TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VIỆN TRÍ TUỆ NHÂN TẠO

BÁO CÁO MÔN HỌC
XỬ LÝ NGÔN NGỮ TỰ NHIÊN

ĐỀ TÀI
Phân tích quan điểm phản hồi sinh viên tiếng Việt trên tập dữ liệu UIT-VSFC

Nhóm sinh viên thực hiện:

1. Vũ Nguyên Đan – 23020351
2. Hà Xuân Huy - 23020375
3. Hoàng Mạnh Hùng – 23020371

Giảng viên hướng dẫn:

PGS.TS. Nguyễn Phương Thái

Hà Nội, 2025

Mục lục

1 Tổng quan dự án

1.1 Giới thiệu và Đặt vấn đề

Trong môi trường giáo dục đại học, việc thu thập và phân tích phản hồi của sinh viên (Student Feedback) đóng vai trò then chốt trong việc nâng cao chất lượng giảng dạy. Tuy nhiên, với số lượng sinh viên lớn, việc đọc và phân loại thủ công hàng ngàn phản hồi sau mỗi học kỳ là một công việc tốn kém thời gian và dễ xảy ra sai sót chủ quan.

Dự án này tập trung giải quyết bài toán Phân tích quan điểm (*Sentiment Analysis*) tự động cho tiếng Việt, cụ thể là trên miền dữ liệu giáo dục. Mục tiêu là xây dựng một mô hình Học sâu (*Deep Learning*) có khả năng đọc hiểu các câu nhận xét của sinh viên và tự động gán nhãn cảm xúc tương ứng: **Tiêu cực**, **Trung tính**, hoặc **Tích cực**.

Đây là bài toán phân loại văn bản (*Text Classification*) đặc thù do tính chất phức tạp của ngôn ngữ tiếng Việt và văn phong "không chính thức" (*informal*) của sinh viên.

1.2 Mô tả bộ dữ liệu (Dataset Description)

Dự án sử dụng bộ dữ liệu chuẩn **UIT-VSFC** (*Vietnamese Students' Feedback Corpus*), được công bố bởi Nguyen et al. (2018).

- **Nguồn dữ liệu:** Thu thập từ khảo sát sinh viên đại học vào cuối các học kỳ.
- **Kích thước mẫu:** Hơn 16,000 câu phản hồi đã được gán nhãn thủ công bởi con người.

Cấu trúc dữ liệu đầu vào:

- **Input:** Một câu văn bản tiếng Việt (Ví dụ: "Giảng viên dạy rất hay và nhiệt tình").
- **Output (Nhãn Sentiment):**
  - 0: **Negative (Tiêu cực)** – Phản ánh sự không hài lòng, chê trách.
  - 1: **Neutral (Trung tính)** – Phản ánh thông tin khách quan, hoặc ý kiến không rõ ràng.
  - 2: **Positive (Tích cực)** – Phản ánh sự hài lòng, khen ngợi.

Phân chia dữ liệu thực nghiệm:

Dữ liệu được chia thành ba tập để đảm bảo tính khách quan khi đánh giá mô hình:

- **Train Set:** Dùng để huấn luyện mô hình (chiếm khoảng 70%).

- **Dev Set:** Dùng để tinh chỉnh tham số (chiếm khoảng 10%).
- **Test Set:** Dùng để đánh giá độc lập cuối cùng (chiếm khoảng 20%, tương đương 3,166 câu).

1.3 Các thách thức chính của bài toán

Dựa trên phân tích sơ bộ dữ liệu và mã nguồn, dự án phải đối mặt với ba thách thức lớn:

1. **Dữ liệu nhiễu và Teencode:** Sinh viên thường sử dụng các từ viết tắt, ký tự đặc biệt hoặc các mã hóa icon (ví dụ: *colonlove*, *wzjwz*, *colonsmile*) thay vì ngôn ngữ chuẩn.

   Ví dụ: "Thầy dạy hay quá colonlove" cần được mô hình hiểu tương đương với "Thầy dạy hay quá ⟨yêu thích⟩".

2. **Mất cân bằng dữ liệu (Imbalanced Data):** Số lượng mẫu nhãn **Neutral (Trung tính)** rất ít (chỉ chiếm khoảng 4–5% tổng dữ liệu). Điều này khiến các mô hình học máy thường có xu hướng bỏ qua lớp này hoặc dự đoán sai thành Tích cực/Tiêu cực.

