**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN MÔN HỌC**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**Đề tài:**

**HATE SPEECH DETECTION ON VIETNAMESE SOCIAL MEDIA TEXTS**

**Sinh viên thực hiện: Trần Tuyết Nhi - 21520383**

**Nguyễn Trọng Ân - 21520548**

Tp. Hồ Chí Minh, tháng 1 năm 2024

Nội dung đồ án

[**1. Bộ ngữ liệu** 1](#_Toc157772988)

[**2. Phương pháp** 10](#_Toc157772989)

[2.1. Text-CNN 10](#_Toc157772990)

[2.2. BERT 20](#_Toc157772991)

[2.3. Các metric đánh giá mô hình 28](#_Toc157772992)

[**3. Cài đặt** 29](#_Toc157772993)

[3.1. Tiền xử lý ngữ liệu 29](#_Toc157772994)

[3.2. Trích xuất đặc trưng 31](#_Toc157772995)

[3.3. Cài đặt mô hình 32](#_Toc157772996)

[**4. Kết quả đánh giá** 33](#_Toc157772997)

[4.1. Kết quả đạt được 33](#_Toc157772998)

[4.2. Phân tích các trường hợp sai 33](#_Toc157772999)

# **1. Bộ ngữ liệu**

Bộ ngữ liệu nhóm sử dụng cho đồ án là bộ ngữ liệu ViHSD. Đây là bộ ngữ liệu bằng tiếng Việt, được thu thập từ các bình luận trên các mạng xã hội như FaceBook và Youtube. Các nhãn trong bộ ngữ liệu được chú thích thủ công nhằm mục đích phục vụ cho các nghiên cứu phát hiện lời nói chán ghét (hate speech) một cách tự động trên các mạng xã hội.

Bộ ngữ liệu bao gồm 33400 bình luận đã được gán nhãn với tổng số lượng từ vựng là 21239 từ. Các bình luận trong bộ ngữ liệu được chia làm 3 nhóm tương ứng với 3 nhãn sau: HATE (2), OFFENSIVE (1) và CLEAN (0). Trong bộ ngữ liệu ViHSD, có hai nhãn thể hiện bình luận chứa lời nói chán ghét và một nhãn chỉ đến bình luận bình thường. Các ý nghĩa chi tiết về các nhãn được mô tả trong bảng dưới đây:

|  |  |
| --- | --- |
| **Label** | **Description** |
| CLEAN | Các bình luận không có bất kỳ hành vi quấy rối nào |
| OFFENSIVE | Các bình luận chứa nội dung quấy rối, kể cả những từ thô tục, nhưng không tấn công một đối tượng cụ thể. |
| HATE | Các ý kiến có nội dung quấy rối và thô tục, trực tiếp nhắm vào một cá nhân hoặc một nhóm người dựa trên đặc điểm cá nhân, tôn giáo và quốc tịch.  Một số trường hợp xảy ra với nhãn HATE:   * Trường hợp 1: Các ý kiến chứa những từ thô tục và tấn công một đối tượng cụ thể như một cá nhân, cộng đồng, quốc gia hoặc tôn giáo. Trường hợp này dễ nhận biết là lời nói chán ghét. * Trường hợp 2: Các ý kiến có tính chất phân biệt chủng tộc, quấy rối và có ý nghĩa chứa đựng sự căm ghét, tuy nhiên, không chứa các từ ngữ rõ ràng. * Trường hợp 3: Các ý kiến có tính chất phân biệt chủng tộc, quấy rối và có ý nghĩa chứa đựng sự căm ghét, nhưng được thể hiện dưới hình thức ẩn dụ. Để nhận biết ý kiến này, người dùng cần có kiến thức đặc biệt về xã hội. |

*Bảng 1: Bảng ý nghĩa các nhãn trong bộ ngữ liệu ViHSD*

Một số mẫu bình luận và nhãn tương ứng của nó trong tập ngữ liệu như sau đây:

|  |  |  |
| --- | --- | --- |
| **Label** | **Comment** | **Explain** |
| CLEAN | Mọi người ơi, cho mik hỏi mik theo dõi cô ấy mà mik hk pít cô ấy là con gái thiệt hả m.n | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Trước khi cắm lên đầu em 4 cái sừng nó cũng bảo y như thế :(((( | Bình luận này xét về nghĩa, nó đang mô tả một tình huống xấu nhưng bình luận không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| 1 ng thì giết 1 mạng . 1 ng thì feed tụt lol | Bình luận này có chứa từ thô tục ‘lol’ nhưng viết theo kiểu tiếng lóng. Theo định nghĩa để phân biệt các nhãn, nội dung đầu tiên cần xét tới là có chứa nội dung quấy rối hay không thì bình luận này không chứa nội dung quấy rối nên được gán nhãn sạch. |
| Được anh ưi, lâu rồi không nghe phúc du rap 🤣 | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Cực kì thuyế...hác xì....á chết mẹ.. | Bình luận này có chứa từ ngữ thô tục “Chết mẹ” nhưng nội dung không quấy rối, tấn công bất kì ai nên được gán nhãn sạch |
| Nó giống vãi lol a ưi! | Bình luận này có chứa từ ngữ thô tục “vãi lol” nhưng chỉ mang ý nghĩa nhấn mạnh, nội dung không quấy rối, tấn công bất kì ai nên được gán nhãn sạch |
| chất vlon a ưii | Bình luận này có chứa từ ngữ thô tục “vlon” nhưng chỉ mang ý nghĩa nhấn mạnh, nội dung không quấy rối, tấn công bất kì ai nên được gán nhãn sạch |
| Đinh Phước sống nhanh chậm t chửi hết👌 | Bình luận này mang ý nghĩa tiêu cực nhưng bình luận không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai nên được gán nhãn sạch |
| Thuyết phục vl | Bình luận này có chứa từ ngữ thô tục “vl” nhưng chỉ mang ý nghĩa nhấn mạnh, nội dung không quấy rối, tấn công bất kì ai nên được gán nhãn sạch |
| hợp lý a ưi 🤣🤣 | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Xin phép cả nhà nhaaaa!!! | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Tự hào quá VN ơi 😂 | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| MV khá chất lượng | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Tào Hưng đm khô máu :))) | Bình luận này có chứa từ ngữ thô tục “đm” nhưng chỉ mang ý nghĩa biểu cảm, nội dung không quấy rối, tấn công bất kì ai nên được gán nhãn sạch |
| Vc | Bình luận này có chứa từ ngữ thô tục “Vc” nhưng chỉ mang ý nghĩa biểu cảm, nội dung không quấy rối, tấn công bất kì ai nên được gán nhãn sạch |
| T đã đi ngủ từ lúc 10r nhưng đ thể nào ngủ nổi 🙂 | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Phan Quỳnh Anh bóp d' là gì m | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| **Bé bú trên để bố bú dưới, =))** | **Bình luận này có từ ngữ thô tục ‘bú’ và cũng mang ý nghĩa không phù hợp, nên gán nhãn sạch là chưa hợp lý** |
| Oiiiii nàng công chúa | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
|  | **Đoàn Quang Vinh dm quảng cáo** | **Bình luận này chứa từ ngữ thô tục “dm” và cũng mang ý nghĩa tiêu cực nên gán nhãn sạch là chưa hợp lý** |
| OFFENSIVE | Đồ khùng | Bình luận này chứa từ xúc phạm "khùng". Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| Bảo dan tri thap thì tự ái :) | Bình luận này chứa từ xúc phạm "dan tri thap". Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| Bố lạy mày đừng "hiếp dâm" thị giác của bố nữa ad à 😂😂😂 | Bình luận này chứa từ thô tục "hiếp dâm". Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| Lưu Bảo Toàn đọc hại não vcl | Bình luận này chứa từ ngữ thô tục “vcl”. Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| lolzzzz | Bình luận này chứa từ ngữ thô tục “lolzzzz” mang ý nghĩa biểu thị cảm xúc tiêu cực. Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| Ncc | Bình luận này chứa từ ngữ thô tục “Ncc” mang ý nghĩa biểu thị cảm xúc tiêu cực. Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| đọc bài báo xong buồn ĩa đéo chịu đc | Bình luận này chứa từ ngữ thô tục “ĩa” mang ý nghĩa biểu thị thái độ tiêu cực. Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| mặc cái quần giống thằng biến thái | Bình luận này mang ý nghĩa miệt thị, tiêu cực nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| Im mẹ i | Bình luận này mang ý nghĩa miệt thị, tiêu cực nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| Về vn ăn mau lol à. | Bình luận này chứa từ ngữ thô tục “mau lol” và mang ý nghĩa quấy rối nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| **Phúc Hưng clm :)))** | **Bình luận này chứa từ ngữ thô tục “clm” nhưng về ý nghĩa lại thể hiện cảm xúc vui vẻ, tích cực nên gán nhãn offensive là chưa hợp lý** |
| đéo ai thèm làm ,vì họ biết khả năng của họ đến đâu | Bình luận này chứa từ ngữ thô tục “đéo” và mang ý nghĩa quấy rối, tiêu cực nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| Toàn ngu muội. Đi ra đó mà đánh đi. | Bình luận này mang ý nghĩa miệt thị, tiêu cực nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| Đéo ai bắt xem ??=}} | Bình luận này chứa từ ngữ thô tục “đéo” và mang ý nghĩa quấy rối, tiêu cực nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| Lê Thị Trang đắt vcd | Bình luận này chứa từ ngữ thô tục “vcl” nhưng không nhắm vào một người hoặc một nhóm cụ thể |
| bắn bỏ | Bình luận này mang ý nghĩa quấy rối, tiêu cực nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| **tự hào vì có nhiều admin fanpage ngu nhất thế giới nữa :))** | **Bình luận này mang ý nghĩa quấy rối, tiêu cực và chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể “admin fanpage”** |
| Kkkk...iu nhau là phải có đụ.. | Bình luận này có từ ngữ thô tục “đụ” và mang ý nghĩa quấy rối nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể. |
| **Mu này tởm thật** | **Bình luận này mang ý nghĩa quấy rối hướng đến một đối tượng cụ thể nên gán nhãn offensive là chưa hợp lý** |
| Người ta gọi là đú mà. | Bình luận này mang ý nghĩa quấy rối, miệt thị nhưng không chứa từ ngữ nhắm vào một người hoặc một nhóm cụ thể |
| **Tạ Thị Ngọc Duyên mày cũng v chứ khác đéo đâu xớ** | **Bình luận này có từ ngữ thô tục “đéo” và mang ý nghĩa quấy rối, tiêu cực, đồng thời nhắm tới một đối tượng cụ thể nên gán nhãn 1 là chưa lý** |
| HATE | Dành cho lũ quan ngại | Bình luận này chứa một cụm từ “lũ quan ngại” đề cập đến một nhóm người với ý nghĩa tiêu cực |
| Dm Có a mới không ổn. Mày rình mày chịch riết ổn cái lol | Bình luận này chứa nhiều từ thô tục “Dm”, “chịch”, “cái lol”. Ngoài ra nó còn chứa đại từ nhân xưng “mày” có ý nghĩa nhắm đến một người cụ thể |
| Ở đấy ngột ngạt quá thì đưa nó qua vũ hán cho nó thoáng mát | Bình luận này chứa cụm từ vũ hán đề cập đến một địa điểm cụ thể, mang ý nghĩa phân biệt chủng tộc. Tuy nhiên, bình luận này không chứa bất kỳ từ ngữ xấu nào. |
| Xem cái này chỉ muốn ỉa vào mặt mấy thằng cho xây tượng đài... | Bình luận này chứa từ ngữ thô tục ‘ỉa’ đồng thời mang nội dung tiêu cực nhắm đến một đối tượng cụ thể “mấy thằng” |
| Tủnn Tủnn trẩu | Bình luận này chứa nội dung quấy rối, tiêu cực và nhắm đến một người cụ thể “Tủnn Tủnn” |
| **Tui nghĩ là xe Merc sẽ có camera hành trình phía sau lưu lại xe nào quay được cảnh này. Tội cắt ghép clip này là bôi nhọ lãnh đạo, vu khống, xuyên tạc chứ ít gì.** | **Bình luận này không chứa từ ngữ thô tục và không mang nội dung quấy rối nhắm đến một người hay nhóm nào nên gán nhãn hate là chưa hợp lý** |
| xuất cai L, đm quần áo Vn toàn nhập siêu từ TQ hết 70% chứ xuất xuất clm chúng mày | Bình luận này chứa nhiều từ ngữ thô tục “cái L”, “đm”, “clm” và có từ “chúng mày” nhắm đến một nhóm đối tượng cụ thể |
| Hoàng Ngọc Hòa sũa cc | Bình luận này chứa từ ngữ thô tục “sũa cc” và có nội dung quấy rối nhắm dến một người cụ thể “Hoàng Ngọc Hoà” |
| Nguyễn Việt Cường ơ thằng này ra tù rồi à | Bình luận này không chứa từ thô tục nào nhưng có nội dung chế giễu, quấy rối nhắm đến một người cụ thể “thằng này” nên được gán nhãn hate |
| Trần dần bệnh tâm thần chuyện có nói ko chuyện ko nói có làm cho cả cộng đồng người việt ở Mỹ quá bức xúc truyền thông pho bolsa tv giống Trần dần bịa đặt chuyện có nói ko chuyện ko nói có | Bình luận này không chứa từ thô tục nào nhưng có nội dung chế giễu, quấy rối nhắm đến một người cụ thể “Trần dần” và cả nhóm người “người Việt ở Mỹ” nên được gán nhãn hate |
| Bài này như lồn | Bình luận này chứa từ ngữ thô tục “lồn” và có nội dung quấy rối nhắm đến một đối tượng cụ thể “bài này” |
| Những chú dâm loạn viên đâu tui muốn thấy cái mõm chó của mấy bạn | Bình luận này chứa từ ngữ thô tục “dâm loạn”, “mõm chó” và có nội dung quấy rối nhắm đến một nhóm đối tượng cụ thể “mấy bạn” |
| Nhìn nhà tiên tri vủ trụ trần dần giống con chó quá trời luôn ta ơi . | Bình luận này chứa từ ngữ thô tục “chó” và có nội dung quấy rối nhắm đến một người cụ thể “trần dần” |
| **loại đàn bà độc ác** | **Bình luận này chứa từ ngữ tiêu cực “độc ác” và có nội dung quấy rối, miệt thị nhưng không chứa từ ngữ nhắm đến một người hay nhóm cụ thể nên gán nhãn hate là chưa hợp lý** |
| Một số lớn người VN phải xếp vào một trong những dân tộc tàn độc nhất thế giới ,mục tiêu là chủ nghĩa cá nhân là trên hết. | Bình luận này chứa từ ngữ tiêu cực “tàn độc” và có nội dung quấy rối nhắm đến một nhóm đối tượng cụ thể “người VN” |
| Lũ chó chết. Đkm loạn mẹ nó rồi. Xh như lồn | Bình luận này chứa nhiều từ ngữ tiêu cực “chó chết”, “Đkm”, “mẹ”, “lồn” và có nội dung quấy rối, miệt thị và chứa từ ngữ nhắm đến một người hay nhóm cụ thể nên gán nhãn hate là hợp lý |
| Ca sĩ k-icm đéo gì 🙂 ca sĩ câm | Bình luận này chứa từ ngữ tiêu cực “đéo”, “câm” và có nội dung quấy rối nhắm đến một người cụ thể “Ca sĩ k-icm” |
| Vân Thuỳ xàm vl :)) | Bình luận này chứa từ ngữ thô tục “vl” và có nội dung quấy rối nhắm đến một người cụ thể “Vân Thuỷ” |
| Con này là bê đê bong lộ xấu gớm... xấu xúc phạm ng nhìn... | Bình luận này mang nội dung quấy rối, miệt thị, phân biệt và nhắm đến một người cụ thể “con này” nên được gán nhãn hate |
| Chúng bắt đầu cho báo chí đổ tại người dân dọn nhà đón Tết gây ra ô nhiễm kìa | Bình luận này mang ý nghĩa tiêu cực và nhắm tới một đối tượng cụ thể |

