PHÂN TÍCH CẢM XÚC ĐÁNH GIÁ TIẾNG VIỆT SỬ DỤNG MÔ HÌNH PHOBERT

Mã học phần: MAT3508 – Nhập môn Trí tuệ Nhân tạo

Nhóm 17 thực hiện:
Đồng Quốc Đại - 23001513
Chu Thành Dũng - 23001506
Nguyễn Mạnh Dũng - 23001507

Hà Nội, Tháng 11 năm 2025

Nội dung trình bày

1. 1. Giới thiệu Bài toán
2. 2. Phương pháp Triển khai
3. 3. Kết quả Phân tích
4. 4. Demo Ứng dụng thực tế
5. 5. Kết luận Hướng phát triển

1.1. Bối cảnh thực tế

Sự bùng nổ của Thương mại điện tử (TMĐT):

- Các nền tảng như Tiki, Shopee, Lazada phát triển mạnh mẽ tại Việt Nam.
- Mỗi ngày có hàng triệu lượt tương tác, mua hàng và để lại đánh giá (review).
- Bình luận của khách hàng là ”mỏ vàng”dữ liệu để doanh nghiệp thấu hiểu thị trường.

Vấn đề của phương pháp thủ công:

- Số lượng quá lớn: Con người không thể đọc hết hàng chục nghìn comment mỗi ngày.
- Thiếu nhất quán: Đánh giá của con người mang tính chủ quan, không đồng bộ.
- Tốn kém chi phí nhân sự và thời gian vận hành.

1.2. Giải pháp: Sentiment Analysis

Định nghĩa (Sentiment Analysis - Phân tích cảm xúc):

- Là lĩnh vực thuộc Xử lý ngôn ngữ tự nhiên (NLP).
- Sử dụng các thuật toán Học máy (Machine Learning) hoặc Học sâu (Deep Learning) để tự động nhận diện thái độ.

Mục tiêu cốt lõi:

- Chuyển đổi dữ liệu văn bản phi cấu trúc (text) thành dữ liệu định lượng (nhãn cảm xúc).
- Giúp máy tính ”hiểu”được người viết đang Khen, Chê hay Bình thường.

1.3. Phân loại bài toán

Trong dự án này, nhóm tập trung vào bài toán **Phân loại cực tính (Polarity Classification)** ở cấp độ câu (Sentence-level).

Hệ thống nhãn (Labels) được định nghĩa gồm 3 lớp:

1. **Positive (Tích cực):**
   - Tương ứng đánh giá 4-5 sao.
   - Thể hiện sự hài lòng, khen ngợi, yêu thích sản phẩm.
2. **Neutral (Trung lập):**
   - Tương ứng đánh giá 3 sao.
   - Cảm xúc không rõ ràng, đánh giá trung bình, hoặc chỉ hỏi thông tin.
3. **Negative (Tiêu cực):**
   - Tương ứng đánh giá 1-2 sao.
   - Thể hiện sự thất vọng, phàn nàn, giận dữ.

1.4. Thách thức của Tiếng Việt (1/3) - Tách từ

Đặc thù ngôn ngữ:

- Tiếng Anh dùng dấu cách (space) để phân tách các từ (word).
- Tiếng Việt dùng dấu cách để phân tách âm tiết. Đơn vị có nghĩa là ”Từ ghép”.

Ví dụ minh họa:

- Câu: ”giao hàng nhanh”.
- Nếu không tách từ: Máy hiểu là 3 từ rời rạc ”giao”, ”hàng”, ”nhanh”(Mất nghĩa ngữ cảnh).
- Xử lý đúng: ”giao_hàng”(1 từ) và ”nhanh”(1 từ).

$\rightarrow$ *Yêu cầu bắt buộc phải có bước Word Segmentation trước khi đưa vào mô hình.*

1.4. Thách thức của Tiếng Việt (2/3) - Sự đa nghĩa

Ngữ cảnh quyết định ý nghĩa:

- Một từ có thể mang nghĩa tích cực ở ngữ cảnh này nhưng tiêu cực ở ngữ cảnh khác.
- Ví dụ: ”Sản phẩm màu xanh”.
  - Ngữ cảnh 1: Khách đặt màu xanh → Đúng mô tả → **Positive/Neutral**.
  - Ngữ cảnh 2: Khách đặt màu đỏ → Giao sai màu → **Negative**.