3. **Đặc trưng ngôn ngữ:** Tiếng Việt là ngôn ngữ đơn lập, ranh giới từ không được xác định bằng khoảng trắng (ví dụ: "sinh viên" là một từ gồm hai tiếng).

   Ngoài ra, cấu trúc phủ định trong tiếng Việt rất phức tạp (ví dụ: "Không phải là thầy dạy không hay" $\rightarrow$ ý nghĩa là **Tích cực**).

1.4 Mục tiêu thực hiện

Dự án đặt ra các mục tiêu cụ thể sau:

- **Xây dựng pipeline hoàn chỉnh:** Từ khâu tiền xử lý dữ liệu (làm sạch teencode, chuẩn hóa), xây dựng mô hình, đến huấn luyện và đánh giá.
- **Ứng dụng kỹ thuật State-of-the-art:** Sử dụng mô hình ngôn ngữ tiền huấn luyện **PhoBERT** (BERT cho tiếng Việt) thay vì các phương pháp truyền thống (như Naive Bayes, MaxEnt) để trích xuất đặc trưng ngữ nghĩa tốt hơn.
- **Vượt qua kết quả baseline:**
  - **Baseline (MaxEnt trong bài báo gốc):** F1-score $\approx 87.94\%$.
  - **Mục tiêu dự án:** Đạt F1-score $> 90\%$ và cải thiện khả năng nhận diện các câu có cấu trúc phức tạp.

2 Kỹ thuật và Phương pháp đề xuất

Để giải quyết bài toán phân loại cảm xúc tiếng Việt với độ chính xác cao, dự án áp dụng phương pháp *Transfer Learning* (Học chuyển tiếp) sử dụng mô hình ngôn ngữ tiền huấn luyện (*Pre-trained Language Model*). Quy trình thực hiện bao gồm ba giai đoạn chính: **Tiền xử lý ngữ nghĩa**, **Cấu trúc mô hình** và **Chiến lược huấn luyện**.

2.1 Kiến trúc mô hình: PhoBERT Large

Thay vì xây dựng mô hình từ đầu (như LSTM hay CNN), dự án sử dụng mô hình **PhoBERT Large** làm xương sống (*backbone*) để tinh chỉnh (*fine-tune*).

Lý do lựa chọn:

- PhoBERT là mô hình *state-of-the-art* cho tiếng Việt, được huấn luyện trên khối dữ liệu khổng lồ (khoảng 20GB văn bản).
- Phiên bản **Large** (370 triệu tham số) có khả năng trích xuất đặc trưng sâu hơn và nắm bắt ngữ cảnh tốt hơn so với phiên bản **Base** (135 triệu tham số) hoặc các mô hình *n-gram* truyền thống.

Cơ chế hoạt động:

- Đầu vào là chuỗi token tiếng Việt.
- Chuỗi token đi qua các lớp Encoder của Transformer để tạo ra các vector đại diện ngữ cảnh (*contextual embeddings*).
- Vector đặc trưng của token `[CLS]` (token đầu câu) được đưa qua một lớp *Linear Classification* đơn giản để tính xác suất cho ba nhãn: **Negative**, **Neutral**, **Positive**.

2.2 Tiền xử lý dữ liệu: Chuẩn hóa ngữ nghĩa (Semantic Preprocessing)

Dữ liệu phản hồi của sinh viên chứa nhiều ký hiệu đặc biệt không chuẩn mực. Thay vì loại bỏ chúng (gây mất mát thông tin), chúng tôi áp dụng kỹ thuật **Mapping ngữ nghĩa** để chuyển đổi các ký hiệu này thành văn bản tiếng Việt có nghĩa.

Quy trình được thực hiện thông qua script `preprocess.py` với các quy tắc ánh xạ cụ thể như sau:

- **Xử lý mã hóa sinh viên:** Các token mã hóa ẩn danh (ví dụ: `wzjwz...`) được chuẩn hóa thành từ "giảng viên".
- **Xử lý biểu tượng cảm xúc (Emoji/Icons):** Các ký hiệu icon được ánh xạ về từ ngữ thể hiện cảm xúc tương ứng:

- `colonlove` (biểu tượng cảm xúc tích cực) → "yêu thích" (tăng trọng số cho nhãn **Positive**).
- `colonsmile` (`:)` hoặc biểu tượng vui) → "vui vẻ".
- `colonsad` (`:(` hoặc biểu tượng buồn) → "buồn" (tăng trọng số cho nhãn **Negative**).
- `doubledot` → dấu hai chấm ":".