*Bảng 2: Ví dụ một số trường hợp cụ thể của các nhãn*

Thực tế, nhiều ý kiến trong bộ dữ liệu được viết dưới hình thức không chính thức. Các ý kiến thường chứa viết tắt như M.n (mọi người), mik (mình) trong và Dm, cùng với những ngôn ngữ lóng như “chịch”, “cái lol”. Ngoài ra, các ý kiến thường mang ý nghĩa ẩn dụ thay vì ý nghĩa rõ ràng. Ví dụ, từ "lũ quan ngại" thường được sử dụng bởi nhiều người dùng Facebook Việt Nam trên nền tảng truyền thông xã hội để đề cập đến một nhóm người luôn suy nghĩ tiêu cực và đăng nội dung tiêu cực.

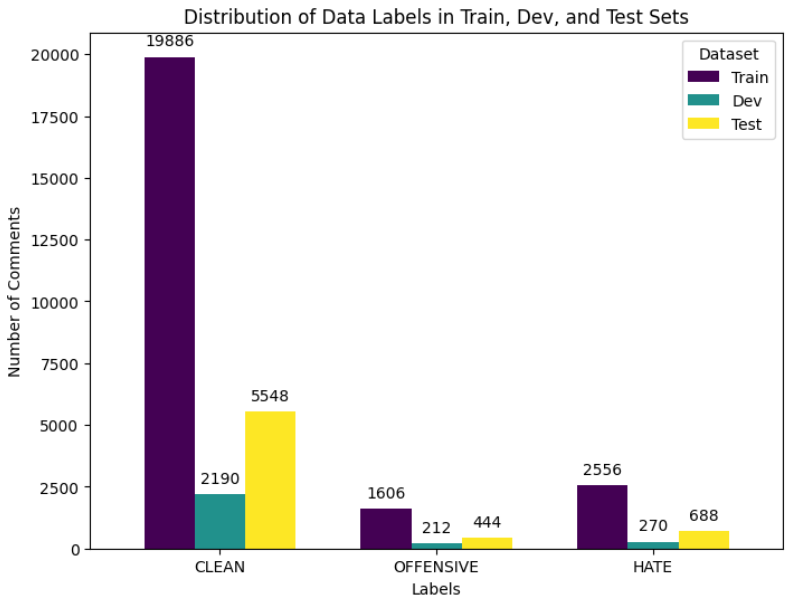
Mỗi bình luận trong tập ngữ liệu bao gồm 1 hoặc 2 câu ngắn, có thể chứa các biểu tượng cảm xúc. Tuy nhiên các biểu tượng cảm xúc này không mang ý nghĩa quá nhiều cho việc xác định nhãn của bình luận. Ví dụ một số mẫu câu có biểu tượng cảm xúc như sau:

|  |  |  |
| --- | --- | --- |
| **Comments** | **Label** | **Description** |
| Cũng được đấy chứ a Độ ộ ộ ộ ộ ộ....😂😂😂 | CLEAN | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai |
| Mang tiếng fan cứng đăng nhưng đéo đc duyệt 🙂 | OFFENSIVE | Bình luận này chứa từ thô tục "đéo". Tuy nhiên, nó không chứa bất kỳ từ ngữ nào nhằm vào một người hoặc một nhóm. |
| Ca sĩ k-icm đéo gì 🙂 ca sĩ câm | HATE | Bình luận này chứa cụm từ thô tục như “đéo” và cụm từ “ca sĩ câm” mang ý nghĩa chê bai, tấn công một cá nhân. |
| 🤣🤣🤣🤣🤣 | CLEAN | Bình luận này hoàn toàn sạch, không chứa nội dung quấy rối hoặc ngôn ngữ thô tục, và không tấn công bất kỳ ai. |

*Bảng 3: Một số mẫu câu có biểu tượng cảm xúc*

Từ bảng 3 ta thấy các nhãn được gán cho bình luận liên quan chủ yếu đến các từ hoặc các cụm từ có trong bình luận. Các biểu tượng cảm xúc không có giá trị quá lớn trong việc phân loại nhãn.

Bộ ngữ liệu bộ dữ liệu thành ba phần: tập huấn luyện (train), phát triển (dev) và kiểm thử (test), tương ứng với tỷ lệ 7-1-2. Phân phối ngữ liệu trên ba nhãn của các tập dữ liệu đó như sau:



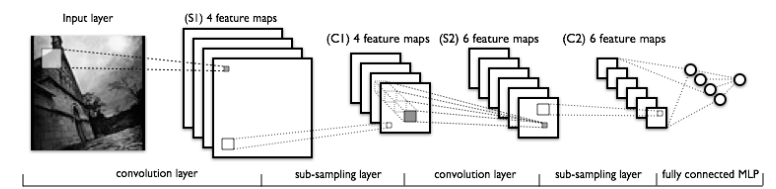
*Hình 1: Phân phối ngữ liệu trên ba nhãn của các tập dữ liệu train, dev và test*

Theo biểu đồ, ta thấy phân phối nhãn dữ liệu trên các tập huấn luyện, phát triển và kiểm thử là giống nhau, và dữ liệu nghiêng về nhãn CLEAN.

# **2. Phương pháp**

## **2.1. Text-CNN**

Mạng thần kinh chuyển đổi ConvNet ban đầu được phát triển trong cộng đồng xử lý hình ảnh mạng thần kinh nơi chúng đạt được kết quả đột phá trong việc nhận dạng một đối tượng từ danh mục được xác định trước (ví dụ: mèo, xe đạp, v.v.). Mạng thần kinh chuyển đổi thường bao gồm hai hoạt động, có thể được coi là trình trích xuất tính năng: tích chập và gộp chung. Đầu ra của chuỗi hoạt động này sau đó thường được kết nối với một lớp được kết nối đầy đủ, về nguyên tắc giống như mạng nơ ron perceptron nhiều lớp truyền thống (MLP).

****

*Hình 2: Kiến trúc mạng thần kinh chuyển đổi được áp dụng để phân loại hình ảnh*

Mạng nơron tích chập thường được ứng dụng chủ yếu trong xử lý hình ảnh nhưng cũng có thể được sử dụng hiệu quả trong các tác vụ xử lý ngôn ngữ tự nhiên (NLP) như text classification. Trong trường hợp tác vụ NLP, chúng ta có mảng 1 chiều biểu thị văn bản. Ở đây, kiến ​​trúc của ConvNet được thay đổi thành các convolution 1D.

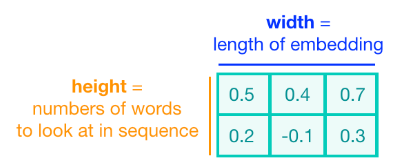
**Phép biến đổi 1D trên văn bản**

Cho một chuỗi các từ , trong đó mỗi từ được liên kết với một vectơ nhúng có kích thước .

* Hạt nhân tích chập (convolutional kernel):

Các lớp tích chập được thiết kế để tìm các mẫu không gian trong hình ảnh bằng cách trượt một cửa sổ hạt nhân nhỏ trên hình ảnh. Các cửa sổ này thường nhỏ, có thể có kích thước 3x3 pixel và mỗi ô nhân có trọng số liên quan. Khi hạt nhân trượt trên một hình ảnh, theo từng pixel, trọng số hạt nhân được nhân với giá trị pixel trong hình ảnh bên dưới, sau đó tất cả các giá trị nhân được cộng lại để có được giá trị pixel đầu ra được lọc.