Hàm ý và mỉa mai (Sarcasm):

- ”Giao hàng nhanh ghê, đặt từ tháng trước giờ mới có.”
- Máy rất dễ bị lừa bởi từ ”nhanh”(tích cực) mà bỏ qua ngữ cảnh thời gian (tiêu cực).

1.4. Thách thức của Tiếng Việt (3/3) - Dữ liệu ”bẩn”

Dữ liệu mạng xã hội và sàn TMĐT chứa rất nhiều nhiễu:

- **Teencode:** ”hng tot”(hàng tốt), ”k”(không), ”ngta”(người ta).
- **Ký tự lạ:** Các icon, emoji rác, ký tự đặc biệt vô nghĩa.
- **Lỗi chính tả:** Viết sai dấu, thiếu ký tự.

→ *Cần quy trình Tiền xử lý (Preprocessing) mạnh mẽ để làm sạch dữ liệu.*

1.5. Mục tiêu nghiên cứu

Mục tiêu tổng quát: Xây dựng thành công mô hình Deep Learning tự động phân loại cảm xúc bình luận Tiếng Việt trên Tiki.

Các nhiệm vụ cụ thể:

- **Nghiên cứu lý thuyết:** Tìm hiểu kiến trúc Transformer và PhoBERT.
- **Xử lý dữ liệu:** Thu thập và làm sạch >140.000 bình luận. Áp dụng kỹ thuật tách từ.
- **Huấn luyện mô hình:** Fine-tuning PhoBERT Large trên GPU.
- **Đánh giá:** Sử dụng các chỉ số Accuracy, F1-Score, ROC-AUC.
- **Ứng dụng:** Xây dựng Web Demo cho phép người dùng tương tác.

1.6. Đối tượng Phạm vi

Đối tượng nghiên cứu:

- Dữ liệu văn bản (text reviews) tiếng Việt.
- Các kỹ thuật NLP hiện đại (BERT, RoBERTa, PhoBERT).

Phạm vi dữ liệu:

- Nguồn: Tiki (Sàn TMĐT uy tín, nhiều bình luận chất lượng).
- Đặc điểm: Dữ liệu tiếng Việt có dấu và không dấu, bao gồm cả tiếng lóng.

Phạm vi công nghệ:

- Sử dụng thư viện `Underthesea` cho tách từ.
- Sử dụng `PyTorch` và `HuggingFace Transformers`.
- Môi trường thực nghiệm: Google Colab (GPU T4).

2.1. Quy trình tổng quan (Pipeline)

Nhóm thực hiện dự án theo quy trình chuẩn của một bài toán Machine Learning:

1. **Data Collection:** Thu thập dữ liệu thô.
2. **Preprocessing:** Làm sạch, chuẩn hóa, tách từ.
3. **Tokenization:** Mã hóa văn bản thành số.
4. **Model Training:** Huấn luyện (Fine-tuning) PhoBERT.
5. **Evaluation:** Đánh giá trên tập kiểm thử.
6. **Deployment:** Triển khai demo.

2.2. Mô tả dữ liệu đầu vào

Nguồn dữ liệu: Dataset `comments.csv` chứa các bình luận về sản phẩm trên Tiki.
Các trường thông tin chính:

- `rating`: Điểm đánh giá (1 đến 5 sao).
- `content`: Nội dung bình luận văn bản.

Số lượng:

- Tổng số mẫu thu thập: 141,281.
- Số mẫu hợp lệ sau khi lọc: 103,000 mẫu.

2.3. Phân phối nhãn Vấn đề mất cân bằng

Sau khi ánh xạ rating sang nhãn cảm xúc, ta có phân phối như sau:

Nhãn	Số lượng	Tỷ lệ	Nhận xét
Positive (4-5 sao)	18,170	64.3%	Chiếm đa số (Dominant)
Negative (1-2 sao)	5,966	21.1%	Khá thấp
Neutral (3 sao)	4,120	14.6%	**Thiểu số (Minority)**

Vấn đề: Dữ liệu bị mất cân bằng (Imbalanced Data).

