HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

XỬ LÝ NGÔN NGỮ TỰ NHIÊN

ĐỀ TÀI: PHÂN LOẠI CẢM XÚC VĂN BẢN SỬ DỤNG MÔ HÌNH NGÔN NGỮ LỚN

Nhóm : 1

Trương Thị Thu Hà : B24CHKH006

Nguyễn Quang Huy : B24CHKH012

Nguyễn Văn Khánh : B24CHKH014

Nguyễn Thế Anh : B24CHKH001

Hà Nội năm 2024

Công việc

- Tìm kiếm model và dataset.
- Finetune llama-3 8B ,BERT bnar 110M ,RoBERTa bản 125M(trên Kaggle) trên tập financial_phrasebank.
- Finetune Llama 3 bản 8B, Gemma2 bản 2B, Gemma 1 bản 8B, và Phi3 bản 8B (trên máy local) trên tập financial_phrasebank .
- Làm slide và thuyết trình phần I (Tổng quan) và II (Các phương pháp được sử dụng).
- Làm slide và thuyết trình phần III (Các mô hình ngôn ngữ lớn) và IV (Kỹ thuật tinh chỉnh tham số hiệu quả).
- Tổng hợp code và demo.
- Viết báo cáo

Mục lục

I. Tổng quan

1.Giới thiệu bài toán

- Bài toán phân loại cảm xúc là một lĩnh vực nghiên cứu quan trọng trong tâm lý học, ngôn ngữ học và trí tuệ nhân tạo, nhằm xác định và phân loại các cảm xúc mà con người thể hiện qua lời nói, văn bản hoặc hành vi. Mục tiêu của bài toán này là phát triển các hệ thống có khả năng nhận diện và hiểu được các cảm xúc như vui vẻ, buồn bã, tức giận, hay lo lắng từ ngữ cảnh giao tiếp. Việc phân loại cảm xúc không chỉ giúp cải thiện khả năng tương tác giữa con người và máy tính, mà còn có ứng dụng rộng rãi trong các lĩnh vực như chăm sóc sức khỏe tâm thần, phân tích thị trường, và truyền thông xã hội. Thông qua việc phân tích cảm xúc, các nhà nghiên cứu và chuyên gia có thể phát hiện các mẫu hành vi và xu hướng, từ đó đưa ra các giải pháp hỗ trợ hiệu quả hơn cho con người.
- Trong lĩnh vực tài chính và kinh tế, bài toán phân loại cảm xúc đóng vai trò then chốt trong việc phân tích tâm lý thị trường và quyết định đầu tư. Cảm xúc của nhà đầu tư có thể ảnh hưởng mạnh mẽ đến hành vi giao dịch và xu hướng của thị trường, khiến cho việc đánh giá các cảm xúc này trở nên quan trọng. Bằng cách áp dụng các kỹ thuật phân tích cảm xúc lên các nguồn dữ liệu như tin tức, báo cáo tài chính, và bình luận của nhà đầu tư trên mạng xã hội, các chuyên gia có thể xác định được những xu hướng tâm lý tích cực hoặc tiêu cực. Thông qua việc phân loại cảm xúc trong bối cảnh tài chính, các nhà đầu tư có thể đưa ra các quyết định đầu tư thông minh hơn, tối ưu hóa lợi nhuận và giảm thiểu rủi ro.
- Trong báo cáo này, chúng tôi sẽ trình bày về quá trình fine-tune mô hình nhận diện cảm xúc trong tài chính và kinh tế, nhằm nâng cao khả năng nhận diện và phân loại cảm xúc trong dữ liệu văn bản.
-

2.Thách thức

- Phân tích cảm xúc là một thách thức lớn do đặc điểm phức tạp của ngôn ngữ tự nhiên. Một số vấn đề thường gặp bao gồm:
    - **Sự đa nghĩa và ngữ cảnh**: Một từ hoặc cụm từ có thể mang nhiều ý nghĩa khác nhau tùy thuộc vào ngữ cảnh, ví dụ như từ "cool" có thể biểu đạt sự thích thú hoặc sự lạnh lẽo. Do đó, việc xác định chính xác cảm xúc trong ngữ cảnh cụ thể là điều không dễ dàng.
    - **Biểu đạt cảm xúc phức tạp**: Người dùng thường biểu đạt cảm xúc qua nhiều cách khác nhau như mỉa mai, châm biếm, hoặc sử dụng các biểu tượng cảm xúc. Những trường hợp này rất khó nhận diện chính xác bằng các phương pháp truyền thống.
    - **Độ lớn của dữ liệu**: Các tập dữ liệu phân tích cảm xúc thường có quy mô lớn và đa dạng, đòi hỏi mô hình phải có khả năng xử lý hiệu quả và chính xác. Điều này càng trở nên khó khăn hơn khi phải xử lý các ngôn ngữ khác nhau hoặc các dạng văn bản ngắn gọn như bình luận trên mạng xã hội.
- Các thách thức trong phân loại cảm xúc văn bản không chỉ nằm ở độ phức tạp của ngôn ngữ mà còn ở tính đa dạng của cảm xúc và ngữ cảnh trong đó chúng xuất hiện. Mỗi phương pháp đều có những ưu và nhược điểm riêng để xử lý những thách thức này.
-

3.Phân loại các lớp

- Việc phân loại thông điệp thành các lớp cảm xúc khác nhau là rất quan trọng để hiểu rõ hơn về tâm lý thị trường và phản ứng của công chúng đối với các sự kiện tài chính.
- Trong phân loại cảm xúc, có nhiều cách tiếp cận, trong đó phân loại đa nhãn (multi-label classification) cho phép một văn bản chứa nhiều loại cảm xúc đồng thời. Ví dụ, một câu có thể vừa mang tính châm biếm (tiêu cực) nhưng cũng thể hiện sự hài hước (tích cực). Những

tập dữ liệu như go_emotions hỗ trợ phân loại này, giúp mô hình nhận diện nhiều loại cảm xúc trong một văn bản.

- Tuy nhiên, trong báo cáo này, chúng tôi sẽ tập trung vào ba lớp cảm xúc chính: **positive** (tích cực), **negative** (tiêu cực) và **neutral** (trung tính). Cách tiếp cận này giúp đơn giản hóa quá trình phân tích và dễ dàng hơn trong việc hiểu rõ hơn về nội dung văn bản:
    - Lớp **positive:** Thể hiện cảm xúc tích cực hoặc kết quả tốt đẹp. Chúng thường liên quan đến các giao dịch, thỏa thuận hoặc sự kiện mà được coi là thành công hoặc có lợi.
    - Lớp **negative:** Thể hiện cảm xúc tiêu cực hoặc kết quả không thuận lợi. Chúng thường liên quan đến sự mất mát, sa thải, hoặc các tình huống xấu khác. Những câu này thường truyền tải thông tin về thiệt hại, khó khăn, hoặc những tin tức không tốt.
    - Lớp **neutral**: Thể hiện thông tin mà không mang tính tích cực hay tiêu cực. Chúng chỉ đơn thuần là thông báo về sự kiện hoặc tình huống mà không bộc lộ cảm xúc mạnh mẽ.
    -

4.Phương pháp đánh giá chất lượng

Trong phân tích cảm xúc, để đánh giá hiệu suất của các mô hình, người ta thường sử dụng các chỉ số đo lường sau:

- **Độ chính xác (Accuracy):**
    - Độ chính xác là tỷ lệ phần trăm số dự đoán đúng của mô hình trên tổng số dữ liệu kiểm tra. Chỉ số này được tính bằng công thức:

$$\boldsymbol{Accuracy = \frac{Số\ dự\ đoán\ đúng}{Tổng\ số\ dự\ đoán}}$$

    - Độ chính xác cho biết mô hình thực hiện tốt như thế nào trên toàn bộ tập dữ liệu, nhưng nó không phản ánh chính xác hiệu suất trong các trường hợp mà dữ liệu không cân bằng (ví dụ: khi số lượng nhãn cảm xúc tích cực lớn hơn nhiều so với nhãn cảm xúc tiêu cực).
- **Độ chính xác (Precision):**
    - Độ chính xác (Precision) là tỷ lệ giữa số lượng dự đoán đúng cho một nhãn cụ thể và tổng số dự đoán cho nhãn đó. Nó cho biết có bao nhiêu dự đoán của mô hình là đúng trên tổng số các dự đoán mà mô hình đưa ra cho nhãn đó. Công thức tính độ chính xác:

$$\boldsymbol{Precision = \frac{Số\ lượng\ dự\ đoán\ đúng\ cho\ một\ nhãn}{Tổng\ số\ dự\ đoán\ cho\ nhãn\ đó}}$$

    - Precision đặc biệt quan trọng trong các trường hợp mà chi phí của việc dự đoán sai là cao, ví dụ như khi mô hình xác định các phản hồi tiêu cực để xử lý ưu tiên.
- **Độ hồi đáp (Recall):**
    - Độ hồi đáp (Recall) là tỷ lệ giữa số lượng dự đoán đúng cho một nhãn và tổng số nhãn thực tế của dữ liệu. Nó cho biết mô hình tìm ra được bao nhiêu phần trăm của các nhãn đó trong dữ liệu kiểm tra. Công thức tính độ hồi đáp:

$$\boldsymbol{Recall = \frac{Số\ lượng\ dự\ đoán\ đúng\ cho\ một\ nhãn}{Tổng\ số\ nhãn\ thực\ tế\ của\ dữ\ liệu}}$$

    - Recall quan trọng khi cần giảm thiểu số lượng các nhãn bị bỏ sót, chẳng hạn như trong các hệ thống phát hiện spam hoặc các phản hồi tiêu cực.
- **F1-Score:**
    - F1-Score là trung bình điều hòa của Precision và Recall, cung cấp một chỉ số cân bằng giữa hai chỉ số này. Nó đặc biệt hữu ích trong các tình huống mà việc đạt được sự cân bằng giữa Precision và Recall là cần thiết. Công thức tính F1-Score:

$$F1 = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

- F1-Score dao động từ 0 đến 1, với giá trị càng gần 1 thì hiệu suất của mô hình càng tốt. Đây là một chỉ số được sử dụng phổ biến trong các bài toán phân tích cảm xúc để đánh giá toàn diện hiệu suất của mô hình.

- **Độ chính xác trung bình có trọng số (Weighted Average):**
  - Khi làm việc với các tập dữ liệu không cân bằng, một số loại nhãn có thể chiếm ưu thế hơn những loại khác. Trong trường hợp này, việc tính toán độ chính xác, độ hồi đáp và F1-Score có trọng số theo từng nhãn sẽ giúp đưa ra một đánh giá chính xác hơn về hiệu suất của mô hình trên toàn bộ tập dữ liệu.
  - Trọng số của mỗi lớp chính là tỉ lệ số lượng mẫu của lớp đó trên tổng số mẫu trong toàn bộ tập dữ liệu. Điều này có nghĩa là các lớp có nhiều mẫu hơn sẽ đóng góp nhiều hơn vào kết quả tổng. Trọng số cho từng lớp được cho bởi công thức:

$$Weight_i = \frac{Số\ lượng\ mẫu\ của\ lớp\ i}{Tổng\ số\ mẫu\ của\ tập\ dữ\ liệu}$$

  - Độ chính xác trung bình có trọng số: Được tính bằng cách nhân precision của từng lớp với trọng số tương ứng, rồi cộng tất cả lại:

$$Weighted\ Average\ Precision = \sum_{i=1}^{n} (Precision_i \times Weight_i)$$

II. Các phương pháp đã được sử dụng

1. Dựa trên từ điển [1]

- Phân tích cảm xúc dựa trên từ điển (lexicon-based sentiment analysis) là một kỹ thuật trong xử lý ngôn ngữ tự nhiên (NLP) được sử dụng để phát hiện cảm xúc của một đoạn văn bản. Phương pháp này sử dụng danh sách các từ và cụm từ (từ điển cảm xúc hoặc từ vựng) được gắn với các trạng thái cảm xúc khác nhau để gán nhãn cho các từ và xác định cảm xúc.
- Các từ được gán nhãn với sự trợ giúp của một loại từ điển gọi là từ điển giá trị cảm xúc (valence dictionary). Mỗi từ trong văn bản có thể mang một giá trị cảm xúc nhất định, từ đó giúp chúng ta có được ấn tượng tích cực hay tiêu cực về một đối tượng, ví dụ như hãng hàng không.
- Ví dụ, với câu: “Good airlines sometimes have bad days.”, một từ điển giá trị cảm xúc sẽ gán nhãn từ "Good" là tích cực, từ "bad" là tiêu cực, và có thể các từ còn lại sẽ được xem là trung lập.
- Sau khi mỗi từ trong văn bản được gán nhãn, ta tính toán tổng điểm cảm xúc bằng cách đếm số lượng từ tích cực và tiêu cực, sau đó kết hợp các giá trị đó để đưa ra một kết quả tổng quát.
- Một công thức phổ biến để tính tổng điểm cảm xúc (StSc) là:

$$StSc = \frac{Số\ từ\ tích\ cực - số\ từ\ tiêu\ cực}{Tổng\ số\ từ}$$

  Nếu điểm cảm xúc là âm, văn bản sẽ được phân loại là tiêu cực. Nếu điểm cảm xúc dương, văn bản được coi là tích cực, và nếu điểm là 0 thì văn bản sẽ được xếp vào loại trung tính.
- Trong phương pháp dựa trên từ điển (lexicon-based), chúng ta có thể bỏ qua toàn bộ quá trình xây dựng mô hình học máy. Thay vào đó, ta có thể phân tích cảm xúc ngay lập tức, tính toán các điểm cảm xúc dựa trên từ điển cảm xúc (valence dictionary) của mình.
- Ưu điểm:
  - **Dễ hiểu và triển khai**: Phương pháp này đơn giản để thực hiện, không yêu cầu mô hình học sâu phức tạp hay lượng lớn dữ liệu huấn luyện. Chỉ cần một từ điển từ ngữ chứa các từ và mức độ cảm xúc tương ứng của chúng.

