LỜI MỞ ĐẦU

LỜI CẢM ƠN

NHẬN XÉT
(Của giảng viên hướng dẫn)

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Giảng viên hướng dẫn
(Ký và ghi rõ họ tên)

UBND TỈNH TRÀ VINH
TRƯỜNG ĐẠI HỌC TRÀ VINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(Của giảng viên hướng dẫn)

Họ và tên sinh viên:…… MSSV: ……

Ngành:…… Khóa:……..

Tên đề tài:…….

……..

……..

Họ và tên Giảng viên hướng dẫn:…….

Chức danh:……... Học vị:……

NHẬN XÉT

1. Nội dung đề tài:

……

……

……

……

……

……

……

2. Ưu điểm:

……

……

……

3. Khuyết điểm:

……

……

……

……

4. Điểm mới đề tài:

……
……
……
……

5. Giá trị thực trên đề tài:

……
……
……
……

6. Đề nghị sữa chửa bổ sung:

……
……
……
……
……

7. Đánh giá

……
……
……
……

Trà Vinh, *ngày tháng năm 2025*
Giảng viên hướng dẫn
(Ký và ghi rõ họ tên)

NHẬN XÉT
(Của giảng viên chấm)

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Giảng viên chấm
(Ký và ghi rõ họ tên)

UBND TỈNH TRÀ VINH
TRƯỜNG ĐẠI HỌC TRÀ VINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(Của giảng viên chấm)

Họ và tên người nhận xét:……

Chức danh:……... Học vị:……

Chuyên ngành:……..

Cơ quan công tác:…….

Tên sinh viên:……

Tên đề tài:…….

……..

……..

I. Ý KIẾN NHẬN XÉT

1. Nội dung:

……

……

……

……

……

……

……

……

2. Điểm mới các kết quả của khóa luận:

……

……

……

3. Ứng dụng thực tế:

……

……

……

……

II. CÁC VẤN ĐỀ CẦN LÀM RÕ

(Các câu hỏi của giáo viên phản biện)

……

……

……

……

……

……

……

……

III. KẾT LUẬN

(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đồ án khóa luận tốt nghiệp)

……

……

……

……

……

Trà Vinh, *ngày tháng năm 2025*

Giảng viên chấm

(Ký và ghi rõ họ tên)

MỤC LỤC

DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

DANH MỤC CÁC CỤM TỪ VIẾT TẮT

NLP:	Natural Language Processing.
ML:	Machine Learning.
DL:	Deep Learning.
BoW:	Bag of Word.
TF-IDF:	Term Frequency - Inverse Document Frequency.

CHƯƠNG 1. TỔNG QUAN

1.1 Tổng quan đề tài

Trong thời đại số hóa hiện tại, đặc biệt là với sự phát triển mạnh mẽ của các nền tảng thương mại điện tử, người tiêu dùng đang có xu hướng chuyển từ hình thức mua sắm trực tiếp sang mua sắm trực tuyến về đa số hàng hóa tiêu dùng. Vì vậy, các cửa hàng, doanh nghiệp cũng đang mở rộng kinh doanh trực tuyến để đáp ứng được nhu cầu của khách hàng.

Qua đó, khi các cửa hàng, doanh nghiệp có một lượng lớn khách hàng và doanh số sản phẩm bán ra được, việc quản lý, chăm sóc lấy ý kiến đánh giá từ khách hàng là rất quan trọng. Nhưng vì khi có lượng lớn ý kiến đánh giá như vậy, việc phân tích theo dõi và tiếp nhận những ý kiến đó theo cách thủ công truyền thống là vô cùng khó khăn. Vì thế, cần một giải pháp thực tiễn nào đó để việc tiếp nhận được ý kiến của lượng lớn khách hàng một cách nhanh chống và hoàn toàn tự động.

Cùng với đó, việc phát triển của trí tuệ nhân tạo những năm gần đây mở ra một ý tưởng trong việc tự động hóa các tác vụ vốn trước đây cần đến sự can thiệp thủ công, trong đó có bài toán phân tích và phân loại cảm xúc từ phản hồi của khách hàng. Với sự hỗ trợ của các kỹ thuật xử lý ngôn ngữ tự nhiên kết hợp cùng các mô hình học máy và học sâu, hoàn toàn có thể xây dựng một hệ thống tự động có khả năng hiểu và phân loại cảm xúc lượng lớn ý kiến đánh giá của khách hàng một cách nhanh chóng, chính xác và hiệu quả.

Đặc biệt, trong bối cảnh ngôn ngữ tiếng Việt vẫn còn hạn chế về tài nguyên xử lý ngôn ngữ so với các ngôn ngữ lớn như tiếng Anh, việc phát triển các hệ thống phân tích cảm xúc dành riêng cho tiếng Việt mang ý nghĩa rất quan trọng cả về mặt ứng dụng lẫn nghiên cứu. Không chỉ giúp cửa hàng, doanh nghiệp nắm bắt nhanh ý kiến cảm nhận của khách hàng, cải thiện chất lượng sản phẩm, dịch vụ mà còn góp phần thúc đẩy sự phát triển của các công nghệ ngôn ngữ tự nhiên cho tiếng Việt.

"Phân loại cảm xúc tiếng Việt bằng mô hình NLP" không chỉ có giá trị trong phạm vi học thuật mà còn mang tính thực tiễn cao. Nó hướng đến việc ứng dụng công nghệ hiện đại nhằm giải quyết bài toán thực tế đang được quan tâm, đồng thời là cơ hội để người học tiếp cận và làm chủ các công cụ, kỹ thuật mới trong lĩnh vực trí tuệ nhân tạo, lĩnh vực đang ngày càng phát triển mạnh mẽ và đầy tiềm năng trong thời đại số hóa hiện nay.

1.2 Mục tiêu đề tài

Mục tiêu chính của đề tài "Phân loại cảm xúc tiếng Việt bằng mô hình NLP" là nghiên cứu và xây dựng một hệ thống có khả năng tự động phân tích và phân loại cảm xúc từ các bình luận tiếng Việt, cụ thể là các bình luận người dùng trên các nền tảng thương mại điện tử. Hệ thống này cần có khả năng nhận diện và phân loại chính xác bình luận thành ba nhóm cảm xúc: tích cực, tiêu cực và trung lập, từ đó hỗ trợ các cửa hàng và doanh nghiệp hiểu rõ hơn về cảm nhận của khách hàng đối với sản phẩm, dịch vụ của mình.

Về mục tiêu học tập, nghiên cứu:

Nhằm củng cố và mở rộng kiến thức, kỹ năng chuyên môn trong lĩnh vực trí tuệ nhân tạo, đặc biệt là xử lý ngôn ngữ tự nhiên và học máy.

Hiểu và áp dụng kiến thức về xử lý ngôn ngữ tự nhiên tiếng Việt: Tìm hiểu sâu về đặc trưng ngôn ngữ tiếng Việt, bao gồm tách từ, chuẩn hóa văn bản, xử lý văn bản, đặc biệt là làm việc với ngôn ngữ thực tế trong các bình luận.

Làm quen và sử dụng các mô hình học máy và học sâu trong xử lý ngôn ngữ tự nhiên: Học cách sử dụng và so sánh hiệu quả giữa các thuật toán học máy truyền thống như Naive Bayes và các mô hình học sâu tiên tiến như BERT, đặc biệt là các mô hình tiền huấn luyện dành riêng cho tiếng Việt như PhoBERT.

Cải thiện khả năng nghiên cứu khoa học và tư duy phản biện: Tìm hiểu, phân tích tài liệu nghiên cứu, đánh giá kết quả thực nghiệm, so sánh mô hình, và đưa ra giải pháp cải tiến là những bước quan trọng giúp rèn luyện tư duy khoa học và nâng cao năng lực nghiên cứu độc lập.

1.3 Phạm vi nghiên cứu

Đề tài tập trung vào việc ứng dụng các kỹ thuật xử lý ngôn ngữ tự nhiên, học máy và học sâu để tự động phân loại cảm xúc trong các bình luận tiếng Việt. Cụ thể, phạm vi của đề tài được giới hạn trong ngữ cảnh phân tích cảm xúc của người dùng khi phản hồi sản phẩm trên các nền tảng thương mại điện tử phổ biến tại Việt Nam. Đây là môi trường dữ liệu thực tế, nơi người dùng thường xuyên để lại nhận xét, đánh giá về chất lượng sản phẩm, dịch vụ đã sử dụng. Việc lựa chọn dữ liệu từ các trang thương mại điện tử giúp đề tài tiếp cận sát với nhu cầu ứng dụng trong thực tiễn, đồng thời mang đến những thách thức thú vị trong quá trình xử lý ngôn ngữ tự nhiên tiếng Việt.

