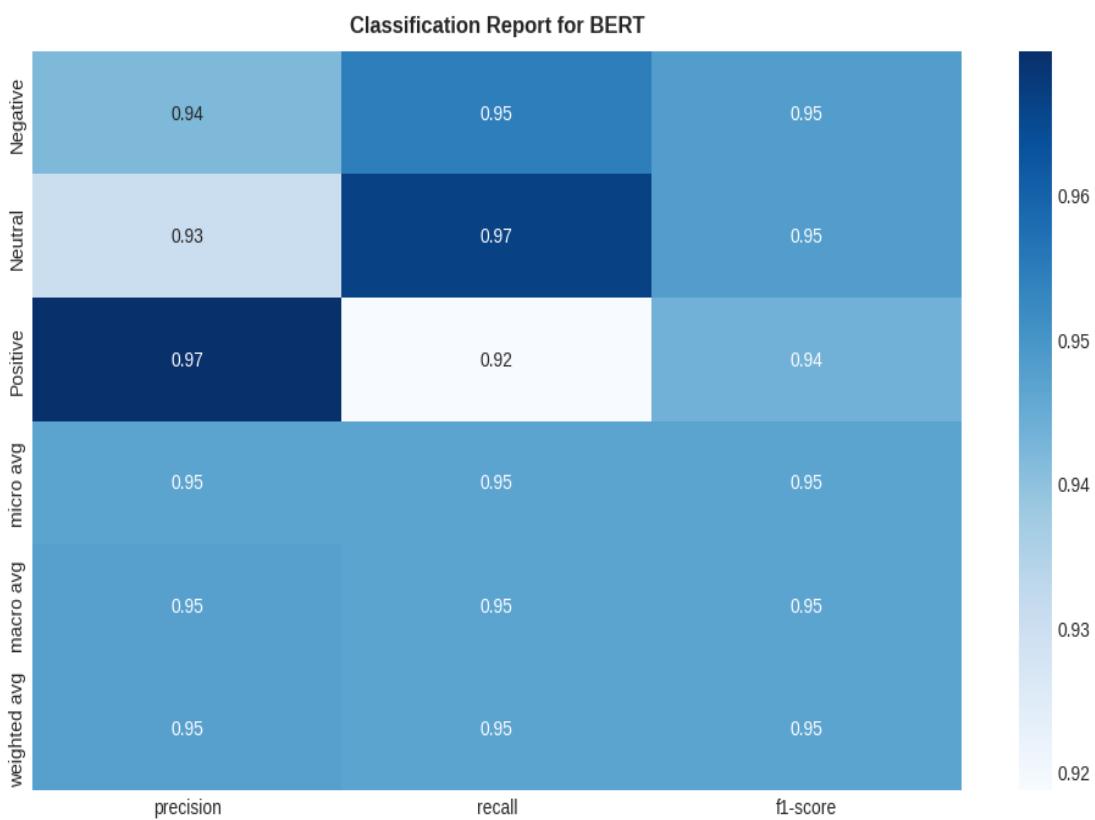
Đánh giá mô hình: Đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra để đo lường độ chính xác và các độ đo đánh giá khác như precision, recall, F1-score, ...

Tinh chỉnh: Có thể điều chỉnh các siêu tham số của mô hình như learning rate, số lượng epochs để cải thiện hiệu suất của mô hình.

## 4.2.2 Kết quả cài đặt



Hình 4.1: Confusion matrix trên tập test



Hình 4.2: Classification report trên tập test

```
print("Accuracy: ", accuracy_score(y_test.argmax(1), y_pred_bert.argmax(1)))  
Accuracy: 0.946718499885662
```

Hình 4.3: Accuracy trên tập test

## 4.3 Cài đặt phương pháp so sánh

### 4.3.1 Mô hình Logistic Regression

#### 4.3.1.1 Phân tích dữ liệu:

Xử lý giá trị thiếu sử dụng `fillna()` để thêm vào các giá trị còn thiếu

Sử dụng LabelEncoder từ thư viện ‘sklearn.preprocessing’ để mã hóa các cột phân loại trong dữ liệu huấn luyện và dữ liệu kiểm tra. Mã hóa nhãn (label encoding) là quá trình chuyển đổi các giá trị phân loại (categorical values) thành các giá trị số (numerical values) để chúng có sử dụng trong mô hình học máy

