ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ - VIỄN THÔNG

NHÓM 6

ỨNG DỤNG MÔ HÌNH LSTM VÀO PHÂN LOẠI VĂN BẢN TIẾNG VIỆT

ĐỒ ÁN MÔN HỌC NGÀNH KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG

NHẬP MÔN TRÍ TUỆ NHÂN TẠO

CHUYÊN NGÀNH: VIỄN THÔNG – MẠNG

NGƯỜI HƯỚNG DẪN KHOA HỌC:

ThS. NGUYỄN THÁI CÔNG NGHĨA

Thành phố Hồ Chí Minh, ngày 08 tháng 06 năm 2025

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA ĐIỆN TỬ - VIỄN THÔNG

ĐỒ ÁN MÔN HỌC NGÀNH KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG

NHẬP MÔN TRÍ TUỆ NHÂN TẠO

CHUYÊN NGÀNH: VIỄN THÔNG – MẠNG

ỨNG DỤNG MÔ HÌNH LSTM VÀO PHÂN LOẠI VĂN BẢN TIẾNG VIỆT

Nhóm 6

STT	Họ và tên	MSSV	Chức vụ
1	Nguyễn Huy Hoàng	21200091	Trưởng nhóm
2	Chu Quang Vinh	21200256	Thành viên
3	Hàng Hải Sơn	20200327	Thành viên

NHẬN XÉT

Tp. HCM, Ngày ... Tháng ... Năm ....

Giảng viên Nhập môn Trí tuệ Nhân tạo

Nguyễn Thái Công Nghĩa

LỜI CẢM ƠN

Đầu tiên, chúng em xin được gửi lời cảm ơn chân thành đến các thầy cô trong Khoa Điện tử – Viễn thông nói chung và Bộ môn Viễn thông – Mạng Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM nói riêng đã tận tình giảng dạy và giúp đỡ chúng em rất nhiều.

Báo cáo môn học với đề tài **Phân loại văn bản Tiếng Việt bằng mô hình LSTM** là kết quả của quá trình học tập, nghiên cứu của chúng em. Đặc biệt, chúng em xin được bày tỏ lòng biết ơn sâu sắc đến thầy **ThS. Nguyễn Thái Công Nghĩa** giảng viên bộ môn Viễn thông-Mạng, Khoa Điện tử - Viễn thông đã trực tiếp giảng dạy và tận tình chỉ bảo, đưa ra những kiến thức rất bổ ích nhưng cũng không thiếu phần thực tế để chúng em hoàn thành đồ án này. Tuy nhiên, do vốn kiến thức và kỹ năng còn hạn chế nên bài báo cáo này không tránh khỏi những sự thiếu sót trong quá trình thực hiện, vì vậy chúng em kính mong quý thầy cô góp ý để chúng em có điều kiện hoàn thiện hơn nữa kiến thức của chúng em.

Nhóm chúng em xin chân thành cảm ơn ạ!

TP. Hồ Chí Minh, ngày 08 tháng 06 năm 2025

Ký tên Trưởng nhóm

Nguyễn Huy Hoàng

MỤC LỤC

DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

CHỮ VIẾT TẮT		
AI	Artificial intelligence	Trí tuệ nhân tạo
TF-IDF	Term Frequency – Inverse Document Frequency	Tần suất xuất hiện của từ - Tần suất nghịch đảo của văn bản
SVD	Singular Value Decomposition	Phân tích giá trị đơn lẻ
LSTM	Long Short – Term Memory	Mạng nơ-ron hồi quy có bộ nhớ dài - ngắn hạn
TC	text/document classification/categorization	Phân loại văn bản/tài liệu
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
LLM	Large Language Model	Mô hình ngôn ngữ lớn
FNN	Feedforward Neural Network	Mạng nơ-ron truyền thẳng
TP	True Positive	Mô hình dự đoán đúng kết quả positive
TN	True Negative	Mô hình dự đoán đúng kết quả negative
FP	False Positive	Mô hình dự đoán sai kết quả positive
FN	False Negative	Mô hình dự đoán sai kết quả negative
TPR	True Positive Rate	Tỉ lệ tất cả các trường hợp dương thực tế

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

LỜI MỞ ĐẦU

Trong những thập kỷ gần đây, trí tuệ nhân tạo (Artificial Intelligence - AI) đã trở thành một trong những lĩnh vực công nghệ phát triển nhanh chóng và có ảnh hưởng sâu rộng đến nhiều mặt của đời sống xã hội. Từ các hệ thống nhận diện khuôn mặt, trợ lý ảo cho đến xe tự hành và y học chính xác, AI đang dần chứng minh vai trò then chốt trong cuộc cách mạng công nghiệp lần thứ tư.

Với mục tiêu tìm hiểu những nền tảng cơ bản và ứng dụng thực tiễn của trí tuệ nhân tạo, bài báo cáo này được thực hiện như một phần của học phần "Nhập môn Trí tuệ nhân tạo".

Bài báo cáo bao gồm: 5 chương.

Chương I: TỔNG QUAN:

Giới thiệu tổng quan về đề tài nghiên cứu, bao gồm lý do chọn đề tài, mục tiêu cần đạt được, đối tượng và phạm vi nghiên cứu cụ thể. Chương này cũng trình bày tóm tắt phương pháp được áp dụng để nghiên cứu đề tài.

Chương II. CƠ SỞ LÝ THUYẾT:

Trình bày các kiến thức nền tảng và lý thuyết liên quan đến lĩnh vực xử lý ngôn ngữ tự nhiên, bài toán phân loại văn bản. Đi sâu vào các kỹ thuật tiền xử lý văn bản tiếng Việt, phương pháp trích xuất đặc trưng TF-IDF, kỹ thuật giảm chiều SVD, và kiến trúc, nguyên lý hoạt động của mạng nơ-ron hồi quy LSTM. Các độ đo đánh giá mô hình cũng được giới thiệu trong chương này.

Chương III: DỮ LIỆU VÀ PHƯƠNG PHÁP THỰC NGHIỆM:

Mô tả chi tiết về bộ dữ liệu văn bản tiếng Việt được sử dụng trong đồ án, bao gồm nguồn gốc, đặc điểm và cách thức tổ chức. Trình bày cụ thể quy trình thực nghiệm, từ các bước tiền xử lý dữ liệu, cách trích xuất đặc trưng, đến việc thiết kế kiến trúc mô hình LSTM và các tham số trong quá trình huấn luyện mô hình.

Chương IV: KẾT QUẢ VÀ THẢO LUẬN:

Trình bày các kết quả thực nghiệm thu được sau quá trình huấn luyện và đánh giá mô hình. Các kết quả này bao gồm các độ đo hiệu suất như Accuracy, Precision, Recall, F1-score và phân tích chi tiết ma trận nhầm lẫn. Chương này cũng đưa ra những thảo luận, nhận xét và đánh giá về hiệu quả của mô hình, những ưu điểm, hạn chế và các yếu tố ảnh hưởng đến kết quả.

Chương V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN:

Tóm tắt lại những kết quả chính mà đồ án đã đạt được, đối chiếu với các mục tiêu đề ra ban đầu để đưa ra kết luận cuối cùng. Đồng thời, đề xuất những hướng nghiên cứu và phát triển tiếp theo có thể thực hiện để cải thiện mô hình hoặc mở rộng phạm vi ứng dụng của đề tài.

Chương I: Tổng quan

1.1. Đặt vấn đề

Thời đại số với những đột phá trong thu thập dữ liệu đã nhanh chóng tạo ra dữ liệu lớn, cho phép các chủ thể truyền thông dễ dàng nhận diện công chúng. Đây là cơ sở để cá nhân hoá đối tượng tiếp nhận thông tin truyền thông. Tuy nhiên, sự gia tăng nhanh chóng của lượng lớn dữ liệu văn bản trên internet, mạng xã hội, báo chí điện tử, tài liệu doanh nghiệp,… đang mang đến nhu cầu cấp thiết cho việc tự động tổ chức, tìm kiếm, và hiểu nội dung từ khối lượng văn bản khổng lồ này. Hơn nữa, với nhu cầu thực tế của người dùng, tìm kiếm thông tin với những chủ đề chỉ định là một thách thức thật sự. Từ đó, bài toán "tìm kiếm văn bản theo chủ đề" trở thành một giải pháp hợp lý cho nhu cầu trên.

1.2. Mục tiêu của đồ án

Phân loại văn bản (Text classification) là một bài toán phổ biến trong xử lý ngôn ngữ tự nhiên. Đây là bài toán thuộc nhóm học có giám sát trong học máy. Bài toán này yêu cầu dữ liệu cần có nhãn. Mô hình sẽ học từ dữ liệu có nhãn đó, sau đó được dùng để dự đoán nhãn cho các dữ liệu mới mà mô hình chưa gặp.

Chúng em quyết định chọn đề tài "Phân loại văn bản tiếng Việt theo chủ đề", với mục tiêu xây dựng một giải pháp có thể ứng dụng nhằm giải quyết bài toán tìm kiếm nói trên. Từ đó giảm thời gian và công sức của con người hơn việc tìm kiếm thủ công.

1.3. Đối tượng và phạm vi nghiên cứu

Trong khuôn khổ đồ án, đối tượng nghiên cứu chính của đề tài bao gồm:

- Dữ liệu: các văn bản tiếng Việt, được tổng hợp từ các trang báo điện tử, các diễn đàn thảo luận và các mẫu tin tức được thu thập từ các nguồn trực tuyến.
- Các thuật toán và kỹ thuật xử lý ngôn ngữ tự nhiên cho tiếng Việt:
    - Kỹ thuật tiền xử lý văn bản: Bao gồm các bước chuẩn hoá văn bản cơ bản (sử dụng hàm *simple_process* của thư viện *Gensim*), tách từ tiếng Việt (sử dụng thư viện *pyvi* với hàm *ViTokenizer.tokenize*), loại bỏ các ký tự đặc biệt và số, chuyển đổi văn bản thành chữ thường, và loại bỏ stopword tiếng Việt dựa trên một danh sách stopword được định sẵn.

  - Phương pháp biểu diễn văn bản (trích dẫn đặc trưng): Tập trung vào kỹ thuật Term Frequency – Inverse Document Frequency (TF-IDF) và phương pháp giảm chiều dữ liệu Singular Value Decomposition (SVD).
- Mô hình học sâu cho phân loại văn bản: Nghiên cứu và triển khai mô hình dạng Long Short – Term Memory (LSTM), một dạng của mạng nơ-ron hồi quy, để thực hiện nhiệm vụ phân loại văn bản đa chủ đề. Kiến trúc mô hình cụ thể bao gồm các lớp Input, Reshape, LSTM, Dense và Dropout.
- Các độ đo đánh giá hiệu suất mô hình: Sử dụng các độ đo phổ biến trong bài toán phân loại như Accuracy, Precision, Recall, F1-score và phân tích Confussion Matrix để đánh giá mức độ hiệu quả của mô hình.

