HATE SPEECH AND OFFENSIVE LANGUAGE DETECTION USING BERT-CNN AND BERT-LSTM

Nguyễn Đình Vũ - 22521692@gm.uit.edu.vn, Nguyễn Thành Vinh - 22521676@gm.uit.edu.vn

Abstract

Trong nghiên cứu này, chúng em áp dụng các mô hình ngôn ngữ tiên tiến vào nhiệm vụ phát hiện hate speech bằng cách sử dụng BERT kết hợp với CNN và BERT kết hợp với LSTM. Các phương pháp kết hợp này nhằm tận dụng sự hiểu biết ngữ cảnh sâu sắc do BERT cung cấp cùng với khả năng trích xuất đặc trưng mạnh mẽ của CNN và sức mạnh mô hình hóa tuần tự của LSTM. Bằng cách tích hợp các mô hình này, chúng em có thể giải quyết các thách thức trong việc phát hiện hate speech trong dữ liệu văn bản đa dạng và phức tạp. Các phương pháp đề xuất được kỳ vọng sẽ đóng góp vào việc tạo ra môi trường trực tuyến an toàn hơn.

1 Giới thiệu

"Twitter" thường hay được sử dụng để truyền tải ngôn từ gây kích động thù hận, đặc biệt là với khả năng ẩn danh mà nền tảng này cung cấp. Thế nên, bất kỳ chiến lược nào dùng để xác định nội dung như thế đều rất quan trọng trong thế giới hiện đại, để giữ cho mạng xã hội trở thành một môi trường an toàn. Việc phát hiện nội dung gây kích động thù hận là bước đầu trong việc phát triển hệ thống có thể "đánh dấu" những mục này và có những hành động đúng đắn. Những người có công việc "chú thích" những nội dung như vậy được các công ty mạng sử dụng để xóa bớt những mẫu này và người dùng có thể "đánh dấu" bất kì thứ gì họ cho là có hại cho công chúng. Thế nhưng, các quy trình này lại tốn rất nhiều thời gian, chi phí và lại phụ thuộc vào sự đánh giá từ con người. Do đó, các phương pháp phát hiện ngôn từ gây kích động thù hận "tự động" đã trở thành "mối quan tâm" lớn trong thời đại ngày nay. Nhằm đạt được mục đích trên, trong các nghiên cứu ban đầu, nhiều người đã nỗ lực xây dựng dựa trên các thuật toán học máy (machine learning algorithms) và các kĩ thuật trích xuất đặc trưng khác nhau, nhưng trong những năm gần đây, bằng cách sử dụng các mô hình ngôn ngữ dựa trên Transformers (Transformers based Language Models), hay BERT, nhiều thành tựu đáng nể đã gặt hái trong mảng xử lý ngôn ngữ tự nhiên (NLP). Hơn thế nữa, các phương pháp mạng nơ ron cũng đã giảm bớt áp lực trong việc tạo đặc trưng được triển khai kết hợp với biểu diễn của các văn bản (text) dưới dạng "word vector" thông qua các model word embedding. Các mô hình máy học (ML) và mạng nơ-ron (NN) thường yêu cầu phải có một bộ dataset lớn để huấn luyện cho hiệu quả, ngược lại, các mô hình dựa trên BERT đôi khi có thể hoạt động với một lượng nhỏ các dữ liệu gán nhãn. Thế nên, để tận dung được lợi thế của cả 2 phương pháp, chúng em đã tích hợp lại để xây dựng một kiến trúc sâu hơn, nhằm phát hiện ngôn từ kích động thù hận trong bộ dữ liệu cơ sở của Twitter một cách hiệu quả hơn.

2 Mô tả bài toán

Mô tả bài toán: Từ một đoạn tweet (comment trên MXH), đưa ra nhận định rằng đoạn tweet đó có chứa ngôn ngữ thù ghét hay xúc phạm hay không. Input: 1 đoạn tweet (comment trên MXH), không quá dài. Output: Trả về 1 trong 3 nhãn: Thù ghét (Hate), Xúc phạm (Offensive) hoặc Bình Thường (Neither).

3 Phương pháp

3.1 Dataset

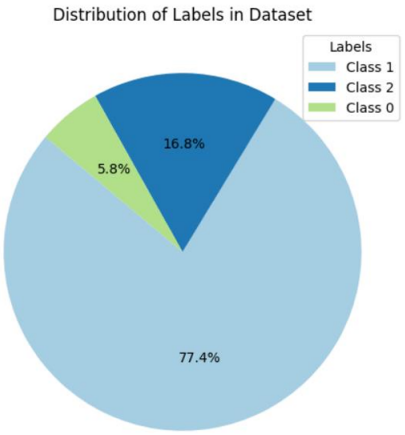
3.1.1 Giải thích về các nhãn

- **Phát ngôn thù địch (HATE)**: Chứa ngôn ngữ lạm dụng, nhằm mục đích xúc phạm cá nhân hoặc nhóm người với những lời lẽ thù địch, xúc phạm và hạ thấp. Một bài viết hoặc bình luận được xác định là HATE nếu nó:
  - Nhắm vào cá nhân hoặc nhóm người dựa trên đặc điểm của họ (màu da, chủng tộc,...)
  - Thể hiện rõ ý định gây hại hoặc kích động hận thù.
  - Có thể hoặc không sử dụng các từ xúc phạm hoặc lời lẽ tục tĩu.
- **Ngôn ngữ xúc phạm (OFFENSIVE)**: Là một bài viết hoặc bình luận có thể chứa các từ ngữ xúc phạm, nhưng không nhắm vào cá nhân hoặc nhóm người nào.
- **Không xúc phạm cũng không phải phát ngôn thù địch (NEITHER)**: Là bài viết hoặc bình luận bình thường. Nó là cuộc trò chuyện, thể hiện cảm xúc một cách bình thường và không chứa ngôn ngữ xúc phạm hoặc thù địch.

