**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

*Người thực hiện*: **NGUYỄN QUANG BẢO – MSHV:20083601**

**NGUYỄN XUÂN GIANG - MSSV: 20079601**

Lớp **: 420300138501**

Khoá  **: 16**

*Người hướng dẫn*: **TS BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

Người thực hiện: **NGUYỄN QUANG BẢO – MSHV:20083601**

**NGUYỄN XUÂN GIANG - MSSV: 20079601**

Lớp **: 420300138501**

Khoá  **: 16**

Người hướng dẫn: **TS. BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Tôi xin được gửi lời tri ân đến các thầy cô và nhà trường đã tạo điều kiện thuận lợi cho tôi trong quá trình học tập và thực hiện đề tài này. Tôi cũng xin bày tỏ lòng biết ơn đến bạn bè đã luôn động viên và chia sẻ kinh nghiệm với tôi.

Tôi xin chân thành cảm ơn Thầy TS. Bùi Thanh Hùng - người đã truyền đạt cho tôi những kiến thức bổ ích và hướng dẫn tận tình trong suốt quá trình nghiên cứu đề tài. Nhờ có sự chỉ bảo của Thầy, tôi mới có thể hoàn thành được bài tiểu luận này một cách tốt nhất.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS. Bùi Thanh Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Công nghiệp TP Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Quang Bảo*

*Nguyễn Xuân Giang*

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Bài toán phân lớp văn bản tiếng Việt là bài toán gán nhãn cho một đoạn văn bản tiếng Việt theo một số chủ đề xác định trước, ví dụ như thể thao, giáo dục, kinh tế, văn hóa, chính trị, khoa học, sức khỏe, giải trí, v.v. Bài toán này có ứng dụng rộng rãi trong các lĩnh vực như phân tích cảm xúc, phát hiện tin giả, tóm tắt nội dung, khai thác thông tin, v.v.Trong đề tài này, chúng tôi sử dụng hướng tiếp cận dựa trên học sâu để phân lớp văn bản tiếng Việt. Kết quả đạt được cho thấy hướng tiếp cận dựa trên học sâu là một phương pháp hiệu quả để phân lớp văn bản tiếng Việt.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc153182263)

[PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc153182264)

[TÓM TẮT iv](#_Toc153182265)

[MỤC LỤC 1](#_Toc153182266)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 2](#_Toc153182267)

[DANH MỤC CÁC HÌNH VẼ 3](#_Toc153182268)

[1.1 Giới thiệu về bài toán 5](#_Toc153182269)

[1.2 Phân tích yêu cầu của bài toán 6](#_Toc153182270)

[1.2.1 Yêu cầu của bài toán 6](#_Toc153182271)

[1.2.2 Các phương pháp giải quyết bài toán 6](#_Toc153182272)

[1.2.3 Phương pháp đề xuất giải quyết bài toán 8](#_Toc153182273)

[1.3 Phương pháp giải quyết bài toán 15](#_Toc153182274)

[1.3.1. Mô hình tổng quát 15](#_Toc153182275)

[1.3.2 Đặc trưng của mô hình đề xuất 16](#_Toc153182276)

[1.3.2.1. Dữ liệu 16](#_Toc153182277)

[1.3.2.2 Xử lý dữ liệu 16](#_Toc153182278)

[1.3.2.3 Phương pháp huấn luyện 20](#_Toc153182279)

[1.4 Thực nghiệm 23](#_Toc153182280)

[1.4.1 Dữ liệu 23](#_Toc153182281)

[1.4.2 Xử lý dữ liệu 24](#_Toc153182282)

[1.4.3 Công nghệ sử dụng 28](#_Toc153182283)

[1.4.4 Cách đánh giá 28](#_Toc153182284)

[1.5 Kết quả đạt được 28](#_Toc153182285)

[1.6 Kết luận 32](#_Toc153182286)

[1.6.1 Kết luận 32](#_Toc153182287)

[1.6.2 Hướng phát triển 32](#_Toc153182288)

[TÀI LIỆU THAM KHẢO 33](#_Toc153182289)

[LÀM VIỆC NHÓM 36](#_Toc153182290)

[TỰ ĐÁNH GIÁ 37](#_Toc153182291)

# DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

LSTM Long Short-Term Memory

RNN Recurrent Neural Network

CNN Convolutional Neural Network

BERT Bidirectional Encoder Representations from Transformers

DANH MỤC CÁC HÌNH VẼ

[**Hình 1:** Kiến trúc LSTM 8](#_Toc153182868)

[**Hình 2:** Mô hình tổng quát 15](file:///C:\Users\hungp\OneDrive%20-%20Industrial%20University%20of%20HoChiMinh%20City\Desktop\NLP.docx#_Toc153182869)

[**Hình 3:** So sánh accuracy của hai mô hình học sâu PhoBert và LSTM 29](#_Toc153182870)

[**Hình 4:** So sánh loss của hai mô hình học sâuPhoBert và LSTM 29](#_Toc153182871)

[**Hình 5:** So sánh accuracy trên từng epochs 30](#_Toc153182872)

[**Hình 6:** So sánh accuracy của từng mô hình 31](#_Toc153182873)

[**Hình 7:** So sánh loss của từng mô hình 32](#_Toc153182874)

DANH MỤC CÁC BẢNG

[**Bảng 1:** Công nghệ sử dụng 27](#_Toc153182947)

**Phân lớp văn bản tiếng Việt bằng mô hình PHO-BERT**

* 1. Giới thiệu về bài toán

Bài toán Phân lớp văn bản tiếng Việt bằng mô hình PHO-BERT [1] và Phân lớp văn bản tiếng Việt bằng phương pháp học sâu là bài toán gán nhãn cho một đoạn văn bản tiếng Việt theo một số chủ đề xác định trước, ví dụ như thể thao, giáo dục, kinh tế, văn hóa, chính trị, khoa học, sức khỏe, giải trí, v.v. Bài toán này có ứng dụng rộng rãi trong các lĩnh vực như phân tích cảm xúc, phát hiện tin giả, tóm tắt nội dung, khai thác thông tin, v.v.

Mô hình PHO-BERT là một phiên bản của mô hình BERT [2] được tiền huấn luyện trên dữ liệu tiếng Việt. Mô hình BERT là một mô hình học sâu dựa trên kiến trúc Transformer, có khả năng mã hóa một đoạn văn bản thành một vector đặc trưng có khả năng biểu diễn được ngữ nghĩa và ngữ cảnh của văn bản. Mô hình PHO-BERT kế thừa các ưu điểm của mô hình BERT và được tinh chỉnh để phù hợp với đặc trưng của tiếng Việt, ví dụ như phân biệt được các từ đồng âm, đồng nghĩa, đồng hình, v.v., cũng như bắt được các từ ghép, từ láy, từ mượn, v.v., trong tiếng Việt. Bằng cách sử dụng mô hình PHO-BERT, ta có thể học được các đặc trưng phong phú, phức tạp và bắt được ngữ cảnh của văn bản tiếng Việt, từ đó cải thiện hiệu năng của bài toán phân lớp văn bản.

Phương pháp học sâu là một nhóm các phương pháp học máy dựa trên các mạng nơ-ron nhân tạo có nhiều lớp ẩn. Các phương pháp học sâu có thể học trực tiếp từ dữ liệu văn bản và phân lớp văn bản mà không cần phải trích xuất đặc trưng thủ công. Các phương pháp học sâu thường sử dụng các mô hình như mạng nơ-ron tích chập (CNN), mạng nơ-ron hồi quy (RNN), mạng nơ-ron biểu diễn từ (BERT), v.v., để học và phân lớp văn bản. Các phương pháp học sâu có ưu điểm là có thể xử lý được các văn bản dài và đa dạng, cũng như có thể áp dụng cho các bài toán khác liên quan đến xử lý ngôn ngữ tự nhiên.

Ý nghĩa của bài toán:

Bài toán phân lớp văn bản tiếng Việt có ý nghĩa là giúp phân loại các văn bản tiếng Việt theo các chủ đề khác nhau, từ đó hỗ trợ cho các công việc liên quan đến tìm kiếm, chiết lọc, tổng hợp thông tin. Bài toán này cũng góp phần nâng cao khả năng xử lý ngôn ngữ tự nhiên tiếng Việt, một ngôn ngữ có nhiều đặc trưng riêng biệt và phong phú. Bài toán này có nhiều ứng dụng trong các lĩnh vực như phân tích cảm xúc, phát hiện tin giả, tóm tắt nội dung, khai thác thông tin, v.v. Bài toán này cũng có thể áp dụng cho các văn bản tiếng Việt về các chuyên ngành khác nhau, ví dụ như tài nguyên và môi trường, khoa học và công nghệ, giáo dục và đào tạo, v.v.

