ĐỀ XUẤT THUẬT TOÁN PHÂN LOẠI VĂN BẢN TIẾNG VIỆT SỬ DỤNG MẠNG LSTM VÀ WORD2VEC

Nguyễn Thị Minh Ánh, Nguyễn Hữu Phát*

Tóm tắt: *Hiện nay, phân loại văn bản là một trong những ứng dụng cơ bản của xử lý ngôn ngữ tự nhiên. Đặc biệt, những thành tựu gần đây của các mạng học sâu cho thấy các phương pháp học sâu đang làm rất tốt trong việc phân loại văn bản. Các phương pháp này cho thấy hiệu quả trong việc phân loại văn bản với ngôn ngữ tiếng Anh. Tuy nhiên, hiện nay, trong phạm vi nghiên cứu của chúng tôi, không có nhiều nghiên cứu đối với văn bản tiếng Việt. Do đó, trong nghiên cứu này, chúng tôi sử dụng mạng Long Short Term Memory (LSTM) kết hợp với Word2vec để phân loại văn bản nhằm cải thiện hiệu suất và độ chính xác. Kết quả đánh giá mô hình trên bộ văn bản tiếng Việt VNTC [1] đã cho thấy sự khả thi và hứa hẹn áp dụng trong thực tế.*

Từ khóa: Phân loại văn bản; Xử lý ngôn ngữ tự nhiên; Xử lý dữ liệu; Long Short Term Memory; Word2vec.

1. ĐẶT VẤN ĐỀ

Với xu thế bùng nổ dữ liệu như hiện nay, lượng dữ liệu văn bản phải xử lý ngày càng nhiều. Vấn đề đặt ra là làm thế nào để khai thác thông tin từ nguồn dữ liệu này. Rõ ràng, ta phải hiểu rõ bản chất của dữ liệu văn bản, hiểu rõ đặc trưng của các dữ liệu này để có thể có được những phương pháp luận cần thiết. Việc phân loại văn bản tự động sẽ giúp tiết kiệm thời gian và công sức.

Các mô hình mạng nơ-ron nhân tạo học sâu(Deep Neural Network - DNN) đã đạt nhiều thành công trong việc xử lý ngôn ngữ tự nhiên. Những ứng dụng của Deep Learning trong xử lý ngôn ngữ tự nhiên có thể kể đến như thuật toán phân cụm từ [12], rút trích nội dung chính của văn bản tiếng Việt [13],… Đối với phân loại văn bản, đã có nhiều nghiên cứu và mô hình được sử dụng.

Với ngôn ngữ tiếng Anh, các bài toán phân loại văn bản cũng đã được nghiên cứu dựa trên các mạng nơ-ron nhân tạo học sâu [14, 19]. Trong [14], tác giả sử dụng LSTM, LSTM + Word2vec. Việc sử dụng mô hình LSTM + Word2Vec được đào tạo trước có thể giải quyết tốt vấn đề xử lý dữ liệu nhiều chiều do các phương pháp truyền thống gây ra. Trong phương pháp [19], tác giả sử dụng mạng Convolutional Neural Network (CNN) và Recurrent Neurral Network (RNN) để phân loại, kết quả trong phương pháp này cho thấy, hiệu quả của mạng RNN tốt hơn so với mạng CNN, tuy nhiên, độ chính xác còn chưa cao do bộ dữ liệu chưa đủ lớn. Còn trong [15], tác giả đã kết hợp giữa hai mô hình là CNN và Bidirectional Recurrent Neural Network (BRNN). Trong mô hình này, tác giả đã sử dụng lớp bidirectional để thay thế cho lớp pooling trong CNN nhằm để giảm những thông tin chi tiết và giúp lưu trữ những phụ thuộc dài hạn của các chuỗi đầu vào. Ngoài ra, để giảm bớt tính phức tạp trong tính toán, các tác giả đã sử dụng thuật toán Support Vector Machine (SVM) [16]. Thuật toán SVM có hiệu quả trong việc xử lý trên không gian số chiều cao, sử dụng tốt trong bài toán phân loại văn bản. Thuật toán giúp tiết kiệm bộ nhớ do chỉ có một tập hợp con các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định. SVM cũng có tính linh hoạt bởi cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính. Tuy nhiên, SVM chưa thể hiện rõ tính xác suất do việc phân lớp của nó chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng.

Đối với phân loại văn bản tiếng Việt, cũng có những nghiên cứu về các phương pháp để

phân loại. Phân loại văn bản tiếng Việt sử dụng phương pháp truyền thống và phương pháp phân loại dựa trên mô hình chủ đề [17]. Mô hình chủ đề là một loại mô hình thống kê để phát hiện ra các "chủ đề" trừu tượng có trong một tập các tài liệu. Phương pháp này đảm bảo tính ổn định và cho độ chính xác tương đối nhưng để xây dựng chủ đề là khá khó và tốn nhiều thời gian cũng như chi phí. Thuật toán SVM cũng được áp dụng cho phân loại văn bản trong tiếng Việt [18]. Bộ phân loại này có khả năng tự động điều chỉnh các tham số để tối ưu hóa hiệu suất phân loại. Tuy nhiên, so với ngôn ngữ là tiếng Anh, bộ phân loại văn bản cho tiếng Việt cho kết quả thấp hơn nhiều (độ chính xác đối với ngôn ngữ tiếng Việt là 80.72% so với 89% của bộ phân loại văn bản tiếng Anh [16]). Trong [20], tác giả đã sử dụng các phương pháp truyền thống như Naïve Bayes (NB) và Maximum entropy (Maxent). NB là một thuật toán đơn giản, được sử dụng để giải quyết vấn đề liên quan tới phân loại dữ liệu dựa trên phương pháp thống kê. Tuy nhiên, nhược điểm của phương pháp này là không có liên kết giữa các đặc trưng. Maxent là phương pháp tối đa hóa entropy bằng cách ước tính xác suất nhãn dựa trên các đặc trưng của câu. Ngoài ra, cũng trong [20], tác giả cũng sử dụng phương pháp học sâu là LSTM và Bidirectional Long Short Term Memory (Bi-LSTM). Trong [22], tác giả đã đề xuất một kiến trúc mạng nơ-ron mới để thực hiện phân loại và đã đạt được kết quả tốt trong nghiên cứu. Trong nghiên cứu [28], tác giả đã sử dụng kết hợp CNN và LSTM cho việc phân tích nội dung câu văn. CNN và LSTM có những ưu điểm riêng, CNN đưa ra những liên kết các từ cục bộ còn với LSTM sẽ liên kết các từ theo dạng chuỗi và liên kết các từ trong quá khứ.

