PHÂN LOẠI VĂN BẢN TIẾNG VIỆT DỰA TRÊN HỌC SÂU

Hoàng Thị Hảo

1. Introduction

Chữ viết là phương tiện để con người lưu trữ tri thức. Xã hội càng phát triển, lượng tri thức cần lưu trữ càng nhiều, kèm theo đó là hàng triệu văn bản với các thể loại đa dạng khác nhau. Khả năng để một người tìm kiếm, phân loại số lượng văn bản đồ sộ đó là không thể. Sự ra đời của trí tuệ nhân tạo hay nói cách khác là các thuật toán thông minh đã giúp ích rất nhiều trong công việc này.

Đầu tiên là thời kỳ của các giải thuật học máy. Từ đầu vào là một danh sách các từ có trong văn bản, một giải thuật học máy đơn giản như Naive Bayes đã có thể dựa vào các từ quan trọng, xuất hiện nhiều trong từng thể loại để đưa ra kết luận một văn bản thuộc về thể loại nào. Tiếp đến là thời kỳ học sâu, với những mạng nơron phức tạp, mô phỏng theo bộ não của con người. Một mạng RNN có thể học được những ngữ nghĩa có trong một văn bản, ý nghĩa mà văn bản muốn truyền tải để từ đó đưa ra thể loại của văn bản. Sẽ thật lý tưởng khi có thể kết hợp điểm mạnh của mỗi mô hình lại với nhau. Trong báo cáo này, tôi thí nghiệm một mô hình mạng nơron Bi-LSTM cùng cơ chế attention nhằm học các mối quan hệ chuỗi của các từ trong văn bản, trọng số của mỗi từ xuất hiện trong văn bản, xem từ nào là từ khóa gắn với chủ đề, từ nào chỉ mang tính chất liên kết mạch văn.