Chuyển đổi văn abnr sang chữ thường, chuyển đổi tất cả dữ liệu thành chuỗi và loại bỏ kí tự đặc biệt bằng cách sử dụng biểu thức chính quy (regex). Quá trình này được áp dụng cho tất cả tập dữ liệu huấn luyện (‘train\_data’) và tập dữ liệu kiểm (‘val\_data’)

Từ gói dữ liệu ‘punkt’ từ thư viện NLTK tách văn bản thành các token (từ), và đếm số lượng từ duy nhất trong tập dữ liệu huấn luyện

Sử dụng thư viện NLTK để lấy danh sách từ dùng (stopwords) trong tiếng anh.

Stopwords là những từ phổ biến như “the”, “is”, “in”, “and” mà thường không mang nhiều ý nghĩa khi phân tích văn bản, và do đó thường được loại bỏ trong các bài toán xử lý ngôn ngữ tự nhiên

Chia tập dữ liệu huấn luyện (‘train\_data’) thành hai tập con: một tập huấn luyện nhỏ hơn là (‘review\_train’) và một tập kiểm tra (‘review\_test’) việc chia dữ liệu là cần thiết để đánh giá hiệu suất mô hình học máy bằng cách huấn luyện mô hình trên tập huấn luyện và kiểm tra nó trên tập kiểm tra

#### **4.3.1.2 Xây dựng mô hình**

Cài đặt Bag Of Word:

Sử dụng CountVectorizer từ thư viện sklearn.feature\_extraction.text để tạo một biểu diễn Bag of Words (BoW) cho các văn bản trong dữ liệu

- CountVectorizer: Là một công cụ trong sklearn dùng để chuyển đổi một bộ văn bản thành ma trận các token đếm (count matrix), nó tính toán số lần xuất hiện của mỗi từ trong tập dữ liệu.
- Tham số tokenizer: ‘tokenizer=word\_tokenize’ word\_tokenize là một hàm từ thư viện nltk.tokenize dùng để phân tích câu thành các từ (token). Nó sẽ tách câu thành các thành phần cơ bản là các từ.
- Tham số stop\_words: ‘stop\_words=stop\_words’ stop\_words là danh sách các từ dừng (stopwords), các từ không mang ý nghĩa quan trọng trong việc phân loại văn bản và thường bị loại bỏ trong quá trình xử lý dữ liệu. Trong trường hợp này, stop\_words được truyền vào từ biến stop\_words, có thể là danh sách các stopwords tiếng Anh từ thư viện nltk.corpus.
- Tham số ngram\_range: ‘ngram\_range=(1, 1)’ ngram\_range xác định phạm vi của các n-gram mà chúng ta muốn xây dựng. Trong trường hợp này, (1, 1) chỉ ra rằng chúng ta chỉ quan tâm đến các từ đơn (unigram), tức là các từ riêng lẻ, không xây dựng các n-gram có kích thước lớn hơn 1.

Chuẩn bị dữ liệu cho mô hình:

- bow\_counts là một đối tượng của lớp CountVectorizer từ thư viện sklearn.feature\_extraction.text. Đối tượng này đã được khởi tạo với các tham số như tokenizer, stop\_words và ngram\_range như đã mô tả trong câu hỏi trước đó.
- fit\_transform(reviews\_train.lower): Phương thức fit\_transform được gọi trên bow\_counts để biến đổi dữ liệu huấn luyện (reviews\_train.lower). Trước tiên, nó sẽ học từ vựng (vocabulary) từ dữ liệu huấn luyện và sau đó biến đổi các văn

bản trong reviews\_train.lower thành ma trận đếm (count matrix) theo biểu diễn Bag of Words.

- transform(reviews\_test.lower): Phương thức transform được gọi sau khi đã học từ vựng từ dữ liệu huấn luyện ( thông qua fit\_transform). Nó sử dụng từ vựng đã học để biến đổi dữ liệu kiểm tra (reviews\_test.lower) thành ma trận đếm tương tự như dữ liệu huấn luyện, bao gồm các từ xuất hiện trong từ vựng đã học.

