JTE
HCMUTE
ISSN: 1859-1272

TẠP CHÍ KHOA HỌC GIÁO DỤC KỸ THUẬT
Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Toxic Text Detection In Vietnamese Language

Dong Tran, Minh Phuoc Huynh, Mai Thanh Nhat Van, Nhat Quang Tran, Minh Tan Le*
Ho Chi Minh City University of Technology and Education, Vietnam

*Corresponding author. *Email: tanlm@hcmute.edu.vn*

ARCTICAL INFO

Received: 04/02/2024
Revised: 11/03/2024
Accepted: 27/03/2024
Published: 28/04/2024

KEYWORDS

Machine Learning;
Natural Language Processing;
Text Classification;
Long Short-Term Memory;
Gated Recurrent Unit.

ABSTRACT

The rapid growth of online platforms in recent years, such as social In recent years, the online world has seen an explosion of platforms for communication and sharing. Social networks, forums, and countless websites have created a vast and diverse online landscape. This abundance of content, while exciting, has also introduced new challenges, particularly when it comes to protecting children. The ease of access to the internet can expose them to potential risks, such as encountering toxic language and online bullying. Traditional methods of mitigation, like blocking connections or restricting screen time, can be cumbersome and may not be entirely effective. This paper proposes a novel solution that leverages the power of deep learning. By training deep learning models to identify malicious phrases, our models can recognize various forms of inappropriate language, including both sensitive words and seemingly harmless words used with harmful intent. This intelligent filtering system can be implemented on both the server-side and client-side of online platforms, offering a robust layer of protection for users as they navigate the digital world.

Nhận Diện Ngôn Ngữ Độc Hại Tiếng Việt

Trần Đông, Huỳnh Minh Phước, Văn Mai Thanh Nhật, Trần Nhật Quang, Lê Minh Tân*
Trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh, Việt Nam

*Tác giả liên hệ. *Email: tanlm@hcmute.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận bài: 04/02/2024
Ngày hoàn thiện: 11/03/2024
Ngày chấp nhận đăng: 27/03/2024
Ngày đăng: 28/04/2024

TỪ KHÓA

Học máy;
Xử lí ngôn ngữ tự nhiên;
Phân loại văn bản;
Bộ nhớ dài-ngắn hạn;
Bộ nhớ tái phát.

TÓM TẮT

Trong những năm gần đây, thế giới trực tuyến đã chứng kiến sự bùng nổ của các nền tảng giao tiếp và chia sẻ. Mạng xã hội, diễn đàn và vô số trang web đã tạo ra một không gian trực tuyến rộng lớn và đa dạng. Mặc dù sự phong phú về nội dung này thú vị và có thể hữu ích, nhưng nó cũng mang đến những thách thức mới, đặc biệt là về vấn đề bảo vệ trẻ em. Việc dễ dàng truy cập internet có thể khiến trẻ em tiếp xúc với các rủi ro tiềm ẩn, chẳng hạn như gặp phải ngôn ngữ độc hại và bắt nạt trực tuyến. Các phương pháp giảm thiểu truyền thống, chẳng hạn như chặn kết nối hoặc hạn chế thời gian sử dụng màn hình, có thể không thực sự hiệu quả. Bài báo này đề xuất một giải pháp sử dụng sức mạnh của học sâu. Bằng cách đào tạo các mô hình học sâu để xác định các cụm từ độc hại, các mô hình của chúng tôi có thể nhận dạng các dạng ngôn ngữ không phù hợp khác nhau, bao gồm cả từ nhạy cảm và những từ có vẻ vô hại được sử dụng với mục đích gây hại. Hệ thống lọc thông minh này có thể được triển khai trên cả phía máy chủ và phía máy khách của các nền tảng trực tuyến, cung cấp một lớp bảo vệ tốt hơn cho người dùng trong thế giới kỹ thuật số.

Doi: https://doi.org/10.54644/jte.2024.1528

1. Giới thiệu

Sự phát triển nhanh chóng của các nền tảng trực tuyến những năm gần đây, như mạng xã hội, diễn đàn, và trang mạng, đã làm nội dung trên internet ngày càng phong phú và dễ dàng tiếp cận hơn cho mọi người. Thông qua các kênh thông tin này, chúng ta có thể tiếp nhận và chia sẻ thông tin một cách nhanh

chóng, thuận tiện. Các trang web và mạng xã hội này có một lượng lớn người sử dụng từ đa dạng các thành phần xã hội khác nhau. Bùng nổ theo là nội dung trên Internet cũng ngày càng trở nên phong phú và khó đoán hơn. Mọi người ngày nay coi mạng xã hội không chỉ là một kênh phương tiện giải trí mà còn là nơi giao lưu và chia sẻ thông tin với nhau. Không chỉ người lớn mới sử dụng Internet và mạng xã hội, trẻ em ngày nay cũng được tiếp cận với Internet từ sớm để phục vụ học tập và giải trí. Tuy nhiên ngoài những lợi ích có thể thấy được, Internet mang lại những nguy cơ không lường trước được cho những em nhỏ, và một trong số đó là những ngôn từ không phù hợp trên Internet, hay còn gọi là ngôn ngữ độc hại. Vào năm 2020, Việt Nam đã lọt vào Top 5 những nước kém văn minh nhất thế giới trên Internet dựa trên kết quả khảo sát của Microsoft [1]. Điều này cho thấy môi trường mạng Việt Nam tiềm ẩn nhiều nguy cơ cho các em nhỏ tiếp cận tới những ngôn ngữ không phù hợp với thuần phong mĩ tục, làm xấu đi hình ảnh đất nước trong mắt bạn bè quốc tế.

Hiểu được điều đó, với mục tiêu đưa ra giải pháp ngăn chặn sự tiêu cực mà ngôn ngữ độc hại đem đến, góp phần tạo nên môi trường mạng lành mạnh ở Việt Nam, bài báo là sự tổng hợp quá trình nghiên cứu, xây dựng mô hình nhận dạng mức độ độc hại ngôn ngữ hiệu quả. Nội dung bao gồm trình bày các khái niệm cơ bản và các công trình liên quan, mô tả về dữ liệu và công cụ được sử dụng, cách tiếp cận của bài toán. Để chứng minh hiệu quả mô hình, kết quả thực nghiệm và thảo luận cũng được nêu ra. Phần cuối dành cho kết luận, đánh giá và đưa ra hướng phát triển của đề tài.