Dữ liệu của đề tài là các bình luận dạng văn bản, được trích xuất từ trang thương mại điện tử Tiki. Cảm xúc của các bình luận sẽ được phân loại thành ba nhóm cơ bản: tích cực, tiêu cực và trung lập. Đây là cách phân loại cảm xúc đơn giản nhưng hiệu quả, phù hợp với bài toán phân tích cảm xúc ở cấp độ ứng dụng. Việc gán nhãn được thực hiện một cách thủ công, tổng hợp từ nhiều người.

Đề tài áp dụng các bước tiền xử lý văn bản phù hợp với tiếng Việt, bao gồm: chuẩn hóa chữ viết, tách từ, loại bỏ ký tự đặc biệt, dấu câu. Sau khi tiền xử lý, dữ liệu sẽ được đưa vào các mô hình học máy và học sâu để huấn luyện. Các mô hình truyền thống như Naive Bayes được sử dụng như cơ sở so sánh, sau đó mô hình PhoBERT sẽ được áp dụng để nâng cao độ chính xác. Việc huấn luyện và đánh giá mô hình được thực hiện trên máy tính cá nhân cùng với nền tảng Google Colab.

1.4 Phương pháp nghiên cứu

Để đạt được mục tiêu của đề tài, quá trình thực hiện sẽ được chia thành các bước cụ thể, tuần tự theo quy trình xử lý dữ liệu và phát triển mô hình học máy và học sâu trong lĩnh vực xử lý ngôn ngữ tự nhiên. Cụ thể, các phương pháp được áp dụng bao gồm:

Khảo sát lý thuyết và tổng quan nghiên cứu: Tìm hiểu các khái niệm liên quan đến phân tích cảm xúc, các cấp độ cảm xúc và các loại cảm xúc phổ biến. Nghiên cứu các phương pháp dựa trên học máy và học sâu. Khảo sát các công trình

nghiên cứu trước đó liên quan đến phân tích cảm xúc trong tiếng Việt và các ngôn ngữ khác.

Thu thập và xử lý dữ liệu

Thu thập dữ liệu các bình luận của người dùng trên các nền tảng thương mại điện tử phổ biến tại Việt Nam là Tiki. Các bình luận này được viết bằng tiếng Việt, với văn tự nhiên, đôi khi có cả các từ viết tắt hoặc không có dấu câu. Điều này mang lại tính thực tế cao, phản ánh đúng với thực tế trong việc ứng dụng bài toán phân loại cảm xúc.

Sau khi thu thập dữ liệu, tiếp theo là xử lý và gán nhãn cảm xúc cho từng bình luận. Xử lý loại bỏ các ký tự đặc biệt, các biểu tượng cảm xúc, các dấu câu,… Để phục vụ cho mục tiêu nghiên cứu, mỗi bình luận sẽ được phân loại vào một trong ba nhãn cảm xúc: tích cực, tiêu cực, hoặc trung lập theo các tiêu chí riêng về các từ cảm xúc và được thực hiện bởi nhiều người, sau đó tổng hợp thống nhất về nhãn của dữ liệu.

Xây dựng mô hình học máy và học sâu: Thử nghiệm các mô hình học máy cổ điển để làm cơ sở so sánh, triển khai các mô hình học sâu phổ biến cho xử lý chuỗi văn bản, tinh chỉnh tham số và tối ưu mô hình.

Huấn luyện và đánh giá mô hình: Chia dữ liệu thành các tập huấn luyện, kiểm thử và kiểm tra theo tỷ lệ hợp lý, sử dụng các thước đo đánh giá như Accuracy, Precision, Recall, F1-score, Confusion Matrix để đo lường hiệu quả mô hình, so sánh kết quả giữa các mô hình học máy, học sâu và mô hình tiền huấn luyện để chọn ra phương án tối ưu.

Triển khai ứng dụng thử nghiệm: Phát triển một ứng dụng web đơn giản cho phép người dùng nhập vào đường dẫn của sản phẩm trên các sàn thương mại điện tử như Shopee, Tiki, Lazada, sau đó hệ thống sẽ tự động đánh giá các bình luận của sản phẩm đó và trả kết quả sau khi phân tích về cho người dùng, cụ thể ở đây là ứng dụng cho các cửa hàng và doanh nghiệp.

Tổng hợp kết quả và đề xuất hướng phát triển: Đưa ra nhận xét, phân tích những ưu, nhược điểm của các mô hình đã thử nghiệm. So sánh hiệu quả mô

hình với các kết quả trong các công trình nghiên cứu khác. Đề xuất hướng phát triển tiếp theo như mở rộng phạm vi cảm xúc, áp dụng cho các lĩnh vực khác như y tế, giáo dục,...

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về xử lý ngôn ngữ tự nhiên

2.1.1 Khái niệm

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một lĩnh vực liên ngành giữa khoa học máy tính, trí tuệ nhân tạo và ngôn ngữ học, nhằm mục tiêu xây dựng các hệ thống có khả năng hiểu, diễn giải và tạo ra ngôn ngữ của con người một cách tự động. NLP đóng vai trò trung gian trong việc giao tiếp giữa con người và máy tính bằng ngôn ngữ tự nhiên, cho phép máy tính xử lý và phân tích lượng lớn văn bản, lời nói mà không cần đến các lệnh lập trình phức tạp từ phía người dùng.

Với sự phát triển mạnh mẽ của dữ liệu văn bản trên các nền tảng số như mạng xã hội, thương mại điện tử, diễn đàn trực tuyến,... NLP ngày càng trở thành một công cụ không thể thiếu trong việc phân tích dữ liệu phi cấu trúc. Các ứng dụng phổ biến của NLP có thể kể đến như: phân tích cảm xúc, chatbot, dịch máy, trích xuất thông tin, tóm tắt văn bản, nhận dạng thực thể, và nhiều ứng dụng khác.

Trong những năm gần đây, sự phát triển của các mô hình ngôn ngữ lớn như BERT, GPT,... đã mang lại bước tiến vượt bậc cho NLP, đặc biệt là trong các bài toán phức tạp như phân loại cảm xúc, dịch máy, và hiểu ngữ cảnh sâu. Đối với tiếng Việt, mô hình PhoBERT được huấn luyện riêng cho dữ liệu tiếng Việt đã chứng minh hiệu quả vượt trội trong nhiều tác vụ, bao gồm phân tích cảm xúc.

2.1.2 Quy trình xử lý ngôn ngữ tự nhiên

Quy trình xử lý ngôn ngữ tự nhiên thường bao gồm nhiều giai đoạn:

Thu thập và tiền xử lý dữ liệu văn bản: Dữ liệu văn bản ban đầu thường chứa nhiều yếu tố không cần thiết như dấu câu, ký tự đặc biệt, từ dừng, từ viết tắt, lỗi chính tả,... nên cần được làm sạch và chuẩn hóa. Đối với tiếng Việt, bước tách từ là cực kỳ quan trọng do tiếng Việt là ngôn ngữ đơn âm tiết, không có dấu cách rõ ràng giữa các từ như tiếng Anh.

Biểu diễn văn bản: Sau khi tiền xử lý, văn bản cần được chuyển đổi thành dạng số để mô hình có thể xử lý được. Một số kỹ thuật phổ biến gồm có: Bag of Words (BoW), TF-IDF, word embeddings (Word2Vec, GloVe)…

Xây dựng mô hình học máy và học sâu: Tùy theo bài toán và yêu cầu về độ chính xác, người dùng có thể lựa chọn mô hình học máy truyền thống như Naive Bayes, SVM, Decision Tree,... hoặc các mô hình học sâu như CNN, RNN, LSTM, Transformer.

Đánh giá và tối ưu mô hình: Các mô hình sau khi huấn luyện cần được đánh giá bằng các chỉ số như độ chính xác (accuracy), độ bao phủ (recall), độ chính xác (precision), F1-score,... để so sánh và lựa chọn mô hình tốt nhất.