3.1.2 Một số thông tin về dataset

Davidson dataset: Được tạo bởi Davidson và cộng sự (2017). Bao gồm 24,783 tweet và 3 nhãn (thù địch, xúc phạm, và không phải cả hai), được tạo bằng cách sử dụng nền tảng Figure Eight crowdsourcing. Những tweet này được chọn từ 85.4 triệu tweet lưu trữ, tập trung vào các từ khóa HateBase (hatebase.org), và được chú thích bởi 3 người.

3.1.3 Phân bố nhãn trong dataset



Qua hình trên, ta có thể nhận thấy, bộ dữ liệu này có vẻ không thực sự quá đa dạng, phân bố giữa các lớp không thực sự "cân bằng": dữ liệu của lớp 1 gần như "áp đảo" so với 2 lớp còn lại. Thế nên, để tránh ảnh hưởng đến việc mô hình gặp khó khăn trong việc nhận diện và phân loại 2 lớp còn lại, chúng em đã áp dụng một số phương pháp tăng cường dữ liệu (data augmentation) cho bộ dataset trên, cụ thể là:

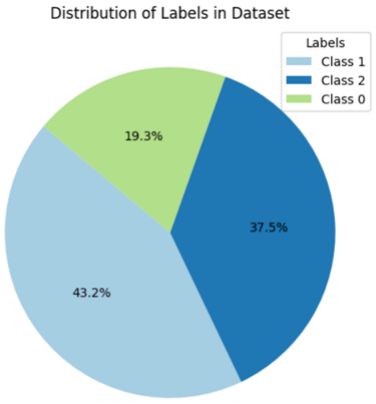
Thay thế từ đồng nghĩa (Synonyms replacement):

- **Cơ chế**: Phương pháp này thay thế một số từ trong câu gốc bằng các từ đồng nghĩa của chúng. Ví dụ, trong câu ”The cat is sitting on the mat”, từ ”sitting” có thể được thay thế bằng ”resting”, tạo thành câu ”The cat is resting on the mat”.
- **Tác dụng**: Việc thay thế từ đồng nghĩa giúp tạo ra nhiều phiên bản khác nhau của cùng một câu, làm tăng sự đa dạng của dữ liệu huấn luyện mà không làm thay đổi ý nghĩa cơ bản của câu. Điều này giúp mô hình học được các ngữ cảnh khác nhau mà từ đồng nghĩa có thể xuất hiện, từ đó cải thiện khả năng tổng quát hóa của mô hình.

Hoán đổi ngẫu nhiên (Random swap):

- **Cơ chế**: Phương pháp này hoán đổi vị trí của các từ trong câu một cách ngẫu nhiên. Ví dụ, câu ”The cat is sitting on the mat” có thể được hoán đổi thành ”On the mat is sitting the cat”.
- **Tác dụng**: Việc hoán đổi từ ngẫu nhiên giúp mô hình tiếp xúc với các cấu trúc câu đa dạng hơn, giúp mô hình trở nên linh hoạt hơn trong việc xử lý các câu có cấu trúc không quen thuộc. Điều này đặc biệt hữu ích trong các ngữ cảnh mà trật tự từ có thể thay đổi mà không ảnh hưởng đến ý nghĩa chính của câu.

Dataset sau khi được áp dụng 2 phương pháp trên



Do nhận thấy tỉ lệ mẫu vẫn bị lệch giữa lớp 0 (Hate speech) so với 2 lớn còn lại, thế nên chúng em đã quyết định “đánh trọng số” (weightclass) cho từng lớp, tránh việc mô hình sẽ “thiên vị” 2 lớp có số lượng mẫu lớn hơn. Với công thức:

$$\text{weight}_{\text{class}} = \frac{n_{\text{samples}}}{\text{number of classes} \times n_{\text{class}}}$$

Trong đó:

- n_{samples}: Tổng số mẫu trong tập dữ liệu.
- n_{class}: Số mẫu trong lớp hiện tại.
- Số lớp: Tổng số lớp trong bài toán phân loại.
- *Kết quả*:
  - Weight for class 0: 1.73
  - Weight for class 1: 0.77
  - Weight for class 2: 0.89

3.2 Tiền xử lý

3.2.1 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu (data preprocessing) là một bước cực kỳ quan trọng trong quá trình phát hiện ngôn từ kích động thù hận và ngôn ngữ xúc phạm. Bởi lẽ, các dữ liệu trên MXH thường chứa nhiều thành phần không liên quan như

URLs, hashtag, biểu tượng cảm xúc (emojis) hoặc khó hơn là từ viết tắt, đôi khi các dữ liệu này có thể không tuân theo các quy tắc ngữ pháp hay viết đúng chính tả. Thế nên, việc tiền xử lý sẽ giúp loại bỏ những yếu tố này, giúp dữ liệu trở nên đồng nhất, sạch sẽ và dễ dàng xử lý hơn.

- **Các kĩ thuật được tiền xử lý dữ liệu được sử dụng**:
  - Lowercase (Chuyển tất cả chữ cái về dạng chữ thường).
  - Remove URLs (Xóa URLs).
  - Remove HTML tags (Xóa tag HTML).
  - Remove punctuations (Xóa dấu câu).
  - Remove newline characters (Xóa xuống dòng).
  - Remove words containing digits (Xóa từ có dính số).
  - Remove stopwords (Xóa các từ không mang nhiều ý nghĩa).
  - Apply stemming (Đưa từ về dạng gốc/ cơ bản nhất).