1.2 Phân tích yêu cầu của bài toán

1.2.1 Yêu cầu của bài toán

Bài toán này yêu cầu sử dụng mô hình PhoBert và một hoặc nhiều các phương pháp học sâu để phân loại văn bản. Dữ liệu có thể tham khảo trên các nguồn hoặc có thể tự crawl dữ liệu.

1.2.2 Các phương pháp giải quyết bài toán

Có nhiều phương pháp khác nhau để giải quyết bài toán phân lớp văn bản tiếng Việt, tùy thuộc vào cách trích xuất đặc trưng và cách học từ dữ liệu văn bản. Dưới đây là một số phương pháp mà chúng tôi tham khảo:

Phương pháp dựa trên đặc trưng thủ công (hand-crafted features) và học máy truyền thống (traditional machine learning) :

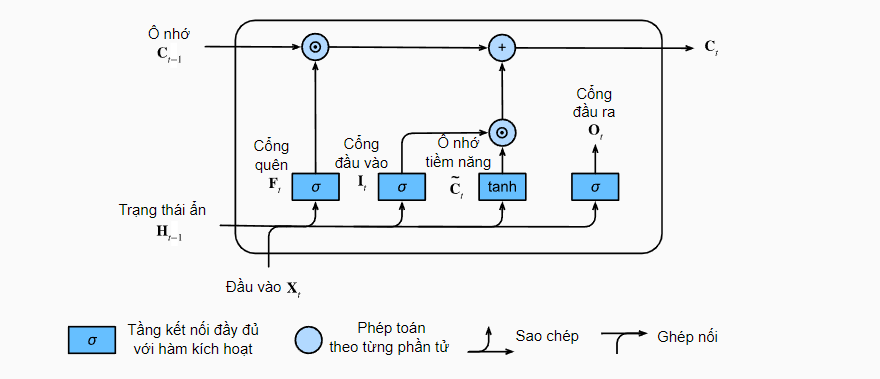
Phương pháp này sử dụng các kỹ thuật trích xuất đặc trưng từ văn bản, ví dụ như TF-IDF [3], n-gram [4], word2vec [5], v.v., và áp dụng các thuật toán học máy truyền thống, ví dụ như SVM, Naive Bayes, Decision Tree, v.v., để phân lớp văn bản. Phương pháp này có ưu điểm là đơn giản, dễ hiểu và có thể đạt kết quả khá tốt trên các bộ dữ liệu nhỏ và ít nhiễu. Tuy nhiên, phương pháp này cũng có nhược điểm là phụ thuộc nhiều vào chất lượng và độ đa dạng của các đặc trưng, cũng như khó khăn trong việc xử lý các văn bản dài, phức tạp và có nhiều ngữ cảnh. Một ví dụ của phương pháp này là bài báo “PHÂN LOẠI VĂN BẢN VỚI MÁY HỌC VECTOR HỖ TRỢ VÀ CÂY QUYẾT ĐỊNH” của Nguyễn Thị Thanh Huyền và Nguyễn Thị Thu Hà, năm 2012 [6]. Bài báo này sử dụng các đặc trưng TF-IDF và n-gram để biểu diễn văn bản, và sử dụng các thuật toán SVM và Decision Tree để phân lớp văn bản. Bài báo này sử dụng bộ dữ liệu VNTC [7] để thực nghiệm, và đạt được kết quả khá tốt, với độ chính xác trung bình là 0.87 cho SVM và 0.85 cho Decision Tree. Tuy nhiên, bài báo này cũng có một số hạn chế, ví dụ như không xử lý được các từ ghép, từ láy, từ mượn trong tiếng Việt, cũng như không xét đến các yếu tố ngữ cảnh và ngữ nghĩa của văn bản.

Phương pháp dựa trên học sâu (deep learning):

Phương pháp này sử dụng các mô hình học sâu, ví dụ như mạng nơ-ron tích chập (CNN), mạng nơ-ron hồi quy (RNN), mạng nơ-ron biểu diễn từ (BERT), v.v., để học trực tiếp từ dữ liệu văn bản và phân lớp văn bản. Phương pháp này có ưu điểm là có thể học được các đặc trưng phong phú, phức tạp và bắt được ngữ cảnh của văn bản, cũng như có thể xử lý được các văn bản dài và đa dạng. Tuy nhiên, phương pháp này cũng có nhược điểm là đòi hỏi nhiều dữ liệu để huấn luyện, cũng như tốn nhiều tài nguyên tính toán và thời gian. Một ví dụ của phương pháp này là sử dụng các mô hình học sâu, ví dụ như mạng nơ-ron tích chập (CNN), mạng nơ-ron hồi quy (RNN), mạng nơ-ron biểu diễn từ (BERT), v.v., để học trực tiếp từ dữ liệu văn bản và phân lớp văn bản. Phương pháp này có ưu điểm là có thể học được các đặc trưng phong phú, phức tạp và bắt được ngữ cảnh của văn bản, cũng như có thể xử lý được các văn bản dài và đa dạng. Tuy nhiên, phương pháp này cũng có nhược điểm là đòi hỏi nhiều dữ liệu để huấn luyện, cũng như tốn nhiều tài nguyên tính toán và thời gian. Một ví dụ của phương pháp này là bài báo Hoang, Vu Cong Duy, et al. "A Comparative Study on Vietnamese Text Classification Methods, Research, Innovation and Vision for the Future." 2007 IEEE [8]. Bài báo tập trung vào vấn đề phân loại văn bản tiếng Việt, nghĩa là tự động gán các đoạn văn bản (hoặc tài liệu) cho các danh mục (hoặc chủ đề) được định trước.Bài báo trình bày hai phương pháp khác nhau cho vấn đề phân loại văn bản tiếng Việt. Cụ thể là sử dụng phương pháp "Bag Of Words - BOW" và "Statistical N-Gram Language Modeling - N-Gram." Đây là hai phương pháp phổ biến được sử dụng trong phân loại văn bản.Nghiên cứu dựa trên một nguồn dữ liệu văn bản tiếng Việt từ một bộ sưu tập tin tức.Bài báo cho biết rằng các phương pháp đã sử dụng có thể đạt được độ chính xác trung bình trên 95%, với thời gian phân loại trung bình là 79 phút cho khoảng 14,000 văn bản. Bài báo cũng thực hiện phân tích ưu và nhược điểm của mỗi phương pháp để tìm ra phương pháp tốt nhất trong các tình huống cụ thể. 1.2.3 Phương pháp đề xuất giải quyết bài toán

Như đã trình bày ở phần trên, trong bài toán này, đề xuất sử dụng phương pháp học sâu và sử dụng mô hình PhoBert để giải quyết . Chúng tôi có làm thêm hai mô hình học máy là Navie Bayes và Linear Regression để so sánh với hai mô hình mà bài toán yêu cầu phía trên.

LSTM [9] là viết tắt của Long Short-Term Memory, là một loại mạng nơ-ron hồi quy nâng cao, có thể xử lý các dữ liệu dạng chuỗi, như văn bản, âm thanh, video, v.v. LSTM được thiết kế để khắc phục vấn đề phụ thuộc dài hạn mà các mạng nơ-ron hồi quy cơ bản (RNN) gặp phải. LSTM có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, bằng cách sử dụng một bộ nhớ đặc biệt được gọi là trạng thái tế bào (cell state). LSTM cũng có ba loại cổng, là cổng quên (forget gate), cổng đầu vào (input gate) và cổng đầu ra (output gate), giúp điều chỉnh việc lưu trữ, cập nhật và truy xuất thông tin từ trạng thái tế bào.



**Hình 1:** Kiến trúc LSTM [9]

Mô hình LSTM có nhiều ưu điểm trong bài toán phân loại văn bản, ví dụ như:

* Có thể học được các đặc trưng ngữ nghĩa và ngữ cảnh của văn bản, từ đó cải thiện hiệu năng của bài toán phân loại.
* Có thể xử lý được các văn bản dài và có nhiều mối quan hệ dài hạn giữa các từ, cụm từ hoặc câu.
* Có thể áp dụng cho các bài toán khác liên quan đến xử lý ngôn ngữ tự nhiên, ví dụ như dịch máy, sinh văn bản, trả lời câu hỏi, v.v.