Sự phát triển các mô hình học sâu trong đề tài phân loại văn bản đã cho thấy những cải tiến trong việc phân loại văn bản tiếng Việt. Việc thử nghiệm của chúng tôi đã cải thiện kết quả trong việc phân loại văn bản tiếng Việt so với các phương pháp truyền thống và cả các mạng học sâu.

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. Bài báo này sử dụng để giải quyết một số vấn đề như phân loại đề tài trong văn bản, phân loại nhận xét tích cực - tiêu cực,… Phân loại văn bản là phương pháp gán nhãn một tài liệu văn bản với các thể loại từ bộ dữ liệu được xác định từ trước. Ví dụ có 1 bộ dữ liệu của các tin tức được ký hiệu là:

$$N = (n_1, n_2, \dots, n_n) \tag{1}$$

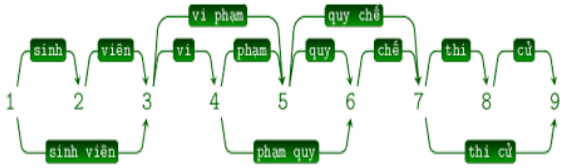
Các nhãn dãn ứng với các tin tức được ký hiệu là:

$$C = (c_1, c_2, \dots, c_m) \tag{2}$$

Và trong bài viết này, chúng tôi sẽ nghiên cứu để tự động gán nhãn các bài viết theo các nhãn C.

Phương pháp phân loại văn bản cổ điển thường được giải quyết bằng cách dựa vào bộ từ điển, các phương pháp học máy. Tuy nhiên, khi các phương pháp học sâu ra đời đã cho hiệu quả tốt hơn rất nhiều. Để phân loại văn bản bằng phương pháp học sâu, CNN và LSTM được sử dụng rộng rãi hơn cả. Tuy nhiên, các phương pháp này thường được áp dụng cho tiếng Anh, còn đối với tiếng Việt, chúng được sử dụng rất hạn chế. Hai ngôn ngữ này có nhiều điểm khác biệt và điểm khác biệt gây khó khăn trong việc xử lý tiếng Việt đó là xác định ranh giới của từ. Đối với tiếng Anh, "từ là một nhóm các ký tự có nghĩa được tách biệt với nhau bởi khoảng trắng trong câu" (Từ điển Webster), do vậy, việc tách từ trở nên rất đơn giản. Còn đối với tiếng Việt, ranh giới từ không được xác định và mặc định là khoảng trắng mà phụ thuộc vào ngữ cảnh dùng câu tiếng Việt. Ví dụ các từ trong tiếng Anh là "book", "cat", "language" thì trong tiếng Việt là "quyển sách", "con mèo", "ngôn ngữ",… Một số đặc tính chính của từ trong tiếng Việt là:

- Từ ở dạng nguyên thể, hình thức và ý nghĩa của từ độc lập với cú pháp.
- Từ bao gồm từ đơn và từ phức, bao gồm từ láy, từ ghép.
- Từ được cấu trúc từ "tiếng". Việc nhận biết từ trong tiếng Việt được gọi là phân cụm từ:



Hình 1. *Vấn đề phân cụm từ trong tiếng Việt [10].*

Trong hình 1, có nhiều hơn một cách để hiểu câu văn này:

1. Sinh_viên vi_phạm quy_chế thi_cử.

2. Sinh_viên vi phạm_quy chế thi_cử. – Câu văn này không mang ý nghĩa.

Phân cụm từ là bước quan trọng trong tiền xử lý văn bản tiếng Việt. Nếu câu văn được hiểu theo cách hiểu thứ nhất, ta có thể phân loại câu này với nhãn "pháp luật". Tuy nhiên, nếu câu văn được phân cụm theo cách thứ 2, ta có thể phân loại nhầm sang một nhãn khác. Do vậy, độ chính xác của bước phân cụm từ là rất quan trọng. Nếu việc phân cụm từ đạt kết quả không tốt có thể dẫn đến phân loại nhãn của văn bản bị sai lệch.

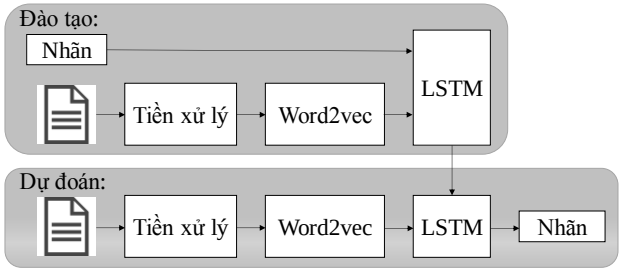
Trong bài viết này, chúng tôi sử dụng mô hình LSTM để phân loại văn bản. LSTM có các cổng giúp lọc thông tin như cổng đầu vào, cổng nhớ. Nhờ đó, các dữ liệu có ích từ thời điểm rất lâu trong quá khứ vẫn có tác dụng điều chỉnh. Trong bài báo chúng tôi kết hợp sử dụng các phương pháp LSTM và Word2vec. Khi sử dụng kết hợp với Word2vec, hiệu quả mô hình được cải thiện.