2. Các nghiên cứu trước đây và kiến thức liên quan

2.1. Tổng quan các nghiên cứu về nhận dạng ngôn ngữ độc hại

Hướng tiếp cận phổ biến trong bài toán này là sử dụng các mô hình học sâu chuyên dùng cho dữ liệu chuỗi và xử lí ngôn ngữ tự nhiên. Như trong công bố [9], nhóm tác giả sử dụng các phương pháp tokenization kết hợp với mô hình học sâu LSTM. Hướng tiếp cận này cho ra một mô hình với kết quả cho ra khá cao precision 94.49%, recall 92.79% và accuracy 94.94%. Một nghiên cứu khác sử dụng một phiên bản nâng cấp hơn của LSTM là Bidirectional LSTM nhằm cải thiện thêm độ chính xác của mô hình dự đoán [10]. Ngoài việc chỉ sử dụng mạng thần kinh hồi quy, nghiên cứu [11] còn sử dụng thêm mạng thần kinh tích chập (CNN) song song với mô hình LSTM. Mặc dù kết quả đánh giá cho thấy CNN cũng đạt hiệu quả khá tốt, nhưng LSTM vẫn vượt trội hơn về cả độ chính xác và hiệu suất thời gian khi sử dụng cùng số epoch. Nghiên cứu [12] đã chỉ ra rằng không chỉ việc áp dụng các mô hình phức tạp mà còn việc sử dụng các phương pháp tiền xử lý và nhúng từ cơ bản cũng có thể ảnh hưởng đến hiệu suất phân loại. Để chứng minh điều này, nhóm tác giả đã tiến hành đánh giá thực nghiệm kiến trúc kết hợp BiLSTM + CNN, mô hình ngôn ngữ BERT (Bidirectional Encoder Representation from Transformer) với nhiều phương pháp tiền xử lý và nhúng từ khác nhau.

Trong nghiên cứu này, chúng tôi đề xuất, huấn luyện và đánh giá thử nghiệm các mô hình có thể nhận diện ngôn ngữ độc hại cho văn bản tiếng Việt, sử dụng kết hợp các phương pháp tokennization cho tiếng Việt và các mô hình học sâu tiên tiến bao gồm Bidirectional LSTM và Bidirectional GRU. Nền tảng lý thuyết của những mô hình này sẽ được trình bày trong các phần bên dưới.

2.2. Mạng thần kinh hồi quy

2.2.1. Giới thiệu

Kiến trúc mạng thần kinh nhân tạo (artificial neural networks) được thiết kế để nhận các điểm dữ liệu đầu vào có số chiều cố định, và thứ tự các điểm này không quan trọng. Tuy nhiên, trong thực tế có nhiều loại dữ liệu có số chiều không cố định, hay có những dữ liệu mà thứ tự của nó quan trọng. Điển hình là dữ liệu có dạng chuỗi các giá trị thực (real-valued) như chuỗi thời gian (time-series), hay dữ liệu là chuỗi các ký hiệu có thứ tự (symbolic) như văn bản, dữ liệu sinh học [2].

Để khắc phục điều này, người ta đã tạo ra một cấu trúc khác cho mạng thần kinh nhân tạo, đó là mạng thần kinh hồi quy (recurrent neural network - RNN). Mạng thần kinh hồi quy là các mạng thần kinh nhân tạo kết nối với nhau để tạo thành đồ thị có hướng dọc theo một trình tự thời gian. Ý tưởng căn bản khi tạo ra mạng thần kinh hồi quy là xem mạng như một vòng lặp, các vòng lặp liên tiếp nhau cho phép mạng sử dụng thông tin từ các dữ liệu trước đó như một cơ chế ghi nhớ. Mạng có thể tăng giảm số lượng

thần kinh tùy ý. Điều này cho phép nó tiếp nhận dữ liệu với bất kỳ chiều dài nào nên nó có thể áp dụng cho các tác vụ như nhận dạng chữ viết tay hay nhận dạng tiếng nói, vốn có tính chất kết nối, không phân đoạn. Mạng thần kinh hồi quy mô phỏng hoạt động của não bộ con người, cho phép máy tính có thể nhận diện các khuôn mẫu sẵn có để xử lí các vấn đề thông thường. Mạng được tạo thành từ nhiều lớp thời gian (temporal layer) tiếp nối nhau và có những hoạt động tương tự như hoạt động của các nơron trong não người. Từ đó, mạng thần kinh hồi quy có thể dự đoán các dữ liệu chuỗi theo một cách mà mạng thần kinh thường không thể.

2.2.2. Phân loại

Kiến trúc của RNN có thể thay đổi dựa trên bài toán cần xử lí, bài toán có thể có một hay nhiều dữ liệu vào và ra. Dưới đây là một số kiến trúc RNN [4]:

• Một-một (one-to-one): Chỉ có một cặp dữ liệu vào - ra ở kiến trúc này. Kiến trúc này thường được sử dụng trong các mạng neuron truyền thống.

• Một-nhiều (one-to-many): Một dữ liệu đầu vào duy nhất có thể tạo ra rất nhiều dữ liệu đầu ra khác nhau. Kiến trúc này thường được sử dụng cho quá trình sản xuất nhạc.

• Nhiều-một (many-to-one): Ở kiến trúc này, một kết quả đầu ra duy nhất được tạo ra bằng cách kết hợp nhiều đầu vào từ các mốc thời gian khác nhau. Kiến trúc này thường được sử dụng trong các bài toán phân tích và nhận dạng cảm xúc, nơi các nhãn được định nghĩa bởi các chuỗi từ.

• Nhiều-nhiều (many-to-many): Kiến trúc này sử dụng một chuỗi nhiều dữ liệu đầu vào để sinh ra một chuỗi nhiều kết quả đầu ra. Ví dụ điển hình cho kiến trúc này là các hệ thống dịch thuật ngôn ngữ.

2.2.3. Lan truyền ngược theo thời gian

Một trong những vấn đề tính toán khi huấn luyện mạng hồi quy là đầu vào có thể rất dài, do đó số lượng lớp thời gian trong mạng cũng có thể rất lớn. Điều này có thể dẫn đến các vấn đề khi phải tính toán và sử dụng bộ nhớ quá nhiều, cùng với tốc độ hội tụ quá chậm. Vấn đề này được giải quyết thông qua phương pháp lan truyền ngược một phần theo thời gian (truncated backpropagation through time). Kỹ thuật này có thể được coi là biến thể của thuật toán xuống đồi ngẫu nhiên (stochastic gradient descent) cho mạng hồi quy [3].

Trong phương pháp này, ban đầu, thực hiện tính toán lan truyền xuôi một cách bình thường, tới bước lan truyền ngược thì chỉ tính đạo hàm của mất mát đối với k lớp thời gian trước đó. Nghĩa là nếu lan truyền xuôi được thực hiện qua 1000 lớp thì lan truyền chỉ cần tính từ lớp 1000 đến lớp 900 nếu $k = 100$, điều này làm giảm đi đáng kể khối lượng tính toán cũng như bộ nhớ cần phải lưu trữ.