Tuy nhiên, mô hình LSTM cũng có một số nhược điểm, ví dụ như:

* Đòi hỏi nhiều dữ liệu để huấn luyện, cũng như tốn nhiều tài nguyên tính toán và thời gian.
* Khó khăn trong việc giải thích cơ chế hoạt động và kết quả của mô hình, do tính phức tạp và phi tuyến của mô hình.
* Có thể bị ảnh hưởng bởi nhiễu hoặc sai sót trong dữ liệu, do mô hình học trực tiếp từ dữ liệu mà không cần trích xuất đặc trưng thủ công.

Kết luận, mô hình LSTM là một mô hình học sâu hiệu quả và phổ biến trong bài toán phân loại văn bản, nhưng cũng có một số hạn chế cần được cải thiện.

Mô hình PHO-BERT [10] là một mô hình học sâu dựa trên kiến trúc Transformer, có khả năng mã hóa một đoạn văn bản tiếng Việt thành một vector đặc trưng có khả năng biểu diễn được ngữ nghĩa và ngữ cảnh của văn bản. Mô hình PHO-BERT được tiền huấn luyện trên một lượng lớn dữ liệu tiếng Việt, bao gồm Wikipedia và các bài báo trên mạng. Mô hình PHO-BERT có thể được sử dụng cho các bài toán xử lý ngôn ngữ tự nhiên tiếng Việt, ví dụ như phân lớp văn bản, phân tích cảm xúc, trả lời câu hỏi, v.v.

Mô hình PHO-BERT có hai phiên bản là “base” và “large”, tương ứng với 135 triệu và 370 triệu tham số. Mô hình PHO-BERT được tối ưu hóa dựa trên mô hình RoBERTa, một phiên bản cải tiến của mô hình BERT. Mô hình PHO-BERT cũng được tinh chỉnh để phù hợp với đặc trưng của tiếng Việt, ví dụ như phân biệt được các từ đồng âm, đồng nghĩa, đồng hình, v.v., cũng như bắt được các từ ghép, từ láy, từ mượn, v.v., trong tiếng Việt.

Mô hình PHO-BERT đã đạt được kết quả cao nhất trên bốn bài toán xử lý ngôn ngữ tự nhiên tiếng Việt, là gán nhãn từ loại, phân tích cú pháp, nhận dạng thực thể và suy luận ngôn ngữ tự nhiên. Mô hình PHO-BERT cũng đã vượt qua các mô hình học sâu khác, ví dụ như CNN, RNN hay BERT tiếng Anh.

Naive Bayes [11] là một thuật toán học máy được sử dụng để giải quyết các bài toán phân loại. Nó dựa trên định lí Bayes, một quy tắc toán học cho phép tính xác suất của một sự kiện dựa trên các sự kiện liên quan đã biết. Naive Bayes được gọi là “naive” (ngây thơ) vì nó giả định rằng các đặc trưng (tính năng) của dữ liệu là độc lập với nhau, điều này có thể không đúng trong thực tế.

Naive Bayes là một thuật toán phân lớp được mô hình hoá dựa trên định lý Bayes trong xác suất thống kê:

Trong đó:

* P(y|X) gọi là posterior probability: xác suất của mục tiêu y với điều kiện có đặc trưng X
* P(X|y) gọi là likelihood: xác suất của đặc trưng X khi đã biết mục tiêu y
* P(y) gọi là prior probability của mục tiêu y
* P(X) gọi là prior probability của đặc trưng X

Ở đây, X là vector các đặc trưng, có thể viết dưới dạng:

Khi đó, đẳng thức Bayes trở thành:

Trong mô hình Naive Bayes, có hai giả thiết được đặt ra:

* Các đặc trưng đưa vào mô hình là độc lập với nhau. Tức là sự thay đổi giá trị của một đặc trưng không ảnh hưởng đến các đặc trưng còn lại.
* Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau đối với đầu ra mục tiêu.

Khi đó, kết quả mục tiêu y để P(y|X) đạt cực đại trở thành:

Chính vì hai giả thiết gần như không tồn tại trong thực tế trên, mô hình này mới được gọi là "naive". Tuy nhiên, chính sự đơn giản của nó với việc dự đoán rất nhanh kết quả đầu ra khiến nó được sử dụng rất nhiều trong thực tế trên những bộ dữ liệu lớn, đem lại kết quả khả quan. Một vài ứng dụng của Naive Bayes có thể kể đến như: lọc thư rác, phân loại văn bản, dự đoán sắc thái văn bản, …

Trong đồ án này chúng tôi sử dụng Multinomial Naive Bayes[12]. Mô hình này chủ yếu được sử dụng trong phân loại văn bản. Đặc trưng đầu vào ở đây chính là tần suất xuất hiện của từ trong văn bản đó.

Naive Bayes có nhiều ưu điểm trong bài toán phân loại văn bản, ví dụ như:

* Đơn giản, dễ hiểu và có thể đạt kết quả khá tốt trên các bộ dữ liệu nhỏ và ít nhiễu.
* Nhanh, hiệu quả và dễ cài đặt, có thể sử dụng các thư viện sẵn có như sklearn, nltk, v.v.
* Có thể xử lý được các bài toán phân loại nhị phân hoặc đa lớp, cũng như có thể kết hợp với các kỹ thuật khác như feature selection, smoothing, v.v.

Tuy nhiên, Naive Bayes cũng có một số nhược điểm, ví dụ như:

* Phụ thuộc nhiều vào chất lượng và độ đa dạng của các đặc trưng, cũng như khó khăn trong việc xử lý các văn bản dài, phức tạp và có nhiều ngữ cảnh.
* Giả định về tính độc lập của các đặc trưng có thể không đúng trong thực tế, dẫn đến sai lệch trong việc ước lượng xác suất.
* Có thể bị ảnh hưởng bởi hiện tượng zero-frequency, tức là khi một đặc trưng không xuất hiện trong tập huấn luyện, xác suất có điều kiện của nó sẽ bằng không, làm giảm xác suất hậu nghiệm của chủ đề.

Mô hình Naive Bayes là mô hình phân lớp đơn giản dễ cài đặt, có tốc độ xử lý nhanh. Tuy nhiên có nhược điểm lớn là yêu cầu các đặc trưng đầu vào phải độc lập, mà điều này khó xảy ra trong thực tế làm giảm chất lượng của mô hình.

Linear Regression [13] là một thuật toán học có giám sát (supervised learning) trong Machine Learning, nó là một phương pháp thống kê dùng để ước lượng mối quan hệ giữa các biến độc lập (input features) và biến phụ thuộc (output target). Linear Regression giả định rằng sự tương quan giữa các biến là tuyến tính, từ đó tìm ra hàm tuyến tính tốt nhất để biểu diễn mối quan hệ này. Thuật toán này dự báo giá trị của biến output từ các giá trị của các biến đầu vào.

Có hai loại chính của Linear Regression:

1. Simple Linear Regression [14]: Mô hình này chỉ có một biến độc lập (input feature) mô tả mối quan hệ tuyến tính giữa biến phụ thuộc (output target) và biến độc lập. Phương trình của Simple Linear Regression có dạng + , trong đó a là điểm giao với trục tung (chỉ số độc lập), b là hệ số góc (độ dốc) của đường thẳng, và là sai số.

2. Multiple Linear Regression [15]: Mô hình này có nhiều hơn một biến độc lập, biểu diễn mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc. Phương trình của Multiple Linear Regression có dạng:

trong đó a là điểm giao với trục tung là các hệ số góc, và là sai số.

Mục tiêu của Linear Regression là tìm ra hệ số góc và điểm giao với trục tung sao cho hàm dự đoán tuyến tính đạt được sai số nhỏ nhất. Một trong những cách phổ biến để ước lượng các hệ số là sử dụng phương pháp Ordinary Least Squares (OLS), trong đó chúng ta cần tối thiểu hóa tổng bình phương sai số (sum of squared error).

Chúng ta xem xét mô hình hồi quy tuyến tính đa biến, phương trình của nó có dạng:

Trong đó a là điểm giao với trục tung là các hệ số góc, và là sai số.