Nội dung bài báo được tổ chức như sau. Sau phần giới thiệu, chúng tôi sẽ trình bày các công việc liên quan. Phần 3 trình bày chi tiết về cơ sở lý thuyết. Phần 4 cho thấy kết quả thử nghiệm của mô hình. Phần 5 là kết luận và hướng phát triển mới.

2. NỘI DUNG CẦN GIẢI QUYẾT

2.1. Xây dựng lý thuyết

Quy trình thực hiện bài toán:



Hình 2. *Sơ đồ cấu trúc hệ thống.*

Với những yêu cầu và mục đích đã đặt ra cho hệ thống, bài toán bao gồm hai bước chính là đào tạo và dự đoán.

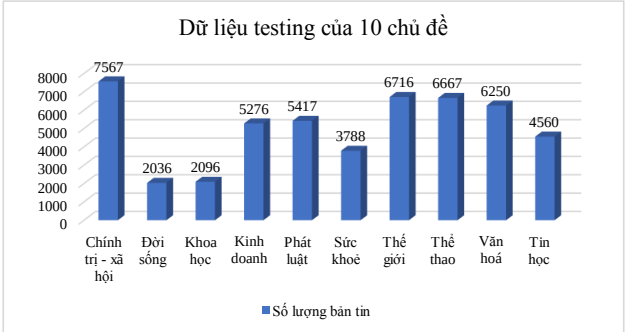
2.1.1. Thu thập dữ liệu

Khó khăn trong việc phân loại văn bản tiếng Việt là bộ dữ liệu các chủ đề bằng tiếng Việt còn hạn chế và độ chính xác chưa cao. Các bộ dữ liệu thường chỉ gồm từ 50-100 văn bản thô. Tuy nhiên, hiện nay có nhóm nghiên cứu của Hoàng Công Duy Vũ và đồng nghiệp [1] đã tạo ra bộ dữ liệu VNTC [1] có thể đáp ứng được điều kiện, nhu cầu nghiên

cứu bài toán này. Bộ dữ liệu được cập nhật và thu thập tin tức điện tử online từ các báo điện tử là: VnExpress [6], TuoitreOnline [7], ThanhNienOnline [8], NguoiLaoDong [9] theo từng chủ đề được người dùng chọn lựa. Bộ dữ liệu bao gồm 10 chủ đề với 33756 bài viết cho tập train và 50373 bài viết cho tập test:



Hình 3. *Tập dữ liệu train [1].*



Hình 4. *Tập dữ liệu test [1].*

2.1.2. Tiền xử lý dữ liệu

a. Tách từ tiếng Việt

Do vấn đề đã nêu từ phần 1, việc tách từ tiếng Việt sẽ ảnh hưởng nhiều tới kết quả của bài toán. Hiện nay, có nhiều nghiên cứu về tách từ tiếng Việt như trong [23], tác giả sử dụng phương pháp Single Classification Ripple Down Rules (SCRDR) cho tách từ và đạt hiệu quả 97.9%. Trong [24], tác giả đã sử dụng mô hình mạng nơ-ron để gắn thẻ part-of-speech (POS) và phân tích cú pháp phụ thuộc và trong nghiên cứu cũng đã đạt 97.9%. Một phương pháp mới cũng được sử dụng trong tách từ tiếng Việt là trong [25], tác giả sử dụng tiếp cận lai (hybrid) để tự động hóa văn bản tiếng Việt. Cách tiếp cận này kết hợp các kỹ thuật trạng thái hữu hạn, phân tích cú pháp biểu thức và việc chiến lược khớp tối đa được tăng cường bằng phương pháp thống kê giải quyết vấn đề mơ hồ của phân đoạn từ. Và kết quả của nghiên cứu này đã đạt độ chính xác là 98.5%. Trong nghiên cứu lần này, chúng tôi sử dụng phương pháp Tokenizer trong [25] để thực hiện tách từ. Với phương pháp này cũng có nhiều công cụ hỗ trợ như Coccoc-tokenizer cho ngôn ngữ C++, VnCoreNLP cho ngôn ngữ Java,… Đối với ngôn ngữ Python, có một số thư viện hỗ trợ là Pyvi, vTools, Vietseg,… Để giải quyết vấn đề tách từ trong bài này, chúng tôi đã sử dụng thư viện tách từ tiếng Việt Pyvi [2]. Bộ tách từ đã cho kết quả khá tốt trên bộ dữ liệu VNTC.

b. Làm sạch dữ liệu

Sau khi tách từ, văn bản còn xuất hiện nhiều ký tự đặc biệt, dấu câu,… những thành phần này làm giảm hiệu quả trong quá trình xử lí. Trong phần này, chúng tôi chuyển tất cả những từ in hoa về chữ in thường, xóa bỏ các dấu câu.

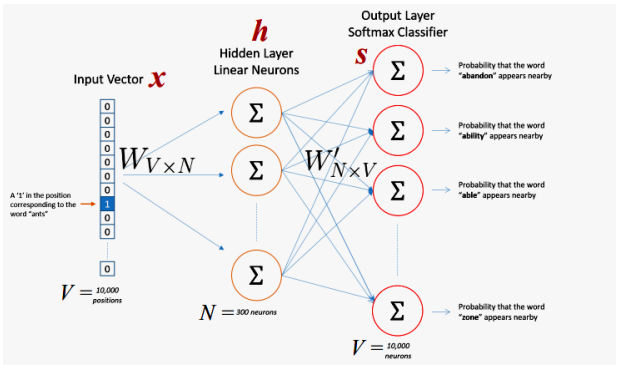
c. Tách bỏ stop-word

Tiếp theo, chúng tôi sẽ loại bỏ "stop-word". "stop-word" được hiểu là những từ có mức độ không quan trọng đối với các văn bản và cả trong việc phân loại văn bản. Một số từ nối, từ chỉ số lượng ("và", "mặc_dù", "tuy_nhiên",…) không mang tính phân biệt khi phân loại. Ngoài ra, "stop-word" còn là những từ không có giá trị phân loại như từ xuất hiện ở hầu khắp các văn bản hay dùng không phổ biến trong văn bản. Việc loại bỏ những từ này sẽ làm giảm thời gian tính toán và giảm bộ nhớ trong quá trình xử lí. Trong bước này, chúng tôi sử dụng bộ từ điển stopword-vietnamese [3] để loại bỏ những từ không cần thiết trong văn bản. Sau khi loại bỏ "stop-word", văn bản có thể xem như là một tập hợp các từ quan trọng còn lại để biểu diễn văn bản.

