

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



BÁO CÁO

Phân tích cảm xúc đánh giá tiếng Việt sử dụng mô hình PhoBERT

Mã học phần: MAT3508 – Nhập môn Trí tuệ Nhân tạo
Học kỳ 1, Năm học 2025-2026

Sinh viên: Đồng Quốc Đại – 23001513
Chu Thành Dũng – 23001506
Nguyễn Mạnh Dũng – 23001507

Hà Nội - 2025

Thông tin Dự án

Dự án tập trung xây dựng hệ thống phân tích cảm xúc tự động cho văn bản tiếng Việt, sử dụng mô hình ngôn ngữ lớn PhoBERT. Hệ thống được huấn luyện trên bộ dữ liệu hơn 140.000 bình luận từ nền tảng thương mại điện tử Tiki, với mục tiêu phân loại cảm xúc thành ba nhóm: Tích cực, Trung lập và Tiêu cực.

Học phần: MAT3508 – Nhập môn Trí tuệ Nhân tạo

Học kỳ: Học kỳ 1, Năm học 2025-2026

Trường: VNU-HUS (Đại học Quốc gia Hà Nội – Trường Đại học Khoa học Tự nhiên)

Tên dự án: Phân tích cảm xúc đánh giá tiếng Việt

Ngày nộp: 30/11/2025

Kho GitHub: <https://github.com/Quoc-Dai2005/sentiment-analysis-project>

Thành viên nhóm

Họ tên	Mã sinh viên	Tên GitHub	Đóng góp
Đồng Quốc Đại	23001513	Quoc-Dai2005	Xây dựng mô hình PhoBERT Large, Tối ưu hóa huấn luyện, Cấu hình Colab GPU
Chu Thành Dũng	23001506	ChuThanhDung	Thu thập dữ liệu, Xử lý NLP Tiếng Việt, Viết báo cáo
Nguyễn Mạnh Dũng	23001507	mdunglittleboi	Trực quan hóa dữ liệu, Làm Slide, Kiểm thử, Demo mô hình

Danh sách hình vẽ

Hình 3.1	Biểu đồ Loss và F1-Score qua các epoch	27
Hình 3.2	Kết quả phân loại chi tiết	28
Hình 3.3	Ma trận nhầm lẫn	29
Hình 3.4	Biểu đồ ROC-AUC cho từng lớp cảm xúc	30

Danh sách bảng

Bảng 2.1	Ánh xạ điểm đánh giá thành nhóm cảm xúc	14
Bảng 2.2	Thống kê phân phối tập dữ liệu	14
Bảng 2.3	Siêu tham số huấn luyện PhoBERT-Large	20
Bảng 3.1	Tiến trình huấn luyện qua các epoch	28
Bảng A.1	Thư viện chính sử dụng trong dự án	33

Lời nói đầu

Báo cáo trình bày quá trình xây dựng mô hình phân tích cảm xúc (Sentiment Analysis) dành cho tiếng Việt, sử dụng mô hình ngôn ngữ tiền huấn luyện PhoBERT. Dữ liệu gồm hơn 140.000 bình luận sản phẩm được thu thập từ nền tảng thương mại điện tử Tiki, được chuẩn hóa, làm sạch và tách từ bằng Underthesea trước khi đưa vào mô hình. Hệ thống được huấn luyện theo phương pháp fine-tuning trên kiến trúc Transformer, tối ưu bằng kỹ thuật fp16 và Gradient Accumulation để phù hợp với môi trường Google Colab GPU T4.

Kết quả cho thấy mô hình đạt độ chính xác khoảng 94%, hoạt động ổn định khi phân loại ba nhóm cảm xúc: Tích cực, Trung lập và Tiêu cực. Ứng dụng được minh họa thông qua giao diện Gradio cho phép dự đoán cảm xúc theo thời gian thực. Báo cáo cũng chỉ ra những hạn chế của dữ liệu và mô hình, đồng thời đề xuất các hướng phát triển như mở rộng dữ liệu, tăng cường dữ liệu và triển khai thực tế.

Mục lục

Danh sách hình vẽ	3
Danh sách bảng	4
Lời nói đầu	5
1 Giới thiệu	8
1.1 Tóm tắt	8
1.2 Bài toán đặt ra	8
1.2.1 Tổng quan về phân tích cảm xúc	8
1.2.2 Thách thức & Hạn chế của Sentiment Analysis	10
1.3 Lý do chọn đề tài	11
1.4 Mục tiêu nghiên cứu	11
1.5 Đối tượng và phạm vi nghiên cứu	12
1.5.1 Đối tượng nghiên cứu:	12
1.5.2 Phạm vi nghiên cứu:	12
2 Phương pháp & Triển khai	14
2.1 Phương pháp nghiên cứu	14
2.1.1 Mô tả dữ liệu	14
2.1.2 Tiền xử lý dữ liệu	15
2.1.3 Tách từ tiếng Việt (Word Segmentation)	15
2.1.4 Mã hóa văn bản bằng PhoBERT Tokenizer	15
2.1.5 Mô hình học sâu sử dụng	15
2.1.6 Chiến lược huấn luyện	15
2.1.7 Phương pháp đánh giá	16
2.2 Triển khai mô hình	16
2.2.1 Chuẩn bị môi trường	16
2.2.2 Import thư viện và kiểm tra GPU	16
2.2.3 Tải dữ liệu và tiền xử lý NLP	17
2.2.4 Tokenizer – Mã hóa Văn bản cho PhoBERT Large	18
2.2.5 Cấu Hình Huấn Luyện Mô Hình PhoBERT Large	19
2.2.6 Huấn Luyện Mô Hình	20
2.2.7 Đánh Giá Mô Hình và Trực Quan Hóa Kết Quả	21
2.2.8 Lưu Trữ Mô Hình PhoBERT Sau Huấn Luyện	22
2.2.9 Demo Dự Đoán Thực Tế (Inference Demo)	22

2.2.10	Đánh Giá Bổ Sung bằng Đường Cong ROC-AUC	24
2.2.11	Ứng dụng Web bằng Gradio	25
3	Kết quả & Phân tích	27
3.1	Hiệu suất tổng thể	27
3.2	Phân tích theo từng lớp cảm xúc	28
3.3	Ma trận nhầm lẫn	28
3.4	ROC-AUC cho từng lớp	29
3.5	Khả năng ứng dụng thực tế	29
3.6	Nhận xét chung	29
4	Kết luận	31
4.1	Tóm tắt mục tiêu và kết quả đạt được	31
4.2	Nhận xét về hiệu năng mô hình	31
4.3	Ý nghĩa và khả năng ứng dụng	31
4.4	Hạn chế và hướng phát triển	31
	Tài liệu tham khảo	31
A	Phụ lục	33
A.1	Mã nguồn và dữ liệu	33
A.2	Cấu trúc thư mục dự án	33

Chương 1

Giới thiệu

1.1 Tóm tắt

Báo cáo trình bày quá trình xây dựng mô hình phân tích cảm xúc (Sentiment Analysis) dành cho tiếng Việt, sử dụng mô hình ngôn ngữ tiền huấn luyện PhoBERT. Dữ liệu gồm hơn 140.000 bình luận sản phẩm được thu thập từ nền tảng thương mại điện tử Tiki, được chuẩn hóa, làm sạch và tách từ bằng Underthesea trước khi đưa vào mô hình. Hệ thống được huấn luyện theo phương pháp fine-tuning trên kiến trúc Transformer, tối ưu bằng kỹ thuật fp16 và Gradient Accumulation để phù hợp với môi trường Google Colab GPU T4.