- Mô hình sẽ có xu hướng học tốt lớp Positive.
- Lớp Neutral có ít dữ liệu nhất, dễ bị dự đoán sai.

2.4. Tiền xử lý (Preprocessing) - Bước 1: Làm sạch

Trước khi đưa vào mô hình, dữ liệu thô cần được làm sạch để loại bỏ nhiễu:

- **Chuẩn hóa Unicode:** Chuyển toàn bộ văn bản về bảng mã Unicode dựng sẵn (NFC) để tránh lỗi font.
- **Xóa ký tự đặc biệt:** Loại bỏ các dấu #, @, URL, email không mang ý nghĩa cảm xúc.
- **Chuẩn hóa định dạng:**
  - Chuyển về chữ thường (lowercase) để giảm số lượng từ vựng cần học.
  - Xóa khoảng trắng thừa đầu/cuối câu.
  - Loại bỏ các dòng trống hoặc dòng có giá trị `NaN`.

2.4. Tiền xử lý - Bước 2: Tách từ (Word Segmentation)

Đây là bước quan trọng nhất đối với tiếng Việt.
Công cụ sử dụng: Thư viện `Underthesea`.
Cơ chế:

- Nhận diện các từ ghép và nối chúng bằng dấu gạch dưới (_).
- Giúp mô hình phân biệt được ”đất nước”(country) với ”đất”(soil) và ”nước”(water).

Ví dụ thực tế:

- Input: ”Giao hàng nhanh, đóng gói cẩn thận.”
- Output: ”Giao_hàng nhanh , đóng_gói cẩn_thận .”

2.5. Tokenization - Mã hóa văn bản

Sau khi tách từ, văn bản được chuyển thành dạng số (IDs) để máy tính xử lý.

Kỹ thuật: Byte-Pair Encoding (BPE)

- PhoBERT sử dụng BPE để mã hóa.
- Các từ phổ biến giữ nguyên, các từ hiếm sẽ bị tách thành các phần nhỏ hơn (sub-words).

Cấu hình:

- `max_length = 128`: Các câu dài hơn sẽ bị cắt (truncation), ngắn hơn sẽ được thêm số 0 (padding).
- Lý do chọn 128: Phủ được hầu hết độ dài bình luận trung bình trên Tiki, tiết kiệm bộ nhớ hơn so với 256 hay 512.

2.6. Giới thiệu Mô hình PhoBERT

PhoBERT (Pre-trained for Vietnamese):

- Được phát triển bởi VinAI Research.
- Là mô hình đầu tiên công bố đạt kết quả SOTA (State-of-the-art) cho tiếng Việt.
- Kiến trúc dựa trên RoBERTa (một biến thể tối ưu hơn của BERT).

Tại sao chọn PhoBERT?

- BERT gốc (Multilingual) không tối ưu cho tiếng Việt vì không hiểu từ ghép.
- PhoBERT được huấn luyện trên 20GB văn bản tiếng Việt chất lượng cao.

Kiến trúc áp dụng:

- Input $\rightarrow$ PhoBERT Large $\rightarrow$ Vector đặc trưng (CLS token) $\rightarrow$ Lớp tuyến tính (Linear Layer) $\rightarrow$ Softmax $\rightarrow$ 3 output (Pos, Neu, Neg).

2.7. Chiến lược huấn luyện (1/2) - Gradient Accumulation

Vấn đề gặp phải:

- Mô hình `phobert-large` rất nặng, ngốn nhiều VRAM.
- Trên Google Colab (GPU T4 16GB), không thể chạy Batch Size lớn (ví dụ 16 hoặc 32) vì sẽ bị lỗi *Out Of Memory (OOM)*.

