UNIVERSITY OF SCIENCE

VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY

BACHELOR'S THESIS

VIETNAMESE SENTIMENT CLASSIFICATION USING DEEP LEARNING METHODS

Student: Le Huy Khiem - 1711135

Supervisor: Dr. Nguyen Thanh Binh

2021

ABSTRACT:

With the booming development of e-commerce platforms in many countries, there is a massive amount of customer review data for different products and services. Understanding customers' feedback on both current and new products can give online retailers the possibility to improve product quality, meet customers' expectations, and increase corresponding revenue. In this thesis, we investigate the Vietnamese sentiment classification problem on two datasets containing customers' reviews. We propose eight different approaches, including Bi-LSTM, Bi-LSTM + Attention, Bi-GRU, Bi-GRU + Attention, Recurrent CNN, Residual CNN, Transformer, and PhoBERT, and conduct all experiments on two datasets, AIVIVN 2019 and our dataset self-collected from multiple Vietnamese e-commerce websites. The experimental results show that all our proposed methods outperform the winning solution of the competition "AIVIVN 2019 Sentiment Champion" by a significant margin. Especially, Recurrent CNN has the best performance in comparison with other algorithms in terms of both AUC (98.48%) and F1-score (93.42%) in this competition dataset and also surpasses other techniques in the dataset we collected. Finally, we aim to publish our codes and these two data sets later to contribute to the current research community related to the field of sentiment analysis.

The thesis was extended as a research publication and has been accepted at the SoMeT Conference 2020 (rank B). We presented our publication on 23rd September 2020.

MỤC LỤC **3**

Từ viết tắt:

ANN	Artificial Neural Networks
ASR	Automatic Speech Recognition
Bi-LSTM	Bidirectional Long Short Term Memory
Bi-RNN	Bidirectional Recurrent Neural Network
CBOW	Continuous Bag Of Words
CNN	Convolutional Neural Networks
CRF	Conditional Random Fields
CV	Computer Vision
DNN	Deep Neural Network
DBN	Deep Belief Network
FL	Focal Loss
GRU	Gated Recurrent Unit
LM	Language Model
LSTM	Long Short Term Memory
NLP	Natural Language Processing
POS	Part-of-Speech
RNN	Recurrent Neural Network
BERT	Bidirectional Encoder Representations from Transformers
PhoBERT	Pre-training approach is based on RoBERTa

I. Giới thiệu đề tài

Ngày nay, phân loại văn bản (Text Classification) đã trở thành một trong những bài toán nền tảng và thiết yếu trong lĩnh vực nghiên cứu Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP),…. Phân loại sắc thái bình luận là một ứng dụng của bài toán phân loại văn bản, ở đó một bình luận về một sản phẩm sẽ được xác định sắc thái cảm xúc, tích cực hoặc tiêu cực. Gần đây, bài toán phân loại sắc thái bình luận không chỉ giới hạn ở sắc thái cảm xúc, mà còn tập trung vào những biểu cảm khác (thích, không thích, vui, không vui, tức giận, …) và thậm chí là ý định (quan tâm hay không quan tâm) của một bình luận. Bài toán này có rất nhiều ứng dụng thực tiễn, đặc biệt trong lĩnh vực Thương mại điện tử. Rất nhiều công ty lớn (Amazon, Tiki, Lazada, Shopee, …) đang không ngừng theo dõi phản hồi từ người dùng và khách hàng, họ sẵn chi những khoản tiền lớn để mua những công nghệ phục vụ cho mục đính phân loại sắc thái bình luận. Nhờ đó họ có thể trích xuất những thông tin có giá trị từ những bình luận hay những nhận xét về sản phẩm. Những bình luận, nhận xét này có thể được tận dụng để cung cấp cho các doanh nghiệp, nhãn hàng về những điểm mạnh, điểm yếu trong sản phẩm của họ.

Có rất nhiều nghiên cứu liên quan đến bài toán phân loại sắc thái bình luận, nhưng đa phần là nghiên cứu về ngôn ngữ Tiếng Anh, hiện tại đã có một vài nghiên cứu về bài toán này trong Tiếng Việt, đa phần lớp sử dụng phương pháp máy học truyền thống. Sơn và đồng nghiệp [1] thực hiện nghiên cứu trên dữ liệu các bình luận được thu thập từ Facebook, Phan và đồng nghiệp [2] sử dụng dữ liệu các bình luận về các địa điểm ăn uống ở Việt Nam hay Thúy và đồng nghiệp [3] sử dụng dữ liệu các bình luận về dịch vụ khách sạn ở Việt Nam. Gần đây, Phú và đồng nghiệp [4] so sánh phương pháp máy học truyền thống và học sâu cho bài toán phân loại sắc thái trên tập dữ liệu những phản hồi từ sinh viên, được thu thập tại một trường đại học ở Việt Nam.

Trong đề tài này, chúng tôi khám phá những cách tiếp cận mới và hiệu quả dựa trên phương pháp học sâu cho bài toán phân loại sắc thái bình luận trong ngôn ngữ Tiếng Việt. Chúng tôi sử dụng nhiều mô hình, bao gồm Bi-LSTM/GRU, Bi-LSTM/GRU kết hợp kỹ thuật Attention [5], Residual CNN [6], Transformer Encoder [7], Học chuyển tiếp và xây dựng các thí nghiệm trên hai tập dữ liệu chứa những bình luận của khách hàng từ các trang web thương mại điện tử ở Việt Nam. Tập dữ liệu thứ nhất được công bố và sử dụng trong cuộc thi Vietnam Sentiment Analysis Challenge 2019 được tổ chức bởi AIVIVN. Tập dữ liệu thứ hai được chúng tôi thu thập từ nhiều trang thương mai điện tử và gán nhãn dựa trên xếp hạng của bình luận. Kết quả thực nghiệm cho thấy những phương pháp chúng tôi đề xuất cho kết quả vượt trội hơn nhiều phương pháp đã giành chiến thắng trong cuộc thi Vietnam Sentiment Analysis Challenge 2019 trên.

II. Lý thuyết liên quan

1. Giới thiệu học sâu

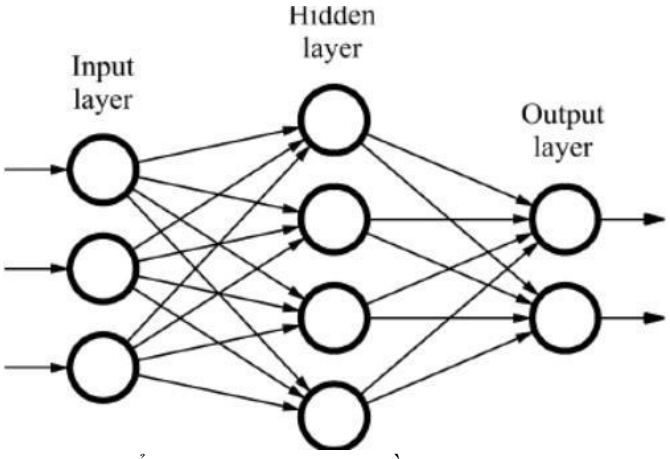
Học sâu là một nhánh quan trong của máy học, dạy máy tính làm những gì mà con người làm một cách tự nhiên: học từ các ví dụmà con người cung cấp cho máy tính. Học sâu là một công nghệ quan trọng đằng sau ô tô không người lái, cho phép chúng nhận ra biển báo dừng hoặc phân biệt người đi bộ với cột đèn. Hay đây là chìa khóa để điều khiển bằng giọng nói trong các thiết bị tiêu dùng như điện thoại, máy tính bảng, TV và loa rảnh tay. Học sâu đang được chú ý rất nhiều gần đây bởi nó đã và đang đạt được những kết quả mà trước đây không thể thực hiện được.