Đồ án được thực hiện trong phạm vi cụ thể như sau:

- Về dữ liệu:
  - Nguồn và cấu trúc: Sử dụng bộ dữ liệu văn bản tiếng Việt đã được thu thập và gán nhãn trước, được tổ chức thành các thư mục con tương ứng với từng chủ đề. Bộ dữ liệu này bao gồm một tập huấn luyện với 50,374 mẫu và một tập kiểm thử với 33,769 mẫu.
  - Cấu trúc: Các dữ liệu được chia thành 10 nhãn với 10 chủ đề chính bao gồm: Chính trị Xã hội, Đời sống, Khoa học, Kinh doanh, Pháp luật, Sức khoẻ, Thế giới, Thể thao, Văn hoá, Vi tính.
- Về công cụ và môi trường:
  - Nghiên cứu được triển khai với ngôn ngữ Python.
  - Các thư viện mã nguồn mở được sử dụng bao gồm *scikit-learn* cho các tác vụ tiền xử lý và trích xuất đặc trưng (TF-IDF, SVD, LabelEncoder), *TensorFlow* (phiên bản 2.18.0) với *Keras API* (phiên bản 3.8.0) để xây dựng và huấn luyện mô hình LSTM, *pyvi* (phiên bản 0.1.1) và *gensim* (phiên bản 4.3.3) cho các tác vụ xử lý văn bản tiếng Việt chuyên biệt.
- Giới hạn của đồ án:

- Đồ án không thực hiện so sánh toàn diện với một loạt các phương pháp trích xuất đặc trưng hoặc các kiến trúc mô hình học sâu khác nhau do giới hạn về thời gian và phạm vi của một đồ án.
- Việc lựa chọn và tối ưu hoá các siêu tham số của mô hình (ví dụ như learning rate, số lượng lớp, số units trong từng lớp, tỉ lệ dropout) chủ yếu được dựa trên các giá trị tham khảo và thực nghiệm, chưa phải là một quá trình tìm kiếm tối ưu hoá siêu tham số một cách hệ thống và toàn diện.
- Đồ án tập trung vào bài toán phân loại đơn nhãn (mỗi văn bản hoặc một đoạn văn chỉ thuộc về một chủ đề duy nhất) và chưa được nghiên cứu để giải quyết cho các trường hợp phức tạp hơn như phân loại đa nhãn.
- Các yếu tố về tốc ưu hoá tốc độ dự đoán cho các ứng dụng thời gian thực cũng không phải là trọng tâm chính của đồ án này.

1.4. Phương pháp nghiên cứu

Phương pháp nghiên cứu của đồ án được thực hiện theo một quy trình có hệ thống, bao gồm các giai đoạn chính:

- Nghiên cứu lý thuyết và tổng quan tài liệu: Tiến hành tìm hiểu về lĩnh vực xử lý ngôn ngữ tự nhiên, đặc biệt là bài toán phân loại văn bản. Nghiên cứu các kỹ thuật xử lý văn bản phổ biến và hiệu quả cho tiếng Việt, các phương pháp trích xuất đặc trưng như TF-IDF, kỹ thuật giảm chiều dữ liệu SVD, và kiến trúc mạng nơ-ron hồi quy Long Short – Term Memory (LSTM) cùng các biến thể của nó. Đồng thời, tiến hành nghiên cứu các công trình nghiên cứu liên quan để có cái nhìn toàn diện vấn đề.
- Thu thập và chuẩn bị dữ liệu: Sử dụng bộ dữ liệu văn bản tiếng Việt đã được thu thập và gán nhãn trước đó, bao gồm các bài báo và các mẫu tin được chia thành 10 chủ đề khác nhau. Dữ liệu được tổ chức thành tập huấn luyện và tập kiểm thử.
- Tiền xử lý văn bản: Áp dụng các kỹ thuật tiền xử lý để làm sạch và chuẩn hoá dữ liệu theo văn bản thô. Bao gồm các bước:
  - Chuẩn hoá văn bản cơ bản từ hàm *simple_preprocess* từ thư viện *gensim*
  - Tách từ tiếng Việt chuyên biệt bằng thư viện *pyvi* với hàm *Vitokenize.tokenize.*

  - Loại bỏ các ký tự đặc biệt, số và chuyển đổi toàn bộ văn bản thành chữ thường (không phải chữ in hoa).
  - Loại bỏ các stop-word tiếng Việt dựa trên danh sách được định sẵn để giảm nhiễu và số chiều đặc trưng của dữ liệu.
- Trích xuất và lựa chọn các đặc trưng (feature): Chuyển đổi dữ liệu văn bản đã tiền xử lý thành dạng vector số mà mô hình máy học có thể hiểu được:
  - Áp dụng phương pháp Term Frequency – Inverse Document Frequency (TF-IDF) để tính trọng số cho từng từ trong mỗi văn bản, với giới hạn 10,000 đặc trưng quan trọng nhất thông qua *TfidVectorizer*.
  - Sử dụng Singular Value Decomposition (SVD), cụ thể là hàm *TruncatedSVD,* nhằm giảm chiều của ma trận TF-IDF xuống chỉ còn 500 thành phần, mục đích là để giữ lại những thông tin quan trọng nhất và loại bỏ nhiễu, đồng thời giảm tính toán cho mô hình học sâu.
- Xây dựng và huấn luyện mô hình: Mô hình được chọn để huấn luyện là Long Short – Term Memory (LSTM), được thực hiện theo các bước:
  - Thiết kế kiến trúc mô hình LSTM bao gồm một lớp *Input* nhận đầu vào là vector 500 chiều, một lớp *Reshape* để định dạng lại dữ liệu cho phù hợp với đầu vào của lớp *LSTM*, một lớp *LSTM* với 256 units, theo sau là các lớp *Dense* và *Dropout* xen kẽ để tăng khả năng học và chống quá khớp, và cuối cùng là một lớp Output với hàm kích hoạt Softmax để phân loại văn bản vào 10 chủ đề.
  - Mã hoá nhãn chủ đề dạng chữ thành dạng số bằng *LabelEncoder*.
  - Phân chia tập dữ liệu huấn luyện ban đầu thành tập huấn luyện thực tế (95%) và tập kiểm định (5%) để theo dõi hiệu suất của mô hình trong quá trình huấn luyện.
  - Tiến hành huấn luyện mô hình với thuật toán tối ưu adam, hàm sai số *sparse_categorical_crossentropy*, độ đo *accuracy*. Quá trình huấn luyện được thực hiện với kích thước batch là 512 và epochs là 50.
- Đánh giá hiệu suất của mô hình: Sau khi huấn luyện, mô hình được đánh giá trên tập kiểm thử (test) chưa từng gặp trước đó. Các độ đo được sử dụng để đánh giá

bao gồm accuracy, Precision, Recall, F1 – score, tính toán cho từng lớp và trung bình và phân tích Confusion Matrix để hiểu rõ hơn về khả năng phân loại của mô hình đối với từng chủ đề cụ thể cũng như các nhầm lẫn thường gặp.

- Lưu trữ mô hình và chuẩn bị cho việc tái sử dụng: Các thành phần quan trọng của quy trình, bao gồm *TfidVectorizer*, *TruncateSVD*, kiến trúc mô hình (*model.json*), trọng số mô hình (*model.weight.h5*) và danh sách các lớp đã mã hoá (*classes.npy*) lưu lại để có thể dễ dàng tải và tái sử dụng mô hình cho việc dự đoán trên dữ liệu mới mà không cần huấn luyện lại từ đầu.

Chương II. Cơ sở lý thuyết

Chương này sẽ trình bày một số cơ sở lý thuyết nền tảng về văn bản, phân loại văn bản, lý thuyết về từ tiếng Việt ứng dụng cho bài toán tách từ trong phân loại văn bản. Ngoài ra, trong chương cũng trình bày các cơ sở lý thuyết toán học về rút trích đặc trưng, chọn lựa đặc trưng, và cuối cùng là các mô hình phân loại văn bản phổ biến trên thế giới và các mô hình thích hợp vào bài toán tìm kiếm văn bản tiếng Việt theo chủ đề.

2.1. Lý thuyết ngôn ngữ cho bài toán tách từ tiếng Việt. [1]

2.1.1. Khái niệm về từ

Trong quá trình học tập và sử dụng ngôn ngữ trong đời sống hằng ngày, mỗi chúng ta đều quen thuộc với khái niệm về "từ". Nhưng để định nghĩa được chính xác "từ là gì?" hoàn toàn không phải là một vấn đề đơn giản. Trong ngành ngôn ngữ học, đã có hàng trăm định nghĩa được đưa ra, nhưng hầu như chưa có định nghĩa nào có thể bao quát hết mọi vấn đề liên quan đến khái niệm "từ". Theo công trình [2] của Đinh Điền, có một số khái niệm tiêu biểu sau đây về từ:

- Theo L.Bloomfield thì: "*từ là một hình thái tự do nhỏ nhất*".
- B.Golovin quan niệm: "*từ là đơn vị nhỏ nhất có nghĩa của ngôn ngữ, được vận dụng độc lập, tái hiện tự do trong lời nói để xây dựng nên câu*".
- Còn Solncev thì lại quan niệm: "*Từ là đơn vị ngôn ngữ có tính hai mặt : âm và nghĩa. Từ có khả năng độc lập về cú pháp khi sử dụng trong lời*"

Trong tiếng Việt, cũng có nhiều định nghĩa về từ như:

- Theo Trương Văn Trình và Nguyễn Hiến Lê thì: "*Từ là âm có nghĩa, dùng trong ngôn ngữ để diễn tả một ý đơn giản nhất, nghĩa là ý không thể phân tích ra được*".
- Nguyễn Kim Thản thì định nghĩa: "*Từ là đơn vị cơ bản của ngôn ngữ, có thể tách khỏi các đơn vị khác của lời nói để vận dụng một cách độc lập và là một khối hoàn chỉnh về ý nghĩa (từ vựng hay ngữ pháp) và cấu tạo*".

- Theo Hồ Lê, "*Từ là đơn vị ngữ ngôn có chức năng định danh phi liên kết hiện thực, hoặc chức năng mô phỏng tiếng động, có khả năng kết hợp tự do, có tính vững chắc về cấu tạo và tính nhất thể về ý nghĩa*".

2.1.2. Hình thái từ tiếng Việt

Như trình bày trong phần trên, có rất nhiều định nghĩa về từ nhưng các nhà ngôn ngữ học vẫn chưa thống nhất quyết định chọn theo lối định nghĩa nào. Điều này cũng xảy ra trong tiếng Việt của chúng ta. Do vậy, với mục đích phục vụ thuận tiện cho việc xử lý tự động ngôn ngữ bằng máy tính, nhưng vẫn phù hợp với các định nghĩa về từ trong ngôn ngữ học đại cương cũng như tính đặc thù của ngôn ngữ đơn lập như tiếng Việt.

2.1.2.1. Hình vị tiếng Việt

Đầu tiên, chúng em sử dụng quan niệm của công trình [2] như sau: tiếng là đơn vị cơ bản trong tiếng Việt vì nó có thể nhận diện tương đối dễ dàng bởi người bản ngữ cũng như nhận diện một cách tự động bởi máy tính. Xét về mặt kỹ thuật trên máy tính, ta cũng có thể thực hiện được các thao tác lưu trữ, xử lý, tìm kiếm và sắp xếp các tiếng một cách dễ dàng do số lượng cũng như chiều dài của các tiếng này là nhỏ[1].