    - **Hiệu quả với dữ liệu nhỏ**: Phương pháp này có thể hoạt động hiệu quả mà không cần phải đào tạo một mô hình phức tạp.
    - **Dễ dàng tùy chỉnh**: Có thể dễ dàng cập nhật và tùy chỉnh từ điển để phản ánh các ngữ nghĩa và từ mới theo thời gian, phù hợp với lĩnh vực hoặc ngữ cảnh cụ thể.
    - **Khả năng giải thích**: Kết quả của phân tích cảm xúc có thể được giải thích dễ dàng hơn, vì bạn có thể theo dõi các từ ngữ cụ thể đã ảnh hưởng đến kết quả cảm xúc.
    - **Khả năng hoạt động trên nhiều ngôn ngữ**: Nếu có từ điển cảm xúc phù hợp, phương pháp này có thể áp dụng cho nhiều ngôn ngữ khác nhau.
- Tuy nhiên, các phương pháp dựa trên từ điển có những hạn chế nhất định, bao gồm:
    - **Không xử lý được ngữ cảnh**: Các từ điển cảm xúc thường không nắm bắt được ý nghĩa của từ trong ngữ cảnh cụ thể, dẫn đến những kết quả không chính xác khi từ đó mang nhiều nghĩa khác nhau.
    - **Không hiệu quả với ngôn ngữ phức tạp**: Những câu có cấu trúc phức tạp hoặc chứa ngôn ngữ châm biếm, ẩn dụ sẽ khó được phân tích chính xác chỉ dựa vào từ điển.

2. Học máy [2]

- Phân tích cảm xúc và học máy đã trở thành những công cụ quan trọng để đánh giá trải nghiệm của khách hàng. Chúng thường được sử dụng trong các trung tâm liên lạc để xác định cảm xúc của khách hàng trong quá trình giao tiếp của họ. Phân tích cảm xúc được thúc đẩy bởi học máy đã xuất hiện trong các nền tảng trò chuyện trực tiếp, phần mềm viết và chỉnh sửa, và nhiều lĩnh vực khác.
- Một số phương pháp học máy phổ biến được sử dụng cho bài toán phân loại cảm xúc:
    - **Support vector machine**
    - **Random forest**
    - **Naive Bayes**
    - **K-Nearest Neighbor**
    - **Hybrid machine learning and rules**
- Ưu điểm: Học máy có khả năng học từ dữ liệu và cải thiện hiệu suất theo thời gian, đặc biệt hữu ích trong việc phân loại các văn bản có khối lượng lớn.
- Hạn chế: Các mô hình học máy thường không nắm bắt được mối quan hệ ngữ nghĩa hoặc ngữ cảnh sâu trong văn bản. Ngoài ra, việc cần phải thực hiện tiền xử lý dữ liệu (như loại bỏ từ dừng, tạo đặc trưng) có thể tốn thời gian và không tối ưu cho các tình huống phức tạp.
-

2.1. Sử dụng Support vector machine [3]

- SVM là một kỹ thuật mạnh mẽ cho phân tích cảm xúc. Nó tìm ra một mặt phẳng tối ưu để phân tách các lớp, xử lý các không gian đặc trưng có chiều cao, và nắm bắt các mối quan hệ phức tạp giữa các từ và cảm xúc. SVM rất phù hợp cho các nhiệm vụ mà không gian đặc trưng và các mối quan hệ phi tuyến đóng vai trò quan trọng.
- Một trong những điểm mạnh chính của SVM nằm ở khả năng xử lý các ranh giới quyết định phi tuyến thông qua việc sử dụng các hàm kernel. Bằng cách sử dụng các kernel này, SVM có thể nắm bắt các mối quan hệ và mẫu phức tạp trong dữ liệu. Nguyên tắc cơ bản của SVM là tìm ra mặt phẳng tối ưu tối đa hóa sự phân tách giữa các lớp khác nhau trong không gian đặc trưng. Mặt phẳng này, còn được gọi là ranh giới quyết định, nhằm đạt được khoảng cách lớn nhất giữa các lớp, cung cấp một giải pháp vững chắc và tổng quát tốt.
- Các điểm dữ liệu nằm gần nhất với ranh giới quyết định, được gọi là vector hỗ trợ (support vectors), đóng vai trò quan trọng trong việc xác định ranh giới quyết định và hiệu suất phân loại tổng thể. SVM cung cấp một số lợi thế, bao gồm khả năng xử lý các tập dữ liệu có tiếng ồn và tính bền vững đối với việc quá khớp (overfitting). Bằng cách tập trung vào việc tối đa

hóa khoảng cách trong khi giảm thiểu lỗi phân loại, SVM có thể cung cấp khả năng tổng quát tốt cho dữ liệu chưa thấy.

- Ngoài ra, SVM cho phép tích hợp các hàm kernel khác nhau, chẳng hạn như tuyến tính, đa thức, Gaussian (RBF) hoặc sigmoid, mang lại tính linh hoạt để điều chỉnh thuật toán cho nhiều loại dữ liệu và lĩnh vực vấn đề khác nhau. Để huấn luyện một mô hình SVM, một bài toán tối ưu hóa lồi được giải quyết, có thể được giải quyết hiệu quả bằng nhiều thuật toán tối ưu hóa khác nhau. Kết quả là một mô hình được tối ưu hóa tốt có thể phân loại chính xác các trường hợp mới dựa trên các mẫu và mối quan hệ đã học trong dữ liệu huấn luyện.
-

2.2. Sử dụng Random Forest [3]

- Random Forest là một phương pháp hiệu quả khác cho phân tích cảm xúc. Nó kết hợp nhiều cây quyết định để đưa ra dự đoán, mang lại khả năng chống quá khớp (overfitting) và xử lý phân phối lớp không cân bằng. Random Forest phù hợp cho các tác vụ mà khả năng giải thích, học tập tập hợp (ensemble learning) và xử lý tập dữ liệu phức tạp là những yếu tố quan trọng.
- Thuật toán hoạt động bằng cách tạo ra một số lượng lớn cây quyết định, mỗi cây được huấn luyện trên một tập con ngẫu nhiên của tập dữ liệu gốc. Những cây quyết định riêng lẻ này được xây dựng bằng cách chọn ngẫu nhiên các đặc trưng tại mỗi nút và thực hiện các phép tách dựa trên tiêu chí tốt nhất có thể, chẳng hạn như độ thuần khiết Gini (Gini impurity) hoặc độ tăng thông tin (information gain). Bằng cách kết hợp các dự đoán của nhiều cây quyết định, Random Forest giảm nguy cơ quá khớp và cải thiện độ chính xác cũng như độ ổn định tổng thể của mô hình.
- Một trong những ưu điểm chính của Random Forest là khả năng xử lý cả các đặc trưng phân loại và liên tục mà không cần chuẩn bị trước nhiều. Nó có thể xử lý dữ liệu thiếu và duy trì hiệu suất dự đoán tốt ngay cả khi có sự hiện diện của các biến nhiễu hoặc không liên quan. Random Forest cũng cung cấp những thông tin giá trị về tầm quan trọng của các đặc trưng, cho phép người dùng xác định các đặc trưng có ảnh hưởng nhất trong quá trình dự đoán. Thông tin này có thể hỗ trợ trong việc chọn lọc các đặc trưng và hiểu các mối quan hệ cơ bản trong tập dữ liệu.
- Thuật toán này cũng rất hiệu quả về mặt tính toán, vì việc huấn luyện các cây quyết định riêng lẻ có thể được thực hiện song song. Nó có thể xử lý các tập dữ liệu lớn với nhiều đặc trưng, khiến nó phù hợp cho cả ứng dụng dữ liệu nhỏ và lớn. Random Forest là một thuật toán đa năng có thể áp dụng trong nhiều lĩnh vực, bao gồm tài chính, y tế, tiếp thị và hơn thế nữa. Sự vững chắc, độ chính xác và khả năng xử lý dữ liệu phức tạp của nó khiến Random Forest trở thành lựa chọn phổ biến trong giới khoa học dữ liệu và các nhà nghiên cứu.
- Kết quả của nghiên cứu sử dụng SVM và Random Forest.

Table 3: Accuracy, Recall, Precision, F1-score for Random Forest and SVM

Algorithm	Accuracy	Recall	Precision	F1-score
Random Forest	0.78564	0.78564	0.78737	0.78527
SVM	0.80394	0.80394	0.80654	0.80347

2.3. Sử dụng Naïve Bayes [4]

- Bộ phân loại Naive Bayes là một thuật toán xác suất hiệu quả cho các nhiệm vụ phân loại, bao gồm phân tích cảm xúc. Nó dựa trên định lý Bayes, với giả định "naive" rằng các đặc trưng độc lập có điều kiện, nghĩa là sự hiện diện hoặc vắng mặt của một đặc trưng không ảnh hưởng đến các đặc trưng khác.
- Có hai biến thể chính của NB:

    - Bernoulli NB: Thiết kế cho dữ liệu nhị phân, nơi mỗi đặc trưng có thể nhận giá trị 0 hoặc 1. Nó tính toán xác suất dựa trên sự hiện diện hoặc vắng mặt của các đặc trưng nhị phân.
    - Multinomial NB: Thích hợp cho các đặc trưng đại diện cho tần số từ, nó tính xác suất dựa trên số lần xuất hiện của từ trong văn bản.
- Ngoài ra, một số cải tiến cho phương pháp này đã được đề xuất, như việc xác định trọng số cho các đặc trưng và áp dụng các kỹ thuật tối ưu hóa để nâng cao hiệu suất của bộ phân loại. Các nghiên cứu cho thấy NB, đặc biệt là Multinomial NB, có hiệu quả cao trong phân loại văn bản, với khả năng dự đoán chính xác các lớp cảm xúc như tích cực hoặc tiêu cực.
- Kết quả của nghiên cứu

Table 2. Performance result of unsupervised sentiment classification.

Items	VADER	Text Blob
Accuracy	0.7072	0.6852
Sensitivity	0.8673	0.9356
Specificity	0.5489	0.4375
Precision	0.6553	0.6219
F1	0.7465	0.7472
Roc_Auc	0.7081	0.6866

2.4. Sử dụng K-Nearest Neighbor [5]

- K-Nearest-Neighbors (KNN) là một thuật toán phân loại giám sát không tham số, đơn giản nhưng hiệu quả trong nhiều trường hợp. Bộ phân loại KNN được coi là bộ phân loại phổ biến nhất cho nhận dạng mẫu nhờ hiệu suất hiệu quả và kết quả hiệu quả của nó cũng như sự đơn giản của nó. Nó được sử dụng rộng rãi trong các lĩnh vực nhận dạng mẫu, học máy, phân loại văn bản, khai thác dữ liệu, nhận dạng đối tượng và nhiều lĩnh vực khác.
- Thuật toán KNN phân loại bằng cách so sánh, tức là so sánh điểm dữ liệu không biết với các điểm dữ liệu trong tập huấn luyện mà nó tương tự. Độ tương đồng được đo bằng khoảng cách Euclid. Các giá trị thuộc tính được chuẩn hóa để ngăn chặn các thuộc tính có khoảng giá trị lớn hơn chiếm ưu thế so với các thuộc tính có khoảng giá trị nhỏ hơn. Trong phân loại KNN, mẫu không biết được gán cho lớp chiếm ưu thế nhất trong số các lớp của các hàng xóm gần nhất của nó. Trong trường hợp có sự hòa giữa hai lớp cho mẫu, lớp có khoảng cách trung bình nhỏ nhất tới mẫu không biết sẽ được gán.
- Thông qua sự kết hợp của một số hàm khoảng cách địa phương dựa trên từng thuộc tính, một hàm khoảng cách toàn cục dist có thể được tính toán.