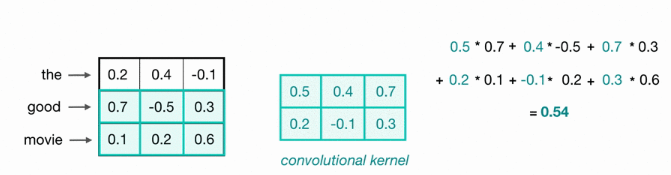
Trong trường hợp phân loại văn bản, hạt nhân tích chập vẫn sẽ là một cửa sổ trượt, công việc duy nhất của nó là xem xét các phần nhúng *d* cho nhiều từ, thay vì các vùng pixel nhỏ trong hình ảnh. Kích thước của hạt nhân tích chập cũng sẽ phải thay đổi theo nhiệm vụ này. Để xem xét chuỗi các từ nhúng, cần có một cửa sổ để xem xét nhiều từ nhúng trong một chuỗi. Các hạt nhân sẽ không còn là hình vuông nữa mà thay vào đó chúng sẽ là một hình chữ nhật rộng với kích thước như nx300 (giả sử *d*=300). Chiều cao của hạt nhân sẽ là số lượng phần nhúng mà nó sẽ nhìn thấy cùng một lúc, tương tự như việc biểu thị một n-gram trong mô hình từ. Chiều rộng của hạt nhân phải dài bằng chiều dài của toàn bộ quá trình nhúng từ.



*Hình 3: Hạt nhân tích chập trong phân loại văn bản*

* Sự biến đổi trên chuỗi từ:

Hãy xem ví dụ về một cặp (2-gam) từ nhúng được lọc sẽ trông như thế nào. Xét một chuỗi từ, mỗi từ được mã hóa dưới dạng vector nhúng có d=3 chiều (thông thường, các vector nhúng sẽ chứa nhiều chiều hơn, nhưng ta lấy 3 chiều để dễ hình dung). Để xem hai từ trong chuỗi ví dụ, chúng ta có thể sử dụng hạt nhân tích chập 2x3. Trong trường hợp này trọng số hạt nhân được đặt trên hai phần nhúng từ; theo hướng đi xuống biểu thị thời gian, do đó, từ “movie” đứng ngay sau “good” trong chuỗi ngắn này. Trọng số hạt nhân và vector nhúng được nhân theo cặp rồi cộng lại để có giá trị đầu ra duy nhất là 0,54.

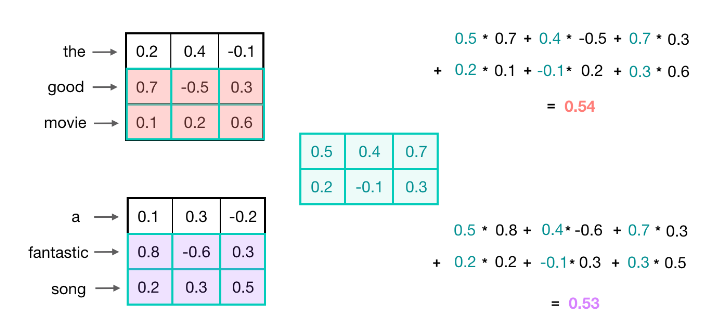


*Hình 4: Cách tính giữa hạt nhân tích chập với hai phần nhúng từ trong chuỗi*

Một mạng lưới thần kinh tích chập sẽ bao gồm nhiều hạt nhân này và khi mạng huấn luyện, các trọng số hạt nhân này sẽ được học. Mỗi hạt nhân được thiết kế để xem xét một từ và (các) từ xung quanh trong một cửa sổ tuần tự và xuất ra một giá trị thể hiện điều gì đó về cụm từ đó. Theo cách này, thao tác tích chập có thể được xem như là trích xuất tính năng dựa trên cửa s , trong đó các tính năng là các mẫu trong các nhóm từ tuần tự biểu thị các đặc điểm như cảm xúc của một văn bản, chức năng ngữ pháp của các từ khác nhau, v.v. Số lượng kênh đầu vào cho lớp chập này là 1 trong vì một nguồn văn bản đầu vào duy nhất sẽ được mã hóa thành một danh sách các từ nhúng.

* Nhận biết các mẫu chung:

Các từ tương tự sẽ có cách nhúng tương tự và phép toán tích chập chỉ là phép toán tuyến tính trên các vectơ này. Vì vậy, khi một hạt nhân tích chập được áp dụng cho các tập hợp từ tương tự khác nhau, nó sẽ tạo ra một giá trị đầu ra tương tự. Trong ví dụ bên dưới, có thể thấy rằng giá trị đầu ra tích chập cho “the good movie” và “a fantastic song” 2-gram đầu vào là gần như nhau vì cách nhúng từ cho các cặp từ đó cũng rất giống nhau.

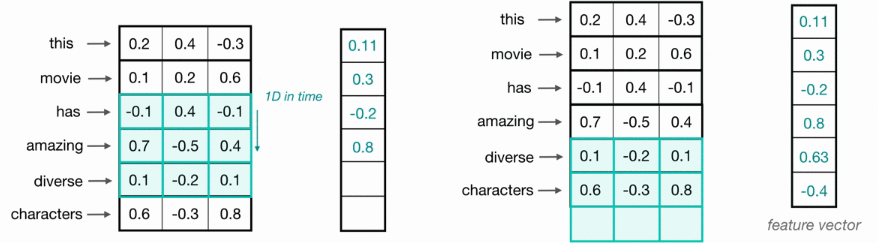


*Hình 5: Ví dụ các từ tương tự sẽ cho ra các giá trị tương tự sau khi áp dụng phép tích chập*

Trong ví dụ này, nhân tích chập đã học cách nắm bắt một tính năng tổng quát hơn; không chỉ là một bộ phim hay bài hát hay mà nói chung là một điều tích cực. Việc nhận biết các loại tính năng cấp cao này có thể đặc biệt hữu ích trong các tác vụ phân loại văn bản thường dựa vào các nhóm chung.

* Phép biến đổi 1D:

Để xử lý toàn bộ chuỗi từ, các hạt nhân này sẽ trượt xuống danh sách các từ nhúng theo trình tự. Đây được gọi là tích chập 1D vì hạt nhân chỉ di chuyển theo một chiều: thời gian. Một hạt nhân sẽ di chuyển từng cái một xuống danh sách các phần nhúng đầu vào, xem xét phần nhúng từ đầu tiên (và một cửa sổ nhỏ chứa các phần nhúng từ tiếp theo), sau đó là phần nhúng từ tiếp theo, v.v. Kết quả đầu ra sẽ là một vectơ đặc trưng chứa nhiều giá trị tương đương với số phần nhúng đầu vào, do đó kích thước chuỗi đầu vào rất quan trọng. Đôi khi một hạt nhân tích chập sẽ không phủ lên các từ nhúng một cách hoàn hảo và do đó có thể cần phải đưa vào một số phần đệm để tính chiều cao của hạt nhân.



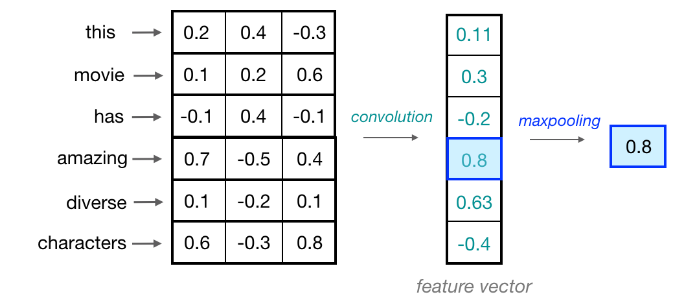
*Hình 6: Minh hoạ đầu ra của tích chập 1D áp dụng cho 1 chuỗi nhúng từ*

* Nhiều hạt nhân:

Giống như trong một mạng nơ-ron tích chập điển hình, một hạt nhân tích chập là không đủ để phát hiện tất cả các loại tính năng khác nhau sẽ hữu ích cho nhiệm vụ phân loại. Để thiết lập một mạng có khả năng học nhiều mối quan hệ khác nhau giữa các từ, chúng ta sẽ cần nhiều hạt nhân có độ cao khác nhau. Các độ cao nắm bắt hiệu quả các mẫu trong các nhóm tuần tự gồm 3, 4 và 5 từ. Giới hạn 5 từ vì những từ ở xa hơn mức đó thường ít liên quan hoặc ít hữu ích hơn trong việc xác định các mẫu trong một cụm từ. Ví dụ:trong câu “this apple is red and a banana is yellow” thì những từ gần nhau như “this”, “apple” và “red” có liên quan nhiều hơn những từ xa nhau như “apple” và “yellow”. Vì vậy, khi một hạt nhân tích chập ngắn trượt lần lượt trên các phần nhúng từ, nó được thiết kế để nắm bắt các tính năng cục bộ hoặc các tính năng trong cửa sổ các từ tuần tự gần đó. Các vectơ đặc trưng đầu ra xếp chồng lên nhau phát sinh từ một số phép toán tích chập này được gọi là lớp tích chập.

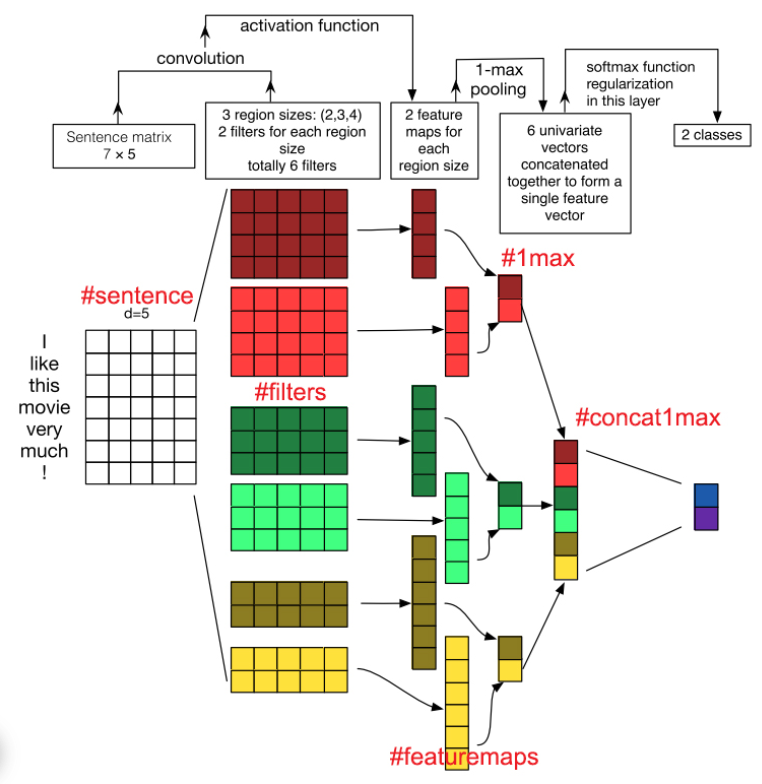
**Maxpooling over Time**

Phép toán tích chập tạo ra một vectơ đặc trưng có thể biểu diễn các đặc điểm cục bộ theo chuỗi các từ nhúng. Vậy vectơ đặc trưng có thể trông như thế nào khi được áp dụng cho một cụm từ quan trọng trong văn bản. Trong trường hợp nhóm đang cố gắng phân loại các bài đánh giá phim với cụm từ “cốt truyện tuyệt vời” đó là một dấu hiệu tốt cho thấy đây là một đánh giá tích cực, bất kể vị trí của nó trong văn bản nguồn. Để chỉ ra sự hiện diện của các đặc trưng cấp cao này, chúng ta cần xác định chúng trong một vectơ, bất kể vị trí nào trong chuỗi đầu vào lớn hơn. Để xác định các đối tượng quan trọng, bất kể vị trí của chúng trong một chuỗi, là loại bỏ thông tin vị trí, ít liên quan hơn. Để thực hiện điều này, chúng ta có thể sử dụng thao tác maxpooling, thao tác này buộc mạng chỉ giữ lại giá trị tối đa trong vectơ đặc tính, đây phải là đặc trưng cục bộ, hữu ích nhất.