Giải pháp: Tích lũy Gradient

- Đặt `per_device_train_batch_size = 4`.
- Đặt `gradient_accumulation_steps = 4`.
- **Cơ chế:** Mô hình tính toán loss cho 4 batch nhỏ liên tiếp nhưng chưa cập nhật trọng số ngay. Nó cộng dồn gradient lại, sau 4 bước mới cập nhật 1 lần.
- **Hiệu quả:** Tương đương với việc huấn luyện Batch Size = $4 \times 4 = 16$, nhưng tốn ít RAM hơn.

2.7. Chiến lược huấn luyện (2/2) - Mixed Precision

Kỹ thuật FP16 (Floating Point 16-bit):

- Mặc định, các mạng nơ-ron sử dụng định dạng FP32 (32-bit) để lưu trữ trọng số.
- Nhóm đã kích hoạt chế độ `fp16=True` trong Trainer.

Lợi ích:

1. **Giảm bộ nhớ:** Trọng số chỉ chiếm một nửa dung lượng so với FP32.
2. **Tăng tốc độ:** GPU dòng Turing (như T4) tính toán FP16 nhanh hơn nhiều.
3. **Độ chính xác:** Vẫn được đảm bảo nhờ kỹ thuật scaling loss tự động.

2.8. Tổng hợp Siêu tham số (Hyperparameters)

Bảng cấu hình cuối cùng được sử dụng để huấn luyện:

Tham số	**Giá trị**	**Ý nghĩa**
Base Model	`vinai/phobert-large`	Mô hình nền tảng
Epochs	4	Số vòng lặp qua toàn bộ dữ liệu
Learning Rate	$1e-5$	Tốc độ học nhỏ để tinh chỉnh nhẹ nhàng
Batch Size thực tế	16	Đảm bảo hội tụ ổn định
Weight Decay	0.01	Tránh overfitting
Optimizer	AdamW	Tối ưu hóa trọng số
Evaluation Strategy	Epoch	Đánh giá sau mỗi vòng lặp

3.1. Tập dữ liệu kiểm thử (Test Set)

Kết quả dưới đây được đo lường trên tập Test hoàn toàn độc lập (mô hình chưa từng nhìn thấy trong quá trình học).

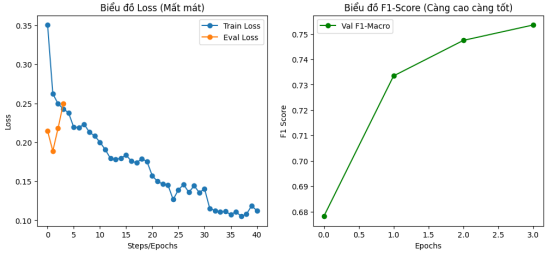
Thông tin tập Test:

- Số lượng mẫu: **20,653** bình luận.
- Tỷ lệ: 20

Các chỉ số đánh giá:

- **Accuracy:** Tỷ lệ dự đoán đúng tổng thể.
- **F1-Score (Macro):** Trung bình điều hòa giữa Precision và Recall (quan trọng vì dữ liệu mất cân bằng).
- **Confusion Matrix:** Phân tích chi tiết sự nhầm lẫn giữa các lớp.

3.2. Biểu đồ huấn luyện (Training History)



Hình: Biểu đồ Loss F1-Score

Phân tích:

- **Train Loss:** Giảm đều từ 0.35 xuống 0.12.
- **Eval Loss:** Giảm đồng biến, không bị Overfitting.
- **F1-Score:** Tăng nhanh và ổn định ở mức 0.75.

3.3. Kết quả tổng thể

Hiệu năng mô hình

Accuracy: 94%

F1-Macro: 0.75

Nhận định sơ bộ:

- Độ chính xác 94
- Tuy nhiên, F1-Macro thấp hơn Accuracy (0.75 vs 0.94) cảnh báo rằng có sự chênh lệch hiệu suất giữa các lớp (do mất cân bằng dữ liệu).