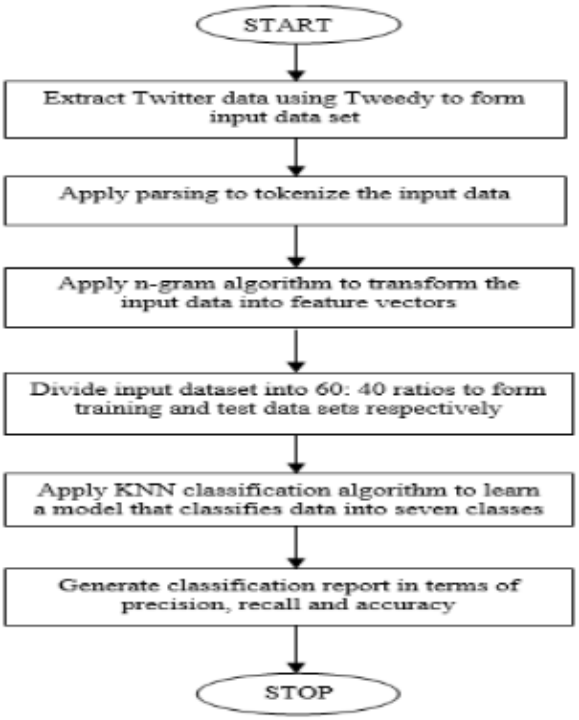


Fig. 1: Proposed Methodology.

- Kết quả của nghiên cứu

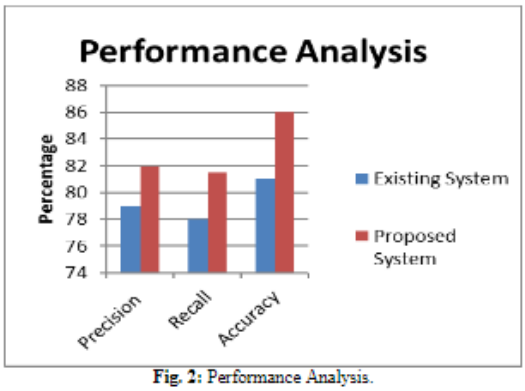


Fig. 2: Performance Analysis.

2.5. Hybrid machine learning and rules [2]

- Một số giải pháp phần mềm phân tích cảm xúc dựa vào quy tắc thay vì học máy. Các hệ thống dựa trên quy tắc bao gồm và loại trừ các từ và cụm từ cụ thể như tích cực hoặc tiêu cực. Những hệ thống này thường hoạt động tốt nhất trong các ngành nghề đặc thù với thuật ngữ thường xuyên được sử dụng, điều này có thể khó trừu tượng hóa đối với các mô hình học máy thông thường.
- Các mô hình hybrid kết hợp cả học máy và các hệ thống dựa trên quy tắc, cho phép cả hai loại hệ thống hoạt động cùng nhau để có cách tiếp cận toàn diện hơn đối với phân tích cảm xúc. Các công ty sử dụng phương pháp này có thể sử dụng một hệ thống học máy để tạo ra các quy tắc hoặc cung cấp chúng vào hệ thống học máy của mình để giúp nó phát triển.
-

3. Học sâu

- Học sâu có ảnh hưởng rất lớn trong cả học có giám sát và không giám sát, nhiều nhà nghiên cứu đang xử lý phân tích cảm xúc bằng cách sử dụng học sâu. Nó bao gồm nhiều mô hình hiệu quả và phổ biến, và những mô hình này được sử dụng để giải quyết nhiều vấn đề một cách hiệu quả.
- Một số phương pháp học sâu đã được sử dụng cho bài toán phân loại cảm xúc:
    - **Convolutional neural networks (CNN).**

    - **Recursive Neural Network (RNN).**
    - **Deep Neural Networks (DNN).**
    - **Recurrent Neural Networks (Recurrent NN).**
    - **Deep Belief Networks (DBN).**
    - **Long Short Term Memory (LSTM).**
    - **Transformers-based Models.**
- Ưu điểm:
    - Khả năng học đặc trưng tự động: Học sâu có khả năng tự động trích xuất và học các đặc trưng phức tạp từ dữ liệu mà không cần phải thiết kế thủ công, giúp tiết kiệm thời gian và công sức.
    - Xử lý dữ liệu lớn: Các mô hình học sâu, đặc biệt là mạng nơ-ron, có khả năng xử lý một lượng lớn dữ liệu, từ đó cải thiện độ chính xác và hiệu suất của mô hình.
    - Khả năng nắm bắt ngữ nghĩa: Học sâu có thể nắm bắt các mối quan hệ ngữ nghĩa phức tạp trong văn bản, giúp cải thiện việc phân loại cảm xúc, đặc biệt trong các ngữ cảnh đa dạng.
    - Hiệu suất cao: Nhiều nghiên cứu đã chỉ ra rằng các mô hình học sâu thường đạt độ chính xác cao hơn so với các phương pháp truyền thống trong nhiều bài toán phân loại cảm xúc.
- Nhược điểm:
    - Cần nhiều dữ liệu: Các mô hình học sâu thường yêu cầu một lượng lớn dữ liệu để được huấn luyện hiệu quả, điều này có thể không khả thi trong một số trường hợp.
    - Thời gian huấn luyện lâu: Quá trình huấn luyện các mô hình học sâu có thể tốn nhiều thời gian, đặc biệt là khi sử dụng các kiến trúc phức tạp.
    - Khó khăn trong việc điều chỉnh siêu tham số: Học sâu có nhiều siêu tham số cần được điều chỉnh, điều này có thể làm cho việc tối ưu hóa mô hình trở nên khó khăn và tốn thời gian.
    - Thiếu khả năng giải thích: Các mô hình học sâu thường được coi là "hộp đen", khiến cho việc hiểu và giải thích quyết định của mô hình trở nên khó khăn hơn so với các phương pháp truyền thống.
    - Nguy cơ overfitting: Nếu không có đủ dữ liệu hoặc nếu mô hình quá phức tạp, có nguy cơ mô hình sẽ bị overfitting, tức là học quá mức vào dữ liệu huấn luyện và không hoạt động tốt trên dữ liệu mới.

3.1. Convolutional neural networks (CNN) [6]

- Convolutional neural networks (CNN) bao gồm các lớp gộp và phức tạp vì nó cung cấp một kiến trúc tiêu chuẩn để ánh xạ các câu có độ dài khác nhau thành các vector phân tán kích thước cố định của câu.
- Để huấn luyện mạng nơ-ron, các thuật toán tối ưu hàm không lồi và phương pháp giảm độ dốc ngẫu nhiên (SGD) đã được sử dụng, và thuật toán lan truyền ngược (backpropagation) được áp dụng để tính toán gradient. Kỹ thuật Dropout được sử dụng để cải thiện khả năng điều chuẩn (regularization) của mạng nơ-ron.
- Phân tích của các nghiên cứu tốt nhất theo sau mạng nơ-ron tích chập (CNN):

TABLE I. ANALYSIS OF CONVOLUTIONAL NEURAL NETWORKS

Researcher Name and Year	Model Used	Purpose	Data Set	Results
J.Islam and Y. Zhang 2016 [25]	Convolutional Neural Networks (CNN)	Visual SA	1269 images from twitter	GoogleNet gave almost 9 % performance progress than AlexNet.
A. Severyn and A. Moschitti, 2015 [26]	Convolutional Neural Networks (CNN)	Phrase level and message level task SA	Semeval-2015	Compared with official system ranked 1st in terms of pharse level subtask and ranked 2nd in terms of message level.
L. Yanmei and C. Yuda, 2015 [27]	Convolutional Neural Networks (CNN)	Micro-Blog SA	1000 micro-blog comments (HuaQiangu)	Proposed model can effectively improve the accuracy of emotional orientation, validation.
Q. You, J. Luo, H. Jin, and J. Yang, 2015 [28]	Convolutional Neural Networks (CNN)	Textual-visual SA	Getty Images, 101 keywords	Joint visual and textual model outperforms the early single fusions.
X. Ouyang, P. Zhou, C. H. Li, and L. Liu, 2015 [15]	Convolutional Neural Networks (CNN)	Sentiments of sentences	rottentomatoes.com (contains movie review excerpts)	The proposed model outperformed the previous models with the 45.5% accuracy.

3.2. Recursive Neural Network (RNN). [6]

- Recursive Neural Network (RNN) thuộc về học có giám sát. Nó chứa một cấu trúc cây được thiết lập trước khi huấn luyện và các nút có thể có các ma trận khác nhau. Trong RNN không cần tái cấu trúc đầu vào.
- Trong nghiên cứu của R. Socher, A. Perelygin, and J. Wu RNTN đạt độ chính xác 80.7% trong dự đoán cảm xúc bằng cách thực hiện gán nhãn chi tiết cho tất cả các cụm từ và vượt qua các mô hình trước đó.
- Phân tích các phương pháp tiếp cận tốt nhất dựa trên mạng nơ-ron hồi quy (RNN):

TABLE II. ANALYSIS OF RECURSIVE NEURAL NETWORKS

Researcher Name and Year	Model Used	Purpose	Data Set	Results
C. Li, B. Xu, G. Wu, S. He, G. Tian, and H. Hao, 2014 [29]	Recursive Neural Deep Model (RNDM)	Chines sentiments analysis of social data	2270 movie reviews from websites	Performs higher (90.8%) than baselines with a great margin.
R. Socher, A. Perelygin, and J. Wu, 2013 [30]	RNTN (Recursive Neural Tensor Network)	Semantic Compositionality	11,855 single sentences from moview review (Pang and Lee 2005)	The RNTN achieved 80.7% accuracy in sentiment prediction , an improvement of 9.7 % over baselines (bag of features).
W. Li and H. Chen, 2014 [31]	Recursive Neural Network (RNN)	Identifying Top Sellers In Underground Economy	Russian carding Forum)	Results have been indicated that Deep learning techniques accomplish superior outcomes than shallow classifiers. Carding sellers have fewer ratings than malware sellers.

3.3. Long Short Term Memory (LSTM) [7]

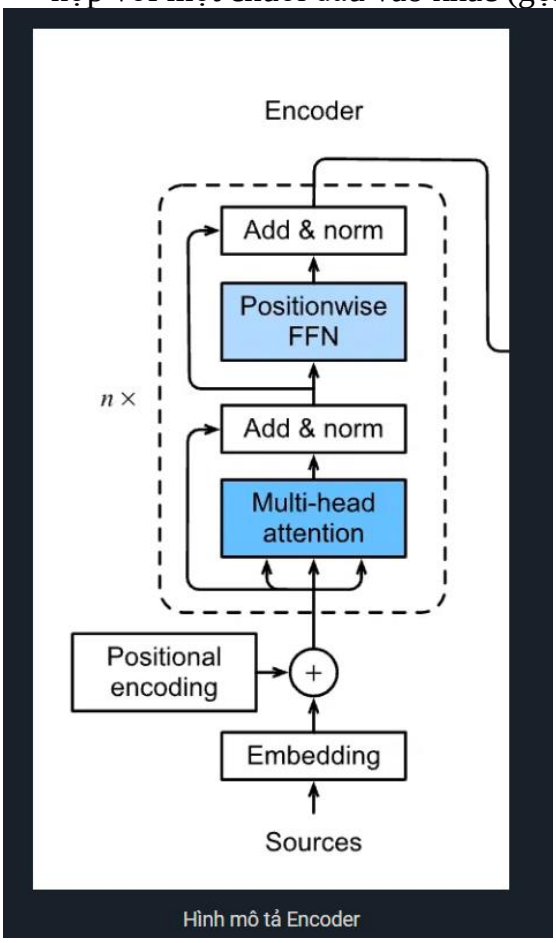
- Mạng LSTM là phần mở rộng của RNN được thiết kế để học dữ liệu tuần tự (thời gian) và các kết nối dài hạn của chúng chính xác hơn so với RNN tiêu chuẩn. Chúng thường được sử dụng trong các ứng dụng học sâu như dự báo chứng khoán, nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên.
- Mạng LSTM là Mạng nơ-ron hồi quy (RNN)) có khả năng xử lý dữ liệu tuần tự và nắm bắt các phụ thuộc lâu dài. Phân tích cảm xúc, kết hợp với mạng LSTM, cung cấp một khung mạnh mẽ để hiểu và khai thác các tông cảm xúc trong dữ liệu văn bản. Khả năng này rất quan trọng trong việc đưa ra quyết định dựa trên dữ liệu trong các bối cảnh kinh doanh và nghiên cứu.

3.4. Transformers-based Models [8]

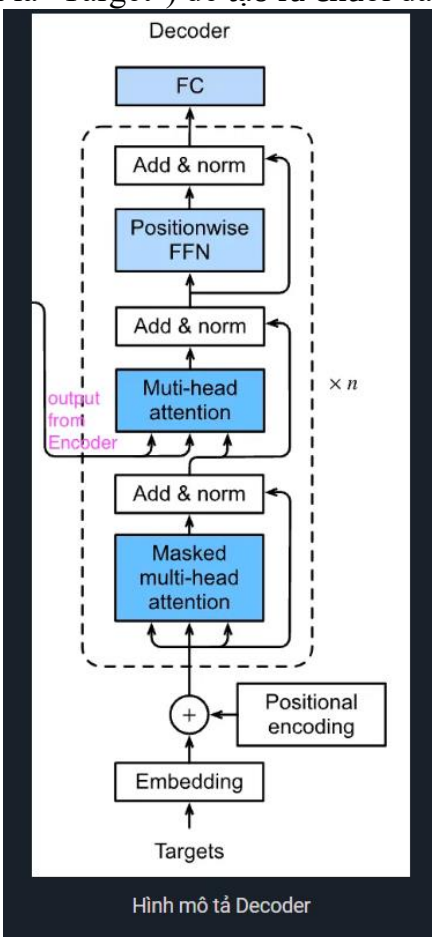
- Transformers là một kiến trúc mô hình học sâu dựa trên cơ chế self-attention, cho phép mô hình này hiểu được mối quan hệ giữa các từ trong một câu mà không cần đến kiến trúc tuần tự truyền thống như RNN (Recurrent Neural Networks) hay LSTM (Long Short-Term

Memory). Transformers có khả năng xử lý toàn bộ câu cùng một lúc, điều này giúp tăng tốc độ huấn luyện và cải thiện hiệu quả xử lý.

