BÁO CÁO DỰ ÁN CUỐI KHÓA

Phát hiện thông tin không đáng tin cậy trên mạng xã hội

Người hướng dẫn: **TS. Vũ Huy Thế**

Người thực hiện: **Nguyễn Văn Sơn (Trưởng nhóm)**

Phạm Ngọc Đông

Đoàn Quang Khải

Nguyễn Thế Hiến

Hà Nội, tháng 1 năm 2021

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

Lời mở đầu

Lý do chọn đề tài

Với sự phát triển vũ bão của internet và mạng xã hội, con người có thêm nhiều những phương tiện để giao tiếp, trao đổi và chia sẻ thông tin, hình ảnh, video mọi lúc mọi nơi mà không bị cản trở bởi yếu tố không gian địa lý. Có thể kể ra một số mạng xã hội phổ biến nhất như Facebook, Twitter, YouTube, WhatsApp, Instagram, Linkedln, Skype,… Sự thu hút của mạng xã hội là rất khó để người dùng chống cưỡng lại, nhất là khi được dùng hoàn toàn miễn phí. Ngày nay, Google, Facebook, Gmail đã trở thành một thứ không thể thiếu trong cuộc sống nhiều người. Theo thống kê của GlobalWeb Index thì Facebook hiện dẫn đầu thế giới với 2,23 tỉ người dùng, có đến 65 triệu doanh nghiệp lập trang thông tin trên mạng này. Kế đến là Youtube với 1,9 tỉ người dùng, WhatsApp là 1,5 tỉ người,… Với sự rộng lớn của nó, việc lan truyền thông tin sẽ là rất khủng khiếp và nó là một mối đe dọa vô cùng lớn.

Như con dao hai lưỡi, mạng xã hội có mặt tích cực và tiêu cực của nó. Nếu biết sử dụng và khai thác đúng đắn chúng sẽ mang lại những lợi ích vô cùng lớn lao cho cá nhân người sử dụng, các doanh nghiệp các cơ quan chính phủ. Ngược lại, đó sẽ là mối hiểm họa tiềm ẩn và gây nhiều phiền lụy cho các cá nhân, tổ chức và trên bình diện lớn hơn là ảnh hướng đến anh ninh của một quốc gia.

Bản thân mạng xã hội không là mối đe dọa, nhưng việc sử dụng chúng thể nào và cách các nhà mạng quản lý và sử dụng thông tin trên mạng xã hội mới là điều đáng nói. Và nó ra một thách thức làm sao để loại bỏ những thông tin như vậy trên mạng xã hội. Và trong báo cáo này, chúng tôi sẽ trình bày ý tưởng sử dụng mô hình học sâu kết hợp học máy truyền thống để phát hiện những thông tin sai lệch này.

Báo cáo này được trình bày thành các chương như sau:

Chương 1. Bài toán phát hiện tin giả: Giới thiệu tin giả là gì, bài toán phát hiện tin giả và hướng tiếp cận giải quyết bài toán
Chương 2. Mô hình ngôn ngữ BERT cho bài toán phân lớp: Giới thiệu mô hình ngôn ngữ BERT áp dụng vào bài toán.
Chương 3. Mô hình giải quyết bài toán: Trình bày quy trình xử lý dữ liệu, xây dựng mô hình.
Chương 4. Kết quả thực nghiệm và đánh giá: Trình bày kết quả và đánh giá mô hình.
Chương 5. Kết luận

CHƯƠNG 1. BÀI TOÁN PHÁT HIỆN TIN GIẢ

1.1 Tin giả trên mạng xã hội

Ngày nay, thông tin giả lan tràn trên mạng xã hội mà không có sự kiểm soát chặt chẽ của các cơ quan chính quyền và các nguy cơ mà chúng mang lại là có thật, bao gồm các môi nguy cơ rất nghiêm trọng sau đây:

Khủng bố không gian mạng

Mối đe dọa lớn nhất đối với an ninh một quốc gia là khủng bố mạng. Trong thời bình, mạng xã hội là mục tiêu được quan tâm hàng đầu của tin tặc. Chúng dùng nhiều cách thức khác nhau: đăng ký nhiều tài khoản giả danh các nhân vật và tổ chức uy tín để tung tin thất thiệt gây hoang mang dư luận. Vài năm trước, tổ chức ISIS đã khai thác Youtube, Twitter, Facebook để chiêu mộ thành viên, tuyên truyền cho chủ nghĩa Hồi giáo cực đoan. Sau kỳ bầu cử Tổng thống Mỹ năm 2016, Cơ quan Điều tra Liên bang (FBI) và Cơ An ninh Quốc gia (NSA) của Mỹ đã cáo buộc tình báo Liên bang Nga đã tìm cách chi phối kết quả bầu cử thông qua việc tung nhiều tin tức thất thiệt trên Facebook.

Kích động bạo lực và nổi loạn ở các quốc gia

Một mạng xã hội với 2 tỉ người dùng như Facebook là một nền tảng lý tưởng để truyền bá thông tin với sức lan tỏa cực kỳ nhanh chóng và rộng khắp. Cuộc nội chiến Syria đã khởi đầu bằng những cuộc biểu tình lớn từ tháng 3.2011, xuất phát từ những lời kêu gọi trên Facebook và Twitter. Trước đó, làn sóng nổi dậy Mùa xuân Ả Rập bùng phát ở một số quốc gia Bắc Phi và Trung Đông cũng có xuất phát điểm tương tự.