2.2.4. Mạng thần kinh hồi quy hai chiều

Một nhược điểm của mạng hồi quy là mỗi một trạng thái ẩn (hidden state) chỉ biết về trạng thái ẩn trước đó nhưng nó không biết về các trạng thái ẩn tương lai. Trong một số ứng dụng như suy luận về ngữ nghĩa của từ, kết quả của mạng sẽ được cải thiện đáng kể nếu biết cả thông tin về quá khứ và tương lai. Ví dụ trong cụm từ "con chó", ta khó có thể dự đoán từ "chó" khi ta chỉ biết từ "con", nhưng có thể dễ dàng dự đoán từ "con" khi đã biết từ "chó". Mạng hồi quy không thể giải quyết vấn đề này vì nó chỉ quan tâm đến các giá trị quá khứ, trong khi trong một số ứng dụng ta cần phải có cái nhìn tổng quát xung quanh thời điểm hiện tại, nghĩa là cả quá khứ lẫn tương lai.

Mạng hồi quy hai chiều được chứng minh là xử lý tốt đối với các ứng dụng cần biết cả giá trị quá khứ lẫn tương lai. Nhưng đối với các ứng dụng không cần biết giá trị tương lai, áp dụng mạng hồi quy hai chiều trong nhiều trường hợp vẫn cho ra độ chính xác cao hơn.

2.2.5. Mạng thần kinh hồi quy nhiều lớp

Trong các ứng dụng thực tế, kiến trúc đa lớp mạng thần kinh được sử dụng để xây dựng các mô hình phức tạp hơn. Thông thường số lượng lớp ẩn tầm hai hoặc ba là đủ [2]. Đối với số lớp ẩn lớn rất dễ bị quá khớp, nên số lượng lớp ẩn nên tỉ lệ thuận với số lượng dữ liệu cho mô hình học.

2.3. Bộ nhớ dài-ngắn hạn

Vấn đề đầu tiên mà chúng ta gặp phải là độ dốc biến mất hoặc độ dốc bùng nổ luôn xảy ra, và ta không thể giải quyết triệt để vấn đề này [2]. Ngoài ra trong mạng hồi quy, kiến trúc của nó làm cho nó hoạt động giống như một bộ nhớ, nhưng bộ nhớ này không phải là vô hạn. Và khi chiều dài dữ liệu tăng lên tương đương với việc số lớp tăng lên, thông tin từ dữ liệu cũ ngày càng mất giá trị so với thông tin mới nhận được. Có thể nói bộ nhớ này bị quên đi ký ức cũ khi lượng ký ức mới nạp vô càng nhiều.

Mạng thần kinh hồi quy học tốt khi sử dụng dữ liệu chuỗi ngắn, mỗi lần chỉ xử lý duy nhất một dữ liệu, và ta gọi mạng hồi quy như vậy có đặc tính của bộ nhớ ngắn hạn (short-term memory). Và ngược lại, để sử dụng dữ liệu chuỗi dài, tức cần tạo ra một mạng thần kinh có đặc tính của bộ nhớ dài hạn (long-term memory), ta phải giải quyết được vấn đề về việc ký ức cũ của bộ nhớ không bị mất đi khi có nhiều ký ức mới. Vì vậy, kiến trúc Bộ nhớ dài-ngắn hạn (Long short-term memory - LSTM) được sinh ra với đặc tính lưu trữ các bộ nhớ ngắn hạn, nhưng trong khoảng thời gian dài hơn.

Một bộ nhớ dài-ngắn hạn là một phiên bản nâng cấp của mạng hồi quy nhiều lớp với sự tinh chỉnh các bước tính toán ở trong trạng thái ẩn khi nó được lan truyền. Để đạt được điều đó ta thêm vô một trạng thái ẩn khác và gọi trạng thái ẩn mới này là một ô trạng thái (cell state). Ta có thể xem ô trạng thái này hoạt động như một bộ nhớ thông thường với hai chức năng chính là "quên" và "nhớ".

Xét một nút ẩn bất kỳ trong mạng thần kinh nhiều lớp, phương trình để tính một trạng thái ẩn bất kỳ là $W_t^{(k)}$. Gọi $\boldsymbol{i}, \boldsymbol{f}, \boldsymbol{o}, \boldsymbol{c}$ là bốn vector tức thời khác nhau thể hiện cho đầu vào (input), quên (forget), đầu ra (output), ô trạng thái (cell state). Ta có thể xem ba vector $\boldsymbol{i}, \boldsymbol{f}, \boldsymbol{o}$ là ba cổng logic (logic gate). Ba cổng này không mang thông tin mà chỉ hoạt động với mục đích là có cho phép nhận đầu vào hay không (input gate), có cho phép quên ký ức cũ hay không (forget gate), và có cho phép kết hợp với đầu ra hay không (output gate). Với vector $\boldsymbol{c}$ là bộ nhớ tức thời của lớp này, ta có phương trình tính toán trong lớp

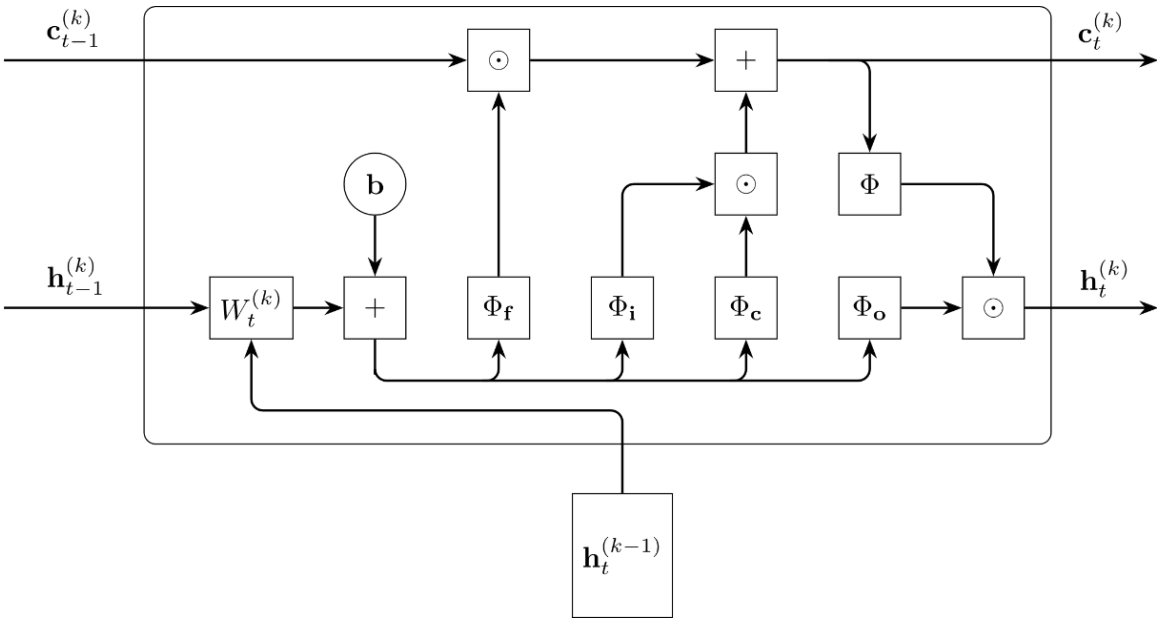
$$\begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{f} \\ \boldsymbol{o} \\ \boldsymbol{c} \end{bmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \Phi \end{pmatrix} \left(W_t^{(k)} \begin{bmatrix} \boldsymbol{h}_t^{(k-1)} \\ \boldsymbol{h}_{t-1}^{(k)} \end{bmatrix} + \boldsymbol{b} \right) \tag{1}$$

$$\boldsymbol{c}_t^{(k)} = \boldsymbol{f} \odot \boldsymbol{c}_{t-1}^{(k)} + \boldsymbol{i} \odot \boldsymbol{c} \tag{2}$$

$$\boldsymbol{h}_t^{(k)} = \boldsymbol{o} \odot \Phi\left(\boldsymbol{c}_t^{(k)}\right) \tag{3}$$