Cài đặt Logistic Regression từ đầu: sử dụng phương pháp One-vs-Rest và thuật toán Gradient Descent, mô hình có thể xử lý các bài toán phân loại đa lớp (multi label)

Sử dụng class Logistic Regression được định nghĩa trước đó với learning\_rate = 0.1 và num\_iterations = 1500

Dữ liệu là ma trận đặc trưng dạng Bag of Words và nhãn tương ứng

Dữ đoán kết quả của dữ liệu bằng cách dùng phương thức ‘predict’ của mô hình Logistic Regression

Độ chính xác sử dụng phương thức accuacy\_score với dự đoán kết quả trên dữ liệu so sánh với dự đoán với nhãn thực tế

#### 4.3.1.3 Đánh giá và tinh chỉnh

Sử dụng GridSearchCV từ thư viện sklearn.model\_selection:

- params là các tham số cần tinh chỉnh cho mô hình Logistic Regression bao gồm learning\_rate và num\_interations
- GridSearch được dùng để tìm kiếm bộ tham số tối ưu bằng cách params và thực hiện cross\_validation (cv=5)
- best\_params là bộ tham số tối ưu nhất được tìm thấy
- sau khi tìm được bộ tham số tối ưu nhất ta sẽ huấn luyện và đánh giá độ chính xác

Kết quả cài đặt:

```
print("Accuracy: ", accuracy_score(y_test_bow, test_pred) * 100)
```

```
→ Accuracy: 83.25634330856263
```

*Hình 4.4: Accuracy trên tập test*

## 4.3.2 Mô hình XGBoost

### 4.3.2.1 Phân tích dữ liệu

Text splitting (Tách từ): Dùng word\_tokenize từ thư viện nltk.tokenize để tách các từ trong nội dung của các câu.

Biến tokens\_text là một danh sách các danh sách từ (list of lists), mỗi danh sách con chứa các từ của một câu.

Đếm số lượng token duy nhất: Sử dụng set(tokens\_counter) để đếm số lượng từ duy nhất trong toàn bộ văn bản. Điều này giúp ta có cái nhìn tổng quan về sự đa dạng của từ vựng trong dữ liệu.

Lựa chọn stopwords tiếng Anh: Sử dụng stopwords của tiếng Anh từ thư viện nltk.corpus.stopwords để loại bỏ các từ không mang tính nghĩa như "a", "an", "the",...

### 4.3.2.2 Biểu diễn dữ liệu

Khởi tạo Bag of Words (BOW):

- Sử dụng CountVectorizer từ sklearn.feature\_extraction.text để biểu diễn các văn bản thành ma trận BOW.
- Thiết lập tokenizer=word\_tokenize để sử dụng tách từ đã chuẩn bị.
- Thiết lập stop\_words=stop\_words để loại bỏ stopwords tiếng Anh.
- ngram\_range=(1, 1) chỉ xem xét các từ đơn.

Chia dữ liệu thành tập huấn luyện và kiểm tra:

- Sử dụng train\_test\_split từ sklearn.model\_selection để chia dữ liệu thành ba phần: huấn luyện (70%), validation (15%) và test (15%).
- Thiết lập random\_state=42 để đảm bảo các lần chia dữ liệu là nhất quán.

Mã hóa dữ liệu:

- Sử dụng fit\_transform để huấn luyện và biến đổi dữ liệu huấn luyện.
- Sử dụng transform để biến đổi dữ liệu validation và test theo mô hình đã huấn luyện.

#### **4.3.2.3 Huấn luyện và dự đoán mô hình**

Huấn luyện mô hình XGBoost:

- Khởi tạo XGBClassifier với các siêu tham số như n\_estimators, colsample\_bytree, subsample.
- Sử dụng fit để huấn luyện mô hình trên dữ liệu huấn luyện đã được biểu diễn bằng BOW.

Dự đoán và đánh giá mô hình:

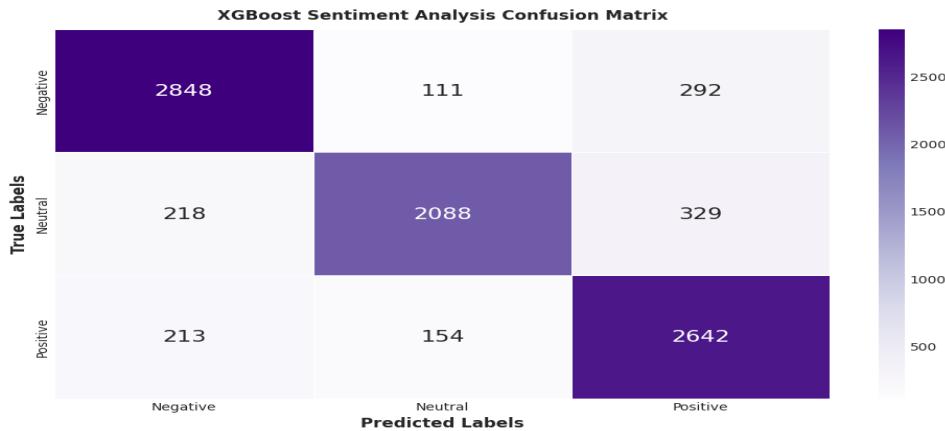
- Sử dụng predict để dự đoán nhãn của dữ liệu validation và test.
- Đánh giá hiệu suất bằng độ chính xác (accuracy\_score).