Lừa đảo

Mạng xã hội là nơi lý tưởng để những kẻ lừa đảo săn tìm con mồi của mình. Thời gian qua, khá nhiều người dùng Facebook trong nước, do sơ hở và dễ tin, cũng là nạn nhân của kẻ lừa đảo nước ngoài. Trên quy mô lớn hơn, các băng nhóm tội phạm có tổ chức sử dụng mạng xã hội làm phương thức liên lạc và trao đổi cũng như thực hiện các hành vi rửa tiền kín đáo khó bị phát hiện bởi cơ quan pháp luật.

Đặc biệt trong thời gian Đại dịch Corona, ở Việt Nam có rất nhiều thông tin giả được đăng tải lên mạng xã hội gây hoang mang dư luận và làm phức tạp hơn cho công tác phòng, chống dịch bệnh. Trước việc lan tràn thông tin giả trên mạng xã hội, Nhà nước đã ban hành Luật An ninh mang và sau đó là Nghị định 15/2020/NĐ-CP với việc chú trọng xử lý các hành vi đưa thông tin sai sự thật trên mạng xã hội. Việc này đã giúp giảm bớt thông tin giả nhưng việc xử lý mất thời gian, mất công sức vì phải sử dụng con người để xác định.

1.2 Bài toán phát hiện tin giả

Trước tình hình phức tạp của thông tin trên mạng xã hội. Chúng ta cần xác định những thông tin giả một cách chính xác và nhanh chóng nhất có thể. Điều đó ngăn chặn được sự lan truyền thông tin khủng khiếp trên mạng xã hội cũng như tránh những điều tiếc mà những thông tin đó gây ra.

Việc phát hiện thông tin sai lệch trên mạng xã hội là không hề đơn giản kể cả với con người. Bởi vì chúng ta không thể đưa ra một định nghĩa chính xác nhất thế nào là tin giả. Có một số định nghĩa về tin giả như sau:

- Trong Fake news on detection on Social Media: Data mining perspective của Kai Shu et al, và các bài viết sau này của Kai Shu đều dùng định nghĩa như sau: "*Fake news is a news article that is intentionally and verifiably false*".
- Trong Social Media and Fake News in 2016 Election của Hunt Allcott et al, ngoài phần định nghĩa như của Kai Shu thì có thêm một bổ sung: " *Fake news could mislead reader*".
- Trong Deception detection for news: Three types of fakes của V. Runbin et al nêu ra 3 loại tin giả và phân loại theo mức độ ảnh hưởng của chúng: *Serious Fabrications , Large-Scale Hoaxes, Humorous Fakes.*

Từ một số định nghĩa và khái niệm cũng như tìm hiểu về bài toán, chúng tôi nhận thấy rằng nếu chỉ dựa vào thông tin nội dung tin thì các công cụ Học máy khó có thể dự đoán được đâu là thông tin giả. Do vậy, để phát hiện được thông tin không tin cậy trên mạng xã hội chúng tôi dựa thêm vào thông tin tác giả bài đăng và sự tương tác của người dùng với bài đăng.

Từ đó, chúng tôi chia bài toán phát hiện tin giả thành bốn bài toán nhỏ hơn:

- Phát hiện độ tin cậy của thông tin từ nội dung của bài đăng
- Tính điểm tin cậy người dùng
- Đo sự tương tác của người dùng với thông tin

Và cuối cùng làm kết hợp các thành phần với nhau để đưa và công cụ phát hiện tin giả chính xác nhất.

1.2.1 Phát hiện độ tin cậy của thông tin từ nội dung bài đăng

Nội dung bài đăng là thông tin quan trọng nhất để đánh giá một thông tin có đáng tin cậy hay không. Nhưng vì thế, nội dung cũng là thành phần khó xử lý nhất. Bởi vì như chúng

tôi đã nói ở phần trên, ngay cả con người cũng khó có thể phát hiện được một tin có đáng tin cậy hay không.

Một thông tin là có đáng tin cậy hay không thì ta phải dựa vào nội dung, từ ngữ, ngữ nghĩa của bài đăng có làm ảnh hướng đến cá nhân hay tổ chức nào không. Ngoài những thông tin không chính xác, những thông tin phản cảm, phản động, bạo lực, đồ trụy,.. cũng được xếp vào những thông không đáng tin cậy. Dựa vào những đặc tính đó của tin giả, chúng tôi sẽ cố gắng trích chọn, tìm kiếm những từ ngữ, thông tin không đáng tin cậy đó.

Phương pháp chúng tôi sử dụng để đánh giá thông tin là sử dụng mô hình Pre-train Bert và biến thể kết hợp với kỹ thuật thống kê số học TF-IDF để có được biểu diễn Vector phản ánh đúng nhất nội dung của thông tin. Sau đó, Vector biểu diễn nội dung thông tin sẽ được đưa qua thêm một Linear Projection Layer (cũng chính là Fully Connected Layer) ở cuối để phân loại. Nội dung chi tiết phương pháp sẽ được trình bày ở Chương III.

1.2.2 Tính điểm tin cậy người dùng

Chúng ta rất khó để phán đoán nội dung của một thông tin là thật hay giả. Vì vậy, phải dựa vào những thông tin khác để phân loại thông tin. Có thể dễ dàng thấy rằng những thông tin sai lệch thường được đăng bởi một nhóm người dùng thiếu hiểu biết đăng thông tin giật gân nhằm mục đích câu like, câu view.

Chúng tôi thực hiện tính điểm tin cậy của người dùng bằng thuật toán học máy thống kê. Đếm số lần người dùng đăng thông tin giả trên cơ sở dữ liệu. Từ đó có được điểm ti cậy người dùng.

1.2.3 Sự tương tác của người dùng với thông tin

Mạng xã hội là nơi mọi người trò chuyện, trao đổi, tương tác với nhau. Sự tương tác thể hiện mức độ lan tỏa của thông tin trên mạng xã hội, vì vậy sự tương tác cũng sẽ là một thuộc tính đóng góp để phát hiện thông tin giả.