Trong học sâu, một mô hình máy tính học cách thực hiện các nhiệm vụ phân loại trực tiếp từ hình ảnh, văn bản hoặc âm thanh. Mô hình học sâu có thể đạt được độ chính xác hiện đại, đôi khi vượt quá hiệu suất ở cấp độ con người. Các mô hình được đào tạo bằng cách sử dụng một tập hợp lớn dữ liệu có nhãn và kiến trúc mạng nơ-ron chứa nhiều lớp.

Trong khi học sâu lần đầu tiên được đưa ra lý thuyết vào những năm 1980, có hai lý do chính khiến nó chỉ trở nên hữu ích gần đây:

- Học sâu yêu cầu một lượng lớn dữ liệu được gắn nhãn. Ví dụ: phát triển ô tô không người lái đòi hỏi hàng triệu hình ảnh và hàng nghìn giờ video.
- Học sâu đòi hỏi khả năng tính toán đáng kể. GPU hiệu suất cao có kiến trúc song song hiệu quả cho việc học sâu. Khi được kết hợp với các cụm hoặc điện toán đám mây, điều này cho phép các nhóm phát triển giảm thời gian đào tạo cho một mạng học sâu từ vài tuần xuống còn vài giờ hoặc ít hơn.

Hầu hết các phương pháp học sâu sử dụng kiến trúc mạng nơ-ron, đó là lý do tại sao các mô hình học sâu thường được gọi là mạng nơ-ron sâu. Thuật ngữ "sâu" thường đề cập đến số lượng các lớp ẩn trong mạng nơ-ron. Mạng nơ-ron truyền thống chỉ chứa 1-2 lớp ẩn, trong khi mạng sâu có thể có tới hàng trăm lớp.



Hình 1. Mạng nơ-ron, được tổ chức theo lớp bao gồm một tập hợp các nút kết nối với nhau.

2. Các lớp sử dụng

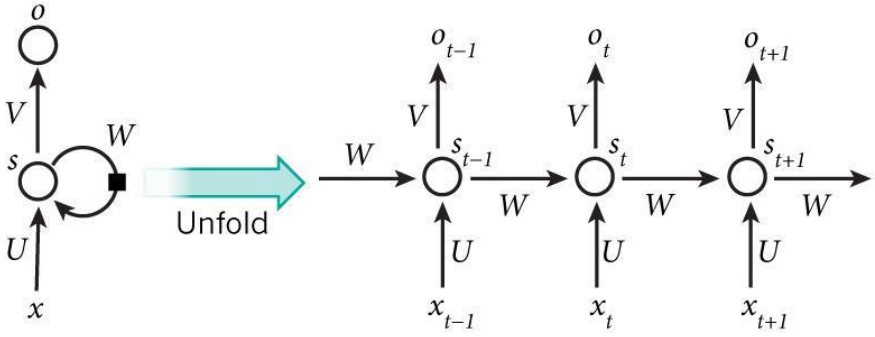
2.1. Nhúng từ

Trong các bài toán Xử lý ngôn ngữ tự nhiên (NLP) với dữ liệu đầu vào ở dạng văn bản, các thuật toán không thể nhận được đầu vào là chữ với dạng biểu diễn chữ cái(a,b,c,…) thông thường. Để máy tính có thể hiểu được, ta cần chuyển các từ trong ngôn ngữ tự nhiên về dạng mà các thuật toán có thể hiểu được ví dụ như dạng số.

Nhúng từ (word embedding) là một kỹ thuật cho việc học mật độ dày đặc thông tin đại diện của từ trong một không gian vector với số chiều khá lớn. Mỗi một từ có thể xem như là một điểm trong không gian này, được đại diện bởi một vector có độ dài cố định. Các vector từ được biểu diễn theo phương pháp nhúng từ này thể hiện được ngữ nghĩa của các từ, từ đó ta có thể nhận ra được mối quan hệ giữa các từ với nhau. Nhúng từ thường được thực hiện trong lớp đầu tiên của một mô hình, lớp này sẽ ánh xạ một từ (chỉ số index của từ trong từ điển từ vựng) trong từ điển sang một vector với kích thước đã cho. Một số phương pháp nhúng từ phổ biến là Word2vec và GloVe.

2.2. Lớp Recurrent và biến thể

Lớp Recurrent (Recurrent Neural Network -RNN) là một loại mạng nơ-ron hồi quy trong đó mỗi nút (node) trong các lớp ẩn có một kết nối với chính bản thân. Chính kết nối này tạo ra các trạng thái nội tại của kiến trúc mạng cho phép mô hình hoá các chuỗi với độ dài bất kỳ.



Hình 2. Minh họa mô hình RNN.

Một RNN có thể nhận vào một chuỗi có chiều dài bất kỳ và tạo ra một chuỗi nhãn có chiều dài tương ứng. Việc tính toán bên trong RNN được thực hiện như sau:

- x_t là chuỗi đầu vào tại t.
- U, W, V là các ma trận trọng số.
- s_t là trạng thái ẩn tại bước t.

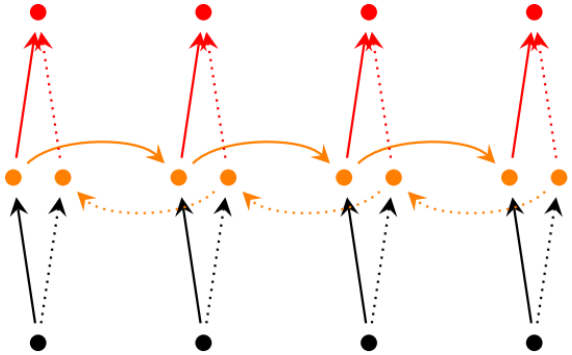
$$s_t = f(Ux_t + Ws_{t-1})$$

Trong đó f thường là một hàm phi tuyến tính như $tanh$

- o_t là đầu ra tại bước t.

$$o_t = softmax(Vs_t)$$

Dựa trên ý tưởng đầu ra (output) tại thời điểm *t* không chỉ phụ thuộc vào các thành phần trước đó mà còn phụ thuộc vào các thành phần trong tương lai. Ví dụ, để dự đoán một từ bị thiếu (missing word) trong chuỗi, ta cần quan sát các từ bên trái và bên phải xung quanh từ đó. Mô hình mạng nơ-ron hồi quy hai chiều gồm hai mạng nơ-ron hồi quy nạp chồng lên nhau. Trong đó, các trạng thái ẩn (hidden state) được tính toán dựa trên cả hai thành phần bên trái và bên phải của mạng.



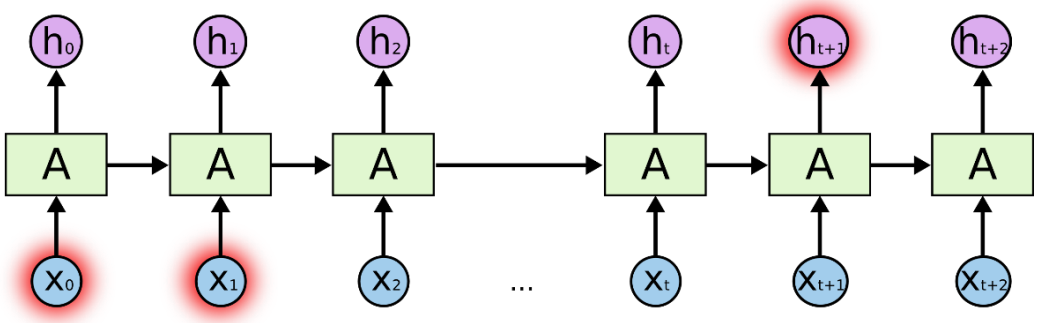
Hình 3. Minh họa mô hình mạng nơ-ron hồi quy 2 chiều.