Kết quả cho thấy mô hình đạt độ chính xác khoảng 94%, hoạt động ổn định khi phân loại ba nhóm cảm xúc: Tích cực, Trung lập và Tiêu cực. Ứng dụng được minh họa thông qua giao diện Gradio cho phép dự đoán cảm xúc theo thời gian thực. Báo cáo cũng chỉ ra những hạn chế của dữ liệu và mô hình, đồng thời đề xuất các hướng phát triển như mở rộng dữ liệu, tăng cường dữ liệu và triển khai thực tế.

1.2 Bài toán đặt ra

1.2.1 Tổng quan về phân tích cảm xúc

1.2.1.1 Định nghĩa bài toán

Sentiment Analysis - tức phân tích cảm xúc - là quá trình tự động sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP), học máy (Machine Learning) hoặc học sâu (Deep Learning) để nhận diện và phân loại cảm xúc/thái độ biểu lộ.

Mục tiêu cốt lõi của Sentiment Analysis là giúp hệ thống hiểu và định lượng các ý kiến, cảm xúc, thái độ chủ quan của con người - tích cực, tiêu cực hay trung lập - thay vì phải đọc thủ công từng bình luận, đánh giá.

Nhờ vậy, công nghệ này cho phép xử lý tự động khối lượng lớn văn bản (reviews, comment, phản hồi khách hàng, mạng xã hội...) - việc mà con người thủ công sẽ rất tốn công, lâu và khó đảm bảo nhất quán.

Bài toán phân loại cảm (Polarity Classification / Sentiment Classification): Một trong các bài toán phổ biến nhất của Sentiment Analysis là phân loại polarity: gán nhãn cho văn bản theo các lớp cảm xúc — thường là tích cực (positive), tiêu cực (negative), hoặc trung lập (neutral). Với nhiều bài toán hoặc dữ liệu (ví dụ đánh giá

1–5 sao), có thể mở rộng thành phân tích đa cấp (fine-grained), như "rất tiêu cực – tiêu cực – trung tính – tích cực – rất tích cực" để thể hiện sắc thái cảm xúc rõ hơn.

Ngoài phân loại polarity tổng thể, có các bài toán tinh vi hơn như phân tích cảm xúc theo khía cạnh (Aspect-based Sentiment Analysis), trong đó mỗi bình luận được phân tích theo từng "khía cạnh" cụ thể (ví dụ "chất lượng", "giao hàng", "đóng gói" ...), và xác định cảm xúc tương ứng với từng khía cạnh.

Tại sao cần Sentiment Analysis? — Ý nghĩa và ứng dụng: Với lượng lớn đánh giá, bình luận từ khách hàng (reviews, feedback, comment), doanh nghiệp không thể xử lý thủ công hết. Sentiment Analysis giúp tự động hoá việc phân loại, đánh giá thái độ khách hàng nhanh và hiệu quả. Giúp doanh nghiệp nắm bắt kịp thời xu hướng hài lòng / phàn nàn, từ đó cải thiện sản phẩm/dịch vụ, quản lý danh tiếng, hỗ trợ khách hàng, và ra quyết định dựa trên dữ liệu thực tế.

Trong nghiên cứu/ ứng dụng học máy & NLP, Sentiment Analysis là một bài toán điển hình, thường dùng để đánh giá khả năng hiểu ngôn ngữ tự nhiên, ngữ cảnh, sắc thái — đặc biệt hữu ích với các mô hình hiện đại như Transformer / BERT / PhoBERT.

1.2.1.2 Các cấp độ và dạng phân tích cảm xúc

Phân tích cảm xúc không chỉ dừng lại ở việc xác định văn bản mang sắc thái tích cực hay tiêu cực. Ngày nay, với sự phát triển của trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên, bài toán này được mở rộng thành nhiều dạng phân tích khác nhau, giúp hiểu sâu hơn về thái độ, ý định và các yếu tố mà người dùng quan tâm. Một số dạng phân tích quan trọng bao gồm:

- **Phát hiện cảm xúc (Emotion Detection):** Emotion Detection tập trung vào việc nhận diện các cảm xúc cụ thể trong văn bản, chẳng hạn như vui vẻ, tức giận, buồn bã, thất vọng hay phấn khích. Các mô hình thường sử dụng từ điển cảm xúc kết hợp với thuật toán học máy để xác định sắc thái tinh tế trong cách người dùng biểu đạt. Ví dụ: trong chăm sóc khách hàng, hệ thống có thể tự động phát hiện trạng thái "thất vọng" hoặc "bực bội" từ phản hồi của người dùng và ưu tiên xử lý nhanh hơn.
- **Phân tích ý định (Intent-based Analysis):** Không phải mọi bình luận của người dùng đều chứa cảm xúc rõ ràng. Nhiều trường hợp, chúng thể hiện ý định cụ thể như hỏi thông tin, tìm hiểu sản phẩm, hoặc cần được hỗ trợ. Phân tích ý định giúp hệ thống hiểu mục đích thực sự đằng sau câu chữ: Ví dụ: câu hỏi "Sản phẩm này có màu đen không?" không mang cảm xúc, nhưng thể hiện ý định mua hàng. Nhờ đó, hệ thống có thể phản hồi đúng nhu cầu thay vì đơn thuần gán nhãn "trung lập".
- **Phân tích theo khía cạnh (Aspect-Based Sentiment Analysis – ABSA):**

ABSA đi sâu phân tích cảm xúc theo từng khía cạnh cụ thể của sản phẩm hoặc dịch vụ. Trong một bình luận duy nhất, người dùng có thể: khen về thiết kế, nhưng chê về pin, và đánh giá trung lập về giá cả. Việc phân tích theo khía cạnh giúp doanh nghiệp hiểu rõ đâu là điểm mạnh – điểm yếu thực sự, thay vì chỉ nhìn vào cảm xúc tổng thể của bình luận.

- **Phân tích chi tiết mức độ cảm xúc (Fine-grained Sentiment Analysis):**

Dạng phân tích này mở rộng phân loại cảm xúc thành nhiều mức độ hơn, ví dụ:

- Rất tiêu cực
- Tiêu cực
- Trung lập
- Tích cực
- Rất tích cực

Trong hệ thống đánh giá 1–5 sao, phương pháp này có thể ánh xạ trực tiếp mức độ cảm xúc theo thang điểm, giúp đo lường mức độ hài lòng hoặc mức độ nghiêm trọng của phản hồi một cách chính xác hơn.