- Transformers được cấu trúc thành hai phần chính là encoder và decoder.
    - Encoder: Encoder xử lý dữ liệu đầu vào (gọi là "Source") và nén dữ liệu vào vùng nhớ hoặc context mà Decoder có thể sử dụng sau đó.
    - Decoder: Decoder nhận đầu vào từ đầu ra của Encoder (gọi là "Encoded input") kết hợp với một chuỗi đầu vào khác (gọi là "Target") để tạo ra chuỗi đầu ra cuối cùng.



Hình mô tả Encoder



Hình mô tả Decoder

- Sử dụng Transformers cho bài toán phân loại cảm xúc là một phương pháp hiệu quả và phổ biến hiện nay. Sử dụng Transformers trong các bài toán như phân loại cảm xúc mang lại nhiều lợi ích vượt trội so với các phương pháp truyền thống.
    - Khả năng nắm bắt ngữ nghĩa: Transformers sử dụng cơ chế attention để nắm bắt mối quan hệ giữa các từ trong câu, cho phép mô hình hiểu ngữ nghĩa một cách sâu sắc hơn. Điều này rất quan trọng trong phân loại cảm xúc, nơi mà ngữ cảnh có thể thay đổi ý nghĩa của từ.
    - Xử lý dữ liệu song song: So với các mô hình tuần tự như RNN, Transformers cho phép xử lý tất cả các từ trong câu cùng một lúc, giúp tăng tốc độ huấn luyện và hiệu suất tính toán.
    - Khả năng mở rộng: Transformers có thể dễ dàng mở rộng với nhiều lớp và số lượng tham số lớn. Những mô hình như BERT và GPT-3 đã chứng minh rằng kích thước lớn có thể cải thiện hiệu suất trong nhiều tác vụ.
    - Khả năng chuyển giao kiến thức: Các mô hình Transformers thường được huấn luyện trên một lượng lớn dữ liệu (pre-trained) và có thể được tinh chỉnh cho các tác vụ cụ thể (fine-tuned), giúp tiết kiệm thời gian và tài nguyên cho người dùng.

- Tính linh hoạt: Transformers không chỉ phù hợp cho phân loại cảm xúc mà còn có thể được áp dụng cho nhiều nhiệm vụ khác trong NLP như dịch máy, sinh văn bản, và trả lời câu hỏi.
- Thích ứng với ngữ cảnh: Khác với các mô hình từ điển tĩnh, Transformers sử dụng embeddings động, cho phép từ có thể có nhiều ý nghĩa khác nhau tùy theo ngữ cảnh, điều này giúp cải thiện độ chính xác trong phân loại cảm xúc.
- Hiệu suất vượt trội: Các nghiên cứu đã chỉ ra rằng mô hình Transformers thường đạt được độ chính xác cao hơn trong các bài toán phân loại cảm xúc so với các phương pháp truyền thống như SVM, Naive Bayes, hay RNN.

III. Các mô hình ngôn ngữ

1.BERT và các biến thể (SiEBERT, RoBERTa, DistilBERT)

- **BERT (Bidirectional Encoder Representations from Transformers)**: BERT là một trong những mô hình ngôn ngữ lớn tiên phong sử dụng kiến trúc transformer theo chiều hai hướng, giúp nắm bắt ngữ cảnh của từ bằng cách xem xét cả hai phía (trái và phải) của từ trong câu. Quá trình huấn luyện trước của BERT dựa trên hai nhiệm vụ chính:
  - Masked Language Modeling (MLM): Một số từ trong câu được "che giấu" và mô hình được yêu cầu dự đoán từ bị che giấu đó dựa trên ngữ cảnh.
  - Next Sentence Prediction (NSP): Mô hình học cách xác định xem một câu có khả năng là câu tiếp theo của một câu trước đó hay không. ***[9]***
- **Tinh chỉnh (fine-tuning)**: Sau khi huấn luyện trước, BERT có thể được tinh chỉnh cho các nhiệm vụ NLP cụ thể như phân loại văn bản, nhận diện thực thể, hoặc phân tích cảm xúc.
- Các biến thể của BERT:
  - **SiEBERT**: Một biến thể của BERT được tinh chỉnh đặc biệt cho các tác vụ phân tích cảm xúc. SiEBERT cải thiện hiệu suất bằng cách học từ một tập dữ liệu cảm xúc được dán nhãn cẩn thận, giúp tăng cường khả năng nhận diện cảm xúc từ văn bản. ***[10]***
  - **RoBERTa** (Robustly Optimized BERT Approach): RoBERTa là một cải tiến của BERT, trong đó quá trình huấn luyện trước được thực hiện trên tập dữ liệu lớn hơn và sử dụng các siêu tham số tối ưu hơn. RoBERTa không sử dụng nhiệm vụ NSP, thay vào đó tập trung vào MLM với một số cải tiến, giúp tăng cường hiệu suất cho các nhiệm vụ NLP. ***[11]***
  - **DistilBERT**: Là một phiên bản nhẹ và nhanh hơn của BERT, được tạo ra bằng cách sử dụng kỹ thuật nén mô hình (knowledge distillation). Mặc dù có kích thước nhỏ hơn, DistilBERT vẫn giữ được khoảng 97% hiệu suất của BERT nhưng nhanh hơn khoảng 60%, làm cho nó phù hợp với các ứng dụng yêu cầu xử lý thời gian thực hoặc tài nguyên tính toán hạn chế. ***[12]***

2.Gemma 2 [13]

- Gemma: Gemma là một mô hình ngôn ngữ lớn được thiết kế với mục tiêu tối ưu hóa khả năng học các đặc trưng cảm xúc từ văn bản. Quá trình huấn luyện của Gemma tập trung vào việc sử dụng các tập dữ liệu chứa nhiều dạng cảm xúc khác nhau, giúp cải thiện hiệu suất trong các tác vụ liên quan đến phân tích cảm xúc.
- Gemma 2 là thế hệ mới nhất trong dòng Gemma của Google, được tạo ra để đáp ứng nhu cầu của các nhà phát triển và nhà nghiên cứu cần các công cụ AI mạnh mẽ và dễ quản lý.
- Gemma 2 tiếp tục truyền thống của dòng Gemma nguyên bản, sử dụng cùng công nghệ và nghiên cứu tiên tiến có trong các mô hình Gemini. Dòng sản phẩm bao gồm các biến thể như CodeGemma, RecurrentGemma và PaliGemma, mỗi biến thể được thiết kế cho các tác vụ AI cụ thể.

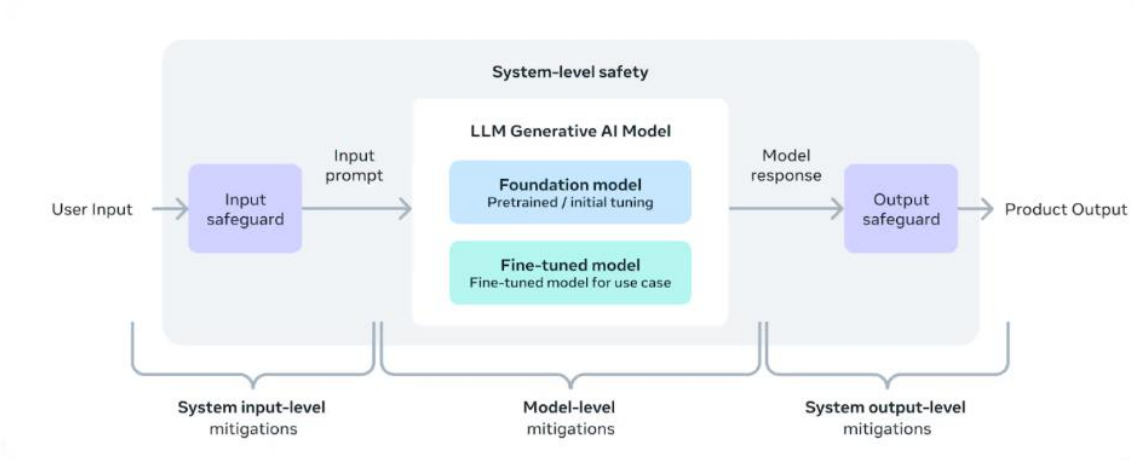
- So với các mô hình khác, Gemma có khả năng phân biệt tốt hơn giữa các trạng thái cảm xúc phức tạp, chẳng hạn như sự hài hước hoặc sự mỉa mai.

3.Phi [14]

- Phi là một mô hình ngôn ngữ nhỏ (SLM) được thiết kế để mang lại hiệu suất tốt trong khi vẫn đủ nhẹ để chạy trên các thiết bị hạn chế về tài nguyên như điện thoại thông minh. Với 3,8 tỷ tham số ấn tượng, Phi-3 là một cột mốc quan trọng trong công nghệ mô hình hóa ngôn ngữ nhỏ gọn.
- Phi-3 là một kiến trúc bộ giải mã transformer với độ dài ngữ cảnh mặc định là 4K, đảm bảo quá trình xử lý dữ liệu đầu vào hiệu quả trong khi vẫn duy trì khả năng nhận thức về ngữ cảnh. Phi-3 cũng cung cấp phiên bản ngữ cảnh dài, Phi-3-mini-128K, mở rộng độ dài ngữ cảnh lên 128K để xử lý các nhiệm vụ yêu cầu hiểu biết ngữ cảnh rộng hơn. Với 32 đầu và 32 lớp, Phi-3 cân bằng giữa độ phức tạp của mô hình và hiệu quả tính toán, làm cho nó phù hợp để triển khai trên các thiết bị di động.
- Phi-3 cho thấy khả năng sử dụng hiệu quả các tham số của mô hình, chứng minh rằng hiệu suất vượt trội có thể đạt được mà không cần tăng kích thước mô hình một cách exponentially. Bằng cách tạo ra sự cân bằng giữa độ phức tạp của mô hình và hiệu quả tài nguyên, Phi-3 thiết lập một tiêu chuẩn mới cho việc mô hình hóa ngôn ngữ quy mô nhỏ, cung cấp một sự lựa chọn hấp dẫn cho các mô hình lớn hơn, đòi hỏi tính toán cao hơn.
- Mặc dù Phi-3 có nhiều khả năng ấn tượng, nhưng nó cũng có một số hạn chế chính:
    - Kiến thức Factual Hạn chế: Phi-3-mini gặp khó khăn trong việc xử lý các nhiệm vụ yêu cầu kiến thức sâu rộng do không lưu trữ được nhiều thông tin thực tế.
    - Giới hạn Ngôn ngữ: Mô hình chủ yếu hoạt động trong tiếng Anh, hạn chế khả năng áp dụng trong môi trường đa ngôn ngữ.
    - Phụ thuộc vào Tài nguyên Bên ngoài: Phi-3-mini có thể cần dựa vào công cụ tìm kiếm để bổ sung kiến thức, tạo ra sự phụ thuộc không tối ưu.