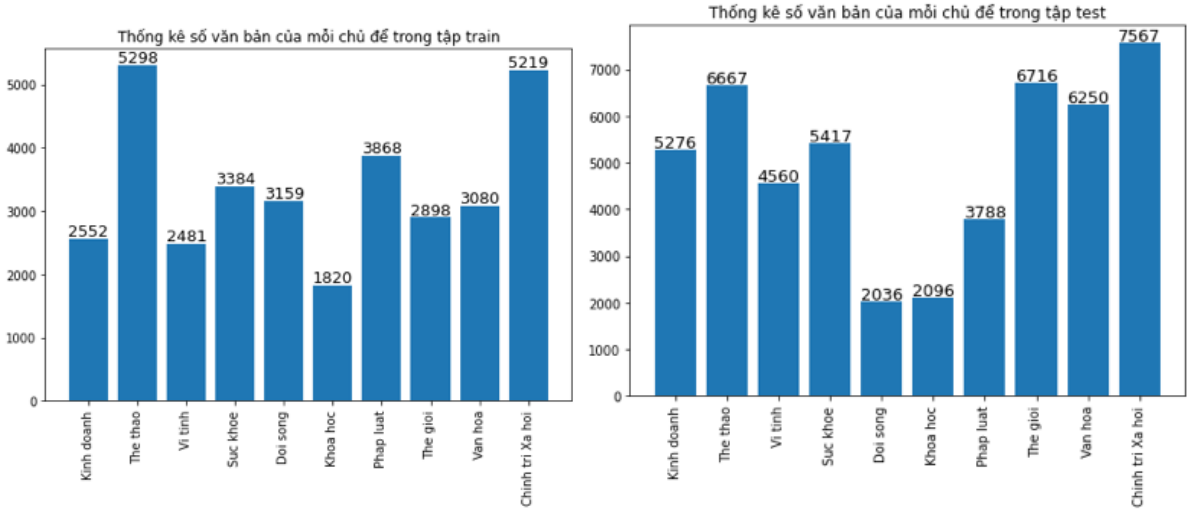
2. Relate work

Các công trình liên quan có thể được chia làm hai hướng: sử dụng học máy và sử dụng học sâu. Trong học máy, Aili Zhang và các cộng sự [2] sử dụng thuật toán máy vectơ hỗ trợ (SVM) để phân loại văn bản nhiều lớp. Phương pháp này chủ yếu sử dụng mô hình không gian vectơ như một đối tượng địa lý, biến văn bản thành một vectơ thưa thớt có kích thước lớn theo các đặc điểm của văn bản và sau đó đưa nó vào bộ phân loại SVM. Liu Hua [3] sử dụng các cụm từ khóa của văn bản để phân loại và cho rằng các từ hoặc cụm từ chính của thông tin danh mục văn bản phản hồi quan trọng hơn, vì vậy các đặc trưng vectơ của các cụm từ khóa trước tiên được trích xuất bằng phương pháp thống kê, và sau đó tính tương tự cosine để xác định phạm trù. Với sự gia tăng của các phương pháp học sâu trong những năm gần đây, K. Kowsari và cộng sự [4] đã giới thiệu một kỹ thuật học sâu mới để phân loại được gọi là Học sâu đa mẫu ngẫu nhiên (RMDL) tự động tìm hiểu các đặc điểm phân loại của tài liệu, và việc phân loại tài liệu hiện tại đã đạt được kết quả tốt. Mô hình có thể được sử dụng cho bất kỳ nhiệm vụ phân loại nào. Tuy nhiên, mỗi mô hình ở trên lại có điểm yếu, điểm mạnh riêng. Trong tài liệu [2] đã chỉ ra rằng các thể loại của văn bản thường gắn với một số cụm từ và từ chính nên chúng được mô hình hóa bằng phương pháp trích từ khóa. Những từ khóa này rất quan trọng, nhưng những từ khác liên kết các từ khóa này với nhau cũng chứa nhiều thông tin về tài liệu và việc bỏ trực tiếp những từ này có thể làm hỏng thông tin mà tài liệu đại diện một cách nghiêm trọng. Trong [4], mạng nơ-ron được sử dụng để nghiên cứu tài liệu, có tính đến các mối tương quan

và trình tự của các từ, có khả năng trích xuất các đặc trưng của văn bản một cách tự động và có hiệu suất mạnh nhất trên các tập dữ liệu cổ điển hiện nay. Tuy nhiên, toàn bộ mô hình không tính đến vai trò của các từ khóa, mà coi tất cả các từ như một mạng lưới đầu vào, không đưa ra bất kỳ sự đối xử đặc biệt nào đối với các từ khóa.

3. Dataset and Features

Tập dữ liệu được sử dùng trong báo cáo này là tập dữ liệu tiếng Việt gồm 10 chủ đề: "Chính trị Xã hội", "Văn hóa", "Thế giới", "Pháp luật", "Khoa học", "Đời sống", "Sức khỏe", "Vi tính", "Thể thao", "Kinh doanh". Phân bố số lượng văn bản thuộc từng chủ đề trong hai tập train và test như sau:



Đơn vị từ trong tiếng Việt bao gồm từ đơn và từ ghép. Nên chúng ta cần phải xác định từ nào là từ đơn, từ nào là từ ghép trước khi đưa vào mô hình. Bởi vì mô hình của chúng ta sẽ coi các từ là đặc trưng, tách nhau theo dấu cách. Do đó, chúng ta phải nối các từ ghép lại thành một từ để không bị tách sai. Ví dụ: Học sinh học sinh học ⇒ Học_sinh học sinh_học. Bài toán này là một bài toán cơ sở trong NLP – bài toán tách từ (word tokenize). Thật may là hiện nay có khá nhiều thư viện mã nguồn mở của bài toán này. Trong bài báo này, chúng ta sẽ sử dụng công cụ Vitk -- A Vietnamese Text Processing Toolkit của tác giả Lê Hồng Phương. Sau khi đã tách từ, mỗi văn bản sẽ được loại bỏ các stopwords, ký tự đặc biệt, dấu câu và cuối cùng là đưa về viết thường.