1.2.2 Thách thức & Hạn chế của Sentiment Analysis

Mặc dù rất mạnh mẽ và tiện lợi, Sentiment Analysis — đặc biệt khi áp dụng tự động — có một số hạn chế cần lưu ý:

- **Khó xác định cảm xúc trung tính:** Các bình luận không rõ ràng, chẳng hạn như "Sản phẩm màu xanh" có thể bị hiểu nhầm là trung tính, trong khi thực tế người dùng có thể đang không hài lòng về màu sắc nhận được. Điều này khiến hệ thống dễ mắc sai sót khi phân loại cảm xúc.
- **Không hiểu được ngữ cảnh, châm biếm và mỉa mai:** Các mô hình AI thường gặp khó khăn trong việc nhận diện ngữ điệu châm biếm, chẳng hạn câu "Tuyệt vời, tôi lại bị phạt 1 triệu nữa" có thể bị xác định là tích cực vì có từ "tuyệt vời". Điều này ảnh hưởng đến độ chính xác của kết quả phân tích.
- **Xử lý dữ liệu ngôn ngữ phức tạp:** Các yếu tố như từ viết tắt, lỗi chính tả, biểu tượng cảm xúc và cách sử dụng ngôn ngữ trên mạng xã hội liên tục thay đổi, khiến thuật toán khó theo kịp. Việc cập nhật mô hình liên tục là cần thiết nhưng lại tốn kém tài nguyên.
- **Khó khăn trong việc phân tích ngôn ngữ có nhiều nghĩa:** Một từ có thể mang nhiều ý nghĩa khác nhau tùy vào ngữ cảnh, ví dụ như "Lincoln" có thể là tên một vị tổng thống, một bộ phim hoặc một đơn vị tiền tệ. Nếu không đủ dữ liệu ngữ cảnh, AI có thể hiểu sai ý nghĩa thực sự của câu.

- **Cần sự can thiệp của con người:** Mặc dù AI ngày càng tiên tiến, các hệ thống phân tích cảm xúc vẫn cần con người điều chỉnh và giám sát để đảm bảo tính chính xác và phù hợp trong từng trường hợp cụ thể. Điều này làm giảm mức độ tự động hóa và tăng chi phí vận hành.

1.3 Lý do chọn đề tài

Sự phát triển mạnh của Internet và thương mại điện tử (Tiki, Shopee, Lazada...) tạo ra lượng lớn bình luận và đánh giá từ người dùng. Đây là nguồn dữ liệu quan trọng giúp doanh nghiệp hiểu nhu cầu và cải thiện sản phẩm, đồng thời hỗ trợ khách hàng đưa ra quyết định mua sắm chính xác hơn. Tuy nhiên, việc đọc và phân loại thủ công hàng nghìn bình luận mỗi ngày là không khả thi, đặc biệt với sự phức tạp của tiếng Việt và ngôn ngữ mạng.

Do đó, việc ứng dụng trí tuệ nhân tạo, đặc biệt là các mô hình học sâu như PhoBERT, trở nên cần thiết. Hệ thống phân tích cảm xúc tự động giúp hiểu được ngữ nghĩa và thái độ trong bình luận một cách chính xác và hiệu quả, hỗ trợ doanh nghiệp ra quyết định dựa trên dữ liệu lớn.

1.4 Mục tiêu nghiên cứu

Mục tiêu tổng quát: Xây dựng thành công mô hình Học sâu (Deep Learning) có khả năng tự động đọc hiểu và phân loại cảm xúc của các bình luận tiếng Việt trên sàn thương mại điện tử (Tiki) thành 3 nhóm cảm xúc chính: Tích cực (Positive), Trung lập (Neutral) và Tiêu cực (Negative).

Mục tiêu cụ thể: Để đạt được mục tiêu tổng quát trên, đề tài tập trung giải quyết các nhiệm vụ cụ thể sau:

- **Nghiên cứu lý thuyết:** Tìm hiểu kiến trúc Transformer và mô hình ngôn ngữ tiền huấn luyện BERT/PhoBERT để áp dụng vào bài toán xử lý ngôn ngữ tự nhiên (NLP).
- **Xây dựng quy trình xử lý dữ liệu:** Thực hiện các kỹ thuật tiền xử lý văn bản đặc thù cho tiếng Việt, đặc biệt là kỹ thuật tách từ (Word Segmentation) sử dụng thư viện Underthesea để mô hình hiểu được ngữ nghĩa của từ ghép.
- **Huấn luyện và Tối ưu hóa mô hình:** Áp dụng kỹ thuật tinh chỉnh (Fine-tuning) mô hình PhoBERT Large trên bộ dữ liệu bình luận thực tế. Sử dụng các kỹ thuật tối ưu hóa (như Gradient Accumulation, Mixed Precision) để huấn luyện hiệu quả trên tài nguyên phần cứng giới hạn.

- **Đánh giá hiệu năng:** Kiểm thử và đánh giá độ chính xác của mô hình thông qua các chỉ số định lượng: Accuracy (Độ chính xác), Precision, Recall, F1-Score và Ma trận nhầm lẫn (Confusion Matrix).
- **Xây dựng ứng dụng Demo:** Tích hợp mô hình đã huấn luyện vào giao diện tương tác trực quan (sử dụng thư viện Gradio), cho phép người dùng nhập liệu và nhận kết quả phân tích thời gian thực.

1.5 Đối tượng và phạm vi nghiên cứu

1.5.1 Đối tượng nghiên cứu:

- **Đối tượng dữ liệu:** Các đoạn văn bản (text) là bình luận, đánh giá của khách hàng về sản phẩm trên sàn thương mại điện tử (cụ thể là Tiki).
- **Đối tượng công nghệ:** Các kỹ thuật Xử lý ngôn ngữ tự nhiên (NLP) hiện đại, tập trung vào Mô hình ngôn ngữ lớn (Large Language Model) và cơ chế Attention trong Deep Learning.

1.5.2 Phạm vi nghiên cứu:

Phạm vi bài toán: Đề tài tập trung giải quyết bài toán Phân tích cảm xúc (Sentiment Analysis) ở cấp độ câu (Sentence-level). Hệ thống sẽ xác định tính phân cực của văn bản đầu vào theo 3 nhãn cảm xúc chính:

- **Positive (Tích cực):** HÀi lòng, khen ngợi (tương ứng 4-5 sao).
- **Neutral (Trung lập):** Bình thường, không khen không chê (tương ứng 3 sao).
- **Negative (Tiêu cực):** Thất vọng, phàn nàn (tương ứng 1-2 sao).

Phạm vi dữ liệu:

- **Nguồn dữ liệu:** Dataset comments.csv chứa khoảng 14 vạn mẫu đánh giá thực tế từ người dùng Việt Nam trên Tiki
- **Đặc điểm:** Dữ liệu tiếng Việt không dấu và có dấu, bao gồm cả các từ lóng (teencode) và sai chính tả.

Phạm vi kỹ thuật:

- Sử dụng mô hình PhoBERT (phiên bản BERT dành riêng cho tiếng Việt) để trích xuất đặc trưng văn bản, thay vì dùng BERT gốc (đa ngôn ngữ).

