**I. Quy Trình Thực Hiện**

Quy trình này gồm **5 bước chính**, từ thu thập dữ liệu đến triển khai ứng dụng.

**1. Thu thập dữ liệu**

📌 **Nguồn dữ liệu**

* **Nền tảng thương mại điện tử**: Shopee, Tiki, Lazada
* **Website đánh giá dịch vụ**: Google Reviews

📌 **Cách thu thập dữ liệu**

* **Web scraping**: Dùng **BeautifulSoup, Scrapy, Selenium** để lấy bình luận
* **Thu thập tay**: với trường hợp web không cho sử dụng tool cào dữ liệu

📌 **Lưu ý**  
✔ Loại bỏ dữ liệu rác (spam, quảng cáo)  
✔ Cân bằng số lượng bình luận tích cực, tiêu cực, trung tính

**2. Tiền xử lý dữ liệu**

📌 **Bước 1: Làm sạch dữ liệu**  
✔ Loại bỏ stop words, dấu câu, số, URL  
✔ Chuyển đổi chữ viết hoa thành chữ thường  
✔ Xóa emoji hoặc chuyển emoji thành text

📌 **Bước 2: Biến đổi dữ liệu**  
✔ **Tokenization**: Chia câu thành các từ nhỏ  
✔ **Padding & Truncation**: Đảm bảo độ dài câu cố định  
✔ **Word Embedding**: Dùng BERT tokenizer để mã hóa văn bản thành vector số

📌 **Bước 3: Gán nhãn dữ liệu**  
✔ **Dữ liệu có nhãn**: Bình luận có sẵn nhãn cảm xúc (positive, neutral, negative)

**3. Huấn luyện mô hình AI**

📌 **Mô hình sử dụng**  
**- BERT-base**

**- BERT-LSTM**

**- BERT-RCNN**

📌 **So sánh mô hình**

* Đánh giá qua **Accuracy, Precision, Recall, F1-score**
* Chọn mô hình có hiệu suất tốt nhất

**4. Đánh giá & Điều chỉnh mô hình**

📌 **Các bước đánh giá**  
✔ Dùng tập test để kiểm tra độ chính xác của mô hình  
✔ Tinh chỉnh tham số **learning rate, batch size** để tối ưu kết quả

**1. BERT-base**

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình NLP mạnh mẽ được huấn luyện trước trên tập dữ liệu lớn. Trong mô hình **BERT-base**, chúng ta sử dụng trọng số từ mô hình đã huấn luyện trước (pre-trained) và có thể tinh chỉnh (fine-tune) trên tập dữ liệu cụ thể để cải thiện hiệu suất.

**Cách hoạt động:**

* **Tokenization**: Mô hình sử dụng một tokenizer đặc biệt để chuyển đổi văn bản thành các mã token.
* **Embedding Layer**: Các token được ánh xạ thành vector nhúng có ý nghĩa ngữ nghĩa.
* **Transformer Layers**: Mô hình sử dụng kiến trúc Transformer với cơ chế tự chú ý (self-attention) để học các mối quan hệ giữa các từ trong câu.
* **Output Layer**: Lớp đầu ra có thể được tùy chỉnh để phù hợp với bài toán phân loại, hồi quy hoặc trích xuất thông tin.

**2. BERT-RCNN**

**BERT-RCNN** kết hợp mô hình BERT với mạng nơ-ron tích chập hồi quy (RCNN - Recurrent Convolutional Neural Network) để tăng khả năng học đặc trưng ngữ cảnh.

**Cách hoạt động:**

* **BERT Embedding**: Văn bản đầu vào được mã hóa bằng BERT để trích xuất vector đặc trưng.
* **BiLSTM hoặc BiGRU**: Một mạng LSTM hoặc GRU hai chiều giúp nắm bắt ngữ cảnh của văn bản theo hướng tiến và lùi.
* **CNN Layer**: Áp dụng tích chập (CNN) để trích xuất thông tin cục bộ từ đầu ra của BiLSTM.
* **Max Pooling & Fully Connected Layer**: Lớp pooling giúp chọn lọc đặc trưng quan trọng nhất và sau đó đưa vào lớp Dense để tạo đầu ra cuối cùng.

**Ưu điểm của BERT-RCNN:**

* Khai thác ngữ cảnh hai chiều nhờ BERT và BiLSTM.
* CNN giúp làm nổi bật các đặc trưng quan trọng trong câu.
* Cải thiện độ chính xác so với mô hình BERT thuần túy.

**3. BERT-LSTM**

**BERT-LSTM** kết hợp BERT với LSTM để tăng cường khả năng mô hình hóa chuỗi thời gian trong NLP.

**Cách hoạt động:**

* **BERT Embedding**: Văn bản được nhúng vào không gian vector sử dụng BERT.
* **LSTM Layer**: Một hoặc nhiều lớp LSTM giúp mô hình hiểu rõ hơn mối quan hệ theo thứ tự thời gian giữa các từ.
* **Dense Layer**: Một hoặc nhiều lớp kết nối đầy đủ để ánh xạ đầu ra của LSTM đến lớp phân loại.
* **Output Layer**: Lớp cuối cùng sử dụng hàm kích hoạt phù hợp (ví dụ: sigmoid cho phân loại nhị phân).

**Ưu điểm của BERT-LSTM:**

* LSTM giúp học các phụ thuộc dài hạn trong câu.
* Giữ lại thông tin quan trọng trong chuỗi văn bản.
* Hoạt động tốt trong các bài toán yêu cầu nhận diện ngữ cảnh theo trình tự.

**Tổng kết:**

* **BERT-base**: Phù hợp cho nhiều tác vụ NLP với độ chính xác cao nhưng có thể cần fine-tuning để tối ưu hóa.
* **BERT-RCNN**: Tốt hơn trong việc học đặc trưng cục bộ và toàn cục, thích hợp cho phân loại văn bản.
* **BERT-LSTM**: Giúp xử lý tốt các văn bản có tính chuỗi, phù hợp cho các bài toán cần nhớ thứ tự từ.*Kết quả mô hình* *NTC-SV Dataset (FOODY)*

| **Model** | **Precision(%)** | **Recall(%)** | **F1(%)** |
| --- | --- | --- | --- |
| BERT-base | 90 | **91** | 90 |
| BERT-LSTM | **91** | 88 | 90 |
| BERT-RCNN | 86 | 89 | 87 |

**1. Precision (Độ chính xác theo lớp dương)**

* Đánh giá mức độ chính xác của các dự đoán dương (cảm xúc tích cực nếu dùng nhãn "positive" là lớp dương).
* Công thức:

A black background with white text

AI-generated content may be incorrect.

* Nếu Precision cao, mô hình ít mắc lỗi khi dự đoán một văn bản có cảm xúc dương.

**2. Recall (Độ nhạy, tỷ lệ phát hiện đúng)**

* Đánh giá khả năng phát hiện đúng các văn bản có cảm xúc dương.
* Công thức:

A black background with white text

AI-generated content may be incorrect.

* Nếu Recall thấp, có nghĩa là mô hình bỏ sót nhiều văn bản dương.

**3. F1-score (Chỉ số cân bằng giữa Precision và Recall)**

* Là trung bình điều hòa của Precision và Recall.
* Công thức:

A black background with white text

AI-generated content may be incorrect.

* **Ưu điểm**: Dùng khi có mất cân bằng dữ liệu, giúp cân đối Precision và Recall.