*Hình 7: Maxpooling để giữ lại đặc trưng hữu ích trong câu*

**Mô hình thuật toán textCNN**



*Hình 8: Mô hình thuật toán textCNN*

* Đầu vào của mô hình

Đầu vào của mô hình là bình luận cần gán nhãn đã được biến đổi thành một chuỗi số nguyên có chiều dài là sequence\_length. Đây là biểu diễn số hoá của các bình luận (Quá trình tiền xử lý ngữ liệu xem thêm ở mục 3.1 và 3.2)

* Embedding layer

Để tối ưu hoá thời gian huấn luyện và tài nguyên, nhóm sử dụng các từ nhúng đã được đào tạo trước. Những phần nhúng này có được từ việc biểu diễn các từ giống nhau trong cùng một không gian vectơ. Điều này có nghĩa là những từ tiêu cực sẽ được nhóm lại gần nhau và những từ tích cực cũng vậy. Ma trận nhúng từ được tạo ra trước khi thực hiện mô hình (xem thêm ở mục 3.2) với số chiều của word vector là 300 (tức là mỗi từ sẽ là một vector kích thước 1×300). Sử dụng ma trận nhúng làm trọng số cho lớp nhúng Keras. Các bình luận trong bộ ngữ liệu sau khi thực hiện padding có chiều dài là sequence\_length = 100. Như vậy kích thước ma trận của một mẫu bình luận sau khi qua lớp embedding sẽ là 100 x 300.

Ta có kích thước từ vựng của bộ ngữ liệu sau khi loại bỏ các stopwords, các biểu tượng cảm xúc và các dấu câu là 19657 từ. Như vậy ma trận trọng số của lớp embedding sẽ có 19657\*300=5897100 parameters.

* Convolution layers

Nhóm sử dụng 3 lớp tích chập với các hạt nhân tích chập có kích thước lần lượt là 2,3,5. Mỗi lớp tích chập có 32 bộ lọc và không sử dụng padding. Kích thước đầu ra của các lớp tích chập có dạng (V,1,32) với 32 là số lượng bộ lọc áp dụng, V là chiều dài của vector đặc trưng được tính bằng công thức:

V = chiều dài của chuỗi đầu vào - kích thước hạt nhân tích chập.

Số lượng trọng số của mỗi lớp được tính như sau:

32\*(số chiều của word vector\*kích thước hạt nhân tích chập +1)

* Lớp tích chập có kích thước hạt nhân là 2, ta có:

Output Shape = (100-2+1, 1, 32) = (99, 1, 32)

Parameters = 32\*(300\*2+1) = 19232

* Lớp tích chập có kích thước hạt nhân là 3:

Output Shape = (100-3+1, 1, 32) = (98, 1, 32)

Parameters = 32\*(300\*3+1) = 28832

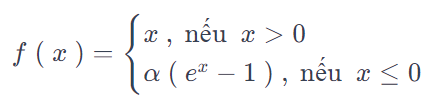
* Lớp tích chập có kích thước hạt nhân là 3:

Output Shape = (100-5+1, 1, 32) = (96, 1, 32)

Parameters = 32\*(300\*5+1) = 48032

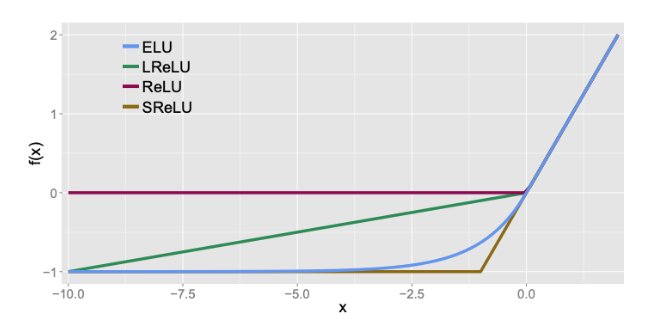
Để bảo đảm giá trị của map đặc trưng, ta cần sử dụng một activation function. Nhóm sử dụng hàm kích hoạt eLU:

Exponential Linear Unit (ELU) là một hàm kích hoạt giúp cải thiện độ chính xác của mô hình và giảm thời gian đào tạo. Nó được biểu diễn về mặt toán học như sau:



Trong công thức trên, α thường được đặt thành 1. Nó xác định mức độ bão hòa của đầu vào âm.

Hàm kích hoạt ReLU trở nên nổi tiếng khi giải quyết được vấn đề gradient biến mất (cụ thể là gradient của các hàm kích hoạt, như sigmoid, trở nên rất nhỏ gây khó khăn cho việc huấn luyện các mô hình lớn hơn). Tuy nhiên, cùng lúc đó, ReLU đã tạo ra một vấn đề cho chính mình, vấn đề này xảy ra khi ReLU xuất ra 0 trên bất kỳ đầu vào nào. Ngược lại, ELU (như batch normalization) có các giá trị âm giúp đưa giá trị trung bình về gần 0. ELU là một hàm trơn tru đối với các giá trị âm, giúp nó chống nhiễu tốt hơn.



*Hình 9: Biểu đồ các hàm kích hoạt*

* Max pooling layers:

Kích thước của các vector đặc trưng phụ thuộc vào ma trận văn bản và ma trận bộ lọc. Nói cách khác, vector đặc trưng sẽ có kích thước thay đổi chứ không cố định. Để đưa ma trận đặc trưng này về kích thước như nhau hoặc chỉ muốn giữ lại các đặc trưng tiêu biểu, chúng ta có thể sử dụng max-pooling để lấy ra các giá trị lớn nhất trong map đặc trưng. Điều này giúp giảm chiều dữ liệu, tăng tốc độ tính toán. Chúng ta lấy ra 1 giá trị lớn nhất trong từng vector đặc trưng. Việc này giúp ta có ma trận output có cùng kích thước. Như vậy, ta chỉ lấy 1 đặc trưng trội nhất ở tất cả các lớp cnn để phục vụ cho bài toán. Kích thước đầu ra của các lớp max-pooling này là (1,1,32).

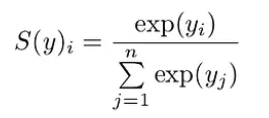
* Concatenate layers

Sau khi áp dụng 1-max pooling, chúng ta đã có những vector có kích thước cố định là 1×1 của 32 thành phần (số bộ lọc). Lớp concatenate kết hợp các đầu ra của các lớp MaxPooling thành một tensor duy nhất có kích thước (3,1,32). Điều này tạo ra một biểu diễn tổng hợp của các đặc trưng cấp độ cao nhất từ các kernel khác nhau.

* Flatten layer: Lớp flatten dùng để làm phẳng tensor từ dạng nhiều chiều thành dạng một chiều để chuẩn bị cho các lớp fully connected tiếp theo. Kích thước đầu ra của flatten layer là 3\*1\*32 = 96
* Dense layer:

Một vector xác suất tương từng loại nhãn cho bình luận cần phân loại sẽ được dự đoán thông qua hàm softmax. Nhãn có giá trị xác suất cao nhất trong vector là nhãn ta cần gán. Số parameters của layer này được tính như sau:  
Parameters = (32\*3 + 1)\*3 = 291

Softmax là một hàm kích hoạt chia tỷ lệ số/log thành xác suất.Đầu ra của Softmax là một vectơ v với xác suất của từng kết quả có thể xảy ra. Xác suất trong vectơ v có tổng bằng 1 đối với tất cả các kết quả hoặc loại có thể xảy ra. Về mặt toán học, Softmax được định nghĩa như sau:



Trong đó:

|  |  |
| --- | --- |
| *y* | Một vector đầu vào của hàm softmax, nó bao gồm n phần tử ứng với n nhãn đầu ra |
| *y*i | Tất cả các giá trị yi là các phần tử của vectơ đầu vào của hàm softmax và chúng có thể nhận bất kỳ giá trị thực nào, dương, 0 hoặc âm. Ví dụ: mạng nơ-ron có thể tạo ra một vectơ như (-0,62, 8,12, 2,53), đây không phải là phân phối xác suất hợp lệ, do đó tại sao softmax lại cần thiết |
| *exp(yi)* | Hàm mũ tiêu chuẩn được áp dụng cho từng phần tử của vectơ đầu vào. Điều này mang lại giá trị dương trên 0, giá trị này sẽ rất nhỏ nếu đầu vào âm và rất lớn nếu đầu vào lớn. Tuy nhiên, nó cố định trong phạm vi (0, 1), đây là yêu cầu bắt buộc của xác suất. |
|  | Là một số hạng chuẩn hóa. Nó đảm bảo rằng tất cả các giá trị đầu ra của hàm sẽ có tổng bằng 1 và mỗi giá trị nằm trong phạm vi (0, 1), do đó tạo thành một phân bố xác suất hợp lệ. |

*Bảng 4: Các thành phần của hàm softmax*

**Hàm loss: Categorical crossentropy**

Hàm loss này thích hợp cho bài toán phân loại đang giải quyết với 3 lớp (multiclass classification) và dữ liệu đầu ra được biểu diễn dưới dạng one-hot encoding. Đối với mỗi mẫu dữ liệu, công thức tính loss như sau:

Trong đó:

* y\_true là vector one-hot encoding của nhãn thực tế.
* y\_pred là vector xác suất dự đoán cho từng lớp được tính từ softmax

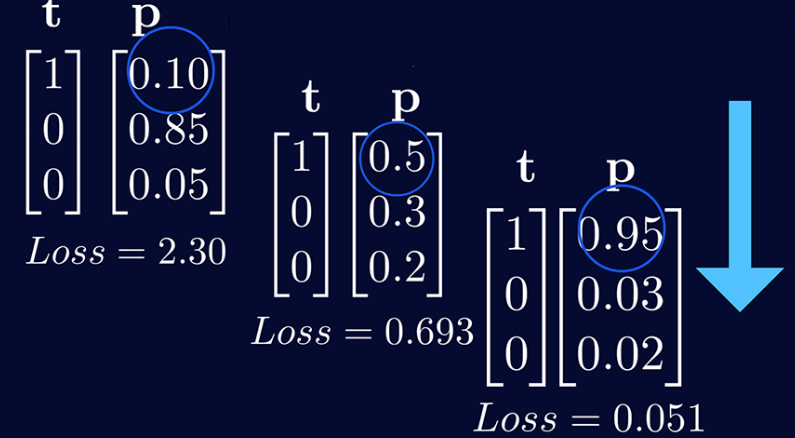
Hàm loss này đo lường sự chênh lệch giữa phân phối xác suất dự đoán và phân phối xác suất thực tế. Nếu các giá trị xác suất dự đoán gần với nhãn thực tế, loss giảm; ngược lại, nếu chúng chênh lệch nhiều, loss tăng.

Phân phối thực ytrue là một vectơ one-hot có giá trị bằng 1 tại một trong các chỉ số và bằng 0 ở mọi nơi khác. Nếu một bình luận nhất định thuộc về lớp i thì trong vectơ phân phối thực ytrue , ta có:

ytrue [i] = 1 và ytrue[j] = 0, ji

Ta thấy rằng N - 1 số hạng trong phép tính tổng bằng 0 và biểu thức trên sẽ trở thành biểu thức đơn giản hóa sau:

Hình bên dưới trình bày một số ví dụ về phân phối đúng và dự đoán. Trong bài toán phân loại phân loại văn bản gồm 3 nhãn: HATE (2), OFFENSIVE (1) và CLEAN (0), nếu lớp mục tiêu là CLEAN, thì tổn thất entropy chéo phân loại sẽ được giảm thiểu khi mạng dự đoán điểm xác suất gần bằng 1 cho lớp chính xác. Điều này hoạt động tương tự với các lớp mục tiêu khác.