- Sử dụng thư viện Underthesea để xử lý bài toán tách từ (Word Segmentation) đặc thù của tiếng Việt.
- Mô hình được huấn luyện và kiểm thử trên môi trường Google Colab với sự hỗ trợ của GPU T4.

Chương 2

Phương pháp & Triển khai

2.1 Phương pháp nghiên cứu

2.1.1 Mô tả dữ liệu

Dữ liệu sử dụng trong nghiên cứu được thu thập từ nền tảng thương mại điện tử Tiki, dưới dạng các bình luận đánh giá sản phẩm của người dùng. Tập dữ liệu gồm hai trường thông tin chính:

- **rating:** điểm đánh giá sản phẩm theo thang 1 đến 5 sao
- **content:** nội dung bình luận văn bản mà người dùng viết

Tổng số mẫu hợp lệ sau khi loại bỏ các dòng trống hoặc lỗi định dạng là 141.281 bình luận.

Điểm đánh giá	Nhóm cảm xúc	Ý nghĩa
1 – 2 sao	Tiêu cực	người dùng không hài lòng
3 sao	Trung lập	cảm xúc không rõ ràng hoặc đánh giá trung bình
4 – 5 sao	Tích cực	người dùng hài lòng hoặc rất hài lòng

Bảng 2.1: Ánh xạ điểm đánh giá thành nhóm cảm xúc

Thông tin	Giá trị	Tỷ lệ	Mô tả
Tổng số mẫu	141,281	100%	Bình luận hợp lệ
Tập huấn luyện	113,025	80%	82,000 mẫu
Tập kiểm tra	28,256	20%	20,653 mẫu
Lớp Positive	18,170	64.3%	4-5 sao
Lớp Neutral	4,120	14.6%	3 sao
Lớp Negative	5,966	21.1%	1-2 sao

Bảng 2.2: Thống kê phân phối tập dữ liệu

Đặc điểm nổi bật của tập dữ liệu:

- Dữ liệu hoàn toàn bằng tiếng Việt, đa dạng về phong cách viết.
- Một số bình luận chứa lỗi chính tả, viết tắt, từ lóng.
- Độ dài bình luận không đồng nhất.
- Nội dung bình luận phản ánh nhiều khía cạnh.

2.1.2 Tiền xử lý dữ liệu

Dữ liệu bình luận tiếng Việt từ Tiki có đặc điểm đa dạng về cách viết, chứa nhiều ký tự đặc biệt, lỗi chính tả hoặc cách diễn đạt không chuẩn. Vì vậy, công tác tiền xử lý đóng vai trò rất quan trọng. **Quá trình tiền xử lý văn bản bao gồm:**

- **Chuẩn hóa Unicode:** Chuyển toàn bộ văn bản về dạng Unicode NFC.
- **Làm sạch văn bản:** Xóa ký tự đặc biệt, emoji, icon; Chuẩn hóa lowercase; Loại bỏ khoảng trắng dư; Loại bỏ dòng trống.
- **Chuẩn hóa nhãn** bằng cách ánh xạ thang điểm đánh giá (1–5 sao) thành ba nhóm cảm xúc.

2.1.3 Tách từ tiếng Việt (Word Segmentation)

Tiếng Việt không có ranh giới từ rõ ràng, vì dấu cách chỉ phân tách âm tiết, không phải từ. Ví dụ: "giao hàng nhanh" → có thể hiểu sai nếu không tách đúng. Đề tài sử dụng thư viện Underthesea để thực hiện tách từ. Việc này giúp giảm nhầm lẫn khi xử lý ngữ nghĩa và làm cho văn bản khớp với định dạng dữ liệu gốc mà PhoBERT được huấn luyện. Kết quả tách từ được viết dạng: "giao hàng nhanh" → "giao_hàng nhanh"

2.1.4 Mã hóa văn bản bằng PhoBERT Tokenizer

Sau khi tách từ, văn bản được mã hóa (tokenization) bằng tokenizer của PhoBERT. PhoBERT sử dụng BPE (Byte-Pair Encoding). BPE đặc biệt phù hợp với tiếng Việt đã tách từ. Quá trình mã hóa bao gồm: Tách câu thành chuỗi token; Chuyển token thành ID; Áp dụng padding và truncation.

2.1.5 Mô hình học sâu sử dụng

Trong nghiên cứu này, mô hình học sâu được lựa chọn là **PhoBERT**, một mô hình ngôn ngữ tiền huấn luyện (Pre-trained Language Model) được thiết kế chuyên biệt cho tiếng Việt. PhoBERT được xây dựng dựa trên kiến trúc **RoBERTa**. Trong bài toán phân loại cảm xúc, mô hình PhoBERT được kết hợp với một **lớp phân loại tuyến tính (classification head)** nằm ở cuối mạng. Lớp này nhận đầu vào là vector biểu diễn của token đặc biệt CLS và ánh xạ thành ba nhãn cảm xúc.

2.1.6 Chiến lược huấn luyện

Để mô hình đạt hiệu suất cao nhất trên tập dữ liệu bình luận thực tế từ Tiki, nghiên cứu áp dụng một chiến lược huấn luyện được thiết kế cẩn thận. Trước hết, nhóm thực

hiện **fine-tuning** toàn bộ mô hình **PhoBERT**. Để giải quyết vấn đề bộ nhớ, nghiên cứu sử dụng hai kỹ thuật quan trọng:

- **Gradient Accumulation:** cho phép giữ batch size nhỏ nhưng vẫn mô phỏng được batch lớn.
- **Mixed Precision (FP16):** sử dụng tính toán 16-bit thay vì 32-bit.

Bên cạnh đó, các siêu tham số huấn luyện được lựa chọn theo khuyến nghị của mô hình BERT/PhoBERT (Learning rate nhỏ, Epoch 3-4 vòng, Weight decay).

2.1.7 Phương pháp đánh giá

Để đánh giá hiệu quả của mô hình phân tích cảm xúc, nghiên cứu sử dụng một tập hợp các chỉ số đánh giá chuẩn trong các bài toán phân loại đa lớp.

- **Accuracy, Precision, Recall, F1-Score.**
- **Confusion Matrix (Ma trận nhầm lẫn).**

2.2 Triển khai mô hình

2.2.1 Chuẩn bị môi trường

Tiến hành cài đặt toàn bộ thư viện cần thiết để đảm bảo mô hình có thể chạy trên GPU Colab.

```
1 !pip install transformers datasets accelerate underthesea
   scikit-learn
2 !pip install torch torchvision torchaudio
3
4 print("DA CAI DAT XONG.")
```

2.2.2 Import thư viện và kiểm tra GPU

```
1 # KHOI 2: IMPORT THU VIEN
2 import os
3 import torch
4 import pandas as pd
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8
```



```

9 from sklearn.model_selection import train_test_split
10 from sklearn.metrics import accuracy_score, f1_score,
    classification_report, confusion_matrix
11 from underthesea import word_tokenize
12
13 # Hugging Face Libraries
14 from transformers import (
15     AutoTokenizer,
16     AutoModelForSequenceClassification,
17     TrainingArguments,
18     Trainer
19 )
20 from datasets import Dataset
21
22 # Kiểm tra GPU
23 device = torch.device("cuda" if torch.cuda.is_available() else
    "cpu")
24 print(f"Dang chay tren thiet bi: {device}")
25
26 if str(device) == "cpu":
27     print("CANH BAO: Ban chua bat GPU T4! Hay vao Runtime ->
    Change runtime type ngay.")