Một số ví dụ sau khi tiền xử lý dữ liệu:

```
Data gốc:
"His big ass smile, his eyes is chink."

Data được EDA:
1) His big ass grinning his eyes is chink -> synonyms replacement

2) His big ass smile his is eyes chink -> random swap


Data gốc:
@EakaErick I wouldnt share a wave with those trash cans

Data được EDA:

1) @EakaErick I wouldnt share a flourish with those trash cans
-> synonyms replacement

2) @EakaErick I wouldnt share a with wave those trash cans
-> random swap
```

3.2.2 Wordclouds

Định nghĩa: WORDCLOUD (hay còn gọi là đám mây từ) là một hình ảnh trực quan thể hiện tần suất xuất hiện của các từ trong một văn bản. Trong một wordcloud, các từ xuất hiện "thường xuyên nhất" sẽ được hiển thị "lớn hơn và nổi bật hơn" so với những từ ít xuất hiện hơn. Wordcloud thường được sử dụng để nhận dạng nhanh các từ khóa, chủ đề hoặc xu hướng chính trong một tập hợp văn bản lớn.

Wordcloud của lớp HateSpeech

Wordcloud của lớp Offensive

Wordcloud của lớp Neither

3.3 Xây dựng các kiến trúc kết hợp

3.3.1 Bert là gì?

BERT (Bidirectional Encoder Representation from Transformer) là mô hình biểu diễn từ theo 2 chiều, sử dụng kỹ thuật Transformer. BERT được thiết kế để huấn luyện trước các biểu diễn từ nhúng (pre-train word embedding). Điểm đặc biệt ở BERT đó là có thể điều hòa cân bằng ngữ cảnh theo cả 2 chiều trái và phải. Không như những kỹ thuật khác, BERT không sử dụng LSTM (Long short-term memory) để trích xuất đặc trưng ngữ cảnh của từ, mà thay vào đó sử dụng Transformer, là cơ chế dựa trên sự tập

trung (attention) chứ không dựa trên sự lặp lại. So với các mô hình dựa trên huấn luyện theo 2 chiều riêng biệt trái, phải như ELMO, BERT có thể trích xuất nhiều đặc trưng ngữ cảnh hơn từ cùng một văn bản. Nhờ vào khả năng xử lý một cách hiệu quả sự không tường minh (ngữ nghĩa tiềm ẩn), và đây chính là điểm khác biệt nhất của việc hiểu 10 ngôn ngữ tự nhiên, BERT có thể đạt độ chính xác cao trong việc phân tích các ngôn ngữ gần gũi với con người. Các nghiên cứu đã cho thấy, nhờ sử dụng một lượng lớn dữ liệu văn bản huấn luyện, BERT đã đạt được kết quả tối ưu tốt nhất trên 11 tác vụ xử lý ngôn ngữ tự nhiên (NLP). Quá trình huấn luyện trước bên trái và bên phải của BERT đạt được bằng cách sử dụng mặt nạ mô hình ngôn ngữ đã được sửa đổi, gọi là masked language model (MLM). Mục đích của MLM là che giấu một từ ngẫu nhiên trong một câu với xác suất nhỏ. Khi mô hình che một từ, nó sẽ thay thế từ đó bằng một token [MASK]. Sau đó, mô hình cố gắng dự đoán từ bị che bằng cách sử dụng ngữ cảnh của cả bên trái và phải của từ bị che, với sự hỗ trợ của transformer. Ngoài việc trích xuất ngữ cảnh trái và phải bằng MLM, BERT có thêm một mục tiêu quan trọng khác so với các nghiên cứu trước, đó là dự đoán câu tiếp theo (Next Sentence Prediction).

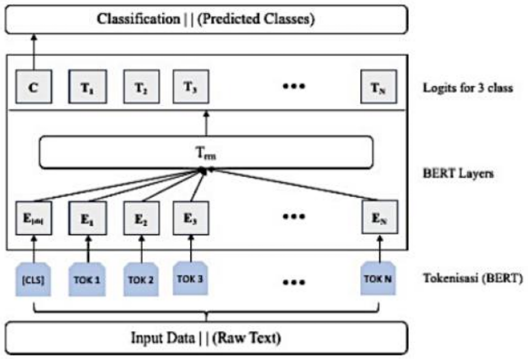
- **Biểu diễn đầu vào**: Văn bản đầu vào của mô hình BERT trước tiên được xử lý thông qua một phương pháp được gọi là tách từ wordpiece. Kết quả sẽ tạo ra một tập các token, mỗi token đại diện cho một từ. Chuỗi đầu vào BERT biểu diễn một cách tường minh cho cả dạng văn bản đơn và cặp văn bản. Với văn bản đơn, chuỗi đầu vào BERT là sự ghép nối của token đặc biệt “”, token của chuỗi văn bản, và token phân tách “”. Trạng thái ẩn cuối cùng tương ứng với token phân loại [CLS] được sử dụng làm véc-tơ biểu diễn tổng hợp cho các nhiệm vụ phân loại. Với cặp văn bản, chuỗi đầu vào BERT là sự ghép nối của “”, token của chuỗi văn bản đầu, “”, token của chuỗi văn bản thứ hai, và “”. Nhờ vậy, BERT có thể được sử dụng để so sánh một cặp hai câu. Tập hợp các token của cặp câu này sau đó được xử lý thông qua ba lớp nhúng khác nhau có cùng kích thước, rồi được cộng lại với nhau và chuyển đến lớp mã hóa (encoder layer). Ba lớp nhúng đó là: lớp nhúng token (token embedding layer), lớp nhúng đoạn (segment embedding layer) và lớp nhúng vị trí (position embedding layer).

- **Transformer**: Các nghiên cứu về mô hình hóa chuỗi trước đây sử dụng một kiểu kiến trúc chung là seq2seq, dựa trên các kỹ thuật như RNN và LSTM. Kiến trúc transformer không dựa trên RNN mà dựa trên kỹ thuật tập trung (attention). Nó quyết định những chuỗi nào là quan trọng trong mỗi bước tính toán. Bộ mã hóa không chỉ ánh xạ đầu vào thành véc-tơ không gian nhiều chiều hơn mà còn sử dụng các từ khóa quan trọng làm đầu vào bổ sung cho bộ giải mã. Nhờ vậy có thể cải thiện bộ giải mã vì có thêm thông tin bổ sung về chuỗi quan trọng và từ khóa cung cấp ngữ cảnh cho câu.