Mục tiêu của chúng ta là tìm ra các hệ số của phương trình để tối thiểu hóa tổng bình phương sai số (RSS):

Trong đó, m là số lượng mẫu dữ liệu, là giá trị thực tế của biến phụ thuộc tại mẫu thứ i và là giá trị dự đoán tại mẫu thứ i .Để tối thiểu hóa tổng bình phương sai số, chúng ta có thể sử dụng phương pháp Gradient Descent [16] hoặc phương pháp Normal Equation [17].

Linear Regression được ứng dụng rộng rãi trong nhiều lĩnh vực, như:

* Dự báo giá cả: dự đoán giá nhà, giá cổ phiếu, giá nhiên liệu dựa trên các yếu tố như vị trí, kích thước, chất lượng, lượng cung cầu, …
* Dự báo điểm số: dự đoán điểm số của học sinh dựa trên thời gian học, nỗ lực, kỹ năng, trình độ giáo viên, …
* Dự báo sản phẩm: dự đoán đầu ra sản xuất dựa trên thời gian, công suất, nguyên liệu, lao động, …
* Phân tích chuỗi thời gian: dự đoán xu hướng và chu kỳ của các chuỗi dữ liệu, như bất động sản, thời tiết, xu hướng sản xuất, …

Linear Regression có nhiều ưu điểm trong bài toán phân loại văn bản, ví dụ như:

* Đơn giản, dễ hiểu và có thể đạt kết quả khá tốt trên các bộ dữ liệu nhỏ và ít nhiễu.
* Nhanh, hiệu quả và dễ cài đặt, có thể sử dụng các thư viện sẵn có như sklearn, nltk, v.v.
* Có thể xử lý được các bài toán phân loại nhị phân hoặc đa lớp, cũng như có thể kết hợp với các kỹ thuật khác như feature selection, regularization, v.v.

Tuy nhiên, Linear Regression cũng có một số nhược điểm, ví dụ như:

* Phụ thuộc nhiều vào chất lượng và độ đa dạng của các đặc trưng, cũng như khó khăn trong việc xử lý các văn bản dài, phức tạp và có nhiều ngữ cảnh.
* Giả định về tính tuyến tính của mối quan hệ giữa các biến có thể không đúng trong thực tế, dẫn đến sai lệch trong việc ước lượng nhãn.
* Có thể bị ảnh hưởng bởi hiện tượng overfitting, tức là mô hình quá khớp với dữ liệu huấn luyện, làm giảm khả năng tổng quát hóa của mô hình.

1.3 Phương pháp giải quyết bài toán

1.3.1. Mô hình tổng quát

**Hình 2:** Mô hình tổng quát

Vietnamese-text-

classification

Process raw data

Extract labels and content

Label encoding

Encode text sentences

Create mask tensor

Create DataLoader

Training model

PhoBert

LSTM

Naive Bayes

Linear Regression

Save model

1.3.2 Đặc trưng của mô hình đề xuất

1.3.2.1. Dữ liệu

Ở đồ án này chúng tôi sử dụng bộ dữ liệu Vietnamese-text-classification, một bộ dữ liệu phân lớp văn bản tiếng Việt được thu thập từ trang web báo chí vnexpress.net . Bộ dữ liệu này gồm 8 chủ đề, mỗi chủ đề có khoảng 10.000 văn bản. Tôi chia bộ dữ liệu này thành ba tập: tập huấn luyện (80%), tập kiểm tra (20%)

1.3.2.2 Xử lý dữ liệu

Nhóm của chúng tôi xử lý dữ liệu với năm bước như sau :

* Extract labels and content
* Label encoding
* Encode text sentences
* Create mask tensor
* Create dataloader

Extract labels and content

Chúng tôi sử dụng hàm make\_data được sử dụng để lấy nội dung và nhãn của các tệp văn bản trong thư mục huấn luyện hoặc thư mục kiểm thử. Hàm này duyệt qua các thư mục con trong thư mục đầu vào, mỗi thư mục con là một nhãn của văn bản. Sau đó, hàm duyệt qua các tệp văn bản trong mỗi thư mục con và đọc nội dung của chúng. Tiếp theo, hàm sử dụng thư viện rdrsegmenter để phân đoạn từ cho nội dung của mỗi tệp văn bản.Thư viện rdrsegmenter là một thư viện Python được sử dụng để tách văn bản tiếng Việt thành các từ. Thư viện này được phát triển bởi nhóm nghiên cứu Vietnamese Natural Language Processing (VNLP) thuộc Đại học Quốc gia Hà Nội. Cuối cùng, hàm nối các từ trong mỗi câu thành một chuỗi, và nối các chuỗi này thành một chuỗi duy nhất, là nội dung đã được xử lý của mỗi tệp văn bản. Hàm cũng thêm nội dung và nhãn của mỗi tệp văn bản vào hai danh sách texts và labels.

Label encoding

Sử dụng công cụ LabelEncoder từ thư viện scikit-learn để mã hóa nhãn văn bản thành dạng số nguyên. Mục đích của việc này là để các mô hình học máy có thể hiểu và xử lý nhãn một cách dễ dàng hơn.

Cụ thể, thực hiện các bước sau:

* Tạo một đối tượng LabelEncoder.
* Huấn luyện LabelEncoder trên nhãn của tập huấn luyện. Gọi phương thức fit trên đối tượng lb\_encoder với tham số là tập nhãn huấn luyện (train\_labels). Phương thức này sẽ học được các nhãn duy nhất trong tập nhãn huấn luyện và gán cho chúng một giá trị số nguyên từ 0 đến n\_classes - 1, trong đó n\_classes là số lượng nhãn khác nhau.
* Mã hóa nhãn của tập huấn luyện và tập kiểm thử thành dạng số nguyên.Gọi phương thức transform trên đối tượng lb\_encoder với tham số là tập nhãn huấn luyện và tập nhãn kiểm thử (train\_labels và test\_labels). Phương thức này sẽ chuyển đổi các nhãn thành các giá trị số nguyên tương ứng đã được học ở bước trước. Kết quả của phương thức này được gán cho các biến en\_train\_labels và en\_test\_labels, là các tập nhãn đã được mã hóa của tập huấn luyện và tập kiểm thử.
* In ra các lớp (nhãn) mà LabelEncoder đã tìm hiểu.

Encode text sentences

Để chuyển đổi một danh sách các câu văn thành một tensor các số nguyên chúng tôi sử dụng một mô hình BPE [18] và một bộ từ vựng đã được học trước. Mô hình BPE là một kỹ thuật phân đoạn từ dựa trên tần suất xuất hiện của các cặp ký tự liên tiếp. Mô hình này có thể giúp cải thiện độ chính xác của quá trình mã hóa văn bản thành các số nguyên. Bộ từ vựng là một tập hợp các từ và các đơn vị con, được gán cho các số nguyên duy nhất. Bộ từ vựng này được sử dụng để mã hóa các câu văn thành các số nguyên.Sau khi đã chuyển đổi tất cả các câu văn trong tập dữ liệu thành các số nguyên, cần đệm các danh sách các số nguyên sao cho chúng có cùng độ dài. Điều này giúp các mô hình học máy có thể xử lý các câu văn có độ dài khác nhau. Để thực hiện việc này, chúng tôi sử dụng hàm pad\_sequences. Ví dụ, giả sử có một câu văn sau:

"Đây là một câu văn"

Sau khi sử dụng mô hình BPE và bộ từ vựng, câu văn này sẽ được chuyển đổi thành một chuỗi các đơn vị con sau:

['<s>', 'Đây', 'là', 'một', 'câu', 'văn', '</s>']

Sau khi sử dụng hàm vocab.encode\_line, chuỗi này sẽ được chuyển đổi thành một danh sách các số nguyên sau:

[1, 2, 3, 4, 5, 6, 7]

Cuối cùng, danh sách này sẽ được đệm thành một tensor có độ dài 256 bằng cách sử dụng hàm pad\_sequences. Tensor này sẽ có dạng sau:

[1, 2, 3, 4, 5, 6, 7, 0, 0, 0, ..., 0]

Tương tự, tất cả các câu văn trong tập dữ liệu sẽ được chuyển đổi thành các tensor các số nguyên có cùng độ dài 256. Với 256 là độ dài tối đa của mỗi câu văn sau khi được mã hóa. Độ dài này được xác định dựa trên kích thước của mô hình BPE và bộ từ vựng.