Ngoài ra, tiếng còn được xem là "từ chính tả". Tuy nhiên, nếu xét trên các tiêu chí của ngôn ngữ học, thì tiếng không thể được xem là một từ thực sự. Thậm chí, tiếng cũng chưa hoàn toàn đủ tư cách để được xem là "hình vị thực sự" vì chưa thỏa tiêu chí về nội dung (phải có ý nghĩa hoàn chỉnh). Vì vậy, trong luận văn này, chúng em dựa theo quan điểm của Đinh Điền trong công trình [3] là xem tiếng chỉ là "hình vị tiếng Việt":

Hình vị tiếng Việt ở đây phải được hiểu là: bên cạnh khái niệm hình vị như trong ngôn ngữ học đại cương, còn phải xét đến yếu tố hình tố, là yếu tố thuần túy hình thức biểu hiện những kiểu quan hệ bên trong giữa các thành tố trong từ. Ta có thể gọi đây

là những "tha hình vị" hay "á hình vị". Như vậy, trong tiếng Việt sẽ có 3 loại hình vị như sau:

- **Hình vị gốc:** là những nguyên tố, đơn vị nhỏ nhất, có nghĩa, chúng có thể là hình vị thực (là những từ vựng) hay hình vị hư (ngữ pháp), chúng có thể đứng độc lập hay bị ràng buộc.
- **Tha hình vị:** vốn cũng là hình vị gốc, nhưng vì mối tương quan với các thành tố khác trong từ mà chúng biến đổi đi về âm, nghĩa,... Tha hình vị bao gồm:
  - **Tha hình vị láy nghĩa:** trong các từ ghép bội nghĩa, như: giá cả, hỏi han, tuổi tác,...; nhà cửa, yêu thương, ngược xuôi,...
  - **Tha hình vị láy âm:** chùm chím, đo đỏ, chúm chím,...; lẻ đẻ, đủng đỉnh,...
  - **Tha hình vị định tính:** là các yếu tố phụ để miêu tả thuộc tính, như: xanh lè, tối om, cười khẩy,...
  - **Tha hình vị tựa phụ tố:** là đơn vị hoạt động giống như những phụ tố (affix) trong các ngôn ngữ biến hình, như: giáo viên, hiện đại hoá, tân tổng thống,...
- **Á hình vị:** là những chiết đoạn ngữ âm được phân xuất một cách tiêu cực, thuần túy dựa vào hình thức, không rõ nghĩa, song có giá trị khu biệt, làm chức năng cấu tạo từ. Ví dụ: dưa hấu, dưa gang, bí ư, đậu nành, cà niễng,...

2.1.2.2. Từ tiếng Việt

Trong đồ án này, chúng em sử dụng định nghĩa từ theo công trình [2], "*từ được cấu tạo bởi những hình vị*". Theo công trình này, thì "*từ tiếng Việt được cấu tạo bởi những hình vị tiếng Việt*".

Từ tiếng Việt ở đây bao gồm*: từ đơn, từ ghép, từ láy* và *từ ngẫu hợp*.

Xuất phát từ nhu cầu xử lý tự động ngữ liệu tiếng Việt bằng máy tính, Đinh Điền đã đề nghị cách thức hình thức hoá các quan niệm về hình vị tiếng Việt và từ tiếng Việt nói trên trong công trình [3] như sau:

- Do "*hình vị tiếng Việt*" cũng chính là từ chính tả (từng chữ độc lập), nên việc hình thức hoá rất đơn giản, không cần đặt ra. Trong ngữ liệu tiếng Việt cũng như tiếng Anh, đơn vị cơ bản được lưu cũng chính là từ chính tả này. Tuy nhiên, nếu chỉ lưu trữ ở cấp độ hình vị như vậy, thì lượng thông tin trong kho ngữ liệu sẽ rất hạn chế và chúng ta sẽ không thể khai thác hiệu quả vốn có của nó được.
- Để lưu trữ thông tin về ranh giới từ tiếng Việt, chúng em sử dụng khái niệm từ từ điển học được trình bày trong công trình [3]. Từ từ điển học ở đây được định nghĩa là "*những đơn vị mà căn cứ vào đặc điểm ý nghĩa của nó phải xếp riêng trong từ điển và có đánh dấu đây là đơn vị từ của ngôn ngữ*". Việc chọn lựa những từ nào sẽ đưa vào từ điển là hoàn toàn do các nhà ngôn ngữ hay người xây dựng kho ngữ liệu quyết định, dựa theo quan điểm về từ đã nêu trên. Trong đồ án này chúng em sử dụng từ điển tiếng Việt của công trình [4] của GS Hoàng Phê.

Do có nhiều thuật ngữ về "*từ*" khác nhau (từ chính tả, từ từ điển học ...), vì vậy, từ đây trở về sau, thuật ngữ "*từ*" được sử dụng trong đồ án được quy ước là để chỉ "*từ từ điển*".

2.2. Cơ sở lý thuyết về văn bản và phân loại văn bản

2.2.1. Khái niệm văn bản

Theo **Wikipedia** [5] thì văn bản (text, document) có 1 số khái niệm sau:

- Trong ngôn ngữ (language), văn bản là 1 thuật ngữ rộng nói về 1 thứ gì đó mà chứa các từ ngữ diễn đạt 1 sự việc.
- Trong ngôn ngữ học (linguistics), văn bản là 1 hoạt động giao tiếp, thi hành 7 nguyên tắc cấu thành cơ bản và 3 nguyên tắc điều khiển của văn bản học. Cả tiếng nói, ngôn ngữ viết hay ngôn ngữ thông thường đều có thể xem như văn bản trong ngôn ngữ học.
- Trong lý thuyết văn học, văn bản là 1 đối tượng (object) được nghiên cứu, dù nó là 1 cuốn tiểu thuyết, 1 bài thơ, 1 vở phim, 1 mẫu quảng cáo hay bất cứ thứ gì có thành phần thuộc về ký hiệu. Cách dùng rộng rãi thuật ngữ này được bắt

nguồn từ sự xuất hiện của ký hiệu những năm 1960 và được củng cố vững chắc bằng những nghiên cứu văn hóa sau đó trong những năm 1980.

- Trong truyền thông các thiết bị di động, văn bản (hay tin nhắn văn bản) là 1 đoạn tin nhắn số hóa ngắn giữa những thiết bị.
- Trong tin học, văn bản liên hệ đến dữ liệu ký tự (character data), hay đến 1 trong những thành phần của chương trình trong bộ nhớ.
- Trong học thuật, văn bản thường được dùng như là 1 hình thức viết tắt của sách giáo khoa.

2.2.2. Khái niệm phân lớp

Phân lớp (classification, categorization) là 1 tiến trình trong đó các đối tượng và sự việc được nhận ra, được phân biệt và hiểu được. Sự phân lớp hàm ý rằng các đối tượng được nhóm thành các bộ phận loại, thường thì phục vụ cho 1 vài mục đích đặc biệt. Nói 1 cách cơ bản, 1 bộ phận loại mô tả mối quan hệ giữa các chủ thể và đối tượng tri thức. Có rất nhiều cách tiếp cận phân lớp, nhưng nói chung có 2 cách cơ bản nhất:

- Phân lớp học có giám sát (supervised learning)
- Phân lớp học không có giám sát (unsupervised learning).

2.2.3. Khái niệm phân loại văn bản

Phân loại văn bản (text/document classification/categorization - TC) là 1 quá trình gán nhãn cho những tài liệu được diễn đạt trong ngôn ngữ tự nhiên vào 1 trong những bộ phân lớp (category, class), các bộ phân lớp này đã được định nghĩa trước [6]

Nói 1 cách toán học, phân loại văn bản là 1 quá trình xấp xỉ hàm mục tiêu chưa biết $\Psi: D \times C \rightarrow \{T, F\}$ bằng trung gian của hàm $\Phi: D \times C \rightarrow \{T, F\}$, hàm này được gọi là hàm phân lớp. Trong đó:

- $C = \{c_1, \dots, c_m\}$ là tập các nhãn phân lớp có kích thước cố định đã được định nghĩa trước.
- D là phạm vi các tài liệu.
- Giá trị của T $(True)$ được gán cho (d_j, c_i) chỉ định rằng 1 quyết định tài liệu d_j thuộc về lớp c_i

- Giá trị của F (False) cho biết quyết định d_j không thuộc về lớp c_i

Một số lưu ý:

- Chúng ta thường có giả sử rằng các bộ phận loại chỉ là những nhãn ký hiệu. Không có 1 tri thức bổ sung nào từ ý nghĩa (meaning) của các bộ phận loại có thể giúp xây dựng bộ phận lớp.
- Các thuộc tính của các tài liệu liên quan đến bộ phận lớp nên được nhận ra dựa trên bản là nội dung của tài liệu.
- Đưa ra nội dung của 1 tài liệu mang tính chủ quan, điều này có nghĩa tài liệu trong bộ phận loại này không được quyết định 1 cách chắc chắn. Tùy vào từng ứng dụng cụ thể mà phân loại văn bản có thể chia thành: [7]

2.2.3.1 Phân loại văn bản đơn nhãn và đa nhãn

Ràng buộc khác biệt ở đây có lẽ bị phụ thuộc vào nhiệm vụ phân loại (TC task), vào ứng dụng cụ thể . Chúng ta có thể lấy ví dụ như sau: cho trước 1 số nguyên k (hoặc lớn hơn k hoặc nhỏ hơn k), k thành phần của tập C (tập các loại) được gán cho mỗi tài liệu $d_j \in D$.

- Trường hợp chỉ có chính xác một phân lớp (category) được gán cho tài liệu $d_j \in D$ được gọi là phân loại nhãn đơn (single – label, nonoverlapping category).
- Trường hợp có 1 số lượng nhãn (từ 0 cho đến $|C|$) được gán cho tài liệu $d_j \in D$) được gọi là phân loại đa nhãn (multi – label, overlapping category).
- Trường hợp đặc biệt của phân loại nhãn đơn là phân loại nhị phân trong đó mỗi tài liệu $d_j \in D$ có thể được gán cho bộ phận loại c_i hay không thuộc bộ phận loại c_i.

Trên quan điểm lý thuyết, trường hợp phân loại đơn nhãn (nhị phân) tổng quát hơn trường hợp đa nhãn. 1 thuật toán cho phân lớp đơn nhãn cũng có thể áp dụng cho phân lớp đa nhãn, chỉ đơn giản là chúng ta biến đổi vấn đề phân lớp đa nhãn trên tập $\{c_1, \dots, |C|\}$ thành $|C|$ vấn đề phân lớp đơn nhãn độc lập với nhau. Tuy nhiên, điều

ngược lại là không đúng, 1 thuật toán cho phân lớp đa nhãn không thể áp dụng cho phân lớp đơn nhãn (cũng như phân lớp nhị phân).