*Hình 10: Minh hoạ cách tính loss trong một số trường hợp*

**Tối ưu hoá Adam**

Adaptive Moment Estimation - Adam là một thuật toán tốc độ học tập thích ứng được thiết kế để cải thiện tốc độ đào tạo trong mạng lưới thần kinh sâu và đạt được sự hội tụ một cách nhanh chóng. Nó tùy chỉnh tốc độ học của từng tham số dựa trên lịch sử độ dốc của nó và việc điều chỉnh này giúp toàn bộ mạng lưới thần kinh học hiệu quả.

Xem xét gradient descent tiêu chuẩn:

Trong đó: θ là tham số mô hình, α là tốc độ học và gₜ là độ dốc của hàm chi phí đối với các tham số.

Trong thuật toán gradient descent tiêu chuẩn, tốc độ học α ​​là cố định. Chúng ta cần bắt đầu ở tốc độ học cao và thay đổi alpha theo cách thủ công theo từng bước hoặc theo một lịch trình học nào đó. Tốc độ học thấp hơn khi bắt đầu sẽ dẫn đến sự hội tụ rất chậm, trong khi tốc độ học rất cao khi bắt đầu có thể bỏ lỡ cực tiểu. Adam giải quyết vấn đề này bằng cách điều chỉnh tốc độ học α ​​cho từng tham số θ, cho phép hội tụ nhanh hơn so với phương pháp giảm độ dốc tiêu chuẩn với tốc độ học toàn cầu không đổi.

Bằng cách cộng thêm một phần nhỏ của hướng di chuyển trước đó, Momentum giúp cho quá trình di chuyển của mô hình mượt mà và ổn định hơn, tránh bị rung lắc do những thay đổi đột ngột của độ dốc. Điều này giúp cho quá trình huấn luyện nhanh hơn và hội tụ chính xác hơn. Kỹ thuật momentum chỉ ra rằng vì một số bước cuối cùng gần như theo cùng một hướng, nên có thể áp dụng mức tăng cho bước hiện tại để di chuyển theo hướng đó, do đó cần thực hiện ít bước hơn.

Trong đó:

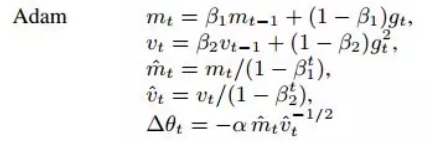
* là vector momentum v tại thời gian t, là một là một hàm của vectơ động lượng trước đó
* Siêu tham số γ (momentum decay) là độ giảm động lượng, nó kiểm soát mức độ ảnh hưởng của vectơ động lượng trước đó. Giá trị γ càng lớn, ảnh hưởng của vectơ động lượng trước đó càng mạnh. Ngược lại, giá trị γ càng nhỏ, vectơ động lượng càng nhanh trở nên "quên" các hướng di chuyển trước đó.
* η là tỷ lệ học điều khiển độ lớn của bước di chuyển theo hướng gradient. Giá trị η càng lớn, bước di chuyển càng lớn và quá trình huấn luyện có thể nhanh hơn nhưng dễ bị dao động. Ngược lại, giá trị η càng nhỏ, bước di chuyển càng nhỏ và quá trình huấn luyện có thể chậm hơn nhưng ổn định hơn.

Ta thấy được cập nhật hiệu quả hơn so với lúc ban đầu bằng cách trừ đi momentum.

Xem xét “độ dốc” của bề mặt lỗi đối với từng tham số để cập nhật tốc độ học một cách thích ứng. Các tham số có độ dốc cao nhận được các bước cập nhật nhỏ hơn và độ dốc thấp cho phép các bước cập nhật lớn hơn. Vì vậy, trong khi momentum tập trung vào việc tăng tốc theo các hướng nhất quán thì RMSProp dựa trên việc kiểm soát độ vọt bằng cách điều chỉnh kích thước bước.

Phương trình đầu tiên là một giá trị trung bình di chuyển có trọng số của các bình phương độ dốc (gradient), về cơ bản nó chính là phương sai của độ dốc. Ở đây, chúng ta thấy rằng tỷ lệ học trong việc cập nhật θ được chia cho căn bậc hai của giá trị trung bình di chuyển của các bình phương độ dốc. Điều này có nghĩa là khi phương sai của độ dốc cao, chúng ta giảm tỷ lệ học vì muốn cẩn thận hơn. Ngược lại, khi phương sai của độ dốc thấp, chúng ta tăng tỷ lệ học để tiến nhanh hơn tới điểm tối ưu.

Công thức của adam được tính như sau:



Các siêu tham số chính trong Adam là:

α - Kích thước bước để tối ưu hóa.

β₁ — Tốc độ suy giảm của động lượng.

β₂ — Tốc độ phân rã đối với gradient bình phương.

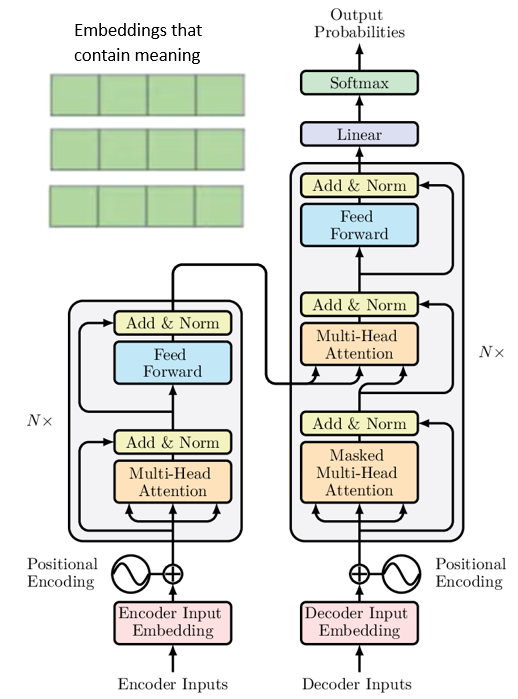
— Giá trị nhỏ để tránh chia cho 0. Nó thường là khoảng 1e-8.

Cùng với nhau, các tham số này cho phép Adam hội tụ nhanh hơn trong khi vẫn ổn định về mặt số lượng.

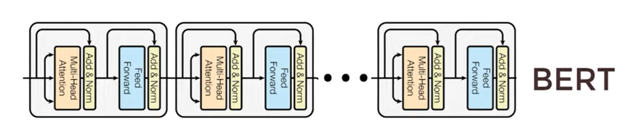
## **2.2. BERT**

**2.2.1. Kiến trúc mô hình Transformer và mô hình BERT**

Khối encoder tạo ra embedding cho tất cả các từ một cách đồng thời. Những embedding này là những vectors chứa đựng ý nghĩa của các từ. Các từ tương tự nhau có giá trị gần nhau trong không gian vectơ đó. Khối decoder sử dụng nhưng embedding này từ khối encoding và các từ đã được tạo ra trước đó ở trong câu, sau đó sử dụng chúng để sinh ra các từ tiếp theo. Và chúng ta cứ tiếp tục sinh ra từng từ một cho đến hết câu. Chúng ra có thể thấy rõ sự phân biệt trong công việc. Khối encoder học ngôn ngữ là gì, cú pháp là gì, và quan trọng nhất, ngữ cảnh là gì. Khối decoder học về cách các từ ngữ liên quan đến nhau. Cả hai, một cách độc lập, đều có sự hiểu biết cơ bản về ngôn ngữ. Và chính vì sự hiểu biết này mà chúng ra có thể tách kiến trúc này ra và xây dựng các hệ thống để hiểu về ngôn ngữ. Nếu chúng ta xếp chồng các khối encoder lại với nhau chúng ta có được mô hình BERT (Bidirectional Encoder Representation from Transformer)

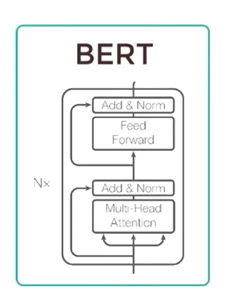


*Hình 11: Kiến trúc mô hình Transformer*

**

*Hình 12: Kiến trúc mô hình BERT*

Chúng ta có thể sử dụng BERT để học:

- Language translation

- Question answering

- Sentiment analysis

- Text classification

- …

→ Chúng ta có thể huấn luyện BERT để hiểu ngôn ngữ và fine-tune BERT cho các nhiệm vụ cụ thể dựa trên vấn đề mà ta muốn giải quyết

Việc huấn luyện BERT có 2 phần:

1. Pretraining:

* Masked Language Model (MLM): Nhiệm vụ này kích hoạt khía cạnh học sâu hai chiều của mô hình. Trong nhiệm vụ này, một phần của các input tokens được ẩn đi (thay thế bằng token [MASK]) một cách ngẫu nhiên và mô hình cố gắng dự đoán những đơn vị này đã được ẩn - không phải toàn bộ chuỗi input. Các tokens được dự đoán từ mô hình sau đó được đưa vào một lớp softmax output qua từ vựng để có được từ output cuối cùng.

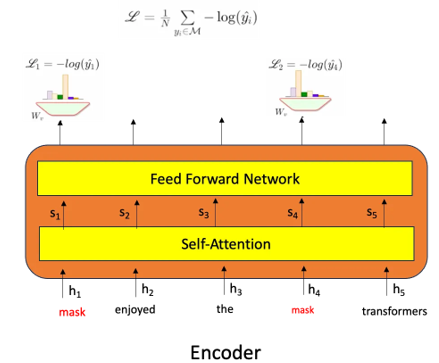
Tuy nhiên, điều này tạo ra một sự mâu thuẫn giữa 2 nhiệm vụ pre-training và fine-tuning vì nhiệm vụ sau không liên quan đến việc dự đoán các từ bị ẩn trong hầu hết các nhiệm vụ phụ thuộc. Điều này được giảm nhẹ bằng cách ẩn các input tokens một cách tinh tế.

Khoảng 15% từ được ẩn trong quá trình huấn luyện, nhưng không phải tất các các từ bị ẩn đến được thay thế bằng token [MASK]

· 80% của số đó được thay bằng token [*MASK*].

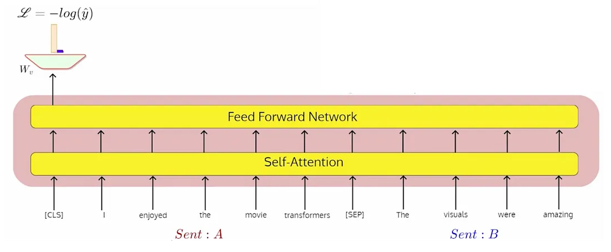
· 10% được thay bằng các tokens ngẫu nhiên

· 10% còn lại được thay bằng giá trị cũ của từ trước khi bị ẩn đi



*Hình 13: Masked Language Model*

* Next Sentence Prediction (NSP): Mô hình Ngôn Ngữ không trực tiếp nắm bắt mối quan hệ giữa hai câu, điều này quan trọng trong nhiều nhiệm vụ phụ thuộc như Question Answering (QA) và Natural Language Inference (NLI). Mô hình được dạy về mối quan hệ giữa các câu thông qua việc huấn luyện trên nhiệm vụ NSP được nhị phân hóa.