Create mask tensor

Chúng tôi tạo ra một tensor các giá trị nhị phân, để biểu diễn vị trí của các token hợp lệ trong một tensor các số nguyên. Mục đích của việc này là để giúp mô hình học máy tập trung vào các token có giá trị trong câu văn bản, và bỏ qua các token padding. Cụ thể, thực hiện các bước sau:

* Tạo một hàm make\_mask() nhận vào một batch các câu văn bản (batch\_ids) dưới dạng tensor, và trả về một batch các mask tensor tương ứng.
* Duyệt qua từng câu văn bản trong batch.
* Tạo một mask tensor cho câu văn bản hiện tại. Giá trị của mỗi phần tử trong mask là 1 nếu token\_id tương ứng lớn hơn 0 (nghĩa là token không phải là token padding), ngược lại là 0.
* Thêm mask tensor của câu văn bản vào danh sách batch\_mask.
* Trả về danh sách batch\_mask dưới dạng tensor của PyTorch.

Ví dụ, một batch các câu văn bản dưới dạng tensor như sau:

([[1, 2, 3, 4, 5, 0], [6, 7, 8, 9, 0, 0], [10, 11, 12, 0, 0, 0]])

Sẽ được chuyển đổi thành các mask tensor như sau:

([[1, 1, 1, 1, 1, 0], [1, 1, 1, 1, 0, 0], [1, 1, 1, 0, 0, 0]])

Trong các mask tensor này, các phần tử có giá trị 1 tương ứng với các token có giá trị, và các phần tử có giá trị 0 tương ứng với các token padding.

Các mask tensor này sẽ được sử dụng trong quá trình huấn luyện mô hình để bỏ qua các vị trí của token padding trong quá trình tính toán loss. Điều này giúp mô hình tập trung vào các token có giá trị, và tránh bị ảnh hưởng bởi các token padding. Điều này giúp mô hình học cách phân loại các câu văn bản chính xác hơn.

Create dataloader

Tạo các đối tượng DataLoader cho các tập huấn luyện, tập kiểm thử và tập kiểm tra, bằng cách sử dụng các tensor các số nguyên, các tensor các giá trị nhị phân và các tensor các nhãn đã được tạo ra trước đó.

DataLoader là một đối tượng có thể lặp lại trên tập dữ liệu, với cách lấy mẫu tuần tự và kích thước batch được chỉ định. Đối tượng này sẽ được sử dụng để đưa dữ liệu vào mô hình mạng nơ-ron. Tạo DataLoader để đưa dữ liệu vào mô hình mạng nơ-ron một cách hiệu quả.

Cụ thể, DataLoader sẽ giúp thực hiện các công việc sau:

* Phân chia tập dữ liệu thành các batch nhỏ, phù hợp với khả năng xử lý của mô hình mạng nơ-ron.
* Lấy mẫu các batch dữ liệu một cách ngẫu nhiên hoặc tuần tự.
* Cung cấp các batch dữ liệu cho mô hình mạng nơ-ron một cách hiệu quả.

Việc sử dụng DataLoader giúp cho quá trình huấn luyện và kiểm tra mô hình mạng nơ-ron trở nên hiệu quả hơn, đồng thời giúp giảm thiểu thời gian huấn luyện và kiểm tra.

1.3.2.3 Phương pháp huấn luyện

PhoBERT

Trong nghiên cứu này, chúng tôi trình bày quá trình huấn luyện và đánh giá một mô hình phân loại văn bản sử dụng PhoBERT [1], một biến thể của RoBERTa được tối ưu hóa cho tiếng Việt. Mô hình được xây dựng và huấn luyện trên tập dữ liệu phân loại văn bản để hiểu và dự đoán các phân loại liên quan.

Chuẩn Bị Mô Hình:

Mô hình được khởi tạo với cấu hình từ tệp tin config.json và trọng số được load từ tệp model.bin. Số lượng lớp đầu ra (num\_labels) được xác định dựa trên số lớp trong tập dữ liệu, được xác định bởi lb\_encoder.classes\_. Mô hình sau đó được chuyển đến thiết bị GPU để tối ưu hóa tính toán.

Tối Ưu Hóa và Trình Tạo Mô Hình:

Tối ưu hóa AdamW [19] được áp dụng với tốc độ học là 1e-5. Trọng số của mô hình được nhóm lại để áp dụng tỷ lệ giảm trọng lượng khác nhau cho các nhóm. Quá trình này thực hiện bởi một trình tạo mô hình, áp dụng tối ưu hóa cho danh sách tham số cần tối ưu hóa.

Quá Trình Huấn Luyện:

Huấn luyện diễn ra qua nhiều epochs, mỗi epoch bao gồm quá trình đào tạo và đánh giá. Mỗi batch từ tập dữ liệu huấn luyện được đưa vào mô hình để cập nhật trọng số thông qua quá trình lan truyền ngược. Các thống kê như Accuracy và F1-score được tính toán trên tập huấn luyện để đánh giá hiệu suất.

Đánh Giá trên Tập Validation:

Sau mỗi epoch, mô hình được chuyển sang chế độ đánh giá và đánh giá trên tập validation. Các thống kê như Accuracy và F1 score được tính toán để đánh giá khả năng tổng quát hóa của mô hình.

Kết Quả và Theo Dõi Tiến Tiến:

Các kết quả như loss, accuracy, và F1 score được theo dõi và lưu lại sau mỗi epoch. Điều này giúp theo dõi tiến triển của mô hình qua thời gian và cung cấp thông tin về hiệu suất của nó trên cả tập dữ liệu huấn luyện và tập validation.

LSTM

Chúng tôi trình bày một mô hình mạng nơ-ron sử dụng lớp LSTM (Long Short-Term Memory) được xây dựng bằng thư viện Keras, nhằm giải quyết bài toán phân loại văn bản. Mục tiêu là tận dụng khả năng của LSTM trong việc xử lý thông tin tuần tự và chuỗi dài hạn để cải thiện hiệu suất phân loại. Kiến Trúc Mô Hình:

Lớp Embedding:

Mô hình bắt đầu bằng một lớp Embedding để biểu diễn từng từ trong câu dưới dạng vector có số chiều xác định bởi embedding\_dim. Điều này giúp mô hình hiểu được mối quan hệ ngữ nghĩa giữa các từ.

Lớp LSTM:

Chúng tôi tích hợp hai lớp LSTM, mỗi lớp với hidden\_size đơn vị ẩn, giúp mô hình học được các mô hình phức tạp và chuỗi dài hạn. Lớp LSTM đầu tiên được cấu hình để trả về chuỗi đầu ra, trong khi lớp thứ hai chỉ trả về đầu ra của bước cuối cùng. Điều này giúp mô hình hiểu và tổng hợp thông tin từ chuỗi đầu vào.

Lớp Dropout:

Để ngăn chặn hiện tượng quá mức học (overfitting), chúng tôi sử dụng lớp Dropout sau mỗi lớp LSTM, tắt ngẫu nhiên một số đơn vị đầu ra trong quá trình huấn luyện.

Lớp Dense:

Sau khi xử lý chuỗi đầu ra từ LSTM, chúng tôi áp dụng hai lớp Dense, mỗi lớp điều có số đơn vị và hàm kích hoạt tương ứng. Lớp Dense cuối cùng sử dụng hàm kích hoạt softmax để tạo ra xác suất cho mỗi lớp đầu ra trong bài toán phân loại.

Biên Dịch và Huấn Luyện:

Sau khi xây dựng kiến trúc mô hình, chúng tôi sử dụng hàm mất mát là 'categorical\_crossentropy' và thuật toán tối ưu hóa 'adam' để biên dịch mô hình. Độ đo 'accuracy' được chọn để đánh giá hiệu suất của mô hình trong quá trình huấn luyện.

Linear Classifier

Chúng tôi trình bày quá trình xây dựng và huấn luyện một mô hình phân loại văn bản đa đầu ra, sử dụng Logistic Regression và chuyển đổi TF-IDF. Mô hình được xây dựng dựa trên một pipeline tích hợp các bước tiền xử lý và mô hình hóa để hiểu và phân loại các văn bản đa nhãn.

Mô hình được thiết kế dưới dạng một pipeline, bắt đầu bằng bước CountVectorizer để vector hóa văn bản sử dụng biểu diễn Bag of Words (BoW). Bước này cấu hình để sử dụng từng từ đơn lẻ, loại bỏ từ quá thường xuyên (max\_df=0.8), và không giới hạn số lượng từ vựng (max\_features=None).

Tiếp theo, một bước TfidfTransformer được thực hiện để chuyển đổi biểu diễn BoW thành biểu diễn TF-IDF (Term Frequency-Inverse Document Frequency).