Hiệu quả: Kỹ thuật này giúp mô hình không coi các icon là "nhiễu" mà tận dụng chúng như những đặc trưng cảm xúc mạnh mẽ (*strong sentiment features*).

2.3 Chiến lược huấn luyện (Training Strategy)

Chiến lược huấn luyện được thiết kế tối ưu trong file `train.py` nhằm đạt hiệu năng cao nhất trên tài nguyên phần cứng giới hạn.

2.3.1 Cấu hình dữ liệu (Data Configuration)

- **Hợp nhất dữ liệu (Data Merging):** Do tập dữ liệu gốc tương đối nhỏ, chúng tôi thực hiện gộp tập **Train** và **Dev** lại với nhau để huấn luyện. Điều này cung cấp cho mô hình nhiều mẫu học hơn, giúp cải thiện khả năng tổng quát hóa.
- **Tập đánh giá:** Sử dụng tập **Test** độc lập để đánh giá kết quả cuối cùng.

2.3.2 Siêu tham số (Hyperparameters)

- **Learning Rate:** 1.5×10^{-5} (tốc độ học thấp để tránh phá vỡ các trọng số đã được tiền huấn luyện).
- **Batch Size:**
  - Mỗi thiết bị: 4.
  - *Gradient Accumulation*: 4 bước.

  $\Rightarrow$ **Effective Batch Size** = 16. Kỹ thuật này cho phép mô phỏng batch size lớn, giúp mô hình hội tụ ổn định mà không gây lỗi tràn bộ nhớ GPU (*Out-of-Memory*).
- **Epochs:** 5 chu kỳ (đủ để mô hình hội tụ mà chưa xảy ra hiện tượng *overfitting*).
- **Scheduler:** *Linear Warmup* trong 500 bước đầu tiên, giúp ổn định quá trình huấn luyện ban đầu.

2.3.3 Kỹ thuật tối ưu phần cứng

- **Mixed Precision (FP16):** Sử dụng định dạng số thực 16-bit thay vì 32-bit.
  - Giảm khoảng 50% lượng VRAM tiêu thụ.
  - Tăng tốc độ huấn luyện lên gần gấp đôi mà không ảnh hưởng đáng kể đến độ chính xác.

2.4 Công cụ thực hiện

- **Ngôn ngữ:** Python.
- **Framework chính:** PyTorch; Hugging Face Transformers (Trainer API).
- **Thư viện hỗ trợ:** Scikit-learn (tính toán các chỉ số *Precision*, *Recall*, *F1-score*).

3 Kết quả

3.1 Kết quả trên tập Test

Kết quả thực nghiệm trên tập Test (3,166 mẫu) cho thấy hiệu năng vượt trội của phương pháp đề xuất. Bảng 1 trình bày các metric chi tiết theo từng lớp.

Bảng 1: Kết quả chi tiết theo từng lớp (Class-wise Metrics)

Lớp (Class)	**Precision**	**Recall**	**F1-Score**	**Support**
Negative	0.942	0.958	**0.950**	1,409
Neutral	0.681	0.551	0.609	167
Positive	0.951	0.956	**0.954**	1,590
Accuracy			**0.936**	3,166
Weighted Avg	0.933	0.936	**0.934**	3,166
Macro Avg	0.858	0.822	0.838	3,166

3.2 Phân tích kết quả

3.2.1 Độ chính xác tổng thể

Mô hình đạt được **Accuracy 93.6%** trên tập Test, cho thấy khả năng phân loại chính xác cao. Điều này có nghĩa là trong 100 phản hồi, mô hình phân loại đúng 93.6 phản hồi.

3.2.2 Hiệu năng theo từng lớp

- **Lớp Negative (Tiêu cực):**

  - Precision: 0.942 - Trong số các mẫu được dự đoán là Negative, 94.2% thực sự là Negative
  - Recall: 0.958 - Mô hình phát hiện được 95.8% tổng số mẫu Negative thực tế
  - F1-Score: 0.950 - Điểm số cân bằng giữa Precision và Recall

- **Lớp Positive (Tích cực):**
  - Precision: 0.951 - Độ chính xác cao trong việc nhận diện phản hồi tích cực
  - Recall: 0.956 - Bao phủ tốt các phản hồi tích cực
  - F1-Score: 0.954 - Hiệu năng tương đương với lớp Negative