2.1.3. Trích xuất đặc trưng

Sau khi loại bỏ stopword, ta đã thu được văn bản với các từ "quan trọng". Tuy nhiên, có thể thấy rằng, số đặc trưng của tất cả văn bản đang xử lí là rất lớn. Chính vì vậy, phân loại dựa trên các đặc trưng này cần phải có cách xử lí, lựa chọn đặc trưng nhằm rút ngắn số chiều của không gian đặc trưng.

Tiếp theo, các từ trong văn bản sẽ được vector hóa, cụ thể mỗi từ được biểu diễn bằng một vector. Việc biểu diễn từ thành một vector đóng vai trò quan trọng trong bài toán này. Word Embedding có nhiệm vụ ánh xạ từ một hoặc một cụm từ trong bộ từ vựng tới một vector số thực từ không gian một chiều cho mỗi từ tới không gian các vector liên tục. Trong phần này, chúng tôi sử dụng mô hình Word2vec. Word Embedding là việc biểu diễn các từ dưới dạng các vector số thực với số chiều xác định. Word2vec là một mô hình học tập không giám sát được tạo ra năm 2013. Word2vec là một trong những mô hình đầu tiên về Word Embedding sử dụng mạng neural, 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 một tập các từ (vocabulary) sang một không gian vector ở đây mỗi vector được biểu diễn bởi n số thực. Word2vec là một mạng neural gồm 2 lớp với duy nhất 1 tầng ẩn. Đầu vào là một tập lớn các từ và sinh ra không gian vector, trong đó, với mỗi từ duy nhất được gắn với một vector tương ứng trong không gian theo biểu thức:



Hình 5. *Mô hình Word2Vec [5].*

- Đầu vào của mô hình là one – hot – vector mỗi từ sẽ có dạng $x_1, x_2,\ldots, x_v$, trong đó, V là số các từ vựng(vocabulary). Mỗi từ là một vector, trong đó có giá trị 1 tương đương với thứ tự của từ trong vocabulary và còn lại sẽ là 0.

- Ma trận giữa đầu vào và lớp ẩn là ma trận W (có số chiều là VxN) có hàm kích hoạt là linear function, ma trận giữa lớp ẩn và đầu ra là W' (có số chiều là NxV) có hàm kích hoạt là softmax.

- Mỗi hàng của W là vector N chiều đại diện cho v_w là mỗi từ trong lớp đầu vào. Mỗi

hàng của W là v_w^T.

- Từ lớp ẩn đến đầu ra có ma trận là W' = $w'_{i,j}$. Tính toán score u_i cho mỗi từ :

$$u_j = v'_{w_j}.h \tag{3}$$

Trong đó, v'_{w_j} là cột của ma trận j trong W'. Sau đó, ta sử dụng hàm kích hoạt softmax như sau:

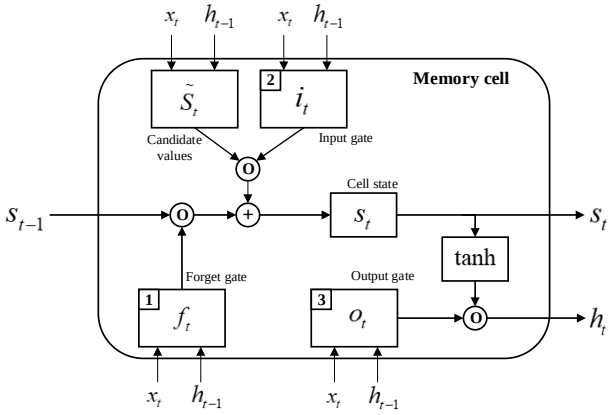
$$P(w_j|w_I) = y_j = \frac{exp(u_j)}{\sum_{j'=1}^{V} exp(u_{j'})} = \frac{exp\left(v'^T_{w_j} v_{w_I}\right)}{\sum_{j'=1}^{V} exp\left(v'^T_{w_j}, v_{w_I}\right)} \tag{4}$$

Với v_w và $v_{w'}$ là 2 vector đại diện cho từ w đến từ ma trận W và W'.

Sau quá trình huấn luyện mô hình, trọng số của các vector của từng từ được cập nhật liên tục. Từ đó, chúng ta có thể thực hiện tính toán bằng các khoảng cách. Những từ càng gần nhau về khoảng cách thường là những từ hay xuất hiện cùng nhau trong văn cảnhhoặc các từ đồng nghĩa, hay các từ thuộc cùng một trường từ vựng.

2.1.4. Long Short Term Memory (LSTM)

LSTM là một phiên bản mở rộng của mạng Recurrent Neural Network (RNN), nó được thiết kế để giải quyết các bài toán về phụ thuộc xa (long-term dependencies). RNN là mạng nơ-ron có chứa vòng lặp. Mạng này có khả năng lưu trữ thông tin, thông tin được truyền từ lớp này sang lớp khác. Đầu ra của lớp ẩn phụ thuộc vào thông tin của các lớp tại mọi thời điểm. RNN đã được sử dụng phổ biến trong xử lý ngôn ngữ tự nhiên hay các bài toán có dữ liệu tuần tự. Tuy nhiên, do kiến trúc của RNN khá đơn giản nên khả năng liên kết các lớp có khoảng cách xa là không tốt. Nó cơ bản không có khả năng ghi nhớ thông tin từ các dữ liệu có khoảng cách xa, và do đó, những phần tử đầu tiên trong chuỗi đầu vào thường không có nhiều ảnh hưởng đến kết quả dự đoán phần tử cho chuỗi đầu ra các bước sau. Nguyên nhân của việc này là do RNN chịu ảnh hưởng bởi việc đạo hàm bị thấp dần trong quá trình học – biến mất đạo hàm (vanishing gradient). Mạng LSTM được thiết kế để khắc phục vấn đề này. Cơ chế hoạt động của LSTM là chỉ ghi nhớ những thông tin liên quan, quan trọng cho việc dự đoán, còn các thông tin khác sẽ được bỏ đi.