4.Llama-3

- Llama 3 là một mô hình ngôn ngữ lớn (LLM) được phát triển bởi Meta AI, nổi tiếng với khả năng tạo văn bản, dịch ngôn ngữ và viết các loại nội dung sáng tạo khác nhau. ***[15]***
- Llama 3 là một mô hình ngôn ngữ lớn được thiết kế để hiểu và tạo ra văn bản giống như con người. Nó được xây dựng trên kiến trúc transformer tối ưu hóa, cho phép nó xử lý một loạt các nhiệm vụ, từ trả lời câu hỏi đến tạo mã. Mô hình có nhiều phiên bản, trong đó phiên bản 8 tỷ (8B) và 70 tỷ (70B) tham số là nổi bật nhất. Các mô hình này đã được tinh chỉnh để thực hiện theo hướng dẫn của con người một cách chính xác hơn, giúp chúng đặc biệt hiệu quả trong các ứng dụng dựa trên đối thoại như chatbot. ***[16]***
- Kiến trúc của mô hình Llama: Các mô hình Llama 3 có một tokenizer với vốn từ vựng của 128K tokens. Một vốn từ vựng lớn hơn có nghĩa là các mô hình có thể hiểu và xử lý văn bản tốt hơn. Ngoài ra, các mô hình hiện sử dụng chú ý truy vấn được nhóm (GQA) để cải thiện hiệu quả suy luận. GQA là một kỹ thuật mà bạn có thể coi là điểm nhấn giúp các mô hình tập trung vào các phần có liên quan của dữ liệu đầu vào để tạo ra các phản hồi nhanh hơn và chính xác hơn. ***[17]***



- Cải tiến công nghệ trong Llama 3
  - Cải tiến về kiến trúc và mã hóa: Meta đã triển khai kiến trúc bộ chuyển đổi chỉ giải mã chuẩn trong các mô hình Llama 3, đánh dấu sự chuyển dịch có chủ đích sang một khuôn khổ đơn giản và hiệu quả hơn cho xử lý ngôn ngữ. Kết hợp với một bộ mã hóa tự hào có vốn từ vựng 128K, các mô hình mã hóa ngôn ngữ hiệu quả hơn so với các mô hình tiền nhiệm của chúng. Kích thước vốn từ vựng lớn này đảm bảo rằng các mô hình có thể hiểu và tạo ra nhiều phản hồi giống con người hơn, tăng cường đáng kể khả năng áp dụng của chúng trong các bối cảnh hội thoại phức tạp.
  - Hiệu quả suy luận và đào tạo mô hình: Meta đã cải thiện hiệu quả suy luận của Llama 3 thông qua việc sử dụng grouped query attention (GQA) trên cả mô hình 8 tỷ và 70 tỷ tham số. GQA tối ưu hóa quá trình xử lý truy vấn bằng cách cho phép mô hình tập trung vào các phân đoạn dữ liệu có liên quan hiệu quả hơn, do đó tăng tốc thời gian phản hồi mà không làm giảm độ chính xác. Hơn nữa, các mô hình đào tạo trên chuỗi 8.192 mã thông báo, với các biện pháp cụ thể để ngăn chặn sự tự chú ý qua ranh giới tài liệu. Phương pháp này cải thiện khả năng xử lý tài liệu dài của mô hình bằng cách duy trì tính toàn vẹn ngữ cảnh trong toàn bộ văn bản. ***[18]***
- Những lợi thế của Llama 3:
  - Hiệu suất hiện đại: Llama 3 đã thiết lập một trạng thái tiên tiến mới cho LLM ở thang tham số 8B và 70B, vượt trội hơn các mô hình hàng đầu khác như GPT-4, Claude và Mistral trên các chuẩn mực như MMLU, HumanEval và các chuẩn mực khác với các đánh giá toàn diện của con người trên 12 trường hợp sử dụng chính cho thấy Llama 3 vượt trội trong các nhiệm vụ như lập luận phức tạp, viết sáng tạo và lập trình.
  - Kiến trúc được tối ưu hóa: Llama 3 sử dụng vốn từ vựng 128.000 mã thông báo và sự chú ý truy vấn được nhóm lại để cho phép mã hóa và suy luận hiệu quả hơn so với các phiên bản trước. Nó được đào tạo trên các chuỗi lên đến 8.192 mã thông báo, gấp đôi độ dài ngữ cảnh của Llama 2, để hiểu rõ hơn ở cấp độ tài liệu.
  - Cải thiện lý luận và hướng dẫn sau đây: Llama 3 cho thấy những tiến bộ đáng kể về khả năng lý luận, tạo mã và tuân theo hướng dẫn của con người một cách hiệu quả. Điều này được thực hiện bằng các kỹ thuật như học từ thứ hạng ưu tiên thông qua PPO và DPO trong quá trình đào tạo.
  - Khả năng truy cập nguồn mở: Là một mô hình nguồn mở, Llama 3 được cung cấp miễn phí cho các nhà nghiên cứu, nhà phát triển và tổ chức sử dụng, thúc đẩy sự đổi mới và cộng tác đồng thời cung cấp các công cụ như Llama Guard 2 để đảm bảo an toàn và Torchtune để dễ dàng tinh chỉnh nhằm hỗ trợ hệ sinh thái nguồn mở.
  - Dữ liệu đào tạo mở rộng: Mô hình Llama 3 được đào tạo trên bộ dữ liệu gồm hơn 15 nghìn tỷ mã thông báo, lớn hơn bảy lần so với bộ dữ liệu được sử dụng cho Llama 2.

Bộ dữ liệu khổng lồ này bao gồm dữ liệu từ hơn 30 ngôn ngữ, cho phép mô hình xử lý nhiều phong cách và bối cảnh ngôn ngữ khác nhau.

- Phát triển có trách nhiệm: Llama 3 kết hợp phương pháp tiếp cận cấp hệ thống của Meta vào quá trình phát triển AI có trách nhiệm, bao gồm các công cụ tin cậy và an toàn được cập nhật như Llama Guard 2 và Code Shield. Mục tiêu là cho phép các nhà phát triển tùy chỉnh Llama 3 một cách an toàn cho các trường hợp sử dụng của họ trong khi áp dụng các biện pháp thực hành tốt nhất.

→ Những tính năng cải tiến này giúp chương trình này khác biệt so với các chương trình LLM khác trên thị trường và trở thành một mô hình có khả năng cao và linh hoạt cho nhiều ứng dụng xử lý ngôn ngữ tự nhiên. ***[19]***

IV. Kỹ thuật tinh chỉnh tham số hiệu quả (Parameter-Efficient Fine-Tuning)

1. Parameter-Efficient Fine-Tuning (PEFT)

- PEFT là một kỹ thuật tinh chỉnh mô hình nhằm giảm thiểu số lượng tham số cần cập nhật trong quá trình tinh chỉnh, qua đó giúp tiết kiệm tài nguyên tính toán và giảm chi phí lưu trữ. Thay vì điều chỉnh tất cả các tham số của mô hình, PEFT chỉ cập nhật một tập hợp nhỏ các tham số, hoặc thêm các tham số phụ vào mô hình để tối ưu hóa. ***[20]***
- Mô hình PEFT là mô hình được đào tạo trước đã được tinh chỉnh bằng kỹ thuật tinh chỉnh hiệu quả tham số. Mô hình PEFT bắt đầu như một mô hình mục đích chung được đào tạo trên lượng lớn dữ liệu để học cách hiểu rộng về ngôn ngữ hoặc các mẫu hình ảnh. Sau đó, quá trình tinh chỉnh sẽ điều chỉnh mô hình này để thực hiện tốt các tác vụ cụ thể hơn bằng cách chỉ sửa đổi một số ít tham số được chọn thay vì toàn bộ mạng. Việc cập nhật có chọn lọc này khiến các mô hình PEFT đặc biệt hữu ích cho các ứng dụng mà việc triển khai các mô hình quy mô lớn bị cấm về mặt tính toán hoặc tài chính. Bằng cách tập trung cập nhật vào các tham số có tác động lớn nhất, mô hình PEFT duy trì hiệu suất cao trong khi vẫn hiệu quả và nhanh nhẹn hơn các mô hình được đào tạo lại hoàn toàn. ***[21]***
- PEFT quan trọng vì:
  - Giảm tài nguyên tính toán cần thiết: Nó chỉ điều chỉnh các tham số liên quan nhất thay vì tất cả các tham số, tiết kiệm đáng kể năng lượng, giảm lượng khí thải carbon và chi phí điện toán đám mây liên quan đến việc huấn luyện AI.
  - Thời gian tạo giá trị nhanh hơn: Nó giúp thích ứng với các mô hình tiên tiến như GPT-3 một cách nhanh chóng cho các trường hợp sử dụng mới, nơi việc huấn luyện lại hoàn toàn sẽ mất quá nhiều thời gian và tài nguyên.
  - Ngăn chặn sự quên thảm hại: Nó giữ lại những khả năng được mã hóa trong các mô hình đã được huấn luyện trước. Kiến thức rộng lớn học được trong quá trình huấn luyện trước đó được duy trì.
  - Tiếp cận sức mạnh của các mô hình lớn: Nó giúp các công ty và nhóm nhỏ hơn với nguồn lực hoặc dữ liệu hạn chế tiếp cận sức mạnh của các mô hình lớn bằng cách tránh nhu cầu huấn luyện lại rộng rãi.
  - Đơn giản hóa quy trình làm việc AI: Nó giúp đơn giản hóa và tinh gọn quy trình làm việc AI bằng cách làm cho việc học chuyển giao và thích ứng các mô hình đã được huấn luyện trước trở nên dễ dàng hơn và nhẹ nhàng hơn.
  - Giảm rào cản tùy chỉnh: Nó cho phép các nhóm AI khám phá hiệu quả việc áp dụng các mô hình cho các lĩnh vực và trường hợp sử dụng mới. [22]
- ***Lợi ích của PEFT***
  - Giảm chi phí tính toán và lưu trữ: PEFT liên quan đến việc tinh chỉnh chỉ một số lượng nhỏ các tham số mô hình bổ sung trong khi giữ nguyên hầu hết các tham số

của các LLM đã được huấn luyện trước, do đó giảm đáng kể chi phí tính toán và lưu trữ.

- Vượt qua hiện tượng quên thảm hại: Trong quá trình tinh chỉnh hoàn toàn các LLM, hiện tượng quên thảm hại có thể xảy ra, nơi mà mô hình quên đi kiến thức đã học trong quá trình huấn luyện trước. PEFT có khả năng vượt qua vấn đề này bằng cách chỉ cập nhật một vài tham số.
- Hiệu suất tốt hơn trong các chế độ có dữ liệu thấp: PEFT đặc biệt có lợi trong các tình huống có dữ liệu thưa thớt. PEFT có thể đạt được hiệu suất vượt trội và tổng quát tốt hơn với các miền mới, chưa thấy so với các phương pháp tinh chỉnh truyền thống bằng cách tối ưu hóa một tập con tham số nhỏ hơn.
- Tính di động: Các phương pháp PEFT cho phép người dùng tạo ra các checkpoint nhỏ chỉ vài MB so với các checkpoint lớn của việc tinh chỉnh hoàn toàn. Điều này khiến cho trọng số đã được huấn luyện từ các phương pháp PEFT dễ triển khai và sử dụng cho nhiều nhiệm vụ mà không cần thay thế toàn bộ mô hình. Các mô hình được tinh chỉnh qua PEFT nhỏ hơn và dễ quản lý hơn, dẫn đến các mô hình nhẹ hơn, dễ triển khai trên nhiều nền tảng, bao gồm cả thiết bị di động và các thiết bị hạn chế tài nguyên khác.
- Hiệu suất tương đương với số lượng tham số có thể huấn luyện ít hơn: Mặc dù tinh chỉnh ít tham số hơn, PEFT vẫn có thể đạt được mức hiệu suất tương đương với tinh chỉnh mô hình hoàn toàn. Sự hiệu quả này cho phép mở rộng cho nhiều ứng dụng mà không cần nhiều tài nguyên tính toán.
- Tính bền vững: PEFT đại diện cho một phương pháp bền vững hơn trong việc huấn luyện mô hình, yêu cầu ít năng lượng tính toán và thời gian hơn. Nó giảm lượng khí thải carbon liên quan đến các tác vụ tính toán lớn, phù hợp với các mục tiêu hoạt động thân thiện với môi trường.
- Chu kỳ huấn luyện nhanh hơn: Việc PEFT tập trung vào số lượng tham số ít hơn giúp tăng tốc quá trình huấn luyện, cho phép có các vòng lặp và triển khai nhanh hơn. Điều này rất lý tưởng cho các dự án có lịch phát triển gấp gáp.
- Yêu cầu lưu trữ thấp hơn: Kích thước mô hình kết quả nhỏ hơn với hầu hết các tham số của mô hình gốc không thay đổi. Sự giảm kích thước này tạo điều kiện dễ dàng hơn trong việc quản lý và phân phối các bản cập nhật mô hình, làm cho việc cải tiến liên tục trở nên thực tế hơn trong các môi trường hoạt động. [21]

2. So sánh Parameter-Efficient Fine-Tuning (PEFT) với Standard Fine-Tuning (SFT)

- Standard Fine-Tuning: Là phương pháp tinh chỉnh truyền thống, trong đó tất cả các tham số của mô hình đều được cập nhật. Điều này giúp mô hình thích nghi tốt với tác vụ cụ thể, nhưng đồng thời cũng đòi hỏi nhiều tài nguyên tính toán và thời gian huấn luyện.
- Tinh chỉnh chuẩn được coi là không hiệu quả bằng các phương pháp chuyên biệt để chỉnh sửa mô hình do hiệu suất tương đối kém. Tuy nhiên, nó đơn giản, không phụ thuộc vào các chi tiết kiến trúc của mô hình đang được chỉnh sửa và có thể tận dụng những tiến bộ trong các kỹ thuật đào tạo chuẩn mà không cần thêm công việc (ví dụ: PEFT hộp đen để tăng hiệu quả tính toán), khiến nó trở thành lựa chọn hấp dẫn cho trình chỉnh sửa mô hình. ***[23]***
- Ưu và nhược điểm:
    - Tinh chỉnh chuẩn có thể đạt được hiệu suất cao với việc triển khai tương đối đơn giản. Tuy nhiên, có nguy cơ quá khớp, đặc biệt nếu dữ liệu cụ thể theo miền bị hạn chế hoặc không đại diện cho nhiệm vụ rộng hơn.