3.4. Phân tích chi tiết từng lớp (Classification Report)

Lớp	Precision	Recall	F1-Score	Số mẫu
Negative	0.80	0.74	0.77	1,437
Neutral	**0.49**	**0.54**	**0.51**	**1,046**
Positive	0.98	0.98	**0.98**	18,170

3.5. Đánh giá lớp Positive (Tích cực)

Kết quả: F1-Score đạt **0.98** (Gần như tuyệt đối).
Lý do thành công:

- Dữ liệu huấn luyện dồi dào (chiếm 64
- Các từ ngữ biểu thị tích cực thường rất rõ ràng và mạnh mẽ (VD: ”tuyệt vời”, ”đẹp”, ”nhanh”, ”ưng ý”).
- Mô hình học được các mẫu câu khen ngợi rất tốt.

3.6. Đánh giá lớp Negative (Tiêu cực)

Kết quả: F1-Score đạt **0.77** (Mức khá tốt).
Phân tích:

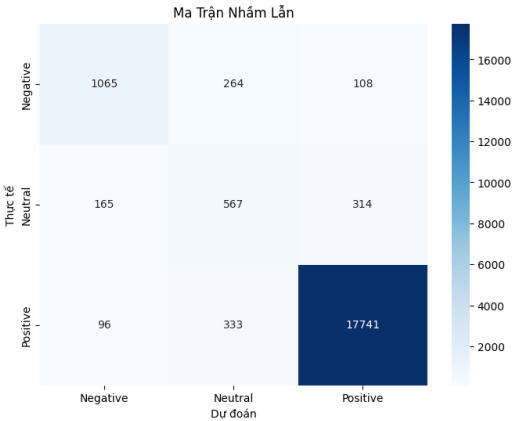
- Precision (0.80) cao hơn Recall (0.74).
- Nghĩa là: Khi mô hình dự đoán là Tiêu cực, nó thường đúng. Nhưng nó vẫn bỏ sót một số trường hợp tiêu cực (gán nhầm sang Trung lập).
- Nguyên nhân: Một số câu chê bai dùng từ ngữ giảm nhẹ (nói giảm nói tránh) hoặc châm biếm mà mô hình chưa bắt được hết.

3.7. Đánh giá lớp Neutral (Trung lập) - Điểm yếu

Kết quả: F1-Score chỉ đạt **0.51** (Thấp nhất).
Nguyên nhân thất bại:

1. **Thiếu dữ liệu:** Lớp Neutral chỉ chiếm 14
2. **Ranh giới mờ nhạt:**
   - Các đánh giá 3 sao trên Tiki rất đa dạng. Có người 3 sao là khen (”Tạm ổn”), có người 3 sao là chê (”Bình thường, không đặc sắc”).
   - Sự nhập nhằng (ambiguity) này khiến mô hình bị bối rối.

3.8. Ma trận nhầm lẫn (Confusion Matrix)

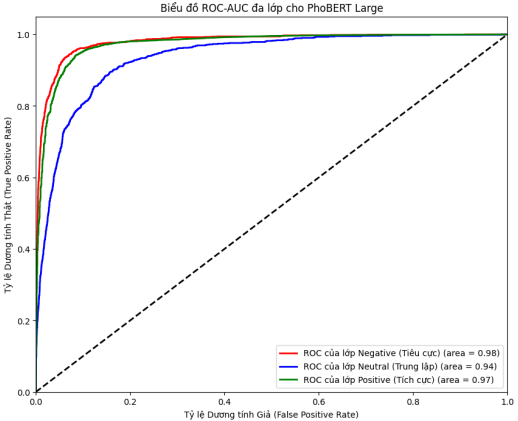


Hình: Ma trận nhầm lẫn thực tế

Nhận xét:

- **Neutral** $\rightarrow$ **Positive:** Nhầm 314 mẫu (Lớn nhất).
- **Negative** $\rightarrow$ **Neutral:** Nhầm 264 mẫu.
- Rất ít khi nhầm lẫn giữa hai cực đối lập (Negative $\leftrightarrow$ Positive).

3.9. Đường cong ROC-AUC



Hình: Đường cong ROC đa lớp

Kết quả AUC:

- Negative: **0.98**
- Positive: **0.97**
- Neutral: **0.94**

Ý nghĩa: AUC > 0.9 chứng tỏ mô hình có khả năng phân tách các lớp rất tốt, bất chấp việc chọn ngưỡng (threshold) nào.