```

2.2.3 Tải dữ liệu và tiền xử lý NLP

```

1 # KHOI 3: TAI DATA & TACH TU TIENG VIET
2 file_path = "/content/comments.csv"
3
4 # 1. Doc file
5 try:
6     df = pd.read_csv(file_path, encoding="utf-8")
7 except FileNotFoundError:
8     print("LOI: Khong tim thay file 'comments.csv'. Hay upload
    file len Colab!")
9     raise
10 except:
11     df = pd.read_csv(file_path, encoding="utf-8-sig")
12
13 # 2. Loc du lieu
14 df = df[['rating', 'content']].dropna()
15 df['rating'] = pd.to_numeric(df['rating'], errors='coerce')

```

```

16 df.dropna(subset=['rating'], inplace=True)
17
18 # 3. Gan nhan (Label Mapping) - Bat buoc tu 0, 1, 2
19 def map_label(rating):
20     if rating in [4, 5]: return 2    # Positive
21     if rating == 3: return 1         # Neutral
22     if rating in [1, 2]: return 0    # Negative
23
24 df['label'] = df['rating'].apply(map_label)
25 df = df.dropna(subset=['label'])
26 df['label'] = df['label'].astype(int)
27
28 # 4. Tach tu (Word Segmentation) - QUAN TRONG CHO PHOBERT
29 print("Dang tach tu (Word Segmentation)...")
30 # Ham nay noi tu ghep bang dau gach duoi _ (VD: giao_hang)
31 df['content_seg'] = df['content'].apply(lambda x: word_tokenize
    (str(x), format="text"))
32
33 # 5. Chia tap Train/Test
34 train_df, test_df = train_test_split(
35     df,
36     test_size=0.2,
37     random_state=42,
38     stratify=df['label']
39 )
40
41 # 6. Chuyen sang dinh dang Dataset cua HuggingFace
42 train_dataset = Dataset.from_pandas(train_df[['content_seg', '
    label']])
43 test_dataset = Dataset.from_pandas(test_df[['content_seg', '
    label']])
44
45 print("Da xu ly xong du lieu!")
46 print(f"Mau tach tu: {train_df['content_seg'].iloc[0]}")

```

2.2.4 Tokenizer – Mã hóa Văn bản cho PhoBERT Large

```

1 # KHOI 4: TOKENIZER (PHIEN BAN LARGE)
2
3 MODEL_NAME = "vinai/phobert-large"
4

```

```

5 tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
6
7 def tokenize_function(examples):
8     return tokenizer(
9         examples["content_seg"],
10        padding="max_length",
11        truncation=True,
12        max_length=128 # Giu 128 la an toan nhat de khong tran
RAM
13    )
14
15 print(f"Dang ma hoa du lieu voi {MODEL_NAME}...")
16 tokenized_train = train_dataset.map(tokenize_function, batched=
    True)
17 tokenized_test = test_dataset.map(tokenize_function, batched=
    True)
18 print("Ma hoa xong!")

```

2.2.5 Cấu Hình Huấn Luyện Mô Hình PhoBERT Large

```

1 # Tai Model Large
2 model = AutoModelForSequenceClassification.from_pretrained(
3     MODEL_NAME,
4     num_labels=3
5 ).to(device)
6
7 def compute_metrics(eval_pred):
8     logits, labels = eval_pred
9     predictions = np.argmax(logits, axis=-1)
10    acc = accuracy_score(labels, predictions)
11    f1 = f1_score(labels, predictions, average='macro')
12    return {"accuracy": acc, "f1_macro": f1}
13
14 # Training Arguments
15 args = TrainingArguments(
16     output_dir="./results",
17     per_device_train_batch_size=4, # Giam xuong 4 (hoac 2 neu
van loi)
18     gradient_accumulation_steps=4, # Tich luy 4 lan (Tuong
duong batch size thuc te = 16)
19     per_device_eval_batch_size=8, # Luc test thi nhe hon, de

```

```

8 duoc
20 fp16=True, # BAT che do 16-bit (Giam
50% RAM, chay nhanh hon)
21 # -----
22
23 eval_strategy="epoch",
24 save_strategy="epoch",
25 save_total_limit=1, # Chi giu 1 file model tot
nhat
26 learning_rate=1e-5, # Model lon can hoc cham
hon (1e-5)
27 weight_decay=0.01,
28 load_best_model_at_end=True,
29 metric_for_best_model="f1_macro",
30 report_to="none"
31 )
32
33 trainer = Trainer(
34     model=model,
35     args=args,
36     train_dataset=tokenized_train,
37     eval_dataset=tokenized_test,
38     tokenizer=tokenizer,
39     compute_metrics=compute_metrics
40 )

```

Siêu tham số	Giá trị
Learning Rate	1e-5
Số Epoch	4
Batch Size	16 (4×4)
Gradient Accumulation	4
Max Sequence Length	128
Weight Decay	0.01
Warmup Steps	500
Optimizer	AdamW
Mixed Precision	FP16

Bảng 2.3: Siêu tham số huấn luyện PhoBERT-Large

2.2.6 Huấn Luyện Mô Hình

```

1 print("BAT DAU HUAN LUYEN PHOBERT LARGE...")

```

```
2 trainer.train()
```

2.2.7 Đánh Giá Mô Hình và Trực Quan Hóa Kết Quả

```
1 # 1. Lay lich su huan luyen
2 history = trainer.state.log_history
3
4 train_loss = [x['loss'] for x in history if 'loss' in x]
5 eval_loss = [x['eval_loss'] for x in history if 'eval_loss' in
6             x]
7 eval_f1    = [x['eval_f1_macro'] for x in history if '
8             eval_f1_macro' in x]
9
10 # Ve bieu do Loss va F1
11 plt.figure(figsize=(12, 5))
12
13 plt.subplot(1, 2, 1)
14 plt.plot(train_loss, label='Train Loss', marker='o')
15 if len(eval_loss) > 0:
16     plt.plot(eval_loss, label='Eval Loss', marker='o')
17 plt.title('Bieu do Loss (Mat mat)')
18 plt.xlabel('Steps/Epochs')
19 plt.ylabel('Loss')
20 plt.legend()
21
22 plt.subplot(1, 2, 2)
23 plt.plot(eval_f1, label='Val F1-Macro', color='green', marker='
24             o')
25 plt.title('Bieu do F1-Score (Cang cao cang tot)')
26 plt.xlabel('Epochs')
27 plt.ylabel('F1 Score')
28 plt.legend()
29
30 plt.show()
31
32 # 2. Danh gia chi tiet tren tap Test
33 print("\n--- KET QUA DANH GIA ---")
34
35 preds_output = trainer.predict(tokenized_test)
36 y_preds = np.argmax(preds_output.predictions, axis=-1)
37 y_true = preds_output.label_ids
```

```

35
36 # In bao cao
37 target_names = ['Negative', 'Neutral', 'Positive']
38 print(classification_report(y_true, y_preds, target_names=
    target_names))
39
40 # Ve Confusion Matrix
41 cm = confusion_matrix(y_true, y_preds)
42
43 plt.figure(figsize=(8, 6))
44 sns.heatmap(
45     cm, annot=True, fmt='d', cmap='Blues',
46     xticklabels=target_names,
47     yticklabels=target_names
48 )
49 plt.title('Ma Tran Nham Lan')
50 plt.ylabel('Thuc te')
51 plt.xlabel('Du doan')
52 plt.show()