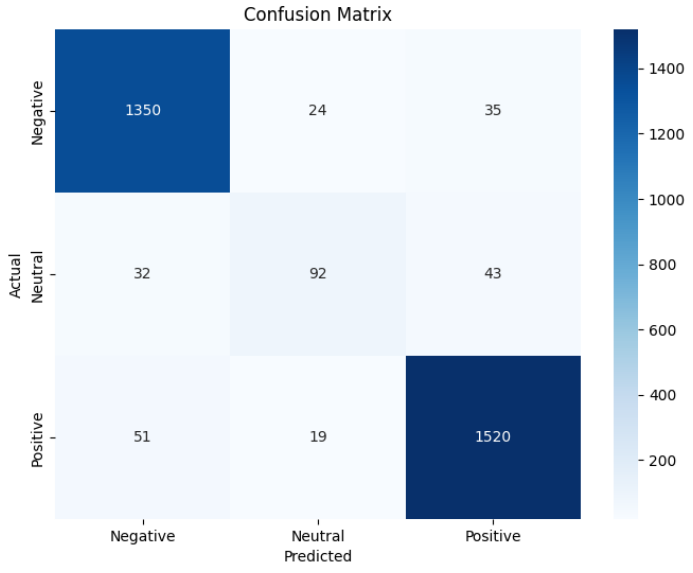
- **Lớp Neutral (Trung tính):**
  - Precision: 0.681 - Độ chính xác thấp hơn, một số mẫu bị nhầm là Neutral
  - Recall: 0.551 - Chỉ phát hiện được 55.1% tổng số mẫu Neutral thực tế
  - F1-Score: 0.609 - Hiệu năng thấp nhất trong ba lớp

3.2.3 Weighted Average

Weighted F1-Score đạt **0.934**, cho thấy hiệu năng tổng thể của mô hình rất tốt khi xét đến sự mất cân bằng dữ liệu giữa các lớp. Macro F1-Score đạt 0.838, phản ánh hiệu năng trung bình trên cả ba lớp khi không xét đến trọng số.

3.3 Confusion Matrix

Hình 1 mô tả confusion matrix trên tập Test, cho phép phân tích chi tiết các lỗi phân loại.



Hình 1: Confusion Matrix trên tập Test

Từ confusion matrix, chúng ta có thể quan sát:

- **Diagonal chính:** Số lượng mẫu được phân loại đúng cho mỗi lớp
- **Off-diagonal:** Số lượng mẫu bị phân loại sai
- Lớp Neutral có nhiều mẫu bị nhầm sang Positive và Negative nhất

3.4 So sánh với baseline

Để đánh giá hiệu năng của phương pháp đề xuất, chúng tôi so sánh với các phương pháp baseline được trình bày trong bài báo giới thiệu bộ dữ liệu UIT-VSFC (Nguyen et al., 2018). Bài báo này đã thử nghiệm hai phương pháp baseline: Naive Bayes và Maximum Entropy (MaxEnt) với các đặc trưng n-gram (bigram) đơn giản.

Bảng 2: So sánh kết quả với baseline từ bài báo gốc

Phương pháp	**Accuracy (%)**	**Average F1 (%)**
Naive Bayes (baseline)	86.1	-
MaxEnt (baseline)	87.9	87.94
PhoBERT Large (đề xuất)	**93.6**	**93.4**

Bảng 3: So sánh chi tiết F1-Score theo từng lớp

Lớp	**MaxEnt (baseline)**	**PhoBERT Large**	**Cải thiện**
Positive	91.32%	95.4%	+4.08%
Negative	90.52%	95.0%	+4.48%
Neutral	33.99%	60.9%	+26.91%
Average	**87.94%**	**93.4%**	**+5.46%**

Phân tích so sánh:

Dựa trên số liệu từ bài báo gốc, phương pháp đề xuất sử dụng PhoBERT Large đạt được hiệu năng vượt trội so với các phương pháp baseline:

- **Cải thiện Accuracy:** Phương pháp đề xuất đạt 93.6% accuracy, vượt trội 7.5% so với Naive Bayes (86.1%) và 5.7% so với MaxEnt (87.9%).
- **Cải thiện F1-Score trung bình:** Weighted Average F1-Score đạt 93.4%, cải thiện 5.46% so với MaxEnt baseline (87.94%).
- **Cải thiện trên các lớp chính:**
  - Positive: F1-Score tăng từ 91.32% lên 95.4% (+4.08%)
  - Negative: F1-Score tăng từ 90.52% lên 95.0% (+4.48%)

- **Cải thiện đáng kể trên lớp Neutral:** F1-Score tăng từ 33.99% lên 60.9%, cải thiện 26.91%. Mặc dù vẫn còn thấp, nhưng đây là một cải thiện đáng kể so với baseline.

