ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC
ĐỀ TÀI: NHẬN DIỆN CẢM XÚC VĂN BẢN BẰNG MÔ HÌNH HỌC SÂU BERT

MÔN HỌC: HỌC THỐNG KÊ
NĂM HỌC: 2024 – 2025

Giảng viên lý thuyết: Ngô Minh Nhựt
Giảng viên hướng dẫn: Lê Long Quốc

THÀNH VIÊN NHÓM

1.	Lê Tiến Tài	22120314	Thành viên
2.	Nguyễn Sinh Trực	22120395	Thành viên
3.	Lê Quốc Vương	22120445	Thành viên
4.	Phạm Tuấn Vương	22120446	Trưởng nhóm

TP. HCM, Ngày 10 tháng 05 năm 2025

MỤC LỤC

I. Bảng phân công công việc

STT	Người thực hiện	Công việc	Mức độ hoàn thành
1	Lê Tiến Tài	Tìm bộ dữ liệu, triển khai web, viết báo cáo	100%
2	Nguyễn Sinh Trực	Tìm bộ dữ liệu, triển khai web, viết báo cáo	100%
3	Lê Quốc Vương	Tìm bộ dữ liệu, train mô hình, viết báo cáo	100%
4	Phạm Tuấn Vương	Tìm bộ dữ liệu, train mô hình, viết báo cáo	100%

II. Giới thiệu bài toán

Trong kỷ nguyên số hiện nay, mạng xã hội và các nền tảng trực tuyến đã trở thành nơi người dùng chia sẻ ý kiến, cảm xúc và đánh giá về sản phẩm, dịch vụ, hay các vấn đề xã hội. Mỗi ngày, hàng triệu bài viết, bình luận, và đánh giá được tạo ra, chứa đựng những thông tin giá trị về cảm xúc và thái độ của người viết. Tuy nhiên, khối lượng dữ liệu khổng lồ này một cách thủ công là điều không thể, đặc biệt khi những phản hồi và cảm xúc của người dùng có thể ảnh hưởng trực tiếp đến các chiến lược kinh doanh và quyết định của doanh nghiệp. Vì vậy, bài toán **phân loại cảm xúc (Sentiment Classification)** lựa chọn đề tài hợp lý và mang tính ứng dụng cao, có thể giải quyết các vấn đề cấp bách của các doanh nghiệp trong việc xử lý thông tin từ khách hàng. Việc hiểu được cảm xúc của người dùng không chỉ giúp các tổ chức cải thiện dịch vụ, tăng cường mối quan hệ với khách hàng và phát triển chiến lược kinh doanh hiệu quả hơn mà còn mở rộng ra cho nhiều lĩnh vực xã hội khác trong cuộc sống.

III. Mô tả chi tiết bộ dữ liệu

1. Giới thiệu

Bộ dữ liệu **dair-ai/emotion** là một tập dữ liệu về **phân loại cảm xúc trong văn bản tiếng Anh**, được thu thập từ các dòng tweet trên Twitter. Mỗi dòng tweet được gán nhãn theo một trong **6 cảm xúc cơ bản**:

- **sadness** (buồn)
- **joy** (vui)
- **love** (yêu)
- **anger** (giận)
- **fear** (sợ)
- **surprise** (ngạc nhiên)

Bộ dữ liệu này thường được dùng để huấn luyện và đánh giá các mô hình học máy cho bài toán **nhận diện cảm xúc từ văn bản**, đặc biệt là trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP).

2. Nguồn gốc

Bộ dữ liệu này được phát triển bởi nhóm DAIR.AI và được sử dụng trong nghiên cứu "CARER: Contextualized Affect Representations for Emotion Recognition" nhằm xây dựng các biểu diễn cảm xúc theo ngữ cảnh từ văn bản.