2.2.3.2. Phân loại văn bản phụ thuộc lớp/loại văn bản so với phụ thuộc tài liệu

Có rất nhiều cách khác nhau để sử dụng bộ phân loại văn bản (text classifier). Cho trước tài liệu $d_j \in D$, chúng ta muốn tìm tất cả các lớp $c_i \in C$ mà tài liệu thuộc vào, cách này được gọi là phân loại dựa vào tài liệu (document-pivoted categorization-DPC). Ngược lại, nếu cho trước $c_i \in C$, chúng ta muốn tìm tất cả các tài liệu $d_j \in D$ mà thuộc vào nó, cách này được gọi là phân loại dựa vào lớp (loại) văn bản (category-pivoted categorization-CPC). Sự khác biệt này thể hiện rõ ở thực tế hơn là ở khái niệm trừu tượng.

DPC thích hợp hơn khi mà các tài liệu sẵn có ở những thời điểm khác nhau ví dụ như bài toán lọc e-mail. Còn CPC thích hợp hơn khi 1 tài liệu $c|C|+1$ được thêm vào tập có sẵn $C = \{c_1, \dots, |C|\}$ sau khi các tài liệu đã được phân lớp dưới C lớp và các tài liệu này cần được xem xét phân lớp lại dưới $c|C|+1$ lớp. Trên thực tế DPC được dùng nhiều hơn CPC.

2.2.3.3. Phân loại văn bản "cứng" so với "mềm"

Trong khi việc tự động hoàn toàn của quá trình phân lớp cần 1 quyết định *T* hay *F* cho mỗi cặp (d_j, c_i) thì việc tự động từng phần của tiến trình có lẽ lại cần những nhu cầu khác nhau.

- Phân loại văn bản "cứng" (hard TC) tức là cung cấp 1 giá trị trong $\{T, F\}$ cho biết d_j có hay không có nằm trong c_i. Điều này rất hữu ích cho các ứng dụng phân lớp tự động (autonomous) [6].
- Phân loại văn bản mềm (soft TC) tức là cung cấp 1 giá trị trong [0,1] cho biết mức độ tin cậy của hệ thống khi quyết định sự phụ thuộc của d_j vào trong c_i. Điều này lại phù hợp hơn với các ứng dụng phân lớp tương tác (interactive) [6].

2.2.3.4. Các ứng dụng của phân loại văn bản. [7]

Phân loại văn bản là bài toán nền tảng trong lĩnh vực truy hồi thông tin (information retrieval) có liên quan 1 phần đến Xử lý ngôn ngữ tự nhiên (Natural Language Processing-NLP). Phân loại văn bản là bài toán ứng dụng rất nhiều trong lĩnh vực xử lý ngôn ngữ hiện nay, ví dụ như: search engines, hệ thống lọc Spam mail, hệ thống phân loại để phục vụ cho việc lưu trữ và tìm kiếm... Ngoài ra, phân loại văn bản kết hợp với một số bài toán khác là cơ sở cho một số ứng dụng như: phân loại giọng nói bằng cách kết hợp giữa nhận dạng giọng nói và phân loại văn bản, phân loại tài liệu số (multimedia) thông qua phân tích chú thích văn bản, định danh tác giả (author identification) cho dạng văn bản tiểu thuyết của những tác giả chưa biết, nhận dạng ngôn ngữ (language identification) của những văn bản chưa biết loại ngôn ngữ, định danh tự động thể loại văn bản (text genre), và chấm điểm bài luận tự động (automated essay grading), ...

Chương III: Dữ liệu và Phương pháp thực nghiệm

Nội dung của chương này trình bày chi tiết về bộ dữ liệu được sử dụng trong đồ án, quy trình các bước tiền xử lý dữ liệu văn bản tiếng Việt, phương pháp trích xuất và lựa chọn đặc trưng, kiến trúc cụ thể của mô hình học sâu Long Short-Term Memory (LSTM) đã được xây dựng, cùng với quy trình huấn luyện và các thiết lập thực nghiệm để giải quyết bài toán phân loại văn bản theo chủ đề.

3.1. Mô tả dữ liệu.

3.1.1. Nguồn thu thập dữ liệu và đặc điểm của dữ liệu

Đồ án sử dụng bộ dữ liệu văn bản tiếng Việt đã được thu thập và gán nhãn trước [8]. Dữ liệu bao gồm các bài báo điện tử, các diễn đàn thảo luận và các mẫu tin tức được thu thập từ các nguồn trực tuyến. Tất cả các văn bản trong bộ dữ liệu đều là tiếng Việt.

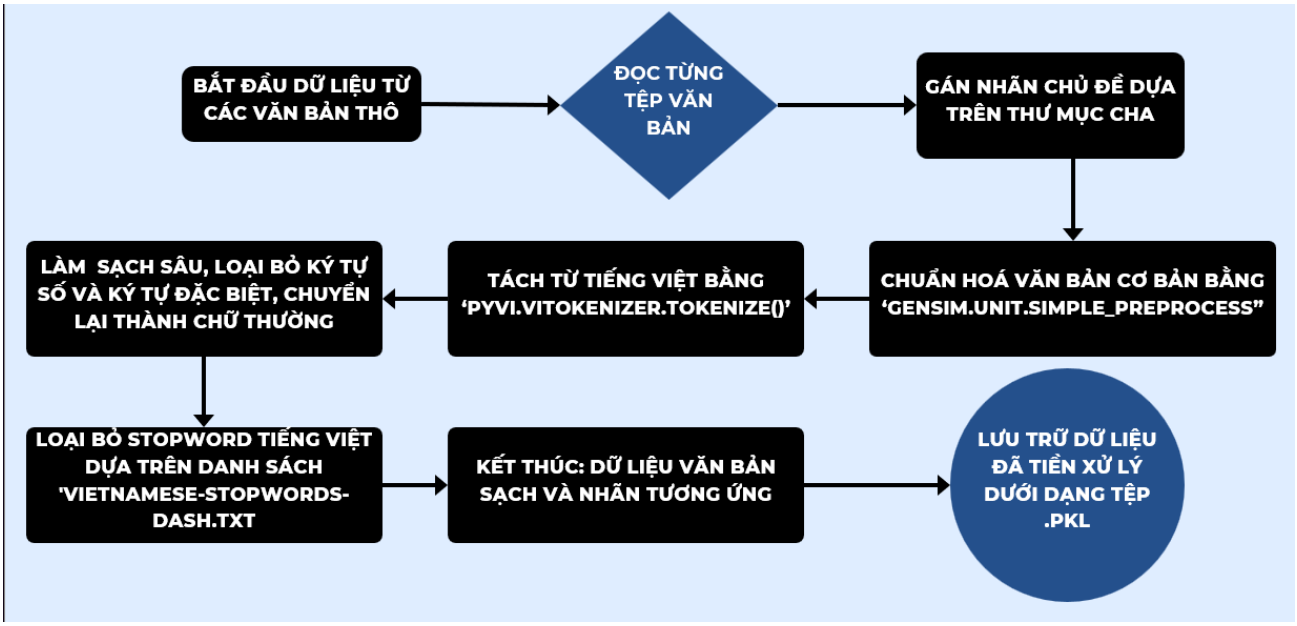
3.1.2. Cấu trúc của bộ dữ liệu

Dữ liệu được tổ chức thành các thư mục con, trong đó tên của mỗi thư mục con đại diện cho một chủ đề cụ thể.

Bộ dữ liệu được chia sẵn thành hai tập chính: tập huấn luyện (*Train_Full*) và tập kiểm thử (*Test_Full*). Tập huấn luyện bao gồm 50,374 mẫu văn bản và tập kiểm thử bao gồm 33,769 mẫu văn bản.

3.2. Các kỹ thuật Tiền xử lý văn bản tiếng Việt

Tiền xử lý văn bản trước khi đưa vào huấn luyện là một bước cực kỳ quan trọng trong các bài toán NLP, đặc biệt với tiếng Việt do tính phức tạp và đặc thù của ngôn ngữ. Mục tiêu của tiền xử lý là làm sạch dữ liệu, giảm nhiễu, và chuẩn hóa văn bản để cải thiện hiệu suất của các bước tiếp theo như trích xuất đặc trưng và xây dựng mô hình.



Hình 1. Quy trình tiền xử lý văn bản

Trong đồ án này, các kỹ thuật tiền xử lý văn bản tiếng Việt chính được áp dụng bao gồm:

3.2.1. Chuẩn hóa văn bản cơ bản

Trước khi thực hiện các bước chuyên sâu hơn, văn bản thường được xử lý sơ bộ để loại bỏ các yếu tố không cần thiết. Trong đồ án, *gensim.utils.simple_preprocess* được sử dụng để chuyển đổi văn bản thành chữ thường (lowercase), tách các từ dựa trên khoảng trắng và các ký tự không phải chữ cái, đồng thời loại bỏ các dấu cơ bản.

Ngoài ra, các ký tự đặc biệt khác và số cũng được loại bỏ bằng cách sử dụng các biểu thức chính quy hoặc phương thức *strip* như trong file notebook.

3.2.2. Tách từ tiếng Việt (Word Segmentation/Tokenization)

Tiếng Việt có 4 đặc trưng cơ bản đó là đơn tiết, không biến hình; sử dụng hư từ; sử dụng trật tự từ; sử dụng trọng âm và ngữ điệu, trong đó ranh giới giữa các từ không phải lúc nào cũng là khoảng trắng (ví dụ: "học sinh", "sinh học"). Do đó, việc tách từ chính xác là rất cần thiết để hiểu đúng ngữ nghĩa của câu.

Đồ án sử dụng thư viện *pyvi*, cụ thể là hàm *ViTokenizer.tokenize()*. Công cụ này giúp tách các âm tiết thành các từ đơn hoặc từ ghép có nghĩa trong tiếng Việt, ví dụ, "học_sinh"

thay vì "học" và "sinh" riêng lẻ. Kết quả của *ViTokenizer.tokenize()* là một chuỗi với các từ được nối với nhau bằng dấu gạch dưới ("_").

3.2.3. Loại bỏ Stopwords

Stopwords là những từ xuất hiện rất thường xuyên trong ngôn ngữ nhưng thường không mang nhiều ý nghĩa riêng biệt để phân loại văn bản (ví dụ: "*là*", "*và*", "*của*", "*thì*", "*các*", "*một*",...).

Việc loại bỏ *stopwords* giúp giảm số chiều của không gian đặc trưng, giảm nhiễu và giúp mô hình tập trung vào những từ mang thông tin quan trọng hơn.

Trong đồ án, một danh sách các stopwords tiếng Việt được định nghĩa sẵn trong tệp *vietnamese-stopwords-dash.txt* được sử dụng để lọc bỏ các từ này ra khỏi văn bản.

Sau khi thực hiện các bước tiền xử lý, dữ liệu sẽ trở nên đồng nhất và sẵn sàng cho giai đoạn trích xuất các đặc trưng của dữ liệu.

3.3. Các phương pháp Trích xuất đặc trưng (Feature Extraction)

Sau khi tiền xử lý, văn bản cần được chuyển đổi thành dạng biểu diễn số học (thường là vector) mà các thuật toán có thể hiểu và xử lý. Đây được gọi là quá trình trích xuất các đặc trưng của dữ liệu.