Nếu trạng thái ẩn có số chiều là p, thì ma trận $W_t^{(k)}$ bây giờ sẽ có số chiều là $4p \times 2p$, vì nó cần tạo ra bốn vector trung gian ở đầu ra. Chi tiết về cách hoạt động của bộ nhớ dài-ngắn hạn được trình bày trong Hình 1.

JTE
HCMUTE
ISSN: 1859-1272



Hình 1. *Kiến trúc của LSTM*

2.4. Bộ nhớ tái phát

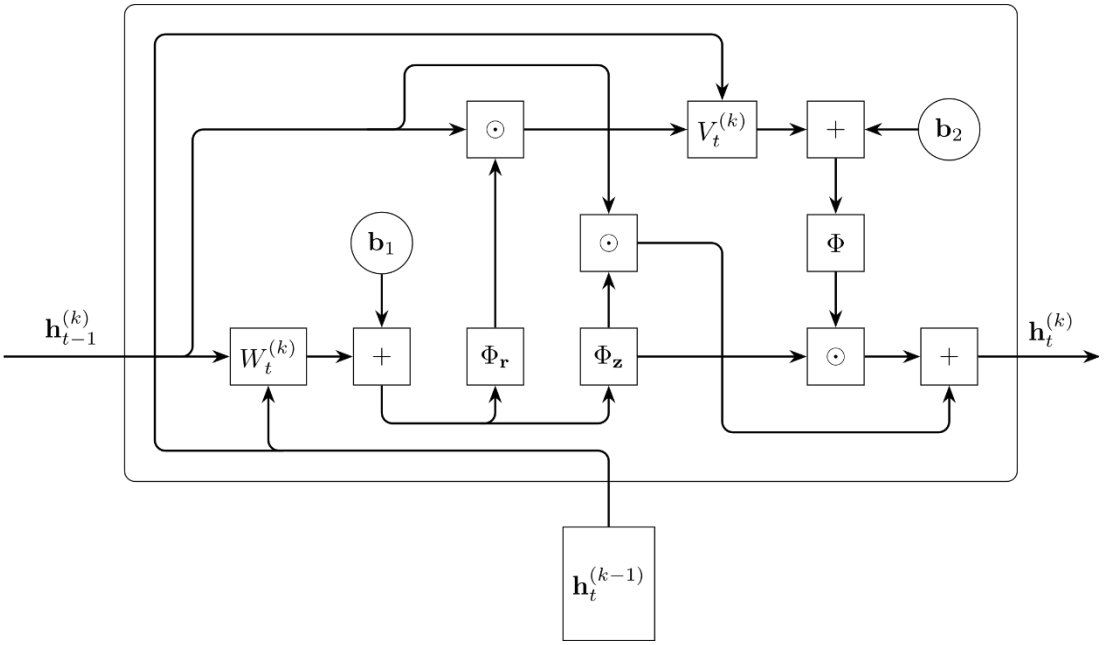
Bộ nhớ tái phát (Gated Recurrent Unit - GRU) có thể được xem xét là một phiên bản đơn giản của bộ nhớ dài-ngắn hạn. Đối với bộ nhớ dài-ngắn hạn, nó điều chỉnh trực tiếp lượng thông tin của trạng thái ẩn bằng cổng quên và cổng đầu ra, còn bộ nhớ tái phát chỉ sử dụng một cổng đặt lại (reset gate). Điểm khác biệt lớn là bộ nhớ tái phát không dùng ô trạng thái làm bộ nhớ.

Xét một nút ẩn bất kỳ trong mạng thần kinh nhiều lớp, thay vì phải tính đến bốn vector trung gian thì bộ nhớ tái phát chỉ tính hai vector trung gian là cổng đặt lại và cổng cập nhật (update gate).

$$\begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{r} \end{bmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \end{pmatrix} \left(W_t^{(k)} \begin{bmatrix} \boldsymbol{h}_t^{(k-1)} \\ \boldsymbol{h}_{t-1}^{(k)} \end{bmatrix} + \boldsymbol{b}_1 \right) \tag{4}$$

$$\boldsymbol{h}_t^{(k)} = \boldsymbol{z} \odot \boldsymbol{h}_{t-1}^{(k)} + (1 - \boldsymbol{z}) \odot \Phi \left(V_t^{(k)} \begin{bmatrix} \boldsymbol{h}_t^{(k-1)} \\ \boldsymbol{h}_{t-1}^{(k)} \end{bmatrix} + \boldsymbol{b}_2 \right) \tag{5}$$

Ma trận $W_t^{(k)}$ bây giờ chỉ có số chiều là $2p \times 2p$ vì nó chỉ cần tạo ra hai vector trung gian ở đầu ra, còn ma trận $V_t^{(k)}$ sẽ có số chiều là $p \times 2p$. Vì bộ nhớ tái phát không sử dụng ô trạng thái để làm bộ nhớ, nên hai cổng của nó phải đảm nhiệm nhiệm vụ này. Chi tiết về cách hoạt động của bộ nhớ dài-ngắn hạn được trình bày trong Hình 2.



Hình 2. *Kiến trúc của GRU*

2.5. Kỹ thuật nhúng từ

Thông thường, trong học máy, để máy móc xử lí ngôn ngữ thì các từ sẽ được biểu diễn dưới dạng one-hot encoding, dữ liệu của các từ sẽ được chuyển về dạng các vector với số chiều tương ứng với tổng số từ. Tuy nhiên cách thể hiện one-hot encoding có một số vấn đề: Chi phí lớn do số chiều của vector tương ứng với tổng số từ khác nhau trong dữ liệu, các vector hầu như không mang thông tin giá trị hay sự liên kết giữa các từ với nhau và thiếu sự khái quát khi ngôn ngữ có nhiều từ có nghĩa giống nhau [5].

Ở word embedding xử lý được những vấn đề nêu trên, bằng cách dùng một không gian vector để biểu diễn dữ liệu có khả năng miêu tả được mối liên hệ, sự tương đồng về mặt ngữ nghĩa, văn cảnh (context) của dữ liệu. Không gian này bao gồm nhiều chiều và các từ trong không gian đó mà có cùng văn cảnh hoặc ngữ nghĩa sẽ có vị trí gần nhau.