CHƯƠNG 2. MÔ HÌNH NGÔN NGỮ BERT CHO BÀI TOÁN PHÂN LỚP

2.1 Phương pháp Transformer

2.1.1 Encoder và Decoder trong BERT

Trước khi tìm hiểu về BERT, chúng ta nhìn lại về Transformer. Đây là một lớp mô hình seq2seq gồm 2 phrase encoder và decoder. Mô hình hoàn toàn không sử dụng các kiến trúc Recurrent Neural Network của RNN mà chỉ sử dụng các layers attention để embedding các từ trong câu. Kiến trúc cụ thể của mô hình như sau:

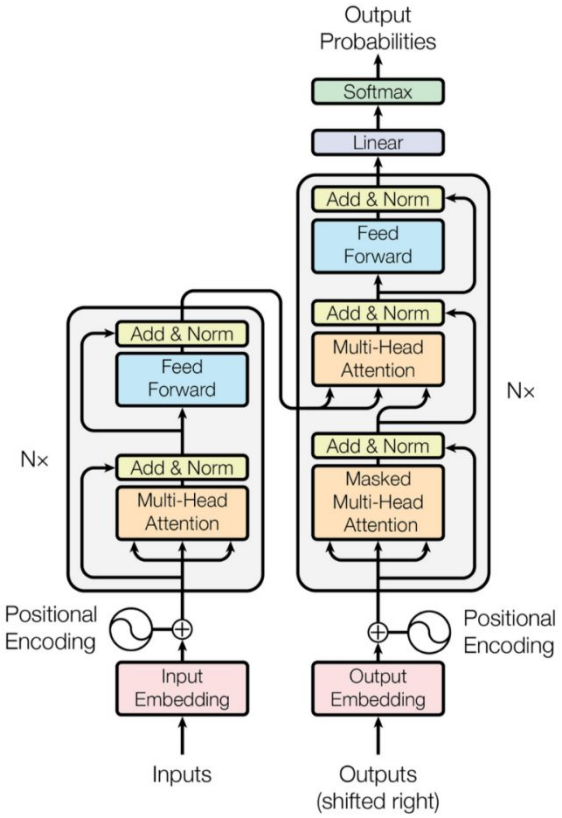


Figure 1: The Transformer - model architecture.

Hình 1: Sơ đồ kiến trúc transformer kết hợp attention

Mô hình sẽ bao gồm 2 phase:

- Encoder: Bao gồm 2 layers liên tiếp nhau. Mỗi một layer sẽ bao gồm một sub-layer là Multi-Head Attention kết hợp với fully-connected layer như mô tả ở nhánh encoder bên trái của hình vẽ. Kết thúc quá trình encoder ta thu được một vector embedding output cho mỗi từ.

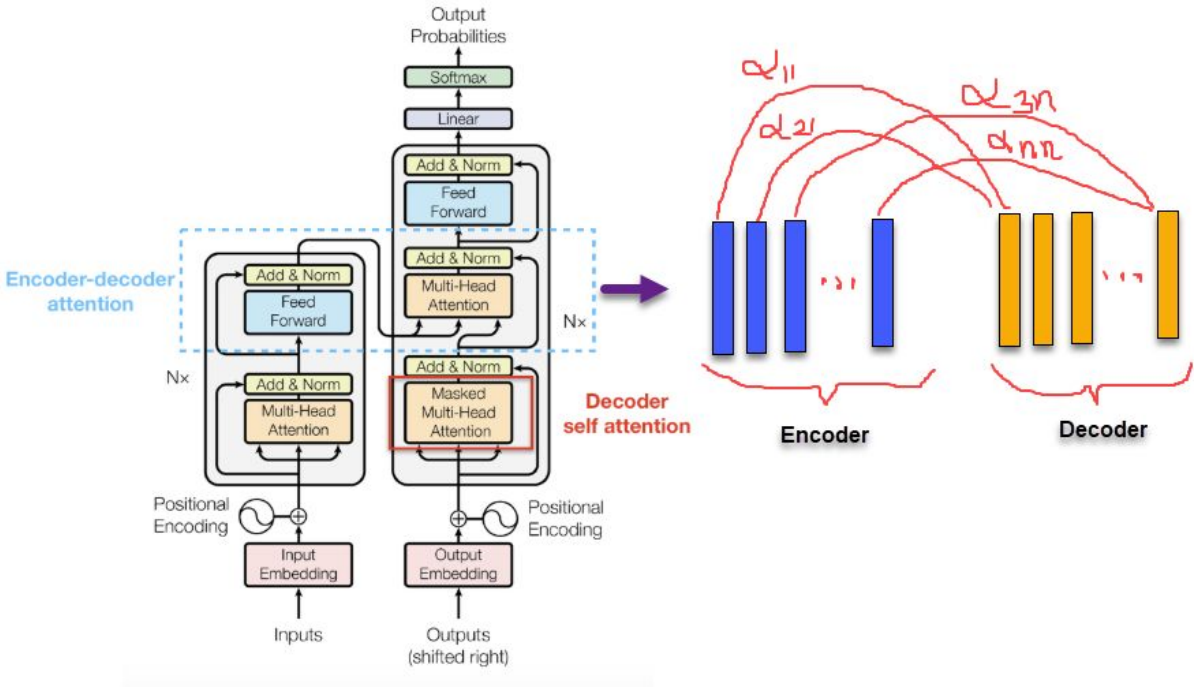
- Decoder: Kiến trúc cũng bao gồm các layers liên tiếp nhau. Mỗi một layer của Decoder cũng có các sub-layers gần tương tự như layer Encoder như bổ sung thêm sub-layer đầu tiên là Masker Multi-Head Attention có tác dụng loại bỏ các từ trong tương lai khỏi quá trình attention.