Trong đồ án này, hai phương pháp chính được sử dụng để trích xuất và lựa chọn đặc trưng:

3.3.1. TF – IDF (Term Frequency – Inverse Document Frequency)

TF – IDF là một thống kê số học nhằm phản ánh tầm quan trọng của một từ đối với một văn bản trong một tập hợp hay một ngữ liệu văn bản. TF – IDF thường dùng dưới dạng là một trọng số trong tìm kiếm truy xuất thông tin, khai thác văn bản, và mô hình hóa người dùng.

- Term Frequency (TF): Đo lường tần suất xuất hiện của một từ trong cùng một văn bản:

$$TF\ (t,d) = \frac{f(t,d)}{\max\{f(w,d);\ w \in d\}}$$

Với $f(t,d)$ là số lần xuất hiện từ *t* trong văn bản *d.*

$\max\{f(w,d);\ w \in d\}$ là số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản.

- Inverse Document Frequency (IDF): Tần số nghịch của một từ trong tập văn bản:

$$IDF(t,D) = \log\frac{|D|}{|\{d \in D : t \in d\}|}$$

Với $|D|$ là tổng số văn bản trong tập D.

$|\{d \in D : t \in d\}|$ là số văn bản chứa từ t. Nếu từ đó không xuất hiện ở bất cứ văn bản nào trong tập dữ liệu thì mẫu số sẽ bằng 0, dẫn đến phép chia không hợp lệ. Do đó, người ta thường thay mẫu số bằng

$$1 + |\{d \in D : t \in d\}|$$

Giá trị TF – IDF:

$$tfidf(t,d,D) = tf(t,d) \times idf\ (t,D)$$

Những từ có giá trị TF – IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít hơn trong các văn bản khác. Điều này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

Trong đồ án, thư viện *scikit – learn* với lớp *TfidVectorizer* được sử dụng để tính toán ma trận TF – IDF cho toàn bộ tập dữ liệu văn bản. Cùng với đó, tham số *max_features=10000* được thiết lập, điều này nghĩa là mô hình chỉ giữ lại 10000 từ có điểm TF – IDF cao nhất trong toàn bộ tập dữ liệu để làm đặc trưng.

3.3.2. Giảm chiều dữ liệu với SVD (Singular Value Decomposition – Phân rã giá trị suy biến)

SVD là là phép phân tích một ma trận thực hoặc phức thành một phép quay, sau đó là phép thay đổi tỉ lệ rồi đến phép quay khác. Nó khái quát hóa phép phân tích riêng của một ma trận chuẩn vuông với một cơ sở riêng chuẩn trực giao thành bất kỳ $m \times n$ ma trận.

Cho ma trận A là một ma trận cấp $m \times n$, thuật toán SVD phân tích ma trận A thành dạng:

$$A = U \sum V^T$$

Ở đây, Σ là một ma trận đường chéo cấp $m \times n$ với các giá trị trên đường chéo là không âm và được sắp xếp theo thứ tự giảm dần:

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_r \geq 0 = 0 = \cdots = 0$$

Khi đó, ma trận A xấp xỉ bằng tổng của k ma trận có hạng bằng 1 (với k < n).

$$A \approx A_k = U_k \Sigma_k (V_k)^T = \Sigma_{i=1}^{k} \sigma_i u_i v_i^T$$

Trong đồ án, sau khi có ma trận TF – IDF với 10000 đặc trưng, ma trận này vẫn còn thưa và có số chiều lớn. *TruncatedSVD* từ *scikit – learn* được sử dụng để giảm số chiều của ma trận TF – IDF xuống còn 500 thành phần. Điều này giúp tạo ra một biểu diễn đặc trưng dày đặc hơn, gọn hơn, có thể nắm bắt được các mối quan hệ ngữ nghĩa tiềm ẩn giữa các từ và văn bản. Việc giảm chiều này cũng giúp giảm tải tính toán cho mô hình LSTM và có thể giúp mô hình tổng quát hoá tốt hơn, tránh việc bị *Overfitting* (quá khớp).

Kết quả của quá trình này là các văn bản được xử lý và biểu diễn bằng một vector 500 chiều, sẵn sàng được đưa vào mô hình học sâu để huấn luyện.

3.4. Mô hình Học máy/Học sâu được sử dụng.

Để giải quyết bài toán phân loại tiếng Việt theo chủ đề, chúng em tập trung vào việc xây dựng một mô hình học sâu dựa trên kiến trúc Mạng Nơ – ron Hồi quy LSTM.

3.4.1. Giới thiệu về mạng Nơ – ron Hồi quy

Mạng nơ-ron hồi quy (RNN) là một mô hình học sâu được đào tạo để xử lý và chuyển đổi đầu vào dữ liệu tuần tự thành đầu ra dữ liệu tuần tự cụ thể. Dữ liệu tuần tự là dữ liệu, chẳng hạn như từ, câu hoặc dữ liệu chuỗi thời gian, trong đó các thành phần tuần tự tương quan với nhau dựa trên ngữ nghĩa phức tạp và quy tắc cú pháp.

RNN là một hệ thống phần mềm gồm nhiều thành phần được kết nối với nhau theo cách con người thực hiện chuyển đổi dữ liệu tuần tự, chẳng hạn như dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác. Phần lớn RNN đang được thay thế bằng trí tuệ nhân tạo (AI) dựa

trên công cụ biến đổi và các mô hình ngôn ngữ lớn (LLM), hiệu quả hơn nhiều trong việc xử lý dữ liệu tuần tự.

3.4.2. Mạng Long Short Term – Memory (LSTM)

Long Short – Term Memory là một dạng đặc biệt của mạng nơ – ron hồi quy (RNN), được trang bị các cơ chế cổng (gating mechanisms). Không giống như các mạng thần kinh truyền thẳng (FNN) tiêu chuẩn, LSTM có chứa các kết nối phản hồi. Mạng không chỉ xử lý các điểm dữ liệu đơn lẻ mà còn xử lý toàn bộ chuỗi dữ liệu. Cơ chế cổng của mạng LSTM có cấu trúc như sau:

- **Cell state (bộ nhớ dài hạn):** hoạt động như một băng chuyền chuyên chở thông tin xuyên suốt chuỗi thời gian.
- **3 cổng điều khiển và cơ chế hoạt động:**
  - **Cổng quên (Forget Gate):** Quyết định phần nào của cell state cũ (C_{t-1}) cần bị loại bỏ, phần nào sẽ được giữ lại.

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

    Trong đó: σ là hàm sigmoid

    Hàm sigmoid cho đầu ra giá trị từ [0,1]:

    1 = "giữ nguyên", 0 = "loại bỏ"

  - **Đầu vào (Input Gate):**

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

    Input Gate (i_t): Lọc thông tin mới từ đầu vào (x_t) để lưu trữ vào cell state

    Candidate Cell State ($\tilde{C}_t$): Tạo ra giá trị đề xuất cập nhật cho cell state, được "nén" trong khoảng $[-1, 1]$ bởi hàm *tanh.*
  - **Cell State mới:**

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

Phần $f_t \odot C_{t-1}$ sẽ "giữ lại" thông tin cũ theo tỉ lệ do Fogget Gate quyết định.

Phần $i_t \odot \tilde{C}$ bổ sung thông tin mới, đảm bảo ra rằng chỉ dữ liệu có giá trị mới được lưu giữ.

- **Đầu ra (Output Gate):**

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

$$h_t = o_t \odot tanh(C_t)$$

Output Gate (o_t): Quyết định phần nào của cell state sẽ được "trích xuất" ra làm hidden state.

Hidden State (h_t): Kết quả cuối cùng được đưa đi làm đầu ra cho bước thời gian hiện tại, sau khi cell state được "đi qua" hàm tanh.

3.4.3. Kiến trúc mô hình LSTM trong đồ án

Kiến trúc mô hình được định nghĩa bởi hàm lstm_model():

- **Lớp Input:** Nhận đầu vào là vector đặc trưng có 500 chiều (kết quả từ TF-IDF + SVD).

```
input_layer = Input(shape=(500,))
```

- **Lớp Reshape:** Đầu vào 500 chiều được định hình lại thành (1, 500) vì các lớp LSTM trong Keras thường mong đợi đầu vào có dạng 3D (batch_size, timesteps, features). Ở đây, mỗi văn bản được coi là một chuỗi có 1 "bước thời gian" (timestep) và 500 "đặc trưng" (features) tại bước thời gian đó.

```
layer = Reshape((1, 500))(input_layer)
```

- **Lớp LSTM:** Một lớp LSTM với 256 units (nơ-ron). Hàm kích hoạt *activation='relu'* được sử dụng. Cùng với đó, *dropout=0.65* và *recurrent_dropout=0.3* được áp dụng để giảm thiểu hiện tượng *overfitting* bằng cách ngẫu nhiên bỏ qua một số kết nối trong quá trình huấn luyện.

```
layer = LSTM(256, activation='relu', dropout=0.65, recurrent_dropout=0.3)(layer)
```

- **Các Lớp Dense và Lớp Dropout:** Sau lớp LSTM, một chuỗi các lớp Dense (fully connected) với hàm kích hoạt relu và các lớp Dropout được thêm vào để học các biểu diễn phức tạp hơn và tiếp tục chống *overfitting*.

```
layer = Dense(512, activation='relu')(layer)
layer = Dropout(0.5)(layer)
layer = Dense(512, activation='relu')(layer)
layer = Dense(256, activation='relu')(layer)
layer = Dropout(0.5)(layer)
layer = Dense(256, activation='relu')(layer)
layer = Dense(128, activation='relu')(layer)
layer = Dropout(0.5)(layer)
layer = Dense(128, activation='relu')(layer)
```

- **Lớp Output**: Một lớp Dense với 10 units tương ứng với 10 chủ đề cần phân loại và hàm kích hoạt *softmax*. Hàm *softmax* sẽ chuyển đổi *output* của lớp Dense thành một phân phối xác suất trên 10 lớp, trong đó tổng các xác suất bằng 1. Lớp có xác suất cao nhất sẽ được coi là dự đoán của mô hình.

```
output_layer = Dense(10, activation='softmax')(layer)
```

- **Tiến hành biên dịch chương trình:** Mô hình được biên dịch với thuật toán tối ưu *adam*, hàm sai số *sparse_categorical_crossentropy* (vì nhãn được mã hóa dưới dạng số nguyên và đây là bài toán phân loại đa lớp), và độ đo *accuracy* để theo dõi trong quá trình huấn luyện.