Hình 6. *Mô hình LSTM [11].*

Mạng LSTM có thể bao gồm nhiều tế bào LSTM liên kết với nhau. Ý tưởng của LSTM là bổ sung thêm trạng thái bên trong tế bào (cell internal state) s_t và ba cổng sàng lọc thông tin đầu vào và đầu ra cho tế bào bao gồm cổng quên f_t, cổng đầu vào i_t và cổng đầu ra o_t. Tại mỗi bước thời gian t, các cổng lần lượt nhận giá trị đầu vào x_t (đại diện cho một

phần tử trong chuỗi đầu vào) và giá trị h_{t-1} có được từ đầu ra của các ô nhớ từ bước thời gian trước đó *t – 1*. Các cổng đều có chức năng sàng lọc thông tin với mỗi mục đích khác nhau. Các cổng được định nghĩa như sau:

- Cổng quên: Có chức năng loại bỏ những thông tin không cần thiết nhận được khỏi trạng thái tế bào bên trong.

- Cổng đầu vào: Giúp sàng lọc những thông tin cần thiết để được thêm vào trạng thái tế bào bên trong.

- Cổng đầu ra: Có chức năng xác định những thông tin nào từ các trạng thái tế bào bên trong được sử dụng như đầu ra.

Trong quá trình thực hiện, s_t và các giá trị đầu ra h_t được tính toán như sau:

Ở bước đầu tiên, tế bào LSTM quyết định những thông tin cần được loại bỏ từ các trạng thái tế bào bên trong ở bước thời gian trước đó s_{t-1}. Giá trị f_t của cổng quên tại bước thời gian t được tính toán dựa trên giá trị đầu vào hiện tại x_t, giá trị đầu ra h_{t-1} từ tế bào LSTM ở bước trước đó và độ lệch (bias) b_f của cổng quên. Hàm sigmoid biến đổi tất cả các giá trị kích hoạt (activation value) về miền giá trị trong khoảng từ 0 và 1 theo công thức:

$$f_t = \sigma(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \quad (5)$$

Ở bước thứ 2, tế bào LSTM xác định những thông tin nào cần được thêm vào các trạng thái tế bào bên trong s_t. Bước này bao gồm hai quá trình tính toán đối với $\tilde{s}_t$ và f_t. $\tilde{s}_t$ biểu diễn những thông tin có thể được thêm vào các trạng thái tế bào bên trong:

$$\tilde{s}_t = tanh(W_{\tilde{s},x}x_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}}) \quad (6)$$

Giá trị i_t của cổng đầu vào tại bước thời gian t được tính:

$$i_t = \sigma(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \quad (7)$$

Ở bước tiếp theo, giá trị mới của trạng thái tế bào bên trong s_t được tính toán dựa trên kết quả thu được từ các bước trên:

$$s_t = f_t * s_{t-1} + i_t * \tilde{s}_t \quad (8)$$

Cuối cùng, giá trị đầu ra h_t:

$$o_t = \sigma(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \quad (9)$$

$$h_t = o_t * tanh(s_t) \quad (10)$$

Trong đó:

$W_{\tilde{s},x}$, $W_{\tilde{s},h}$, $W_{f,x}$, $W_{f,h}$, $W_{i,x}$, $W_{i,h}$ là các ma trận trọng số trong mỗi tế bào LSTM.

b_f, $b_{\tilde{s}}$, b_i, b_o là các vector bias.

2.2. Thực nghiệm

2.2.1. Dataset

Như đã trình bày ở trên, chúng tôi sử dụng tập dữ liệu VNTC [1] bao gồm 33756 bài viết cho tập train và 50373 bài viết cho tập test.

2.2.2. Tiền xử lý đầu vào

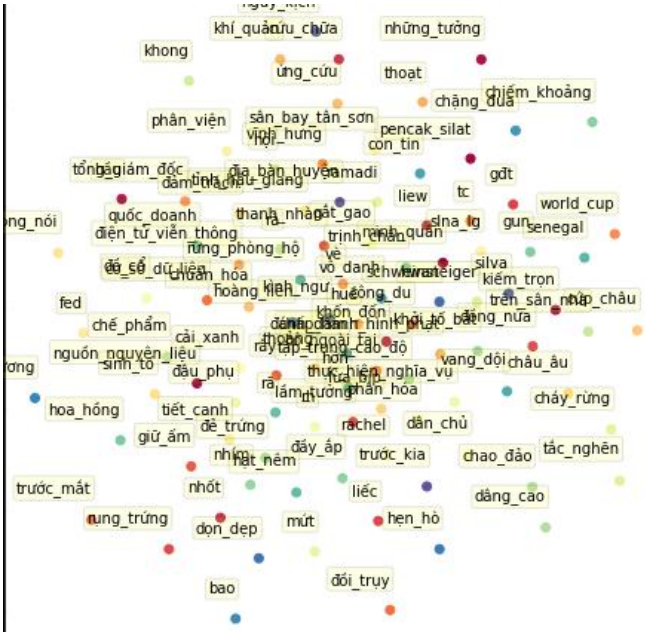
Các văn bản đầu vào sẽ được tách từ thông qua thư viện pyvi [3] và được làm sạch, tách stopword, xóa bỏ các ký tự đặc biệt, dấu câu.