2.1.2 Các tiến trình self-attention và encoder-decoder attention

Trong kiến trúc transformer chúng ta áp dụng 2 dạng attention khác nhau tai từng bước huấn luyện.

Self-Attention: Được sử dụng trong cùng một câu input, tại encoder hoặc tại decoder. Đây chính là attention được áp dụng tại Multi-Head Attention ở đầu của cả 2 phase encoder và decoder.

Encoder-Decoder Attention:



Hình 2: Thể hiện cách tính trọng số attention khi kết hợp mỗi vector embedding ở decoder với toàn bộ các vector embedding ở encoder

Sở dĩ được gọi là encoder-decoder attention vì đây là kiến trúc attention tương tác giữa các vector embedding của encoder và decoder. Vector context được tính toán trên encoder đã được tính tương quan với vector decoder nên sẽ có ý nghĩa giải thích bối cảnh

của từ tại vị trí time step decoder tương ứng. Sau khi kết hợp giữa các vector context và vector decoder ta sẽ chiếu qua một Fully Connected Layer để tính phân phối xác suất cho output.

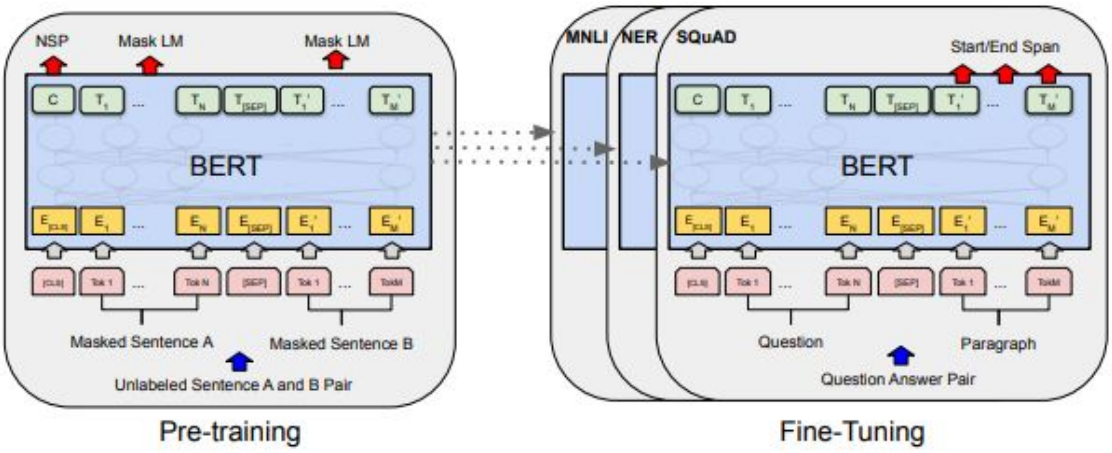
2.2 Giới thiệu về BERT

BERT là viết tắt của cụm từ Bidirectional Encoder Representation from Transformer có nghĩa là mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer. BERT được thiết kế để huấn luyện trước các biểu diễn từ (pre-train-word embedding). Điểm đặc biệt ở BERT đó là nó có thể điều hòa cân bằng bối cảnh theo cả 2 chiều trái và phải.

Cơ chế attention của Transformer sẽ truyền toàn bộ các từ trong câu văn đồng thời vào mô hình một lúc mà không cần quan tâm đến chiều của câu Do đó Transformer được xem như là huấn luyện hai chiều (bidirectional) mặc dù trên thực tế chính xác hơn chúng ta có thể nói rằng đó là huấn luyện không chiều (non-directional). Đặc điểm này cho phép mô hình học được bối cảnh của từ dựa trên toàn bộ các từ xung quanh nó bao gồm cả từ bên trái và từ bên phải.

2.2.1 Fine-tuning model Bert

Một điểm đặc biệt ở BERT mà các model embedding trước đây chưa từng có đó là kết quả huấn luyện có thể fine-tuning được. Chúng ta sẽ thêm vào kiến trúc model một output layer để tùy biến theo các tác vụ huấn luyện.



Hình 3: Toàn bộ tiến trình pre-training và fine-tuning của BERT.

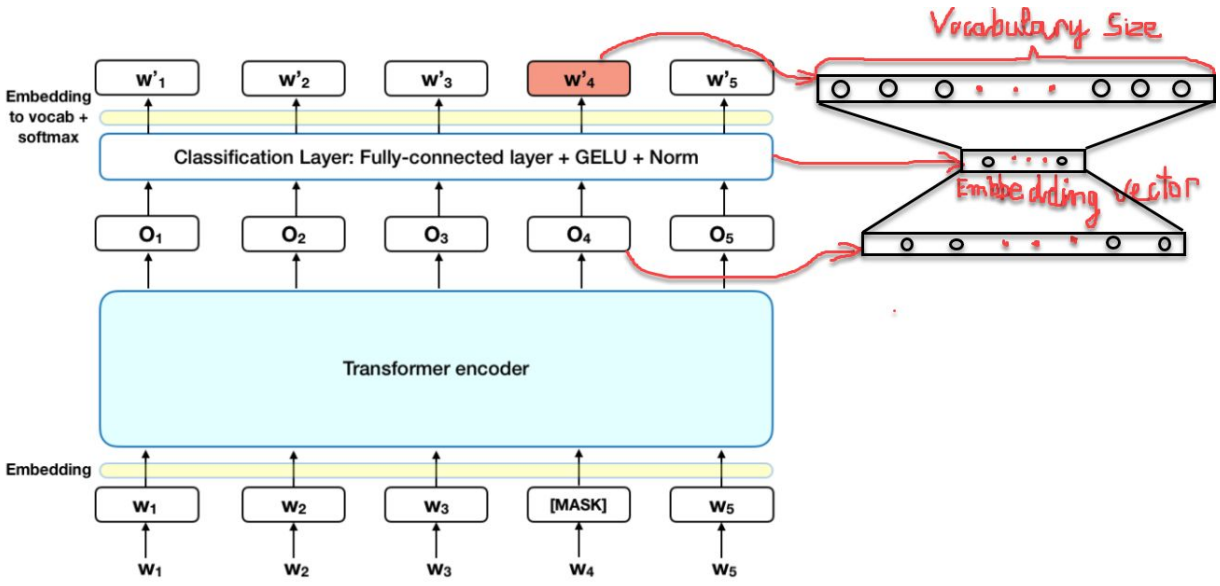
Một kiến trúc tương tự được sử dụng cho cả pretrain-model và fine-tuning model. Chúng ta sử dụng cùng một tham số pretrain để khởi tạo mô hình cho các tác vụ down stream