4.1. Kiến trúc ứng dụng Web

Nhóm đã xây dựng một giao diện demo đơn giản để kiểm chứng mô hình trong thực tế.
Công nghệ: Thư viện `Gradio` (Python).
Luồng xử lý (Inference Flow):

1. Người dùng nhập text vào ô trống.
2. Hệ thống gọi hàm `gradio_predict`.
3. Văn bản đi qua các bước: Tách từ $\rightarrow$ Tokenize $\rightarrow$ Model $\rightarrow$ Softmax.
4. Trả về biểu đồ thanh ngang thể hiện độ tin cậy của 3 lớp.

4.2. Các trường hợp thử nghiệm (Test Cases)

Case 1 (Rõ ràng): ”Giao hàng nhanh, đóng gói cẩn thận, sách đẹp.”→ Dự đoán: **Positive** (Độ tin cậy > 99
Case 2 (Từ nối đối lập): ”Sách bìa thì đẹp *nhưng* nội dung sáo rỗng.”→ Dự đoán: **Negative**. *(Mô hình chú ý vào vế sau từ ”nhưng cơ chế Attention hoạt động tốt).*
Case 3 (Ẩn dụ): ”Shop làm ăn bát nháo, treo đầu dê bán thịt chó.”→ Dự đoán: **Negative**. *(Mô hình hiểu được thành ngữ tiêu cực).*

5.1. Tổng kết kết quả đạt được

1. **Quy trình hoàn thiện:** Đã xây dựng thành công pipeline xử lý NLP từ A-Z cho tiếng Việt.
2. **Hiệu năng cao:** Đạt độ chính xác 94
3. **Làm chủ công nghệ:** Đã nắm vững cách sử dụng các mô hình ngôn ngữ lớn (PhoBERT) và các kỹ thuật tối ưu huấn luyện (Gradient Accumulation).

5.2. Nhìn nhận hạn chế

Bên cạnh thành công, dự án còn tồn tại các điểm yếu:

- **Hiệu suất lớp Neutral thấp:** Chưa giải quyết triệt để vấn đề mất cân bằng dữ liệu.
- **Tài nguyên tính toán:**
  - PhoBERT Large có kích thước lớn (>1GB).
  - Tốc độ dự đoán (Inference speed) còn chậm nếu chạy trên CPU, khó triển khai trên thiết bị di động.
- **Phạm vi:** Chưa xử lý được các bình luận chứa cả khen và chê cùng lúc cho các khía cạnh khác nhau (Aspect-based).

5.3. Hướng phát triển (Future Work)

Để nâng cấp hệ thống, nhóm đề xuất các hướng đi sau:

1. Cải thiện dữ liệu (Data Augmentation):

- Sử dụng kỹ thuật dịch ngược (Back-translation) hoặc thay thế từ đồng nghĩa để sinh thêm dữ liệu cho lớp Neutral và Negative.

2. Cải thiện thuật toán:

- Thay thế hàm Loss: Sử dụng **Focal Loss** để phạt nặng hơn khi mô hình dự đoán sai các lớp thiểu số.
- Thử nghiệm mô hình nhẹ hơn: **DistilPhoBERT** hoặc lượng tử hóa (Quantization) để tăng tốc độ.

Tài liệu tham khảo

- Nguyen, D. Q. et al. (2020). *PhoBERT: Pre-trained language models for Vietnamese*. EMNLP Findings.
- Wolf, T. et al. (2020). *Transformers: State-of-the-art natural language processing*.
- Underthesea Project. *Vietnamese NLP Toolkit*.
- Liu, B. (2012). *Sentiment analysis and opinion mining*.
- Abid, A. et al. (2019). *Gradio: Hassle-free sharing and testing of ML models*.

XIN CẢM ƠN
THẦY VÀ CÁC BẠN ĐÃ LẮNG NGHE!

Mã nguồn dự án:
`https://github.com/Quoc-Dai2005/sentiment-analysis-project`

(Nhóm rất mong nhận được câu hỏi và góp ý từ mọi người)