2.2.1. Lớp LSTM

Một điểm nổi bật của mạng nơ-ron chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Thật không may là với khoảng cách càng lớn dần thì mạng nơ-ron bắt đầu không thể nhớ và học được nữa. Đây được gọi là vấn đề phụ thuộc xa (Long-term Dependency) của RNN.

Kiến trúc mạng nơ-ron hồi quy cổ điển rất khó để áp dụng trong thực tế vì vấn đề liên quan đến việc mất mát hoặc bùng nổ giá trị (Vanishing and Exploding gradient) được dùng để cập nhật các trọng số của mạng thông qua quá trình học khi phải mô hình hoá các chuỗi rất dài.

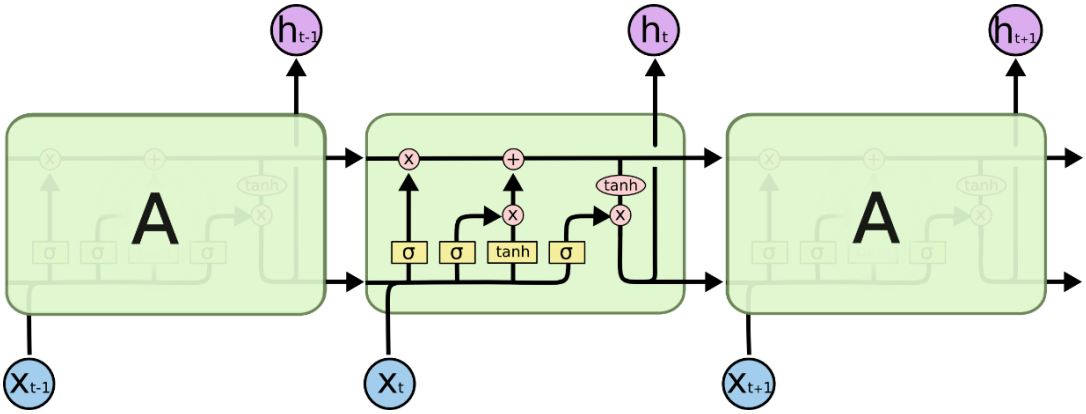
Khi đó, mạng bộ nhớ dài-ngắn(LSTM) [8] - một biến thể nổi bật của mạng nơ-ron hồi quy, được đề xuất như là một giải pháp cho vấn đề vừa được nêu ở trên.



Hình 4. Minh họa vấn đề phụ thuộc xa của RNN.

Mạng bộ nhớ dài-ngắn là một dạng đặc biệt của mạng nơ-ron hồi quy, nó có khả năng học được các phụ thuộc xa. Mạng bộ nhớ dài-ngắn được giới thiệu bởi Hochreiter và Schmidhuber (1997) và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần trở nên phổ biến như hiện nay.

Mạng bộ nhớ dài-ngắn được thiết kế để tránh được vấn đề phụ thuộc xa. Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, ta không cần phải huấn luyện chúng để có thể nhớ được. Tức là ngay bản thân chúng đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

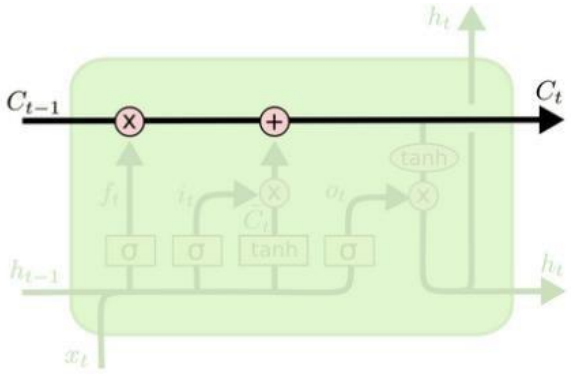


Hình 5. Minh họa mô hình LSTM.

Điểm chính trong kiến trúc mạng bộ nhớ dài-ngắn chính là các tế bào nhớ với các cổng cho phép lưu trữ hoặc truy xuất thông tin. Các cổng này cho phép ghi đè (Input gate), loại bỏ dư thừa (Forget gate) và truy xuất (Output gate) các thông tin được lưu trữ bên trong các memory cell.

- **Trạng thái tế bào (Cell state)**

Trạng thái tế bào là một dạng giống như băng chuyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi xuyên suốt mà không sợ bị thay đổi.

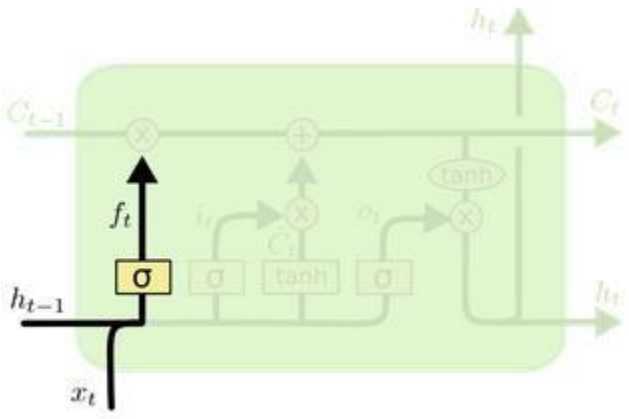


Hình 6. Trạng thái tế bào.

- **Tầng cổng quên (Forget gate layer)**

Tầng này quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là tầng cổng quên (Forget gate layer).

Tầng sẽ lấy đầu vào là h_{t-1} và x_t rồi đưa ra kết quả là một số trong khoảng [0,1] cho mỗi số trong trạng thái tế bào C_t. Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.

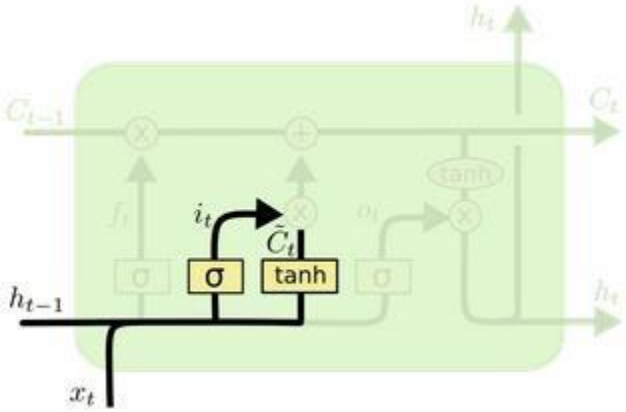


Hình 7. Tầng cổng quên.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]) + b_f$$

- **Tầng cổng vào (Input gate layer)**

Tầng cổng vào sử dụng một tầng sigmoid để quyết định giá trị nào ta sẽ cập nhật. Tiếp theo là một tầng tanh tạo ra một vector cho giá tị mới $\tilde{C}_t$. nhằm thêm vào cho trạng thái.



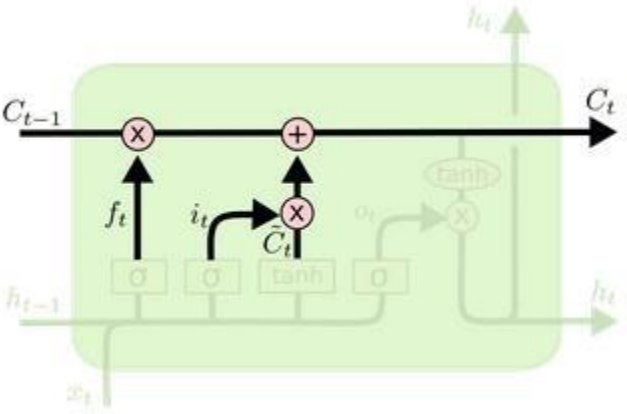
Hình 8. Tầng cổng vào.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) + b_i$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t]) + b_C$$

- **Cập nhật trạng thái tế bào (Update the cell state)**

Ta sẽ nhân trạng thái cũ với f_t để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm $i_t * \tilde{C}_t$.