- **Mô hình ngôn ngữ**: Việc xây dựng một mô hình ngôn ngữ thông thường cần sử dụng một lượng lớn dữ liệu văn bản không chú thích, được gọi là huấn luyện trước (pre-training). Mục đích chung của mô hình ngôn ngữ là học biểu diễn theo ngữ cảnh của từ. Mô hình ngôn ngữ là thành phần quan trọng trong việc giải quyết các bài toán NLP, tìm hiểu sự xuất hiện của từ và các mẫu dự đoán từ dựa trên dữ liệu văn bản không có chú thích. Mô hình ngôn ngữ học ngữ cảnh bằng cách sử dụng các kỹ thuật như nhúng từ, tức là sử dụng véc-tơ để đại diện cho các từ trong không gian véc-tơ. Dựa trên lượng lớn dữ liệu huấn luyện, mô hình

ngôn ngữ học các biểu diễn của các từ tùy thuộc vào ngữ cảnh, nhờ đó cho phép các từ tương tự nhau có biểu diễn tương tự

- **Masked Language Model**: BERT sử dụng token mặt nạ [MASK] để huấn luyện trước các biểu diễn sâu hai chiều cho mô hình ngôn ngữ. Trái ngược với các mô hình ngôn ngữ khác thực hiện huấn luyện từ trái sang phải hoặc từ phải sang trái để dự đoán các từ, và từ được dự đoán được đặt ở cuối hoặc ở đầu chuỗi văn bản, thì BERT che dấu một từ ngẫu nhiên trong chuỗi. Cụ thể hơn, BERT sử dụng token mặt nạ [MASK] để huấn luyện trước các biểu diễn sâu hai chiều cho mô hình ngôn ngữ. Trái ngược với các mô hình ngôn ngữ khác thực hiện huấn luyện từ trái sang phải hoặc từ phải sang trái để dự đoán các từ, và từ được dự đoán được đặt ở cuối hoặc ở đầu chuỗi văn bản, thì BERT che dấu một từ ngẫu nhiên trong chuỗi.
  Cụ thể hơn, văn bản đầu vào được chia thành các tokens bằng bộ token hóa BERT, với các tokens đặc biệt như [CLS] và [SEP] được thêm vào. Khoảng 15% các tokens được chọn ngẫu nhiên để che đi (mask) bằng token [MASK]. Các tokens này sau đó được chuyển thành embeddings (biểu diễn vector), bao gồm thông tin về vị trí và loại token. Các embeddings này được đưa qua một chuỗi các lớp transformer, nơi cơ chế attention giúp mô hình hiểu được ngữ cảnh của từng token. Sau đó, mô hình dự đoán các token ban đầu của mỗi token bị che đi dựa trên ngữ cảnh xung quanh. Loss được tính toán từ sự khác biệt giữa token dự đoán và token thực tế, được sử dụng để điều chỉnh trọng số của mô hình.
  Ví dụ, với câu ”The quick brown fox jumps over the lazy dog”, một số tokens được che đi thành ”The quick [MASK] fox jumps over [MASK] lazy dog”. Mục tiêu của mô hình là dự đoán từ ”brown” và ”the” dựa trên ngữ cảnh xung quanh, giúp mô hình BERT nắm bắt ngữ cảnh và ngữ nghĩa của văn bản một cách sâu sắc và hiệu quả.
- **Dự đoán câu tiếp theo (NSP)**: Dự đoán câu tiếp theo được sử dụng để hiểu mối quan hệ giữa hai câu văn bản. BERT đã được huấn luyện trước để dự đoán liệu có tồn tại mối quan hệ giữa hai câu hay không. Mỗi câu có kích thước nhúng riêng.



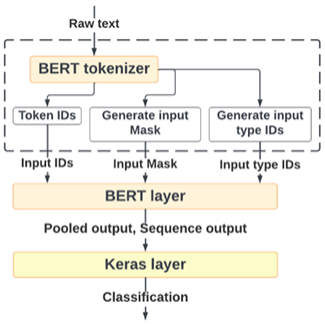
Cách hoạt động của BERT cho bài toán phân loại : Trong lớp đầu tiên là lớp đầu vào. Quá trình bắt đầu với dữ liệu văn bản thô từ các tweet cần được phân loại. TOK1, TOK2, TOK3, TOK4, ... TOKN đại diện cho quá trình tạo token, liên quan đến việc chuyển đổi văn bản thô thành đại diện số mà mô hình có thể xử lý. Văn bản được chuyển đổi thành các mã định danh token (token IDs) bằng cách sử dụng bộ token hóa BERT. Các token [CLS] và [SEP] được bộ token hóa tự động thêm vào nhưng không hiển thị rõ ràng. Kết quả của quá trình này là một tập hợp các token ids sẵn sàng cho xử lý tiếp theo. E1, E2, E3, E4, ... EN đại diện cho quá trình tạo vector, nơi các mã định danh token được chuyển đổi thành các biểu diễn vector bằng cách sử dụng ma trận tạo vector BERT. Mô hình BERT sử dụng bao gồm một lớp tạo vector chuyển đổi các token ids thành các biểu diễn vector có ý nghĩa ngữ nghĩa.
Kiến trúc mô hình: với các vector đã được

