ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

---------o0o---------

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN VIỄN THÔNG

Chuyên ngành: Điện Tử - Viễn Thông

ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP

NGHIÊN CỨU PHÂN LỚP TỰ ĐỘNG VĂN BẢN TIẾNG VIỆT

Vietnamese texts classification

Sinh viên thực hiện : Cao Đức Thành

MSSV : 1810514

Giáo viên hướng dẫn : Ths.Nguyễn Khánh Lợi

Tp. Hồ Chí Minh, 5/2022

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các Thầy (Cô) của trường Đại học Bách khoa – ĐHQG TPHCM, các Thầy (Cô) khoa Điện – Điện Tử đã tạo điều kiện cho chúng em có được môi trường học tập với nhiều trải nghiệm. Đặc biệt em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Khánh Lợi - người đã trực tiếp hướng dẫn em trong suốt quá trình thực hiện đề cương luận văn. Thầy luôn tạo điều kiện tốt nhất về kiến thức, cơ sở vật chất, thời gian và luôn giải đáp tận tình mọi thắc mắc, cũng như hỗ trợ em những tài liệu và thông tin vô cùng bổ ích. Trong cả quá trình may mắn được thầy hướng dẫn, em học được từ thầy không chỉ kinh nghiệm, kỹ năng, phương pháp làm việc nghiêm túc, hiệu quả, mà còn là nhiệt huyết, là lòng yêu nghề và niềm đam mê cháy bỏng chưa bao giờ thôi rực rỡ. Tất cả đã giúp ích cho em rất nhiều trong quá trình thực hiện luận văn và kể cả trong những nghiên cứu, làm việc sau này.

Trong quá trình thực hiện đề tài, khó tránh khỏi sai sót do lối diễn đạt và kiến thức chuyên môn còn nhiều hạn chế, rất mong nhận được ý kiến đóng góp từ Thầy (Cô) để em có thể học thêm được nhiều kinh nghiệm và tiếp thu để hoàn thành tốt hơn bài báo cáo.

Kính chúc quý Thầy (Cô) thật nhiều sức khoẻ và công tác tốt.

Em xin chân thành cảm ơn!

TÓM TẮT

1. Vấn đề cần giải quyết

Bài toán phân loại văn bản: phân loại các thể loại tin tức trong các bài báo.

Đầu vào: đoạn văn, bài báo.

Đầu ra: thể loại của đoạn văn, bài báo đó.

2. Phương pháp giải quyết

Phương pháp Deep Learning LSTM kết hợp với Word2Vector.

3. Kết quả sơ khởi đạt được

Thu thập được dữ liệu từ nhiều nguồn khác nhau.

Phân loại được các thể loại tin tức.

4. Kết quả dự kiến đạt được

Tóm tắt được các tin tức sau khi thu thập.

MỤC LỤC

CÁC TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ
NLP	Natural Language Processing (Xử lý ngôn ngữ tự nhiên)
LSTM	Long short term memory (Bộ nhớ ngắn-dài hạn)
GD	Gradient descent
CBOW	Continuous Bag-of-Words
RNN	Recurrent Neuron Network
CNN	Convolution Neuron Network
DNN	Deep Neuron Network
CNTT	Công nghệ thông tin
ReLU	Rectified Linear Unit

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

GIỚI THIỆU

Phân lớp văn bản là bài toán cơ bản trong khai phá dữ liệu văn bản. Bài toán phân lớp văn bản là việc gán tên các chủ đề (tên lớp/nhãn lớp) đã được xác định trước vào các văn bản dựa trên nội dung của chúng. Phân lớp văn bản là công việc được sử dụng để hỗ trợ trong quá trình tìm kiếm thông tin, chiết lọc thông tin, lọc văn bản hoặc tự động dẫn đường cho các văn bản tới những chủ đề xác định trước. Phân lớp văn bản có thể thực hiện thủ công hoặc tự động sử dụng các kỹ thuật học máy có giám sát. Các hệ thống phân lớp có thể ứng dụng trong việc phân loại tài liệu của các thư viện điện tử, phân loại văn bản báo chí trên các trang tin điện tử,… những hệ thống tốt, cho ra kết quả khả quan, giúp ích nhiều cho con người.

Đề tài *"Nghiên cứu phân lớp tự động văn bản tiếng Việt* ", vận dụng những kiến thức về kỹ thuật khai phá văn bản, kỹ thuật phân lớp văn bản nói riêng, và kiến thức về CNTT nói chung. Với mong muốn ứng dụng hệ thống phân lớp này vào phục vụ nghiên cứu khoa học và công tác quản lý, phân loại các tài liệu văn bản.

Nội dung và phạm vi đề tài: Trình bày khái niệm khai phá dữ liệu, khai phá văn bản, một số kỹ thuật khai phá văn bản và phân lớp văn bản. Nghiên cứu một số đặc điểm đặc trưng của ngôn ngữ tiếng Việt, phương pháp tách từ tiếng Việt và loại bỏ stop word. Nghiên cứu, sử dụng thuật toán, xây dựng bộ phân lớp văn bản báo chí tiếng Việt. Đầu vào của bộ phân lớp là văn bản báo chí tiếng Việt, đầu ra là kết quả phân lớp văn bản báo chí tiếng Việt vào một trong các chủ đề thông tin chuyên ngành: chính trị xã hội, đời sống, kinh doanh, khoa học, pháp luật, sức khỏe, thế giới, thể thao, văn hóa, vi tính. Bố cục của luận văn bao gồm:

Chương 1: Khái quát về phân lớp văn bản. Chương này trình bày khái quát về khai phá văn bản, Phân lớp văn bản.

Chương 2: Cơ sở lý thuyết. Chương này trình bày đặc điểm cơ bản của tiếng Việt, kỹ thuật tách từ văn bản tiếng Việt, thuật toán LSTM, mô hình Word Embedding.

Chương 3: Tiến hành và đánh giá kết quả. Trình bày cách xây dựng model và kết quả model cùng với các đánh giá.

Chương 4: Tổng kết. Trình bày các vấn đề đã đạt được, chưa đạt được và đưa ra hướng phát triển cho đề tài.

CHƯƠNG 1: KHÁI QUÁT VỀ PHÂN LOẠI VĂN BẢN

1.1 Khai phá dữ liệu văn bản

Khai phá dữ liệu văn bản là quá trình trích chọn ra các tri thức mới, có giá trị và tác động được, đang tiềm ẩn trong các văn bản, để sử dụng các tri thức này vào việc tổ chức thông tin tốt hơn nhằm hỗ trợ con người.

Dữ liệu văn bản thường được chia thành hai loại:

Dạng phi cấu trúc: là dạng văn bản chúng ta sử dụng hằng ngày được thể hiện dưới dạng ngôn ngữ tự nhiên của con người và không có một cấu trúc định dạng cụ thể nào. Ví dụ: các văn bản lưu dưới dạng tệp tin.TXT, .DOC.

Dạng bán cấu trúc: là các loại văn bản không được lưu trữ dưới dạng các bản ghi chặt chẽ mà được tổ chức qua các thẻ đánh dấu để thể hiện nội dung chính của văn bản. Ví dụ: dạng tệp tin HTML, email,…

Tùy từng mục đích sử dụng cụ thể mà việc xử lý văn bản được thực hiện trên dạng cấu trúc nào. Trong đề cương luận văn này, chúng ta quan tâm xử lý các dữ liệu văn bản ở dạng phi cấu trúc (biểu diễn dưới dạng tệp tin .TXT, .DOC).

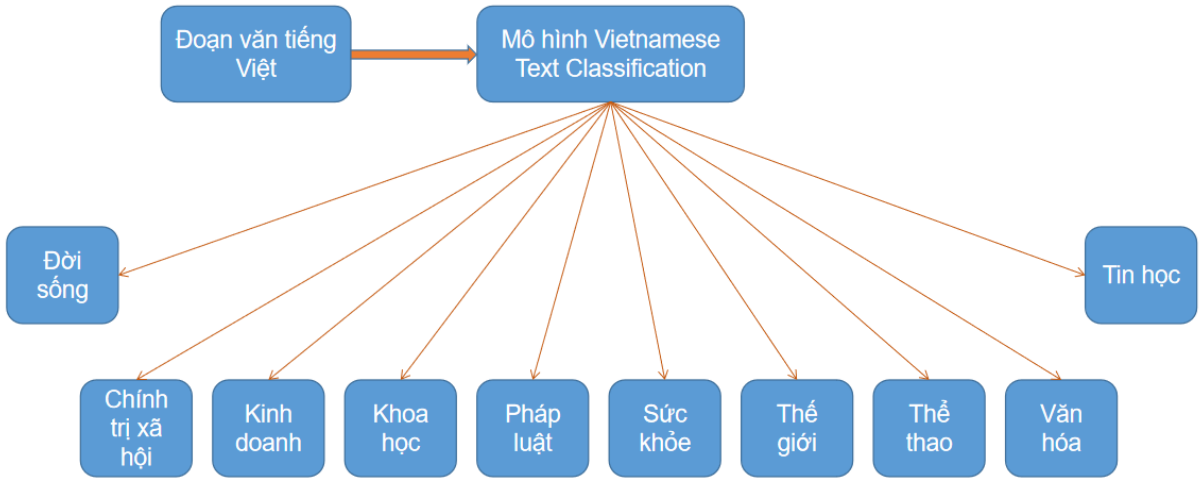
1.2 Tổng quan phân loại văn bản

Bài toán phân loại văn bản thực chất có thể xem là bài toán phân lớp. Phân loại văn bản tự động là việc gán các nhãn phân loại lên một văn bản mới dựa trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện. Nhiều kỹ thuật máy học và khai phá dữ liệu đã được áp dụng vào bài toán phân loại văn bản, chẳng hạn: phương pháp quyết định dựa vào thuật toán Naive Bayes, cây quyết định (decision tree), k–láng giềng gần nhất (KNN), mạng nơron (neural network),…

Phân loại văn bản là một bài toán xử lí văn bản cổ điển, đó là ánh xạ một văn bản vào một chủ đề đã biết trong một tập hữu hạn các chủ đề dựa trên ngữ nghĩa của văn bản. Ví dụ một bài viết trong một tờ báo có thể thuộc một (hoặc một vài) chủ đề nào đó (như thể thao, sức khỏe, công nghệ thông tin,…). Việc tự động phân loại văn bản vào một chủ đề nào đó giúp cho việc sắp xếp, lưu trữ và truy vấn tài liệu dễ dàng hơn về sau.

Đặc điểm nổi bật của bài toán này là sự đa dạng của chủ đề văn bản và tính đa chủ đề của văn bản. Tính đa chủ đề của văn bản làm cho sự phân loại chỉ mang tính tương đối và có phần chủ quan, nếu do con người thực hiện, và dễ bị nhập nhằng khi phân loại tự động. Rõ ràng một bài viết về Giáo dục cũng có thể xếp vào Kinh tế nếu như bài viết bàn về tiền nong đầu tư cho giáo dục và tác động của đầu tư này đến kinh tế - xã hội. Về bản chất, một văn bản là một tập hợp từ ngữ có liên quan với nhau tạo nên nội dung ngữ nghĩa của văn bản. Từ ngữ của một văn bản là đa dạng do tính đa dạng của ngôn ngữ (đồng nghĩa, đa nghĩa, từ vay mượn nước ngoài,…) và số lượng từ cần xét là lớn. Ở đây cần lưu ý rằng, một văn bản có thể có số lượng từ ngữ không nhiều, nhưng số lượng từ ngữ cần xét là rất nhiều vì phải bao hàm tất cả các từ của ngôn ngữ đang xét.

Trên thế giới đã có nhiều công trình nghiên cứu đạt những kết quả khả quan, nhất là đối với phân loại văn bản tiếng Anh. Tuy vậy, các nghiên cứu và ứng dụng đối với văn bản tiếng Việt còn nhiều hạn chế do khó khăn về tách từ và câu. Có thể liệt kê một số công trình nghiên cứu trong nước với các hướng tiếp cận khác nhau cho bài toán phân loại văn bản, bao gồm: phân loại với máy học vectơ hỗ trợ, cách tiếp cận sử dụng lý thuyết tập thô, cách tiếp cận thống kê hình vị, cách tiếp cận sử dụng phương pháp học không giám sát và đánh chỉ mục, cách tiếp cận theo luật kết hợp. Theo các kết quả trình bày trong các công trình đó thì những cách tiếp cận nêu trên đều cho kết quả khá tốt.