*Hình 14: Next Sentence Prediction*

Trong nhiệm vụ này, 2 câu A và B được chọn để pre-training:

Chúng ta có 2 tokens đặc biệt là [CLS] (class) và [SEP] (separator). Mục đích ở đây là kiểm tra xem câu B có phải là câu tiếp theo của câu A hay không. Vì vậy, đầu tiên chúng ta lấy tokens [CLS] để biểu diễn và giải quyết nhiệm vụ nhị phân này. Ta đã có các câu tự nhiên với A, B là một cặp câu và nhãn (1/0) đại diện cho nhãn ‘Next sentence’. Trong quá trình huấn luyện, chúng ta truyền vào inputs A, B và một nhãn, nhiệm vụ của mô hình là tối đa hoá xác suất của nhãn (1/0)

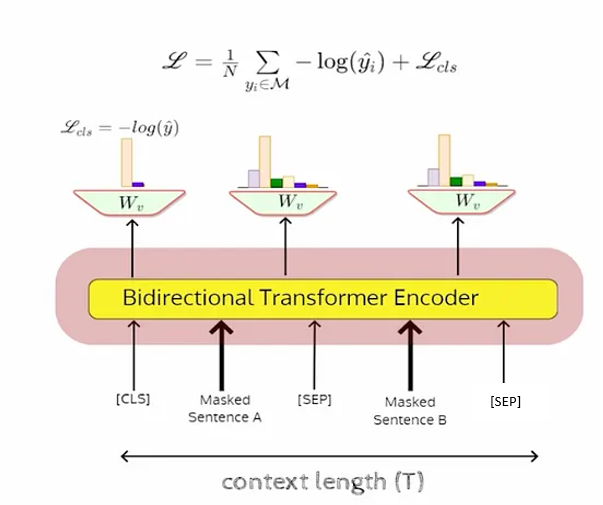
· 50% B là câu tiếp theo của câu A.

· 50% B là một câu ngẫu nhiên trong dữ liệu.

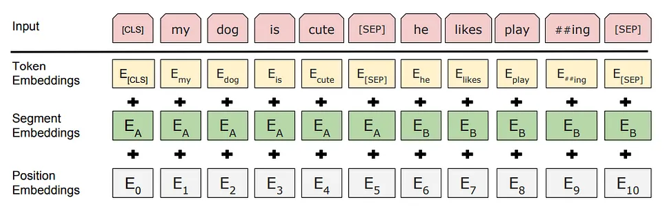
**2.2.2. Training mô hình BERT**

**Input mô hình và quá trình embedding**

Mô hình được huấn luyện với cả hai nhiệm vụ trên một cách đồng thời, nghĩa là các từ bị ẩn không ảnh hưởng đến việc tính toán các từ bị ẩn khác và tính toán câu sau có phải câu tiếp theo không. Việc này có thể được thực hiện bằng việc sử dụng inputs và outputs một cách khôn ngoan.

·

*Hình 14: Mô hình được huấn luyện với cả hai nhiệm vụ trên một cách đồng thời*

**

*Hình 15: Input mô hình và quá trình embedding*

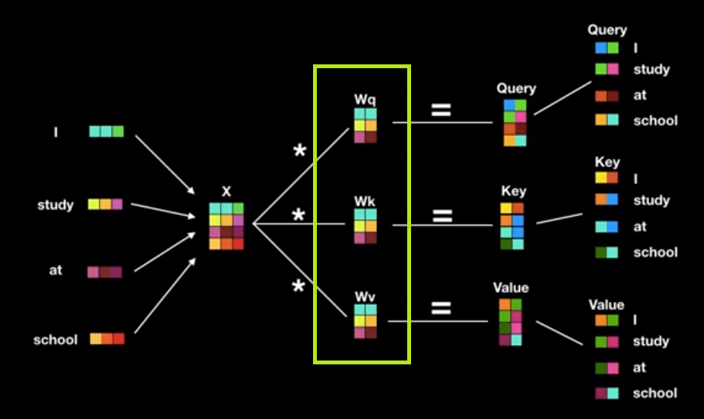
* Token Embedding: Một token [CLS] được thêm vào chuỗi input ở vị trí đầu tiên của chuỗi và token [SEP] được thêm vào cuối mỗi câu.
* Segment Embedding: Một đánh dấu xác định câu A hay câu B được thêm vào mỗi token. Việc này cho phép khối encoder phân biệt giữa các câu trong chuỗi input.
* *Position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

**Cơ chế Attention**

* Scale dot-product attention

Đây chính là một cơ chế self-attention cho phép mỗi từ 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 là ma trận X kích thước m x n, với m,n lần lượt là độ dài câu và số chiều của một vector nhúng từ (768).

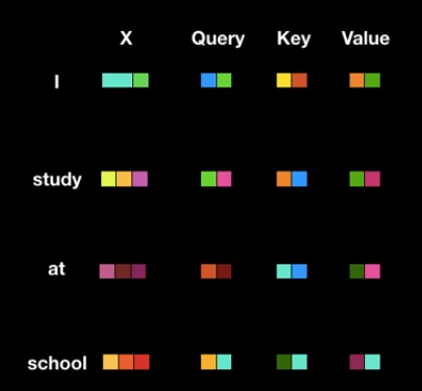
* Selt attention



*Hình 16: Quá trình thực hiện self attention*

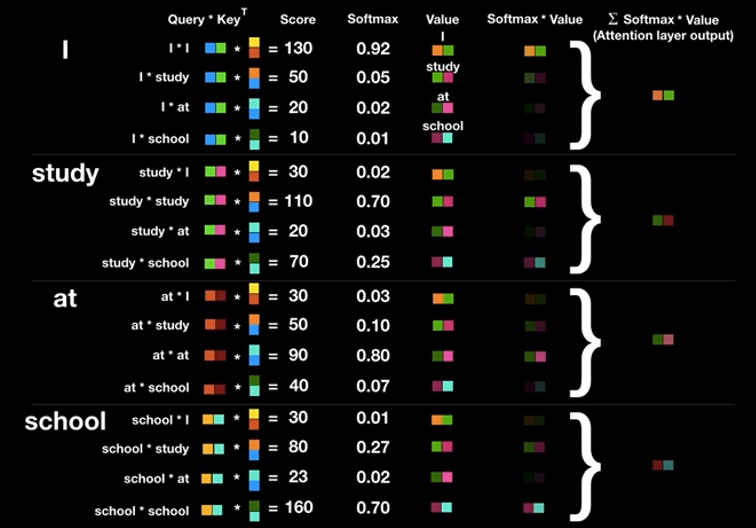
Trong khung màu vàng là 3 ma trận Wq, Wk, Wv  chính là những hệ số mà model cần huấn luyện. Sau khi nhân các ma trận này với ma trận đầu vào X ta thu được ma trận Q,K,V (tương ứng với trong hình là ma trận Query, Key và Value). Ma trận Query và Key có tác dụng tính toán ra phân phối score cho các cặp từ. Ma trận Value sẽ dựa trên phân phối score để tính ra vectơ 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



*Hình 17: Các vector Q, K, V ứng với từng từ trong câu input*

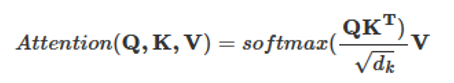
Để tính ra score cho mỗi cặp từ (wi, wj), chúng ta sẽ tính toán dot-product giữa query với key, phép tính này nhằm tìm ra mối liên hệ trọng số của các cặp từ. Tuy nhiên điểm số sau cùng là điểm số chưa được chuẩn hóa. Do đó chúng ta chuẩn hóa bằng một hàm softmax để đưa về một phân phối xác suất mà độ lớn sẽ đại diện cho mức độ attention của từ query tới từ key. Trọng số càng lớn càng chứng tỏ từ wi trả về một sự chú ý lớn hơn đối với từ wj. 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 18: Kết quả tính attention vector cho toàn bộ các từ trong câu*

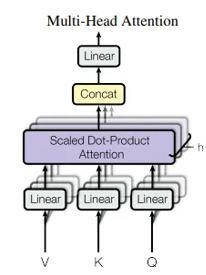
Như vậy, đầu vào để tính attention sẽ bao gồm ma trận Q (mỗi dòng của nó là một vector query đại diện cho các từ input), ma trận K (tương tự như ma trận Q, mỗi dòng là vector key đại diện cho các từ input). Hai ma trận Q,K được sử dụng để tính attention mà các từ trong câu trả về cho 1 từ cụ thể trong 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).

Trong thực hành chúng ta tính toán hàm attention trên toàn bộ tập các câu truy vấn một cách đồng thời được đóng gói thông qua ma trận Q. keys và values cũng được đóng gói cùng nhau thông qua matrix K và V. Phương trình Attention như sau:



Việc chia cho dk là số dimension của vector key nhằm mục đích tránh tràn luồng nếu số mũ là quá lớn.

* Multi head Attention



*Hình 19: Multi-Head attention*

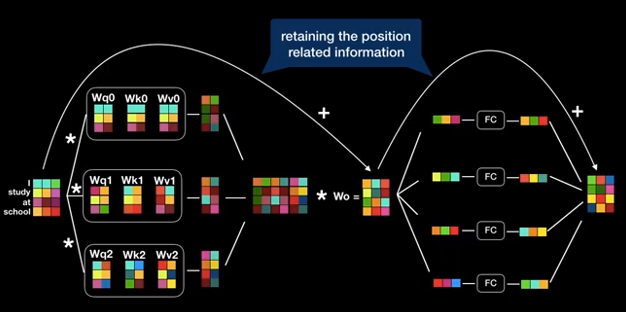
Như vậy sau quá trình Scale dot production chúng ta sẽ thu được 1 ma trận attention. Các tham số mà model cần tinh chỉnh chính là các ma trận Wq, Wk, W. Mỗi quá trình như vậy được gọi là 1 head của attention. Khi lặp lại quá trình này nhiều lần ta sẽ thu được quá trình Multi-head Attention.

Sau khi thu được các 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.

*MultiHead(Q,K,V)=concatenate(head1,head2,...,headh)W0*

*Headi = Attention(Qi,Ki,Vi)*

Để trả về output có cùng kích thước với ma trận input chúng ta chỉ cần nhân với ma trận W0 chiều rộng bằng với chiều rộng của ma trận input.

**

*Hình 20: Sơ đồ của 1 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 sublayer 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**

Việc triển khai cho Mạng Nơ-ron Lan truyền Xuôi (FFN) là rất đơn giản. Nó yêu cầu hai lớp tuyến tính, hoặc lớp dense. Lớp dense đầu tiên có kích thước là (d\_model, d\_ffn), và lớp thứ hai có kích thước là (d\_ffn, d\_model).

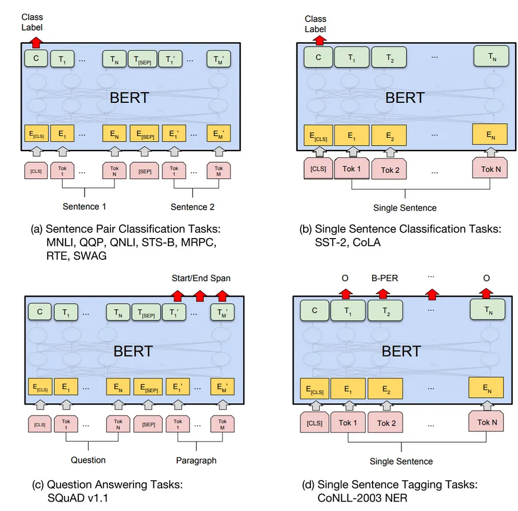


**2.2.3. Fine Tuning**

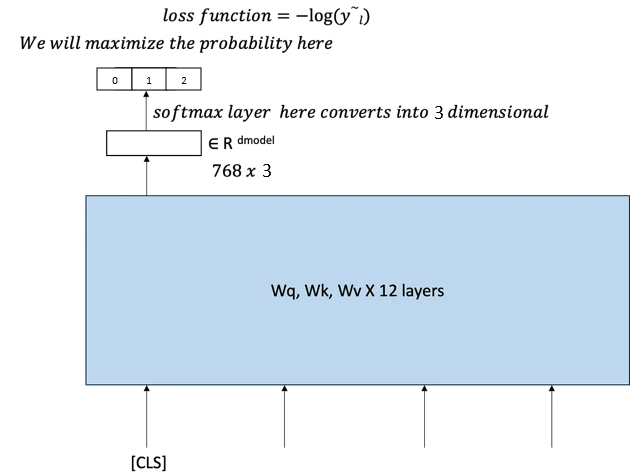
Fine-tuning cho những nhiệm vụ phụ thuộc được thực hiện bằng cách thay đổi các inputs hoặc outputs thích hợp. Trong quá trình tổng quát, để huấn luyện các mô hình cụ thể cho từng nhiệm vụ, chúng ta thêm một lớp output bổ sung cho BERT và fine-tune mô hình kết quả - tất cả các tham số, từ đầu đến cuối. Một kết quả tích cực của việc thêm lớn input/output mà không thay đổi mô hình BERT là chỉ cần học một số lượng tham số tối thiểu từ đầu, là cho quy trình nhanh chóng, giảm chi phí và sử dụng tài nguyên một cách hiệu quả.