chuẩn bị, dữ liệu được đưa vào kiến trúc mô hình BERT, được điều chỉnh cho các nhiệm vụ phân loại.TFBertForSequenceClassification là một mô hình BERT với một lớp bổ sung cho phân loại. Mô hình này bao gồm một số lớp transformer xử lý các token embeddings và tạo ra các biểu diễn trừu tượng hơn của đầu vào. Đầu ra từ lớp này là logits, là các điểm số thô chỉ ra khả năng đầu vào thuộc về mỗi lớp đã được định nghĩa. Đầu ra: Là kết quả từ mô hình, hiển thị các điểm số chưa chuẩn hóa cho mỗi lớp. Mỗi điểm số đại diện cho sự tự tin của mô hình rằng đầu vào thuộc về một lớp cụ thể nào đó. Phân loại (Các lớp được dự đoán) là kết quả cuối cùng thu được. Lớp có xác suất cao nhất là lớp được mô hình dự đoán cho đầu vào đó. Quá trình này cung cấp đầu ra cuối cùng dưới dạng nhãn lớp, chỉ định thể loại hoặc lớp của tweet dựa trên phân tích được thực hiện bởi mô hình BERT.

3.3.2 Tổng quan

Khi được huấn luyện trên lượng lớn data, BERT chắc chắn sẽ cho hiệu suất rất tốt, nó cũng trả về các vector và ngữ cảnh nhúng (contextual embedding) khác nhau cho cùng một từ, điều đó giúp trích xuất nhiều thông tin hơn từ văn bản. Mặt khác, DeepLearning lại có lợi thế cho mảng xử lý ngôn ngữ tự nhiên (NLP), trong đó CNN và RNN thường được sử dụng cho phân loại văn bản (text classification). Thế nên, chúng em đã triển khai 2 mô hình kết hợp, giữa BERT với các mô hình phổ biến như CNN và LSTM.

3.3.3 Pipeline



(Kiến trúc tổng quát: BERT + Deep Neural Networks)

Ban đầu, chúng em đánh giá thông tin ngữ cảnh có được từ BERT, tinh chỉnh (fine-tuning) bằng cách sử dụng các bộ dữ liệu của mình để có được các biểu diễn ngữ cảnh của nó và sau đó, tích hợp mô hình với một số kỹ thuật ensemble learning: tổng hợp (aggregation) và xếp chồng (stacking), nhằm cải thiện hiệu suất và độ bền vững, nhằm được phân loại tốt hơn. Dữ liệu văn bản cần được chuyển đổi thành các "token ids" và sau đó được sắp xếp thành các Tensors trước khi được đưa vào mô hình BERT. Tại đây, TensorFlow Hub cung cấp một bộ tiền xử lý BERT (BERT-processor) phù hợp (cụ thể là tokenizer) cho mỗi mô hình BERT, thực hiện việc chuyển đổi này bằng cách sử dụng thư viện TensorFlow. Ngoài ra, chúng em còn sử dụng BERT TensorFlow Hub để tính toán các biểu diễn không gian vector của các bộ dữ liệu, nhằm triển khai 2 mô hình học sâu khác nhau.

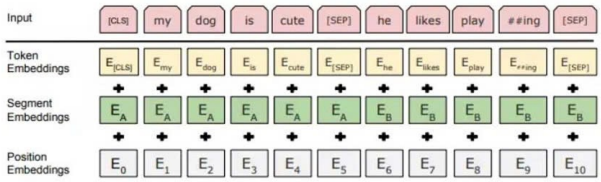
3.3.4 Mô hình

Mô hình BERT mà chúng em sử dụng cho cả 2 mô hình sắp tới là "BERT uncased L-2 H-128 A-2 model": 2 hidden layers (L=2), 128 hidden size (H=128).

Giới thiệu sơ lược về mô hình

- Mô hình BERT được điều chỉnh để hoạt động như một lớp embedding từ để trích

xuất thông tin từ dữ liệu. Mô hình BERT được chọn vì nó vượt trội so với các phương pháp mô hình ngôn ngữ đơn ngữ và đa ngữ được tiền huấn luyện trước đó, đạt hiệu suất mới tiên tiến nhất trên nhiều nhiệm vụ NLP. Kiến trúc của mô hình BERT là một kiến trúc nhiều lớp bao gồm nhiều lớp bộ mã hóa Bidirectional Transformer. Nó nhận biểu diễn của một câu văn bản bao gồm một chuỗi từ ngữ cảnh làm đầu vào. Biểu diễn đầu vào của mô hình BERT được xây dựng bằng cách cộng các token đó với các vector phân đoạn và các vị trí tương ứng của các từ trong chuỗi.



Quá trình biểu diễn đầu vào của mô hình BERT

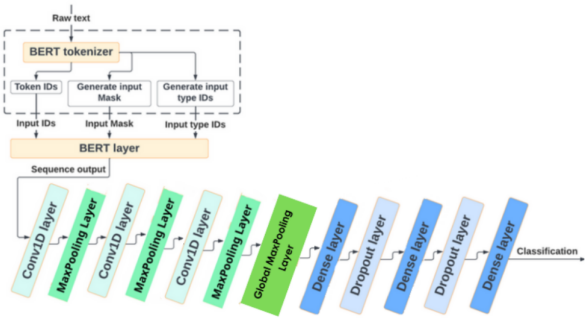
- Sử dụng Embedding Vị trí với độ dài câu tối đa là 20 từ.
- Từ đầu tiên của mỗi chuỗi mặc định là từ đặc biệt [CLS]. Đầu ra của trạng thái ẩn cuối cùng tương ứng với từ [CLS] sẽ được sử dụng để đại diện cho toàn bộ câu trong phân loại.
- Khi một chuỗi chỉ bao gồm một câu duy nhất, embedding phân đoạn có thể được áp dụng trực tiếp cho câu đó.
- Trong trường hợp chuỗi chứa nhiều hơn hai câu, phân biệt các câu trong hai bước: tách các câu bằng từ đặc biệt gọi là [SEP] và thêm embedding phân đoạn độc lập cho mỗi câu.
- Tiếp theo, lớp Fully-connected ở cuối mô hình BERT tiền huấn luyện được thay thế bằng một kiến trúc mạng CNN. Vì CNN hiện là mô hình thành công nhất trong việc giải quyết các nhiệm vụ phân loại văn bản ngắn[10], nó được sử dụng thay vì các mạng neural sâu điển hình khác như LSTM, Bi-LSTM và GRU.