Cuối cùng, mô hình sử dụng MultiOutputClassifier để hỗ trợ đa đầu ra, và LogisticRegression được chọn làm mô hình tuyến tính với các tham số cụ thể như solver='lbfgs', multi\_class='auto', và max\_iter=10000.

Huấn Luyện Mô Hình:

Mô hình được huấn luyện trên tập dữ liệu huấn luyện bằng cách gọi phương thức fit trên pipeline. Thời gian huấn luyện được đo lường để đánh giá hiệu suất và tiến triển của mô hình.

Thời gian hoàn thành quá trình huấn luyện và các thống kê hiệu suất khác như độ chính xác và F1 score được ghi nhận và đánh giá. Các kết quả này cung cấp cái nhìn toàn diện về khả năng phân loại và tổng quát hóa của mô hình.

Naive Bayes

Chúng tôi trình bày một mô hình phân loại văn bản đa đầu ra được xây dựng và huấn luyện bằng cách sử dụng thuật toán Naive Bayes, được tích hợp vào một pipeline với các bước tiền xử lý và biểu diễn TF-IDF.

Mô hình được thiết kế dưới dạng một pipeline với các bước chính:

* + CountVectorizer: Biểu diễn văn bản dưới dạng Bag of Words (BoW) với các cấu hình nhất định như sử dụng từng từ đơn lẻ, loại bỏ từ quá thường xuyên, và không giới hạn số lượng từ vựng.
  + TfidfTransformer: Chuyển đổi biểu diễn BoW thành biểu diễn TF-IDF để cân nhắc mức độ quan trọng của từng từ trong văn bản.
  + MultinomialNB: Sử dụng thuật toán Naive Bayes với đa đầu ra thông qua MultiOutputClassifier, với Multinomial Naive Bayes làm mô hình cụ thể.

Huấn Luyện Mô Hình:

Mô hình được huấn luyện trên tập dữ liệu huấn luyện thông qua phương pháp fit trên pipeline. Thời gian huấn luyện được đo lường để đánh giá thời gian thực hiện và tiến triển của mô hình.

Kết Quả và Đánh Giá Hiệu Suất:

Kết quả của quá trình huấn luyện và thông tin về hiệu suất, như độ chính xác và thời gian hoàn thành, được in ra để đánh giá sự hiệu quả của mô hình. Những kết quả này mang lại cái nhìn chi tiết và toàn diện về khả năng phân loại của mô hình Naive Bayes đa đầu ra trong ngữ cảnh phân loại văn bản.

* 1. Thực nghiệm

1.4.1 Dữ liệu

Để huấn luyện và kiểm tra mô hình, tôi sử dụng bộ dữ liệu Vietnamese-text-classification, một bộ dữ liệu phân lớp văn bản tiếng Việt được thu thập từ trang web báo chí vnexpress.net . Bộ dữ liệu này gồm 8 chủ đề, mỗi chủ đề có khoảng 10.000 văn bản.

1.4.2 Xử lý dữ liệu

Nhóm của chúng tôi xử lý dữ liệu với năm bước như sau :

* Extract labels and content
* Label encoding
* Encode text sentences
* Create mask tensor
* Create dataloader

Extract labels and content

Chúng tôi sử dụng hàm make\_data được sử dụng để lấy nội dung và nhãn của các tệp văn bản trong thư mục huấn luyện hoặc thư mục kiểm thử. Hàm này duyệt qua các thư mục con trong thư mục đầu vào, mỗi thư mục con là một nhãn của văn bản. Sau đó, hàm duyệt qua các tệp văn bản trong mỗi thư mục con và đọc nội dung của chúng. Tiếp theo, hàm sử dụng thư viện rdrsegmenter để phân đoạn từ cho nội dung của mỗi tệp văn bản.Thư viện rdrsegmenter là một thư viện Python được sử dụng để tách văn bản tiếng Việt thành các từ. Thư viện này được phát triển bởi nhóm nghiên cứu Vietnamese Natural Language Processing (VNLP) thuộc Đại học Quốc gia Hà Nội. Cuối cùng, hàm nối các từ trong mỗi câu thành một chuỗi, và nối các chuỗi này thành một chuỗi duy nhất, là nội dung đã được xử lý của mỗi tệp văn bản. Hàm cũng thêm nội dung và nhãn của mỗi tệp văn bản vào hai danh sách texts và labels.

Label encoding

Sử dụng công cụ LabelEncoder từ thư viện scikit-learn để mã hóa nhãn văn bản thành dạng số nguyên. Mục đích của việc này là để các mô hình học máy có thể hiểu và xử lý nhãn một cách dễ dàng hơn.

Cụ thể, thực hiện các bước sau:

* Tạo một đối tượng LabelEncoder.
* Huấn luyện LabelEncoder trên nhãn của tập huấn luyện. Gọi phương thức fit trên đối tượng lb\_encoder với tham số là tập nhãn huấn luyện (train\_labels). Phương thức này sẽ học được các nhãn duy nhất trong tập nhãn huấn luyện và gán cho chúng một giá trị số nguyên từ 0 đến n\_classes - 1, trong đó n\_classes là số lượng nhãn khác nhau.
* Mã hóa nhãn của tập huấn luyện và tập kiểm thử thành dạng số nguyên.Gọi phương thức transform trên đối tượng lb\_encoder với tham số là tập nhãn huấn luyện và tập nhãn kiểm thử (train\_labels và test\_labels). Phương thức này sẽ chuyển đổi các nhãn thành các giá trị số nguyên tương ứng đã được học ở bước trước. Kết quả của phương thức này được gán cho các biến en\_train\_labels và en\_test\_labels, là các tập nhãn đã được mã hóa của tập huấn luyện và tập kiểm thử.
* In ra các lớp (nhãn) mà LabelEncoder đã tìm hiểu.

Encode text sentences

Để chuyển đổi một danh sách các câu văn thành một tensor các số nguyên chúng tôi sử dụng một mô hình BPE và một bộ từ vựng đã được học trước.Mô hình BPE là một kỹ thuật phân đoạn từ dựa trên tần suất xuất hiện của các cặp ký tự liên tiếp. Mô hình này có thể giúp cải thiện độ chính xác của quá trình mã hóa văn bản thành các số nguyên. Bộ từ vựng là một tập hợp các từ và các đơn vị con, được gán cho các số nguyên duy nhất. Bộ từ vựng này được sử dụng để mã hóa các câu văn thành các số nguyên.Sau khi đã chuyển đổi tất cả các câu văn trong tập dữ liệu thành các số nguyên, cần đệm các danh sách các số nguyên sao cho chúng có cùng độ dài. Điều này giúp các mô hình học máy có thể xử lý các câu văn có độ dài khác nhau. Để thực hiện việc này, chúng tôi sử dụng hàm pad\_sequences. Ví dụ, giả sử có một câu văn sau:

"Đây là một câu văn"

Sau khi sử dụng mô hình BPE và bộ từ vựng, câu văn này sẽ được chuyển đổi thành một chuỗi các đơn vị con sau:

['<s>', 'Đây', 'là', 'một', 'câu', 'văn', '</s>']

Sau khi sử dụng hàm vocab.encode\_line, chuỗi này sẽ được chuyển đổi thành một danh sách các số nguyên sau:

[1, 2, 3, 4, 5, 6, 7]

Cuối cùng, danh sách này sẽ được đệm thành một tensor có độ dài 256 bằng cách sử dụng hàm pad\_sequences. Tensor này sẽ có dạng sau:

[1, 2, 3, 4, 5, 6, 7, 0, 0, 0, ..., 0]

Tương tự, tất cả các câu văn trong tập dữ liệu sẽ được chuyển đổi thành các tensor các số nguyên có cùng độ dài 256 là độ dài tối đa của mỗi câu văn sau khi được mã hóa. Độ dài này được xác định dựa trên kích thước của mô hình BPE và bộ từ vựng.

Create mask tensor

Chúng tôi tạo ra một tensor các giá trị nhị phân, để biểu diễn vị trí của các token hợp lệ trong một tensor các số nguyên.Mục đích của việc này là để giúp mô hình học máy tập trung vào các token có giá trị trong câu văn bản, và bỏ qua các token padding.Cụ thể, thực hiện các bước sau:

* Tạo một hàm make\_mask() nhận vào một batch các câu văn bản (batch\_ids) dưới dạng tensor, và trả về một batch các mask tensor tương ứng.
* Duyệt qua từng câu văn bản trong batch.
* Tạo một mask tensor cho câu văn bản hiện tại. Giá trị của mỗi phần tử trong mask là 1 nếu token\_id tương ứng lớn hơn 0 (nghĩa là token không phải là token padding), ngược lại là 0.
* Thêm mask tensor của câu văn bản vào danh sách batch\_mask.
* Trả về danh sách batch\_mask dưới dạng tensor của PyTorch.