Kiến trúc mô hình sử dụng nhiều lớp Dense và Dropout cho việc học các mẫu phức tạp và kiểm soát overfitting. Đồng thời, việc kết hợp SVD để giảm chiều và sau đó là một mạng LSTM sâu (bao gồm cả phần Dense), là một cách tiếp cận để xử lý dữ liệu văn bản đã được trích xuất đặc trưng.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 500)	0
reshape (Reshape)	(None, 1, 500)	0
lstm (LSTM)	(None, 256)	775,168
dense (Dense)	(None, 512)	131,584
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262,656
dense_2 (Dense)	(None, 256)	131,328
dropout_1 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 256)	65,792
dense_4 (Dense)	(None, 128)	32,896
dropout_2 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 128)	16,512
dense_6 (Dense)	(None, 10)	1,290

Total params: 1,417,226 (5.41 MB)
Trainable params: 1,417,226 (5.41 MB)
Non-trainable params: 0 (0.00 B)

Hình 2. Kiến trúc mô hình LSTM được xây dựng trong đồ án

3.5. Quy trình huấn luyện mô hình

Phân chia dữ liệu huấn luyện và kiểm định: Tập dữ liệu huấn luyện và nhãn đã mã hóa số tương ứng được phân chia thành tập huấn luyện thực tế và tập kiểm định theo tỷ lệ 95% cho huấn luyện và 5% cho kiểm định. Thông số *random_state=2019* được sử dụng.

Huấn luyện mô hình: Mô hình được huấn luyện bằng phương thức *fit()* với kích thước batch (*batch_size*) là 512. Qua những thử nghiệm thực tế dựa trên việc quan sát đồ thị loss/accuracy, số epochs được đặt là 50 để tránh hiện tượng overfitting.

Lưu mô hình: Sau khi huấn luyện, kiến trúc mô hình được lưu dưới dạng tệp JSON (*model.json*) và trọng số mô hình được lưu dưới dạng tệp HDF5 (*model.weights.h5*).

3.6. Đánh giá mô hình (Evaluation Metrics).

Để đánh giá chất lượng và hiệu suất của mô hình phân loại văn bản đã xây dựng, nhóm em sử dụng các chỉ số đo lường (metrics) giúp theo dõi và đánh giá mức độ chính xác của mô hình, khả năng của nó trong việc xác định đúng các lớp và những loại lỗi mà nó mắc phải. Các chỉ số đo lường được sử dụng trong đồ án bao gồm:

3.6.1. Accuracy (Độ chính xác tổng thể):

Accuracy là số liệu đo lường tần suất mô hình học máy dự đoán đúng kết quả.

$$Accuracy = \frac{\text{Correct Predictions}}{\text{All Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

trong đó: TP: True Positives,

TN: True Negatives,

FP: False Positives,

FN: False Negatives, tính trên toàn bộ các lớp.

Trong đồ án, Accuracy được theo dõi trên tập huấn luyện, tập kiểm định trong quá trình huấn luyện và được tính toán trên tập kiểm thử sau khi huấn luyện xong.

3.6.2. Precision

Precision là tỉ lệ phần trăm tất cả các kết quả phân loại dương tính của mô hình thực sự là dương tính. Giá trị này được xác định theo toán học như sau:

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

Precision được sử dụng trong đồ án cho biết trong số những mẫu mà mô hình dự đoán là thuộc lớp *i*, có bao nhiêu phần trăm thực sự thuộc lớp *i*. Precision cao cho thấy mô hình ít khi dự đoán nhầm một mẫu không thuộc lớp i thành lớp *i*.

3.6.3. Recall (hoặc True Positive Rate)

True Positive Rate hoặc tỉ lệ tất cả các trường hợp dương thực tế được phân loại chính xác là dương tính, còn được gọi là Recall. Giá trị này được định nghĩa về mặt toán học như sau:

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Giá trị Recall trong đồ án cho biết trong số những mẫu thực sự thuộc lớp i, mô hình đã dự đoán đúng được bao nhiêu phần trăm. Recall cao cho thấy mô hình bỏ sót ít các mẫu thực sự thuộc lớp *i*.

3.6.4. F1-score

F1-score là một độ đo cân bằng giữa Precision và Recall, đặc biệt hữu ích khi có sự mất cân bằng giữa các lớp hoặc khi cả FP và FN đều quan trọng. Công thức của F1-score cho lớp thứ *i* được thể hiện như sau:

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i}$$

3.6.5. Confussion Matrix

Confussion Matrix là một ma trận tổng hợp kết quả dự đoán của mô hình phân loại. Các hàng của ma trận thường biểu diễn các lớp thực tế, và các cột biểu diễn các lớp được dự đoán bởi mô hình hoặc ngược lại. Ma trận này chia các dự đoán thành bốn loại:

- True Positive (TP): Mô hình dự đoán đúng kết quả positive, tức là kết quả thực tế là positive.
- True Negative (TN): Mô hình dự đoán đúng kết quả negative, tức là kết quả thực tế là negative.
- False Positive (FP): Mô hình dự đoán sai kết quả positive, tức là kết quả thực tế là negative. Nó còn được gọi là lỗi loại I.
- False Negative (FN): Mô hình dự đoán sai kết quả negative tức là kết quả thực tế là positive. Nó cũng được gọi là lỗi loại II.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Confussion Matrix cung cấp một cái nhìn chi tiết về hiệu suất của mô hình trên từng lớp, giúp xác định những lớp nào mô hình phân loại tốt, những lớp nào hay bị nhầm lẫn với nhau, và loại nhầm lẫn. Đây là công cụ trực quan và mạnh mẽ để phân tích lỗi.

Trong đồ án, việc sử dụng *sklearn.metrics.classification_report* sẽ cung cấp một bảng tóm tắt Precision, Recall, F1-score cho từng lớp cũng như các giá trị trung bình, bên cạnh việc

trực quan hóa ma trận nhầm lẫn để có cái nhìn sâu sắc về hiệu suất của mô hình LSTM đã xây dựng.

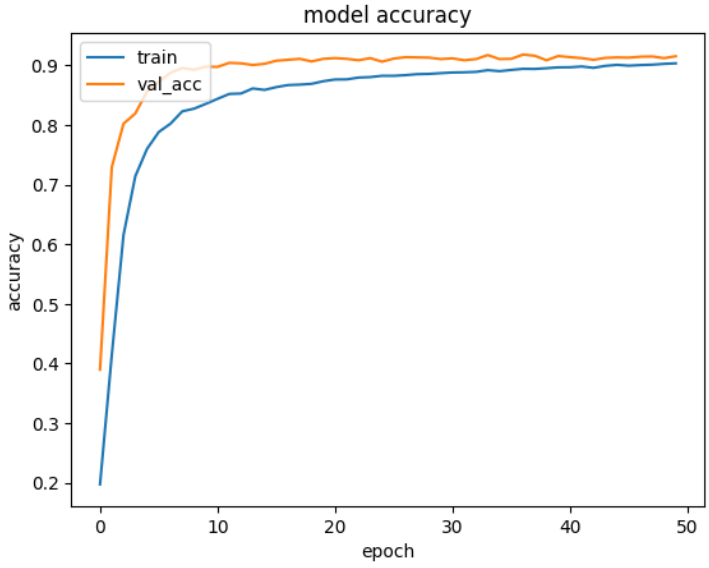
Chương IV: Kết quả và thảo luận

Chương này trình bày và phân tích chi tiết các kết quả thực nghiệm thu được từ quá trình huấn luyện và đánh giá mô hình Long Short-Term Memory (LSTM) cho bài toán phân loại văn bản tiếng Việt theo 10 chủ đề. Các kết quả bao gồm hiệu suất của mô hình trên các tập dữ liệu, phân tích các độ đo như Accuracy, Precision, Recall, F1-score, và Confussion matrix. Từ đó, đưa ra những thảo luận sâu hơn về hiệu quả, những ưu điểm, hạn chế của phương pháp đã áp dụng, và các yếu tố có thể đã ảnh hưởng đến kết quả cuối cùng.

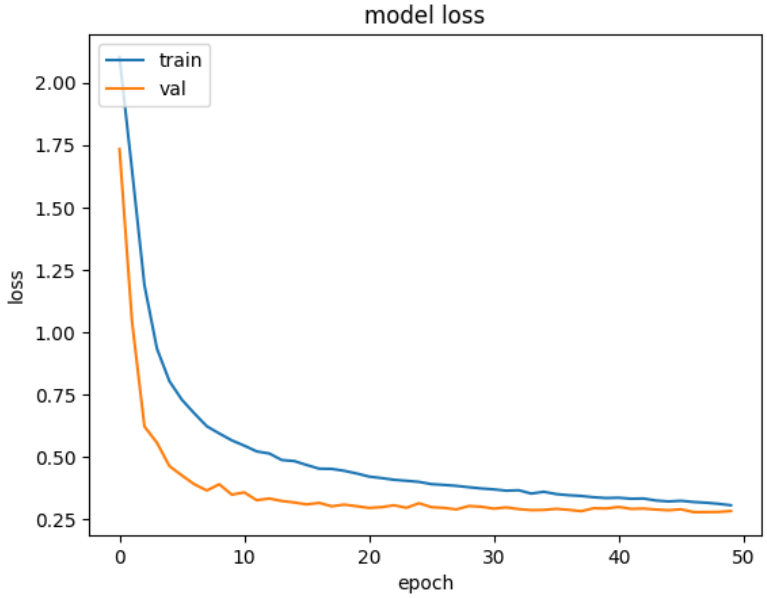
4.1. Kết quả huấn luyện mô hình

Quá trình huấn luyện mô hình LSTM được thực hiện với các thông số được thiết lập và mô tả chi tiết trong Chương 3 (mục 3.4.4). Mô hình sử dụng bộ tối ưu *adam*, hàm sai số *sparse_categorical_crossentropy,* và theo dõi độ chính xác với *accuracy*. Dữ liệu huấn luyện sau khi trích xuất đặc trưng được chia thành 95% cho huấn luyện thực tế và 5% cho tập kiểm định với *random_state=2019* để đảm bảo tính nhất quán. Mô hình được huấn luyện với *batch_size=512* với số epoch là 50.

Trong quá trình huấn luyện, độ chính xác (accuracy) và giá trị của hàm sai số (loss) trên cả tập huấn luyện và tập kiểm định được ghi nhận sau mỗi epoch để theo dõi khả năng học và tổng quát hoá của mô hình.



Hình 3. Biểu đồ Accuracy



Hình 4. Biểu đồ Loss

4.1.1. Phân tích kết quả đạt được

- **Về độ chính xác:**
  - Độ chính xác trên tập huấn luyện cho thấy sự tăng trưởng ổn định qua từng epoch bắt đầu từ khoảng **0.1491** và đạt đến giá trị cao là **0.9059** (**90.59%**) ở epoch cuối cùng (epoch thứ 50). Điều này cho thấy mô hình đã học được các mẫu và đặc trưng từ dữ liệu huấn luyện.
  - Độ chính xác trên tập kiểm định cũng tăng từ **0.3898** ở epoch thứ nhất và đạt **0.9154** ở epoch thứ 50.
  - Quan sát hai đường cong accuracy, có thể thấy rằng *val_accuracy* bám khá sát *train_accuracy* trong suốt quá trình huấn luyện. Sự chênh lệch giữa hai giá trị này không quá lớn (ví dụ, ở epoch cuối là **0.9059** so với **0.9154**), điều này là một dấu hiệu tích cực cho thấy mô hình không bị overfitting một cách nghiêm trọng. Mô hình có khả năng tổng quát hóa tốt trên dữ liệu mà nó chưa từng gặp trong quá trình huấn luyện ở mỗi epoch.
- **Về hàm sai số:**

- Giá trị hàm sai số trên tập huấn luyện giảm đáng kể từ epoch đầu tiên (**2.2054**) xuống chỉ còn **0.2985** ở epoch cuối. Điều này khẳng định mô hình đang tối ưu hoá tốt các trọng số của nó dựa trên dữ liệu huấn luyện.
- Hàm sai số trên tập kiểm định cũng cho ra giá trị giảm tương tự (từ **1.7341** xuống còn **0.2829**).
- Đường cong loss (Hình 4) cho thấy *val_loss* giảm song song với *train_loss* và không có xu hướng tăng trở lại một cách rõ rệt, tiếp tục củng cố nhận định rằng mô hình được kiểm soát tốt, tránh được hiện tượng quá khớp.