khác nhau. Trong suốt quá trình fine-tuning thì toàn bộ các tham số của layers học chuyển giao sẽ được fine-tune.

2.2.2 Masked Language Model

Masked Language Model (MLM) là một tác vụ cho phép chúng ta fine-tuning lại các biểu diễn từ trên các bộ dữ liệu unsupervised-text bất kỳ. Chúng ta có thể áp dụng MLM cho những ngôn ngữ khác nhau để tạo ra biểu diễn embedding cho chúng. Các bộ dữ liệu của tiếng anh có kích thước lên tới vài trăm tới vài nghìn GB được huấn luyện trên BERT đã tạo những kết quả khác ấn tượng.

Bên dưới là sơ đồ huấn luyện BERT theo tác vụ Masked LM.



Hình 4: Sơ đồ kiến trúc BERT cho tác vụ Masked LM

Theo đó:

- Khoảng 15% các token của câu input được thay thế bởi [MASK] token trước khi chuyển vào model đại diện cho những từ bị che dấu (masked). Mô hình sẽ dựa trên các từ không được che (non-masked) dấu xung quanh [MASK] và đồng thời là bối cảnh của [MASK] để dự báo giá trị gốc của từ được che dấu. Số lượng từ được che dấu được lựa chọn là một số ít (15%) để tỷ lệ bối cảnh chiếm nhiều hơn (85%).
- Bản chất của kiến trúc BERT vẫn là mô hình seq2seq gồm 2 phase encoder giúp embedding các từ input và decoder giúp tìm ra phân phối xác suất của các từ output. Kiến trúc Transformer encoder được giữ lại cho tác vụ Masked LM.

Sau khi thực hiện self-attention và feed-forward ta sẽ thu được các vector embedding ở output.

- Để tính toán phân phối xác suất cho từ output, chúng ta thêm một Fully connected layer ngay sau Transformer Encoder. Hàm softmax có tác dụng tính toán phân phối xác suất. Số lượng units của Fully connected layer phải bằng kích thước của từ điển.
- Cuối cùng ta thu được vector nhúng của mỗi một từ tại vị trí MASK sẽ là embedding vector giảm chiều của vector O_i sau khi đi qua Fully connected layer như mô tả trên hình vẽ bên phải.

Hàm loss function của BERT sẽ bỏ qua mất máy từ những từ không bị che dấu và chỉ đưa vào mất mát của những từ bị che dấu. Do đó mô hình sẽ hội tụ lâu hơn nhưng đây là đặc tính bù trừ cho sự gia tăng ý thức về bối cảnh. Việc lựa chọn ngẫu nhiên 15% số lượng các từ bị che dấu cũng tạo ra vô số các kịch bản input cho mô hình huấn luyện rất lâu mới học được toàn diện các khả năng.

2.2.3 Next Sentence Prediction (NSP)

Đây là một bài toán phân loại học có giám sát với 2 nhãn (hay còn gọi là phân loại nhị phân). Input đầu vào của mô hình là một cặp câu (pair-sequence) sao cho 50% câu thứ 2 được lựa chọn là câu tiếp theo của câu thứ nhất và 50% được lựa chọn một cách ngẫu nhiên từ bộ văn bản mà không có mối liên hệ gì với câu thứ nhất. Nhãn của mô hình sẽ tương ứng với IsNext khi cặp câu là liên tiếp hoặc NotNext nếu cặp câu không liên tiếp.

Thông tin Input được Preprocessing trước khi đưa vào mô hình huấn luyện bao gồm:

- Ngữ nghĩa của từ (token embeddings): Thông qua các embedding vector cho từng từ. Các vector được khởi tạo từ pretrain model.

Ngoài embedding biểu diễn từ của từ trong câu, mô hình còn embedding thêm một số thông tin:

- Loại câu (segment embeddings): Gồm hai vector là E_A nếu từ thuộc câu thứ nhất và E_B nếu từ thuộc câu thứ hai.
- Vị trí của từ trong câu (position embedding): là các vector E_0,…, $E_{10.}$ Tương tự như positional embedding trong transformer.

Vector input sẽ bằng tổng của ba thành phần embedding theo từ, câu và vị trị.