Có 2 phương pháp chủ yếu được sử dụng trong word embedding là Count based method và Predictive method [6]. Các phương pháp này đều dựa trên một giả thuyết đó là những từ nào xuất hiện cùng nhau trong một ngữ cảnh sẽ có vị trí gần nhau trong không gian vector.

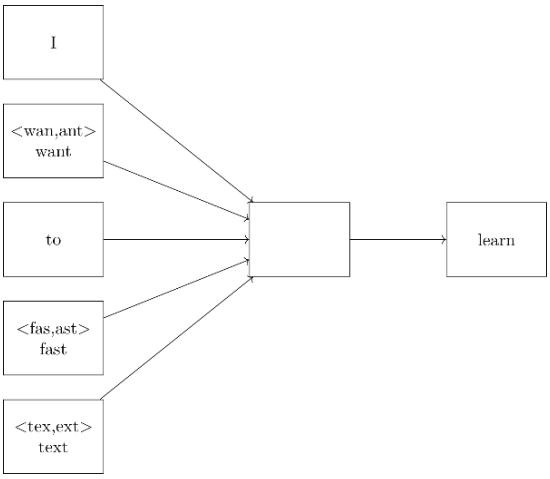
Phương pháp Count-based method tính toán mức liên quan về mặt ngữ nghĩa giữa các từ bằng cách thống kê số lần đồng xuất hiện của một từ so với các từ khác. Khác so với Count-based method, Predictive method tính toán sự tương đồng ngữ nghĩa giữa các từ để dự đoán từ tiếp theo bằng cách đưa qua một mạng neural network có một hoặc vài layer dựa trên input là các từ xung quanh (context word). Một context word có thể là một hoặc nhiều từ khác nhau, một mô hình điển hình cho phương pháp này là Word2vec.

Trong bài toán này, chúng tôi sử dụng mô hình mô hình Fasttext embeddings, Fasttext có thể được coi là một sự mở rộng của Word2vec, tuy nhiên khác với Word2vec hoạt động ở mức độ các từ riêng biệt, Fasttext hoạt động ở mức n-gram kí tự đối với mỗi từ. Ví dụ chúng ta có từ “apple” và n = 3 thì n-gram của từ này là: <ap, app, ppl, ple, le> và <apple>. Và vector đại diện cho từ “apple” có thể được tính bằng tổng các vector n-gram của chính nó [7].

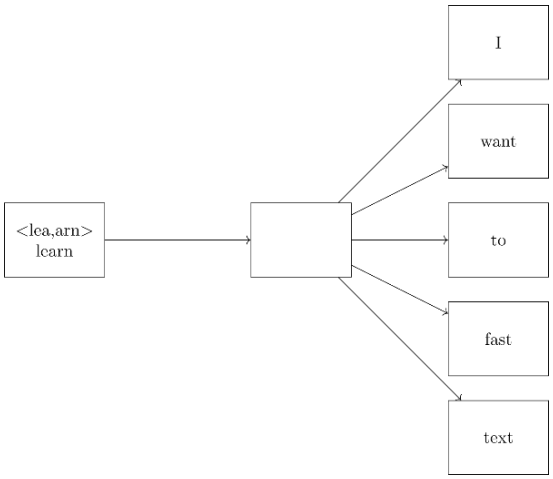
Fasttext hoạt động dựa trên hai phương thức: Continue Bag Of Words và Skip-gram.

Trong mô hình Continue Bag Of Words (CBOW), chúng ta lấy ngữ cảnh của từ mục tiêu làm đầu vào và dự đoán từ xuất hiện trong ngữ cảnh đó. Ví dụ, trong câu "I want to learn FastText". Trong câu này, các từ "I", "want", "to" và "FastText" được đưa vào làm đầu vào, và mô hình dự đoán "learn" là đầu ra. Tất cả dữ liệu đầu vào và đầu ra có cùng kích thước và được mã hóa one-hot (one-hot encoding). Mô

hình CBOW sử dụng mạng thần kinh để huấn luyện. Mạng thần kinh này có một lớp đầu vào, một lớp ẩn và một lớp đầu ra. Hình 3 minh họa cách hoạt động của CBOW.



Hình 3. *Kiến trúc CBOW*



Hình 4. *Kiến trúc Skip-gram*

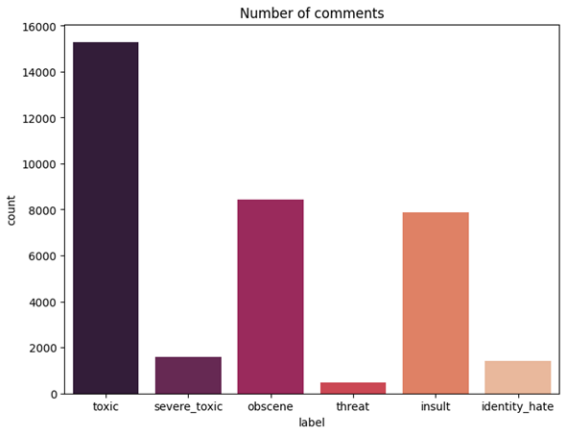
Mô hình Skip-gram hoạt động giống như CBOW, nhưng đầu vào là từ mục tiêu và mô hình dự đoán ngữ cảnh của từ đó. Mô hình này cũng sử dụng mạng thần kinh để huấn luyện. Hình 4 minh họa cách hoạt động của Skip-gram.

3. Dữ liệu và Phương pháp

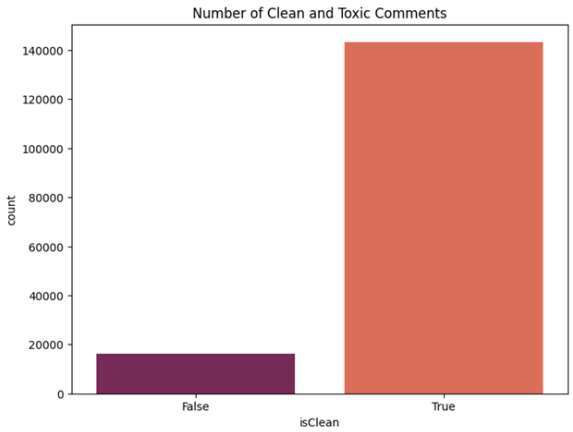
3.1. Dữ liệu

3.1.1. Mô tả tập dữ liệu

Dữ liệu gốc được thu thập từ Kaggle, một nền tảng trực tuyến dành cho cộng đồng khoa học dữ liệu và học máy [8]. Nguồn dữ liệu này xuất phát từ thử thách phân loại bình luận độc hại, được tổ chức và cung cấp bởi Conversation AI, một nhóm nghiên cứu do Jigsaw và Google thành lập. Theo mô tả, dữ liệu thô được trích xuất từ các bình luận trong phần thảo luận (talk page) trên Wikipedia, một bách khoa toàn thư trực tuyến đa ngôn ngữ.