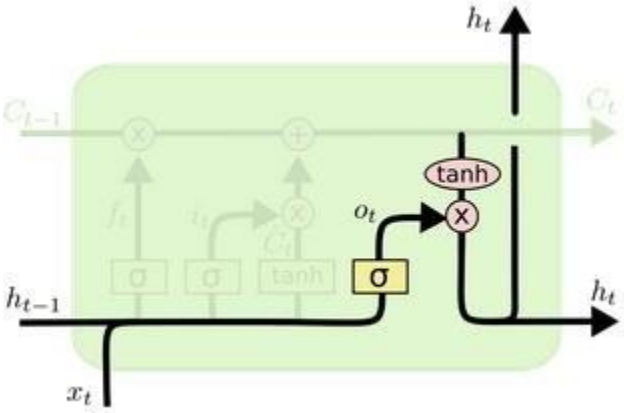


Hình 9. Cập nhật trạng thái tế bào.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

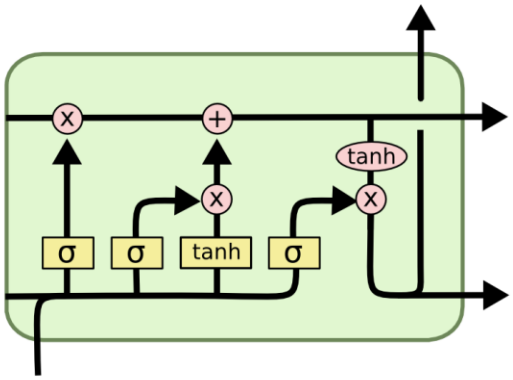
- **Tầng cổng ra (Output gate layer)**

Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa qua một hàm tanh để co giá trị nó về khoảng [-1,1]. Cuối cùng nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra mong muốn.



Hình 10. Tầng cổng ra.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) + b_o h_t$$
$$= o_t * \tanh\ (C_t)$$

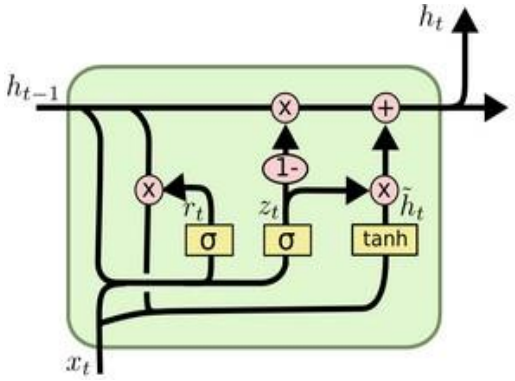


Hình 11. Kiến trúc LSTM đầy đủ.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]) + b_f$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) + b_i$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t]) + b_C$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) + b_o h_t$$
$$= o_t * \tanh\ (C_t)$$

2.2.2. Lớp GRU

Một biến thể khá đặc biệt của LSTM là GRU được giới thiệu bởi Cho và các cộng sự (2014). Nó kết hợp các cổng loại trừ và đầu vào thành một cổng "Cổng cập nhật" (update gate). Nó cũng kết hợp trạng thái tế bào và trạng thái ẩn với nhau tạo ra một thay đổi khác. Kết quả là mô hình của ta sẽ đơn giản hơn mô hình LSTM chuẩn và ngày càng trở nên phổ biến.



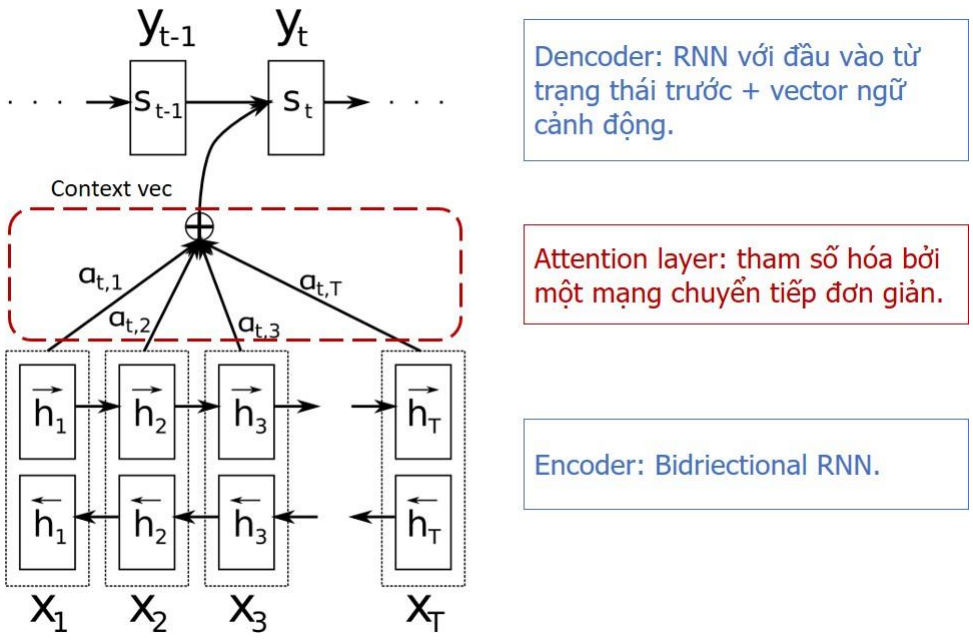
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Hình 12. Minh họa mô hình GRU.

2.3. Lớp Attention

Cơ chế Tập trung(Attention) là một cơ chế cho phép mô hình có thể tập trung học hiệu quả hơn đặc biệt với những chuỗi đầu vào có độ dài lớn. Phương pháp này đã dành được sự quan tâm lớn của cộng đồng nghiên cứu. Hệ thống dịch máy tự động của Google, Google Translate hiện đang áp dụng mô hình Sequence to sequence (seq2seq) hay tên gọi khác là Encoder-Decoder, với cơ chế Attention và cho chất lượng vượt trội so với những phương pháp trước kia.

Cơ chế Attention đã trở nên phổ biến trong những năm gần đây trong việc đào tạo mạng lưới thần kinh. Năm 2014, Bahdanau và các cộng sự đã đề xuất và áp dụng thành công cơ chế này để dịch và sắp xếp các từ cho nhiệm vụ dịch máy. Một cách lý tưởng, cơ chế này thường được áp dụng ở giữa các lớp của bộ mã hóa và bộ giải mã (Encoder - Decoder), nhằm tập trung có chọn lọc vào các phần của đầu ra lớp mã hóa, tương ứng với đầu vào của giải mã. Cụ thể, trong lớp Attention trước tiên lấy đầu ra của lớp mã hóa làm đầu vào để tính phân phối xác suất của đầu ra bộ mã hóa cho mỗi từ x_t ở bước t của lớp giải mã.



Hình 13. Mô hình Encoder-Decoder với cơ chế Attention.