Hình 1.1 Mô hình phân loại văn bản

1.3 Các phương pháp giải quyết bài toán phân loại văn bản

1.3.1 phương pháp cây quyết định

Phương pháp cây quyết định có thể áp dụng vào bài toán phân loại văn bản. Dựa vào tập các văn bản huấn luyện (gọi tắt là tập huấn luyện), xây dựng một cây quyết định. Cây quyết định có dạng là cây nhị phân, mỗi nút trong tương ứng với việc phân hoạch tập văn bản dựa trên một thuộc tính nào đó. Việc xây dựng cây quyết định phụ thuộc vào việc lựa chọn thuộc tính để phân hoạch. Theo [9], việc lựa chọn thuộc tính phân hoạch dựa trên độ lợi thông tin (information gain) lớn nhất, đó là hiệu giữa độ hỗn loạn thông tin trước và sau phân hoạch với thuộc tính đó. Độ lợi thông tin được tính toán dựa vào độ hỗn loạn thông tin (Entropy) theo công thức (1). Giả sử tập huấn luyện S chứa các văn bản thuộc k chủ đề, thì độ hỗn loạn thông tin của tập S là:

$$Entropy(S) = \sum_{i=1}^{k}(-p_i \, log_2 \, p_i) \quad (1)$$

Trong đó pi là xác suất để một phần tử (1 văn bản) thuộc vào chủ đề thứ i. pi chính là tần suất xuất hiện một văn bản thuộc chủ đề thứ i trong tập S. Độ lợi thông tin khi dùng thuộc tính *a* phân hoạch tập S thành các tập con tùy theo giá trị của a (kí hiệu *Values(a)* trong công thức) là :

$$Gain(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

1.3.2 phương pháp support vector machine

Máy học vectơ hỗ trợ (SVM) là một giải thuật phân lớp có hiệu quả cao và đã được áp dụng nhiều trong lĩnh vực khai phá dữ liệu và nhận dạng. Tuy nhiên SVM chưa được áp dụng một cách có hiệu quả vào phân loại văn bản vì đặc điểm của bài toán phân loại văn bản là không gian đặc trưng thường rất lớn. Nghiên cứu phương pháp máy học vector hỗ trợ (SVM), áp dụng nó vào bài toán phân loại văn bản và so sánh hiệu quả của nó với hiệu quả của giải thuật phân lớp cổ điển, rất phổ biến đó là cây quyết định. Nghiên cứu chỉ ra rằng SVM với cách lựa chọn đặc trưng bằng phương pháp tách giá trị đơn (SVD) cho kết quả tốt hơn so với cây quyết định.

1.3.3 phương pháp Deep Learning

Phương pháp Deep Learning Neural Network: những thập niên gần đây, với sự phát triển nhanh chóng tốc độ xử lý của CPU, GPU và chi phí cho phần cứng ngày càng giảm, các dịch vụ hạ tầng điện toán đám mây ngày càng phát triển, làm tiền đề và cơ hội cho phương pháp học sâu Deep Learning Neural Network phát triển mạnh mẽ. Trong đó, bài toán phân tích văn bản đã được giải quyết bằng mô hình học Recurrent Neural Network (RNN) với một biến thể được dùng phổ biến hiện nay là Long Short Term Memory Neural Network (LSTMs), kết hợp với mô hình vector hóa từ (vector representations of words) Word2Vector với kiến trúc Continuous Bag-of-Words (CBOW). Mô hình này cho độ chính xác hơn 90%. Ưu điểm của phương pháp này là văn bản đầu vào có thể là 1 câu hay 1 đoạn văn. Để thực hiện mô hình này đòi hỏi phải có dữ liệu văn bản càng nhiều càng tốt để tạo Word2Vector CBOW chất lượng cao và dữ liệu gán nhãn lớn để huấn luyện (training), xác minh (validate) và kiểm tra (test) mô hình học có giám sát (Supervised Learning) LSTMs.

Dựa trên các phân tích trên, em quyết định chọn phương pháp deep learning LSTMs kết hợp với Word2Vector để giải quyết bài toán phân loại văn bản. Mô hình này tỏ ra sát với yêu cầu ứng dụng thực tiễn với văn bản đầu vào là một đoạn văn bất kỳ, có thể là bài báo online,... Đầu ra cho biết các văn bản đó thuộc thể loại nào.

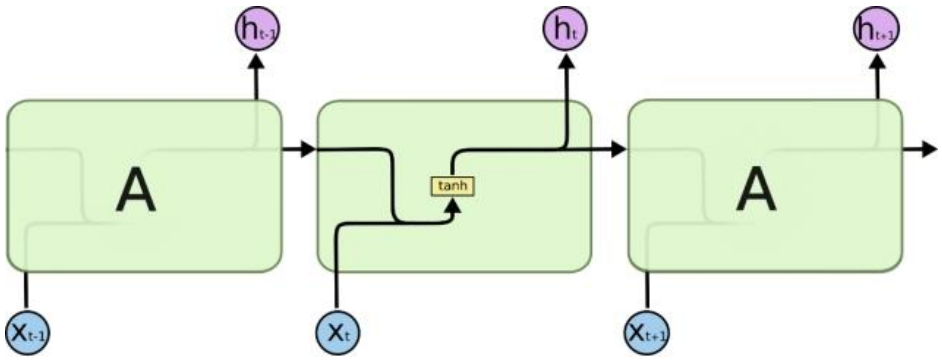
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Mạng LSTM và các thành phần liên quan

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & 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 đã trở nên phổ biến như hiện nay.

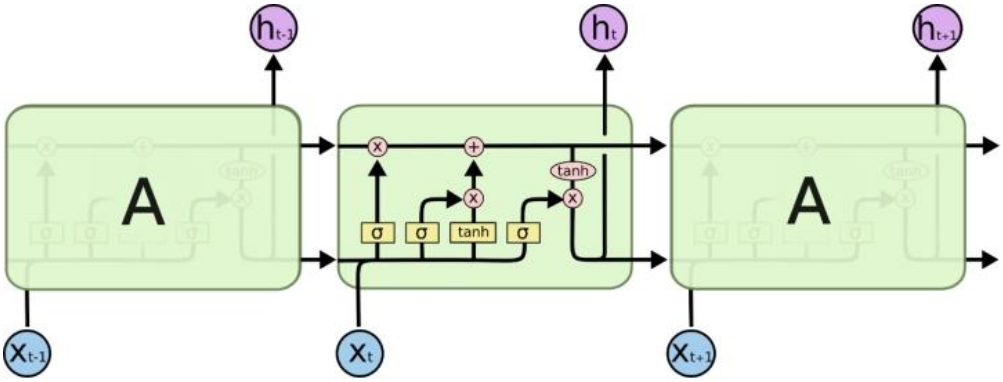
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các module lặp đi lặp lại của mạng neural. Với mạng RNN chuẩn, các module này có cấu trúc rất đơn giản, thường là một tầng Tanh.



Hình 2.1 Mô hình RNN

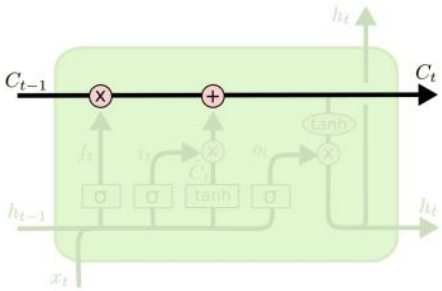
LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các module trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng neural, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



Hình 2.2 Mô hình LSTM

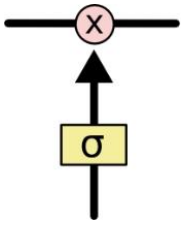
Ý tưởng cốt lõi của LSTM:

Chìa khóa của LSTM là trạng thái tế bào (cell state). Trạng thái tế bào là một dạng giống như băng truyền. 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 thông suốt mà không sợ bị thay đổi.



Hình 2.3 Trạng thái tế bào LSTM

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế bào, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate). Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



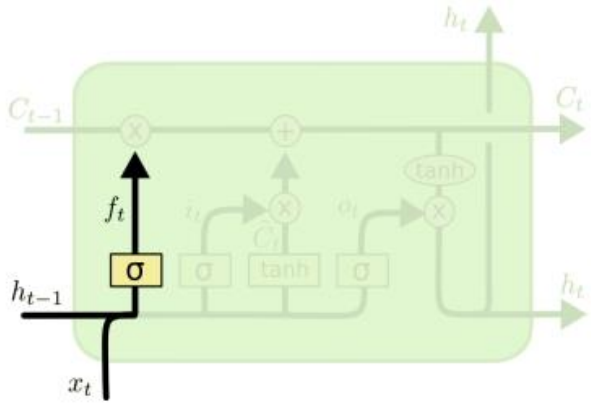
Hình 2.4 Cổng kết nối của LSTM

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, 1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Cụ thể cách hoạt động của LSTM:

Bước đầu tiên của LSTM là 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). Nó 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-1}. 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.

Ví dụ mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó. Với những bài toán như vậy, trong trường hợp trạng thái tế bào mang thông tin về giới tính của một nhân vật nào đó giúp ta sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật lúc trước nữa, vì nó không còn tác dụng gì với chủ thế mới này. Vì thế, chúng ta có tầng cổng quên:



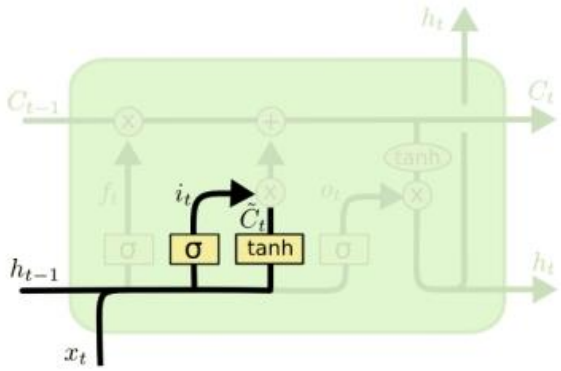
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hình 2.5 Tầng cổng quên LSTM

Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là "tầng cổng vào" (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một

tầng Tanh tạo ra một vector cho giá trị mới *Ct* nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.

Chẳng hạn với ví dụ mô hình ngôn ngữ của chúng ta, chúng ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.

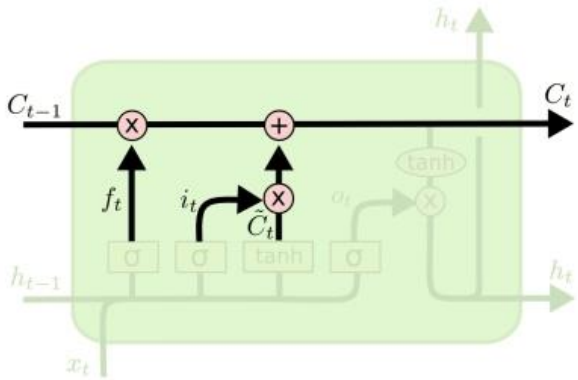


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 2.6 Tầng cổng vào và Tanh của LSTM

Tiếp theo là cập nhập trạng thái tế bào cũ C_{t-1} thành trạng thái mới *Ct*. Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện. Ta sẽ nhân trạng thái cũ với *ft* để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm *it*∗*Ct*~. Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.

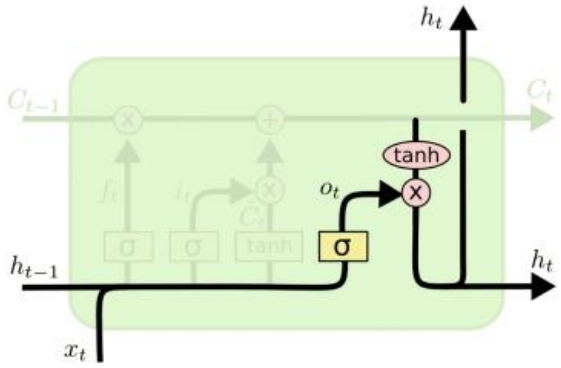
Với bài toán mô hình ngôn ngữ, việc này chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.



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

Hình 2.7 Tầng cập nhật trạng thái

Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. 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 nó trạng thái tế bảo qua một hàm Tanh để đưa giá trị nó về khoảng [-1, 1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.