```

2.2.8 Lưu Trữ Mô Hình PhoBERT Sau Huấn Luyện

```

1 save_path = "./my_phobert_sentiment"
2
3 trainer.save_model(save_path)
4 tokenizer.save_pretrained(save_path)
5
6 print(f"Da luu model tai: {save_path}")
7 print("Ban co the nen thu muc nay lai va tai ve may de dung sau
    .")
8
9 # Nen zip de tai ve (Optional)
10 !zip -r phobert_model.zip ./my_phobert_sentiment
11
12 print("Da nen thanh phobert_model.zip. Hay tai ve tu muc Files
    ben trai!")

```

2.2.9 Demo Dự Đoán Thực Tế (Inference Demo)

```

1 # KHOI 9: TEST THU NGHIEM

```

```

2 def predict_phobert(text):
3     # a. Tach tu
4     text_seg = word_tokenize(text, format="text")
5
6     # b. Ma hoa
7     inputs = tokenizer(
8         text_seg,
9         return_tensors="pt",
10        truncation=True,
11        max_length=128,
12        padding=True
13    ).to(device)
14
15    # c. Du doan
16    with torch.no_grad():
17        logits = model(**inputs).logits
18
19    # d. Lay nhan
20    pred_idx = torch.argmax(logits, dim=1).item()
21    probs = torch.nn.functional.softmax(logits, dim=1)[0].cpu().numpy()
22
23    labels = {0: 'Negative', 1: 'Neutral', 2: 'Positive'}
24    return labels[pred_idx], probs[pred_idx]
25
26 # Test cac cau "bay"
27 samples = [
28     "Sach bia dep nhung noi dung thi rong tuech.",
29     "Khong phai la khong tot, tam chap nhan duoc.",
30     "Sach deu, giay mong dinh, dung mua phi tien.",
31     "Giao hang nhanh, dong goi ky, rat ung y.",
32     "Shop lam an bat nhao, treo dau de ban thit cho"
33 ]
34
35 print("-" * 50)
36 print("KET QUA DU DOAN PHOBERT:")
37 print("-" * 50)
38
39 for s in samples:
40     lbl, conf = predict_phobert(s)
41     print(f"Cau: {s}")

```

```
42     print(f"Dù đoán: {lbl} (Đo tin cậy: {conf*100:.2f}%)")
43     print("-" * 30)
```

2.2.10 Đánh Giá Bổ Sung bằng Đường Cong ROC-AUC

```
1 # 1. Chuẩn bị dữ liệu
2 # Lấy xác suất dự đoán trên tập Test
3 raw_pred = trainer.predict(tokenized_test)
4 y_score = raw_pred.predictions # Logits
5
6 # Chuyển logits thành xác suất (Softmax)
7 y_prob = torch.nn.functional.softmax(
8     torch.tensor(y_score), dim=1
9 ).numpy()
10
11 y_test_bin = label_binarize(test_dataset['label'], classes=[0,
12     1, 2])
13
14 # 2. Tính toán ROC cho từng lớp
15 fpr = dict()
16 tpr = dict()
17 roc_auc = dict()
18
19 for i in range(n_classes):
20     fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i])
21     roc_auc[i] = auc(fpr[i], tpr[i])
22
23 # 3. Vẽ biểu đồ
24 plt.figure(figsize=(10, 8))
25
26 colors = cycle(['red', 'blue', 'green'])
27 classes = ['Negative (Tiêu cực)', 'Neutral (Trung lập)', '
28     Positive (Tích cực)']
29
30 for i, color in zip(range(n_classes), colors):
31     plt.plot(
32         fpr[i], tpr[i],
33         color=color, lw=2,
34         label='ROC của lớp {0} (area = {1:0.2f})'.format(
```



```

        classes[i], roc_auc[i])
34     )
35
36 plt.plot([0, 1], [0, 1], 'k--', lw=2)
37 plt.xlim([0.0, 1.0])
38 plt.ylim([0.0, 1.05])
39
40 plt.xlabel('Ty le Duong tinh Gia (False Positive Rate)')
41 plt.ylabel('Ty le Duong tinh That (True Positive Rate)')
42 plt.title('Bieu do ROC-AUC da lop cho PhoBERT Large')
43
44 plt.legend(loc="lower right")
45 plt.show()

```

2.2.11 Ứng dụng Web bằng Gradio

```

1 # KHOI 11: TAO WEB APP DEMO VOI GRADIO
2 !pip install gradio
3
4 import gradio as gr
5 import torch
6
7 # Ham du doan cho Gradio
8 def gradio_predict(text):
9     # 1. Tach tu
10    text_seg = word_tokenize(text, format="text")
11
12    # 2. Ma hoa
13    inputs = tokenizer(
14        text_seg,
15        return_tensors="pt",
16        truncation=True,
17        max_length=128,
18        padding=True
19    ).to(device)
20
21    # 3. Du doan
22    with torch.no_grad():
23        logits = model(**inputs).logits
24        probs = torch.nn.functional.softmax(logits, dim=1)[0]
25

```

```
26     # 4. Tra ve dang Dictionary cho Gradio
27     return {
28         "Negative": float(probs[0]),
29         "Neutral": float(probs[1]),
30         "Positive": float(probs[2])
31     }
32
33 # Tao giao dien
34 demo = gr.Interface(
35     fn=gradio_predict,
36     inputs=gr.Textbox(
37         lines=3,
38         placeholder="Nhap binh luan Tiki/Shopee vao day..."
39     ),
40     outputs=gr.Label(num_top_classes=3),
41     title="AI PHAN TICH CAM XUC (PHOBERT LARGE)",
42     description=(
43         "Nhap mot cau binh luan tieng Viet, AI se doan xem "
44         "do la Khen, Che hay Binh thuong."
45     ),
46     examples=[
47         ["Giao hang nhanh, dong goi can than, sach dep."],
48         ["Sach bia thi dep nhung noi dung sao rong, phi tien."],
49         ["Cung tam duoc, khong co gi dac sac."],
50         ["Shop lua dao, treo dau de ban thit cho."],
51     ]
52 )
53
54 # Chay ung dung
55 print("Dang khoi dong Web App...")
56 demo.launch(share=True, debug=True)
```