4.2. Đánh giá hiệu suất mô hình trên tập kiểm thử

Sau khi hoàn tất quá trình huấn luyện và lựa chọn mô hình dựa trên hiệu suất trên tập kiểm định, mô hình cuối cùng được đánh giá một cách khách quan trên tập kiểm thử. Tập kiểm thử này bao gồm 33,769 mẫu văn bản hoàn toàn mới, chưa từng được sử dụng trong quá trình huấn luyện hay kiểm định.

4.2.1. Độ chính xác tổng thể

Độ chính xác tổng thể là tỉ lệ phần trăm các mẫu văn bản trong tập kiểm thử được mô hình phân loại chính xác vào đúng chủ đề của chúng.

Kết quả đánh giá cho thấy mô hình đạt độ chính xác tổng thể là **89.75%** trên tập kiểm thử. Kết quả này được tính toán dựa trên dự đoán của mô hình (*test_predictions*) so với nhãn thực tế của tập kiểm thử (*y_test_one_hot*).

4.2.2. Báo cáo phân loại chi tiết

Để có cái nhìn sâu hơn về hiệu suất của mô hình đối với từng chủ đề cụ thể, báo cáo phân loại bao gồm các độ đo Precision, Recall và F1-score đã được tính toán. Bảng 1 dưới đây trình bày chi tiết các kết quả này:

	Precision	Recall	F1-score	Support
Chính trị xã hội	0.8557	0.8191	0.8370	5219
Đời sống	0.8317	0.8528	0.8421	3159
Khoa học	0.8405	0.7874	0.8130	1820

Kinh doanh	0.8222	0.9259	0.8710	2552
Pháp luật	0.9594	0.8555	0.9045	3868
Sức khoẻ	0.8846	0.9427	0.9127	3384
Thế giới	0.9091	0.9110	0.9100	2898
Thể thao	0.9737	0.9857	0.9796	5298
Văn hoá	0.9118	0.9333	0.9224	3090
Vi tính	0.9296	0.9254	0.9275	2481
Accuracy			0.8975	33769
Macro avg	0.8918	0.8939	0.8920	33769
Weight avg	0.8985	0.8975	0.8972	33769

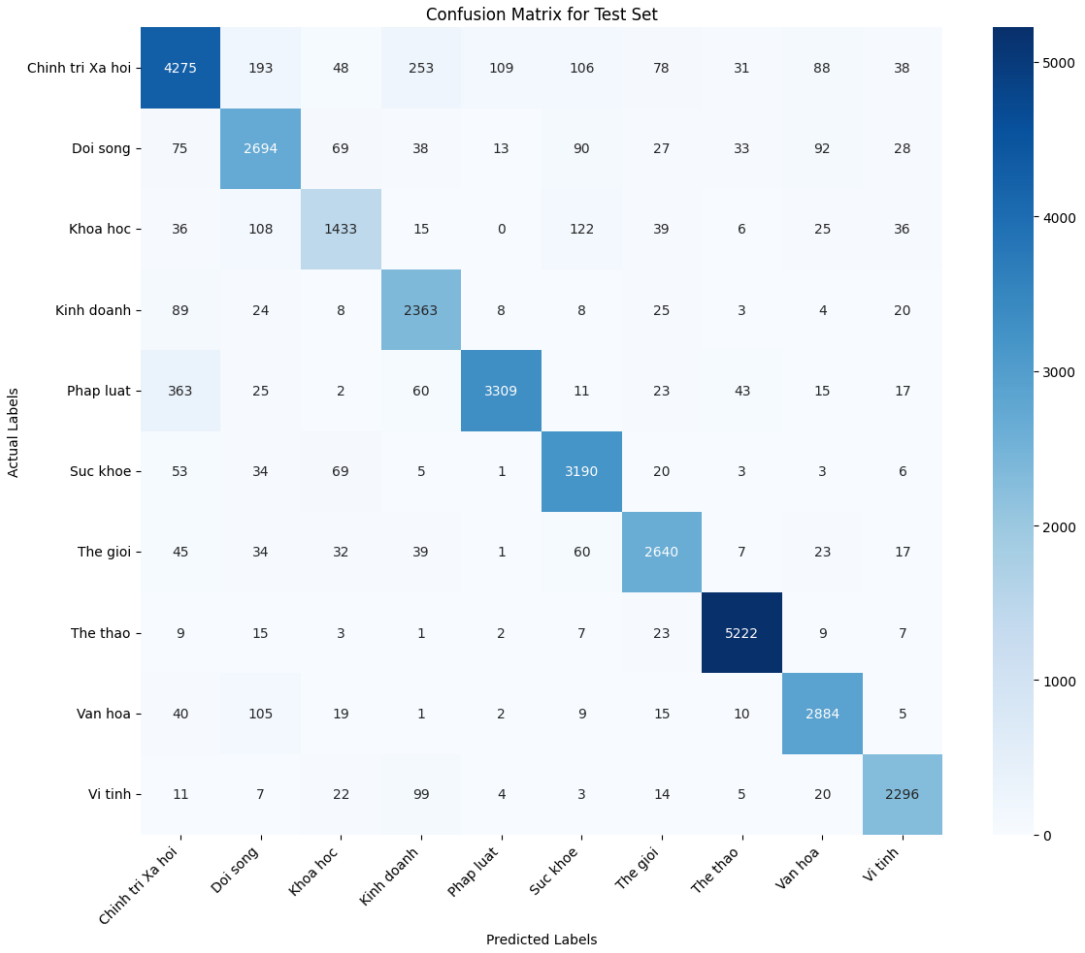
Bảng 1. Bảng báo cáo phân loại chi tiết

Nhận xét Classification Report:

- Các chủ đề như "*Pháp luật*", "*Thể thao*", "*Vi tính*", "*Sức khoẻ*", "*Thế giới*" đạt Precision cao (từ 0.88 trở lên), cho thấy khi mô hình dự đoán một văn bản thuộc các chủ đề này, khả năng cao là dự đoán đó chính xác.
- Chủ đề "*Thể thao*" có F1-score cao nhất (0.9796), cho thấy sự cân bằng tốt giữa Precision và Recall.
- Các chủ đề "*Đời sống*" và "*Khoa học*" có Precision thấp hơn (khoảng 0.83 - 0.84), nghĩa là mô hình có xu hướng dự đoán nhầm các văn bản từ chủ đề khác vào hai chủ đề này.
- Chủ đề "*Pháp luật*" có Precision rất cao (0.9604) nhưng Recall lại thấp hơn (0.8555), điều này ngụ ý mô hình rất chắc chắn khi phân loại vào "*Pháp luật*" nhưng có thể bỏ sót nhiều văn bản thực sự thuộc chủ đề này.

4.2.3. Phân tích Confussion Matrix

Confussion Matrix cung cấp cái nhìn trực quan về hiệu suất phân loại của mô hình, cho thấy số lượng các mẫu được phân loại đúng và sai giữa các cặp chủ đề. Các hàng của ma trận biểu diễn các lớp thực tế, và các cột biểu diễn các lớp được dự đoán bởi mô hình.



Hình 5. Confussion matrix của mô hình

Các giá trị trên đường chéo chính thể hiện số lượng các mẫu được phân loại đúng cho từng chủ đề. Chủ đề "*Thể thao*" có số lượng dự đoán đúng cao nhất. Chủ đề "*Khoa học*" có số lượng dự đoán đúng thấp hơn.

Dựa trên Confussion Matrix, có thể thấy các nhầm lẫn đáng chú ý:

- Chủ đề "*Pháp luật*" (Nhãn thật) bị nhầm lẫn nhiều nhất sang "*Chính trị Xã hội*" (Dự đoán).
- Chủ đề "*Chính trị Xã hội*" (Nhãn thật) bị nhầm lẫn nhiều sang "*Đời sống*" (Dự đoán) và "*Kinh doanh*" (Dự đoán).
- Chủ đề "*Đời sống*" (Nhãn thật) bị nhầm lẫn nhiều sang "*Khoa học*" (Dự đoán) và "*Văn hoá*" (Dự đoán).

- Chủ đề "*Văn hoá*" (Nhãn thật) bị nhầm lẫn nhiều sang "*Đời sống*" (Dự đoán).

Nguyên nhân là vì các cặp chủ đề này thường có sự tương đồng về từ vựng và ngữ cảnh, do đó dẫn đến việc mô hình gặp khó khăn trong việc phân biệt một cách rạch ròi.

4.3. Thảo luận kết quả

Từ các kết quả đánh giá chi tiết ở trên, nhóm em có thể đưa ra một số thảo luận và nhận định về hiệu quả của mô hình và phương pháp đã áp dụng:

4.3.1. Về hiệu quả tổng thể của mô hình

Với độ chính xác tổng thể đạt 89.75% và F1-score trung bình có trọng số là 0.8972 trên tập kiểm thử, mô hình LSTM kết hợp với phương pháp trích xuất đặc trưng TF-IDF và SVD đã chứng minh được khả năng phân loại văn bản tiếng Việt đa chủ đề ở mức độ khá tốt và đáng tin cậy.

Sự ổn định của quá trình huấn luyện (không có dấu hiệu overfitting nghiêm trọng) cho thấy kiến trúc mô hình và các kỹ thuật regularization (dropout) đã được áp dụng một cách hợp lý.

4.3.2. Ưu điểm của mô hình và phương pháp đã chọn

Khả năng học các biểu diễn phức tạp: Kiến trúc mạng LSTM sâu với nhiều lớp Dense đã cho phép mô hình học được các mối quan hệ phi tuyến tính và các đặc trưng trừu tượng từ dữ liệu đầu vào đã qua xử lý SVD, giúp phân biệt các chủ đề.

Kiểm soát hiệu quả hiện tượng quá khớp: Việc sử dụng các lớp Dropout với tỉ lệ tương đối cao (0.3 đến 0.65) đã đóng vai trò quan trọng trong việc giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới, tránh việc chỉ học thuộc lòng dữ liệu huấn luyện.

Hiệu quả của TF-IDF và SVD: Sự kết hợp của TF-IDF để nắm bắt tầm quan trọng của từ và SVD để giảm chiều, loại bỏ nhiễu và tạo ra các đặc trưng tiềm ẩn đã cung cấp một đầu vào chất lượng cho mạng LSTM. Việc giảm từ 10,000 đặc trưng TF-IDF xuống còn 500 đặc trưng SVD giúp giảm đáng kể độ phức tạp tính toán cho các lớp học sâu tiếp theo.