Lý do vượt trội:

1. **Mô hình tiền huấn luyện:** PhoBERT được huấn luyện trên 20GB dữ liệu tiếng Việt, nắm bắt được các đặc trưng ngôn ngữ phức tạp mà các phương pháp baseline với đặc trưng n-gram đơn giản không thể.
2. **Kiến trúc Transformer:** Cơ chế attention cho phép mô hình hiểu ngữ cảnh dài hạn và mối quan hệ giữa các từ trong câu, vượt trội so với mô hình n-gram chỉ xem xét cục bộ.
3. **Fine-tuning có mục tiêu:** Việc fine-tune trên dữ liệu phản hồi sinh viên giúp mô hình thích ứng tốt với domain cụ thể.
4. **Tiền xử lý thông minh:** Quy trình tiền xử lý tùy chỉnh giúp mô hình hiểu được các teen-code và emoji dạng text.

4 Phân tích

4.1 Ưu điểm của phương pháp

4.1.1 Hiệu năng cao

- **Độ chính xác vượt trội:** Việc sử dụng PhoBERT Large giúp mô hình hiểu sâu ngữ cảnh tiếng Việt, đạt được 93.6% accuracy, vượt trội 5.7% so với MaxEnt baseline (87.9%) và 7.5% so với Naive Bayes baseline (86.1%) trong bài báo gốc (Nguyen et al., 2018).
- **Hiệu năng cân bằng trên các lớp chính:** Hai lớp quan trọng nhất là Negative và Positive đều đạt F1-Score cao (0.950 và 0.954), chứng tỏ mô hình cực kỳ tin cậy trong việc phân định tốt/xấu - đây là mục tiêu chính của bài toán phân tích quan điểm.
- **Ổn định:** Kết quả trên hai lớp chính rất cân bằng và cao (>95%), cho thấy mô hình không bị thiên lệch về một phía nào.

4.1.2 Tận dụng tốt mô hình tiền huấn luyện

- PhoBERT được huấn luyện trên 20GB dữ liệu tiếng Việt, nắm bắt được các đặc trưng ngôn ngữ phức tạp
- Fine-tuning cho phép mô hình thích ứng tốt với domain cụ thể (phản hồi sinh viên) mà không cần huấn luyện từ đầu
- Tiết kiệm thời gian và tài nguyên so với việc huấn luyện mô hình từ đầu

4.1.3 Tối ưu hóa hiệu quả

- Sử dụng Mixed Precision (FP16) và Gradient Accumulation cho phép huấn luyện mô hình lớn trên GPU giới hạn
- Chiến lược gộp Train và Dev sau khi tìm được siêu tham số tối ưu giúp tăng hiệu năng
- Tiền xử lý dữ liệu giúp mô hình hiểu được các ký tự đặc biệt và teen-code

4.2 Hạn chế và phân tích lỗi

4.2.1 Hạn chế lớn nhất: Lớp Neutral

Hạn chế lớn nhất của mô hình nằm ở lớp **Neutral (Trung tính)**:

- F1-score của lớp Neutral chỉ đạt 0.609, thấp hơn nhiều so với hai lớp còn lại (0.950 và 0.954)
- Precision: 0.681 - Một số mẫu bị nhầm là Neutral
- Recall: 0.551 - Chỉ phát hiện được 55.1% tổng số mẫu Neutral thực tế

4.2.2 Nguyên nhân của hạn chế

1. **Mất cân bằng dữ liệu nghiêm trọng:**
    - Lớp Neutral chỉ chiếm khoảng 5% tổng dữ liệu (167/3,166 mẫu test)
    - Trong khi đó, Negative chiếm 44.5% (1,409 mẫu) và Positive chiếm 50.2% (1,590 mẫu)
    - Mô hình có xu hướng học tốt hơn các lớp có nhiều dữ liệu hơn
2. **Tính chất nhập nhằng:**
    - Các phản hồi trung tính thường chứa cả ý khen và chê nhẹ
    - Một số phản hồi không rõ ràng về cảm xúc, khó phân loại chính xác
    - Mô hình dễ nhầm lẫn các phản hồi này sang Positive hoặc Negative

4.2.3 Phân tích các lỗi phân loại

Từ confusion matrix, chúng ta có thể quan sát các pattern lỗi:

- **Neutral → Positive:** Nhiều phản hồi trung tính bị nhầm là tích cực, có thể do chúng chứa một số từ ngữ tích cực