2.2 Phân tích cảm xúc

2.2.1 Khái niệm

Phân tích cảm xúc (Sentiment Analysis) hay còn gọi là phân loại cảm xúc, là một nhánh quan trọng trong xử lý ngôn ngữ tự nhiên (NLP), tập trung vào việc xác định thái độ, quan điểm hoặc cảm xúc của người viết đối với một chủ đề cụ thể trong văn bản. Mục tiêu chính của phân tích cảm xúc là xác định liệu văn bản thể hiện cảm xúc tích cực, tiêu cực hay trung lập, từ đó giúp các hệ thống hiểu được "ý định" hoặc "cảm nhận" của người dùng khi giao tiếp bằng ngôn ngữ tự nhiên.

Phân tích cảm xúc có rất nhiều ứng dụng trong thực tế, đặc biệt là trong lĩnh vực kinh doanh và chăm sóc khách hàng. Các doanh nghiệp có thể sử dụng phân tích cảm xúc để theo dõi phản hồi của người tiêu dùng, đánh giá mức độ hài lòng về sản phẩm, dịch vụ hoặc thương hiệu, từ đó đưa ra chiến lược cải thiện phù hợp. Ngoài ra, phân tích cảm xúc còn được ứng dụng trong phân tích dư luận xã hội, đánh giá hiệu quả chiến dịch truyền thông, quản lý danh tiếng và nhiều lĩnh vực khác.

Về mặt kỹ thuật, phân tích cảm xúc có thể được triển khai dưới nhiều mức độ khác nhau:

Mức câu: Xác định cảm xúc tổng thể của từng câu.

Mức đoạn văn, bình luận: Đánh giá cảm xúc chung của toàn bộ văn bản hoặc bài viết.

Mức thực thể: Phân tích cảm xúc hướng đến các khía cạnh cụ thể (ví dụ: người dùng thích "giá cả" nhưng không hài lòng với "chất lượng sản phẩm").

Đối với ngôn ngữ tiếng Việt, phân tích cảm xúc là một thách thức lớn do đặc điểm ngữ pháp đặc biệt, cách viết không chuẩn, sử dụng từ láy, tiếng lóng, viết tắt, cũng như hiện tượng từ đồng âm, dị nghĩa phổ biến. Do đó, việc áp dụng các mô hình hiện đại được thiết kế riêng cho tiếng Việt như PhoBERT sẽ mang lại kết quả tốt hơn so với việc sử dụng các mô hình học từ tiếng Anh hoặc ngôn ngữ khác.

Phân tích cảm xúc là một công cụ hữu ích giúp biến đổi dữ liệu văn bản phi cấu trúc thành thông tin có giá trị. Trong thời đại số hóa và dữ liệu lớn, việc áp dụng các kỹ thuật phân tích cảm xúc không chỉ mang lại lợi thế cạnh tranh cho doanh nghiệp mà còn mở ra nhiều hướng nghiên cứu và ứng dụng trong lĩnh vực trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên.

2.2.2 Các phương pháp phân tích cảm xúc

Phương pháp dựa trên từ điển cảm xúc: Sử dụng một tập hợp từ điển chứa các từ có gán nhãn cảm xúc (ví dụ: “tuyệt vời” là tích cực, “tệ” là tiêu cực). Văn bản được phân tích dựa trên việc đếm và tổng hợp các từ có trong từ điển để đưa ra dự đoán về cảm xúc. Tuy đơn giản và dễ triển khai, phương pháp này thường thiếu hiệu quả trong việc xử lý các ngữ cảnh phức tạp hoặc câu mang tính mỉa mai, ẩn dụ.

Phương pháp học máy truyền thống: Bao gồm các mô hình như Naive Bayes, SVM, KNN,... Các mô hình này học từ dữ liệu huấn luyện đã được gán nhãn để rút ra quy tắc phân loại. Trước khi đưa vào mô hình, văn bản thường được chuyển thành dạng vector thông qua các kỹ thuật biểu diễn như TF-IDF hoặc Bag of Words.

Phương pháp học sâu: Các mô hình học sâu như CNN, RNN, LSTM hoặc Transformer (BERT, RoBERTa, PhoBERT) cho phép khai thác tốt hơn mối quan hệ ngữ cảnh và cấu trúc ngôn ngữ trong văn bản. Đặc biệt, các mô hình tiền

huấn luyện như BERT hoặc PhoBERT có khả năng hiểu ngữ nghĩa theo ngữ cảnh sâu, giúp cải thiện đáng kể độ chính xác trong phân tích cảm xúc, đặc biệt đối với ngôn ngữ nhiều biến thể như tiếng Việt.

2.3 Kỹ thuật biểu diễn văn bản

Trong xử lý ngôn ngữ tự nhiên, việc biểu diễn văn bản dưới dạng mà máy tính có thể hiểu và xử lý được là một bước quan trọng, đặc biệt trong các bài toán phân loại văn bản như phân loại cảm xúc. Văn bản ở dạng tự nhiên là một chuỗi ký tự hoặc từ ngữ, nhưng để xử lý bằng các thuật toán học máy hoặc học sâu, văn bản cần được chuyển đổi sang dạng số hoặc vector đặc trưng.

2.3.1 Kỹ thuật Bag of Word (BoW)

Bag of Words là một trong những kỹ thuật biểu diễn văn bản đơn giản và phổ biến nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên. Ý tưởng chính của BoW là đại diện cho một văn bản như một tập hợp các từ ngữ (không quan tâm đến thứ tự) và biểu diễn văn bản dưới dạng vector dựa trên tần suất xuất hiện của từng từ trong tập từ vựng.

Cụ thể, với một tập hợp các văn bản, trước tiên người ta xây dựng một tập từ vựng bao gồm tất cả các từ xuất hiện trong tập dữ liệu huấn luyện. Mỗi văn bản sau đó sẽ được biểu diễn bằng một vector có chiều bằng kích thước của tập từ vựng. Mỗi phần tử trong vector đại diện cho số lần xuất hiện của một từ cụ thể trong văn bản đó. Ví dụ, nếu từ "vui" xuất hiện ba lần trong văn bản, thì phần tử tương ứng với từ "vui" trong vector sẽ có giá trị là 3.

BoW có ưu điểm là dễ hiểu, dễ triển khai và hiệu quả trong các bài toán phân loại cơ bản. Tuy nhiên, kỹ thuật này tồn tại nhiều hạn chế. Trước hết, nó bỏ qua thứ tự từ trong câu, do đó không nắm bắt được ngữ nghĩa đầy đủ của văn bản. Ngoài ra, BoW không thể hiện được mối quan hệ ngữ cảnh giữa các từ, ví dụ như sự khác biệt giữa "không vui" và "rất vui" có thể bị mất đi khi chỉ xét đến tần suất của từ "vui".

Một vấn đề khác là độ hiệu quả tính toán: với các tập từ vựng lớn, vector BoW sẽ có kích thước rất cao và thường rất thưa, gây tốn bộ nhớ và ảnh hưởng đến tốc độ xử lý.

Mặc dù có nhiều điểm hạn chế, BoW vẫn là một kỹ thuật nền tảng quan trọng trong lĩnh vực NLP và thường được dùng như một bước khởi đầu để so sánh với các mô hình phức tạp hơn như TF-IDF hay Word Embedding. Trong các bài toán phân loại cảm xúc tiếng Việt, BoW có thể hoạt động tốt với dữ liệu nhỏ và khi kết hợp với các kỹ thuật tiền xử lý, chuẩn hóa từ hoặc phân tách từ chính xác.

2.3.2 Kỹ thuật TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency) là một kỹ thuật biểu diễn văn bản nhằm đánh giá mức độ quan trọng của một từ trong một văn bản so với toàn bộ tập hợp các văn bản. Đây là sự cải tiến của mô hình Bag of Words, với mục tiêu không chỉ xét đến tần suất xuất hiện của từ trong văn bản, mà còn điều chỉnh dựa trên mức độ phổ biến của từ đó trên toàn bộ tập dữ liệu.

Cụ thể, TF-IDF là tích của hai thành phần:

TF (Term Frequency): đo tần suất xuất hiện của từ trong văn bản. Có thể tính bằng công thức:

TF(t, d) = (số lần từ t xuất hiện trong văn bản d) / (tổng số từ trong d)

IDF (Inverse Document Frequency): đo mức độ đặc trưng của từ, bằng cách giảm trọng số của các từ xuất hiện nhiều trong toàn bộ tập văn bản. Công thức thường dùng là:

$$IDF(t) = \log(N / (df(t) + 1))$$

Trong đó, N là tổng số văn bản trong tập dữ liệu, còn df(t) là số văn bản chứa từ t. Việc cộng thêm 1 vào mẫu số giúp tránh chia cho 0.