Mô hình BERT+CNN

- **Lý do BERT+CNN**: BERT hoạt động bằng cách tạo ra các trạng thái ẩn (hidden states) cho mỗi từ trong câu dựa trên ngữ cảnh của các từ xung quanh. Các trạng thái ẩn này chứa thông tin ngữ cảnh phong phú về mỗi từ, giúp mô hình hiểu ngữ nghĩa và quan hệ giữa các từ trong câu. Khi sử dụng CNN dựa trên hidden states của BERT, ta có thể tận dụng các tính năng ngữ cảnh phong phú mà BERT đã học được để trích xuất các mẫu cục bộ quan trọng, CNN đặc biệt mạnh mẽ trong việc phát hiện các mẫu cục bộ trong dữ liệu, nhờ vào việc sử dụng các bộ lọc (kernels). Các bộ lọc của CNN có thể phát hiện các cụm từ quan trọng hoặc các chuỗi từ có ý nghĩa đặc biệt trong ngữ cảnh của câu, nhận diện các n-grams hữu ích giúp mô hình hiểu rõ hơn các cấu trúc từ và cách chúng đóng góp vào ngữ nghĩa của câu. Đồng thời, các bộ lọc của CNN cũng học cách tách biệt các thông tin chính và thông tin phụ, giúp mô hình tập trung vào các đặc điểm quan trọng hơn của dữ liệu. Các bộ lọc trượt qua các hidden states được tạo ra bởi BERT, thực hiện phép nhân tích chập và tạo ra các feature maps. Sau đó, các feature maps được đưa qua hàm kích hoạt (activation function) để tạo ra các giá trị phi tuyến tính. Cuối cùng, lớp pooling có thể được sử dụng để giảm kích thước của các feature maps và tập trung vào các thông tin quan trọng nhất.
  Các kỹ thuật tích chập và pooling của CNN giúp trích xuất các khái niệm chính

và từ khóa của văn bản dưới dạng các đặc trưng, dẫn đến việc cải thiện đáng kể hiệu suất của mô hình phân loại. Tuy nhiên, mạng CNN có một hạn chế đáng kể là không phù hợp cho văn bản dạng chuỗi. Để giải quyết hạn chế này, mô hình ngôn ngữ đơn ngữ lớn được tiền huấn luyện BERT là sự kết hợp phù hợp do BERT có nhiệm vụ trích xuất các đặc trưng từ câu để làm đầu vào cho mô hình Text-CNN. Tiếp theo, embedding từ ngữ cảnh của các bình luận từ BERT được đưa vào mô hình Text-CNN để lấy các bản đồ đặc trưng. Cuối cùng, các nhãn dự đoán được đưa ra thông qua lớp softmax.

- **Cấu trúc mô hình**



- **Mô tả mô hình:**

  – **Input Layer**: Mô hình nhận dữ liệu đầu vào dưới dạng chuỗi văn bản. Dữ liệu này sau đó được chuyển đến lớp tiền xử lý.

  – **Preprocessing Layer**: Sử dụng mô hình tiền xử lý BERT từ TensorFlow Hub để chuyển đổi văn bản đầu vào thành các định dạng phù hợp cho mô hình BERT, bao gồm việc tạo ra các token ids, các nhúng vị trí và các nhúng đoạn.

  – **BERT Encoder Layer**: Văn bản đã được tiền xử lý sau đó được đưa vào mô hình BERT để mã hóa ngữ cảnh. Đầu ra của BERT bao gồm các biểu diễn ngữ cảnh của từng token.

  – **Sequence Output Extraction**: Đầu ra chuỗi từ BERT được trích xuất để làm đầu vào cho các lớp CNN tiếp theo.

  – **CNN Layer 1**: Thực hiện tích chập trên đầu vào với 64 bộ lọc, kích thước kernel là 3, sử dụng hàm kích hoạt ReLU và padding 'same'. Dùng để phát hiện các đặc trưng cơ bản của dữ liệu, như các từ hoặc cụm từ ngắn.

  – **MaxPooling Layer 1**: Thực hiện MaxPooling với kích thước pool là 2, giúp giảm kích thước dữ liệu và giữ lại các đặc trưng quan trọng.

  – **CNN Layer 2**: Thực hiện tích chập với 128 bộ lọc, kích thước kernel là 3, sử dụng hàm kích hoạt ReLU và padding 'same'. Nắm bắt các đặc trưng phức tạp như ngữ cảnh hoặc cấu trúc câu phức tạp.

  – **MaxPooling Layer 2**: Thực hiện MaxPooling với kích thước pool là 2.

  – **CNN Layer 3**: Thực hiện tích chập với 256 bộ lọc, kích thước kernel là 3, sử dụng hàm kích hoạt ReLU và padding 'same'. Phát hiện các đặc trưng cao cấp và tổng quát hóa từ các lớp trước, giúp mô hình hiểu sâu hơn về ngữ cảnh và ý nghĩa.

  – **MaxPooling Layer 3**: Thực hiện MaxPooling với kích thước pool là 2.

  – **Global MaxPooling Layer**: Tầng Global MaxPooling để lấy giá trị cực đại toàn cầu từ các đầu ra của lớp CNN, giúp giảm số lượng các tham số cần huấn luyện và giữ lại các đặc trưng quan trọng nhất.

– **Dense Layer 1**: Một lớp Dense với 512 đơn vị và kích hoạt ReLU để tăng cường khả năng học của mô hình.