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

Hình 2.8 Tầng đầu ra

2.2 Mô hình Word Embedding

2.2.1 Mô hình Word2Vec

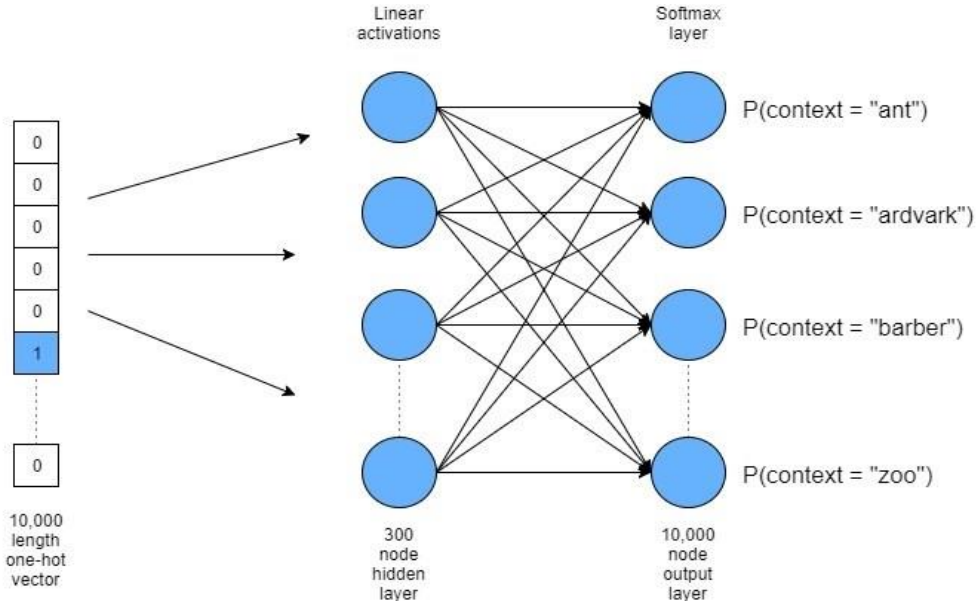
Word Embedding là việc biểu diễn các từ dưới dạng các vector số thực với số chiều xác định. Word2Vec là 1 trong những mô hình đầu tiên về Word Embedding sử dụng mạng neural, vẫn khá phổ biến ở thời điểm hiện tại, mô hình có khả năng vector hóa từng từ dựa trên tập các từ chính và các từ ngữ cảnh,... Về mặt toán học, thực chất Word2Vec là việc ánh xạ từ từ 1 tập các từ (vocabulary) sang 1 không gian vector, mỗi vector được biểu diễn bởi N số thực. Mỗi từ ứng với 1 vector cố định. Sau quá trình huấn luyện mô hình bằng thuật toán back probagation, trọng số các vector của từng từ được cập nhật liên tục. Từ đó, ta có thể thực hiện tính toán bằng các khoảng cách quen thuộc như euclide, cosine, mahattan,... Những từ càng "gần" nhau về mặt khoảng cách thường là các từ hay xuất hiện cùng nhau trong văn cảnh, các từ đồng nghĩa, các từ thuộc cùng 1 trường từ vựng,...

2.2.2 Các hướng tiếp cận của Word2Vec

Word2Vec bao gồm 2 cách tiếp cận chính bao gồm:

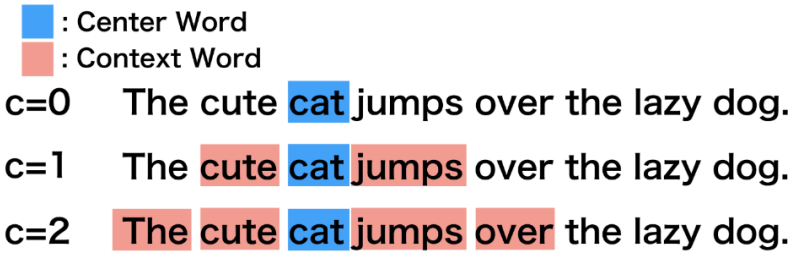
- CBOW model
- Skip-gram model

Mô hình chung của Word2Vec (cả CBOW và Skip-gram) đều dựa trên 1 mạng neural network khá đơn giản. Gọi V là tập các tất cả các từ hay vocabulary với N từ khác nhau. Layer input biểu diễn dưới dạng one-hot encoding với N node đại diện cho N từ trong vocabulary. Activation function (hàm kích hoạt) chỉ có tại layer cuối là softmax function, loss function là cross entropy loss, tương tự như cách biểu diễn mô hình của các bài toán classification thông thường vậy. Ở giữa 2 layer input và output là 1 layer trung gian với size = k, chính là vector sẽ được sử dụng để biểu diễn các từ sau khi huấn luyện mô hình.



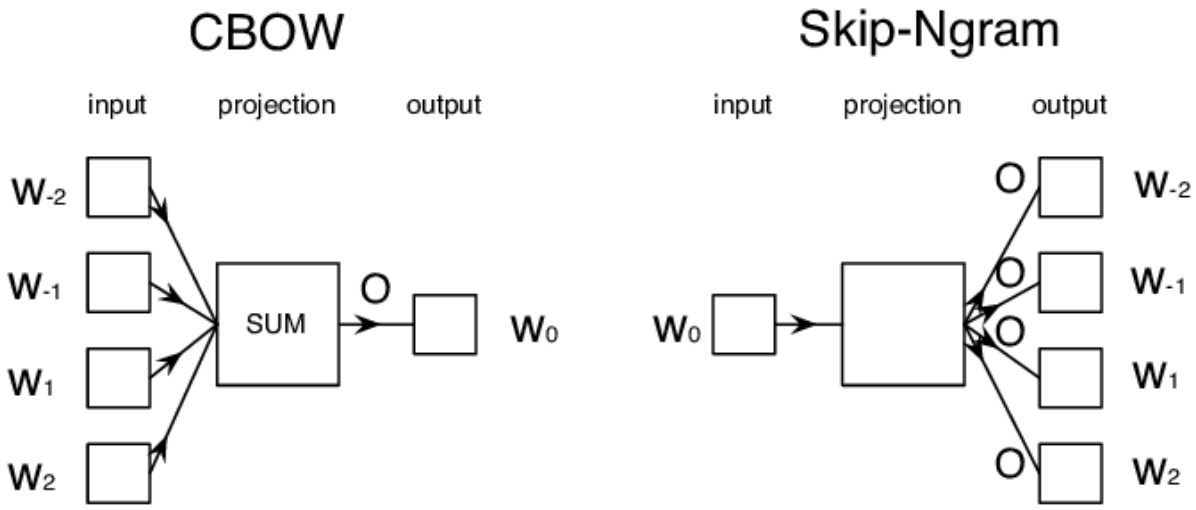
Hình 2.9 Mô hình Word2Vec

Ta có 2 khái niệm quan trọng là: target word (center word) và context words**.** Hiểu đơn giản là ta sẽ sử dụng từ ở giữa (target word hay center word) cùng với các từ xung quanh nó (context words) để mô hình thông qua đó để tiến hành huấn luyện:



Hình 2.10 Mô hình context word và center word

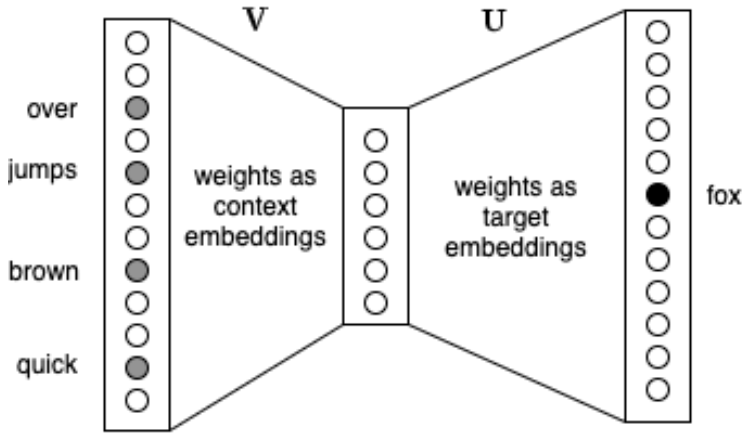
Cùng với đó, quy định 1 tham số ***c*** hay window là việc sử dụng bao nhiêu từ xung quanh, gồm 2 bên trái phải của target word, gần như cách biểu diễn theo N-grams cho từ.



Hình 2.11 Mô hình CBOW và Skip-gram

2.2.2.1 Mô hình CBOW

Ý tưởng chính của CBOW là dựa vào các context word (hay các từ xung quanh) để dự đoán center word (từ đích). CBOW có điểm thuận lợi là training mô hình nhanh hơn so với mô hình skip-gram, thường cho kết quả tốt hơn với frequence words (hay các từ thường xuất hiện trong văn cảnh).

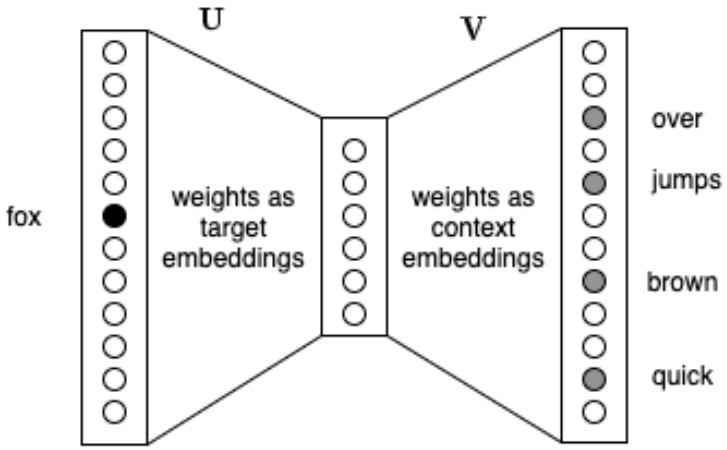


Hình 2.12 Minh họa CBOW dưới dạng mạng neural

Như hình trên ta thấy: từ các context word như: ‘over’, ‘jumps’, ‘brown’, ‘quick’ ta sẽ dự đoán được center word là từ ‘fox’ sau khi đi qua các lớp hidden layers.

2.2.2.2 Mô hình Skipgrams

Skip-gram thì ngược lại với CBOW, dùng target word để dự đoán các từ xung quanh. Skip-gram huấn luyện chậm hơn. Thường làm việc khá tốt với các tập data nhỏ, đặc biệt do đặc trưng của mô hình nên khả năng vector hóa cho các từ ít xuất hiện tốt hơn CBOW.



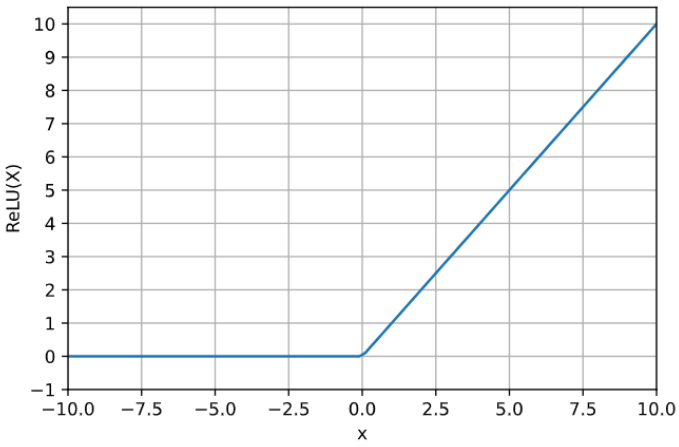
Hình 2.13 Minh họa Skip-gram dưới dạng mạng neural

Như hình trên ta thấy: center word là từ ‘fox’ sau khi đi qua các lớp hidden layers ta sẽ dự đoán được các context word như: ‘over’, ‘jumps’, ‘brown’, ‘quick’ .

2.3 Lý thuyết các hàm kích hoạt

2.3.1 Hàm Relu

Công thức: $f(x) = \max(0, x)$



Hình 2.14 Hàm Relu

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0 đưa về 0 và giữ nguyên các giá trị lớn hơn 0.

Ưu điểm:

Tốc độ hội tụ nhanh hơn hẳn. ReLU có tốc độ hội tụ nhanh gấp 6 lần Tanh, điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sigmoid và Tanh.

Tính toán nhanh hơn. Tanh và Sigmoid sử dụng hàm exp và công thức phức tạp hơn ReLU rất nhiều do vậy sẽ tốn nhiều chi phí hơn để tính toán.

Nhược điểm:

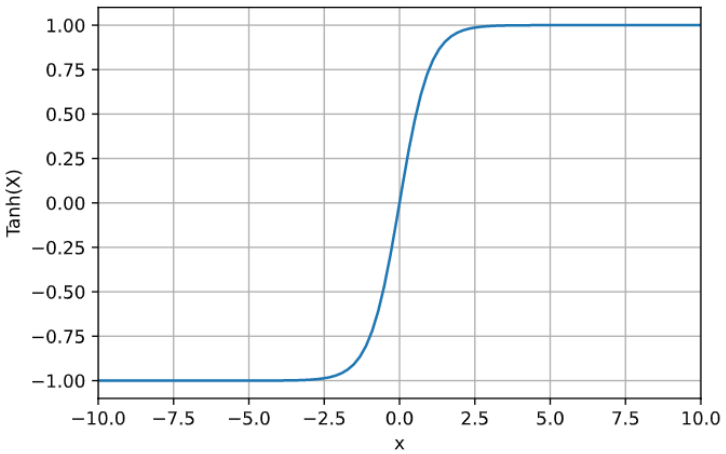
Với các node có giá trị nhỏ hơn 0, qua ReLU activation sẽ thành 0, hiện tượng đấy gọi là "Dying ReLU". Nếu các node bị chuyển thành 0 thì sẽ không có ý nghĩa với bước linear activation ở lớp tiếp theo và các hệ số tương ứng từ node đấy cũng không được cập nhật với gradient descent.

Khi learning rate lớn, các trọng số (weights) có thể thay đổi theo cách làm tất cả neuron dừng việc cập nhật.