Khi nhân TF với IDF, ta được giá trị TF-IDF thể hiện độ quan trọng tương đối của từ trong một văn bản cụ thể. Các từ phổ biến như “và”, “là”, “của” có thể có TF cao nhưng do xuất hiện nhiều trong hầu hết các văn bản nên sẽ có IDF thấp, dẫn đến TF-IDF thấp. Ngược lại, các từ hiếm nhưng mang tính phân biệt cao (ví

dụ: “phấn khích”, “tức giận”, “buồn bã”) sẽ có trọng số cao, từ đó giúp mô hình học được những đặc điểm đặc trưng của từng cảm xúc.

Ưu điểm chính của TF-IDF là khả năng làm nổi bật các từ khóa quan trọng và loại bỏ các từ mang tính chung chung. Trong các bài toán phân loại cảm xúc, TF-IDF giúp mô hình nhận diện được các từ thể hiện cảm xúc rõ ràng, từ đó nâng cao hiệu quả phân loại.

Tuy nhiên, tương tự như Bag of Words, TF-IDF không xét đến ngữ cảnh hoặc thứ tự từ trong câu. Điều này có thể gây khó khăn trong việc hiểu các cấu trúc ngữ nghĩa phức tạp như phủ định ("không vui") hoặc các cụm từ có sắc thái cảm xúc đặc biệt. Ngoài ra, biểu diễn văn bản dưới dạng TF-IDF vẫn tạo ra vector thưa và kích thước cao, gây tốn tài nguyên tính toán với các tập dữ liệu lớn.

Mặc dù vậy, TF-IDF vẫn là một kỹ thuật mạnh mẽ và thường được dùng làm baseline trong các mô hình phân loại văn bản, bao gồm cả phân loại cảm xúc tiếng Việt. Khi kết hợp với các thuật toán học máy truyền thống như Naive Bayes, SVM hay Logistic Regression, TF-IDF có thể đạt hiệu quả tốt trên các tập dữ liệu vừa và nhỏ.

2.4 Tổng quan về học máy

2.4.1 Khái niệm

Học máy (Machine Learning – ML) là một lĩnh vực thuộc Trí tuệ nhân tạo (AI) cho phép máy tính học từ dữ liệu mà không cần được lập trình một cách rõ ràng. Mục tiêu của học máy là xây dựng các mô hình có khả năng suy luận hoặc đưa ra dự đoán dựa trên các mẫu đã học được từ dữ liệu quá khứ.

2.4.2 Phân loại học máy

Học máy có thể chia thành ba nhóm chính:

Học có giám sát (Supervised Learning): Là hình thức học dựa trên dữ liệu có nhãn. Mô hình được cung cấp các cặp dữ liệu đầu vào và đầu ra tương ứng để học cách ánh xạ giữa chúng. Ví dụ phân loại văn bản, hồi quy giá trị,...

Học không giám sát (Unsupervised Learning): Mô hình học từ dữ liệu chưa có nhãn, thường dùng trong các bài toán như phân cụm, giảm chiều dữ liệu,... Ví dụ nhóm các bình luận có nội dung tương tự nhau.

Học tăng cường (Reinforcement Learning): Mô hình học thông qua việc tương tác với môi trường và nhận phần thưởng (reward) hoặc hình phạt (penalty). Lĩnh vực này phổ biến trong robot, trò chơi, hoặc tối ưu hệ thống.

2.4.3 Một số thuật toán học máy phổ biến trong NLP

Trong các bài toán xử lý ngôn ngữ tự nhiên, đặc biệt là phân loại văn bản và cảm xúc, học máy truyền thống vẫn đóng vai trò quan trọng nhờ khả năng huấn luyện nhanh, dễ triển khai và hiệu quả tốt khi dữ liệu không quá lớn.

Thuật toán học máy phổ biến trong bài toán xử lý ngôn ngữ tự nhiên:

Naive Bayes là một trong những thuật toán đơn giản và hiệu quả nhất trong phân loại văn bản. Dựa trên định lý Bayes và giả định độc lập giữa các đặc trưng, mô hình này tính xác suất một văn bản thuộc về một nhãn cụ thể dựa trên tần suất xuất hiện của các từ. Mặc dù giả định độc lập hiếm khi đúng trong thực tế, Naive Bayes vẫn cho kết quả đáng tin cậy trong nhiều ứng dụng thực tế, đặc biệt là với dữ liệu văn bản ngắn và đơn giản như bình luận hoặc đánh giá sản phẩm.

Hồi quy logistic (Logistic Regression) là một mô hình phân loại nhị phân truyền thống, hoạt động bằng cách học một hàm tuyến tính giữa các đặc trưng đầu vào và xác suất đầu ra thuộc một lớp cụ thể. Mặc dù mang tên "hồi quy", nhưng Hồi quy logistic thực chất là một mô hình phân loại, thường được mở rộng cho các bài toán đa lớp. Mô hình này có thể diễn giải tốt, dễ huấn luyện và hoạt động ổn định với các đặc trưng rời rạc như từ vựng trong văn bản.

Support Vector Machine (SVM) là một thuật toán mạnh mẽ và phổ biến trong phân loại văn bản, đặc biệt là trong các bài toán yêu cầu độ chính xác cao. SVM hoạt động bằng cách tìm một siêu phẳng tối ưu để phân tách các lớp dữ liệu. Trong không gian đặc trưng cao chiều như biểu diễn văn bản (TF-IDF hoặc Word Embedding), SVM thường cho kết quả rất tốt nhờ khả năng tối đa hóa biên phân cách giữa các lớp.

Ngoài ra, còn có một số thuật toán ít phổ biến hơn trong việc xử lý ngôn ngữ tự nhiên như K láng giềng gần nhất (K-Nearest Neighbors – KNN), Cây quyết định (Decision Tree) và Rừng ngẫu nhiên (Random Forest), do hiệu quả đem lại không cao mà chi phí tính toán cao khi dữ liệu càng lớn.

Các thuật toán học máy trên đều cần dữ liệu đầu vào là dạng số, cho nên khi xử lý ngôn ngữ tự nhiên, bước tiền xử lý dữ liệu và trích xuất đặc trưng là vô cùng quan trọng. Việc lựa chọn đúng phương pháp và kết hợp với thuật toán học máy phù hợp sẽ mang lại kết quả tốt cho mô hình bài toán cần huấn luyện.

2.4.4 Ưu nhược điểm của học máy

2.4.4.1 Ưu điểm

Tự động học từ dữ liệu: Thay vì xây dựng các luật thủ công để phân loại, học máy cho phép mô hình học trực tiếp từ dữ liệu đầu vào và đưa ra quyết định dựa trên mẫu đã học. Điều này giúp tiết kiệm thời gian và công sức khi xử lý các bài toán có nhiều tình huống không thể lập trình tường minh.

Tổng quát hóa tốt: Với một tập dữ liệu đủ lớn và đa dạng, mô hình học máy có thể khái quát hóa và dự đoán chính xác cho các trường hợp chưa từng gặp, đặc biệt hiệu quả trong các bài toán như phân loại văn bản hay cảm xúc.

Hiệu quả với dữ liệu vừa phải: Các thuật toán như Naive Bayes, SVM hay Logistic Regression có thể đạt hiệu suất tốt ngay cả khi không cần lượng dữ liệu quá lớn.

Tốc độ huấn luyện nhanh, dễ triển khai: Các mô hình học máy có tốc độ huấn luyện nhanh, ít tốn tài nguyên tính toán, phù hợp với các bài toán vừa và nhỏ hoặc yêu cầu triển khai nhanh.

Dễ giải thích và kiểm soát: Một số mô hình học máy (Logistic Regression, Cây quyết định,…) có khả năng giải thích rõ ràng cách mô hình đưa ra dự đoán, hỗ trợ tốt cho các hệ thống cần tính minh bạch.

2.4.4.2 Nhược điểm

Phụ thuộc nhiều vào trích xuất đặc trưng: Hiệu quả của mô hình học máy phụ thuộc rất lớn vào cách biểu diễn dữ liệu đầu vào. Nếu đặc trưng không

phản ánh đúng ngữ nghĩa hoặc không loại bỏ được nhiễu, mô hình dễ đưa ra kết quả sai lệch.