Trong Sentence Classification, trạng thái cuối cùng tương ứng với token [*CLS*] được sử dụng làm input cho các lớp bổ sung thực hiện công việc dự đoán.

* Lấy một câu có độ dài nhỏ hơn 512 và đưa nó làm input cho BERT.
* Thêm vào một đầu phân loại (có thể là bất kỳ mô hình máy học phù hợp nào). Khởi tạo các tham số của đầu phân loại một cách ngẫu nhiên.
* Bây giờ, huấn luyện toàn bộ mô hình bào gồm cả các tham số của BERT đã được pre-train cho tập dữ liệu mới. Tuy nhiên chúng ta không ẩn các từ trong chuỗi input (đây là lý do tại sao chúng ta thay thế 10% các từ bị ẩn bằng các từ ngẫu nhiên trong quá trình pre-train). Quan sát thấy rằng mô hình được sử dụng trong đầu phân loại nhanh chóng hội tụ với số lượng mấu huấn luyện ít hơn so với phương pháp feature-base.



*Hình 21: Các nhiệm vụ của mô hình BERT*



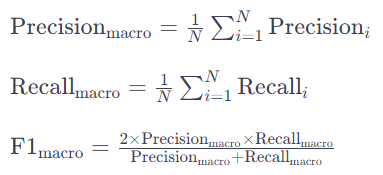
*Hình 22: Fine tuning mô hình BERT cho text classification*

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 x 3) và cũng như học toàn bộ trọng số Wq, Wk, Wv. Điều này có nghĩa là mọi thứ trong mạng bắt đầu chuyên sâu cho nhiệm vụ Text Classification, nghĩa là nó sẽ cố gắng điều chỉnh các tham số để thực hiện tốt trên nhiệm vụ này.

## **2.3. Các metric đánh giá mô hình**

**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:

**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:



Trong đó: Precisioni ​và Recalli ​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 nhầm lẫn 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).

# **3. Cài đặt**

Bước đầu tiên trong phân lớp văn bản là biến đổi văn bản từ chuỗi ký tự về dạng phù hợp với các giải thuật học máy. Đặc điểm chung của nguồn dữ liệu văn bản là không có cấu trúc (độ dài khác nhau) trong khi đa số các giải thuật đòi hỏi dữ liệu huấn luyện phải có cấu trúc (chiều dài các vectơ đặc trưng phải giống nhau chẳng hạn). Bước tiếp theo là huấn luyện mô hình học tự động từ bảng dữ liệu này.

## **3.1. Tiền xử lý ngữ liệu**

Nhóm tiến hành tiền xử lý dữ liệu qua bốn bước sau:

*(1) Phân đoạn văn bản thành các từ riêng lẻ bằng công cụ pyvi tool:*

Tách từ (Word Segmentation) là một bước quan trọng không thể thiếu trong xử lý ngôn ngữ tự nhiên, nhằm xác định được ranh giới các từ có trong văn bản. Đối với các ngôn ngữ như tiếng Anh, tiếng Pháp, tiếng Đức việc tách từ được thực hiện khá đơn giản dựa vào các ký tự phân cách như: khoảng trắng, ký tự tab, các dấu câu, dấu ngoặc, v.v. Ngược lại, trong tiếng Việt, ngoài từ đơn (một âm tiết), còn có từ ghép (đa âm tiết). Khoảng trắng ngoài việc ngăn cách các từ với nhau, còn được dùng để ngăn cách các âm tiết (syllable) của một từ ghép. Điều này gây khó khăn cho việc tách từ tự động một cách chính xác, ảnh hưởng đến kết quả của các bài toán phân lớp văn bản. Hai tiếp cận chính để tách từ tiếng Việt là dựa trên từ điển và tiếp cận thống kê. Ngoài ra còn có một số phương pháp kết hợp cả hai tiếp cận trên.

Nhóm sử dụng công cụ pyvi tool - một gói mã nguồn mở của Python - để chia nhỏ các bình luận thành các từ riêng lẻ. Pyvi thực hiện tách từ dựa trên một mô hình máy học sử dụng tiếp cận thống kê, cho phép nó tự động học và điều chỉnh theo cấu trúc ngôn ngữ thực tế của tiếng Việt. Mỗi từ hoặc cụm từ được phân tách bằng một khoảng trắng.

Ví dụ, mẫu bình luận được trích từ bộ ngữ liệu “Chúc anh em năm mới vui vẻ, an khang thịnh vượng.” sau khi được xử lý bằng công cụ pyvi tool sẽ cho ra kết quả:

Chúc anh\_em năm mới vui\_vẻ , an\_khang thịnh\_vượng .

Ta thấy các từ hoặc các dấu câu như “chúc”, “năm”, “mới”, “,”, “.” và các cụm từ như “anh\_em”, “vui\_vẻ”, “an\_khang”, “thịnh\_vượng” đã được phân tách ra bằng một khoảng trắng.

*(2) Loại bỏ các stopwords*

Stopwords là những từ thường gặp trong ngôn ngữ nhưng không mang nhiều thông tin, chẳng hạn như các từ chỉ quan hệ, giới từ, đại từ,... Việc loại bỏ stopwords có thể giúp đơn giản hóa dữ liệu, giảm thiểu nhiễu và cải thiện hiệu quả của các thuật toán xử lý ngôn ngữ tự nhiên. Có nhiều cách để loại bỏ stopwords, một cách phổ biến là sử dụng từ điển stopwords. Từ điển stopwords là một tập hợp các từ được xác định là stopwords. Để loại bỏ stopwords, ta có thể sử dụng một hàm để kiểm tra xem một từ có nằm trong từ điển stopwords hay không.

Với mẫu bình luận "Chúc anh em năm mới vui vẻ, an khang thịnh vượng." ở ví dụ trên, sau khi thực hiện word segmentation và loại bỏ stopwords, ta được kết quả:

Chúc anh\_em vui\_vẻ , an\_khang thịnh\_vượng .

Ta thấy các từ “năm” và “mới” đã bị loại bỏ. Các từ còn lại là những từ mang nhiều thông tin, có thể giúp chúng ta hiểu được nội dung của bình luận.

*(3) Loại bỏ các biểu tượng cảm xúc*

Các biểu tượng cảm xúc (emoji) là những ký tự đặc biệt được sử dụng để biểu đạt cảm xúc, thái độ hoặc ý tưởng. Trong một số trường hợp, các biểu tượng cảm xúc có thể mang lại nhiều thông tin trong việc xác định nhãn cho bình luận, chẳng hạn như trong các bình luận thể hiện cảm xúc. Tuy nhiên, trong nhiều trường hợp khác, các biểu tượng cảm xúc lại không mang lại nhiều thông tin, thậm chí có thể gây nhiễu cho quá trình xử lý ngôn ngữ tự nhiên.

Trong bài toán phát hiện lời nói chán ghét trên mạng xã hội, nhóm đã nhận thấy rằng các biểu tượng cảm xúc thường không mang lại nhiều thông tin trong việc xác định nhãn cho bình luận. Do đó, nhóm đã tiến hành loại bỏ các biểu tượng cảm xúc khỏi tập dữ liệu.

*(4) Chuyển đổi tất cả văn bản thành chữ thường (không áp dụng cho mô hình pre-trained transformer)*

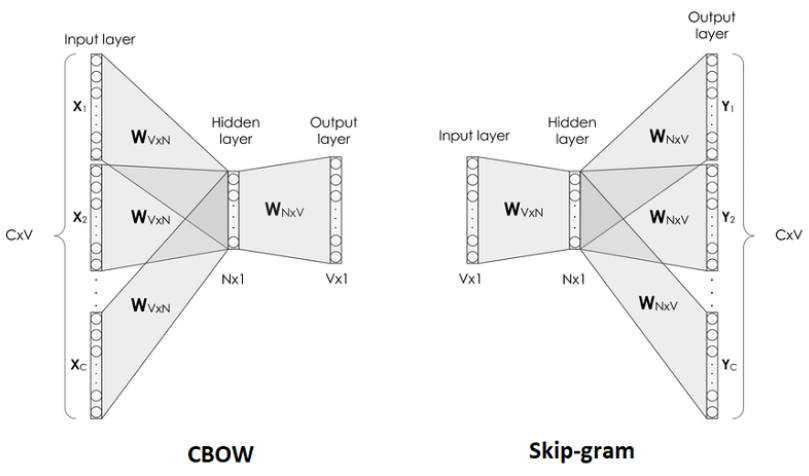
Trong tiếng Việt, chữ viết in hoa thường được sử dụng để nhấn mạnh, thể hiện cảm xúc hoặc thể hiện chức năng ngữ pháp. Tuy nhiên, trong một số trường hợp, việc sử dụng chữ viết in hoa có thể gây nhiễu cho quá trình xử lý ngôn ngữ tự nhiên. Trong nhiều trường hợp, các thuật toán xử lý ngôn ngữ tự nhiên (NLP) hoạt động hiệu quả hơn với văn bản ở dạng chữ thường. Điều này giúp giảm kích thước từ vựng và tăng tính đồng nhất của dữ liệu.

Lưu ý rằng sau các bước tiền xử lý dữ liệu, một số bình luận sẽ trở thành chuỗi rỗng, nhóm thực hiện xoá các chuỗi rỗng này ra khỏi tập ngữ liệu.

## **3.2. Trích xuất đặc trưng**

Word embedding là một kỹ thuật biểu diễn từ ngữ dưới dạng các vector số thực trong không gian đa chiều. Mục tiêu của word embedding là ánh xạ các từ ngữ vào các vector sao cho các từ có ý nghĩa tương tự sẽ có các vector gần nhau trong không gian này. Kỹ thuật word embedding word2vec cung cấp các biểu diễn vectơ riêng biệt cho các từ trong từ vựng. Có hai cách xây dựng mô hình Word2vec để biểu diễn sự phân tán của từ trong không gian vector là sử dụng ngữ cảnh để dự đoán mục tiêu (Continuous Bag Of Word) và sử dụng một từ để dự đoán ngữ cảnh mục tiêu (Continuous skip-gram). Thuật toán CBOW tốn ít thời gian huấn luyện mô hình hơn Skip-gram. Tuy nhiên, Skip-gram có độ chính xác cao hơn và có chứa cả những từ ít xuất hiện.

Trong word2vec, ta không trực tiếp sử dụng thông tin hình thái học. Trong cả mô hình skip-gram và CBOW, ta sử dụng các vector khác nhau để biểu diễn các từ ở các dạng khác nhau. Chẳng hạn, “dog” và “dogs” được biểu diễn bởi hai vectơ khác nhau, trong khi mối quan hệ giữa hai vectơ đó không biểu thị trực tiếp trong mô hình. Điều này dẫn đến sự thiếu hiểu biết về cấu trúc bên trong của ngôn ngữ. Đây là một hạn chế đối với ngôn ngữ giàu hình thái vì nó bỏ qua quan hệ cú pháp của từ.