Một cách cụ thể hơn về mô hình Encoder - Decoder với cơ chế Attention. Giả sử, chúng ta có một chuỗi đầu vào x có độ dài n và cố gắng xuất ra một chuỗi mục tiêu y có độ dài m:

$$x = [x_1, x_2, \ldots, x_n]$$

$$y = [y_1, y_2, \ldots, y_m]$$

Bộ Encoder là BiRNN với một trạng thái ẩn tới $\overrightarrow{h}_t$ và một trạng thái ẩn lùi $\overleftarrow{h}_t$.

$$h_i = [\overrightarrow{h}_t^T, \overleftarrow{h}_t^T]$$

$$i = 1, \ldots, n$$

Bộ Decoder có trạng thái ẩn $s_t = f(s_{t-1}, y_{t-1}, c_t)$ cho từ đầu ra ở vị trí $t, t = 1, \ldots, m$. Trong đó vector bối cảnh (context vector) c_t là tổng các trạng thái ẩn của chuỗi đầu vào, được tính theo điểm số căn chỉnh (alignment scores) $\alpha_{t,i}$.

$$c_t = \sum_{i=1}^{n} \alpha_{t,i} h_i$$

$$\alpha_{t,i} = align(y_t, x_i) = \frac{\exp(score(s_{t-1}, h_i))}{\sum_{j=1}^{n} \exp(score(s_{t-1}, h_j))}$$

Mô hình căn chỉnh (alignment model) gán một số điểm $\alpha_{t,i}$ cho cặp đầu vào ở vị trí i và đầu ra ở vị trí t, (y_t, x_i), dựa trên mức độ phù hợp. Tập hợp $\{\alpha_{t,i}\}$ là các trọng số xác định mức độ của mỗi trạng thái ẩn nguồn cần được xem xét cho mỗi đầu ra. Trong bài báo của

Bahdanau, điểm số căn chỉnh α được tham số hóa bởi một mạng chuyển tiếp (feed-forward network) với một lớp ẩn duy nhất và mạng này được đào tạo chung với các phần khác của mô hình. Hàm score được sử dụng như một hàm kích hoạt phi tuyến tính ví dụ như hàm tanh:

$$score(s_t, h_i) = v_\alpha{}^T \tanh\left(W_\alpha[s_t, h_i]\right)$$

Trong đó v_α và W_α là ma trận trọng số sẽ được học trong mô hình căn chỉnh.

2.4. Lớp Convolution

Lớp Mạng nơ-ron tích chập(Convolution-CNN) là một lớp của mạng nơ-ron sâu được sử dụng nhiều trong thị giác máy tính hoặc phân tích hình ảnh trực quan. Ngoài ra CNN còn được ứng dụng nhiều trong Xử lý ngôn ngữ tự nhiên.

Trong Xử lý ngôn ngữ tự nhiên, thay vì đầu vào là một ảnh được biểu thị dưới một ma trận điểm ảnh, đầu vào của mô hình mạng nơ-ron tích chập trong các bài toán Xử lý ngôn ngữ tự nhiên là các mệnh đề, các văn bản được biểu diễn như một ma trận. Mỗi dòng của một ma trận tương ứng với một mã, đa phần đó là một từ, nhưng nó cũng có thể là một kí tự. Mỗi hàng chính là một vector đại diện cho một từ. Thông thường các vector từ này được trình bày ở mức thấp như dạng Word2vec hay Glove, nhưng nó cũng có thể là một vector với việc các từ sẽ được đánh trọng số thuộc một bộ từ vựng.

3. Các độ đo hiệu suất

Khi xây dựng một mô hình Machine Learning, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không cũng như để so sánh khả năng của các mô hình.

3.1. Accuracy

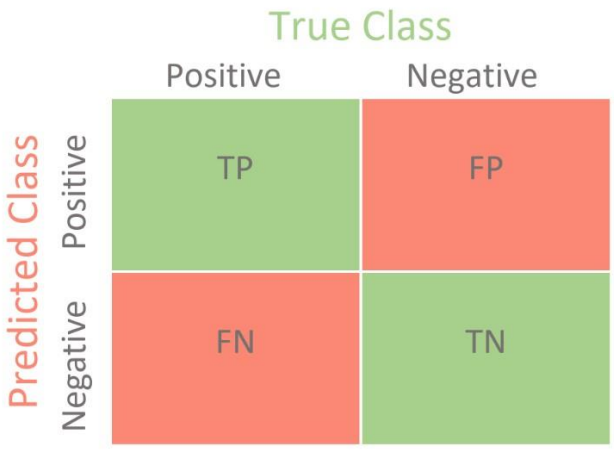
Accuracy (Độ chính xác) là độ đo đơn giản nhất để đánh giá một mô hình phân lớp. Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

3.2. F1 score

3.2.1. Precision và Recall

Với bài toán phân loại mà tập dữ liệu của các lớp là chênh lệch nhau rất nhiều, có một phép đó hiệu quả thường được sử dụng là Precision-Recall.

Trước tiên, ta định nghĩa *True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN)* dựa trên confusion matrix như sau:



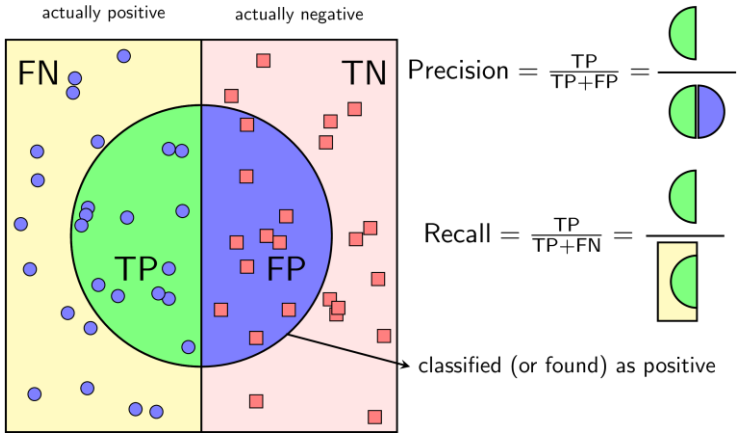
Hình 14. Confusion matrix.

- Precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive (TP + FP).
- Recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN).

Một cách toán học, Precision và Recall là hai phân số có tử số bằng nhau nhưng mẫu số khác nhau:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$



Hình 15. Cách tính Precision và recall.

Khi Precision = 1, mọi điểm tìm được đều thực sự là positive, tức không có điểm negative nào lẫn vào kết quả. Tuy nhiên, Precision = 1 không đảm bảo mô hình là tốt, vì câu hỏi đặt ra là liệu mô hình đã tìm được tất cả các điểm positive hay chưa. Nếu một mô

hình chỉ tìm được đúng một điểm positive mà nó chắc chắn nhất thì ta không thể gọi nó là một mô hình tốt. Khi Recall = 1, mọi điểm positive đều được tìm thấy. Tuy nhiên, đại lượng này lại không đo liệu có bao nhiêu điểm negative bị lẫn trong đó. Nếu mô hình phân loại mọi điểm là positive thì chắc chắn Recall = 1, tuy nhiên dễ nhận ra đây là một mô hình cực tồi.

Một mô hình phân lớp tốt là mô hình có cả Precision và Recall đều cao, tức càng gần một càng tốt. Có một cách đo chất lượng của bộ phân lớp dựa vào Precision và Reall: F1-score.

3.2.2. F1-score

F1-score, là harmonic mean của precision và recall (giả sử rằng hai đại lượng này khác không):

$$F1 = 2.\frac{precision * recall}{recision + recall}$$

F1-score có giá trị nằm trong nửa khoảng (0,1], F1 càng cao, bộ phân lớp càng tốt. Khi cả recall và precision đều bằng 1 (tốt nhất có thể), F1-score = 1.

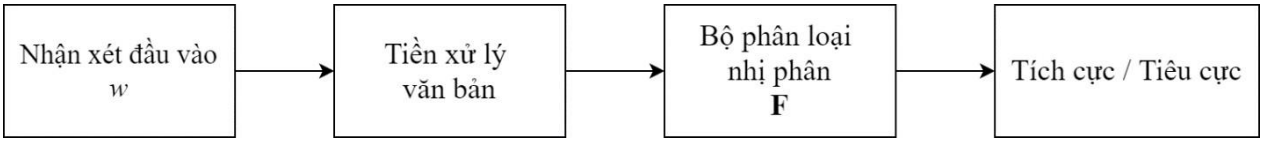
III. Phương pháp đề xuất

1. Phát biểu bài toán

Trong khuôn khổ đề tài này, chúng tôi tập trung nghiên cứu bài toán phân loại sắc thái bình luận như một bài toán phân loại nhị phân, cụ thể mỗi một bình luận Tiếng Viết sẽ được phân loại vào hai sắc thái cảm xúc, tích cực và tiêu cực. Ví dụ:

- Tích cực: *sản phẩm tốt tiki giao hàng nhanh tôi rất hài lòng.*
- Tiêu cực: *mới mua máy được 1 hôm dùng thử lần đầu thì lỗi luôn, lỗi e2 lại phải gửi máy đi sửa chữa ở tỉnh thì hết sức bất tiện, khá nản.*

Giả sử rằng *w* là một bình luận từ khách hàng về một sản phẩm, ký hiệu "0" là nhãn của sắc thái tích cực và "1" là nhãn của sắc thái tiêu cực. Bài toán được phát biểu: Xây dựng bộ phân loại nhị phân **F** để dự đoán sắc thái của *w*. Sơ đồ một hệ thống phân loại sắc thái bình luận được thể hiện trong Hình 16.