Khó nắm bắt ngữ cảnh và ngữ nghĩa sâu: Các mô hình học máy truyền thống thường dựa vào tần suất từ vựng (như TF-IDF), nên không thể hiểu được ngữ nghĩa sâu hoặc mối liên hệ dài giữa các từ trong văn bản, đặc biệt là với các ngôn ngữ giàu ngữ cảnh như tiếng Việt.

Dễ bị ảnh hưởng bởi dữ liệu nhiễu hoặc mất cân bằng nhãn: Khi tập dữ liệu chứa nhiều bình luận rác, lỗi chính tả, từ không dấu hoặc chênh lệch quá nhiều giữa các nhãn (ví dụ nhãn "tích cực" chiếm đa số), mô hình dễ bị học lệch và suy giảm độ chính xác.

Không thích hợp với dữ liệu lớn hoặc quá phức tạp: Với những bài toán yêu cầu hiểu sâu về ngôn ngữ hoặc xử lý dữ liệu rất lớn, các mô hình học máy truyền thống có thể không tốt về hiệu năng và độ chính xác.

2.5 Tổng quan về học sâu

2.5.1 Khái niệm

Học sâu (Deep Learning – DL) là một nhánh mở rộng của học máy, tập trung vào việc xây dựng và huấn luyện các mạng nơ-ron nhân tạo nhiều, nhằm mô phỏng cách thức hoạt động của bộ não con người trong việc xử lý dữ liệu và trích xuất thông tin. Điểm nổi bật của học sâu so với học máy truyền thống là khả năng tự động học biểu diễn đặc trưng từ dữ liệu thô mà không cần sự can thiệp thủ công từ con người.

Trong học sâu, các mô hình mạng nơ-ron bao gồm nhiều lớp ẩn (hidden layers), cho phép hệ thống học được các đặc trưng ở nhiều cấp độ khác nhau. Với dữ liệu văn bản, các lớp đầu tiên thường học thông tin cơ bản như từ vựng hoặc ký tự, trong khi các lớp sâu hơn có thể học được các cấu trúc ngữ pháp, ngữ nghĩa hoặc thậm chí cả sắc thái cảm xúc của câu. Chính nhờ khả năng học được biểu diễn ngữ nghĩa phức tạp, học sâu đã đạt được nhiều bước tiến vượt bậc trong các bài toán xử lý ngôn ngữ tự nhiên (NLP) như dịch máy, phân loại cảm xúc, tóm tắt văn bản, hoặc trả lời câu hỏi.

2.5.2 Một số kiến trúc học sâu tiêu biểu trong NLP

Mạng nơ-ron tích chập (Convolutional Neural Network – CNN): Dù ban đầu được thiết kế cho xử lý ảnh, CNN cũng được sử dụng hiệu quả trong phân loại văn bản nhờ khả năng nhận diện các mẫu cục bộ trong chuỗi từ, ví dụ như cụm từ biểu thị cảm xúc.

Mạng nơ-ron hồi tiếp (Recurrent Neural Network – RNN): Đây là các mô hình phù hợp với dữ liệu chuỗi như văn bản, có khả năng ghi nhớ thông tin từ các bước trước đó trong câu. RNN được sử dụng rộng rãi trong các bài toán phân tích cảm xúc, phân tích ý định, hoặc sinh văn bản.

Transformer: Là kiến trúc mang tính đột phá trong NLP hiện đại, Transformer giúp xử lý ngữ cảnh dài tốt hơn nhờ cơ chế tự chú ý (self-attention). Các mô hình như BERT, GPT, RoBERTa, XLM-R,... đều dựa trên kiến trúc này. Đặc biệt, các mô hình tiền huấn luyện như BERT đã chứng minh khả năng hiểu ngữ nghĩa và cảm xúc sâu sắc từ văn bản đầu vào, kể cả trong tiếng Việt, nhờ được huấn luyện trên kho dữ liệu lớn.

2.5.3 Ưu nhược điểm của học sâu

2.5.3.1 Ưu điểm

Tự động trích xuất đặc trưng: Một trong những điểm mạnh nổi bật nhất của học sâu là khả năng tự động học đặc trưng từ dữ liệu đầu vào, mà không cần phải xử lý thủ công các đặc trưng như trong học máy truyền thống. Điều này giúp giảm thiểu công sức tiền xử lý và loại bỏ sự phụ thuộc vào kiến thức chuyên môn trong việc lựa chọn đặc trưng.

Hiểu ngữ nghĩa và ngữ cảnh tốt hơn: Các kiến trúc học sâu như RNN, LSTM, và đặc biệt là Transformer (BERT, GPT,...) có khả năng nắm bắt quan hệ ngữ cảnh dài trong văn bản, hiểu được ý nghĩa câu và sắc thái cảm xúc phức tạp – điều mà các mô hình học máy truyền thống khó thực hiện.

Hiệu suất vượt trội với dữ liệu lớn: Khi được huấn luyện trên tập dữ liệu đủ lớn, mô hình học sâu thường đạt độ chính xác cao hơn đáng kể so với các mô hình học máy, nhờ khả năng học được các biểu diễn ngôn ngữ trừu tượng hơn.

Khả năng mở rộng linh hoạt: Học sâu có thể áp dụng cho nhiều bài toán NLP khác nhau như phân loại cảm xúc, sinh văn bản, dịch máy, phân tích thực thể,... chỉ cần điều chỉnh đầu ra và kiến trúc phù hợp.

2.5.3.2 Nhược điểm

Đòi hỏi tài nguyên tính toán lớn: Việc huấn luyện các mô hình học sâu yêu cầu phần cứng mạnh (như GPU), đặc biệt với các mô hình nhiều tham số như BERT hoặc GPT. Điều này làm tăng chi phí triển khai và khó tiếp cận với các dự án có nguồn lực hạn chế.

Cần lượng dữ liệu lớn để đạt hiệu quả cao: Mô hình học sâu hoạt động tốt nhất khi có một tập dữ liệu huấn luyện phong phú và đa dạng. Trong các trường hợp dữ liệu ít hoặc mất cân bằng nhãn (ví dụ: số lượng nhãn "trung lập" ít hơn nhiều so với "tích cực"), mô hình có thể bị lệch hoặc quá khớp.

Khó giải thích và thiếu minh bạch: Các mô hình học sâu thường khó lý giải cụ thể vì sao mô hình đưa ra một dự đoán nhất định. Điều này là một rào cản trong các hệ thống yêu cầu giải thích rõ ràng, như các ứng dụng trong y tế, pháp luật, hoặc tài chính.

Thời gian huấn luyện lâu và khó điều chỉnh: Quá trình huấn luyện mô hình học sâu thường kéo dài, đòi hỏi phải thử nghiệm nhiều lần với các siêu tham số khác nhau (learning rate, batch size, số lớp, v.v.). Việc tinh chỉnh mô hình hiệu quả cũng đòi hỏi kinh nghiệm và hiểu biết sâu về mạng nơ-ron.

2.6 Thuật toán Naïve Bayes

2.6.1 Khái niệm

Naïve Bayes là một trong những thuật toán học máy có giám sát đơn giản nhưng hiệu quả, đặc biệt phù hợp với các bài toán phân loại văn bản như phân loại cảm xúc, phân loại thư rác, hoặc phân loại chủ đề văn bản. Đây là thuật toán dựa trên định lý Bayes trong xác suất thống kê, kết hợp với giả định “ngây thơ” (naïve) rằng các đặc trưng là độc lập có điều kiện với nhau.

2.6.2 Định lý Bayes

Thuật toán Naive Bayes dựa trên định lý Bayes, định lý cung cấp một phương pháp để tính xác suất hậu nghiệm *P(C|X),* tức là xác suất để một mẫu dữ liệu *X* thuộc về lớp *C,* dựa trên xác suất tiên nghiệm của lớp *C* và xác suất có điều kiện *P(X|C)*:

$$P(C|X) = \frac{P(X|C)\,.P(C)}{\mathrm{P(X)}}$$

Trong đó:

C: là nhãn phân lớp (ví dụ: tích cực, tiêu cực, trung lập).

X: là đặc trưng đầu vào (thường là một chuỗi các từ trong văn bản).

P(C|X): xác suất một văn bản có đặc trưng ***X*** thuộc lớp ***C***.