Khi đã tiền xử lý xong, mỗi văn bản sẽ được biểu diễn theo các từ có trong nó, tức là một ma trận kích thước MxN. Với M là số từ có trong văn bản và N là kích thước vector nhúng của từ. Vector nhúng của từ được lấy từ mô hình pre-train word embedding Word2Vec.

4. Methods

4.1. Tầng đầu vào của mạng

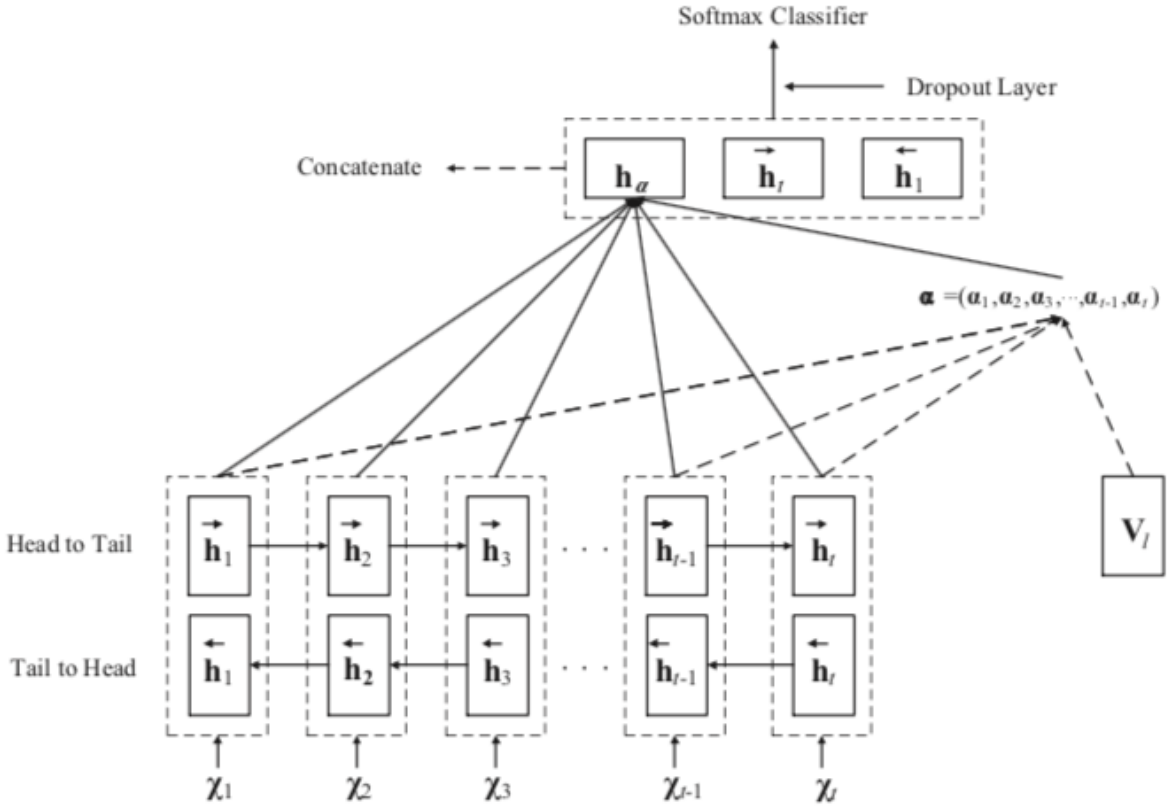
Ý nghĩa của một từ không thể chỉ được quyết định bởi chính nó mà còn do ngữ cảnh mà có từ đó. Tức cùng một từ với ngữ cảnh khác nhau thì từ đó sẽ mang ý nghĩa khác nhau.Vì vậy, thay vì cho trực tiếp vector pre-train của từ làm đầu vào của mạng thì vector của từ sẽ được tính bằng trung bình cộng vector của từ đứng liền trước từ đó, vector của từ đó và vectơ của từ đứng liền sau từ đó.