2.3.2 Hàm tanh

Công thức:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



Hình 2.15 Hàm Tanh

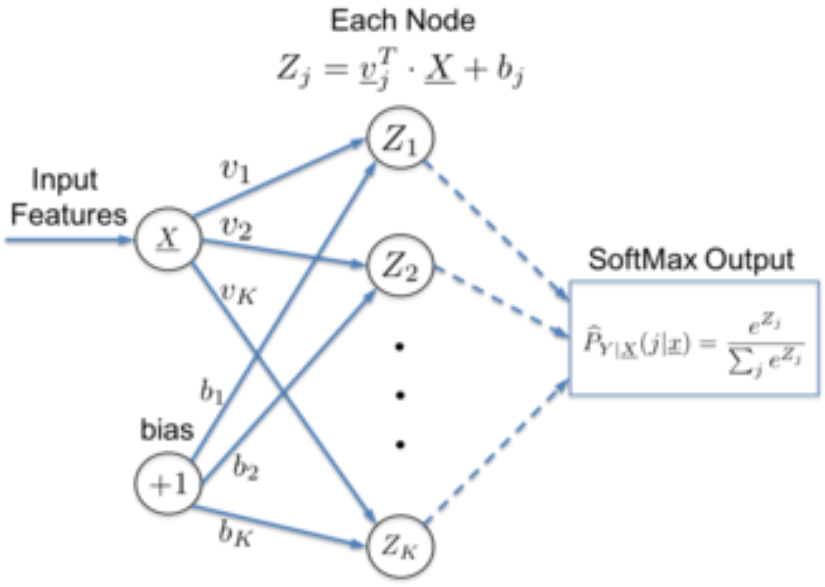
Hàm tanh nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (-1, 1). Hàm Tanh bị bão hoà ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm Tanh lại đối xứng qua 0 nên khắc phục được nhược điểm của hàm Relu.

2.3.3 Hàm softmax

Công thức hàm softmax:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}$$

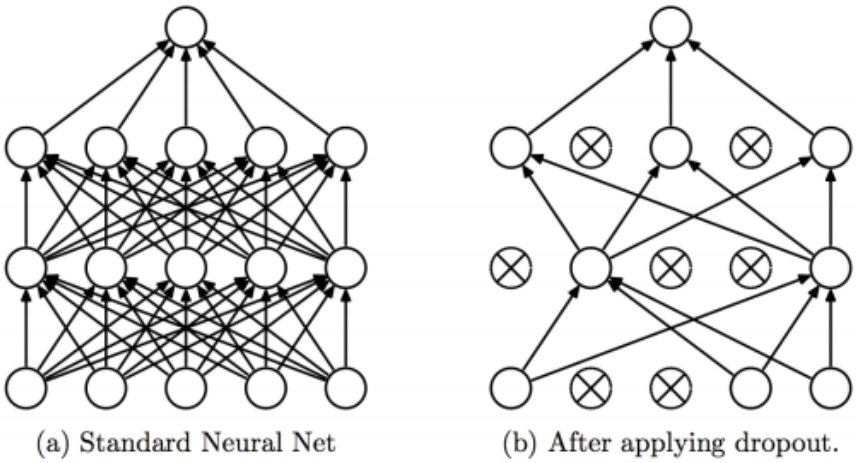
Hàm Softmax nhận vector đầu vào và đầu ra xuất ra các số đại diện cho xác suất, giá trị của mỗi số nằm trong khoảng từ 0 đến 1 là phạm vi giá trị hợp lệ của xác suất. Phạm vi được biểu thị là [0, 1]. Toàn bộ vectơ đầu ra tổng bằng 1. Có nghĩa là khi tất cả các xác suất được tính, đó là 100%.



Hình 2.16 Biểu diễn hàm softmax

2.4 Kỹ thuật Dropout

Trong mạng neural network, kỹ thuật dropout là việc chúng ta sẽ bỏ qua một vài unit trong suốt quá trình train trong mô hình, những unit bị bỏ qua được lựa chọn ngẫu nhiên. Ở đây, chúng ta hiểu "bỏ qua - ignoring" là unit đó sẽ không tham gia và đóng góp vào quá trình huấn luyện (lan truyền tiến và lan truyền ngược).



Hình 2.17 Mô hình Dropout

Dropout ép mạng neural phải tìm ra nhiều đặc trưng hơn, với đặc điểm là chúng phải hữu ích hơn, tốt hơn khi kết hợp với nhiều neuron khác. Dropout đòi hỏi phải gấp đôi quá trình huấn luyện để đạt được sự hội tụ. Tuy nhiên, thời gian huấn luyện cho mỗi epoch sẽ ít hơn.

2.5 Các thuật toán tối ưu hóa (Optimizer)

2.5.1 Thuật toán tối ưu hóa là gì?

Thuật toán tối ưu là cơ sở để xây dựng mô hình neural network với mục đích học được các features (hay pattern) của dữ liệu đầu vào, từ đó có thể tìm 1 cặp weights và bias phù hợp để tối ưu hóa model.

2.5.2 Thuật toán Gradient descent

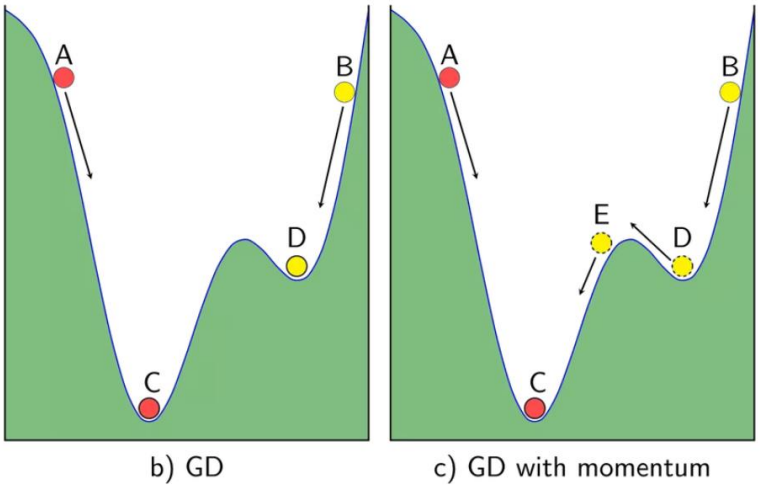
Trong các bài toán tối ưu, chúng ta thường tìm giá trị nhỏ nhất của 1 hàm số nào đó, mà hàm số đạt giá trị nhỏ nhất khi đạo hàm bằng 0. Nhưng không phải lúc nào hàm số cũng có thể đạo hàm được, đối với các hàm số nhiều biến thì đạo hàm rất phức tạp, thậm chí là bất khả thi. Nên thay vào đó người ta tìm điểm gần với điểm cực tiểu nhất và xem đó là nghiệm bài toán. Gradient Descent dịch ra tiếng Việt là giảm dần độ dốc, nên hướng tiếp cận ở đây là chọn 1 nghiệm ngẫu nhiên cứ sau mỗi vòng lặp (hay epoch) ta sẽ tối ưu hóa các giá trị weights và bias cho nó tiến dần đến điểm cần tìm.

Công thức : $x_{new} = x_{old} - learningrate.gradient(x)$

Gradient descent phụ thuộc vào nhiều yếu tố : như nếu chọn điểm x ban đầu khác nhau sẽ ảnh hưởng đến quá trình hội tụ; hoặc tốc độ học (learning rate) quá lớn hoặc quá nhỏ cũng ảnh hưởng: nếu tốc độ học quá nhỏ thì tốc độ hội tụ rất chậm ảnh hưởng đến quá trình training, còn tốc độ học quá lớn thì tiến nhanh tới đích sau vài vòng lặp tuy nhiên thuật toán không hội tụ, quanh quẩn quanh đích vì bước nhảy quá lớn.

2.5.3 Thuật toán Momentum

Để khắc phục các hạn chế trên của thuật toán Gradient Descent người ta dùng gradient descent with momentum.



Hình 2.18 so sánh gradient descent với momentum

Để giải thích được Gradient with Momentum thì trước tiên ta nên nhìn dưới góc độ vật lí: Như hình b phía trên, nếu ta thả 2 viên bi tại 2 điểm khác nhau A và B thì viên bị A sẽ trượt xuống điểm C còn viên bi B sẽ trượt xuống điểm D, nhưng ta lại không mong muốn viên bi B sẽ dừng ở điểm D (local minimum) mà sẽ tiếp tục lăn tới điểm C (global minimum). Để thực hiện được điều đó ta phải cấp cho viên bi B một vận tốc ban đầu đủ lớn để nó có thể vượt qua điểm E tới điểm C.

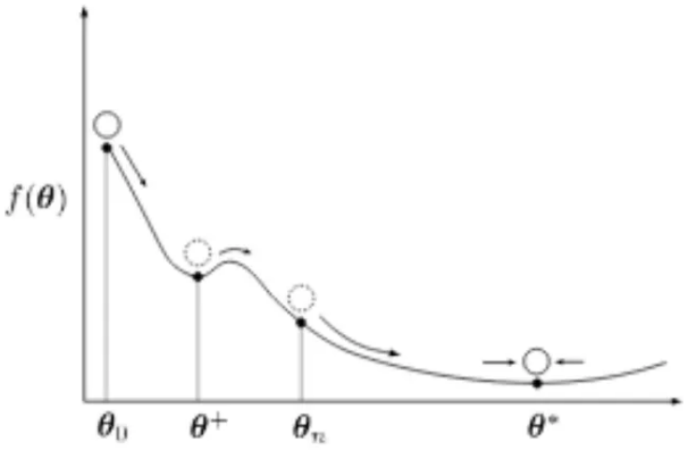
Công thức Momentum: $x_{new} = x_{old} - (\gamma . v + learningrate . gradient(x))$
x_{new}: tọa độ mới
x_{old} : tọa độ cũ
γ: parameter, thường =0.9
learningrate : tốc độ học
gradient : đạo hàm của hàm f

Ưu điểm : thuật toán tối ưu giải quyết được vấn đề: Gradient Descent không tiến được tới điểm global minimum mà chỉ dừng lại ở local minimum.

Nhược điểm : tuy momentum giúp hòn bi vượt dốc tiến tới điểm đích, tuy nhiên khi tới gần đích, nó vẫn mất khá nhiều thời gian giao động qua lại trước khi dừng hẳn, điều này được giải thích vì viên bi có đà.

2.5.4 Thuật toán Adam

Adam là sự kết hợp của Momentum và RMSprop . Nếu giải thích theo hiện tượng vật lí thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như 1 quả cầu rất nặng có ma sát, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn.

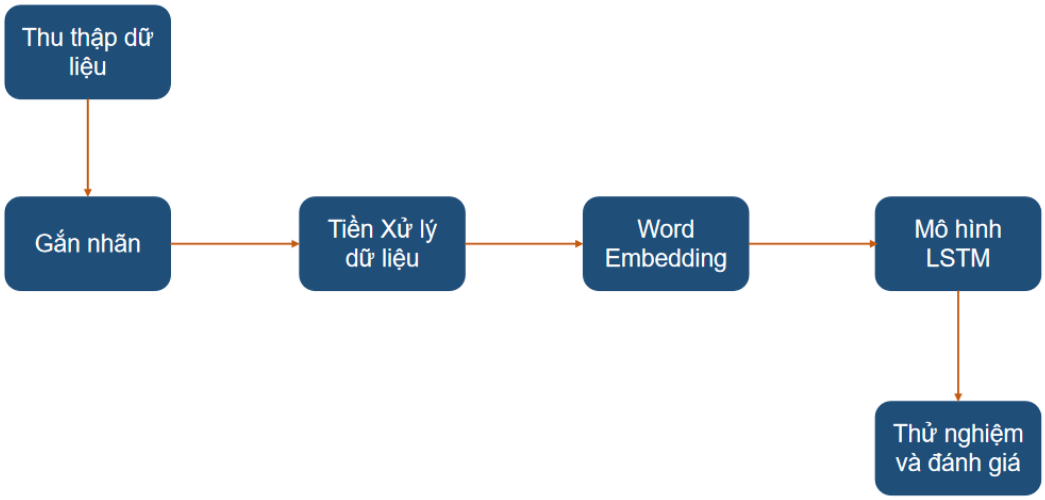


Hình 2.19 Ví dụ mô hình Adam

CHƯƠNG 3: TIẾN HÀNH VÀ ĐÁNH GIÁ KẾT QUẢ

Bài toán phân loại văn bản có thể được phát biểu như sau: đầu vào của bài toán là một văn bản và đầu ra là xác suất văn bản đó thuộc một trong các lớp mà chúng ta đã gắn nhãn.

Quy trình thực hiện bài toán tóm tắt văn bản theo sơ đồ bên dưới:



Hình 3.1 Quy trình tóm tắt văn bản

3.1 Thu thập và gắn nhãn dữ liệu

Thu thập dữ liệu: dữ liệu có thể được thu thập từ nhiều nguồn trên internet: facebook, các trang báo điện tử như: VnExpress, Thanh Niên online, Tuổi trẻ online,... Dữ liệu bao gồm: 30000 bài báo cho tập train, 10000 bài báo cho tập test và 10000 bài báo cho tập validation.

Gắn nhãn dữ liệu: dữ liệu sau khi thu thập được sẽ được tiến hành gắn nhãn với 10 nhãn: chính trị xã hội, đời sống, kinh doanh, khoa học, pháp luật, sức khỏe, thế giới, thể thao, văn hóa, tin học.