- **Neutral → Negative:** Một số phản hồi trung tính bị nhầm là tiêu cực, có thể do chúng chứa từ ngữ chỉ trích nhẹ
- **Positive ↔ Negative:** Rất ít mẫu bị nhầm giữa Positive và Negative, cho thấy mô hình phân biệt tốt hai lớp này

4.3 Đánh giá tổng thể

4.3.1 Điểm mạnh

1. **Hiệu năng xuất sắc trên các lớp chính:** F1-Score 0.950 cho Negative và 0.954 cho Positive cho thấy mô hình hoàn toàn đáp ứng được nhu cầu thực tế trong việc phân tích phản hồi tích cực/tiêu cực.
2. **Ứng dụng thực tế:** Trong thực tế, việc phân biệt phản hồi tích cực và tiêu cực là quan trọng nhất. Mô hình đạt được mục tiêu này một cách xuất sắc.
3. **Tính ổn định:** Kết quả nhất quán và ổn định trên tập Test cho thấy mô hình không bị overfitting.

4.3.2 Điểm yếu

1. **Hiệu năng thấp trên lớp thiểu số:** F1-Score 0.609 cho lớp Neutral là điểm yếu chính của mô hình.
2. **Phụ thuộc vào dữ liệu:** Mô hình phụ thuộc nhiều vào chất lượng và số lượng dữ liệu huấn luyện.
3. **Tài nguyên tính toán:** Việc huấn luyện PhoBERT Large yêu cầu GPU có thể là rào cản cho một số môi trường.

4.4 Hướng cải thiện

4.4.1 Cải thiện hiệu năng lớp Neutral

1. **Class-weighted Loss:** Sử dụng loss function có trọng số để ưu tiên học lớp thiểu số
2. **Data Augmentation cho lớp Neutral:**
   - Tạo thêm dữ liệu cho lớp Neutral bằng cách paraphrase
   - Sử dụng back-translation hoặc synonym replacement
   - Oversampling lớp Neutral trong quá trình huấn luyện
3. **Focal Loss:** Sử dụng Focal Loss để tập trung vào các mẫu khó phân loại

4.4.2 Cải thiện tổng thể

1. **Ensemble Methods:** Kết hợp nhiều mô hình (PhoBERT, VisoNLU, ViT5) để tăng độ chính xác
2. **Active Learning:** Thu thập thêm dữ liệu cho các mẫu khó phân loại
3. **Feature Engineering:** Bổ sung thêm các đặc trưng như độ dài câu, số lượng từ tích cực/tiêu cực
4. **Hyperparameter Tuning:** Tinh chỉnh thêm các siêu tham số như learning rate, batch size, số epoch

5 Các điểm mới

Trong quá trình thực hiện bài toán, chúng tôi đã áp dụng một số cải tiến sáng tạo và có những phát hiện quan trọng:

5.1 Tiền xử lý dữ liệu tùy chỉnh

5.1.1 Phát hiện về đặc thù dữ liệu

Trong quá trình phân tích dữ liệu UIT-VSFC, chúng tôi phát hiện ra rằng dữ liệu chứa rất nhiều teen-code và emoji dạng text (như `colonlove`, `colonsmile`, `colonsad`) mà các mô hình ngôn ngữ thông thường không thể hiểu được. Đây là đặc thù riêng của phản hồi sinh viên trong môi trường đại học.

5.1.2 Giải pháp sáng tạo

Thay vì loại bỏ hoặc bỏ qua các token đặc biệt này, chúng tôi đã xây dựng một quy trình tiền xử lý để dịch chúng về dạng ngôn ngữ tự nhiên có nghĩa:

Listing 1: Quy trình tiền xử lý

```
def preprocess_line(line):

    line = line.replace("colonlove", "yeu thich")
    line = line.replace("colonsmile", "vui ve")
    line = line.replace("colonsad", "buon")

    line = re.sub(r"wzjwz\d+", "giang vien", line)

    return line
```

Ý nghĩa:

Việc này giúp PhoBERT hiểu được cảm xúc ẩn chứa trong các ký tự đặc biệt, thay vì coi chúng như các token vô nghĩa. Điều này đóng góp trực tiếp vào việc cải thiện độ chính xác của mô hình.