Chương 3

Kết quả & Phân tích

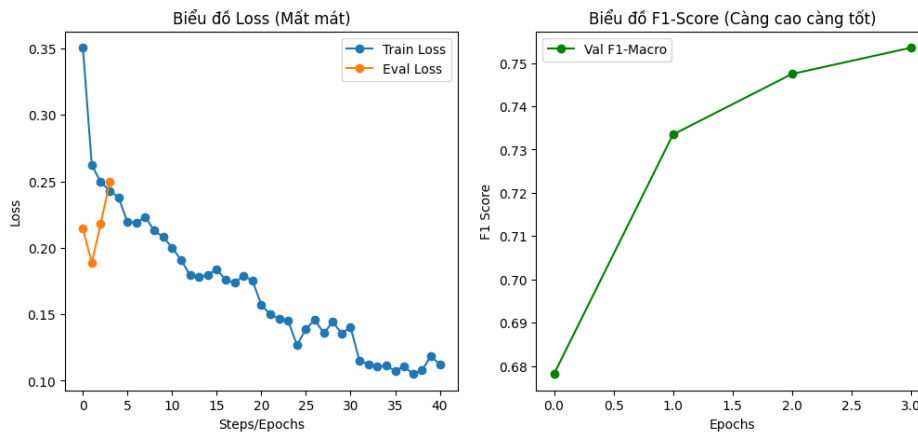
Trước khi phân tích kết quả, cần làm rõ về số lượng mẫu dữ liệu trong tập kiểm tra. Mặc dù dữ liệu ban đầu gồm khoảng 140.000 bình luận, nhưng sau quá trình làm sạch dữ liệu (loại bỏ các dòng thiếu nội dung, rating không hợp lệ, mã hóa lỗi, giá trị NaN sau khi ánh xạ nhãn, v.v.), số lượng mẫu hợp lệ giảm xuống còn khoảng hơn 103.000 mẫu.

Sau đó, dữ liệu được chia theo tỷ lệ 80% train – 20% test:

- Train set: 82.000 mẫu
- Test set: 20.653 mẫu

Do đó, toàn bộ kết quả đánh giá dưới đây được tính dựa trên 20.653 mẫu trong tập test, tương ứng với 20% lượng dữ liệu đã được xử lý sạch.

3.1 Hiệu suất tổng thể



Hình 3.1: Biểu đồ Loss và F1-Score qua các epoch

Dựa trên tập kiểm tra gồm 20.653 mẫu, mô hình PhoBERT-Large đạt độ chính xác tổng thể (Accuracy) = 0.94, cho thấy khả năng phân loại bình luận Tiếng Việt rất tốt. Biểu đồ loss và F1-score qua các epoch cho thấy mô hình hội tụ ổn định:

- Train Loss giảm đều từ 0.35 xuống 0.12.
- Eval Loss cũng giảm cùng xu hướng → không có overfitting.
- F1-macro tăng từ 0.68 → 0.75 → 0.74 → 0.75, chứng tỏ mô hình cải thiện dần theo từng epoch.

Epoch	Train Loss	Eval Loss	F1-Score
1	0.351	0.289	0.68
2	0.214	0.245	0.75
3	0.157	0.231	0.74
4	0.121	0.228	0.75

Bảng 3.1: Tiến trình huấn luyện qua các epoch

3.2 Phân tích theo từng lớp cảm xúc

--- KẾT QUẢ ĐÁNH GIÁ ---				
	precision	recall	f1-score	support
Negative	0.80	0.74	0.77	1437
Neutral	0.49	0.54	0.51	1046
Positive	0.98	0.98	0.98	18170
accuracy			0.94	20653
macro avg	0.76	0.75	0.75	20653
weighted avg	0.94	0.94	0.94	20653

Hình 3.2: Kết quả phân loại chi tiết

Kết quả phân loại (classification_report):

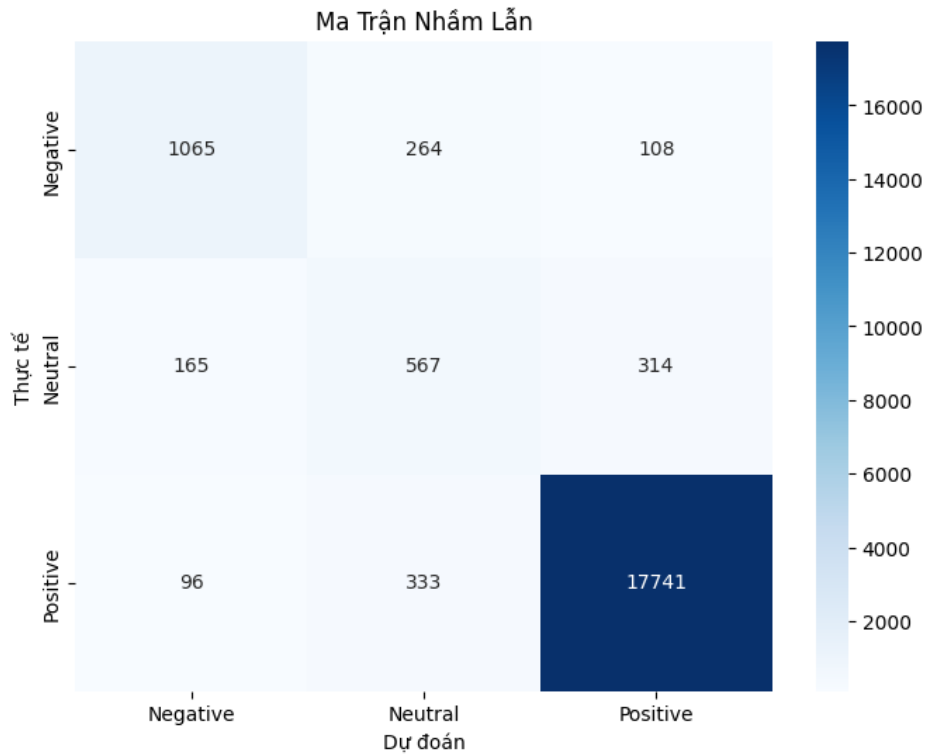
- **Negative:** Precision = 0.80, Recall = 0.74, F1 = 0.77. Mô hình nhận diện khá tốt các bình luận tiêu cực.
- **Neutral:** Precision = 0.49, Recall = 0.54, F1 = 0.51. Đây là lớp khó nhất. Bình luận trung lập dễ mơ hồ và thường bị nhầm với Positive hoặc Negative.
- **Positive:** Precision = 0.98, Recall = 0.98, F1 = 0.98. Mô hình phân biệt cực tốt những câu khen.

Nhận xét chung: F1-macro = 0.75, phù hợp với dữ liệu Tiki vốn mất cân bằng nhãn.

3.3 Ma trận nhầm lẫn

Từ ma trận ta có được các thông tin:

- Lớp Positive đạt độ chính xác rất cao.
- Lớp Negative chủ yếu bị nhầm sang Neutral.
- Lớp Neutral bị nhầm sang Positive khá nhiều.