#### **4.3.2.4 Đánh giá và tinh chỉnh**

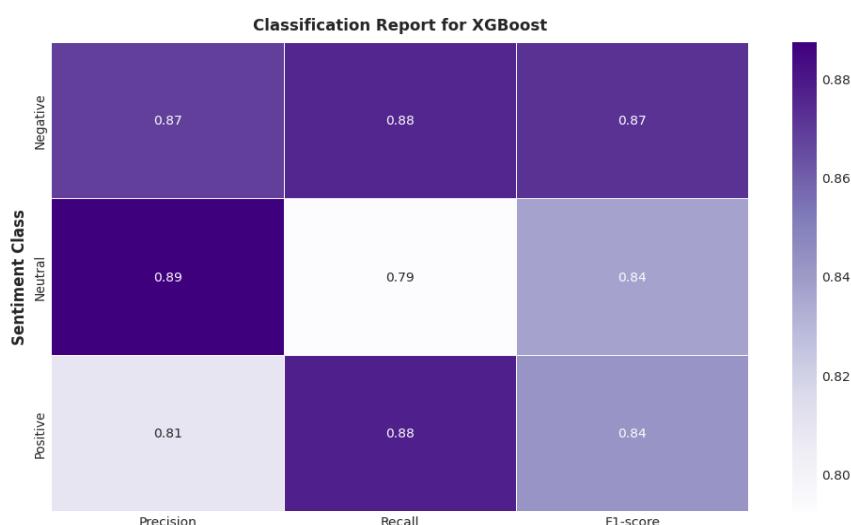
Để cải thiện hiệu suất của mô hình, có thể thực hiện:

- Tinh chỉnh siêu tham số (Hyperparameter tuning): Điều chỉnh các tham số của mô hình như n\_estimators, max\_depth, learning\_rate, để tối ưu hóa hiệu suất.
- Cross-validation: Sử dụng kỹ thuật cross-validation để đánh giá hiệu suất trung bình của mô hình và giảm thiểu overfitting.
- Grid Search hoặc Random Search: Thủ nghiệm các giá trị khác nhau của các tham số để tìm ra bộ tham số tối ưu.

Kết quả cài đặt:



Hình 4.5: Confusion matrix trên tập test



Hình 4.6: Classification report trên tập test

```
print("Accuracy: ", accuracy_score(y_test_bow_num, test_pred_2) * 100)

Accuracy: 85.19392917369308
```

Hình 4.7: Accuracy trên tập test

#### 4.4 Phân tích một vài mẫu sai

Đối với mô hình BERT, mô hình có thể học được ngữ cảnh nhưng khi gặp những mẫu câu mang hàm ý không rõ ràng, không thể hiện qua những dấu hiệu mà mô hình đã được học thì mô hình vẫn chưa nhận biết chính xác.