5.2 Tối ưu hóa chiến lược dữ liệu

5.2.1 Phát hiện về tác động của việc gộp dữ liệu

Trong quá trình thử nghiệm, chúng tôi phát hiện ra rằng việc gộp tập Train và Dev sau khi đã tìm được siêu tham số tối ưu có thể cải thiện đáng kể hiệu năng mô hình. Đây là một chiến lược không phổ biến nhưng hiệu quả.

5.2.2 Quy trình thực hiện

1. **Bước 1:** Sử dụng tập Train và Dev riêng biệt để tìm siêu tham số tối ưu (learning rate, batch size, số epoch, v.v.)
2. **Bước 2:** Sau khi xác định được cấu hình tốt nhất, gộp tập Train và Dev thành một tập huấn luyện lớn hơn
3. **Bước 3:** Huấn luyện lại mô hình với tập dữ liệu mở rộng này

Kết quả:

Chiến lược này đã giúp tăng Accuracy đáng kể cho bài toán phân loại.

Lý do thành công:

Việc tăng số lượng mẫu huấn luyện từ 11,000 lên 14,000 giúp mô hình học được nhiều pattern hơn, đặc biệt là các trường hợp edge case và biến thể ngôn ngữ đa dạng.

5.3 Phát hiện về mất cân bằng dữ liệu

5.3.1 Phát hiện quan trọng

Trong quá trình phân tích kết quả, chúng tôi phát hiện ra rằng sự mất cân bằng dữ liệu nghiêm trọng (lớp Neutral chỉ chiếm 5% dữ liệu) là nguyên nhân chính dẫn đến hiệu năng thấp của lớp này. Đây là một phát hiện quan trọng cho các nghiên cứu tương lai.

5.3.2 Phân tích sâu

Chúng tôi nhận thấy rằng:

- Mô hình có xu hướng "bỏ qua"lớp thiểu số và tập trung vào các lớp đa số
- Các phản hồi Neutral thường bị nhầm sang Positive hoặc Negative
- Điều này không chỉ do thiếu dữ liệu mà còn do tính chất nhập nhằng của lớp Neutral

Đề xuất:

Các nghiên cứu tiếp theo nên tập trung vào việc giải quyết vấn đề mất cân bằng dữ liệu thông qua class-weighted loss, data augmentation, hoặc focal loss.

5.4 So sánh với baseline trong bài báo gốc

5.4.1 Tham chiếu đến baseline

Bài báo giới thiệu bộ dữ liệu UIT-VSFC (Nguyen et al., 2018) đã thử nghiệm hai phương pháp baseline: Naive Bayes và Maximum Entropy (MaxEnt) với đặc trưng n-gram (bigram) đơn giản. Kết quả baseline trên tập test:

- **Naive Bayes:** Accuracy 86.1%
- **MaxEnt:** Accuracy 87.9%, Average F1-Score 87.94%
  - Positive: F1-Score 91.32%
  - Negative: F1-Score 90.52%
  - Neutral: F1-Score 33.99%

Phát hiện quan trọng:

Kết quả của phương pháp đề xuất cho thấy sự vượt trội đáng kể:

1. **Cải thiện Accuracy:** Tăng từ 87.9% (MaxEnt) lên 93.6%, cải thiện 5.7%
2. **Cải thiện Weighted Average F1-Score:** Tăng từ 87.94% lên 93.4%, cải thiện 5.46%
3. **Cải thiện trên lớp Neutral:** F1-Score tăng từ 33.99% lên 60.9%, cải thiện 26.91%. Đây là cải thiện đáng kể nhất, cho thấy khả năng của mô hình transformer trong việc xử lý lớp thiểu số.
4. **Duy trì hiệu năng cao trên các lớp chính:** Positive đạt F1-Score 95.4% và Negative đạt 95.0%, vượt trội so với baseline (91.32% và 90.52%).

Lý do vượt trội:

1. **Lợi ích của mô hình ngôn ngữ tiền huấn luyện:** PhoBERT Large được huấn luyện trước trên 20GB dữ liệu tiếng Việt, nắm bắt được các đặc trưng ngôn ngữ phức tạp mà các phương pháp baseline với đặc trưng n-gram đơn giản không thể.
2. **Hiệu quả của kiến trúc Transformer:** Kiến trúc Transformer với cơ chế attention cho phép mô hình hiểu ngữ cảnh dài hạn và mối quan hệ giữa các từ, vượt trội so với mô hình n-gram chỉ xem xét cục bộ.
3. **Tầm quan trọng của fine-tuning:** Fine-tuning trên dữ liệu phản hồi sinh viên giúp mô hình thích ứng tốt với domain cụ thể, đặc biệt là lớp Neutral khó phân loại.