Ví dụ: “Hôm nay trời đẹp”

Vector’(nay) = (Vector(Hôm) + Vector(nay) + Vector(trời))/ 3

Với những từ đứng đầu hoặc cuối văn bản, dùng thêm ký hiệu </s> để tính và vector của </s> là vector không.

4.2. Mô hình Bi-LSTM cùng self-attention



Ý nghĩa của một từ một phần phụ thuộc vào ngữ cảnh chứa nó, nên mô hình sẽ sử dụng mạng Long short-term memory hai chiều (Bi-LSTM) để học các đặc trưng của văn bản để có thể học ý nghĩa của một từ mà có tính cả đến những những từ đứng trước nó và cả từ đứng sau nó. Sau đấy thêm cơ chế attention để học được trọng số của từng từ, sao cho từ quan trọng hơn có trọng số cao hơn, từ ít quan trọng có trọng số thấp hơn. Từ đấy tổng hợp được vector của văn bản mà mang được các đặc trưng quan trọng.

Trong hình trên, chuỗi đầu vào x_1, x_2, x_3,…,x_t được đưa trực tiếp vào tầng Bi-LSTM. Với chiều tiến, ta sẽ có các trạng thái ẩn $\overrightarrow{h_1}$, $\overrightarrow{h_2}$, $\overrightarrow{h_3}$,…, $\overrightarrow{h_t}$, và chiều tiến có các trạng thái ẩn là $\overleftarrow{h_t}$, $\overleftarrow{h_{t-1}}$, $\overleftarrow{h_{t-2}}$,…, $\overleftarrow{h_1}$. Trạng thái ẩn của từ x_i là h_i = [$\overrightarrow{h_i}$, $\overleftarrow{h_i}$] (i = 1,…t) – một vector 2k chiều được tổng hợp từ hai vector trạng thái ẩn k chiều của hai mạng LSTM.

Tiếp đến sử dụng cơ chế attention để tính trọng số cho từng từ. V_l là vector biểu diễn cho chủ đề, α_i (i = 1,…t) biểu diễn độ tương tự giữa trạng thái ẩn của từ i và vector V_l và cũng là trọng số của từ i, được tính theo phương trình:

$$\alpha_i = \frac{e^{h_i^T M V_l}}{\sum_{j=1}^{t} e^{h_j^T M V_l}}$$

Với $i = 1,\ldots t$. M là ma trận tham số để tính sự tương tự giữa h_i và V_l.
Đặt $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_t)$ là vector trọng số của các từ. Khi đó ta tính được vector đặc trưng có trọng số h_α bằng:

$$h_\alpha = \sum_{i=1}^{t} \alpha_i \, h_i$$

h_α biểu diễn các thông tin có trọng số của văn bản, những đặc trưng quan trọng của văn bản sẽ có trọng số cao và ngược lại. Kết hợp h_α và hai trạng thái ẩn cuối cùng $\overrightarrow{h_t}$, $\overleftarrow{h_1}$ ta được vector đặc trưng cuối cùng của văn bản:

$$s = [h_\alpha, \overrightarrow{h_t}, \overleftarrow{h_1}]$$

Cuối cùng cho s làm đầu vào một tầng kết nối đầy đủ có sử dụng dropout (nhằm tránh việc overfitting) có số đầu ra bằng số thể loại.

$$o = f(W_c sp + b_c)$$

Với p là tỷ lệ đầu vào bị loại bỏ, W_c, b_c lần lượt là ma trận tham số và bias của lớp này. f là hàm kích hoạt Relu. Cuối cùng, áp dụng softmax cho đầu ra cuối để dự đoán thể loại của văn bản. Xác xuất để văn bản thuộc vào lớp j là:

$$p(j|doc) = \frac{e^{o_j}}{\sum_{i=1}^{N} e^{o_i}}$$

Hàm mục tiêu của mô hình là cross-entropy:

$$L = \sum_{i=1}^{T} \log p(y_i | X_i, \theta) + \lambda \|\theta\|_2^2$$

Với (X_i, y_i) là các cặp ví dụ học, T là số ví dụ học, θ là tất cả các tham số của mô hình, λ là tham số của thành phần regularization. Các trọng số được cập nhật bằng giải thuật Adam.