P(X|C): xác suất xuất hiện của đặc trưng ***X*** khi biết văn bản thuộc lớp ***C***.

P(C): xác suất tiên nghiệm của lớp ***C***.

P(X): xác suất xảy ra của đặc trưng ***X***.

2.6.3 Giả định "Naive" – Độc lập có điều kiện

Naive Bayes đơn giản hóa việc tính toán xác suất *P(X|C)* bằng giả định rằng các đặc trưng $x_1, x_2, \ldots x_n$ trong *X* là độc lập với nhau khi biết lớp *C*:

$$P(X|C) = P(x_1, x_2, \ldots, x_n|C) = \prod_{i=1}^{n} P(x_i|C)$$

Giả định này làm giảm đáng kể độ phức tạp trong việc tính toán, đặc biệt khi số lượng đặc trưng lớn như trong phân tích văn bản, mỗi từ trong từ điển có thể được xem như một đặc trưng.

Khi áp dụng vào bài toán phân loại văn bản, đặc trưng thường là các từ hoặc n-gram trong văn bản. Mỗi văn bản được biểu diễn như một vector từ vựng với các giá trị là số lần xuất hiện, tần suất hoặc nhị phân. Do đó, mô hình Naive Bayes sẽ tính xác suất văn bản thuộc từng lớp dựa trên tần suất xuất hiện của các từ trong các văn bản đã biết nhãn.

2.6.4 Ưu điểm và nhược điểm của Naïve Bayes

2.6.4.1 Ưu điểm

Đơn giản và dễ triển khai: Naive Bayes là thuật toán học máy có cấu trúc lý thuyết đơn giản. Việc cài đặt, huấn luyện và suy luận dựa trên mô hình này không đòi hỏi nhiều tài nguyên tính toán, giúp tiết kiệm thời gian và công sức trong giai đoạn phát triển.

Tốc độ xử lý nhanh: Cả quá trình huấn luyện và dự đoán của Naive Bayes đều có độ phức tạp thấp, chủ yếu dựa vào việc đếm tần suất và tính xác suất, giúp mô hình hoạt động rất nhanh, đặc biệt với các tập dữ liệu lớn hoặc yêu cầu phản hồi theo thời gian thực.

Ít bị ảnh hưởng bởi dữ liệu nhiễu: Do mô hình dựa trên xác suất tổng thể, nên những dữ liệu nhiễu hoặc ngoại lệ nhỏ ít ảnh hưởng nghiêm trọng đến kết quả phân loại.

Hoạt động tốt với tập dữ liệu nhỏ: Khác với các mô hình học sâu đòi hỏi tập dữ liệu lớn để học biểu diễn tốt, Naive Bayes có thể huấn luyện hiệu quả ngay cả khi dữ liệu huấn luyện còn hạn chế về số lượng.

2.6.4.2 Nhược điểm

Giả định độc lập không thực tế: Naive Bayes giả định rằng các đặc trưng là độc lập có điều kiện với nhau, điều này thường không đúng trong thực tế, đặc biệt trong ngôn ngữ tự nhiên, các từ có mối quan hệ về ngữ nghĩa và cú pháp. Sự vi phạm giả định này có thể làm giảm hiệu quả phân loại.

Không xét đến thứ tự từ và ngữ cảnh: Thuật toán không thể hiểu hoặc xử lý được mối quan hệ theo ngữ cảnh giữa các từ, cũng như không thể học được các cấu trúc cú pháp hoặc ngữ nghĩa phức tạp trong câu.

Dễ bị sai lệch nếu có từ chưa từng xuất hiện: Nếu một từ trong văn bản kiểm tra chưa từng xuất hiện trong tập huấn luyện (thuộc một lớp cụ thể), xác suất $P(x_i|C)$ sẽ bằng 0, khiến xác suất toàn bộ văn bản bằng 0. Tuy có thể khắc phục bằng một số kỹ thuật, nhưng nó vẫn là điểm yếu.

Độ chính xác thấp hơn so với các mô hình hiện đại: So với các thuật toán học sâu như LSTM, Transformer (BERT, GPT), hoặc các mô hình học máy mạnh như SVM, độ chính xác của Naive Bayes thường thấp hơn trong những bài toán phức tạp hoặc yêu cầu hiểu sâu về ngữ nghĩa.

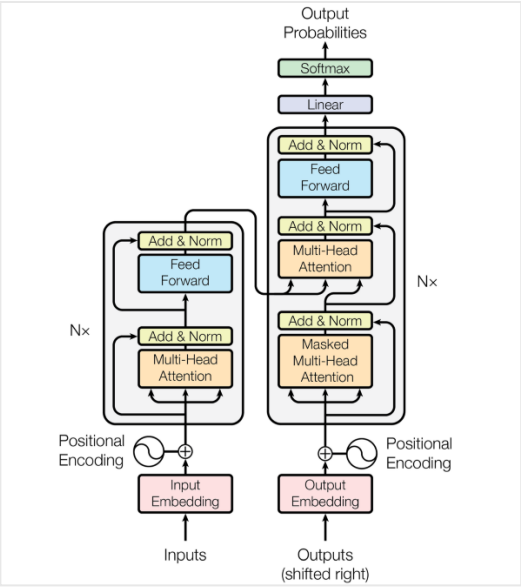
2.7 Mô hình PhoBERT

2.7.1 Khái niệm

PhoBERT là một mô hình ngôn ngữ tiền huấn luyện (pre-trained language model) dành riêng cho tiếng Việt, được phát triển dựa trên kiến trúc BERT (Bidirectional Encoder Representations from Transformers) của Google. Đây là bước tiến quan trọng trong việc áp dụng các kỹ thuật học sâu tiên tiến vào xử lý ngôn ngữ tự nhiên tiếng Việt, một ngôn ngữ phức tạp về ngữ pháp, cú pháp và có sự khác biệt rõ rệt so với tiếng Anh.

2.7.2 Kiến trúc tổng quan

PhoBERT sử dụng kiến trúc Transformer Encoder tương tự như BERT gốc. Transformer là một mô hình mạng nơ-ron dựa trên cơ chế self-attention, cho phép mô hình học được mối quan hệ giữa các từ trong câu mà không cần dựa vào thông tin vị trí tuyệt đối như trong RNN hay CNN. Mô hình học các biểu diễn ngữ nghĩa của từ trong ngữ cảnh hai chiều, giúp nắm bắt tốt hơn ý nghĩa và cảm xúc trong câu.



Hình 2.1: Kiến trúc Transformer

PhoBERT có hai phiên bản chính:

PhoBERT-base: Có cấu hình tương đương BERT-base với 12 lớp Transformer, 768 chiều ẩn, và 12 heads.

PhoBERT-large: Tương đương BERT-large với 24 lớp Transformer, 1024 chiều ẩn, và 16 heads.

2.7.3 Đặc điểm của PhoBERT

Điểm khác biệt quan trọng giữa PhoBERT và BERT gốc là ngôn ngữ và dữ liệu huấn luyện:

Dữ liệu tiền huấn luyện: PhoBERT được huấn luyện trên một tập dữ liệu văn bản tiếng Việt khổng lồ, khoảng 20GB văn bản sạch được trích xuất từ Common Crawl (gọi là Vietnamese Wikipedia + CC100-Viet). Giúp mô hình hiểu rõ hơn ngữ pháp, cách dùng từ, cấu trúc câu và sắc thái biểu cảm của tiếng Việt.

Tiền xử lý bằng từ ghép Byte-Pair Encoding (BPE): Không giống tiếng Anh, tiếng Việt có nhiều từ đa âm tiết được viết cách nhau như "máy tính", "trí tuệ nhân tạo". PhoBERT sử dụng kỹ thuật BPE giống RoBERTa để mã hóa từ, giúp tránh việc tách sai nghĩa của các từ tiếng Việt.

Dựa trên kiến trúc RoBERTa: PhoBERT thực chất không phải là một bản sao trực tiếp của BERT mà là bản mở rộng dựa trên RoBERTa, một cải tiến của BERT được huấn luyện với nhiều dữ liệu hơn và không sử dụng nhiệm vụ NSP (Next Sentence Prediction). Do đó, PhoBERT có khả năng biểu diễn ngữ nghĩa mạnh mẽ hơn.