4.3.3. Hạn chế và các yếu tố ảnh hưởng đến kết quả.

Sự tương đồng ngữ nghĩa giữa các chủ đề: Hạn chế lớn nhất của mô hình là việc nhầm lẫn giữa các chủ đề có nội dung và từ vựng giao thoa cao như "*Pháp luật*" và "*Chính trị Xã hội*", hay "*Văn hoá*" và "*Đời sống*". Điều này cho thấy các đặc trưng dựa trên tần suất từ (TF-IDF) và các thành phần tiềm ẩn từ SVD có thể chưa đủ mạnh để nắm bắt hoàn toàn các sắc thái ngữ nghĩa tinh vi và ranh giới mờ giữa các chủ đề này.

Chất lượng và phạm vi của danh sách *stopword*: Hiệu quả của việc loại bỏ stopword ảnh hưởng trực tiếp đến chất lượng đặc trưng. Một danh sách *stopword* không đủ toàn diện hoặc loại bỏ nhầm các từ quan trọng có thể làm giảm hiệu suất.

Sự mất cân bằng dữ liệu: Mặc dù chưa phân tích chi tiết trong báo cáo này, nhưng nếu có sự chênh lệch lớn về số lượng mẫu giữa các chủ đề trong tập huấn luyện, mô hình có thể có xu hướng học tốt hơn trên các chủ đề đa số và kém hơn trên các chủ đề thiểu số. (Giá trị 'support' trong Bảng 4.1 cho thấy số lượng mẫu mỗi lớp trong tập test, có sự chênh lệch nhất định, ví dụ "*Thể thao*" có 5298 mẫu trong khi "*Khoa học*" chỉ có 1820 mẫu).

Bản chất của TF-IDF và SVD: Các phương pháp này không xem xét đến thứ tự của từ trong câu hoặc ngữ cảnh rộng hơn mà các từ xuất hiện, điều này có thể làm sai số thông tin ngữ nghĩa quan trọng.

Kiến trúc mô hình và tối ưu hóa siêu tham số: Mặc dù mô hình hiện tại cho kết quả khá, kiến trúc LSTM và các siêu tham số (số lớp, số units, tỉ lệ dropout, learning rate của optimizer, số epochs) được lựa chọn dựa trên kinh nghiệm và thử nghiệm ban đầu. Một quy trình tối ưu hóa siêu tham số kỹ lưỡng hơn (ví dụ: Grid Search, Random Search, Bayesian Optimization) có thể giúp tìm ra cấu hình tốt hơn nữa.

Độ sâu của mạng Dense: Số lượng lớn các lớp Dense sau LSTM có thể làm tăng nguy cơ overfitting nếu không được kiểm soát cẩn thận, mặc dù dropout đã được sử dụng.

4.4. Phân tích một số trường hợp dự đoán lỗi

Để có cái nhìn rõ hơn về những hạn chế của mô hình, chúng em xem xét một vài ví dụ cụ thể:

Ví dụ 1:

Câu văn gốc: *'Tác phẩm Khí chất Nam Bộ qua truyện Sơn Nam được tác giả Đinh Thị Thanh Thủy ấp ủ trong suốt 20 năm và bắt tay thực hiện trong hơn một năm qua*.'

Chủ đề thực tế: Văn hoá

Chủ đề dự đoán: Đời sống.

Nguyên nhân:

- Câu văn này tập trung vào việc giới thiệu một "*tác phẩm*" liên quan đến "*khí chất Nam Bộ*" và "*truyện Sơn Nam*", được một "*tác giả*" thực hiện. Tất cả những yếu tố này đều rất gần gũi và thường xuyên xuất hiện trong các văn bản thuộc chủ đề "*Văn hoá*" (bao gồm văn học, nghệ thuật, nghiên cứu văn hóa vùng miền).
- Mặc dù các yếu tố trên nghiêng về "*Văn hoá*", mô hình có thể nhầm sang "*Đời sống*" nếu trong tập huấn luyện, có nhiều bài viết "*Đời sống*" cũng nhắc đến các tác giả, tác phẩm hoặc các sự kiện văn hóa nhưng theo một khía cạnh đời thường hơn.
- Các từ ngữ như "*ấp ủ*", "*bắt tay thực hiện*" có thể xuất hiện trong nhiều ngữ cảnh đời thường khác. Mô hình không nắm bắt được trọng tâm chính của câu là "*tác phẩm*" và "*khí chất Nam Bộ qua truyện Sơn Nam*".

Ví dụ 2:

Câu văn gốc: "*Dự thảo Luật Đất đai sửa đổi lần này tập trung vào việc giải quyết các vướng mắc liên quan đến đền bù, giải phóng mặt bằng, cũng như các chính sách hỗ trợ tái định cư cho người dân bị ảnh hưởng bởi các dự án phát triển kinh tế - xã hội*"

Chủ đề thực tế: Pháp luật.

Chủ đề dự đoán: Chính trị Xã hội.

Nguyên nhân: Văn bản này đề cập đến "*Luật Đất đai*", "*đền bù*", "*giải phóng mặt bằng*" là các thuật ngữ pháp lý. Tuy nhiên, nó cũng chứa các cụm từ như "*chính sách hỗ trợ*", "*phát triển kinh tế - xã hội*" thường xuất hiện trong các văn bản về chính trị và các vấn đề xã hội. Sự giao thoa này có thể khiến mô hình, vốn dựa trên tần suất từ và các thành phần tiềm ẩn, gặp khó khăn trong việc xác định ranh giới rõ ràng và ưu tiên các yếu tố pháp lý hơn.

Chương V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Phân loại văn bản là một bài toán khó, đặc biệt là những văn bản trong ngôn ngữ đời thường. Tuy nhiên, bên cạnh những vấn đề khó khăn trong xử lý ngôn ngữ tự nhiên, chúng ta vẫn có thể tìm thấy những điều thú vị. Chính những thú vị này là động lực để chúng em nghiên cứu đề tài này.

Trong khuôn khổ đồ án, những vấn đề liên quan đến các bài toán xử lý ngôn ngữ như tách từ tiếng Việt, phân loại văn bản tiếng Việt, tìm kiếm văn bản tiếng Việt, tìm kiếm văn bản tiếng Việt theo chủ đề đã được chúng em tìm hiểu một cách nghiêm túc cả về chiều rộng và chiều sâu của vấn đề. Các kết quả của đồ án cho thấy các phương pháp tách từ và phân loại văn bản tiếng Việt với độ chính xác khá cao. Đây cũng là thành quả cho nhiều tháng nỗ lực không ngừng nghỉ của nhóm.

5.2. Hạn chế của đồ án

Mặc dù đã có những động lực thôi thúc tham gia nghiên cứu để giải quyết những vấn đề còn vướng mắc, vẫn tồn tại những vấn đề về xử lý ngôn ngữ tự nhiên, đặc biệt là tiếng Việt – ngôn ngữ có vốn từ rất rộng và đa dạng về nghĩa. Chính vì vậy, mô hình vẫn còn gặp khó khăn trong việc phân biệt các chủ đề có nội dung và từ vựng có những sự tương đồng về mặt nghĩa và lĩnh vực.

5.3. Hướng phát triển

Để giải quyết tốt hơn các bài toán phân loại văn bản tiếng Việt, việc xây dựng một bộ phân loại có khả năng học tăng cường sẽ giải quyết tốt việc phân loại các chủ đề đa dạng khi áp dụng vào thực tế. Đặc biệt là các hệ thống với khả năng học tăng cường trực tuyến có thể thích nghi khi gặp những bài toán thực tiễn. Thêm vào đó, việc áp dụng các kỹ thuật xử lý mất cân bằng dữ liệu cũng có thể cải tiến hiệu suất của mô hình trên các lớp có ít mẫu.

Những hướng phát triển này không chỉ giúp cải thiện độ chính xác và khả năng ứng dụng của hệ thống phân loại văn bản mà còn đóng góp vào sự phát triển chung của lĩnh vực xử lý ngôn ngữ tự nhiên cho tiếng Việt.

TÀI LIỆU THAM KHẢO

[1] V. Thuy, "*Gán nhãn hình thái cho từ ngữ liệu song ngữ Anh - Việt,*" trường Đại học Khoa học Tự nhiên - ĐHQG TPHCM, thành phố Hồ Chí Minh, 2005.

[2] Đ. Điền, "*Vấn đề ranh giới từ trong ngữ liệu song ngữ Anh - Việt,*" Viện ngôn ngữ học, Hội ngôn ngữ học Tp HCM, ĐH KHXH&NV, Ho Chi Minh City, 2002.

[3] Đ. Điền, "*Xây dựng và khai tác kho ngữ liệu song ngữ Anh - Việt,*" Đại học Khoa học Xã hội và Nhân văn - ĐHQGHCM, Ho Chi Minh city, 2005.

[4] H. Phê, "*Từ điển tiếng Việt,*" Trung tâm từ điển học, NXB Đà Nẵng, 1998.

[5] "*Wikipedia,*" [Online]. Available: https://en.wikipedia.org/wiki/Text. [Accessed 1 06 2025].

[6] F. Sebastiani, "*Text Classifiaction for Web Filtering,*" Pisa, 2004.

[7] F. Sebastiani, "*Machine Learning in Automated Text Categorization,*" ACM Computing Surveys, Vol. 34, No. 1, March 2002, pp. 1-47, 2002.

[8] N. L. N. Hoàng Công Duy Vũ, "T*ìm kiếm văn bản tiếng Việt theo chủ đề,*" trường Đại học Khoa học Tự nhiên - ĐHQGHCM, thành phố Hồ Chí Minh, 2006.

[9] https://github.com/DucLeTrong/vietnamese-text-classify.

[10] L. N. Thành, "*VIBLO,*" 11 02 2025. [Online]. Available: https://viblo.asia/p/tim-hieu-lstm-bi-quyet-giu-thong-tin-lau-dai-hieu-qua-MG24BaezVz3. [Accessed 1 06 2025].

[11] https://github.com/duyvuleo/VNTC.

PHỤ LỤC

BẢNG PHÂN CÔNG ĐÁNH GIÁ

STT	Thành viên	Chức vụ	Nhiệm vụ	Mức độ hoàn thành	Trung bình	Ghi chú	Ký tên
1	Nguyễn Huy Hoàng	Trưởng nhóm	Lựa chọn hướng tiếp cận mô hình, Xây dựng kiến trúc mô hình, Huấn luyện và lưu mô hình, nắm nội dung Chương 2, Chương 3	10 9 9	9		
2	Chu Quang Vinh	Thành viên	Thu thập dữ liệu và các bước tiền xử lý dữ liệu, nắm nội dung Chương 1,	9 9 9	9		

			Làm PowerPoint				
3	Hàng Hải Sơn	Thành viên	Đánh giá mô hình, nắm nội dung Chương 5, Tổng hợp tài liệu tham khảo và chỉnh sửa nội dung báo cáo	9 9 9	9		