5. Experiments

Thí nghiệm được tiến hành với bộ dữ liệu đã được miêu tả ở mục 3. Các tham số của mô hình được cài đặt như sau: số chiều vector nhúng của từ d = 300; số chiều của trạng thái ẩn trong mạng LSTM n = 500; tỷ lệ dropout p = 0.6; tốc độ học của thuật toán Adam α = 0.001; kích thước vector thể loại l = 100. Do giới hạn về phần cứng, các văn bản chỉ có thể được đưa vào mô hình với độ dài là 500 từ. Vì vậy, những văn bản có chứa nhiều hơn 500 từ sẽ bị cắt bớt, và những văn bản ít hơn 500 từ thì sẽ được thêm ký tự "</s>" để lấp đầy khoảng trống.
Để mang lại tính khách quan, ta sẽ so sánh mô hình được đề xuất trong báo cáo này với hai mô hình nữa là Naive Bayes và Bi-LSTM. Mô hình Naive Bayes có hệ số smooth α = 1. Kết quả của ba mô hình có trong bảng sau:

Model	Precision	Recall	F1-Score	Accuracy
Naive Bayes	0.88	0.88	0.88	0.89
Bi-LSTM	0.88	0.86	0.87	0.89
Bi-LSTM&Self-Attention	0.89	0.88	0.88	0.91

Từ kết quả trên ta thấy rằng, mô hình được đề xuất vẫn chưa thực sự hoạt động như mong đợi. Độ chính xác của Bi-LSTM&Self-Attention chỉ hơn hai mô hình kia 0.02%. Nguyên nhân có thể là do kích thước đầu vào đã bị cắt bớt trước khi được đưa vào mạng nên những từ có khả năng là từ khóa của chủ đề lại không được mô hình biết đến.

6. Conclusion

Trong báo cáo này, tôi đã thí nghiệm một mô hình phân loại văn bản dựa trên học sâu kết hợp với cơ chế attention. Kết quả có một chút cải thiện so với các mô hình học máy truyền thống nhưng chưa thực sự đáng kể. Vẫn còn rất nhiều công việc có thể thực hiện để cải tiến mô hình này trong tương lai, chẳng hạn: tăng thêm phần cứng để có thể đưa toàn bộ văn bản vào mạng, sử dụng thêm mạng CNN để trích xuất đặc trưng của văn bản thay vì sử dụng pre-train word embedding.

7. References

[1] Du, Changshun, and Lei Huang. "Text classification research with attention-based recurrent neural networks." *International Journal of Computers Communications & Control* 13.1 (2018): 50-61

[2] Zhang, A.-L., Liu, G.-L., Liu C.-Y. (2004); Research on multiple classes text categorization based o SVM, Journal of Information, 9, 6–10, 2004.

[3] Hua, L. (2007); Text Categorization Base on Key Phrases, Journal of Chinese Information Processing, 21(4), 34–41, 2007.

[4] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, "Random Multimodel Deep Learning for Classification", arXiv, April, 2018.

[5] https://nguyenvanhieu.vn/phan-loai-van-ban-tieng-viet/

[6] https://nttuan8.com/bai-14-long-short-term-memory-lstm/

[7] https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c

[8] https://towardsdatascience.com/using-deep-learning-for-end-to-end-multiclass-text-classification-39b46aecac81

[9] https://machinelearningmastery.com/best-practices-document-classification-deep-learning/