3.2 Tiền xử lý dữ liệu

Dữ liệu sau khi thu thập về được đem đi xử lý cơ bản như: loại bỏ các kí tự đặc biệt, các lỗi phân tách câu, các hastag,... và được chuẩn hóa về dạng chữ thường. Sau đó đưa dữ liệu đi qua bộ tách từ của underthesea để những từ đơn cũng như từ đôi được đưa về đúng với ý nghĩa của nó.

```
def text_preprocess(sentence):
    sentence = word_tokenize(sentence, format="text")
    sentence = re.sub(r"[()\"#/@;:<>{}`+=~|*'.!?,.]", "", sentence)
    sentence = re.sub(r'[^\s\wỉĩịúùủũụưứừửữựýỳỷỹỵđ]',' ',sentence)
    sentence.lower()
    sentence = re.sub(r'\s+', ' ', sentence).strip()
    return sentence
```

Bảng 3.1 xử lý văn bản

```
Điều này thể hiện hai khía cạnh về an ninh ở châu Á - Thái Bình Dương
: Thứ nhất là ngày càng có nhiều nước trên thế giới quan tâm đến khu
vực này , kể cả những nước lớn , và những nước trong khu vực cũng như
ngoài khu vực . Khía cạnh thứ hai là tình hình an ninh châu Á - Thái
Bình Dương lợi ích rất lớn , tương lai rất tốt đẹp nhưng thực tế có
rất nhiều vấn đề đang nổi lên . Đó là vấn đề cạnh tranh giữa các nước
lớn ; tuân thủ luật pháp quốc tế ; sử dụng sức mạnh quân sự trong
việc xử lý các vấn đề giữa các cường quốc với nhau và giữa các cường
quốc với các nước trong khu vực ; tranh chấp lãnh thổ , môi trường
hay những vấn đề mới về chiến tranh trong tương lai . Chính vì vậy
trong năm nay chủ đề của Shangri - La rất rộng . Có thể khái quát là
đối thoại đã đề cập đến những thách thức an ninh ở trong khu vực ,
cũng như làm thế nào để giải quyết nó để tất cả các nước đều được thụ
hưởng lợi ích trong khu vực châu Á - Thái Bình Dương.....
```

Bảng 3.2 Văn bản trước khi xử lý

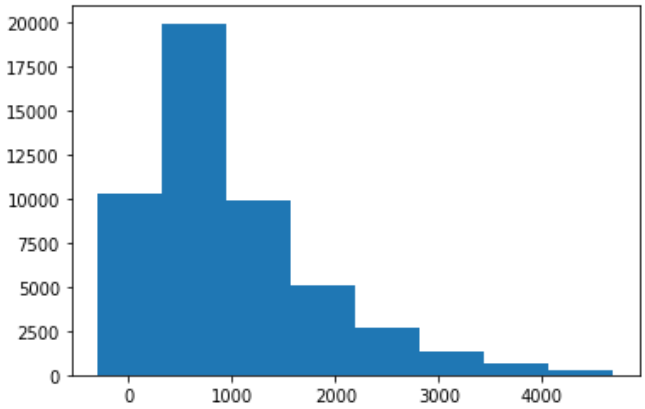
```
Điều này thể_hiện hai khía_cạnh về an_ninh ở châu_Á Thái_Bình_Dương
Thứ nhất là ngày_càng có nhiều nước trên thế_giới_quan tâm đến khu_vực
này kể_cả những nước_lớn và những nước trong khu_vực cũng như ngoài
khu_vực _Khía_cạnh thứ hai là tình_hình an_ninh châu_Á
Thái_Bình_Dương lợi_ích rất lớn tương_lai rất tốt_đẹp nhưng thực_tế
có rất nhiều vấn_đề đang nổi lên _Đó là vấn_đề cạnh_tranh giữa các
nước_lớn tuân_thủ luật_pháp quốc_tế sử_dụng sức_mạnh quân_sự trong
việc xử_lý các vấn_đề giữa các cường_quốc với nhau và giữa các
cường_quốc với các nước trong khu_vực tranh_chấp lãnh_thổ môi_trường
hay những vấn_đề mới về chiến_tranh trong tương_lai Chính vì_vậy trong
năm nay chủ_đề của Shangri La rất rộng _Có_thể khái_quát là đối_thoại
đã đề_cập đến những thách_thức an_ninh ở trong khu_vực cũng như làm
thế_nào để giải_quyết nó để tất_cả các nước đều được thụ_hưởng lợi_ích
trong khu_vực châu_Á Thái_Bình_Dương
```

Bảng 3.3 Văn bản sau khi xử lý

3.3 Word embedding

Dữ liệu sau khi đã làm sạch, tách từ ta tiến hành vector hóa chúng, đưa mỗi từ thành một vector số.

Theo hình 3.2 ta thấy mỗi bài báo trung bình có khoảng 500 từ và để cho các vector của câu có cùng chiều dài ta tiến hành padding chúng với chiều dài tối đa là 500 từ như đã nói ở trên. Với những câu có chiều dài hơn 500 thì ta chỉ lấy 500 từ đầu tiên còn những câu dưới 500 từ thì ta thêm vào những số 0 phía sau từ cuối cùng.



Hình 3.2 Histogram độ dài các bài báo

Ta đưa những dữ liệu ở trên đi qua một pretrained model word2vec (skip-gram) 150 chiều để các từ không chỉ là một vector mà là một vector 150 chiều. Vì vậy mỗi từ sẽ có khoảng cách gần, xa khác nhau trong không gian vector để thể hiện sự tương đồng về mặt ngữ nghĩa cũng như cảm xúc của từng từ. Từ đó, mỗi từ sẽ có sự liên quan về mặt ngữ nghĩa với nhau nhiều hơn và khi đưa vào mô hình máy học thì máy sẽ dễ hơn trong việc phân tích cảm xúc của câu văn.

```
print("Number of word vectors:{}".format(len(word_vectors.vocab))
model.wv.most_similar_cosmul(positive=['phong_phú', 'đa_dạng'],
                             negative=['hạn hẹp'])
```

```
Number of word vectors: 189485
[('gia_vị', 0.9468579888343811),
('ẩm_thực', 0.9415967464447021),
('bắt_mắt', 0.9373512864112854),
('tiện_dụng', 0.9353459477424622),
('tiện_ích', 0.9268515706062317),
('đậm_đà', 0.9252027273178101),
('màu_sắc', 0.925193190574646),
```

Bảng 3.4 Các từ có khoảng cách gần nhau

3.5 Xây dựng Model

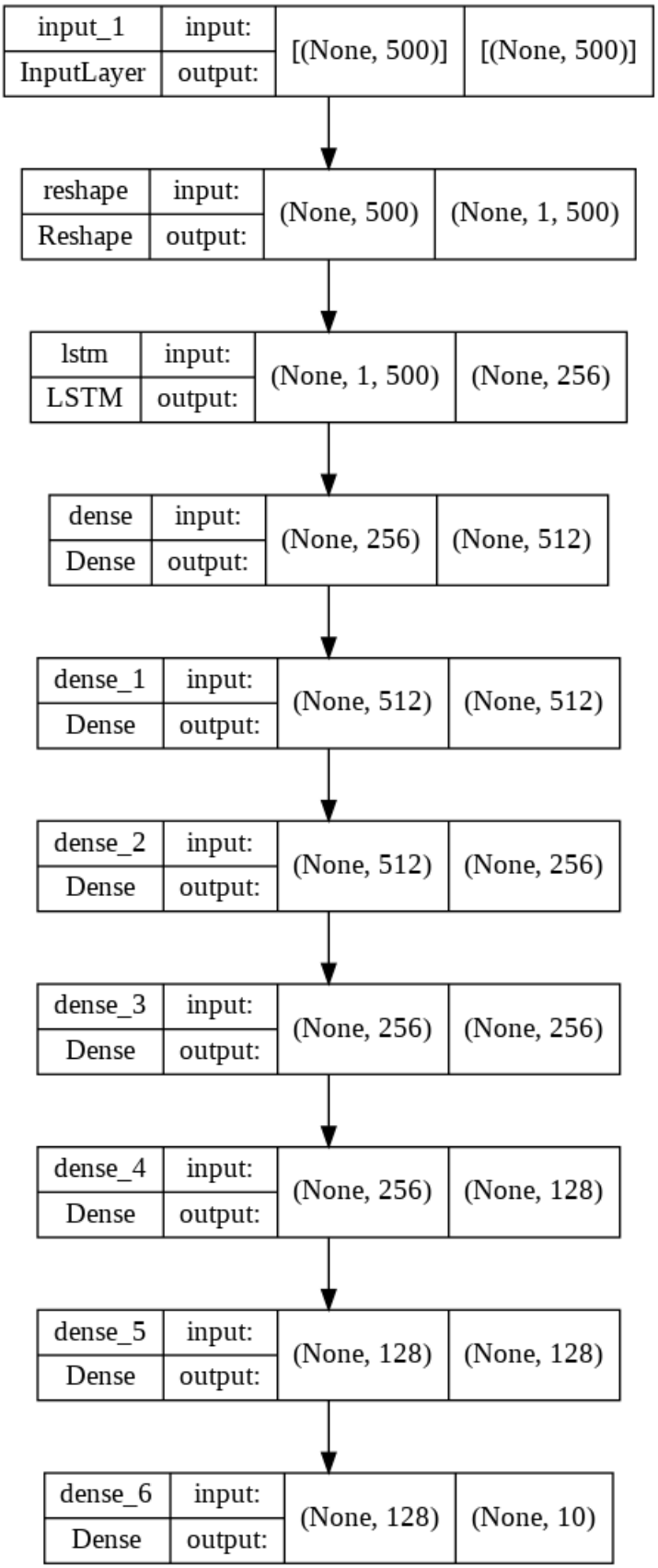
Model CNN với đầu vào là câu đã chuẩn hóa thành 500 từ, sau đó đi qua một lớp Embedding với số chiều là 150. Sau khi đã Embedding để có được mối quan hệ ngữ nghĩa giữa các từ, ta tiếp tục sử dụng lớp LSTM với 256 Neuron. Tiếp theo là hai lớp Dense 512 Neuron, hai lớp Dense 256 Neuron và hai lớp Dense 128 Neuron để mở rộng số đặc trưng mà model phải học để không bỏ sót đặc trưng quan trọng. Lớp Dropout được sử dụng để loại bỏ đi ngẫu nhiên các Neuron của model trong quá trình trainning, điều đó bắt buộc model phải làm việc tốt hơn nữa để điều chỉnh các trọng số của model trong điều kiện bị mất một số dữ liệu. Hàm Activation được sử dụng cho các lớp ở trên là hàm "Relu" với mong muốn đơn giản hóa cho model tính toán nhanh hơn. Cuối cùng ta sử dụng một lớp Dense có 10 Neuron với mục đích phân loại đặc trưng về 10 thành phần tương ứng với 10 thể loại báo mà model sẽ phân loại. Hàm Activation được sử dụng cho lớp này là hàm "softmax" để cho xác suất của các đặc trưng nằm trong khoảng [0, 1] dễ dàng trong việc phân loại đặc trưng hơn.

```
#input layer
input_layer = Input(shape=(500,))
layer = Reshape((1, 500))(input_layer)
embedded_sequences = Embedding(nb_words, output_dim=150,
                               weights=[wv_matrix], input_length=350,
                               trainable=True)(sequence_input)
#hidden layer
layer = LSTM(256, activation='relu',dropout=0.4,recurrent_dropout=0.3)
(embedded_sequences)
layer = Dense(512, activation='relu')(layer)
layer = Dense(512, activation='relu')(layer)
layer = Dense(256, activation='relu')(layer)
layer = Dense(256, activation='relu')(layer)
layer = Dense(128, activation='relu')(layer)
layer = Dense(128, activation='relu')(layer)
#output layer
output_layer = Dense(10, activation='softmax')(layer)
model = models.Model(input_layer, output_layer)
model.compile(optimizer= 'adam', loss='sparse_categorical_crossentropy',
 metrics=['accuracy'])
```

Bảng 3.5 Architecture model

```
Total params: 1,417,226
Trainable params: 1,417,226
Non-trainable params: 0
```

Bảng 3.6 Tổng kết model



Hình 3.3 Architecture của model

3.6 Train model

Sau khi xây dựng xong model ta tiến hành chia dữ liệu thành 3 tập lần lượt là train, validation và test. Model được train trên tập dữ liệu train và ta sẽ quan sát quá trình học của model có tốt hay không, có mang tính tổng thể hay chưa dựa trên tập validation.

```
from sklearn.model_selection import train_test_split
x, x_test, y, y_test = train_test_split(data, labels, test_size=0.2)
x_train, x_val, y_train, y_val = train_test_split(x, y,test_size=0.3)
```

Bảng 3.7 Chia các tập dữ liệu

Ta chọn batch_size là 512 câu cho mỗi batch, tương ứng với train hết 56 batch sẽ train được hết tập dữ liệu, số epoch để train là 10, hệ số learning_rate để cập nhật cho model là 0.0005.

```
predict.compile(loss='categorical_crossentropy',
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0005),
metrics=['acc'])
```