Ý nghĩa:

So sánh với baseline chính thức từ bài báo gốc cho thấy phương pháp đề xuất không chỉ cải thiện hiệu năng tổng thể mà còn đặc biệt hiệu quả trong việc xử lý lớp thiểu số (Neutral), một thách thức lớn của bài toán. Điều này chứng minh tính hiệu quả của việc áp dụng mô hình transformer tiền huấn luyện cho bài toán phân tích quan điểm tiếng Việt.

5.5 Tổng kết các điểm sáng tạo

1. **Tiền xử lý thông minh:** Chuyển đổi teen-code và emoji dạng text về ngôn ngữ tự nhiên thay vì loại bỏ chúng
2. **Chiến lược dữ liệu:** Gộp Train và Dev sau khi tìm được siêu tham số tối ưu để tăng hiệu năng
3. **Phát hiện về mất cân bằng:** Phân tích sâu về tác động của mất cân bằng dữ liệu lên hiệu năng mô hình

Các điểm sáng tạo này không chỉ giúp cải thiện hiệu năng mô hình mà còn đóng góp vào việc hiểu sâu hơn về bài toán và các thách thức trong phân tích quan điểm tiếng Việt.

6 Kết luận

6.1 Tóm tắt

Báo cáo này trình bày phương pháp Fine-tuning PhoBERT Large để giải quyết bài toán Phân tích quan điểm trên bộ dữ liệu UIT-VSFC. Mô hình đạt được độ chính xác 93.6% trên tập Test, với F1-Score 0.950 cho Negative và 0.954 cho Positive.

6.2 Đóng góp chính

1. **Ứng dụng thành công PhoBERT Large:** Fine-tuning thành công mô hình 370 triệu tham số cho bài toán phân loại quan điểm tiếng Việt, đạt hiệu năng cao.
2. **Tiền xử lý dữ liệu thông minh:** Xây dựng quy trình tiền xử lý tùy chỉnh để chuyển đổi teen-code và emoji dạng text về ngôn ngữ tự nhiên, giúp mô hình hiểu được cảm xúc ẩn chứa.
3. **Tối ưu hóa chiến lược:** Gộp tập Train và Dev sau khi tìm được siêu tham số tối ưu đã giúp cải thiện độ chính xác lên 93.6%.
4. **Phân tích chi tiết:** Đánh giá và phân tích kỹ lưỡng các lỗi phân loại, đặc biệt là vấn đề mất cân bằng dữ liệu ở lớp Neutral.

6.3 Kết quả đạt được

- **Accuracy:** 93.6% - Độ chính xác tổng thể cao
- **Weighted F1-Score:** 0.934 - Hiệu năng tốt khi xét đến sự mất cân bằng dữ liệu
- **F1-Score Negative:** 0.950 - Xuất sắc
- **F1-Score Positive:** 0.954 - Xuất sắc
- **F1-Score Neutral:** 0.609 - Cần cải thiện

6.4 Ứng dụng thực tế

Mặc dù còn hạn chế ở lớp Trung tính do thiếu dữ liệu, mô hình hoàn toàn đáp ứng tốt nhu cầu phân tích thực tế cho các phản hồi Tích cực và Tiêu cực. Trong thực tế, việc phân biệt phản hồi tích cực và tiêu cực là mục tiêu chính của bài toán phân tích quan điểm, và mô hình đã đạt được mục tiêu này một cách xuất sắc.

6.5 Hướng phát triển

1. **Cải thiện lớp Neutral:** Áp dụng class-weighted loss, data augmentation, và focal loss để cải thiện hiệu năng trên lớp thiểu số.
2. **Mở rộng ứng dụng:** Áp dụng phương pháp tương tự cho các bài toán phân tích quan điểm khác trong tiếng Việt.
3. **Tối ưu hóa:** Nghiên cứu các kỹ thuật quantization và pruning để giảm kích thước mô hình mà vẫn giữ được hiệu năng cao.
4. **Ensemble:** Kết hợp nhiều mô hình để tăng độ chính xác và tính ổn định.

Tài liệu

[1] K. V. Nguyen, V. D. Nguyen, P. X. V. Nguyen, T. T. H. Truong, and N. L.-T. Nguyen, "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis," in *Proceedings of the 10th International Conference on Knowledge and Systems Engineering (KSE)*, Ho Chi Minh City, Vietnam, 2018, pp. 19–24.

[2] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 1037–1042.

[3] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

Hết báo cáo