2.2.4 Các kiến trúc mô hình BERT

Hiện tại có nhiều phiên bản khác nhau của model BERT. Các phiên bản đều dựa trên việc thay đổi kiến trúc của Transformer tập trung ở 3 tham số ***L***: số lượng các block sub-layers trong Transformer, ***H***: kích thước của embedding vector (hay còn gọi là hidden size), ***A***: Số lượng head trong Multi-head layer, mỗi một head sẽ thực hiện một self-attention. Tên gọi của 2 kiến trúc bao gồm:

- **BERT$_{\text{BASE}}$** (***L*** = 12, ***H*** = 768, ***A*** = 12): Tổng tham số là 110 triệu
- **BERT$_{\text{LARGE}}$** (***L*** = 24, ***H*** = 1024, ***A*** = 16): Tổng tham số là 340 triệu

Như vậy ở kiến trúc BERT$_{\text{LARGE}}$ chúng ta tăng gấp đôi số layer, tăng kích thước hidden size của embedding vector gập 1.33 lần và tăng số lượng head trong Multi-head layer gấp 1.33 lần.

CHƯƠNG 3. MÔ HÌNH GIẢI QUYẾT BÀI TOÁN

3.1 Mô hình giải quyết bài toán

Quy trình xây dựng mô hình bao gồm 4 bước chính:

- Thu thập dữ liệu
- Tiền xử lý dữ liệu
- Xây dựng mô hình
- Thực nghiệm và đánh giá

Trong quá trình xây dựng, các bước được thực hiện lặp đi lặp lại cho đến khi đạt được kết quả tốt nhất.

3.2 Các bước thực hiện

3.2.1 Thu thập dữ liệu

Mạng xã hội là một thế giới thông tin rộng lớn có đầy đủ các loại thông tin. Nhưng việc thu thập dữ liệu gặp rất nhiều khó khăn bởi vì vấn đề bản quyền và an ninh từ các nhà cung cấp mạng xã hội. Việc gán nhãn cho dữ liệu sau khi được thu thập về cũng gặp nhiều khó khăn bởi vì có những thông tin mà con người cũng khó để đánh giá.

Chúng tôi sử dụng dữ liệu từ cuộc thi Reliable Intelligence Identification on Vietnamese SNSs (ReINTEL) xác định độ tin cậy của thông tin trên mạng xã hội Việt Nam. Dữ liệu là những bài đăng trên mạng xã hội của người dùng Việt Nam đã được gán nhãn bởi các chuyên gia. Dữ liệu thu thập được bao gồm 5172 bản ghi đã được gán nhãn 0 và 1 (0 là tin thông tin đáng tin cậy và 1 là thông tin không đáng tin cậy). Dữ liệu được biểu diễn trên 8 thuộc tính bao gồm:

- Id: Mã id cho mỗi bài đăng trên mạng xã hội
- User_name: Id người dùng
- Post_message: Nội dung bài đăng trên mạng xã hội
- Timestamp_post: Thời gian bài đăng được đăng
- Num_like_post: Số lượng lượt thích của bài đăng
- Num_comment_post: Số lượng lượt bình luận của bài đăng
- Num_share_post: Số lượng lượt chia sẻ của bài đăng

Dữ liệu thu thập về được lưu dưới định dạng csv, chúng tôi thực hiện đọc dữ liệu bằng thư viện pandas DataFrame và thực hiện tiền xử lý dữ liệu.

3.2.2 Tiền xử lý dữ liệu

Sau khi khảo sát và đánh giá dữ liệu, nhận thông dữ liệu mang rất nhiều thông tin bị lỗi và thiếu, chúng tôi thực hiện quá trình tiền xử lý với các trường hợp như dưới đây.

a. Lựa chọn thông tin có giá trị trong dữ liệu

Như đã trình bày ở phần thu thập dữ liệu, dữ liệu gồm 8 thuộc tính nhưng không phải tất cả đều mang giá trị để xác định tin giả. Vì vậy, chúng tôi thực hiện đánh giá từng thuộc tính. Có thế dễ dàng thấy rằng trong bài toán phát hiện tin giả nội dung thông tin là quan trọng nhất xong như đã nói ở trên, ngay cả con người cũng gặp khó khăn khi đọc một thông tin và phân biệt đó là thật hay giả nên chúng tôi giữ lại các thuộc tính mã người dùng và tương tác trên mạng xã hội. Dữ liệu sau khi được lựa chọn thuộc tính mang giá trị còn lại các trường thông tin: Post_message, User_name, Num_like_post, Num_comment_post, Num_share_post.

b. Nội dung bản ghi không mang thông tin

Trong quá trình khảo sát, có rất nhiều bản ghi mang dữ [1] liệu lỗi mang giá trị null nên chúng tôi thực hiện xóa hết bản ghi null. Ngoài ra, trong trường dữ liệu nội dung bài đăng, nhiều bản ghi chỉ mang thông tin <URL> là đường dẫn đến thông tin, chúng tôi cũng thực hiện xóa các bản ghi chỉ mang thông tin <URL>.

c. Dữ liệu trùng lặp

Quá trình khảo sát chúng tôi còn thấy rằng nhiều bản ghi bị trùng lặp nhưng nhãn của chúng lại khác nhau, chúng tôi thực hiện loại bỏ đi các phần dữ liệu trùng lặp và gán lại nhãn cho chúng là 1.

d. Dữ liệu mất cân bằng

Dữ liệu gồm hơn năm nghìn bản ghi được gán nhãn là 0 và 1 nhưng chúng tôi nhận thấy dữ liệu nhãn là 1 chỉ có 934 (18.06%) bản ghi và dữ liệu nhãn 0 là 4238 (81.94%). Vì vậy sẽ dẫn đến việc thiên lệch và ảnh hưởng đến quá trình máy học. Để đơn giản, chúng tôi thực hiện downsample dữ liệu nhãn 0 sao cho dữ liệu học là không quá mất cân bằng.