Hình 5. *Tỉ lệ giữa các nhãn thuộc tính*



Hình 6. *Tỉ lệ giữa dữ liệu độc hại (cột False) và không độc hại (cột True)*

Tập dữ liệu bao gồm nhiều cột được mô tả như sau:

- id: Là id của từng bình luận trong tập dữ liệu.
- comment_text: Các bình luận dưới dạng văn bản thuần túy, sẽ được xử lý để máy có thể học được.

- Các nhãn trạng thái thể hiện sự độc hại của bình luận: “toxic”, “severe_toxic”, “obscene”, “threat”, “insult”, “identity_hate”. Các nhãn tiếng Anh này tương ứng với “độc hại”, “cực kỳ độc hại”, “tục tĩu”, “đe doạ”, “xúc phạm”, “thù ghét” trong tiếng Việt.

Tập dữ liệu có khoảng 160 nghìn dòng. Số lượng bình luận của mỗi nhãn trạng thái được thể hiện dưới dạng biểu đồ cột ở Hình 5. Có thể thấy, nhãn “toxic” chiếm số lượng lớn nhất do đây là tiêu chí chung để có thể đánh giá bình luận, các nhãn khác có thể xem như là mở rộng hoặc bổ sung chi tiết cho nhãn “toxic”.

Ngoài ra, chúng ta có thể thấy rằng dữ liệu sạch chiếm đa số trong tập dữ liệu, trong khi dữ liệu độc hại chiếm phần nhỏ hơn (Hình 6). Tuy vậy số lượng dữ liệu độc hại không quá thấp, nên có thể sử dụng để huấn luyện mô hình

3.1.2. Tiền xử lý dữ liệu

Với mục tiêu giải quyết bài toán sử dụng ngôn ngữ tiếng Việt, dữ liệu đã được chuyển đổi sang tiếng Việt thông qua công cụ Google Translate, sử dụng trên Google Sheet để phù hợp với yêu cầu bài toán.

Trong văn bản có thể xuất hiện các kí tự đặc biệt, những kí tự này đôi khi có thể nằm ở những vị trí không phù hợp và gây ảnh hưởng đến cấu trúc của câu. Đồng thời, những dòng dữ liệu trùng lặp thường không mang lại nhiều giá trị trong quá trình huấn luyện và tăng dung lượng bộ nhớ, do đó cũng cần phải được loại bỏ.

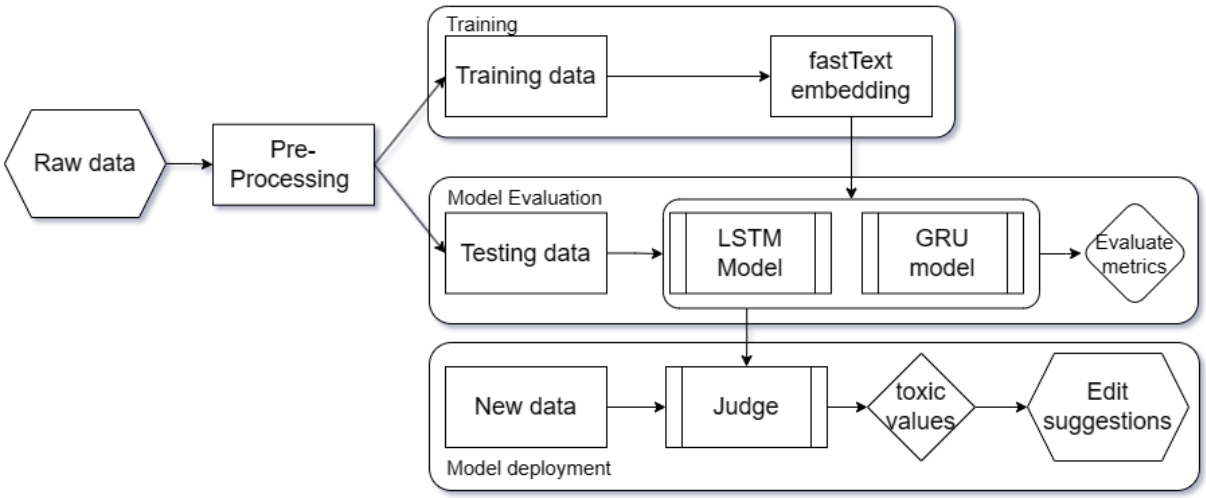
Trước khi bắt đầu quá trình huấn luyện mô hình, tập dữ liệu huấn luyện đã được phân chia thành hai phần là tập train và tập test theo tỉ lệ 9:1. Sau đó, trong quá trình huấn luyện, tập dữ liệu train được chia thành hai phần train và valid với tỉ lệ 9:1. Hai tập dữ liệu train và valid được sử dụng để huấn luyện mô hình, trong khi tập dữ liệu test được dành để kiểm thử hiệu suất của mô hình đã được huấn luyện.

3.2. Phương pháp đề xuất

Kiến trúc tổng quan của hệ thống phát hiện văn bản độc hại được minh họa trong Hình 7.

Hệ thống phân loại độc hại trong ngôn ngữ sẽ bắt đầu với một loạt các bước tiền xử lý dữ liệu để làm sạch và chuẩn hóa dữ liệu thành một định dạng mà mô hình có thể hiểu được bao gồm hai kỹ thuật sẽ được sử dụng: phân tách từ (tokenization) và nhúng từ (word embedding).

Trong nhiệm vụ phân loại, chúng tôi xây dựng hai mô hình mạng thần kinh hồi quy sử dụng hai kiến trúc: LSTM và GRU như đã đề cập. Mỗi mô hình bao gồm sáu lớp: lớp đầu vào nhận dữ liệu đầu vào dưới dạng các token, lớp nhúng từ sử dụng fastText, lớp dropout ở giữa hai lớp mạng thần kinh hồi quy hai chiều để kiểm soát quá khớp (overfitting), và cuối cùng là lớp dense đánh giá mức độ độc hại của văn bản dựa trên các giá trị độc hại đã định nghĩa.



Hình 7. *Sơ đồ hệ thống*

Vì đối mặt với bài toán sử dụng ngôn ngữ tự nhiên, mô hình mạng thần kinh hồi quy hai chiều (bidirectional RNN) được ưu tiên chọn lựa, trong khi mô hình fastText được coi là phù hợp nhất cho tiếng Việt, như đã đề cập trước đó.

Kết quả dự đoán được tạo ra bằng cách lấy trung bình của đầu ra từ hai mô hình, nhằm tối ưu hóa kết quả cuối cùng và cung cấp một số gợi ý chỉnh sửa để phù hợp với các tiêu chuẩn của cộng đồng.

4. Kết quả và thảo luận

4.1. Thiết lập mô hình

4.1.1. Phương pháp đánh giá

Để đánh giá tính chính xác của mô hình, ba phương pháp đánh giá được sử dụng là Precision, Recall và Accuracy. Với TP là True Positive, FP là False Positive, TN là True Negative, FN là False Negative.