2.7.4 Ứng dụng PhoBERT trong phân loại cảm xúc tiếng Việt

Trong bài toán phân loại cảm xúc, mỗi văn bản đầu vào sẽ được mã hóa thành chuỗi token bằng tokenizer của PhoBERT, sau đó đưa vào mô hình để trích xuất vector biểu diễn. Vector này sẽ được đưa vào một lớp phân loại để dự đoán nhãn cảm xúc: tích cực, trung lập, hoặc tiêu cực.

PhoBERT đặc biệt hiệu quả trong phân tích cảm xúc tiếng Việt do:

Nắm bắt tốt mối quan hệ ngữ nghĩa phức tạp giữa các từ trong câu.

Nhận diện chính xác sắc thái biểu cảm dù ngôn từ có thể ngắn gọn, ẩn dụ hoặc có dấu hiệu mỉa mai.

Tăng đáng kể độ chính xác so với các mô hình truyền thống như Naive Bayes hoặc các mô hình word embedding đơn giản như Word2Vec, TF-IDF.

2.7.5 Ưu điểm và nhược điểm của PhoBERT

2.7.5.1 Ưu điểm

Được huấn luyện chuyên biệt cho tiếng Việt: PhoBERT là mô hình đầu tiên được huấn luyện trên tập dữ liệu lớn gồm các văn bản tiếng Việt, giúp mô hình học được đặc trưng ngữ pháp, cú pháp, cấu trúc câu và sắc thái biểu cảm đặc thù của ngôn ngữ này. Điều này mang lại lợi thế lớn so với các mô hình đa ngôn ngữ trong các bài toán phân tích ngữ nghĩa tiếng Việt.

Hiểu ngữ cảnh tốt nhờ kiến trúc Transformer: PhoBERT sử dụng kiến trúc Transformer với cơ chế tự chú ý, cho phép mô hình hiểu được mối quan hệ giữa các từ trong cả hai chiều trái và phải của câu. Điều này rất quan trọng trong việc nắm bắt các biểu hiện cảm xúc thường được ngụ ý hoặc ẩn dụ trong văn bản tiếng Việt.

Khả năng biểu diễn ngôn ngữ mạnh mẽ: Các vector biểu diễn từ do PhoBERT tạo ra mang tính ngữ nghĩa cao, có thể được sử dụng hiệu quả trong nhiều tác vụ như phân loại cảm xúc, phân tích quan điểm, trích xuất thực thể, và nhiều ứng dụng NLP khác.

Dễ dàng tinh chỉnh theo bài toán cụ thể: PhoBERT có thể được fine-tune trên một tập dữ liệu nhỏ hơn cho các tác vụ cụ thể, giúp tăng độ chính xác trong những ứng dụng mục tiêu như phân loại cảm xúc trong đánh giá sản phẩm, ý kiến người dùng,…

2.7.5.2 Nhược điểm

Yêu cầu tài nguyên tính toán lớn: PhoBERT là mô hình học sâu với hàng triệu tham số, do đó việc huấn luyện và suy luận đòi hỏi phần cứng mạnh, đặc biệt là GPU với bộ nhớ lớn. Điều này có thể là rào cản đối với các hệ thống có tài nguyên hạn chế.

Thời gian huấn luyện và suy luận chậm hơn: So với các mô hình truyền thống như Naive Bayes, PhoBERT có tốc độ xử lý chậm hơn đáng kể, đặc biệt khi làm việc với tập dữ liệu lớn hoặc yêu cầu xử lý thời gian thực.

Cần tiền xử lý dữ liệu phù hợp: PhoBERT sử dụng tokenizer riêng dựa trên phương pháp BPE. Do đó, quá trình tiền xử lý dữ liệu đầu vào cần tuân thủ đúng định dạng yêu cầu của mô hình, nếu không sẽ dẫn đến sai lệch trong kết quả phân tích.

Hiệu suất phụ thuộc vào kỹ thuật tinh chỉnh: Nếu không tinh chỉnh đúng cách hoặc không chọn được siêu tham số phù hợp, mô hình PhoBERT có thể không phát huy tối đa hiệu quả. Việc tinh chỉnh cũng đòi hỏi kiến thức chuyên môn và kinh nghiệm trong lĩnh vực học sâu.

2.8 Tổng quan về FastAPI

2.8.1 Khái niệm

FastAPI là một framework hiện đại và hiệu suất cao được sử dụng để xây dựng các ứng dụng web và API bằng ngôn ngữ lập trình Python. Ra đời với mục tiêu tận dụng các tính năng mạnh mẽ của Python, FastAPI đã nhanh chóng trở thành một trong những công cụ phổ biến nhất trong việc triển khai các dịch vụ web hiện đại, đặc biệt trong lĩnh vực trí tuệ nhân tạo và học máy, nơi hiệu suất và khả năng mở rộng là yếu tố quan trọng.

FastAPI được phát triển bởi Sebastián Ramírez và lần đầu ra mắt vào năm 2018. Framework này được thiết kế với các tiêu chí: tốc độ, dễ sử dụng, hỗ trợ tốt cho kiểm thử, và khả năng tự động sinh tài liệu API.

FastAPI được xây dựng trên nền tảng của hai thư viện mạnh mẽ: **Starlette** thư viện hỗ trợ xây dựng các ứng dụng web bất đồng bộ hiệu suất cao và **Pydantic** thư viện để xác thực và phân tích dữ liệu đầu vào dựa trên type annotation. Sự kết hợp này cho phép FastAPI vừa đạt được hiệu năng cao, vừa dễ sử dụng và thân thiện với lập trình viên.

2.8.2 Đặc điểm nổi bật

Hiệu suất cao: FastAPI có hiệu suất gần tương đương với NodeJS, nền tảng nổi tiếng về tốc độ, nhờ tận dụng cơ chế bất đồng bộ (asynchronous I/O) của Python thông qua async/await.

Tự động sinh tài liệu API: FastAPI hỗ trợ tự động sinh tài liệu API theo chuẩn OpenAPI (Swagger), giúp dễ dàng kiểm thử và tích hợp API. Tài liệu này được tạo dựa trên khai báo kiểu dữ liệu các tham số đầu vào và đầu ra, hoàn toàn tự động và đồng bộ với mã nguồn.

Xác thực và phân tích dữ liệu mạnh mẽ với Pydantic: Nhờ sử dụng thư viện Pydantic, FastAPI cho phép xác thực và chuyển đổi dữ liệu đầu vào dựa trên các kiểu dữ liệu rõ ràng, đồng thời hỗ trợ kiểm tra lỗi, cung cấp phản hồi chi tiết và giúp lập trình viên phát hiện lỗi sớm.

Hỗ trợ type hints giúp tăng năng suất: FastAPI tận dụng triệt để type hints trong Python để kiểm tra dữ liệu và sinh mã tài liệu, đồng thời tăng khả năng tương thích với các công cụ như VSCode, PyCharm, giúp lập trình viên có thể tự động gợi ý cú pháp và phát hiện lỗi sớm trong quá trình viết mã.

Tương thích tốt với các thư viện Machine Learning: FastAPI thường được sử dụng để triển khai các mô hình học máy, học sâu và xử lý ngôn ngữ tự nhiên, nhờ khả năng tích hợp mượt mà với các thư viện như TensorFlow, PyTorch, Scikit-learn, và Transformers.

2.8.3 Ưu nhược điểm của FastAPI

2.8.3.1 Ưu điểm

Hiệu suất cao và hỗ trợ bất đồng bộ: FastAPI được xây dựng trên nền Starlette, một framework bất đồng bộ tốc độ cao, nhờ vậy có thể xử lý hàng nghìn request mỗi giây, rất phù hợp với các hệ thống có lưu lượng truy cập lớn hoặc yêu cầu thời gian phản hồi nhanh.

Tự động sinh tài liệu API: Một trong những điểm mạnh nổi bật của FastAPI là khả năng tự động sinh tài liệu API theo chuẩn OpenAPI và ReDoc. Điều này không chỉ giúp người phát triển dễ dàng kiểm thử API mà còn hỗ trợ việc tích hợp với các hệ thống bên ngoài một cách trực quan và hiệu quả.

Hỗ trợ kiểm tra kiểu dữ liệu đầu vào: Với sự hỗ trợ của thư viện Pydantic, FastAPI cho phép khai báo và kiểm tra kiểu dữ liệu một cách rõ ràng, giúp giảm thiểu lỗi trong quá trình xử lý yêu cầu, đồng thời tăng độ an toàn và tin cậy cho hệ thống.