Ví dụ, một batch các câu văn bản dưới dạng tensor như sau:

([[1, 2, 3, 4, 5, 0], [6, 7, 8, 9, 0, 0], [10, 11, 12, 0, 0, 0]])

Sẽ được chuyển đổi thành các mask tensor như sau:

([[1, 1, 1, 1, 1, 0], [1, 1, 1, 1, 0, 0], [1, 1, 1, 0, 0, 0]])

Trong các mask tensor này, các phần tử có giá trị 1 tương ứng với các token có giá trị, và các phần tử có giá trị 0 tương ứng với các token padding.

Các mask tensor này sẽ được sử dụng trong quá trình huấn luyện mô hình để bỏ qua các vị trí của token padding trong quá trình tính toán loss. Điều này giúp mô hình tập trung vào các token có giá trị, và tránh bị ảnh hưởng bởi các token padding.Điều này giúp mô hình học cách phân loại các câu văn bản chính xác hơn.

Create dataloader

Tạo các đối tượng DataLoader cho các tập huấn luyện, tập kiểm thử và tập kiểm tra, bằng cách sử dụng các tensor các số nguyên, các tensor các giá trị nhị phân và các tensor các nhãn đã được tạo ra trước đó.

DataLoader là một đối tượng có thể lặp lại trên tập dữ liệu, với cách lấy mẫu tuần tự và kích thước batch được chỉ định. Đối tượng này sẽ được sử dụng để đưa dữ liệu vào mô hình mạng nơ-ron.Tạo DataLoader để đưa dữ liệu vào mô hình mạng nơ-ron một cách hiệu quả.

Cụ thể, DataLoader sẽ giúp thực hiện các công việc sau:

* Phân chia tập dữ liệu thành các batch nhỏ, phù hợp với khả năng xử lý của mô hình mạng nơ-ron.
* Lấy mẫu các batch dữ liệu một cách ngẫu nhiên hoặc tuần tự.
* Cung cấp các batch dữ liệu cho mô hình mạng nơ-ron một cách hiệu quả.

Việc sử dụng DataLoader giúp cho quá trình huấn luyện và kiểm tra mô hình mạng nơ-ron trở nên hiệu quả hơn, đồng thời giúp giảm thiểu thời gian huấn luyện và kiểm tra.

1.4.3 Công nghệ sử dụng

|  |  |
| --- | --- |
| Ngôn ngữ | Python [20] |
| Thư viện | PyTorch [21], Tensorflow [22], Transformers [23], Keras [24] |
| Môi trường | Kaggle GPU P100 |

**Bảng 1:** Công nghệ sử dụng

1.4.4 Cách đánh giá

1. Độ chính xác (Accuracy)[25]

Độ chính xác được tính bằng cách so sánh số dự đoán đúng với tổng số mẫu trong tập kiểm thử.

2. F1-Score [26]

F1-score là một độ đo kết hợp giữa độ chính xác và độ phủ (recall). Nó được tính theo công thức:

Ở đây, Precision [27] là tỷ lệ số dự đoán đúng tích cực (True Positive) trên tổng số dự đoán tích cực (True Positive + False Positive), và Recall [27] là tỷ lệ số dự đoán đúng tích cực trên tổng số thực tế tích cực (True Positive + False Negative).

# 1.5 Kết quả đạt được

* Với mô hình PhoBert

Số lượng epoch được thiết lập là 5 (EPOCHS = 5).

Tối ưu hóa được thực hiện bằng trình tối ưu hóa AdamW với tỷ lệ học là 1e-5.

* Với mô hình LSTM

Số lượng epoch được thiết lập là 5 (EPOCHS = 5).

Tối ưu hóa được thực hiện bằng trình tối ưu hóa Adam

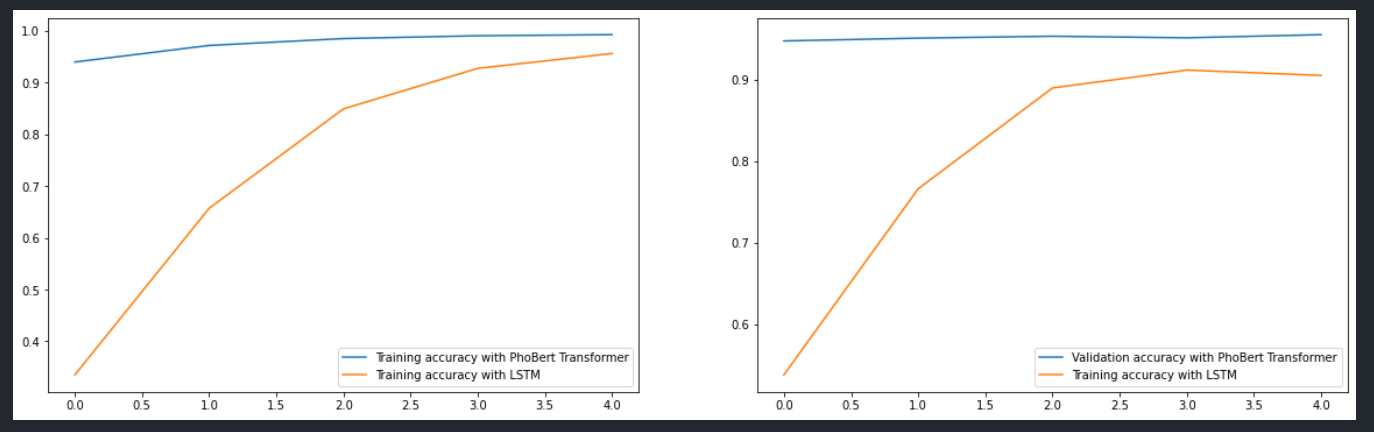
* Với mô hình Linear Regression

Trong LogisticRegression sử dụng solver là 'lbfgs', multi\_class là 'auto', và max\_iter là 10000.

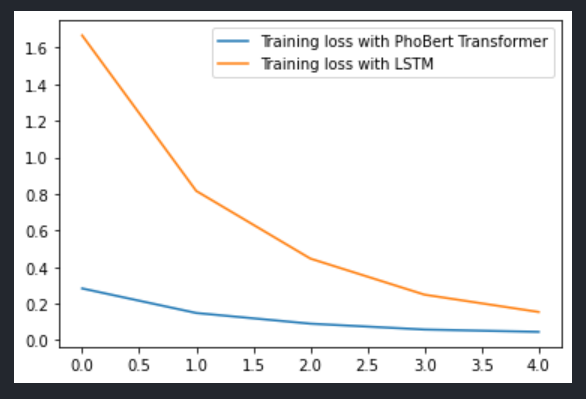
* Với mô hình Naive Bayes

Trong MultiOutputClassifier, MultinomialNB() được sử dụng làm mô hình phân loại cho mỗi đầu ra.

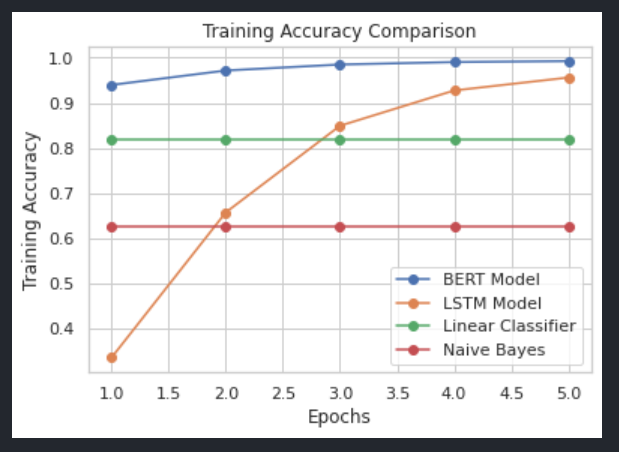
Dựa trên các phương pháp, tham số đã được nêu ra ở trên, chúng tôi tiến hành thực nghiệm để phân loại các văn bản và cho ra các kết quả mô hình như sau.