e. Tiền xử lý mức thông tin chữ

Nhóm chúng tôi sử dụng pretrain model BERT để xử lý dữ liệu vì vậy để đưa dữ liệu chữ qua BERT cũng cần xử lý để BERT đạt được kết quả tốt nhất. Phần đầu tiên chúng tôi thực hiện bỏ các khoảng trắng, xóa đi các ký tự đặc biệt không mang thông tin, đưa về chữ thường. Nhận thấy trong dữ liệu nội dung bài đăng có rất nhiều từ viết tắt, chúng tôi khảo sát và tìm kiếm những từ viết tắt được sử dụng nhiều lần và xây dựng một dictionary chứa các từ viết tắt và chuyển chúng về từ gốc. Ngoài ra, trong tiếng Việt có

những từ mà dấu của nó bị sai vị trí ví dụ như "chúôi" thì dấu sắc bị đặt sai vị trí, chúng tôi cũng thực hiện đưa các từ bị sai về đúng. Cuối cùng là sử dụng Vncorenlp để làm tác vụ segmentation.

f. Tiền xử lý mức thông tin số

Trong trường dữ liệu kiểu số có nhiều bản ghi có chứa phần chữ, chúng tôi loại bỏ phần chữ và dữ liệu phần số của dữ liệu. Ngoài ra, một số bản ghi unknown chúng tôi chuyển cũng thực hiện chuyển về 0.

Sau khi thực hiện các bước như trên chúng tôi chia dữ liệu thành 3 phần để thực hiện xây dựng model, 70% dữ liệu sẽ được đưa vào model để train, 10% cho tập validate và 20% dùng để test dữ liệu.

3.2.3 Xây dựng mô hình xử lý dữ liệu chữ

Như phần giới thiệu về bài toán, chúng tôi đã đưa ra hướng tiệp cận của mình theo từng thành phần. Chúng tôi thực hiện từng phần tách biệt với nhau trước khi đưa vào kết hợp. Ở phần chữ, chúng tôi sử dụng mô hình pretrain BERT (trình bày ở chương 2) để thực hiện xử lý dữ chữ, cụ thể ở đây là PhoBERT cho ngôn ngữ là tiếng Việt.

Sử dụng pretrain PhoBERT tokenizer đưa dữ liệu dạng chữ về vector chiều dài 128, vector qua BERT thể hiện biểu diễn của từ qua ngữ nghĩa. Sau đó, tiếp tục cài đặt TF-IDF biểu diễn các từ theo dạng thống kê với chiều dài 128. Cuối cùng, nối hai vector bên trên lại đưa vào model Roberta cho tác vụ phân lớp cụ thể ở đây là RobertaForSequenceClassification.

Thực hiện cài đặt trên framework Pytorch. Quá trình huấn luyện một model phân lớp trên Pytorch sẽ bao gồm những bước chính sau đây:

- Khởi tạo DataLoader để quản lý dữ liệu đưa vào huấn luyện và thẩm định.
- Thiết lập kiến trúc mô hình
- Khai báo hàm loss function
- Phương pháp optimization giúp tối ưu loss function.
- Huấn luyện mô hình qua các epochs.

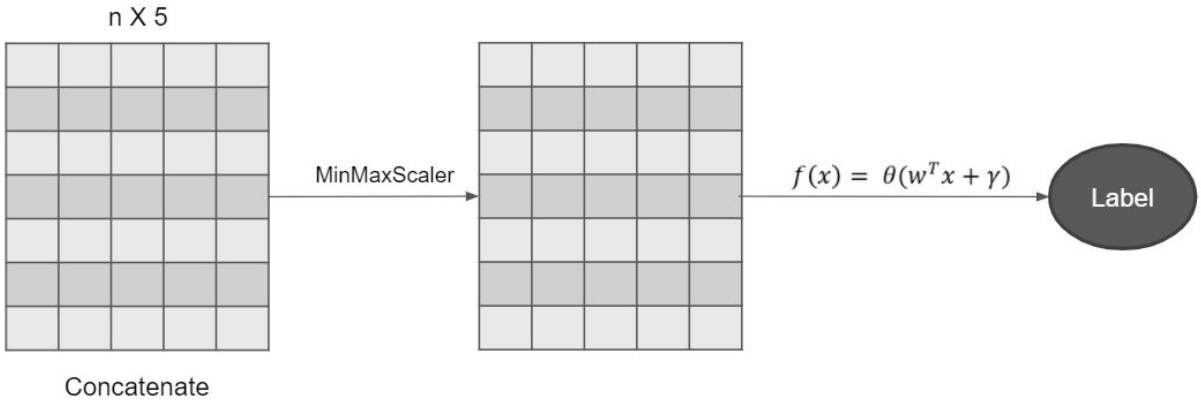
Chúng tôi thực hiện cài đặt và thực nghiệm trên môi trường Colab với GPU miễn phí của Google .Huấn luyện trên 10 epochs và thấy hàm loss validation giảm dần gần với 0.

3.2.4 Xây dựng mô hình tính điểm tin cậy của người dùng

Chúng tôi thực hiện tính điểm tin cậy của người dùng bằng việc thống kê số lần người đó đăng tin giả trên cơ sở dữ liệu. Sử dụng mã người dùng xây dựng bộ dữ liệu lưu thông tin người dùng và số lần đăng tin không chính xác.

3.2.5 Xây dựng mô hình kết hợp

Sau khi có vector chạy qua từng thành phần ở bên trên chúng tôi kết hợp từng phần với nhau, sử dụng MinMaxScaler rồi đưa qua mô hình Logistic Regression.