Bảng 3.8 Compile model

Khi đã thiết lập đầy đủ các thông số ta bắt đầu train model.

```
 Epoch 1/10
56/56 [==============================] - 16s 184ms/step - loss:
2.0517 - accuracy: 0.2290 - val_loss: 1.5165 - val_accuracy: 0.5072
Epoch 2/10
56/56 [==============================] - 8s 146ms/step - loss: 1.1672
- accuracy: 0.6091 - val_loss: 0.6257 - val_accuracy: 0.8164
Epoch 3/10
56/56 [==============================] - 6s 114ms/step - loss: 0.7449
- accuracy: 0.7631 - val_loss: 0.4666 - val_accuracy: 0.8575
Epoch 4/10
56/56 [==============================] - 6s 115ms/step - loss: 0.5733
- accuracy: 0.8211 - val_loss: 0.4338 - val_accuracy: 0.8655
Epoch 5/10
56/56 [==============================] - 7s 117ms/step - loss: 0.5007
- accuracy: 0.8441 - val_loss: 0.3959 - val_accuracy: 0.8840
Epoch 6/10
56/56 [==============================] - 6s 115ms/step - loss: 0.4345
- accuracy: 0.8666 - val_loss: 0.3620 - val_accuracy: 0.8928
Epoch 7/10
56/56 [==============================] - 6s 116ms/step - loss: 0.3893
- accuracy: 0.8780 - val_loss: 0.3613 - val_accuracy: 0.8921
```

Bảng 3.9 Quá trình train model

3.7 Kết quả và phân tích lỗi

3.7.1 Kết quả mô hình

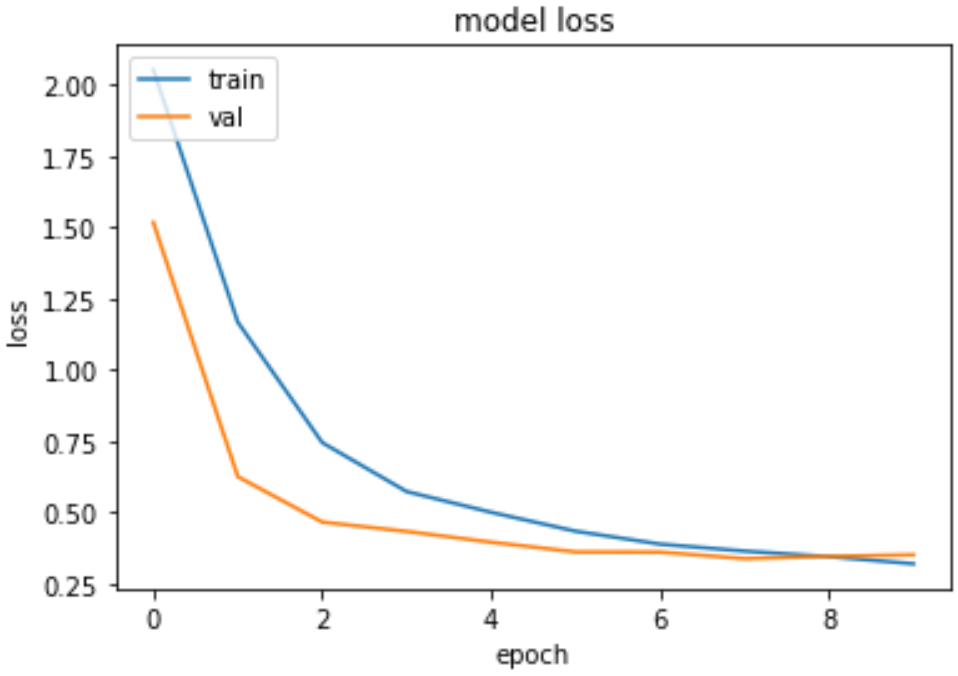
Sau khi train model qua 10 epoch ta nhận được độ chính xác ở tập train là 0.899 và độ chính xác ở tập validation là 0.894.

```
Epoch 8/10
56/56 [==============================] - 7s 117ms/step - loss: 0.3648
- accuracy: 0.8883 - val_loss: 0.3384 - val_accuracy: 0.8971
Epoch 9/10
56/56 [==============================] - 6s 115ms/step - loss: 0.3453
- accuracy: 0.8933 - val_loss: 0.3464 - val_accuracy: 0.8968
Epoch 10/10
56/56 [==============================] - 7s 116ms/step - loss: 0.3204
- accuracy: 0.8999 - val_loss: 0.3519 - val_accuracy: 0.8946

Train accuract 0.9403715258082813
Validation accuracy:  0.8946236559139785
Test accuracy:  0.9031265508684864
```

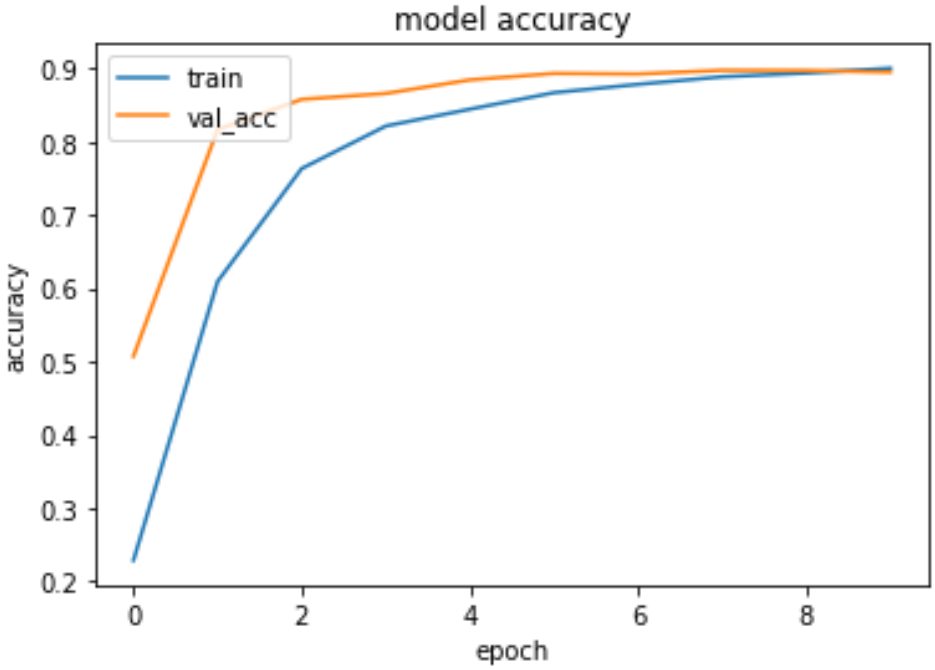
Bảng 3.10 Kết quả train model

Hình 3.4 bên dưới là đồ thị hàm Loss giữa kết quả dự đoán và kết quả đúng, ta thấy đến epoch thứ 10 thì giá trị Loss giữa tập train và tập test tiến đến giá trị gần như nhau nên ta chấp nhận dừng ở epoch thứ 10.



Hình 3.4 Đồ thị giá trị Loss của model

Hình 3.5 bên dưới là đồ thị hàm Accuracy giữa kết quả dự đoán và kết quả đúng, ta thấy đến epoch thứ 10 thì Accuracy của tập train và test dần hội tụ lại với nhau nên ta dừng ở epoch thứ 10.



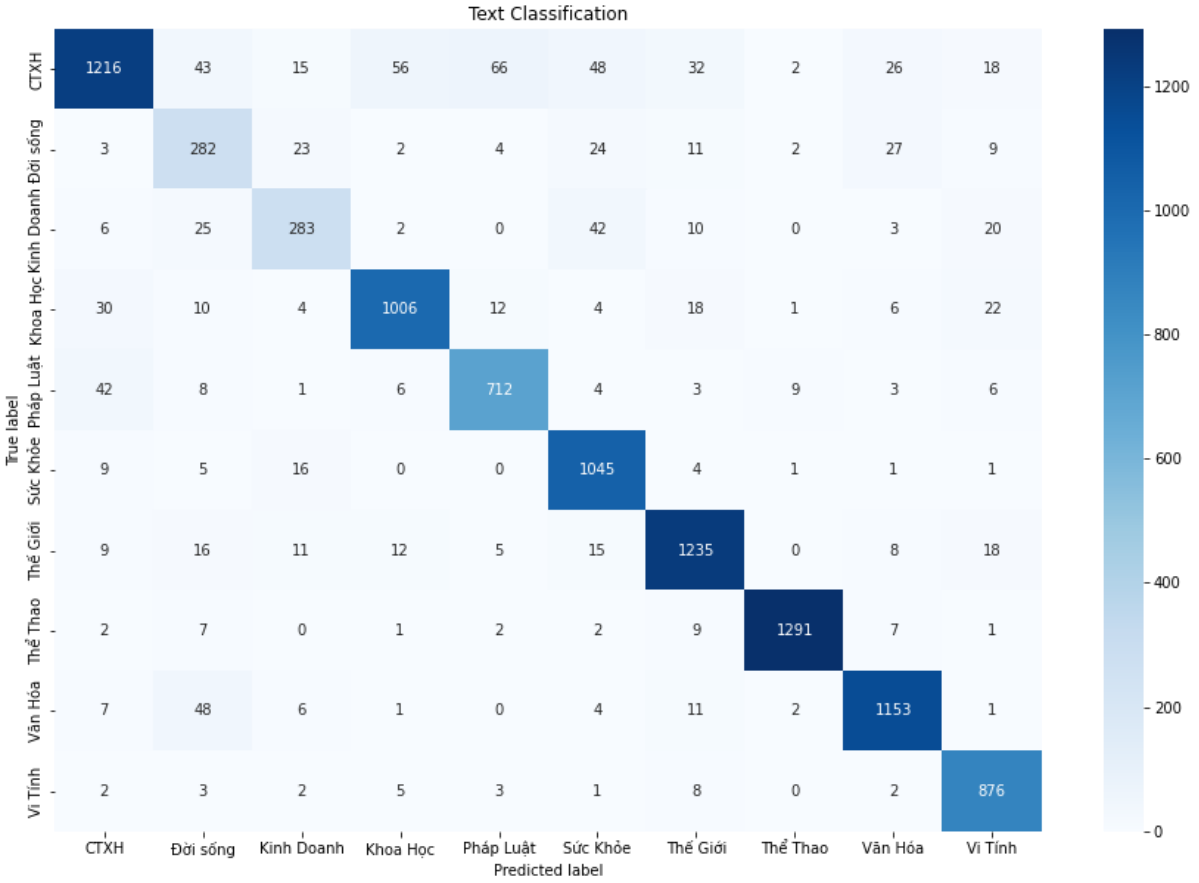
Hình 3.5 Đồ thị giá trị Accuracy của model

Sau khi train xong model ta tiến hành đánh giá model dựa trên tập test.

	precision	recall	f1-score	support
CTXH	0.91	0.82	0.86	1503
Đời Sống	0.70	0.60	0.64	374
Kinh Doanh	0.75	0.78	0.77	439
Khoa Học	0.92	0.87	0.89	1061
Pháp Luật	0.92	0.90	0.91	755
Sức Khỏe	0.89	0.96	0.92	1112
Thế Giới	0.94	0.92	0.93	1333
Thể Thao	0.98	0.97	0.98	1320
Văn Hóa	0.87	0.97	0.92	1245
Vi Tính	0.91	0.97	0.94	933
accuracy			0.90	10075
macro avg	0.88	0.88	0.88	10075
weighted avg	0.90	0.90	0.90	10075

Bảng 3.11 Giá trị dự đoán trên tập test

Để đánh giá model một cách trực quan hơn ta vẽ đồ thị heat_map dựa trên Confussion matrix mà ta tính được.



Hình 3.6 Đồ thị heat_map Confussion matrix

Từ đồ thị cũng như bảng giá trị dự đoán trên tập test, ta nhận được Accuracy trên tập test là 0.9 rất gần với kết quả ở tập validation (0.895). Ta thấy ở hai thể loại báo là Đời Sống và Kinh Doanh có dữ liệu ít hơn so với các thể loại còn lại, điều này làm cho model không đủ lượng data để train cho hai thể loại này dẫn đến dễ dự đoán sai như trên. Minh chứng là kết quả precision của 2 thể loại này chỉ đạt ở mức 0.7-0.75 trong khi các thể loại khác đều ở lân cận 0.9, điều này làm giảm chất lượng model.