*Hình 23: Hai cách xây dựng mô hình word2vec*

Vì nhiều cách tạo từ tuân theo các quy tắc trong các ngôn ngữ giàu hình thái nên có thể cải thiện cách biểu diễn vectơ cho các ngôn ngữ này bằng cách sử dụng thông tin ở cấp độ ký tự. Từ quan điểm này, fastText đề xuất phương thức embedding từ con (subword embedding) để cải thiện khả năng biểu diễn vectơ cho ngôn ngữ giàu hình thái. Nó là một phần mở rộng của mô hình word2vec. Trong khi mô hình Word2Vec cung cấp khả năng nhúng vào các từ, trong khi fastText cung cấp khả năng nhúng vào ký tự n-gram. Giống như mô hình word2vec, fastText sử dụng CBOW và Skip-gram để tính toán các vectơ. FastText cũng có thể xử lý các từ nằm ngoài từ vựng, tức là văn bản nhanh có thể tìm thấy các từ nhúng không có tại thời điểm đào tạo.

Phần Word embedding trong đồ án được sử dụng từ cc.vi.300.vec, đây là tệp .vec nhúng từ đã được đào tạo trước bằng phương pháp fastText. Nhóm tạo một đối tượng tokenizer để tạo một từ điển các từ xuất hiện trong tập huấn luyện, không bao gồm các ký tự đặc biệt, dấu câu không cần thiết, hoặc các thành phần không liên quan khác. Sau đó nhóm tạo một ma trận nhúng có kích thước là (num\_words, embedding\_dim), trong đó num\_words sẽ bằng số lượng các từ xuất hiện trong tập huấn luyện + 1, và embedding\_dim là 300. Sau đó điền vào ma trận nhúng các vector nhúng của từng từ trong tập huấn luyện. Nếu từ trong tập huấn luyện nằm ngoài từ vựng của file .vec thì vector của từ đó sẽ bằng 0. Kích thước của từ điển là 10000 từ, nếu số thứ tự của từ lớn hơn hoặc bằng kích thước từ điển thì vector của từ đó sẽ bằng 0.

Đối với từng tập dữ liệu huấn luyện, phát triển và kiểm thử, nhóm thực hiện chuyển đổi dữ liệu văn bản thành các đặc trưng có thể được sử dụng trong mô hình học máy theo các bước như sau:

Bước 1: Sử dụng tokenizer để chuyển đổi mỗi câu trong tập dữ liệu X thành một chuỗi các số nguyên, trong đó mỗi số nguyên tương ứng với một từ trong từ điển của tokenizer, nhằm tạo ra một biểu diễn số hóa của văn bản.

Bước 2: Đảm bảo rằng tất cả các chuỗi số nguyên có độ dài bằng 100. Các chuỗi dài hơn sẽ bị cắt bớt trong khi các chuỗi ngắn hơn sẽ được đệm bằng số 0.

Bước 3: Đối với tập huấn luyện và tập phát triển, nhóm chuyển đổi các nhãn thành biểu diễn one-hot, làm cho mỗi nhãn trở thành một vectơ có chiều dài bằng 3, trong đó chỉ có một phần tử bằng 1 (đại diện cho nhãn của mẫu), và các phần tử khác bằng 0

## **3.3. Cài đặt mô hình**

**Text-CNN**

Mô hình Text-CNN được đào tạo với 40 epochs, kích thước batch là 256, chiều dài chuỗi là 100, và tỷ lệ dropout là 0.5. Mô hình của chúng tôi sử dụng lớp Convolution 1D với 32 bộ lọc và kích thước 2, 3, 5 tương ứng. Nhóm sử dụng bộ tối ưu hóa adam cho TextCNN.

**BERT**

Mô hình BERT-base-multilingual-cased được train với 4 epochs, kích thước batch là 16, độ dài lớn nhất của chuỗi là 100, manual seed = 4, cho phép sử dụng đa luồng. Sau khi train, mô hình được đánh giá bằng confusion matrix, F1-score và accuracy score.

# **4. Kết quả đánh giá**

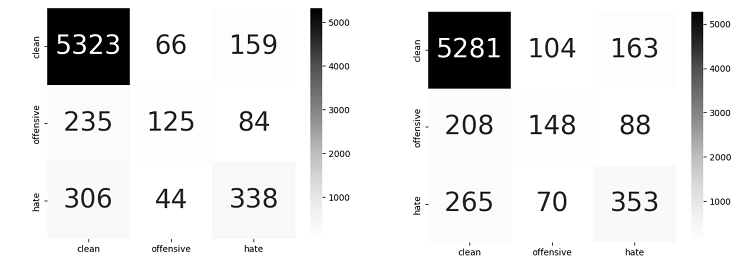
## **4.1. Kết quả đạt được**

Bảng 5 thống kê kết quả đạt được ở hai mô hình huấn luyện trên tập ngữ liệu ViHSD. Kết quả được đo bằng accuracy và F1 - macro. Ta thấy mô hình transformer BERT sử dụng kiến trúc bert-base-multilingual-cased hoạt động tốt hơn mô hình mạng nơron sâu Text-CNN với accuracy và F1-macro lần lượt là 86.56% và 62.25% trên tập ngữ liệu.

|  |  |  |
| --- | --- | --- |
| **Mô hình** | **Accuracy** | **F1 - macro** |
| Text-CNN | 86.61 | 61.13 |
| BERT (bert-base-multilingual-cased) | 86.56 | 62.25 |

*Bảng 5: Kết quả các mô hình trên tập ngữ liệu ViHSD*

Kết quả này chứng tỏ sức mạnh của BERT trong nhiệm vụ phân loại văn bản, đặc biệt là trong việc phát hiện lời nói châm biếm, ngay cả khi chúng được huấn luyện trên nhiều ngôn ngữ khác nhau.



*Hình 24: Confusion matrix của hai mô hình: Bên trái là mô hình Text-CNN, bên phải là mô hình BERT*

Bảng 5 cho ta thấy có khoảng cách lớn giữa giá trị của độ chính xác và điểm F1-score ở cả hai mô hình. Quan sát các confusion matrix, ta thấy đa số bình luận xúc phạm ở cả hai mô hình đều được dự đoán là bình luận sạch (CLEAN), điều này là do sự mất cân bằng dữ liệu như mô tả ở hình 1 mục 1.

## **4.2. Phân tích các trường hợp sai**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Bình luận** | **True** | **Predict** | **Nhận xét** |
| 1 | May quá đéo có tiền mua xe | 0 | 1 | Mẫu chứa từ đặc biệt “đéo”, ta có các vector từ nhúng được đào tạo bằng phương pháp fastText, vì vậy nó chỉ ra mối liên hệ của từ “đéo” đối với các từ trong từ điển nhưng không xét từ đó trong ngữ cảnh câu. |
| 2 | Chưa xem tập nào dơ tay | 0 | 2 | Sau quá trình tiền xử lý, mẫu bình luận sẽ thành “chưa tập dơ”. Mô hình không hoạt động tốt với các từ đồng âm nhưng khác nghĩa như “dơ” trong “dơ tay” (đưa tay ra) và “dơ” trong “dơ bẩn” |
| 3 | Lờ ô lờ ôn lờ ôn lôn huyền CHIM:)) | 0 | 2 | Đây là mẫu bình luận bị gán nhãn sai trong tập ngữ liệu. Căn cứ vào cách gán nhãn, nhãn này nên được gán là 1 nhưng nhìn chung, kết quả phân loại cho thấy mô hình phát hiện khá tốt trong trường hợp này |
| 4 | coin card :3 | 1 | 0 | Mẫu chứa từ đặc biệt là “coin”,“card” và “dell”. Các mẫu từ tiếng lóng này được sử dụng trên mạng xã hội mang ý châm biếm nhưng và không có trong tập huấn luyện nên khả năng cao mô hình dự đoán sai do sự mất cân bằng giữa các lớp. |
| 5 | Trần linh dell hiểu kiểu gì :)) | 1 | 0 |
| 6 | K nói thì tự kiêu,nói thì lại tự ái chứ í thức thế ai mà chịu nổi😡 | 1 | 0 | Mẫu không chứa các từ thô tục nhưng ý nghĩa cả câu đang có hàm ý tấn công, cụm từ “í thức” viết sai chính tả. Do mô hình sẽ học ý nghĩa của các từ nhưng không học theo ngữ cảnh vì vậy dẫn tới kết quả dự đoán sai. |
| 6 | Theo trung cộng đi rồi đến lúc ko còn lợi dụng đc bà sẽ thấy cảnh. Hãy nhớ câu đừng nghe những gì cs nói mà..😂😂😂 | 2 | 0 | Mẫu có chứa các từ “trung cộng” mang ý phân biệt chủng tộc, “Cuong Ly Dinh” và từ thô tục “cc” nhưng mô hình phân loại không xác định được các đối tượng mục tiêu nên cho kết quả phân loại sai |
| 7 | Cuong Ly Dinh chửi cc | 2 | 1 |

*Bảng 6: Phân tích kết quả phân loại sai của mô hình Text-CNN*

Đối với mô hình Text-CNN, do sự mất cân bằng giữa các lớp nên mô hình sẽ có xu hướng dự đoán các từ chưa biết thành nhãn chiếm đa số (CLEAN). Thêm vào đó việc sử dụng các vector từ nhúng theo phương pháp fastText chỉ học được mối quan hệ giữa các từ trong từ điển mà không thể học theo ngữ cảnh dẫn tới các kết quả phân loại sai đối với các từ đồng âm.

## 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Bình luận** | **True** | **Predict** | **Nhận xét** |
| 1 | Đừng cố biện minh =)))) choi lon | 0 | 1 | Mẫu có chứa từ “lon”, “khùng”, “chó” nhưng không có ý quấy rối, đả kích ai nên được gán nhãn sạch nhưng mô hình không phân biệt được |
| 2 | Lý khùng thì có,kk | 0 | 1 |
| 3 | Nguyễn Thị Thủy chó con chứ dúi đâu ra | 0 | 2 |
| 4 | coin card :3 | 1 | 0 | Mẫu chứa từ đặc biệt là “coin”,“card” và “dell”. Các mẫu từ tiếng lóng này được sử dụng trên mạng xã hội mang ý châm biếm nhưng và không có trong tập huấn luyện nên khả năng cao mô hình dự đoán sai do sự mất cân bằng giữa các lớp. |
| 5 | Trần linh dell hiểu kiểu gì :)) | 1 | 0 |
| 6 | Haizz. Nthe này thì dân khổ quá | 1 | 0 | Mẫu có hàm ý chê bai, mỉa mai, tấn công nhưng không chứa các từ ngữ đả kích nên mô hình không nhận biết được |
| 7 | Đề nghị 17, 22,23,32 đến phà vài hơi vào mẹt tk này để nó đi cách ly thế giới hẳn đi. |  |  |
| 8 | K nói thì tự kiêu,nói thì lại tự ái chứ í thức thế ai mà chịu nổi😡 | 1 | 2 | Mẫu không chứa các từ thô tục nhưng ý nghĩa cả câu đang có hàm ý tấn công, cụm từ “í thức” viết sai chính tả. Do mô hình sẽ học ý nghĩa của các từ nhưng không học theo ngữ cảnh vì vậy dẫn tới kết quả dự đoán sai. |
| 9 | Thay Ngu | 2 | 0 | Mẫu có ý nghĩa tấn công, quấy rối nhưng do được viết hoa nên có lẽ mô hình đã nhận định đây là tên riêng nên đã gán nhãn sạch cho mẫu |
| 10 | Chi bà nhiều tiền thì đi làm từ thiện,còn ơ đó nói tục tiểu, giơ bẩn. Làm nhục người vn trên thế giới. Làm bai hoại nền văn hóa vn. Xin yêu các cấp có thẩm quyền vào cuộc sử lý. | 0 | 2 | Mẫu có chứa các từ ngữ thô tục “tục tiểu”, “Làm nhục” nên mô hình nhận định đây là hate speech nhưng đây câu thể hiện quan điểm, mang tính góp ý, xây dựng |

Đối với mô hình BERT, tuy 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. Như những câu có ý nghĩa tấn công nhưng không sử dụng từ ngữ thô tục hoặc ngược lại, những câu có ý nghĩa sạch nhưng lại bị viết sai chính tả khiến mô hình nhầm lẫn.