Phù hợp với các mô hình học máy: FastAPI là sự lựa chọn lý tưởng cho việc triển khai các mô hình học máy như phân loại cảm xúc, nhận diện hình ảnh, chatbot,... vì nó hỗ trợ xử lý JSON, tích hợp dễ dàng với các thư viện như Scikit-learn, PyTorch, TensorFlow,…

2.8.3.2 Nhược điểm

Cộng đồng người dùng chưa lớn: Mặc dù FastAPI đang ngày càng phổ biến, nhưng so với các framework lâu đời như Django hay Flask, hệ sinh thái tài liệu, thư viện mở rộng và diễn đàn hỗ trợ còn hạn chế, đặc biệt với các tính năng phức tạp như quản lý cơ sở dữ liệu hoặc giao diện quản trị.

Thiếu hệ thống quản trị tích hợp sẵn: Khác với Django, FastAPI không tích hợp sẵn hệ thống quản trị, nên nếu cần xây dựng các công cụ quản trị nội bộ, nhà phát triển phải tự xây dựng từ đầu hoặc tích hợp thêm các giải pháp bên ngoài.

Khó khăn khi phát triển ứng dụng quy mô lớn: Khi xây dựng các ứng dụng phức tạp với nhiều tầng logic và mô-đun, FastAPI yêu cầu người phát triển phải tự tổ chức cấu trúc mã nguồn hợp lý. Nếu không có kinh nghiệm, điều này có thể dẫn đến khó bảo trì và mở rộng hệ thống.

Đòi hỏi kiến thức về bất đồng bộ: FastAPI khuyến khích sử dụng lập trình bất đồng bộ (async/await), tuy mang lại hiệu suất cao nhưng cũng có thể gây khó khăn cho người mới bắt đầu.

2.9 Tổng quan về NextJS

2.9.1 Khái niệm

….

2.9.2 Đặc điểm nổi bật của NextJS

….

2.9.3 Ưu nhược điểm của NextJS

2.9.3.1 Ưu điểm

…

2.9.3.2 Nhược điểm

…

CHƯƠNG 3. HUẤN LUYỆN MÔ HÌNH

3.1 Thu thập và tiền xử lý dữ liệu

3.1.1 Thu thập dữ liệu

Để phục vụ cho việc huấn luyện mô hình phân loại cảm xúc tiếng Việt, bước đầu tiên và quan trọng là tiến hành thu thập một tập dữ liệu thực tế, phản ánh đúng các biểu hiện cảm xúc của người dùng trong các tình huống giao tiếp. Trong đề tài này, dữ liệu được thu thập từ trang thương mại điện tử Tiki.vn, nơi khách hàng thường để lại các đánh giá và nhận xét về sản phẩm đã mua, một nguồn dữ liệu phong phú và mang tính cảm xúc rõ rệt, nhất là đối với ngôn ngữ tiếng Việt.

Quá trình thu thập dữ liệu được thực hiện bằng ngôn ngữ lập trình Python, với sự hỗ trợ của thư viện **requests** để gửi các yêu cầu HTTP đến API của trang Tiki và lấy dữ liệu trả về dưới dạng JSON. Dữ liệu sau đó được trích ra, chỉ lấy nội dung các bình luận, sau đó lưu trữ dưới dạng tệp .csv để thuận tiện cho việc xử lý sau này.

3.1.1.1 Các bước thực hiện

Quá trình thu thập dữ liệu được thực hiện theo các bước tuần tự, đảm bảo độ đầy đủ, phong phú và tính hiệu quả của tập dữ liệu. Cụ thể, các bước thực hiện được chia thành hai bước chính: thu thập danh sách sản phẩm từ các danh mục trên Tiki và thu thập các bình luận của người dùng từ từng sản phẩm đó. Dưới đây là mô tả chi tiết cho từng bước trong quy trình:

Bước 1: Thu thập danh sách sản phẩm từ các danh mục

Tiki tổ chức sản phẩm theo hệ thống phân cấp danh mục, với các danh mục lớn như "Sách", "Điện thoại - Máy tính bảng", "Thời trang", "Điện gia dụng",... và các danh mục con nằm bên trong. Việc lựa chọn đa dạng các danh mục giúp đảm bảo sự phong phú trong tập dữ liệu, bao phủ nhiều lĩnh vực, ngữ cảnh và hành vi tiêu dùng khác nhau.

Để truy vấn danh sách sản phẩm thuộc một danh mục cụ thể, Tiki có một API với phương thức GET, ta sử dụng một biến trong python để lưu lại để tiện sử dụng:

```
url_category = "https://tiki.vn/api/personalish/v1/blocks/listings"
```

Khi gọi đến api này để lấy dữ liệu thì ta cần truyền vào các tham số sau:

```
category_params = {
        "limit": 40,
        "include": "advertisement",
        "aggregations": 2,
        "version": "home-persionalized",
        "category": 8322,
        "page": 1,
        "urlKey": "nha-sach-tiki"
}
```

Trong đó có các tham số quan trọng:

limit: là số lượng sản phẩm lấy trên mỗi trang.

page: là số thứ tự của trang hiện tại.

category: là id của danh mục.

Còn có một số tham số khác như: version, include, aggregations,…

Bên cạnh đó cần gửi kèm theo các headers quan trọng và là yêu cầu bắt buộc, bao gồm:

```
headers = {
    "Content-Type": "application/json",
    "Accept": "application/json, text/plain, */*"
}
```

Vì ta cần thu thập dữ liệu từ nhiều danh mục khác nhau để tăng sự phong phú và đa dạng của dữ liệu, nên ta cần lưu id của các danh mục vào một danh sách

để tiện truy vấn, thu thập id danh mục theo một cách thủ công từng danh mục dựa trên đường dẫn của tiki có dạng:

```
https://tiki.vn/url_key/cxxx
```

Trong đó:

url_key: là tham số urlKey trong API lấy sản phẩm từ danh mục ở trên đã giới thiệu.

cxxx: có nghĩa là category xxx và số xxx là id của danh mục đó.

Ví dụ danh mục "Nhà sách Tiki" sẽ có url_key là "nha-sach-tiki" và id tương ứng là 8322 sẽ có dạng:

```
https://tiki.vn/nha-sach-tiki/c8322
```

Ta thu thập thủ công tuần tự qua các danh mục, sau đó lưu nó vào danh sách trong python với định dạng:

```
category_ids = [
    {"id": 8322, "url_key": "sach"},
    {"id": 1883, " url_key": "nha-cua-doi-song"},
    {"id": 1789, " url_key": "dien-thoai-may-tinh-bang"},
    {"id": 15078, " url_key": "cham-soc-nha-cua"},
    …
]
```

Vì đây là danh sách nên ta sử dụng vòng lập để duyệt qua từng danh mục, sau đó sử dụng thư viện requests của python để gọi tới API, mục tiêu là lấy thông tin 40 sản phẩm trên trang đầu của từng danh mục:

```
for category in categories:
  category_params = {
    "limit": 40, # lấy tối đa 40 sản phẩm
    "include": "advertisement",
```

```
    "aggregations": 2,
    "version": "home-persionalized",
    "category": category["id"], # id danh mục
    "page": 1, # trang đầu tiền
    "urlKey": category["url_key"] # url_key của danh mục
  }
  # sử dụng thư viện request để gọi api với phương thức GET
  response_category = requests.get(url_category,
headers=headers, params=category_params)
```

Sau khi gửi yêu cầu HTTP GET tới API, phản hồi trả về là dữ liệu JSON chứa thông tin của các sản phẩm, bao gồm: id, name, thumbnail_url, rating_average, review_count,... Ở đây, quan trọng nhất là các Id sản phẩm và sẽ được lưu lại vào thành một danh sách để sử dụng ở bước tiếp theo:

```
for category in categories:
  … # mã nguồn gọi API bên trên

  if response_category.status_code == 200:
    # trường data trong dữ liệu trả về là danh sách thông tin
sản phẩm
    product_data = response_category.json()["data"]
    # Lấy danh sách product_id
    product_ids = [item["id"] for item in product_data]
```

3.1.1.2 Kết quả thu thập

3.1.2 Tiền xử lý dữ liệu

3.1.3 Gán nhãn dữ liệu

3.2 Trích xuất đặc trưng văn bản

CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG THỬ NGHIỆM

CHƯƠNG 5. ĐÁNH GIÁ VÀ KẾT LUẬN

TÀI LIỆU THAM KHẢO

PHỤ LỤC