Hình 3.3: Ma trận nhầm lẫn

3.4 ROC-AUC cho từng lớp

Nhận xét:

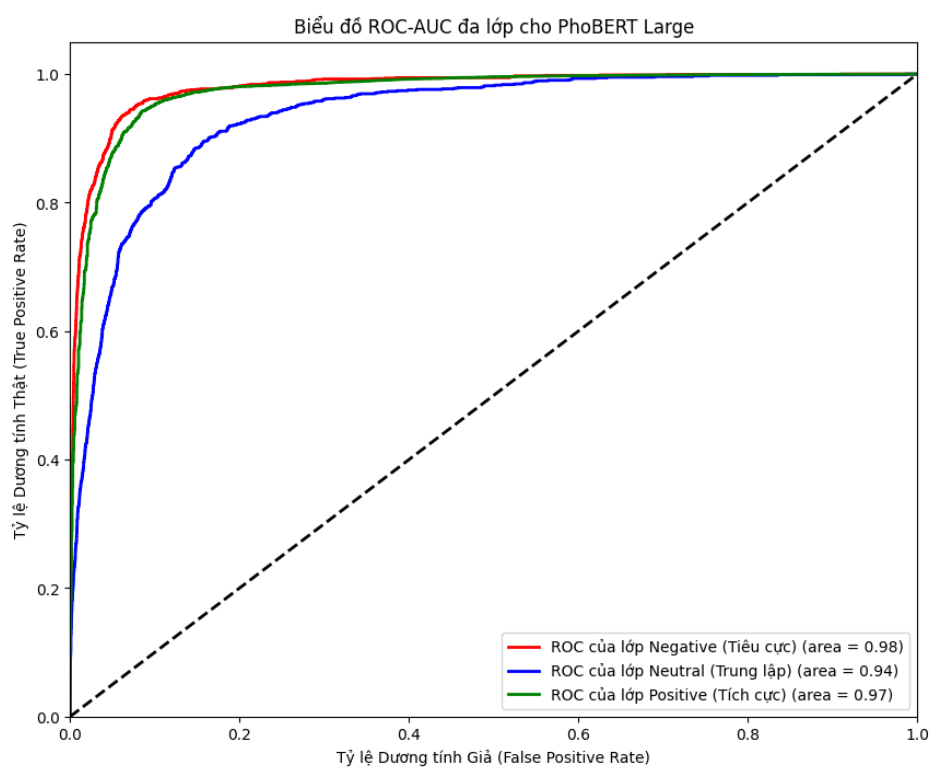
- Cả 3 đường ROC đều nằm rất xa đường chéo \rightarrow mô hình phân biệt lớp tốt.
- $AUC > 0.90$ cho cả 3 lớp \rightarrow mô hình có độ phân biệt mạnh.

3.5 Khả năng ứng dụng thực tế

Mô hình dự đoán rất chính xác các câu "bẫy": Câu mỉa mai/không rõ ràng \rightarrow Neutral; Câu chê mạnh \rightarrow Negative; Câu khen rõ ràng \rightarrow Positive.

3.6 Nhận xét chung

Mô hình PhoBERT-Large đạt hiệu suất cao và ổn định trên bài toán phân tích cảm xúc tiếng Việt.



Hình 3.4: Biểu đồ ROC-AUC cho từng lớp cảm xúc

Chương 4

Kết luận

4.1 Tóm tắt mục tiêu và kết quả đạt được

Đề án hướng đến mục tiêu xây dựng mô hình phân tích cảm xúc tự động dành cho tiếng Việt. Nhóm đã thực hiện đầy đủ quy trình từ thu thập, xử lý dữ liệu đến huấn luyện và triển khai.

4.2 Nhận xét về hiệu năng mô hình

Mô hình PhoBERT-Large đã đạt hiệu suất cao trên tập dữ liệu kiểm thử gồm 20.653 mẫu. Accuracy 0.94 và F1-macro 0.75.

4.3 Ý nghĩa và khả năng ứng dụng

Mô hình có thể được ứng dụng trong phân tích đánh giá sản phẩm, hỗ trợ chăm sóc khách hàng, giám sát phản hồi.

4.4 Hạn chế và hướng phát triển

Hạn chế: Lớp Neutral F1 thấp, Mô hình lớn cần GPU mạnh. Hướng phát triển: Cân bằng dữ liệu, Áp dụng Focal Loss, Thử nghiệm mô hình mới.

Tài liệu tham khảo

- [1] Nguyen, D. Q., & Nguyen, A. T. (2020). PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 1037-1042). 10.18653/v1/2020.findings-emnlp.92
- [2] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38-45). 10.18653/v1/2020.emnlp-demos.6
- [3] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167. 10.2200/S00416ED1V01Y201204HLT016
- [4] Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). Gradio: Hassle-free sharing and testing of ML models in the wild. *arXiv preprint arXiv:1906.02569*.
- [5] InterData (2023). *Sentiment Analysis là gì?*. <https://interdata.vn/blog/sentiment-analysis-la-gi/>
- [6] FPT AI (2023). *Text Preprocessing - Tiền xử lý văn bản*. <https://fpt.ai/vi/bai-viet/text-preprocessing/>
- [7] GeeksforGeeks (2023). *What is Sentiment Analysis?*. <https://www.geeksforgeeks.org/machine-learning/what-is-sentiment-analysis/>
- [8] VNPT AI (2023). *Sentiment Analysis - Phân tích cảm xúc*. <https://vnptai.io/vi/blog/detail/sentiment-analysis>
- [9] Evotek (2023). *Giải mã dữ liệu: Kỹ thuật xử lý ngôn ngữ tiếng Việt (Vietnamese NLP)*. <https://tuyendung.evotek.vn/giai-ma-du-lieu-ky-thuat-xu-ly-ngon-ngu-tieng-viet-vietnamese-nlp/>

Phụ lục A

Phụ lục

A.1 Mã nguồn và dữ liệu

Toàn bộ mã nguồn, dữ liệu và hướng dẫn triển khai có thể được tìm thấy tại kho GitHub của dự án: <https://github.com/Quoc-Dai2005/sentiment-analysis-project>

A.2 Cấu trúc thư mục dự án

```
sentiment-analysis-project/  
data/  
    comments.csv          % Chứa dữ liệu  
.gitignore               % Cấu hình git  
README.md                % Hướng dẫn dự án  
app.py                   % File chạy ứng dụng (Gradio/Streamlit)  
requirements.txt          % Các thư viện cần cài đặt  
trainer-ai.ipynb         % Notebook huấn luyện mô hình
```

Thư viện	Phiên bản	Công dụng
Transformers	4.21.0	Mô hình PhoBERT và Trainer
PyTorch	1.12.0	Nền tảng học sâu
Underthesea	6.0.0	Tách từ tiếng Việt
Pandas	1.5.0	Xử lý dữ liệu
Gradio	3.10.0	Giao diện web demo

Bảng A.1: Thư viện chính sử dụng trong dự án