Hình 16. Sơ đồ hệ thống phân loại sắc thái văn bản

2. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là một trong những bước quan trọng nhất trong xử lý ngôn ngữ tự nhiên, đặc biệt với dữ liệu văn bản được thu thập từ các trang web thương mại điện tử. Trong tập dữ liệu chúng tôi thu thập, tồn tại nhiều câu, đoạn, từ ngữ không chính thống, không phù hợp với tiên chuẩn thông thường của Tiếng Việt. Do đó, tiền xử lý dữ liệu có thể giúp loại bỏ nhiễu trong dữ liệu. Trong đề tài này, với một bình luận đầu vào, chúng tôi áp dụng nhiều bước tiền xử lý trước khi thực hiện huấn luyện và đánh giá các mô hình học sâu.

Đầu tiên, chúng tôi viết thường tất cả các ký tự và sửa tất cả các từ bị kéo dài (ví dụ ***đẹpppppp quá***) về dạng đúng (***đẹp quá***). Sau đó, chúng tôi loại bỏ dấu câu, các ký tự đặc biệt, biểu tượng như ***! @ ? ( )***, chúng tôi cũng loại bỏ các chữ số vì gần như không đóng góp vào sắc thái cảm xúc của một bình luận. Mặc dù tất cả các bình luận được thu thập từ các trang web Tiếng Việt, vẫn tồn tại một tỉ lệ nhỏ các bình luận được viết trong ngôn ngữ khác, bao gồm Tiếng Anh, Tiếng Trung và Tiếng Hàn, do đó trong quá trình tiền xử lý dữ liệu, những bình luận này được loại bỏ. Cuối cùng, chúng tôi quan sát tồn tại nhiều từ viết tắt trong Tiếng Việt, những từ này có những đóng góp vào sắc thái của câu, do đó cần được thay thế về dạng chuẩn. Ví dụ, chúng tôi thay thế ***kp*** thành ***không phải*** hay ***ô kê*** thành ***ok***.

3. Mô hình dự đoán

Trong phần này, chúng tôi mô tả các phương pháp đề xuất cho bài toán phân loại sắc thái bình luận. Thông thường, có hai cách tiếp cận chính cho bài toán. Đầu tiên, dùng một phương pháp nhúng từ đã được huấn luyện (trong đề tài này, chúng tôi dùng Fasttext [9]) và đào tạo một mạng thần kinh học sâu thích hợp để tìm hiểu các đặc trưng ngữ nghĩa và phân loại bình luận. Chúng tôi khám phá 4 kiến trúc mạng nơ-ron khác nhau, bao gồm Bi-LSTM/GRU, Bi-LSTM/GRU kết hợp kỹ thuật Attention, Residual CNN, Transformer Encoder. Học chuyển tiếp là cách tiếp cận thứ hai, chúng tôi chọn một mô hình ngôn ngữ đã được huấn luyện, PhoBERT [10], cho ngôn ngữ Tiếng Việt và tinh chỉnh cho bài toán hiện tại. Với hiểu biết của chúng tôi, các kỹ thuật trên chưa được khám phá cho bài toán phân loại sắc thái bình luận trong Tiếng Việt.

3.1. Bi-LSTM/GRU

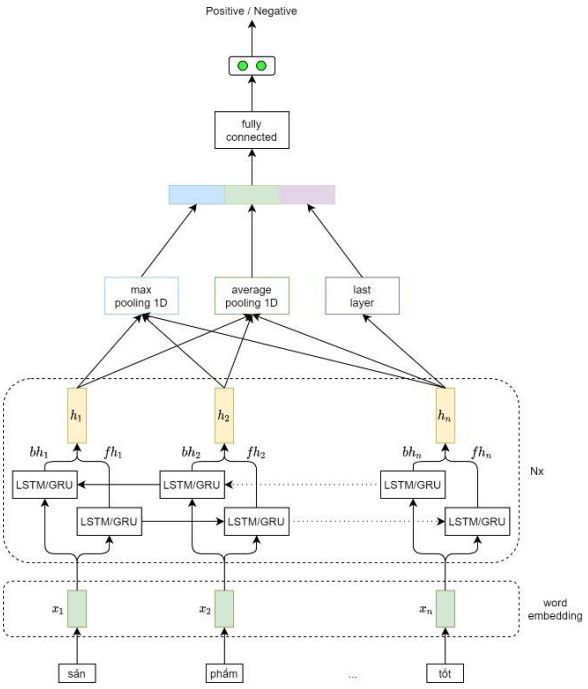
Mạng nơ-ron hồi quy hai hướng(Bidirectional Recurrent Neural Networks-BiRNNs) được sử dụng rộng rãi để giải quyết nhiều vấn đề trong Xử lý ngon ngữ tự nhiên. Cấu trúc đặc biệt của nó cho phép chúng ta nắm bắt được cả thông tin theo chiều xuôi và chiều ngược của một chuỗi. Hai biến thể của RNNs, Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU), được chứng minh nắm bắt thông tin dài hạn một cách hiệu quả hơn. Chúng tôi sử dụng cả Bi-LSTM và Bi-GRU như phương pháp cơ sở cho bài toán.

Giả sử rằng sau bước tiền xử lý dữ liệu, thu được n từ $\{w_1, w_2, \dots, w_n\}$ từ một bình luận đầu vào w. Ký hiệu $\{x_1, x_2, \dots, x_n\}$ là các w_i đã qua phương pháp word embedding và $\{h_1, h_2, \dots, h_n\}$ thể hiện các vector ẩn được tính bởi Bi-LSTM và Bi-GRU.

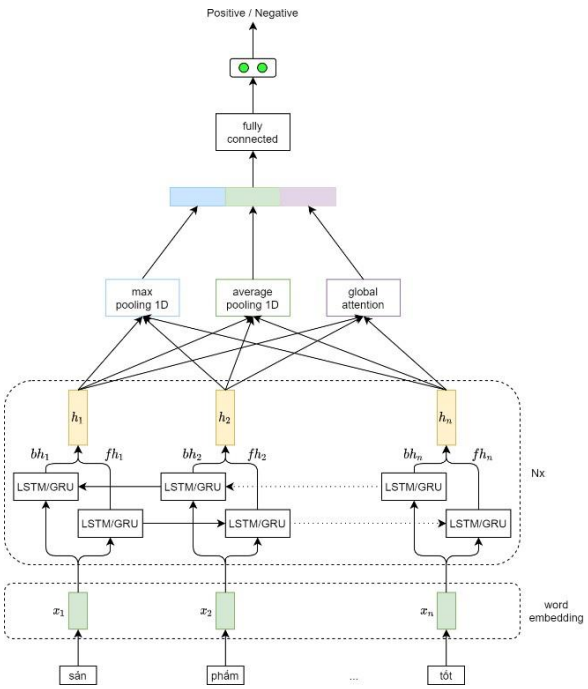
Không như các phương pháp dựa trên RNNs trước đây, chỉ sử dụng vector ẩn cuối cùng h_n để đưa vào một lớp tuyến tính, thu được vector h_{linear} và thực hiện phân lớp bằng một lớp tuyến tính cuối. Chúng tôi đề xuất một cách tiếp cận mới, sử dụng các vector ẩn $\{h_1, h_2, \dots, h_n\}$ đưa vào song song hai lớp Max Pooling 1D và Average Pooling 1D để thu được hai vector ẩn mới h_{max} và h_{avg}. Sau đó, chúng tôi nối ba vector $[h_{max}, h_{avg}, h_{linear}]$ để được vector ẩn cuối cùng và thực hiện phân lớp bằng một lớp tuyến tính cuối. Kiến trúc trên được mô tả trong Hình 17.