Precision là tỉ lệ đối tượng được gán các kết quả giống nhau có kết quả thực sự giống nhau.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall là tỉ lệ các đối tượng giống nhau được gán các kết quả giống nhau.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Accuracy là tỉ lệ các đối tượng được gán đúng với kết quả mẫu.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

4.1.2. Tokenization

Tokenization đánh số cho mỗi từ có trong tập từ. Dưới đây là minh họa sau khi thực hiện tokenize tập dữ liệu.

Bảng 1. *Các từ tương ứng sau tokenization*

tôi	của	một	bạn	là	có	không	và	các	đã
1	2	3	4	5	6	7	8	9	10

4.1.3. Word Embedding

Mô hình nhóm sử dụng là mô hình fasttext. Do mô hình này hoạt động dựa trên cách tách các từ thành nhiều phần nhỏ, sẽ dễ dàng hơn để liên hệ với các từ viết thiếu rõ ràng. Mô hình fasttext được sử dụng là một mô hình đã được huấn luyện từ trước chuyên dụng cho tiếng Việt.

Với 200000 là số từ tối đa có thể xuất hiện và 300 là số chiều của một vector từ, khởi tạo một ma trận với kích thước 200000×300 để chứa các vector từ. Với mỗi từ đã được mô hình hóa bởi fasttext, lấy ra từ đó và index của chúng, sau đó lấy ra vector từ tương ứng. Sau cùng thay thế vector đã lấy vào vị trí tương ứng trong ma trận. Với cách làm này, với mỗi từ sau khi đã trải qua quá trình tokenization sẽ được nhận biết bằng vector nhúng từ tương ứng.

4.2. Tham số và huấn luyện mô hình

Quá trình huấn luyện được thực hiện với hai mô hình học sâu là LSTM và GRU. Bảng 1 và Bảng 2 là tóm tắt của từng mô hình.

Bảng 2. *Bảng tóm tắt mô hình LSTM*

Layer (type)	**Output Shape**	**Param #**
input (InputLayer)	[(None, 200)]	0

embedding (Embedding)	(None, 200, 300)	60000000
bidirectional (Bidirectional)	(None, 200, 128)	186880
dropout (Dropout)	(None, 200, 128)	0
bidirectional_1 (Bidirectional)	(None, 128)	98816
dense (Dense)	(None, 6)	774
Total params: 60286470 (299.97 MB)		
Trainable params: 286470 (1.09 MB)		
Non-trainable params: 60000000 (228.88 MB)		

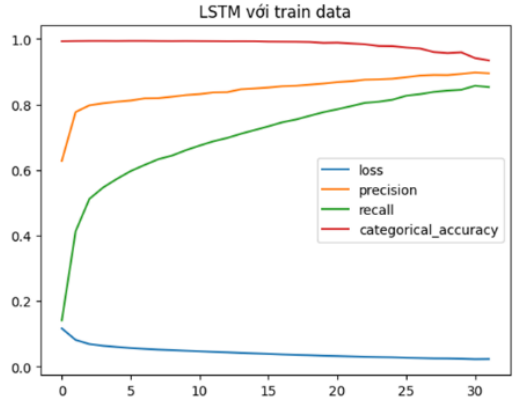
Bảng 3. *Bảng tóm tắt mô hình GRU*

Layer (type)	**Output Shape**	**Param #**
input (InputLayer)	[(None, 200)]	0
embedding (Embedding)	(None, 200, 300)	60000000
bidirectional (Bidirectional)	(None, 200, 128)	140544
dropout (Dropout)	(None, 200, 128)	0
bidirectional_1 (Bidirectional)	(None, 128)	74496
dense (Dense)	(None, 6)	774
Total params: 60215814 (229.71 MB)		
Trainable params: 215814 (843.02 KB)		
Non-trainable params: (228.88 MB)		

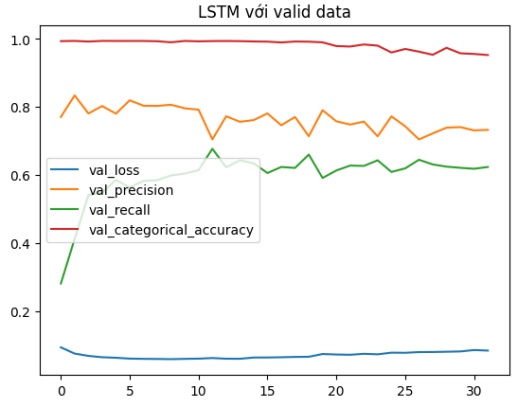
4.3. Kết quả thực nghiệm

Đường học tập (learning curve) của mô hình LSTM được thể hiện trong Hình 8, và Hình 9; mô hình GRU được thể hiện trong Hình 10, và Hình 11. Có thể thấy rằng các mô hình cho ra độ chính xác tương đối cao, trên 90% và giá trị mất mát cũng rất khả quan. Từ đó có thể kết luận rằng các mô hình có thể hoạt động tương đối ổn, trong đó mô hình LSTM có hiệu suất tốt hơn một chút so với GRU trên tập dữ liệu này.

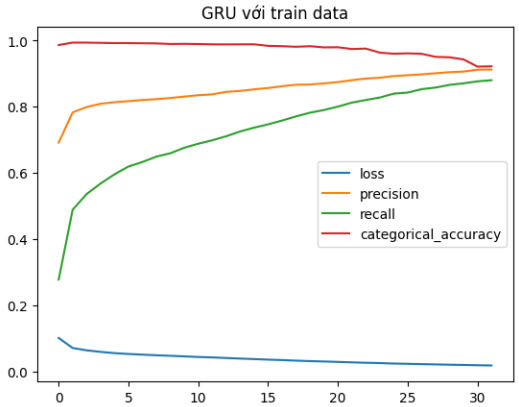
Kết quả thực nghiệm đánh giá mô hình trên tập dữ liệu test được thể hiện trong Bảng 4. Có thể thấy rằng không có nhiều khác biệt giữa kết quả trên tập test và tập validation.