**Hình 3:** So sánh accuracy của hai mô hình học sâu PhoBert và LSTM



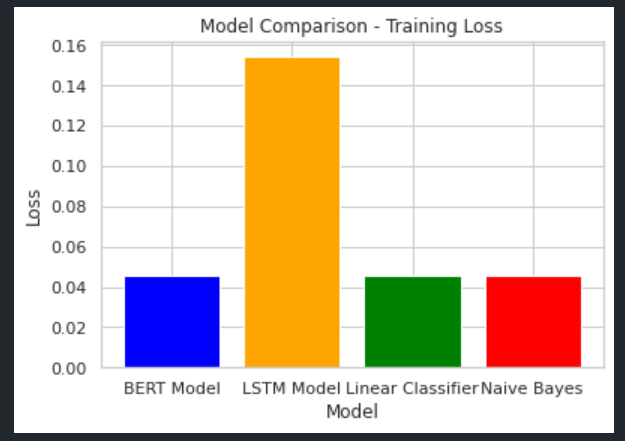
**Hình 4:** So sánh loss của hai mô hình học sâuPhoBert và LSTM



**Hình 5:** So sánh accuracy trên từng epochs



**Hình 6:** So sánh accuracy của từng mô hình



**Hình 7:** So sánh loss của từng mô hình

Kết quả so sánh:

Nhìn vào các biểu đồ ở trên ta thấy mô hình Phobert cho ra kết quả chính xác nhất trong các mô hình.Vì nó có thể tận dụng được kiến thức ngôn ngữ đã được tiền huấn luyện, và có thể tinh chỉnh được theo nhu cầu của bài toán cụ thể. Mô hình PhoBERT cũng có thể xử lý được các vấn đề đặc thù của tiếng Việt, như phân đoạn từ, chuẩn hóa chữ cái, xử lý các từ mượn, v.v.

1.6 Kết luận

1.6.1 Kết luận

Về mặt lý thuyết, Đồ án đã tìm hiểu về các phương pháp giải quyết bài toán dự đoán được trạng thái ngủ , đồng thời Đồ án cũng đề xuất phương pháp học sâu để giải quyết bài toán này.

1.6.2 Hướng phát triển

Trong tương lai mong muốn phát triển thêm hướng giải quyết cách tiếp cận bài toán theo một hướng mới dùng các mô hình sẵn có để kết hợp lại với nhau để thực nghiệm cho ra các phương pháp, các kết quả khả quan hơn. Có thể kết hợp với các mô hình khác như mô hình RNN, mô hình Transformer, hoặc mô hình BERT để cải thiện hiệu năng, hoặc có thể áp dụng mô hình cho các bài toán khác như phân tích cảm xúc, phát hiện tin giả, v.v.

# TÀI LIỆU THAM KHẢO

1. Le Binh, D., Minh, H. L. N., Diem, Q. N., & Bao, D. T. N. (2023, July). An Extraction-based Approach for Vietnamese Legal Text Summarization. In *2023 International Conference on System Science and Engineering (ICSSE)* (pp. 61-66). IEEE.
2. De Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.
3. Soufyane, A., Abdelhakim, B. A., & Ahmed, M. B. (2021, January). An intelligent chatbot using NLP and TF-IDF algorithm for text understanding applied to the medical field. In *Emerging Trends in ICT for Sustainable Development: The Proceedings of NICE2020 International Conference* (pp. 3-10). Cham: Springer International Publishing.
4. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernández, L. (2014). Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, *41*(3), 853-860.
5. Adewumi, T., Liwicki, F., & Liwicki, M. (2022). Word2Vec: Optimal hyperparameters and their impact on natural language processing downstream tasks. *Open Computer Science*, *12*(1), 134-141.
6. Đệ, T. C., & Khang, P. N. (2012). Phân loại văn bản với máy học Vector hỗ trợ và cây quyết định. *Tạp chí Khoa học Đại học cần Thơ*, (21a), 52-63.
7. Github <https://github.com/duyvuleo/VNTC.git>
8. Hoang, V. C. D., Dinh, D., Le Nguyen, N., & Ngo, H. Q. (2007, March). A comparative study on vietnamese text classification methods. In *2007 IEEE international conference on research, innovation and vision for the future* (pp. 267-273). IEEE.
9. Kasthuri, E., & Balaji, S. (2023). Natural language processing and deep learning chatbot using long short term memory algorithm. *Materials Today: Proceedings*, *81*, 690-693.
10. Nguyen, D. Q., & Nguyen, A. T. (2020). PhoBERT: Pre-trained language models for Vietnamese. *arXiv preprint arXiv:2003.00744*.
11. Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. *Encyclopedia of machine learning*, *15*(1), 713-714.
12. Jiang, L., Wang, S., Li, C., & Zhang, L. (2016). Structure extended multinomial naive Bayes. *Information Sciences*, *329*, 346-356.
13. Su, X., Yan, X., & Tsai, C. L. (2012). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *4*(3), 275-294.
14. Zou, K. H., Tuncali, K., & Silverman, S. G. (2003). Correlation and simple linear regression. *Radiology*, *227*(3), 617-628.
15. Eberly, L. E. (2007). Multiple linear regression. *Topics in Biostatistics*, 165-187.
16. Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
17. Fecher, T., Pail, R., Gruber, T., GOCO Consortium University of Bonn, Institute of Geodesy and Geoinformation WD Schuh J Kusche JM Brockmann I Loth S Müller A Eicker J Schall, Graz University of Technology, Institute of Geodesy T Mayer-Gürr A Kvas B Klinger D Rieser N Zehentner, Austrian Academy of Sciences, Space Research Institute O Baur E Höck S Krauss, & University of Bern, Astronomical Institute A Jäggi U Meyer L Prange A Maier. (2017). GOCO05c: a new combined gravity field model based on full normal equations and regionally varying weighting. *Surveys in geophysics*, *38*, 571-590.
18. Preiser, W. F., & Schramm, U. (2012). A process model for building performance evaluation (BPE). *Enhancing building performance*, 19-31.
19. Llugsi, R., El Yacoubi, S., Fontaine, A., & Lupera, P. (2021, October). Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)* (pp. 1-6). IEEE.
20. Python <https://www.python.org/>
21. Pytorch <https://pytorch.org/>
22. Tensorflow <https://www.tensorflow.org/?hl=vi>
23. Transformers <https://pypi.org/project/transformers/>
24. Keras <https://keras.io/>
25. Fleuren, L. M., Klausch, T. L., Zwager, C. L., Schoonmade, L. J., Guo, T., Roggeveen, L. F., ... & Elbers, P. W. (2020). Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive care medicine*, *46*, 383-400.
26. Humphrey, A., Kuberski, W., Bialek, J., Perrakis, N., Cools, W., Nuyttens, N., ... & Cunha, P. A. C. (2022). Machine-learning classification of astronomical sources: estimating F1-score in the absence of ground truth. *Monthly Notices of the Royal Astronomical Society: Letters*, *517*(1), L116-L120.
27. Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240).

# LÀM VIỆC NHÓM

Cách thức làm việc nhóm: gặp nhau trực tiếp để trao đổi, tìm kiếm tài liệu hoặc họp online qua discord,qua các buổi thực hành trên lớp.

Phân chia công việc của các thành viên trong nhóm:

|  |  |  |
| --- | --- | --- |
| Thành viên | Nhiệm vụ | Mức độ hoàn thành |
| Nguyễn Quang Bảo | 1.3 Phương pháp giải quyết bài toán  1.4 Thực nghiệm  1.5 Kết quả đạt được | 100% |
| Nguyễn Xuân Giang | 1.1 Giới thiệu về bài toán 1.2 Phân tích yêu cầu của bài toán 1.6 Kết luận | 100% |

Tổng số lần gặp nhau:

Trực tiếp: 5 lần

Online: 10 lần

Tổng thời gian gặp nhau: Khoảng 25 giờ

# TỰ ĐÁNH GIÁ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Chương | Nội dung | Điểm chuẩn | Tự chấm | Ghi chú |
| 1 | Giới thiệu về bài toán | 0.5 | 0.5 |  |
| 2 | Phân tích yêu cầu của bài toán | 1.5 | 1.5 |  |
| 3 | Phương pháp giải quyết bài toán | 2.5 | 2 |  |
| 4 | Thực nghiệm | 4.5 | 3.5 |  |
| **5** | **Kết luận** | 0.5 | 0.25 |  |
| Nhóm | Điểm nhóm | 0.5 | 0.25 |  |
| Tổng điểm | 8 | | |  |