3.2. Bi-LSTM/GRU + Attention

Kiến trúc của Bi-LSTM/GRU + Attention mà chúng tôi đề xuất được thể hiện trong Hình 18. Điểm khác biệt với kiến trúc Bi-LSTM/GRU nằm ở việc chúng tôi dùng các vector ẩn $\{h_1, h_2, \dots, h_n\}$ đưa vào một lớp General Global Attention để thu được vector ẩn h_{att}. Sau đó, chúng tôi nối ba vector $[h_{max}, h_{avg}, h_{att}]$ để được vector ẩn cuối cùng và thực hiện phân lớp bằng một lớp tuyến tính cuối.



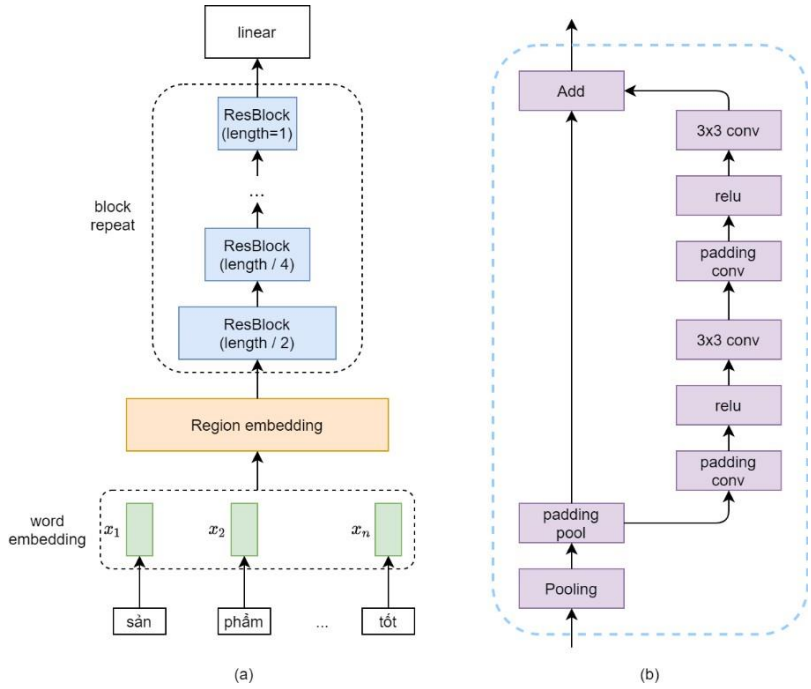
Hình 17. Kiến trúc Bi-LSTM/GRU



Hình 18. Kiến trúc Bi-LSTM/GRU + Attention

3.3. Residual CNN

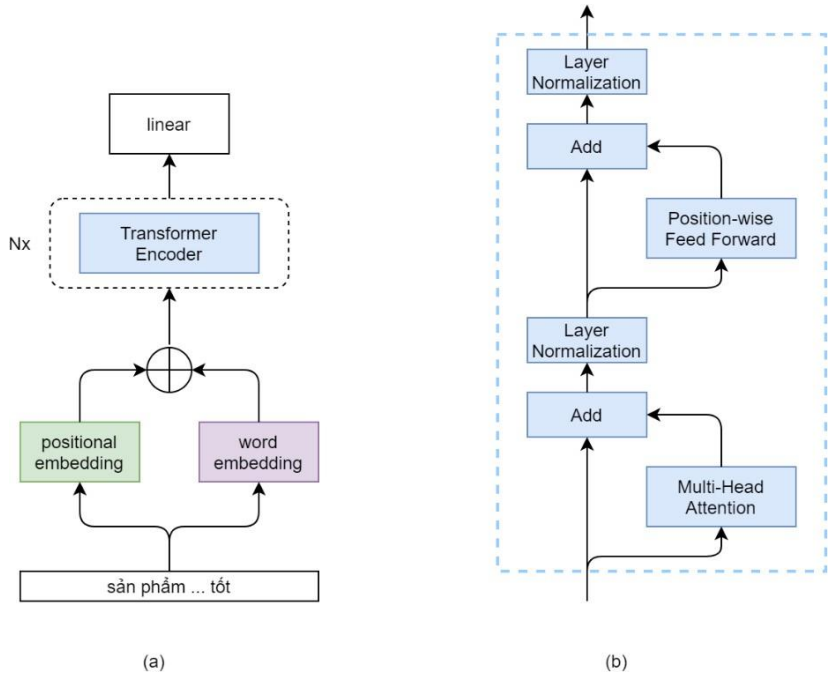
Hầu hết các cách tiếp cận thông thường cho bài toán phân loại văn bản đều dựa trên RNNs. Gần đây, các kiến trúc chỉ dựa trên CNNs với các residual block được khám phá và đạt được những kết quả hứa hẹn. Deep Pyramid CNN (DPCNN) là một trong những phương pháp thành công. Trong đề tài này, chúng tôi sử dụng DPCNN như một Residual CNN để xây dựng mô hình cho bài toán phân loại sắc thái bình luận. Kiến trúc chi tiết của Residual CNN được thể hiện trong Hình 19.



Hình 19. (a) Kiến trúc Residual CNN (b) Kiến trúc chi tiết Residual Block

3.4. Transformer Encoder

Gần đây, Transformer nổi lên là kiến trúc tiên tiến nhất đóng góp vào những kết quả tốt nhất trong Xử lý ngôn ngữ tự nhiên. Transformer cũng được tích hợp trong nhiều mô hình ngôn ngữ tiên tiến khác như BERT, GPT-2. Thông thường, kiến trúc Transformer gồm hai phần, Encoder và Decoder, chúng tôi chỉ sử dụng Encoder để giải quyết bài toán phân loại sắc thái. Trong phần Encoder, có sáu lớp Multi-Head Attention và Feed Forward, vector đầu ra của Encoder được thực hiện phân lớp bằng một lớp tuyến tính cuối. Kiến trúc chi tiết được thể hiện trong Hình 20.



Hình 20. (a) Kiến trúc Transformer Encoder (b) Một tầng trong Encoder

3.5. Học chuyển tiếp

Học chuyển tiếp là một khái niệm trong máy học tận dụng kiến thức đã thu được từ một vấn đề và áp dụng vào một vấn đề khác tương tự. Cụ thể, chúng ta sử dụng một mô hình đã được huấn luyện trên một khối lượng lớn dữ liệu sau đó huấn luyện lại, hay tinh chỉnh trên tập dữ liệu cụ thể. Kỹ thuật này đặc biệt hữu ích khi chúng ta không có một khối lượng lớn dữ liệu hay năng lực tính toán để huấn luyện một kiến trúc khổng lồ. Với bài toán phân loại sắc thái bình luận Tiếng Việt, chúng tôi áp dụng học chuyển tiếp bằng cách sử dụng PhoBERT để trích xuất vector đặc trưng của các bình luận đầu vào và thực hiện phân lớp bằng một lớp tuyến tính cuối.

IV. Thí nghiệm và kết quả

Trong phần này, chúng tôi mô tả chi tiết các tập dữ liệu, triển khai các phương pháp đề xuất và các kết quả thực nghiệm một cách chi tiết. Chúng tôi tiến hành các thi nghiệm trên một server với cấu hình Intel Core i9-7900X CPU, 128GB RAM, và 2 GPU RTX-2080Ti.