Một số kết quả dự đoán:

```
Text Classification
Sentence:  kiến trái chiều thu hằng thi hoa_hậu hoàn_vũ tùy_tiện kiểu
đẹp thông_minh chẳng dám đi thi hoa_hậu thông_minh một_chút_xíu
chẳng_là đẹp lắm đằng gửi thanhle gửi ban biên tậptiêu đề se chang đẹp
lắm đằng hoa_hậu việt_nam chẳng trình_độ sánh_vai hoa_hậu_thế_giới gửi
hau hoan vu vô_cùng nực_cười hai thế_giới phạm thu hằng hoa_khôi bắc
hoa_hậu vn đi thi miss universe lạ cơ_quan_chức_quyền lên_tiếng
thất_vọng gửi du dang gửi ban văn hoátiêu đề phạm thu hằng miss universe
Result predict label:  8 (Van hoa)
True label:  8 (Van hoa)
```

Bảng 3.12 Kết quả dự đoán đúng

```
Text Classification

Sentence:  tịch_thu tàu nước_ngoài vùng_biển quảng_ninh hôm_qua ubnd
tỉnh quảng_ninh quyết_định tịch_thu tàu peace số_hiệu ching alastair
peter alexander sinh quốc_tịch điều_khiển tàu xuất_phát hong_kong
thâm_nhập trái_phép rừng_cấm quốc_gia mùn huyện đảo vân đồn ching
lực_lượng an_ninh điều_tra phát_hiện tạm đảo rồng đi xuống gắn máy tàu
giá triệu đồng vn tự_thuật ching nghề giáo_viên viết báo tự_do mục_đích
ngành tổng công_ty nhà_nước soát_xét_lại hoạt_động ngành kế_hoạch
mở_rộng chiếm_giữ địa_bàn khách_hàng tổ_chức dịch_vụ nước_ngoài
xâm_nhập ta mặt_khác nghiên_cứu thu_hút vốn chủ_động mở_cửa dần chuyển
đi đường thuyền_nhân việt_nam vượt_biên trái_phép hộ_chiếu chính_phủ
ching alastair peter alexander giấy_tờ tùy_thân chiều du_lịch
bất_hợp_pháp dẫn_độ trở_lại hong_kong

Processing...
Result predict label:  4 (Phap luat)
True label:  4 (Phap luat)
```

Bảng 3.13 Kết quả dự đoán đúng

```
Text Classification

Sentence:  vff đứng ngã đường họp trao_đổi quan_chức cao_cấp vff
tổ_chức đứng ngã đường vụ tiêu_cực mùa bóng nhức đầu quan_chức vff vụ
clb đông thép pomina đatp giám_đốc điều_hành vũ tiến thành bắt chủ_tịch
vff nguyễn trọng hỷ bức_xúc tuyên_bố kỷ_luật đội đatp hạng nhì hỷ
nhắc_nhở phát_biểu hớ hớ vụ tiêu_cực đatp báo_chí rầm_rộ pháp_lý con_số
hớ vff gửi công_văn đề_nghị cơ_quan công_an kết_luận căn_cứ xử đatp
văn_bản trả_lời vụ án điều_tra cung_cấp không_lẽ vff xử đatp dựa
báo_chí giúp đatp có_mặt họp tổng_kết giải đội dự league mùa tiến_hành
bốc_thăm xếp lịch thi_đấu vụ đatp vff hết_sức nhức đầu vụ giám_đốc sở
tdtt thừa thiên huế ngô văn trân điện_thoại mua_chuộc cầu_thủ hồ minh
đồng khánh_hòa vụ thoạt_tiên ông hỷ hớ tuyên_bố mạnh_tay song am_hiểu
luật đơn_giản duy_nhất bằng_chứng buộc_tội trân cuộn băng ghi_âm
trao_đổi cầu_thủ đồng song luật băng ghi_âm tham_khảo kết_tội trân
thừa_nhận trao_đổi thăm_dò đội mua_chuộc luật_pháp sờ gáy trân hai vff
trân tổ_chức đơn_giản nhắc khéo vị chủ_tịch qui_chế vff thành_viên
liên_đoàn triệu_tập toàn_thể ban chấp_hành vị thường_trực thường_vụ
toàn_thể bch đồng_cảm trân chờ hội_nghị kéo_dài vff lãnh_đạo đội bóng
giải_quyết hóc_búa bóng_đá vn

Processing...
Result predict label:  7 (The thao)
True label:  7 (The thao)
```

Bảng 3.14 Kết quả dự đoán đúng

```
Text Classification

Sentence:  mở_rộng lĩnh_vực dịch_vụ thành_phần kinh_tế phó thủ_tướng
vũ khoan kế_hoạch đầu_tư tài_chính phối_hợp cơ_quan nghiên_cứu cơ_chế
chính_sách đầu_tư khu_vực dịch_vụ hướng khuyến_khích xã_hội_hoá
mở_rộng thành_phần kinh_tế doanh_nghiệp vốn đầu_tư nước_ngoài tham_gia
phó thủ_tướng thông_qua quỹ phát_triển giúp lĩnh_vực nhà_nước đầu_tư
cơ_chế dịch_vụ đóng_góp ngân_sách chủ_động hội_nhập lĩnh_vực dịch_vụ
thị_trường phó thủ_tướng lĩnh_vực dịch_vụ phát_triển góp_phần đẩy
tăng_trưởng kinh_tế ngành bưu_chính viễn_thông tài_chính ngân_hàng
bảo_hiểm vận_tải du_lịch tốc_độ tăng_trưởng chất_lượng dịch_vụ phó
thủ_tướng lĩnh_vực dịch_vụ trống_không khai_thác quản_lý phân_bố
chủ_yếu hà_nội tp hcm ngành khả_năng xã_hội_hoá phát_triển lĩnh_vực
dịch_vụ phó thủ_tướng vũ khoan thông_báo phép thành_lập tổ công_tác
liên_ngành dịch_vụ thứ_trưởng kế_hoạch đầu_tư tổ_trưởng nhiệm_vụ
theo_dõi tổng_hợp tình_hình phát_triển dịch_vụ giúp kế_hoạch đầu_tư
cơ_quan trình chính_phủ cơ_chế chính_sách vĩ_mô phát_triển dịch_vụ
xây_dựng chỉ_thị thủ_tướng chính_phủ phát_triển dịch_vụ phục_vụ
kế_hoạch giai_đoạn phát_triển kinh_tế xã_hội chiến_lược hoàn_thành quý
iv phó thủ_tướng

Processing...
Result predict label:  3 (Kinh doanh)
True label:  3 (Kinh doanh)
```

Bảng 3.15 Kết quả dự đoán đúng

```
Text Classification

Sentence:  robot đạp xe khoa_học nhật ra_mắt robot nhỏ_bé đi xe_đạp
dừng ngã robot murata boy hãng murata chế_tạo kg cm lái_xe tốc_độ cm
giây điều_khiển máy_tính dây robot công_bố triển_lãm kết_hợp công_nghệ
tiên_tiến makuhari messe tokyo hôm kỹ_sư khâu chế_tạo robot thăng_bằng
đi xe_đạp giải_quyết gắn cảm_biến thân robot phép tốc_độ góc nghiêng
khuyến_nông hà_nội cây_xanh chỗ cây_xanh ủng_hộ đam_mê trồng chỗ
phát_triển mô_hình sinh_thái trồng hoa tận_mắt chứng_kiến ủng_hộ
tham_quan học_tập tổng_kết mô_hình khuyến_nông đất trang_trại sơn_thủy
lấn_chiếm công_trình xây_dựng phép thành_phố hà_nội văn_bản khôi_phục
phát_triển nông_thôn ký lãnh_đạo đổ lỗi anh_em xảy dự_án cục điều_khiển
thăng_bằng murata boy có_giá triệu yên tương_đương usd hãng murata
chế_tạo phiên_bản robot đi xe_đạp phiên_bản dừng xe ngã

Processing...
Result predict label:  2 (Khoa hoc)
True label:  2 (Khoa hoc)
```

Bảng 3.16 Kết quả dự đoán đúng

Từ những ví dụ trên ta thấy đối với các bình luận mang tính rõ ràng, thông tin không bị nhiễu thì model cho kết quả dự đoán rất tốt với độ chính xác rất cao.

Ngoài ra vẫn còn các trường hợp model dự đoán sai:

```
Text Classification

Sentence:  người_mẫu hiếu pêđê bắt hiếu pêđê kẻ môi_giới đường_dây
gái_gọi cao_cấp bắt danh_sách người_mẫu thân_thiết rơi cơ_quan
điều_tra loạt điện_thoại tâm_sự người_mẫu vnexpress liên_lạc người_mẫu
ngọc thuý quen_biết hiếu quan_hệ bình_thường người_mẫu make up chào_hỏi
xã_giao chút bàng_hoàng cầm_đầu đường_dây gái_gọi bắt thúy quen
tiếp_xúc tiếc thúy con_người hiếu đi cơ_quan công_an thu_thập danh_sách
điện_thoại người_mẫu hết_sức bình_thường quan_hệ make up lưu diễn_viên
người_mẫu kèm điện_thoại giả_sử danh_sách thúy nổi_tiếng lạ đổi di_động
cũ đi có_lý_do mấy làm_ăn hiếu người_mẫu ngọc nga dư_luận hết_sức
bức_xúc ảnh_hưởng nghề người_mẫu nghề cộng thông_tin lệch_lạc người_ta
thiện_cảm người_mẫu đối_tượng vụ hiếu pêđê hoạt_động nghệ_thuật
chuyên_nghiệp sàn_diễn đóng vai phụ xướng thành người_mẫu diễn_viên
chuyên_nghiệp thành_kiến dư_luận người_mẫu hy_vọng thoáng thông_cảm
hoạt_động nghệ_thuật chân_chính người_mẫu khao_khát hoàn_thiện
nghề_nghiệp người_mẫu hồ ngọc hà_giới người_mẫu nữ bàn_tán hiếu pêđê
bắt hà quen hà hiếu show diễn thời_trang đọc báo đăng hiếu hà bục_mình
chủ_nhiệm văn_phòng chính_phủ nguyễn công_sự văn_bản truyền_đạt kiến
ủng_hộ trang_trại sơn_thủy triển_khai dự_án khuyến_nông nông_nghiệp
khuyến_nông khuyến_lâm nông_nghiệp phát_triển nông_thôn sở nông_nghiệp
phát_triển nông_thôn hà_nội ủng_hộ dự_án góc_độ du_lịch sinh_thái cộng
khẩu_hiệu dân doanh_nghiệp trồng động_thổ khánh_thành bình_thường đi
dự lễ_lạt kiểu doanh_nghiệp làm_ăn anh_hùng làm_ăn đổ_vỡ tội_phạm
doanh_nghiệp tội_phạm đi đổ lỗi đi động_thổ khánh_thành dư_luận râm_ran
lãnh_đạo lãnh_đạo kia ủng_hộ xấu doanh_nghiệp nhảy bừa chụp ảnh vị
lãnh_đạo cố ảnh trưng hiện_tượng việt_nam thông_thường vị lãnh_đạo đi
trồng dịp đầu xuân_thế trồng lưu_niệm trang_trại sơn thủy dịp trồng
khuyến_nông ủng_hộ dự_án ủng_hộ tưởng du_lịch sinh_thái kết_hợp
buồn_cười nghề người_mẫu vạ_lây gái_gọi hà make up hiếu hà cung_cấp
ủng_hộ đất_đai xây_dựng đình_chỉ ủng_hộ mặt tưởng trang_trại sinh_thái
triển_khai quy_định pháp_luật lấn_chiếm đất thu_hồi toàn_quyền phone
thân hà hiếu phone ghi danh_sách công_an chẳng hiếu hà đồng_nghiệp
người_mẫu hiếu quan_hệ người_mẫu thân thuý hà mấy bữa_nay bận_rộn hà
đọc báo hiếu bắt đồng_nghiệp hơi ngỡ_ngàng hà công_an bắt đối_tượng
mệnh_danh người_mẫu diễn_viên báo_chí đăng đầy_đủ danh_tính viết tắt
ỡm người_mẫu nhột lắm đổi điện_thoại hà cần_thiết

Processing...
Result predict label:  8 (Van hoa)
True label:  4 (Phap luat)
```

Bảng 3.17 Trường hợp dự đoán sai

```
Text Classification

Sentence:  máy_tính gỗ mát_mắt mát môi_trường chán máy_tính màu trắng
công_ty thụy điển tung sản_phẩm diện_mạo phòng góp_phần bảo_vệ
môi_trường máy_tính bàn_phím gỗ triệu máy_tính mỹ kết_thúc cuộc_đời
bãi rác người_ta lo_ngại đống chất_thải điện_tử tiết hoá_chất_độc_hại
nhiễm môi_trường vỏ nhựa chứa hợp_chất ung_thư động_vật ngăn_chặn
công_ty máy_tính swedx sollentuna thụy điển chế_tạo màn_hình máy_tính
chuột bàn_phím bọc khung gỗ hàng nghìn sản_phẩm tiêu_thụ công_ty tung
sản_phẩm năm_ngoái màn_hình phẳng inch bọc gỗ sồi tần bì trị_giá euro
bàn_phím euro chuột euro công_ty gia_nhập thị_trường chuyên_gia
môi_trường eric williams đại_học liên_hợp quốc tokyo nhật nhận_định
máy_tính gỗ thể vị cứu_tinh môi_trường thiết_bị chứa độc_tố nằm
linh_kiện điện_tử sản_xuất máy_tính ngốn nguyên_liệu máy
lượng_hoá_chất nhiên_liệu gấp cân xe_hơi tủ_lạnh nguyên_liệu gấp đôi
cân williams tăng_cường biện_pháp ảnh_hưởng máy_tính môi_trường
liên_hợp quốc dự_định ban_hành pháp_chế sản_xuất máy_tính trách_nhiệm
tái_chế chất_thải điện_tử nghiêm_cấm chất_độc_hại thiết_bị

Processing...
Result predict label:  9 (Vi tinh)
True label:  2 (Khoa hoc)
```