Hình 5: Mô tả mô hình kết hợp

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Kết quả thực nghiệm

Với bài toán phát hiện tin giả chúng tôi sử dụng hai độ đo chính để đánh giá là Accuracy và F1-score (weighted avg).

Kết quả huấn luyện phần nội dung bài đăng:

	Accuracy	F1-score
TF-IDF + RandomForest	0.70	0.73
TF-IDF + SVM	0.85	0.86
TF-IDF + Logistic Regression	0.80	0.80
Pretrain PhoBERT + RobertaForSequenceClassification	0.89	0.89
Pretrain PhoBERT + TF-IDF + RobertaForSequenceClassification	0.90	0.90

Bảng 4-1: Kết quả huấn luyện phần nội dung bài đăng

Đối với dữ liệu số, nhóm chúng tôi thử với một số thuật toán học máy cơ bản dưới đây.

	Accuracy	F1-score
KNN	0.65	0.69
SVM	0.82	0.78
DecisionTree	0.65	0.69
Logistic Regression	0.82	0.79

Sau khi huấn luyện phần nội dung chúng thấy rằng Pretrain PhoBERT + IF-IDF đi qua RobertaForSequenceClassification cho kết quả tốt nhất. Vì vậy, chúng tôi lấy mô hình huấn luyện kết hợp với những phần khác và cho ra kết quả.

	Accuracy	F1-score
Combine Naive Bayes	0.87	0.87
Combine SVM	0.89	0.89
Combine Logistic Regression	0.90	0.90

Bảng 4-2: Kết quả huấn luyện kết hợp

4.2 Đánh giá kết quả

Đầu tiên nhìn vào kết quả của mô hình huấn luyện với phần nội dung, có thể thấy TF-IDF + SVM cũng đem lại kết quả khá cao vì bản thân TF-IDF lấy được những đặc trưng của thông tin. Khi huấn luyện dữ liệu trên pretrain BERT cho kết quả cao hơn vì biểu diễn được ngữ nghĩa của thông tin. Khi kết hợp cả TF-IDF và pretrain BERT, ta lấy được cả hai thông tin đặc trưng và ngữ nghĩa của nội dung nên kết quả được đẩy lên cao hơn.

Về phần kết hợp nội dung và các thành phần điểm người dùng và độ tương tác kết quả không cao hơn so với huấn luyện riêng phần nội dung. Có thể dữ liệu điểm người dùng và độ tương tác không mang nhiều thông cho việc phân lớp dữ liệu. Khi chạy kết hợp qua Naive Bayes và SVM, kết quả còn bị giảm đi nhưng có thể thấy nó đoán tốt hơn với dữ liệu nhãn 0. Điều đó có thể lý giải bằng việc mất cân bằng dữ liệu nên mô hình bị thiên lệch về dữ liệu 0 nhiều hơn. Dù vậy, khi đưa qua Logistic Regression, kết quả không bị giảm so với phần huấn luyện nội dung bài đăng.

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Sau thời gian tìm hiểu và thực hiện đề tài trong thời gian khoảng một tháng, chúng tôi đã đạt được kết quả khá tốt cho bài toán phát hiện tin giả. Mô hình pretrain BERT phân biệt được tốt với những thông tin phản cảm, bạo lực, mang tính thù địch. Nhưng với thông sai không chính xác, mô hình vẫn chưa cho kết quả tốt. Việc kết hợp huấn luyện thêm trên điểm tin cậy người dùng và độ tương tác xã hội không mang lại nhiều thông tin. Cũng dễ hiểu vì dữ liệu trong cơ sở dữ liệu người dùng rất ít nên không mang lại nhiều giá trị và cải thiện được việc dự đoán. Độ tương tác xã hội chỉ thể hiện mức độ ảnh hưởng của tin tức trên mạng xã hội không mang nhiều giá trị trong bài toán phân loại.

Trong phạm vi môn học xử lý ngôn ngữ tự nhiên và trong khoản thời gian khá ngắn nhóm đã được kết quả khá thành công với bài toán của mình. Tuy nhiên, để đưa bài toán áp dụng vào thực tế cần cải thiện thêm rất nhiều. Sau dự án lần này, chúng tôi đã học hỏi được rất nhiều kiến thức trong xử lý ngôn ngữ cũng như xây dựng mô hình huấn luyện trên Pytorch. Học hỏi thêm được kỹ thuật trình bày cũng như làm việc nhóm.

5.2 Hướng phát triển trong tương lai

Trong tương lai để phát triển thêm dự án của mình, chúng tôi dự định sẽ cải thiện thêm mô hình Pretrain BERT+TF-IDF với việc tăng số chiều biểu diễn vector, tinh chỉnh tham số mô hình. Ngoài ra, cố gắng cũng tìm ra phương pháp kết hợp giữa nội dung, điểm người dùng. Phát triển thêm cơ sở dữ liệu người dùng để điểm tin cậy người dùng mang nhiều thông tin hơn. Khi mô hình đã đạt được kết quả tốt, chúng tôi dự định sẽ phát triển Extension giúp đánh giá thông tin trên mạng xã hội Facebook.

TÀI LIỆU THAM KHẢO

[1] J. Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for".

[2] N. Ruchansky, "CSI: A Hybrid Deep Model for Fake News Detection".

[3] K. Shu, "Fake News Detection on Social Media:".

[4] P. Đ. Khánh, "https://phamdinhkhanh.github.io/2020/06/04/PhoBERT_Fairseq.html#8-b%C3%A0i-to%C3%A1n-classification," [Online].

[5] "https://huggingface.co/transformers/v2.9.1/model_doc/roberta.html#robertaforsequenceclassification," [Online].

[6] "https://vlsp.org.vn/vlsp2020/eval/reintel," [Online].