- Tinh chỉnh chuẩn yêu cầu có một tập dữ liệu đủ lớn và có liên quan. Điều chỉnh siêu tham số, chẳng hạn như điều chỉnh tốc độ học và kích thước lô, là rất quan trọng để đảm bảo đào tạo mô hình hiệu quả. ***[24]***

- Bảng so sánh PEFT và SFT: ***[21]***

Tiêu chí	Parameter-Efficient Fine-Tuning (PEFT)	Standard Fine-Tuning (SFT)
Mục đích	Cải thiện hiệu suất của mô hình được huấn luyện trước trên một tác vụ cụ thể với dữ liệu và tính toán hạn chế	Cải thiện hiệu suất của mô hình được huấn luyện trước trên một tác vụ cụ thể với dữ liệu và tính toán phong phú
Dữ liệu huấn luyện	Tập dữ liệu nhỏ	Tập dữ liệu lớn
Thời gian huấn luyện	Nhanh hơn	Tốn nhiều thời gian
Tài nguyên tính toán	Sử dụng ít tài nguyên hơn	Yêu cầu tài nguyên lớn hơn
Tham số mô hình	Chỉ sửa đổi một tập hợp nhỏ các tham số mô hình	Huấn luyện lại toàn bộ mô hình
Quá khớp	Ít có khả năng bị overfitting hơn vì mô hình không bị sửa đổi quá mức	Dễ bị overfitting khi mô hình được sửa đổi nhiều
Hiệu suất huấn luyện	Kém hơn tinh chỉnh chuẩn nhưng vẫn đủ tốt	Thông thường mang lại hiệu suất tốt hơn
Trường hợp sử dụng	Lý tưởng cho môi trường có tài nguyên thấp hoặc không có lượng lớn dữ liệu huấn luyện	Lý tưởng cho các môi trường có nhiều tài nguyên với dữ liệu huấn luyện phong phú và tài nguyên tính toán dồi dào

- Tác động đối với phân tích cảm xúc: Trong bài toán phân tích cảm xúc, PEFT có thể mang lại hiệu suất tương đương hoặc tốt hơn standard fine-tuning, vì các đặc trưng cảm xúc thường có thể được học thông qua một số ít các tham số được cập nhật. Việc tiết kiệm tài nguyên này rất hữu ích khi cần tinh chỉnh mô hình trên nhiều tập dữ liệu khác nhau, như **go_emotions** và **financial_phrasebank**.

3. LoRA (Low-Rank Adaptation)

- LoRA (Low-Rank Adaptation of Large Language Models) là một kỹ thuật huấn luyện phổ biến, tối ưu về mặt tài nguyên, giúp giảm thiểu đáng kể số lượng tham số cần được huấn luyện. Phương pháp này thực hiện bằng cách chèn một số lượng nhỏ ma trận trọng số mới vào mô hình và chỉ những trọng số này được điều chỉnh trong quá trình huấn luyện. Kỹ thuật này mang lại hiệu quả cao về tốc độ huấn luyện, tối ưu hóa bộ nhớ và tạo ra các trọng số mô hình nhỏ gọn (chỉ vài trăm MB), dễ dàng lưu trữ và phân phối. LoRA cũng có thể được kết hợp với các kỹ thuật huấn luyện khác, chẳng hạn như DreamBooth, để tăng tốc độ huấn luyện. ***[25]***
- Cách hoạt động:
  - Thêm vào các ma trận hạng thấp để biểu diễn các thay đổi trong tham số của mô hình gốc.
  - Cập nhật các ma trận hạng thấp này trong quá trình huấn luyện, trong khi giữ nguyên các tham số gốc của mô hình.
- Lợi ích:

- Hiệu quả trong huấn luyện và triển khai: LoRA giảm gánh nặng tính toán, cho phép điều chỉnh mô hình nhanh hơn. Bằng cách yêu cầu ít tham số cần huấn luyện hơn, LoRA giúp dễ dàng tinh chỉnh các mô hình lớn trên phần cứng yếu hơn.
- Duy trì chất lượng và tốc độ mô hình: Mặc dù giảm số lượng tham số, LoRA vẫn giữ nguyên chất lượng và tốc độ suy luận của mô hình ban đầu.
- Giảm kích thước checkpoint: LoRA giảm đáng kể kích thước các checkpoint của mô hình. Ví dụ, với GPT-3, kích thước checkpoint đã giảm từ 1 TB xuống còn 25 MB.
- Không có độ trễ suy luận: LoRA không gây ra độ trễ bổ sung trong quá trình suy luận. Các ma trận dạng hạng thấp được sử dụng trong quá trình huấn luyện, nhưng sẽ được gộp chung với các tham số ban đầu trong quá trình suy luận, đảm bảo không có sự chậm trễ. Điều này cho phép chuyển đổi mô hình nhanh chóng trong thời gian thực mà không gây thêm độ trễ suy luận.
- Tính linh hoạt: LoRA có thể áp dụng cho bất kỳ mô hình nào sử dụng phép nhân ma trận (chẳng hạn như Support Vector Machine), khiến nó trở thành một kỹ thuật được áp dụng rộng rãi trong nhiều trường hợp khác. Thực tế, LoRA được sử dụng phổ biến trong các mô hình Stable Diffusion để truyền tải phong cách trong các mô hình hình ảnh lớn. Ứng dụng trong phân tích cảm xúc: LoRA cho phép tinh chỉnh hiệu quả các mô hình ngôn ngữ lớn để nhận diện cảm xúc từ văn bản mà vẫn duy trì hiệu suất cao, đặc biệt hữu ích khi xử lý các tập dữ liệu lớn hoặc đa dạng. ***[26]***

V. Tinh chỉnh mô hình ngôn ngữ lớn cho bài toán phân tích cảm xúc văn bản (tiếng Anh)

1. Bộ dữ liệu được sử dụng

- Để mô hình được chính xác thì việc lựa chọn dữ liệu vô cùng quan trọng.
- Trong lĩnh vực tài chính và kinh tế, dữ liệu được chú thích (annotated datasets) rất hiếm, phần lớn là tài sản độc quyền.
- Lựa chọn bộ dữ liệu **FinancialPhraseBank:**
    - Để giải quyết vấn đề thiếu dữ liệu đào tạo, các nhà nghiên cứu từ Trường Kinh doanh Đại học Aalto đã giới thiệu vào năm 2014 một bộ dữ liệu với khoảng 5000 câu.
    - Bộ dữ liệu này được chú thích bởi 16 người có kiến thức chuyên môn về thị trường tài chính. Họ được hướng dẫn đánh giá các câu theo quan điểm của nhà đầu tư, dựa trên việc thông tin đó có tác động tích cực, tiêu cực hay trung lập đến giá cổ phiếu.
    - FinancialPhraseBank là một bộ sưu tập bao gồm các tiêu đề tin tức tài chính, được phân loại theo cảm xúc tiêu cực, trung lập hoặc tích cực từ góc nhìn của nhà đầu tư bán lẻ.
- Ứng dụng của bộ dữ liệu:
    - Bộ dữ liệu FinancialPhraseBank này đã được sử dụng trong nhiều nghiên cứu để phân tích và hiểu rõ hơn về cảm xúc trong tin tức tài chính.
    - Nghiên cứu ban đầu của bộ dữ liệu này được công bố bởi Malo, P. và các cộng sự trong bài báo "Good debt or bad debt: Detecting semantic orientations in economic texts" vào năm 2014.

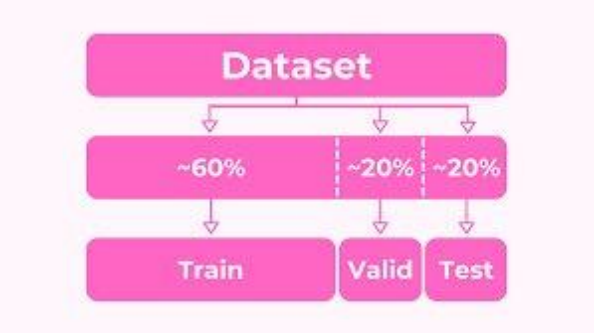
2. Các thư viện được sử dụng

- **accelerate**: Đây là một thư viện hỗ trợ đào tạo phân tán của PyTorch, phát triển bởi HuggingFace. Thư viện này giúp huấn luyện mô hình trên nhiều GPU hoặc CPU cùng lúc (cấu hình phân tán), từ đó tăng tốc quá trình đào tạo khi sử dụng nhiều GPU. Tuy nhiên, trong ví dụ này, accelerate sẽ không được sử dụng.
- **peft**: Đây là một thư viện Python của HuggingFace được thiết kế để điều chỉnh hiệu quả các mô hình ngôn ngữ đã được huấn luyện (PLMs) cho các ứng dụng sau khi đã huấn luyện

mà không cần tinh chỉnh tất cả tham số của mô hình. PEFT chỉ tinh chỉnh một số lượng nhỏ tham số bổ sung, do đó giảm đáng kể chi phí tính toán và lưu trữ.
- **bitsandbytes**: Thư viện này được tạo bởi Tim Dettmers, là một lớp bao bọc nhẹ xung quanh các hàm tùy chỉnh CUDA, đặc biệt là các tối ưu hóa 8-bit, nhân ma trận (LLM.int8()) và các hàm lượng tử hóa. Nó cho phép chạy các mô hình được lưu trữ ở độ chính xác 4-bit, mặc dù trọng số được lưu trữ ở 4-bit, việc tính toán vẫn diễn ra ở độ chính xác 16-bit hoặc 32-bit (float16, bfloat16, float32, v.v.).
- **transformers**: Đây là một thư viện Python cho xử lý ngôn ngữ tự nhiên (NLP). Nó cung cấp một số mô hình đã được huấn luyện sẵn cho các tác vụ NLP như phân loại văn bản, trả lời câu hỏi và dịch thuật.
- trl: Đây là một thư viện đầy đủ các công cụ do HuggingFace phát triển để huấn luyện các mô hình ngôn ngữ transformer bằng phương pháp học tăng cường. Thư viện cung cấp các bước từ Huấn luyện Giám sát (SFT), Mô hình hóa Phần thưởng (RM) cho đến Tối ưu hóa Chính sách Tiệm cận (PPO).

3. Các bước thực hiện

- Tạo dữ liệu:
    - Đọc và chia dữ liệu thành 3 phần: 60% cho train, 20% cho test và 20% cho eval.
    - Chuyển đổi văn bản thành gợi ý cho các mô hình ngôn ngữ lớn. Trong bộ huấn luyện câu trả lời mong đợi (cảm xúc) được bao gồm để mô hình học cách dự đoán.
    - Sử dụng lớp của Hugging Face bọc dữ liệu để tiện sử dụng cho quá trình tinh chỉnh.



Mục	Số lượng
Tổng số bản ghi trong df	4846
Tổng số bản ghi trong X_train	2909
Tổng số bản ghi trong X_eval	967
Tổng số bản ghi trong X_test	970
Tổng số bản ghi trong y_true	970

Tập huấn luyện:	
Loại nhãn	**Số lượng**
Neutral	1728
Positive	818
Negative	363

Tập đánh giá:	
Loại nhãn	**Số lượng**

Neutral	575
Positive	272
Negative	120

Tập kiểm tra:	
Loại nhãn	**Số lượng**
Neutral	576
Positive	273
Negative	121

- Tạo hàm đánh giá mô hình
    - Chuyển đổi nhãn cảm xúc thành dạng số (2: tích cực, 1: trung tính, 0: tiêu cực).
    - Tính toán độ chính xác của mô hình trên bộ dữ liệu kiểm tra: Tính toán tỉ lệ chính xác (accuracy).
    - Tạo báo cáo độ chính xác cho từng nhãn cảm xúc (accuracy cho từng nhãn cảm xúc).
    - Tạo báo cáo phân loại (Classification report) gồm các chỉ số quan trọng cho mô hinh, chẳng hạn như precision (độ chính xác), recall (tỉ lệ phát hiện) và F1-score cho từng nhãn cảm xúc.
    - Tạo ma trận nhầm lẫn (confusion matrix) là một ma trận hiển thị số lượng dự đoán.
- Tải và lượng tử hóa mô hình
    - Tải mô hình ngôn ngữ.
    - Cấu hình cho tối ưu hóa bits and bytes:
        - Tải trọng số dưới dạng 4 bits.
        - Sử dụng lượng tử hóa nf4 (4-bit NormalFloat) .
        - Sử dụng kiểu dữ liệu float16.
        - Không sử dụng lượng tử hóa kép.
    - Tạo đối tượng AutoModelForCausalLM từ mô hình, sử dụng BitsAndBytesConfig để lượng tử hóa mô hình.
    - Vô hiệu hóa tình năng caching cho mô hình.
    - Đặt xác suất token tiền huấn luyện về 1.

→ Tóm lại hàm này được sử dụng để đánh giá toàn diện về hiệu suất của mô hình phân tích cảm xúc sau khi đã tinh chỉnh, giúp phân tích độ chính xác, hiểu rõ hơn về khả năng phân loại của mô hình và các lỗi dự đoán.