Bảng 3.18 Trường hợp dự đoán sai

```
Text Classification

Sentence:   nguyễn công tạn trang_trại sơn_thủy trồng tạo_dựng
trang_trại sinh_thái sơn_thủy trùm sản_xuất heroin trịnh nguyên thủy
vi phạm_luật đất_đai nguyên phó thủ_tướng nguyễn công tạn sơn_thủy
trồng lưu_niệm hầu_như trang_trại bành_trướng bất_lực chính_quyền thưa
ảnh trồng lưu_niệm trang_trại trịnh nguyên thủy người_ta phỏng_đoán
trồng nguyên phó thủ_tướng nguyễn công tạn ảnh trang_trại sơn_thủy
trồng dịp tết chính_phủ đi trồng văn_hóa việt nam tết trồng trồng trồng
chỗ kia hà_nội mời chính_xác xe_cộ đầy_rẫy đông_đủ vị quan_khách mời
trồng trang_trại sơn_thủy mời đi mời đi đi văn_phòng chính_phủ bố_trí
góp cổ_phần trang_trại sơn_thủy thông_tin người_thân góp cổ_phần
thủy_chính mắng thông_tin hôm ta mời dự tiệc tuyệt_đối đừng hòngtrồng
mời hà_nội trang_trại đi ăn_uống tiền sòng_phẳng đừng hòng dư_luận
trịnh nguyên thủy_thường lui_tới hỏi_vợ thủy hoan nguyễn viết bịa_đặt
thủy người_ta đồn_thổi tạn góp vốn mua đất xu mét_vuông đất nguyên
thủy bắt cảm_tưởng trường_hợp xảy đồng_tiền người_ta phạm_tội ngày_mai
tử_hình hoan_nghênh bắt thủy gỡ gia_đình tan_nát ma_túy

Processing...
Result predict label:  4 (Phap luat)
True label:  0 (Chinh tri Xa hoi)
```

Bảng 3.19 Trường hợp dự đoán sai

3.7.2 Phân tích lỗi

Mô hình vẫn còn thiếu sót dẫn đến các lỗi dự đoán sai với các nguyên nhân:

Dữ liệu được sử dụng trong huấn luyện và kiểm thử hoàn toàn là dữ liệu thu thập từ mạng xã hội: mặc dù đã qua quá trình tiền xử lý, nhưng vẫn có thể có những mẫu không theo một chuẩn mực cú pháp, từ đó tạo thành các điểm nhiễu khi mô hình học. (Ví dụ: những câu quá ngắn, những câu không có dấu, câu có sử dụng tiếng nước ngoài, v.v,…).

Sự đa nghĩa của tiếng Việt cũng là một thách thức lớn với nghiên cứu này, đặc biệt với mạng xã hội do các trào lưu sử dụng từ mới, từ lóng, những từ không có trong từ điển, câu có hàm ý mỉa mai xuất hiện cùng với những từ bình thường khiến dữ liệu bị nhiễu.

Do có khá nhiều câu (hơn 2500 câu) có độ dài lớn hơn 500 từ nên việc cắt bỏ các từ phía sau làm ảnh hưởng đến ý nghĩa cũng như nhãn của câu nên khi dự đoán sẽ không đúng với nhãn ban đầu.

Khó khăn trong việc hiểu thông tin của bài báo bởi không phải bài báo nào cũng chỉ mang duy nhất một chủ đề (tính đa chủ đề), ví dụ như bài báo nói về một tội phạm kinh tế thì sẽ có liên quan đến cả kinh doanh và pháp luật, đó cũng là một khó khăn khi phân loại rõ chúng.

Một số hướng giải quyết với các lỗi trên:

Tiếp cận các nguồn dữ liệu khác: có định dạng cấu trúc, ngữ nghĩa, và đa dạng hơn về hình thức sẽ mang lại giá trị cao cho nghiên cứu.

Tập trung thực hiện tốt khâu gắn nhãn dữ liệu, có phân tích và định nghĩa rõ một câu như thế nào được gắn nhãn ở thể loại nào bởi có những bài báo tích hợp bởi nhiều yếu tố, nhiều vấn đề nên sẽ gặp khó khăn nếu không có nguyên tắc rõ ràng.

Trong phần tiền xử lý, ta loại bỏ các câu có độ dài quá lớn gây ảnh hưởng đến kết quả mô hình sau này.

Kết hợp với các phương pháp khác: như đã trình bày tại các phần trên, ở các mô hình phân tích quan điểm, một phương pháp mang lại nhiều hứa hẹn là các mô hình học

lai: kết hợp nhiều phương pháp nhằm mang đến kết quả tốt hơn. Đây cũng là một xu hướng mà nhiều nghiên cứu đang theo đuổi, hứa hẹn mang kết quả đáng mong chờ.

CHƯƠNG 4: TỔNG KẾT

4.1 Tổng kết

Việc phân loại văn bản là công việc cơ bản và có vai trò quan trọng trong việc nghiên cứu, khai thác lượng dữ liệu lớn hiện nay. Trong bài viết này, em đã trình bày phương pháp phân loại văn bản dựa trên mô hình LSTM. Thuật toán LSTM có sự liên kết giữa các đặc trưng trong câu, liên kết giữa từ chính và các từ ngữ cảnh giúp cải thiện kết quả của mô hình, với độ chính xác khoảng 90% về cơ bản hệ thống đã đáp ứng được yêu cầu về phân loại các thể loại báo. Qua thời gian thực hiện đề tài và kết quả đạt được của đề tài, bản thân em nhận thấy đã đem lại lượng kiến thức lớn và có tính ứng dụng cao trong công việc hiện tại. Với những gì đã thực hiện, bản thân em cũng tích lũy được một số kinh nghiệm trong lĩnh vực Machine Learning, Deep Learning trên cơ sở đó việc đề ra các hướng để hoàn thiện và vận dụng vào các lĩnh vực khác nhau ngoài lĩnh vực phân loại văn bản của chương trình.

Diễn giải công việc	Tổng thời gian	Tháng 2		Tháng 3		Tháng 4		Tháng 5	
		Tuần 1&2	Tuần 3&4	Tuần 1&2	Tuần 3&4	Tuần 1&2	Tuần 3&4	Tuần 1&2	Tuần 3
		1-14/2	15-28/2	1-15/3	16-31/3	1-15/4	16/30/4	1-14/5	15-22/5
Tìm hiểu đề bài, yêu cầu bài toán	2 tuần								
Tìm hiểu về phương pháp deep learning cho bài toán phân loại văn bản	2 tuần								
Tìm hiểu và xây dựng chương trình thu thập dữ liệu từ các trang báo điện tử	6 tuần								
Tìm hiểu các phương pháp làm sạch và chuẩn hóa dữ liệu	2 tuần								
Tiến hành xây dựng model	2 tuần								
Chỉnh sửa model đánh giá kết quả	2 tuần								
Viết báo cáo	1 tuần								

Bảng 4.1 kế hoạch thực hiện đề tài

4.2 Những vấn đề đã đạt được

Tìm hiểu, nghiên cứu về thực trạng vấn đề hiện tại, từ đó đưa ra được phát biểu của bài toán và lộ trình nghiên cứu thực hiện. Phần lý thuyết làm rõ khái niệm phân lớp văn bản, xây dựng hệ thống phân lớp văn bản, một số kĩ thuật phân lớp văn bản, lựa chọn đặc trưng văn bản.

Nghiên cứu, tìm hiểu một số công trình trong và ngoài nước có liên quan từ đó học hỏi được nhiều kiến thức, kinh nghiệm trong các lĩnh vực: học máy, xử lý ngôn ngữ tự nhiên, dữ liệu lớn,… áp dụng vào chương trình.

Phát triển, xây dựng bộ từ điển về nội dung các bài báo phục vụ cho quá trình tiền xử lý dữ liệu. Tuy nhiên chủ yếu dựa vào những kiến thức chủ quan và quá trình tìm kiếm trên Internet nên vẫn còn nhiều hạn chế và thiếu sót về từ vựng.

Thu thập dữ liệu liên quan đến các khía cạnh đề tài sử dụng, lên đến khoản 50 nghìn bài báo.

Thực hiện tiến trình tiền xử lý dữ liệu chuẩn hóa dữ liệu, tách từ, loại bỏ stop word, mở rộng dữ liệu bằng các phương pháp tăng cường dữ liệu cho văn bản.

Thực hiện mã hóa dữ liệu vào không gian vector theo mô hình Word2Vec.

Xây dựng, phát triển mô hình sử dụng mạng nơ-ron LSTM ứng dụng cho mô hình phân loại văn bản. Với độ chính xác lên đến 90%.

4.3 Những vấn đề chưa đạt được

Ngoài những vấn đề đã đạt được thì bài báo cáo vẫn còn nhiều thiếu sót như chưa thu thập đủ nhiều dữ liệu để cân bằng dữ liệu cho cả 10 class dẫn đến trường hợp dữ liệu mất cân bằng làm cho các class có ít dữ liệu dễ dự đoán sai gây ảnh hưởng đến kết quả của bài toán.

Do hạn chế về thời gian thực hiện nên việc thu thập dữ liệu chỉ dừng lại ở mức 50000 bài báo, nên khó có thể xây dựng một bộ từng vựng hoàn chỉnh có thể bao quát được hết các chủ đề của các loại báo.

Một hạn chế nữa là về phần cơ sở vật chất, bởi mỗi bài báo trung bình có độ dài lên đến 1000 từ/bài báo nhưng do dung lượng Ram máy tính cũng như Google Colab có giới hạn nên phải thu gọn model, input chỉ đưa vào 500 từ mỗi bài báo. Nếu có thể cho đầu vào nhiều từ hơn thì có thể model sẽ làm việc chính xác hơn.

4.4 Hướng phát triển

Theo kết quả đạt được thì định hướng phát triển về sau sẽ làm tốt hơn về giải thuật, dữ liệu và ứng dụng.

Bên cạnh đề tài cho lĩnh vực phân loại văn bản chương trình có thể mở rộng cho các lĩnh vực khác như: phân tích cảm xúc hay phân loại bình luận,...

Khả năng thu thập dữ sẽ được nâng lên, tạo được nguồn dữ liệu lớn hơn cho các hệ thống dữ liệu từ điển phân tích tiếng việt giúp tạo nguồn dữ liệu lớn, phân tích sâu hơn cho các chương trình có liên quan về sau. Từ đó kết quả sẽ đạt độ chính xác cao hơn.

Cải thiện mô hình phân loại bằng việc sử dụng kết hợp với các kỹ thuật khác để cải thiện kết quả và cải thiện độ chính xác bằng việc sử dụng kết hợp với các mô hình học sâu khác.

Tiếp tục nghiên cứu, thực hiện bài toán tóm tắt văn bản theo như lộ trình đề ra của luận văn tốt nghiệp.

Về mặt học thuật đề tài cũng đã có đưa ra được các nghiên cứu có liên quan gần với nội dung đề tài, nhưng cũng có nhiều nét riêng mà đề tại mạng lại cho thực tiễn cũng như với ngôn ngữ tiếng Việt. Với tính ứng dụng thực tiễn chương trình sẽ tiếp tục được nghiên cứu mở rộng quy mô của chương trình ra nhiều lĩnh vực hơn nữa, thu thập thêm nhiều nguồn dữ liệu, cũng như từng bước hình thành một chương trình có tính ứng dụng cao hơn trong tương lai không xa.

TÀI LIỆU THAM KHẢO

[1]. Gupta, S. (2018, July 9). Text Classification: Applications and Use Cases – Towards Data Science.

[2]. Karani, D. (2020, September 2). Introduction to Word Embedding and Word2Vec - Towards Data Science.

[3]. Pham, G., Kohnert, K., & Carney, E. (2008). Corpora of Vietnamese Texts: Lexical effects of intended audience and publication place. Behavior Research Methods, 40(1), 154–163.

[4]. Phat, H. N., & Anh, N. T. M. (2020). Vietnamese Text Classification Algorithm using Long Short Term Memory and Word2Vec. Informatics and Automation, 19(6), 1255–1279.

[5]. Phi, M. (2020, June 28). Illustrated Guide to LSTM's and GRU's: A step by step explanation.

[6]. Prasad KM, S., & Reddy, D. H. (2019). Text Mining: Classification of Text Documents using Granular Hybrid Classification Technique. International Journal of Research in Advent Technology, 7(6), 1–8.

[7]. Sharma, S. (2021, July 4). Activation Functions in Neural Networks - Towards Data Science.

[8]. Toan Pham Van and Ta Minh Thanh, "Vietnamese news classification based on BoW with Keywords Extraction and Neural Network", 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES), 2017.

[9]. Thangaraj, M., & Sivakami, M. (2018). Text Classification Techniques: A Literature Review. Interdisciplinary Journal of Information, Knowledge, and Management, 13, 117–135.

[10]. Tran Thi Lan Huong (2012). Research on automatic classification of Vietnamese press documents on natural resources and environment.