2.2.3. Trích xuất đặc trưng

Trong bài toán này, đầu tiên chúng chúng tôi chuyển văn bản sang không gian vector là

tạo một thư mục để biểu diễn tài liệu. Mỗi từ sẽ được đưa về dạng vector chỉ bao gồm một số 1 và các thành phần còn lại là 0.

Trong bài toán này, chúng tôi sẽ sử dụng thêm Word2vec giúp cải thiện việc xuất đặc trưng cho văn bản. Khi thực hiện bài toán, sau khi thử nghiệm tập pre-train có sẵn của thư viện gensim không có nhiều sự phù hợp với bộ dữ liệu của mình. Với tập dữ liệu lên tới hơn 80000 văn bản, chúng tôi đã train lại mô hình Word2vec để có thể cải thiện cho bài toán của mình với embedding_size = 300, window_size = 5. Dưới đây là hình ảnh biểu diễn một số từ của model đã train sử dụng t-SNE.



Hình 7. *Các từ trong không gian vector từ Word2Vec.*

2.2.4. Phân loại văn bản

Trong bài toán này, do một bản tin thường dao động từ 500 – 1000 từ nên để có thể giảm kích thước đầu vào cho mạng thì chúng tôi lựa chọn độ dài tối đa của mỗi văn bản là 400, kích thước của word embedding là 300. Việc lựa chọn kích thước văn bản và không gian vector để có thể giải quyết vấn đề về dữ liệu nhiều chiều của các phương pháp truyền thống. Chúng tôi huấn luyện mô hình với 1 lớp ẩn có 128 units, sử dụng RMSProp với tỷ lệ học (learning rate) là 0,001 cho hàm tối ưu, sử dụng dropout là 0,4.

2.2.5. Tiêu chí đánh giá

Để đánh giá hiệu suất của mô hình, chúng tôi sử dụng Precision, Recall và F1-score làm tiêu chí đánh giá của bài toán này. Để minh họa ý nghĩa của các chỉ số này, confusion matrix [4] được thể hiện dưới bảng sau:

Bảng 1. *Confution Matrix.*

	Negative Pre	Positive Pre
Negative Act	True Negative(TN)	False Positive(FP)
Positive Act	False Negative(FN)	True Positive(TP)

Trong đó:

- TN là kết quả trong đó mô hình dự đoán chính xác lớp negative.
- TP là kết quả trong đó mô hình dự đoán chính xác lớp positive.

- FN là kết quả trong đó mô hình dự đoán không chính xác lớp negative.
- FP là kết quả trong đó mô hình dự đoán không chính xác lớp positive.

Precision là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm mô hình dự đoán là Positive:

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

Recall là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm thật sự là Positive (hay tổng số điểm được gán nhãn là Positive ban đầu):

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0):

$$\frac{2}{F1} = \frac{1}{Precision} + \frac{1}{Recall} \tag{13}$$

3. KẾT QUẢ VÀ THẢO LUẬN

Kết quả của mô hình mà chúng tôi đã thực hiện:

Bảng 2. *So sánh kết quả thuật toán.*

	Precision	Recall	F1-score
LSTM	92.36%	91.81%	92,09%
LSTM + Word2Vec (Mô hình đề xuất)	95,55%	95,93%	95,74%

Từ bảng 2 cho thấy kết quả của việc phân loại văn bản tiếng Việt, độ chính xác của mô hình phân loại văn bản sử dụng LSTM kết hợp với Word2vec cao hơn so với mô hình phân loại văn bản chỉ sử dụng LSTM. Do mô hình Word2Vec được đào tạo trước tương đương với việc tăng khối lượng đào tạo một cách gián tiếp. Hơn nữa, Word2vec có thể ngăn chặn sự phù hợp quá mức (over fiting), giảm số lượng các thông số cần thiết trong đào tạo và cải thiện độ chính xác.

Đồng thời, các thử nghiệm của bài toán dựa trên CNN và CNN + Word2vec được thực hiện riêng. Hai đặc điểm quan trọng của CNN là ý nghĩa cục bộ (local sensing) và chia sẻ trọng số (weight sharing). Thuật toán CNN khi được áp dụng vào bài toán này thì coi đoạn văn bản đầu vào là một ma trận 2 chiều có kích thước 400x300. Độ dài cố định của mỗi văn bản là 400 và kích thước word embedding của từng từ là 300.

Bảng 3. *So sánh các thuật toán khác.*

	Precision	Recall	F1-score
CNN	83.48%	83.02%	83,25%
CNN+ Word2Vec	84,13%	84,89%	84,01%

Kết quả trên cho thấy rằng, độ chính xác của mô hình phân loại văn bản dựa trên LSTM + Word2vec cho kết quả tốt hơn nhiều so với thuật toán CNN. CNN có những lợi thế lớn trong việc xử lý hình ảnh, tuy nhiên, đối với phân loại văn bản thì mô hình này cho kết quả không cao. Có thể thấy rằng, LSTM cho kết quả tốt trong việc xử lý các dữ liệu là dạng text. Thuật toán LSTM là phương pháp xử lý dạng chuỗi, trong quá trình học dữ liệu nó sẽ ghi nhớ dữ liệu trong quá khứ và xử lý dựa vào cả dữ liệu hiện tại và quá khứ nên có sự liên kết giữa các từ nên đạt được độ chính xác cao hơn so với CNN.