- Tải và thiết lập Tokenizer
    - Tải Tokenizer cho mô hình.
    - Thiết lập token lấp đầy là toke EOS (End-of-Sequence) – thường được sử dụng để đánh dấu sự kết thúc của chuỗi văn bản, và trong trường hợp này, nó cũng được dùng để lấp đầy các chuỗi ngắn hơn nhằm chuẩn bị cho đầu vào mô hình.
    - Thiết lập hướng lấp đầy (padding side) là "phải": Các chuỗi ngắn hơn sẽ được lấp đầy ở phía bên phải. Nó cần được làm chính xác để mô hình xử lý đầu vào đúng cách.

→ Toàn bộ quá trình nhằm chuẩn bị mô hình và tokenizer cho quá trình kiểm thử mà không tinh chỉnh (fine-tuning), giúp đánh giá hiệu suất mô hình trong trạng thái đã được huấn luyện trước, nhưng chưa được điều chỉnh cho tác vụ cụ thể là phân tích cảm xúc.

- Thiết lập hàm dự đoán cảm xúc

- Các đối số:
    - test: Tập tiêu đề tin tức cần dự đoán.
    - model: Mô hình ngôn ngữ được huấn luyện.
    - tokenizer: Bộ tokenizer được sử dụng.
- Hoạt động:
    - Tạo prompt yêu cầu nó phân tích cảm xúc của tiêu đề và trả về nhãn cảm xúc tương ứng.
    - Áp dụng pipeline function để tạo văn bản từ mô hình ngôn ngữ dựa trên prompt.
    - Trích xuất nhãn dự đoán.
    - Trả về danh sách chứa các nhãn dự đoán tương ứng với tập dữ liệu test.

- Tinh chỉnh mô hình
    - Cấu hình PEFT:
        - lora_alpha: Tốc độ học cho các ma trận cập nhật của LoRA.
        - lora_dropout: Xác suất dropout cho các ma trận cập nhật của LoRA.
        - r: Hạng của các ma trận cập nhật LoRA.
        - bias: Loại bias được sử dụng. Các giá trị có thể là none, additive, và learned.
        - task_type: Loại nhiệm vụ mà mô hình đang được huấn luyện, các giá trị có thể là CAUSAL_LM (mô hình ngôn ngữ nhân quả) và MASKED_LM (mô hình ngôn ngữ mặt nạ).
    - Cấu hình TrainingArguments:
        - output_dir: Thư mục để lưu nhật ký huấn luyện và các điểm kiểm tra.
        - num_train_epochs: Số lượng epoch để huấn luyện mô hình.
        - per_device_train_batch_size: Số lượng mẫu trong mỗi batch trên mỗi thiết bị.
        - gradient_accumulation_steps: Số batch cần tích lũy gradient trước khi cập nhật tham số của mô hình.
        - optim: Optimizer để sử dụng khi huấn luyện mô hình.
        - save_steps: Số bước sau đó sẽ lưu một điểm kiểm tra.
        - logging_steps: Số bước sau đó sẽ ghi nhật ký các số liệu huấn luyện.
        - learning_rate: Tốc độ học cho optimizer.
        - weight_decay: Tham số weight decay cho optimizer.
    - Cấu hình SFTTrainer:
        - model: Mô hình cần huấn luyện.
        - train_dataset: Bộ dữ liệu huấn luyện.
        - eval_dataset: Bộ dữ liệu đánh giá.
        - peft_config: Cấu hình PEFT.
        - dataset_text_field: Tên của trường văn bản trong bộ dữ liệu.
        - tokenizer: Tokenizer được sử dụng.
        - args: Các tham số huấn luyện.
        - packing: Có gói các mẫu huấn luyện hay không.
        - max_seq_length: Độ dài chuỗi tối đa.
- Huấn luyện và kiểm tra mô hình
    - Huấn luyện mô hình từ phương thức train() của SFTTrainer.

- Dự đoán các nhãn cảm xúc trên tập test và đánh giá hiệu năng của mô hình trên tập evaluate.

4. *So sánh kết quả*

4.1. *Sử dụng BERT bản 110 M*

- Trước Fine-tune

```
Accuracy: 0.594
Accuracy for label 0: 0.000
Accuracy for label 1: 1.000
Accuracy for label 2: 0.000

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       121
           1       0.59      1.00      0.75       576
           2       0.00      0.00      0.00       273

    accuracy                           0.59       970
   macro avg       0.20      0.33      0.25       970
weighted avg       0.35      0.59      0.44       970


Confusion Matrix:
[[  0 121   0]
 [  0 576   0]
 [  0 273   0]]
```

- Sau khi Fine-tune

```
Accuracy: 0.680
Accuracy for label 0: 0.050
Accuracy for label 1: 0.903
Accuracy for label 2: 0.491

Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.05      0.09       121
           1       0.73      0.90      0.81       576
           2       0.54      0.49      0.52       273

    accuracy                           0.68       970
   macro avg       0.65      0.48      0.47       970
weighted avg       0.67      0.68      0.64       970


Confusion Matrix:
[[  6  58  57]
 [  1 520  55]
 [  2 137 134]]
```

4.2. *Sử dụng Roberta bản 125 M*

- Trước khi Fine-tune

```
Accuracy: 0.594
Accuracy for label 0: 0.000
Accuracy for label 1: 1.000
Accuracy for label 2: 0.000

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       121
           1       0.59      1.00      0.75       576
           2       0.00      0.00      0.00       273

    accuracy                           0.59       970
   macro avg       0.20      0.33      0.25       970
weighted avg       0.35      0.59      0.44       970


Confusion Matrix:
[[  0 121   0]
 [  0 576   0]
 [  0 273   0]]
```

- Sau khi Fine-tune

```
Accuracy: 0.745
Accuracy for label 0: 0.579
Accuracy for label 1: 0.962
Accuracy for label 2: 0.363

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.58      0.68       121
           1       0.72      0.96      0.83       576
           2       0.83      0.36      0.51       273

    accuracy                           0.75       970
   macro avg       0.79      0.63      0.67       970
weighted avg       0.77      0.75      0.72       970


Confusion Matrix:
[[ 70  47   4]
 [  6 554  16]
 [  9 165  99]]
```

4.3. Sử dụng Phi-3 bản 8B

- Trước khi Fine-tune

```
Accuracy: 0.708
Accuracy for label 0: 0.967
Accuracy for label 1: 0.582
Accuracy for label 2: 0.861

Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.97      0.83       121
           1       0.92      0.58      0.71       576
           2       0.53      0.86      0.65       273

    accuracy                           0.71       970
   macro avg       0.73      0.80      0.73       970
weighted avg       0.79      0.71      0.71       970


Confusion Matrix:
[[117   3   1]
 [ 32 335 209]
 [ 12  26 235]]
```

- Sau khi Fine-tune

```
100%|██████████| 970/970 [02:29<00:00,  6.50it/s]Accuracy: 0.873
Accuracy for label 0: 0.835
Accuracy for label 1: 0.917
Accuracy for label 2: 0.799

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.83      0.87       121
           1       0.88      0.92      0.90       576
           2       0.84      0.80      0.82       273

    accuracy                           0.87       970
   macro avg       0.88      0.85      0.86       970
weighted avg       0.87      0.87      0.87       970


Confusion Matrix:
[[101  19   1]
 [  9 528  39]
 [  1  54 218]]
```

4.4. Sử dụng Gemma 2 bản 2B

- Trước khi Fine-tune

```
Accuracy: 0.599
Accuracy for label 0: 0.000
Accuracy for label 1: 0.997
Accuracy for label 2: 0.026

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       121
           1       0.60      1.00      0.75       576
           2       0.58      0.03      0.05       273

    accuracy                           0.60       970
   macro avg       0.39      0.34      0.27       970
weighted avg       0.52      0.60      0.46       970


Confusion Matrix:
[[  0 118   3]
 [  0 574   2]
 [  0 266   7]]
```

- Sau khi Fine-tune

```
100%|██████████| 970/970 [01:02<00:00, 15.40it/s]Accuracy: 0.868
Accuracy for label 0: 0.876
Accuracy for label 1: 0.899
Accuracy for label 2: 0.799

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.88      0.88       121
           1       0.89      0.90      0.89       576
           2       0.82      0.80      0.81       273

    accuracy                           0.87       970
   macro avg       0.86      0.86      0.86       970
weighted avg       0.87      0.87      0.87       970


Confusion Matrix:
[[106  13   2]
 [ 11 518  47]
 [  2  53 218]]
```

4.5. Sử dụng Gemma 1 bản 8B

- Trước khi Fine-tune

```
Accuracy: 0.594
Accuracy for label 0: 0.000
Accuracy for label 1: 1.000
Accuracy for label 2: 0.000

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       121
           1       0.59      1.00      0.75       576
           2       0.00      0.00      0.00       273

    accuracy                           0.59       970
   macro avg       0.20      0.33      0.25       970
weighted avg       0.35      0.59      0.44       970


Confusion Matrix:
[[  0 121   0]
 [  0 576   0]
 [  0 273   0]]
```

- Sau khi Fine-tune

```
100%|██████████| 970/970 [26:27<00:00,  1.64s/it]Accuracy: 0.869
Accuracy for label 0: 0.860
Accuracy for label 1: 0.911
Accuracy for label 2: 0.784

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.86      0.89       121
           1       0.88      0.91      0.89       576
           2       0.84      0.78      0.81       273

    accuracy                           0.87       970
   macro avg       0.87      0.85      0.86       970
weighted avg       0.87      0.87      0.87       970


Confusion Matrix:
[[104  16   1]
 [ 10 525  41]
 [  0  59 214]]
```

4.6. Sử dụng Llama 3 bản 7B

- Trước khi Fine-tune

```
Accuracy: 0.586
Accuracy for label 0: 0.000
Accuracy for label 1: 0.977
Accuracy for label 2: 0.018

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       121
           1       0.59      0.98      0.74       576
           2       0.28      0.02      0.03       273

    accuracy                           0.59       970
   macro avg       0.29      0.33      0.26       970
weighted avg       0.43      0.59      0.45       970


Confusion Matrix:
[[  0 121   0]
 [  0 563  13]
 [  0 268   5]]
```

- Sau khi Fine-tune

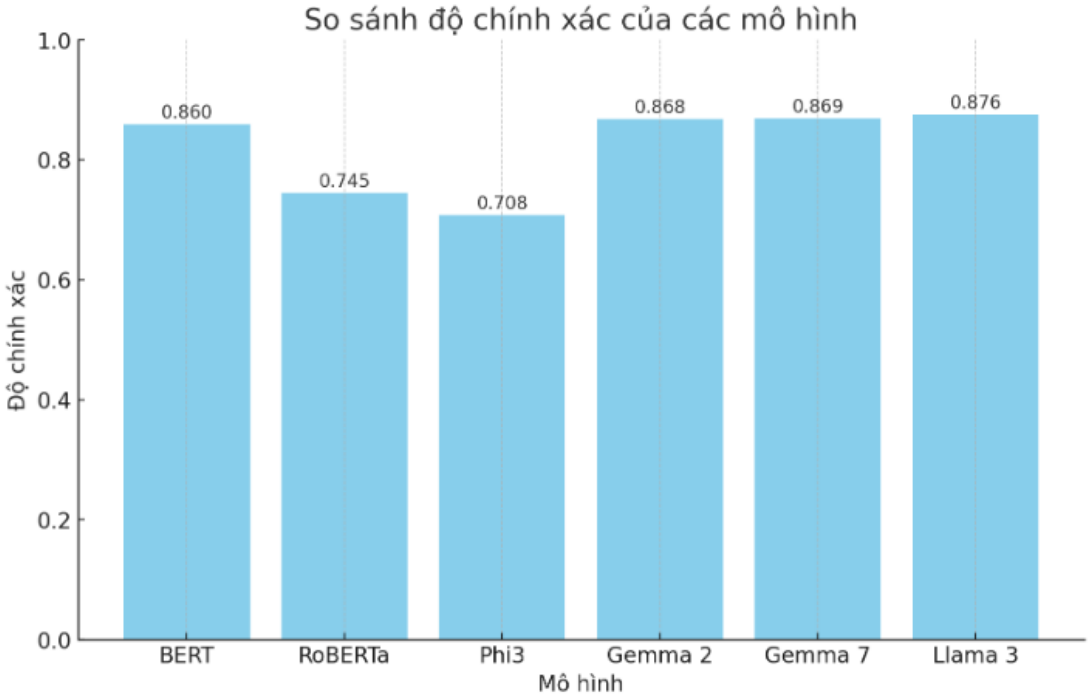
```
100%|██████████| 970/970 [01:58<00:00,  8.16it/s]Accuracy: 0.876
Accuracy for label 0: 0.860
Accuracy for label 1: 0.922
Accuracy for label 2: 0.788

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.86      0.89       121
           1       0.88      0.92      0.90       576
           2       0.85      0.79      0.82       273

    accuracy                           0.88       970
   macro avg       0.88      0.86      0.87       970
weighted avg       0.88      0.88      0.88       970


Confusion Matrix:
[[104  15   2]
 [ 10 531  35]
 [  0  58 215]]
```

4.7. Đánh giá

- Hiệu suất của mô hình sau khi được fine-tune cải thiện đáng kể so với mô hình trước khi fine-tune:
    - Độ chính xác tổng thể (accuracy): Tăng cho thấy mô hình đã có khả năng dự đoán chính xác hơn đáng kể.
    - Precision, Recall, F1-score: Các chỉ số này đều đạt mức cao cho tất cả các nhãn, cho thấy mô hình có khả năng phân loại chính xác các mẫu thuộc từng lớp và ít bị nhầm lẫn giữa các lớp.