– **Dropout Layer 1**: Một lớp Dropout với tỉ lệ 0.3 để tránh hiện tượng overfitting.

– **Dense Layer 2**: Một lớp Dense với 256 đơn vị và kích hoạt ReLU.

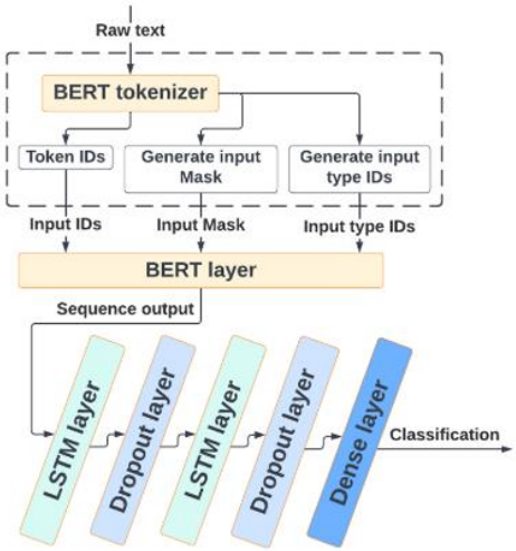
– **Dropout Layer 2**: Một lớp Dropout với tỉ lệ 0.3.

– **Final Dense Layer**: Lớp phân loại cuối cùng với 3 đơn vị, sử dụng hàm kích hoạt Softmax để phân loại văn bản thành ba loại ngôn từ (ngôn từ kích động thù hận, ngôn từ xúc phạm và không phải cả hai).

Mô hình BERT+LSTM

- **Lý do BERT+LSTM**: BERT có khả năng xuất sắc trong việc hiểu ngữ cảnh và mối quan hệ giữa các từ, điều này rất quan trọng để phát hiện ngôn từ kích động thù hận vì ý nghĩa của từ có thể thay đổi dựa trên ngữ cảnh. LSTM (Long Short-Term Memory) mạnh mẽ trong việc xử lý và ghi nhớ thông tin tuần tự, giúp nắm bắt các phụ thuộc dài hạn trong văn bản. Sự kết hợp giữa BERT và LSTM tận dụng ưu thế của cả hai mô hình: BERT mã hóa ngữ cảnh tốt còn LSTM xử lý thông tin tuần tự hiệu quả, làm tăng độ chính xác và khả năng tổng quát hóa của mô hình. Điều này giúp mô hình phát hiện ngôn từ kích động thù hận hiệu quả hơn, đặc biệt khi xử lý các văn bản dài và phức tạp. BERT + LSTM còn giúp mô hình học được các đặc trưng ngữ cảnh và tuần tự từ các bộ dữ liệu huấn luyện, cải thiện khả năng tổng quát hóa khi làm việc với dữ liệu mới, đồng thời ngăn ngừa hiện tượng overfitting và tăng cường khả năng phân loại chính xác.

- **Cấu trúc mô hình**:



- **Mô tả mô hình**:

– **Input Layer**: Mô hình nhận dữ liệu đầu vào dưới dạng chuỗi văn bản. Dữ liệu này sau đó được chuyển đến lớp tiền xửlý.

– **Preprocessing Layer**: Sử dụng mô hình tiền xử lý từ TensorFlow Hub để chuyển đổi văn bản đầu vào thành các token id, mặt nạ đầu vào (input mask) và mã từ (input word id) phù hợp cho BERT.

– **BERT Encoder Layer**: Văn bản đã được tiền xử lý sau đó được đưa vào mô hình BERT từ TensorFlow Hub để mã hóa ngữ cảnh. Đầu ra của BERT bao gồm các biểu diễn ngữ cảnh của từng token và đầu ra tổng hợp.

– **Sequence Output Extraction**: Đầu ra chuỗi từ BERT được trích xuất để làm đầu vào cho các lớp CNN tiếp theo.

– **LSTM Layer 1**: Sử dụng LSTM đơn hướng (unidirectional) với 128 đơn vị để nắm bắt các đặc trưng từ

chuỗi văn bản. Lớp này giúp mô hình học được các mối quan hệ giữa các từ trong văn bản từ hướng trái sang phải.

- **Dropout Layer 1**: Sử dụng lớp Dropout với tỉ lệ 0.1 để ngăn ngừa hiện tượng overfitting bằng cách loại bỏ ngẫu nhiên một số đơn vị (neurons) trong quá trình huấn luyện. Điều này giúp cải thiện khả năng tổng quát hóa của mô hình.
- **LSTM Layer 2**: Một lớp LSTM đơn hướng thứ hai với 128 đơn vị được sử dụng để tăng cường khả năng học ngữ cảnh của mô hình từ chuỗi văn bản. Lớp này không trả về chuỗi kết quả, mà chỉ trả về trạng thái cuối cùng của LSTM, đại diện cho toàn bộ ngữ cảnh của chuỗi.
- **Dropout Layer 2**: Một lớp Dropout thứ hai tiếp tục được áp dụng để giảm nguy cơ overfitting, đảm bảo mô hình hoạt động ổn định trên dữ liệu chưa thấy.
- **Classifier (Dense Layer)**: Lớp phân loại cuối cùng với 3 đơn vị, sử dụng hàm kích hoạt Softmax để phân loại văn bản thành ba loại ngôn từ (ngôn từ kích động thù hận, ngôn từ xúc phạm và không phải cả hai).

3.4 Thực nghiệm

Bộ dataset chúng em chia theo tỉ lệ 8:1:1 (0.8 cho tập train, 0.1 cho tập validation và 0.1 cho tập test). Chúng em còn tạo 1 bộ tối ưu hóa tùy chỉnh (customized optimizer), với số epoch = 20, learning rate = 2e-5 và thử nghiệm với bộ optimizer AdamW, cùng hàm loss là "Cross-entropy".