Bảng 4. *So sánh các phương pháp phân loại văn bản khác đã thực nghiệm.*

Tác giả	Phương pháp	Dữ liệu	Số đặc trưng	Độ chính xác (%)
BùiKhánh Linh, Nguyễn Thị Thu Hà, Nguyễn Thị Ngọc Tú, Đào Thanh Tĩnh [17]	Mô hình chủ đề	VLSP (20000 câu)	829; 339	83; 94.07
Nguyễn Linh Giang, Nguyễn Mạnh Hiển [18]	Support Vector Machine (SVM)	[6] (4162 tài liệu)	7721	80,72
Phu X.V. Nguyen, Tham T.T Hong, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen [20]	Naive Bayes, Maxent, LSTM, Bi-LSTM	Vietnamese Students' Feedback Corpus for Sentiment Analysis (16000 phản hồi) [21]	300	81,2 – 89,6
V. C. D. Hoang, D. Dinh, N. le Nguyen and H. Q. Ngo [1]	SVM kNN NGram	VNTC [1]	N/A	93,4% 84,67% 97,1%
Toan Pham Van, Ta Minh Thanh [22]	SVM Random Forest SVC Neural Network	VNTC [1]	N/A	96,52% 99,21% 99,22% 99,75%
Hung T.Vo, Duc Dung Nguyen [26]	Naive Bayes kNN SVM	N/A	Bag – of – structure (BoS)	72,6% 83,66% 87,95%
Huy – Thanh Duong, Vinh Truong Hoang [27]	Linear SVM	VNNews-01	Tf-Idf	92,3%
Huy – Tien Nguyen, Bac Le [28]	LSTM + CNN	VS	200	87,5%

Từ bảng trên, mặc dù mỗi nghiên cứu thực hiện trên những bộ dữ liệu khác nhau nhưng nhìn chung khi sử dụng phương pháp đề xuất là sử dụng mô hình LSTM + Word2vec đã cải thiện được kết quả so với các phương pháp truyền thống như Naive Bayes, kNN,... So với việc sử dụng các mạng LSTM, Bi-LSTM,... chúng tôi kết hợp với phương pháp Word2vec cũng đã giúp cải thiện kết quả và thời gian thực hiện. Kết quả của chúng tôi cho thấy sự vượt trội so với các phương pháp tương tự như trong [20] tác giả cũng sử dụng LSTM, bộ dữ liệu là 16000 phản hồi và phân loại với 4 nhãn còn đối với nghiên cứu chúng tôi có 10 nhãn và kết quả cải thiện 6% về độ chính xác. Với những nghiên cứu sử dụng cùng bộ dữ liệu với chúng tôi, phương pháp của chúng tôi cũng đã đạt được kết quả tốt hơn hoặc tương đương với các phương pháp truyền thống đã đạt được nhiều kết quả tốt trong xử lý ngôn ngữ tự nhiên là SVM như trong [18, 1, 26, 27]. Việc kết hợp giữa LSTM và phương pháp học sâu khác như CNN trong [28] mặc dù chỉ phân loại 3 nhãn nhưng đạt kết quả vẫn còn hạn chế so với phương pháp của chúng tôi. Trong [22], nghiên cứu đã đạt được kết quả rất tốt trên tập

dữ liệu VNTC, trong đề xuất đã sử dụng các phương pháp vector hóa các từ và phân lớp văn bản khác với chúng tôi. Với kiến trúc mạng mới kết hợp với Bag of Word và Keyword Extraction đã đạt tới 99.75% độ chính xác, tốt hơn so với phương pháp đề xuất của chúng tôi. Qua đây có thể thấy, phương pháp đề xuất của chúng tôi vẫn còn nhiều hạn chế và chúng tôi sẽ tiếp tục cải thiện trong những nghiên cứu tiếp theo.

4. KẾT LUẬN

Việc phân loại văn bản là công việc cơ bản và cũng có vai trò quan trọng trong việc nghiên cứu, khai thác lượng dữ liệu lớn hiện nay. Trong bài viết này, chúng tôi đã trình bày phương pháp phân loại văn bản dựa trên mô hình LSTM. Thuật toán LSTM khắc phục được vấn đề đạo hàm biến mất (vanishing gradient), 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. Đóng góp của chúng tôi là đã đề xuất sử dụng kết hợp mô hình Word2vec giúp giảm số chiều của không gian đặc trưng để làm giảm độ phức tạp của phương pháp. Việc học lại đặc trưng bằng phương pháp Word2vec trên bộ dữ liệu VNTC giúp liên kết đặc trưng phù hợp, liên quan và tương tự nhau giúp nâng cao hiệu quả của mô hình. Với phương pháp Word2vec, chúng tôi sử dụng số chiều là 300, giảm số đặc trưng so với các phương pháp khác giúp tăng tốc độ xử lý, tránh hiện tượng sự bùng nổ tổ hợp (curse of dimensionality). Chúng tôi đã kiểm chứng mô hình này so với việc không sử dụng Word2vec cho thấy khi sử dụng kết hợp với Word2vec kết quả cho tốt hơn. Do việc trích chọn đặc trưng từ Word2vec đã giúp mô hình chọn ra những đặc trưng phù hợp và giúp mô hình cải thiện độ chính xác. Thực nghiệm đã so sánh kết quả phân loại văn bản sử dụng LSTM với CNN, qua đó cho thấy rằng LSTM thực sự tốt hơn CNN. Mô hình của chúng tôi cũng đã cải thiện kết quả so với những phương pháp học truyền thống khác.

Tuy nhiên, khi sử dụng mô hình này, chúng tôi thấy một số nhược điểm của mô hình này như sau:

- Thời gian train mô hình lâu hơn so với các phương pháp như SVM, CNN.

- Mặc dù được cải thiện về vấn đề đạo hàm biến mất của RNN nhưng đối với một số câu văn quá dài thì phương pháp vẫn xuất hiện hiện tượng này, điều này gây ra những trường hợp sai sót trong việc phân loại.

- Hiệu quả không tốt trong thực tế.

Công việc tiếp theo sẽ thực hiện là cải thiện mô hình phân loại bằng việc sử dụng kết hợp với các kỹ thuật khác để cải thiện hiện tượng biến mất đạo hàm như attention và cải thiện độ chính xác bằng việc sử dụng kết hợp với các mô hình học sâu khác.

TÀI LIỆU THAM KHẢO

[1]. V. C. D. Hoang, D. Dinh, N. le Nguyen and H. Q. Ngo, *"A Comparative Study on Vietnamese Text Classification Methods,"* IEEE International Conference on Research, Innovation and Vision for the Future, Hanoi, 2007, pp. 267-273.