- Ma trận nhầm lẫn: Ma trận nhầm lẫn cho thấy mô hình đã giảm đáng kể số lượng mẫu bị phân loại sai, đặc biệt là đối với nhãn 0 và 2.

- Quá trình fine-tune đã mang lại hiệu quả rất tốt trong việc cải thiện hiệu suất của mô hình. Mô hình sau khi fine-tune đã có khả năng phân loại chính xác hơn các mẫu thuộc các lớp khác nhau, đặc biệt là đối với các lớp thiểu số.
- So sánh kết quả giữa các mô hình:

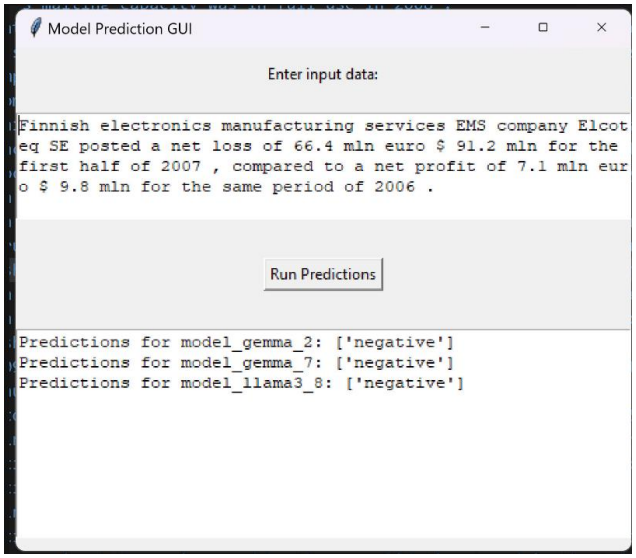


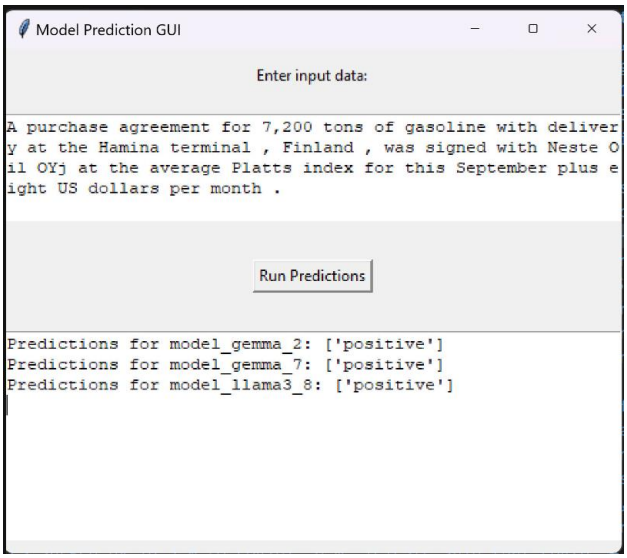
- BERT: Mô hình BERT có độ chính xác khá cao. Điều này cho thấy BERT hiệu quả trong việc hiểu ngữ nghĩa và ngữ cảnh của ngôn ngữ tự nhiên, thường được sử dụng trong nhiều tác vụ NLP.
- Gemma 2 bản 2B và Gemma 1 bản 7B: Hai mô hình này có độ chính xác gần giống nhau, cao hơn BERT với Gemma 1 bản 7B chỉ nhỉnh hơn một chút so với Gemma 2. Chúng đều thể hiện hiệu suất tốt.
- Llama 3 bản 8B : Mô hình Llama 3 đạt độ chính xác ấn tượng, cao hơn cả Gemma 1 bản 7B. Lý do là được đào tạo trên nhiều tham số hơn.
- RoBERTa: RoBERTa có độ chính xác thấp hơn đáng kể so với các mô hình còn lại. Mặc dù RoBERTa là phiên bản cải tiến của BERT, trong một số trường hợp cụ thể, có thể không đạt được hiệu suất tối ưu.
- Phi3: Mô hình này có độ chính xác thấp nhất trong nhóm. Điều này có thể do cấu trúc hoặc cách tiếp cận khác biệt so với các mô hình khác.

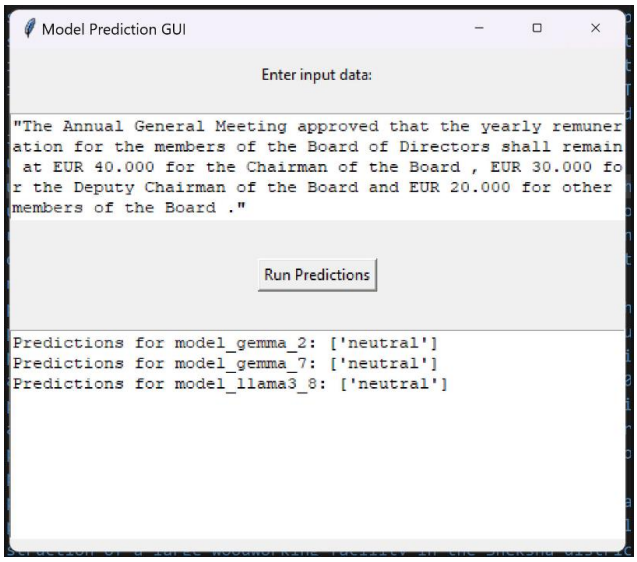
Việc lựa chọn mô hình ngôn ngữ lớn như Llama, 3gemma, Phi3 hay RoBERTa phụ thuộc vào nhiều yếu tố khác nhau. Trước hết, số lượng tham số của mô hình đóng vai trò quan trọng, khi các mô hình có nhiều tham số hơn thường có khả năng xử lý thông tin tốt hơn và cho kết quả chính xác hơn, nhưng đồng thời cũng đòi hỏi nhiều tài nguyên tính toán hơn. Bên cạnh đó, phiên bản của mô hình cũng cần được cân nhắc, vì các phiên bản mới thường có những cải tiến về hiệu suất và độ chính xác. Cuối cùng, yêu cầu về thời gian phản hồi cũng là một yếu tố quan trọng; nếu cần trả lời nhanh, lựa chọn các mô hình nhỏ hơn hoặc các phiên bản đã được tối ưu hóa sẽ phù hợp hơn. Do đó, việc chọn mô hình ngôn ngữ lớn cần được cân nhắc kỹ lưỡng, tùy thuộc vào yêu cầu cụ thể của bài toán và nguồn tài nguyên sẵn có.

4.8. Demo

Phần thực nghiệm demo thử nghiệm ba mô hình ngôn ngữ lớn, gồm **Gemma 2 bản 2B**, **Gemma 1 bản 8B**, và **Llama 3 bản 7B** trên GPU Nvidia RTX 4070 Super. Ba mô hình ngôn ngữ lớn được thử nghiệm, gồm **Gemma 2 bản 2B**, **Gemma 1 bản 8B**, và **Llama 3 bản 7B**, đều cho kết quả tương tự nhau qua nhiều test case, như được mô tả trong ba hình dưới đây:







VI. Tài liệu tham khảo

[1] *A. Bessa, "Lexicon-based sentiment analysis: What it is & how to conduct one," 11 12 2023. [Online]. Available: https://www.knime.com/blog/lexicon-based-sentiment-analysis#:~:text=Lexicon%2Dbased%20sentiment%20analysis%20is%20a%20technique%20used%20in%20natural,or%20neutral)%20and%20detect%20sentiment..*

[2] *T. T. a. CallMiner, "Sentiment analysis & machine learning: 2023 guide," 27 6 2023. [Online]. Available: https://callminer.com/blog/sentiment-analysis-and-machine-learning-2023-guide.*

[3] *R. S. Z. S. M. M. A. M. B. M. S. Talha Ahmed Khan, "Sentiment Analysis using Support Vector Machine and Random Forest,"* Journal of Informatics and Web Engineering, *vol. 3, no. 1, p. 9, February 2024.*

[4] *"Multinomial Naïve Bayes Classifier for Sentiment Analysis of Internet Movie Database,"* Vietnam Journal of Computer Science, *vol. 10, p. 14, 2023.*

[5] *S. P. Soudamini Hota, "KNN classifier based approach for multi-class sentiment analysis of twitter data,"* International Journal of Engineering & Technology, *vol. 7, p. 4, 2018.*

[6] *M. A. A. R. A. N. M. K. B. H. a. A. R. Qurat Tul Ain, "Sentiment Analysis Using Deep Learning Techniques: A Review,"* International Journal of Advanced Computer Science and Applications, *vol. 8, p. 10, 2017.*

[7] *Koushiki, "Sentiment Analysis with LSTM," 13 6 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/01/sentiment-analysis-with-lstm/.*

[8] *T. N. Anh, "Giới thiệu về Transformer - Công nghệ đằng sau ChatGPT và Bard," 22 4 2024. [Online]. Available: https://200lab.io/blog/transformer-cong-nghe-dang-sau-chatgpt-va-bard/.*

[9] *M.-W. C. K. L. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 24 5 2019. [Online]. Available: https://arxiv.org/abs/1810.04805.*

[10] *H. Face, "SiEBERT - English-Language Sentiment Classification," [Online]. Available: https://huggingface.co/siebert/sentiment-roberta-large-english.*

[11] *M. O. N. G. J. D. M. J. D. C. O. L. M. L. L. Z. V. S. Yinhan Liu, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 26 7 2019. [Online]. Available: https://arxiv.org/abs/1907.11692.*

[12] *L. D. J. C. T. W. Victor Sanh, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 1 3 2020. [Online]. Available: https://arxiv.org/abs/1910.01108.*

[13] *ChatLabs, "Google Gemma 2 - Overview and Access," 28 6 2024. [Online]. Available: https://writingmate.ai/blog/google-gemma-2-access-overview.*

[14] *A. Acharya, "Phi-3: Microsoft's Mini Language Model is Capable of Running on Your Phone," 25 4 2024. [Online]. Available: https://encord.com/blog/microsoft-phi-3-small-language-model/.*

[15] *V. inTECH, "Hướng dẫn cách xây dựng LLaMA 3 từ A đến Z bằng Python," 14 6 2024. [Online]. Available: https://intech.vietnamworks.com/article/huong-dan-cach-xay-dung-llama-3-tu-a-den-z-bang-python#:~:text=LLaMA%203%20l%C3%A0%20m%E1%BB%99t%20m%C3%B4,dung%20s%C3%A1ng%20t%E1%BA%A1o%20kh%C3%A1c%20nhau..*

[16] B. Council, "Introduction to LLAMA 3," 11 8 2024. [Online]. Available: https://www.linkedin.com/pulse/introduction-llama-3-blockchaincouncil-1le4c.

[17] A. Vina, "Làm quen với Llama 3 của Meta," 10 5 2024. [Online]. Available: https://www.ultralytics.com/vi/blog/getting-to-know-metas-llama-3.

[18] A. Procter, "Introduction of Llama 3 models by Meta," 26 4 2024. [Online]. Available: https://www.okoone.com/spark/technology-innovation/introduction-of-llama-3-models-by-meta/.

[19] M. Anees, "Complete Breakdown of Llama 3: Features, Applications, and Comparison," 27 6 2024. [Online]. Available: https://workhub.ai/complete-breakdown-of-llama-3/.

[20] Z. Y. yfeng95, "PEFT," 12 4 2024. [Online]. Available: https://github.com/huggingface/peft/blob/main/README.md.

[21] A. Takyar, "Parameter-efficient Fine-tuning (PEFT): Overview, benefits, techniques and model training," [Online]. Available: https://www.leewayhertz.com/parameter-efficient-fine-tuning/.

[22] Moveworks, "What is parameter-efficient fine-tuning?," [Online]. Available: https://www.moveworks.com/us/en/resources/ai-terms-glossary/parameter-efficient-fine-tuning.

[23] K. S. Govind Gangadhar, "Model Editing by Standard Fine-Tuning," 3 6 2024. [Online]. Available: https://arxiv.org/abs/2402.11078.

[24] Y. Kniazieva, "LLM Fine Tuning Methods: Standard & Enhanced," 11 7 2024. [Online]. Available: https://labelyourdata.com/articles/llm-fine-tuning/llm-fine-tuning-methods.

[25] "LoRA," [Online]. Available: https://huggingface.co/docs/diffusers/main/training/lora.

[26] M. Ali, "Mastering Low-Rank Adaptation (LoRA): Enhancing Large Language Models for Efficient Adaptation," 16 1 2024. [Online]. Available: https://www.datacamp.com/tutorial/mastering-low-rank-adaptation-lora-enhancing-large-language-models-for-efficient-adaptation.