1. Tập dữ liệu

Để chứng minh tính hiệu quả của các phương pháp đề xuất cho bài toán phân loại sắc thái bình luận Tiếng Việt, chúng tôi thực nghiệm trên hai tập dữ liệu. Đầu tiên chúng tôi sử dụng tập dữ liệu công khai AIVIVN, chứa các bình luận từ người dùng trên các trang thương mại điện tử Việt Nam. Tập dữ liệu được sử dụng trong cuộc thi Vietnam Sentiment Analysis Challenge 2019, bao gồm 16,073 bình luận trên tập huấn luyện và 10,981 bình

luận trên tập đánh giá. Nhãn của bình luận trên tập đánh giá được gán cẩn thận bởi các thành viên trong nhóm. Bên cạnh đó, chúng tôi thu thập một tập dữ liệu lớn hơn từ nhiều nguồn và gán nhãn cẩn thận. Tập dữ liệu này chứa 358,743 câu bình luận tích cực và 100,699 câu bình luận tiêu cực. Tập huấn luyện và tập đánh giá được chia với tỉ lệ 7:3. Tập dữ liệu này là một đóng góp khác cho cộng đồng nghiên cứu NLP và sẽ được công khai. Chi tiết hai tập dữ liệu được mô tả trong Bảng 1.

AIVIVN	Positive	Negative
Train	8690	7383
Test	5767	5214

Our Dataset	Positive	Negative
Train	251120	70489
Test	107623	30210

Bảng 1. Tập dữ liệu AIVIVN và tập dữ liệu của chúng tôi

2. Triển khai

Chúng tôi triển khai các thí nghiệm bằng PyTorch, công cụ mạnh mẽ hỗ trợ triển khai các phương pháp học sâu. Ngoại trừ mô hình ngôn ngữ PhoBERT cho Tiếng Việt, các kiến trúc khác được thiết kế và triển khai từ đầu. Để sử dụng được PhoBERT cho bài toán, chúng tôi áp dụng phân đoạn từ Tiếng Việt RDRsegmenter để xử lý dữ liệu thô trước khi đưa vào mô hình PhoBERT. Fasttext word embedding cho phiên bản Tiếng Việt có số chiều embedding là 300 được sử dụng cho các phương pháp đề xuất, ngoại trừ phương pháp học chuyển tiếp. Tất cả các mô hình được huấn luyện bằng thuật toán tối ưu Adam, tốc độ học 0.001 trên 2 GPU. Tất cả các siêu tham số của các mô hình được sử dụng chung trên cả 2 tập dữ liệu. Chúng tôi đo lường hiệu suất của các mô hình bằng accuracy và F1 score.

3. Kết quả

Đầu tiên, một điều thú vị có thể thấy là đối với tập dữ liệu AIVIVN, tất cả các phương pháp đề xuất đều vượt trội đáng kể so với phương pháp đã giành chiến thắng trong cuộc thi Vietnam Sentiment Analysis Challenge 2019 với F1 score tốt nhất là 0.90012. Điều đáng nhận xét là phương pháp đã giành chiến thắng là ensemble của nhiều mô hình bao gồm TextCNN, VDCNN, HARNN, SARNN. Có thể thấy Residual CNN là kỹ thuật đạt được kết quả cao nhất với F1 score 0.92621. Điều này chứng minh hiệu quả tuyệt đối của các phương pháp đề xuất cho bài toán phân loại sắc thái bình luận trên tập dữ liệu AIVIVN. Ngoài ra, Bi-GRU cho kết quả tốt hơn Bi-LSTM. Bi-LSTM + Attention dẫn đến một chút cải thiện so với Bi-LSTM/GRU. Cuối cùng, sử dụng PhoBERT có thể cho kết quả tốt hơn phương pháp đã giành chiến thắng nhưng cho kết tệ nhất so với các phương pháp khác trên F1 socre.

Methods	**Accuracy**	**F1 score**
AIVIVN 2019 winner	_	0.90012
Bi-LSTM	0.9198	0.91599
Bi-LSTM + Attention	0.92182	0.91962
Bi-GRU	0.92463	0.92093
Bi-GRU + Attention	0.92071	0.91535
Residual CNN	**0.93037**	**0.92621**
Transformer	0.92091	0.9147
PhoBERT	0.9154	0.91264

Bảng 2. Kết quả thực nghiệm trên tập dữ liệu AIVIVN

Trên tập dữ liệu của chúng tôi, Bi-GRU + Attention vượt trội các phương pháp khác trên Accuracy và cả F1 socre. Khác với tập dữ liệu đầu tiên, cả Bi-LSTM + Attention, Bi-GRU + Attention đều cải thiện kết quả so với Bi-LSTM và Bi-GRU. Một điều thú vị, Transformer và PhoBERT cho kết quả thấp nhất trên cả 2 tập dữ liệu.

Methods	**Accuracy**	**F1 score**
Bi-LSTM	0.9154	0.80129
Bi-LSTM + Attention	0.91551	0.80361
Bi-GRU	0.91556	0.80634
Bi-GRU + Attention	**0.91618**	**0.80838**
Residual CNN	0.91207	0.78404
Transformer	0.89718	0.71807
PhoBERT	0.90477	0.75189

Bảng 3. Kết quả thực nghiệm trên tập dữ liệu của chúng tôi

V. Kết luận

Trong nghiên cứu này, chúng tôi đã khám phá các kỹ thuật khác nhau để giải quyết bài toán phân loại sắc thái bình luận trong ngôn ngữ tiếng Việt, bao gồm Bi-LSTM/GRU, Bi-

LSTM/GRU kết hợp kỹ thuật Attention, Residual CNN, Transformer Encoder, Học chuyển tiếp. Tất cả các phương pháp đề xuất đều đạt được những kết quả hứa hẹn trên cả 2 tập dữ liệu. Chúng tôi sẽ công khai mã nguồn và tập dữ liệu trong thời gian tới.

Chúng tôi sẽ mở rộng nghiên cứu này, phân loại các biểu cảm khác (thích, không thích, vui, không vui, tức giận, …) và thậm chí là ý định (quan tâm hay không quan tâm) của một bình luận. Bên cạnh đó, Bên cạnh đó, chúng tôi cũng có kế hoạch triển khai nghiên cứu này thành các ứng dụng thực tiễn (ứng dụng mobile, ứng dụng web, …).

VI. Tham khảo

[1] Son Trinh, Luu Nguyen, Minh Vo, and Phuc Do. Lexicon-Based Sentiment Analysis of Facebook Comments in Vietnamese Language, volume 642, pages 263–276. 02 2016.

[2] Dang-Hung Phan and Tuan-Dung Cao. Applying skip-gram word estimation and svm-based classification for opinion mining vietnamese food places text reviews. In Proceedings of the Fifth Symposium on Information and Communication Technology, SoICT ’14, page 232–239. Association for Computing Machinery.

[3] Thuy Nguyen-Thanh and Giang Tran. Vietnamese sentiment analysis for hotel review based on overfitting training and ensemble learning. 12 2019.

[4] Phu X. V. Nguyen, Tham T. T. Hong, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Deep learning versus traditional classifiers on vietnamese students’ feedback corpus. 5th NAFOSTED Conference on Information and Computer Science, 2018.

[5] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gul-cehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 2014.

[6] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 1:562–570, 07 2017.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.\

[8] Hochreiter S. and Schmidhuber J. Long short-term memory. Neural Computing, 9:1735–1780, 11 1997.

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, ‘‘Enriching word vectors with subword information,’’ Trans. Assoc. Comput. Linguistics, vol. 5, pp. 135–146, Dec. 2017.

[10] Dat Quoc Nguyen and Anh Tuan Nguyen. Phobert: Pre-trained language models for vietnamese. 2020.