[2]. T.V. Trung, Python Vietnamese Core NLP Toolkit (2019), Online: https://github.com/trungtv/pyvi

[3]. Stopwords/Vietnamese stopwords (2019), Online: https://github.com/stopwords/vietnamese-stopwords

[4]. Ting K.M., *"Confusion Matrix,"* In Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA, 2017.

[5]. Detailed explanation of Word2Vector Skip-gram (2013), Online: http://www.programmersought.com/article/8383114826/

[6]. Vnexpress, Online: https://vnexpress.net/

[7]. Tuoitre, Online: https://tuoitre.vn/
[8]. Thanhnien, Online: https://thanhnien.vn/
[9]. Người lao động, Online: https://nld.com.vn/
[10]. Hông P. L., Huyen T. M. N., Roussanaly A., Vinh H. T., *"A Hybrid Approach to Word Segmentation of Vietnamese Texts. In: Martín-Vide C.,"* Otto F., Fernau H. (eds) Language and Automata Theory and Applications. LATA 2008, pp 240-249.
[11]. Thomas F. and Christopher K., *"Deep learning with long short-term memory networks for financial market predictions,"* European Journal of Operational Research, vol. 270, no. 2, pp. 654-669, 2018.
[12]. Thanh V.Nguyen, Hoang K.Tran, Thanh T.T.Nguyen, Hung Nguyen. *"Word Segmentation for Vietnamese Text Categorization: An online corpus approach".*
[13]. Tạ Nguyễn, Vũ Đức Lung, *"Xây dựng hệ thống rút trích các nội dung chính của văn bản khoa học tiếng Việt dựa trên cấu trúc"*, (2014).
[14]. Lizhong Xiao, Guanghong Wang, Yang Zuo, *"Research on patent text classification based on Word2vec and LSTM"*, (2018).
[15]. Raouf Ganda, *"Efficient Deep Learning Model for Text Classification Based on Recurrent and Convolutional Layers"*, (2017).
[16]. Anurag Sarkar, Saptarshi Chatterjee, Writayan Das, Debabrata Datta. *"Text Classification using Support Vector Machine"*.
[17]. Bùi Khánh Linh, Nguyễn Thị Thu Hà, Nguyễn Thị Ngọc Tú, Đào Thanh Tĩnh. *"Phân loại văn bản tiếng Việt dựa trên mô hình chủ đề."*
[18]. Nguyễn Linh Giang, Nguyễn Mạnh Hiển, *"Phân loại văn bản tiếng Việt với bộ phân loại vectơ hỗ trợ SVM"*, (2005).
[19]. Radhika Kuttala, Bindu K R, Latha Parameswaran, *"A text classification model using convolution neural network and recurrent neural network"*, (2018).
[20]. Phu X.V. Nguyen, Tham T.T Hong, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen. *"Deep Learning versus Traditional Classifiers on Vietnamese Students' Feedback Corpus"*, (2018).
[21]. Kiet Van Nguyen, Vu Duc Nguyen, Phu Xuan-Vinh Nguyen, Tham Thi-Hong Truong, Ngan Luu-Thuy Nguyen, UIT-VSFC: *"Vietnamese Students' Feedback Corpus for Sentiment Analysis"*, 2018 10th International Conference on Knowledge and Systems Engineering (KSE 2018), November 1-3, 2018, Ho Chi Minh City, Vietnam.
[22]. Toan Pham Van, Ta Minh Thanh, *"Vietnamese News Classification based on BoW with Keywords Extraction and Neural Network"*, (2017).
[23]. Dat Quoc Nguyen, Dai Quoc Nguyen, Thanh Vu, Mark Dras, Mark Johnson, *"A Fast and Accurate Vietnamese Word Segmenter"*, (2018)
[24]. Dat Quoc Nguyen, Karin Verspoor, *"An improved neural network model for joint POS tagging and dependency parsing"*, (2018)
[25]. Le Hong Phuong, Huyen Thi Minh Nguyen, Azim Roussanaly, Tuong Vinh Ho, *"A Hyvird Approach to word Segmentation of Vietnamese Texts."*
[26]. Hung T.Vo, Hai C.Lam, Duc Dung Nguyen, Nguyen Huynh Tuong. *"Topic classification and sentiment analysis for Vietnamese education survey system."* (2016)
[27]. Huy – Thanh Duong, Vinh Truong Hoang. *"A survey on the multiple classifier for new benchmark dataset of Vietnamese news classification."* (2019)
[28]. Huy – Tien Nguyen, Bac Le. *"Multi – channel LSTM – CNN model for Vietnamese sentiment analysis."* (2017)

ABSTRACT

PROPOSAL CLASSIFICATION ALGORITHM OF VIETNAMESE TEXT USING LONG SHORT TERM MEMORY AND WORD2VECABSTRACT

Recently, text classification is considered as a fundamental approach in Natural Language Processing (NLP). It can be widely applied into numerous fields namely sentiment analyses, topic labelings and so on. Specifically, recent achievements have shown that Deep Learning (DL) methods obtained great performance in classifying texts. These methods have positive effects on text classification, especially in English. However, there are few studies investigating about their impacts on Vietnamese text classification. Therefore, in this research, Long Short Term Memory (LSTM) network and Word2Vec engine were used in text classification with the aim of improving efficiency and accuracy. The results of model evaluation on Vietnamese text VNTC [1] we concluded were feasible and likely to be applied in real-life contexts in the near future.

Keywords: Text Classification; Natural Language Processing; Long Short Term Memory; Word2vec; Data Processing.

Nhận bài ngày 23 tháng 4 năm 2020
Hoàn thiện ngày 13 tháng 6 năm 2020
Chấp nhận đăng ngày 15 tháng 10 năm 2020

Địa chỉ: Viện Điện tử Viễn thông, Đại học Bách khoa Hà Nội.
*Email: phat.nguyenhuu@hust.edu.vn.