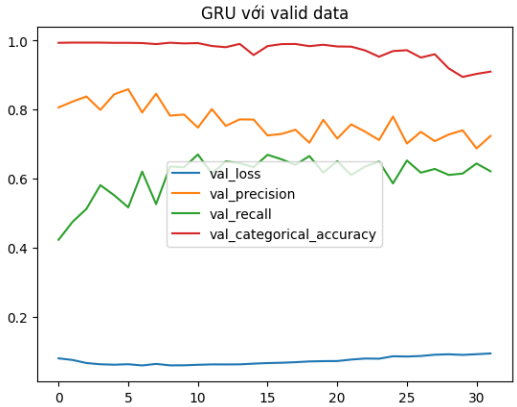
Hình 8. *Kết quả đánh giá dự đoán của mô hình LSTM với dữ liệu huấn luyện*



Hình 9. *Kết quả đánh giá dự đoán của mô hình LSTM với dữ liệu kiểm tra*



Hình 10. *Kết quả đánh giá dự đoán của mô hình GRU với dữ liệu huấn luyện*



Hình 11. *Kết quả đánh giá dự đoán của mô hình GRU với dữ liệu kiểm tra*

Bảng 4. *Thông số đánh giá mô hình trên tập dữ liệu test*

Metrics	LSTM	GRU
Loss	0.07367267459630966	0.09395545721054077
Precission	0.7359092235565186	0.7325249910354614
Recall	0.6160237193107605	0.6094955205917358
Accuracy	0.9714876413345337	0.9583907723426819

5. Kết luận

Bài báo này đã đạt được những kết quả tích cực trong việc xây dựng hai mô hình mạng thần kinh sử dụng kiến trúc LSTM và GRU, kết hợp với kỹ thuật nhúng từ Fasttext, nhằm mục đích phát hiện văn bản độc hại.

Tuy nhiên, còn ghi nhận một số hạn chế cần được thận trọng trong việc đánh giá. Trong đó, việc sử dụng dữ liệu tiếng Anh và sau đó dịch sang tiếng Việt có thể dẫn đến sự mất mát thông tin và khác biệt ngôn ngữ, tạo nên các sai lệch và thiếu sót trong quá trình huấn luyện mô hình. Bên cạnh đó, mặc dù mô hình được đề xuất trong bài báo cho thấy độ chính xác tương đối ổn, nhưng vẫn xuất hiện những trường hợp mà mô hình không thể nhận diện chính xác mức độ độc hại. Điều này có thể được giải thích bởi sự phức tạp khi chuyển đổi giữa các ngôn ngữ khác nhau và bối cảnh văn bản có thể chứa đựng những yếu tố không dễ dàng nhận biết.

Hướng phát triển trong giai đoạn ngắn sẽ tập trung chủ yếu vào việc cải thiện phương pháp dịch và xử lý dữ liệu. Cần áp dụng những phương pháp dịch chính xác hơn, nhằm giảm thiểu ảnh hưởng của sai lệch ngôn ngữ. Bên cạnh đó, sự mở rộng quy mô thu thập dữ liệu thuần Việt sẽ đóng góp vào việc tăng cường độ đa dạng và tính đại diện của dữ liệu. Ngoài phần dữ liệu, quá trình tinh chỉnh mô hình sẽ được tiếp tục với mục tiêu giảm thiểu sai sót. Điều này sẽ đảm bảo rằng mô hình không chỉ chính xác về ngôn ngữ mà còn đáp ứng đúng mức độ độc hại, từ đó tối ưu hóa hiệu suất sử dụng.

Về mặt triển khai ứng dụng, mô hình có thể được tích hợp như một bộ lọc ngôn ngữ độc hại cho cả phía server và client. Phía server có thể triển khai mô hình để tự động chặn người dùng khi họ đăng bình luận có văn bản độc hại. Đồng thời, việc phát triển một extension cho trình duyệt web từ phía client giúp bảo vệ người dùng, tạo ra một giải pháp toàn diện cho vấn đề độc hại trên nền tảng trực tuyến.

Xung đột lợi ích

Các tác giả tuyên bố không có xung đột lợi ích trong bài báo cáo này.

TÀI LIỆU THAM KHẢO

[1] "Vietnam 'in' top 5 countries with poor online behavior." https://vtc.vn/viet-nam-lot-top-5-ung-xu-kem-van-minh-tren-internet-ar529256.html, 2020. Accessed: Dec. 21, 2023.
[2] C. C. Aggarwal, "Neural networks and deep learning: A textbook." Cham, Switzerland: Springer Nature, Jun. 2023.
[3] C. C. Aggarwal, "Machine learning for text." Cham, Switzerland: Springer Nature, May 2022.
[4] S. A. Amidi, "Recurrent neural networks cheatsheet." https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks, 2019. Accessed: Dec. 13, 2023.
[5] "What is word embedding? Why is it important?." https://trituenhantao.io/kien-thuc/word-embedding-la-gi-tai-sao-no-quan-trong/, 2019. Accessed: Dec. 13, 2023.
[6] B. Q. Manh, "Viblo - word embedding - understanding basic concepts in NLP." https://viblo.asia/p/word-embedding-tim-hieu-khai-niem-co-ban-trong-nlp-1Je5E93G5nL, 2020. Accessed: Dec. 13, 2023.
[7] V. A. Krithika, "Introduction to fasttext embeddings and its implication." https://www.analyticsvidhya.com/blog/2023/01/introduction-to-fasttext-embeddings-and-its-implication/, 2023. Accessed: Dec. 13, 2023.
[8] "Toxic comment classification challenge." https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data, 2018. Accessed: Dec. 21, 2023.
[9] K. Dubey, R. Nair, M. U. Khan, and S. Shaikh, "Toxic comment detection using LSTM," in *Proc. 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, Dec. 2020, doi:10.1109/icaecc50550.2020.9339521.
[10] A. K. Bala, "Toxic comments identification and classification using Deep Neural Networks," Academia.edu, https://www.academia.edu/41458366/Toxic_Comments_Identification_and_Classification_Using_Deep_Neural_Networks. Accessed: Dec. 21, 2023.
[11] R. Sharma and M. Patel, "Toxic comment classification using neural networks and machine learning," *IARJSET*, vol. 5, no. 9, pp. 47–52, Sep. 2018, doi:10.17148/iarjset.2018.597.
[12] V. M. Krešňáková, M. Sarnovský, P. Butka, and K. Machová, "Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification," *Applied Sciences*, vol. 10, no. 23, p. 8631, Dec. 2020, doi:10.3390/app10238631.

Trần Đông is pursuing a degree in Information Technology at the Ho Chi Minh City University of Technology and Education (HCMUTE). Email: 20133035@student.hcmute.edu.vn.

Huỳnh Minh Phước is pursuing a degree in Information Technology at the Ho Chi Minh City University of Technology and Education (HCMUTE). Email: 20133082@student.hcmute.edu.vn.

Văn Mai Thanh Nhật is pursuing a degree in Information Technology at the Ho Chi Minh City University of Technology and Education (HCMUTE). Email: 20133076@student.hcmute.edu.vn.

Trần Nhật Quang completed his PhD in Computer Science at Curtin University (Australia). He is currently focusing on applying artificial intelligence to real-world problems, such as predicting agricultural prices to support crop planning and developing systems to support people with hearing and visual impairments. Email: quangtn@hcmute.edu.vn.

Lê Minh Tân graduated from Ho Chi Minh city University of Technology and Education with a bachelor's degree in Information Technology in 2019 and a master's degree in Computer Science in 2021.
Artificial Intelligence, Machine Learning, Deep Learning, Computer Vision
Email: tanlm@hcmute.edu.vn.
ORCID: https://orcid.org/0009-0004-4912-6795