Tweet	Nhận dự đoán	Nhận thực tế	Nhận xét
FIX IT JESUS ! Please FIX IT ! What In the world is going on here. @PlayStation @AskPlayStation @Playstationsup @Treyarch @CallofDuty negative 345 silver wolf error code pic.twitter.com/ziRyhrf59Q	Positive	Negative	Cụm từ " FIX IT JESUS! Please FIX IT biểu thị sự thất vọng hoặc cấp bách, gợi ý một vấn đề cần giải quyết ngay lập tức. Việc sử dụng "Chuyện gì đang xảy ra ở đây" và việc đề cập đến mã lỗi càng nhấn mạnh thêm sự không hài lòng của người dùng. Bất chấp những tín hiệu tiêu cực rõ ràng này, một mô hình có thể phân loại sai nó thành tích cực do dấu chấm than và có thể hiểu sai "FIX IT" là hành động chủ động hoặc tích cực.
The new @CallofDuty for ps5 is 🔥🔥🔥🔥	Positive	Negative	Cụm từ " The new @CallofDuty for ps5 is 🔥🔥🔥🔥" rõ ràng là mang tính tích cực vì nó sử dụng biểu tượng cảm xúc lửa để thể hiện sự phấn khích và tán thành. Nhưng ở đây có lẽ đã sai ở việc gán nhãn dữ liệu là Negative
I'm addicted to call of duty mobile 😊	Positive	Negative	Trong câu tweet trên từ addicted mang nghĩa tiêu cực là nghiện và cụ thể là nghiện trò chơi call of duty nhưng cuối câu lại có icon vui vẻ tích cực nên mô hình đã dự đoán sai
Seems like #Playstation has the marketing deal for #CallOfDutyBlackOpsColdWar	Negative	Neutral	Câu tweet chỉ đơn giản là sự thỏa thuận của PlayStation cho Call Of Duty tuy nhiên mô hình

			lại gán nhãn cho câu là Negative.
the ps5 got fortnite? alr rip xbox	Negative	Positive	Câu "PS5 đã có Fortnite? Tốt rồi, thôi thì Xbox." có thể hiểu là người nói đang đưa ra sự so sánh giữa hai hệ máy chơi game PS5 và Xbox, và việc Fortnite xuất hiện trên PS5 có thể làm cho người chơi lựa chọn PS5 hơn. Do đó có thể có sự hiểu nhầm cảm xúc là sự tiếc nuối dành cho xbox

## Phần 5: KẾT LUẬN

Nghiên cứu này tập trung vào việc áp dụng mô hình BERT để phân tích và phân loại cảm xúc từ dữ liệu văn bản, một bài toán quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và khoa học dữ liệu. Chúng em đã tiến hành các bước chuẩn bị dữ liệu chi tiết như tiền xử lý, loại bỏ dữ liệu không hợp lệ và các mẫu trùng lặp, sau đó tập trung vào các mẫu dữ liệu có nhãn cảm xúc mang tính ý nghĩa.

Thử nghiệm và đánh giá mô hình BERT trên các tập dữ liệu đã cân bằng cho thấy kết quả vô cùng khả quan. Mô hình đã đạt được hiệu suất cao với các chỉ số precision, recall và F1-score, những chỉ số này là độ đo quan trọng để đánh giá khả năng phân loại chính xác của mô hình. Điều này cho thấy mô hình BERT không chỉ đáp ứng được yêu cầu phân loại cảm xúc mà còn cải thiện hiệu quả công việc phân tích dữ liệu văn bản so với các phương pháp truyền thống.

Ngoài ra, việc áp dụng thành công mô hình BERT trong phân tích cảm xúc từ dữ liệu văn bản còn mở ra nhiều triển vọng trong ứng dụng thực tế. Ví dụ, nó có thể được áp dụng để tổng hợp và đánh giá phản hồi từ khách hàng, phân tích tình hình thị trường, và cải thiện trải nghiệm người dùng trong các dịch vụ trực tuyến. Điều này đem lại giá trị lớn cho các tổ chức và doanh nghiệp trong việc nắm bắt ý kiến công khai và dự báo xu hướng thị trường một cách hiệu quả hơn.

Tóm lại, nghiên cứu này đã minh chứng sự hiệu quả và tiềm năng của mô hình BERT trong việc phân tích và phân loại cảm xúc từ dữ liệu văn bản, đồng thời khẳng định vai trò quan trọng của học máy và xử lý ngôn ngữ tự nhiên trong thời đại số ngày nay.

# TÀI LIỆU THAM KHẢO

[1] Phạm Hữu Quang, Hiểu hơn về BERT: Bước nhảy lớn của Google

<https://viblo.asia/p/hieu-hon-ve-bert-buoc-nhay-lon-cua-google-eW65GANOZDO>

[2] Kaggle

<https://www.kaggle.com/code/ludovicocuoghi/twitter-sentiment-analysis-with-bert-vs-roberta/notebook>

[3] Pham Dinh Khanh, Bert Model

<https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html>

[4] Pham Dinh Khanh, Attention is all you need

<https://phamdinhkhanh.github.io/2019/06/18/AttentionLayer.html>