Công thức:

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

Trong đó: M là số lượng các lớp (HATE, OFFENSIVE, NEITHER), log là log tự nhiên, y là chỉ báo nhị phân (0 hoặc 1) nếu nhãn lớp c là phân loại đúng cho quan sát o, p là xác suất dự đoán rằng quan sát o thuộc lớp c.

Để sử dụng các đặc trưng từ mỗi lớp, chúng em đã giới thiệu trọng số trước đó. Do đó, hiệu suất của các bộ phân loại được đo lường thông qua điểm F1-Score, Accuracy, Precision và Recall.

Các độ đo:
Accuracy:

$$\text{Accuracy} = \frac{1}{3}\sum_{i=1}^{3} \frac{tp_i + tn_i}{tp_i + fp_i + tn_i + fn_i}$$

Precision: Tỷ lệ dự đoán đúng trong số các dự đoán dương tính. Độ đo này quan trọng khi chi phí của các dự đoán sai dương tính (false positives) cao.

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

Recall: Tỷ lệ dự đoán đúng trong số các trường hợp dương tính thực sự. Độ đo này quan trọng khi chi phí của các dự đoán sai âm tính (false negatives) cao.

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

F1-Score: Trung bình điều hòa của precision và recall, cung cấp một độ đo cân bằng giữa hai yếu tố này.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Bảng kết quả:
Trước khi augmentation và tiền xử lý dữ liệu:

Model	Acc	Hate Speech			Offensive Language			Neither		
		P	R	F1	P	R	F1	P	R	F1
BERT + CNN	0.88	0.74	0.94	0.83	0.94	0.80	0.87	0.92	0.94	0.93
BERT + LSTM	0.89	0.77	0.94	0.85	0.94	0.82	0.88	0.92	0.94	0.93
BERT	0.86	0.72	0.90	0.80	0.89	0.80	0.84	0.89	0.90	0.89

Sau khi augmentation và tiền xử lý dữ liệu

Model	Acc	Hate Speech			Offensive Language			Neither		
		P	R	F1	P	R	F1	P	R	F1
BERT + CNN	0.90	0.76	0.96	0.85	0.96	0.83	0.89	0.94	0.96	0.95
BERT + LSTM	0.92	0.79	0.97	0.87	0.97	0.85	0.91	0.95	0.97	0.96
BERT	0.88	0.74	0.93	0.82	0.92	0.82	0.87	0.91	0.93	0.92

Các phương pháp và mô hình đã được triển khai và đánh giá trên các bộ dữ liệu mẫu. Các kết quả thực nghiệm cho thấy mô hình BERT-LSTM đạt được hiệu quả tốt hơn so với các mô hình khác.

4 Kết luận

Với mục đích phát hiện ngôn từ gây kích động thù hận trên mạng xã hội, nghiên cứu của chúng em đã sử dụng bộ dữ liệu Davidson, một bộ dữ liệu khá "mất cân bằng" nhưng đã được xử lý bằng nhiều phương pháp khác nhau, để huấn luyện cho các mô hình kết hợp giữa Transformer, cụ thể là BERT với các mô hình học sâu (CNN và LSTM), để tạo ra các bộ phân loại đa nhãn có khả năng phát hiện nội dung thù hận và xúc phạm từ Twitter tương đối chính xác, mặc dù vẫn có một vài nội dung mà mô hình phân loại bị sai. Trong tương lai, với vốn kiến thức ngày càng rộng mở, chúng em sẽ cố gắng khắc phục và nghiên cứu áp dụng các kiến trúc mạng học sâu khác và các kĩ thuật nhúng ngữ cảnh bổ sung để có thể xây dựng những bộ phân loại ngôn từ kích động thù hận một cách mạnh mẽ, và toàn diện hơn.

5 Phân công công việc

STT	Công việc	Nguyễn Đình Vũ	Nguyễn Thành Vinh
1	Xác định bài toán	60	40
2	Chuẩn bị dataset	60	40
3	EDA + Tiền xử lý dữ liệu	70	30
4	Xây dựng mô hình	BERT, BERT + CNN	BERT + LSTM
5	Làm demo	40	60
6	Thiết kế slide	70	30
7	Viết báo cáo	80	20
8	Giải trình và trả lời câu hỏi	70	30
9	Mức độ hoàn thành	65	35

6 Tài liệu tham khảo

References

[1] **BERT-based Ensemble Approaches for Hate Speech Detection**, arXiv:2209.06505, https://arxiv.org/abs/2209.06505

[2] **Advanced BERT-CNN for Hate Speech Detection**, ScienceDirect, https://www.sciencedirect.com/science/article/pii/S2405959519307027

[3] **Cross-entropy vs Sparse Cross-entropy: When to Use One Over the Other**, StackExchange, https://stats.stackexchange.com/questions/326065/cross-entropy-vs-sparse-cross-entropy-when-to-use-one-over-the-other

[4] **Probabilistic Losses in Keras**, Keras Documentation, https://keras.io/api/losses/probabilistic_losses/

[5] **Phân tích nội dung và tạo đám mây từ khóa Wordcloud từ đoạn văn bản tiếng Nhật**, Viblo

[6] **EDA for NLP: Data Augmentation for NLP**, Jason Wei, https://github.com/jasonwei20/eda_nlp

[7] **BERT for Hate Speech Detection**, arXiv:1910.12574, https://arxiv.org/pdf/1910.12574

[8] **Hate Speech and Offensive Language Dataset**, Kaggle, https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

[9] **BERT Model Overview**, Pham Dinh Khanh, https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html

[10] **Using Convolutional Neural Network with BERT for Intent Determination**, He, C., Chen, S., Huang, S., Zhang, J., Song, X., 2019 International Conference on Asian Language Processing (IALP), pp. 65–70, IEEE