3. Phương pháp lấy mẫu

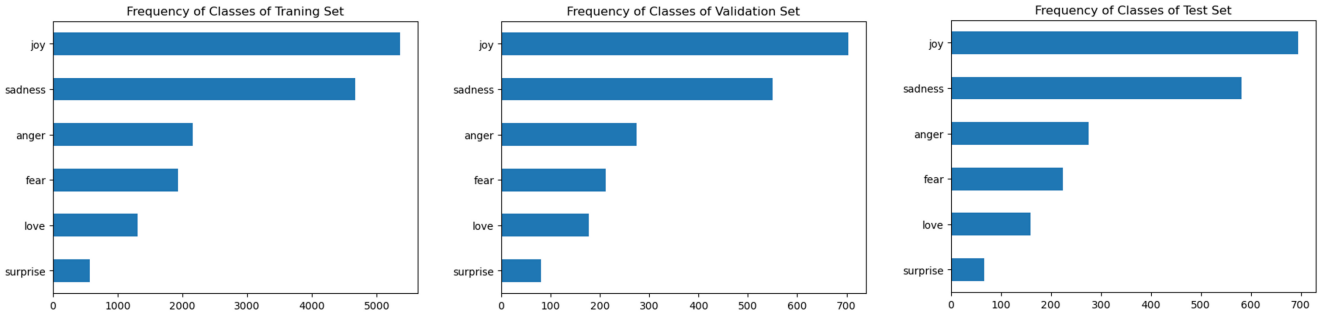
Bộ dữ liệu này là kết quả được tổng hợp, xử lý thông qua hai tập dữ liệu thô trước đó. Hai bộ dữ liệu thô được thu thập qua API của Twitter, cụ thể:

1) **Bộ dữ liệu chủ quan (S)**: Thu thập hơn 2 triệu tweet từ 339 hashtag, sử dụng làm nhãn yếu.
2) **Bộ dữ liệu khách quan (O)**: Thu thập hơn 2 triệu tweet từ các tài khoản tin tức.

4. Cấu trúc bộ dữ liệu

Bộ dữ liệu có hai cấu hình:

- **split**: Gồm 20.000 mẫu, chia thành:
  - Tập train: 16.000 mẫu
  - Tập validation: 2.000 mẫu
  - Tập test: 2.000 mẫu



Hình 1: Phân bố dữ liệu

- **unsplit**: Gồm 416.809 mẫu trong một tập duy nhất

Bộ dữ liệu được nhóm tác giả chia theo cách phân tầng.

Mỗi mẫu dữ liệu gồm hai thuộc tính:

- text: một chuỗi văn bản (string), chính là nội dung của tweet
- label: một nhãn phân loại, với các giá trị có thể bao gồm sadness (0), joy (1), love (2), anger (3), fear (4), surprise(5).

Ví dụ:

{
"text": "im feeling quite sad and sorry for myself but ill snap out of it soon",
"label": 0
}

5. Giấy phép sử dụng

Bộ dữ liệu được sử dụng cho giáo dục và nghiên cứu. Khi sử dụng, vui lòng trích dẫn:

@inproceedings{saravia-etal-2018-carer,
title = "{CARER}: Contextualized Affect Representations for Emotion Recognition",
author = "Saravia, Elvis and
Liu, Hsien-Chi Toby and
Huang, Yen-Hao and
Wu, Junlin and
Chen, Yi-Shin",
booktitle = "Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing",
month = oct # "-" # nov,

year = "2018",
address = "Brussels, Belgium",
publisher = "Association for Computational Linguistics",
url = "https://www.aclweb.org/anthology/D18-1404",
doi = "10.18653/v1/D18-1404",
pages = "3687--3697",
abstract = "Emotions are expressed in nuanced ways, which varies by collective or individual experiences, knowledge, and beliefs. Therefore, to understand emotion, as conveyed through text, a robust mechanism capable of capturing and modeling different linguistic nuances and phenomena is needed. We propose a semi-supervised, graph-based algorithm to produce rich structural descriptors which serve as the building blocks for constructing contextualized affect representations from text. The pattern-based representations are further enriched with word embeddings and evaluated through several emotion recognition tasks. Our experimental results demonstrate that the proposed method outperforms state-of-the-art techniques on emotion recognition tasks.",}

IV. Mô hình BERT

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình ngôn ngữ mạnh mẽ được thiết kế được thiết kế để **pre-train deep bidirectional representations** từ văn bản chưa được gán nhãn bằng cách đồng thời xem xét ngữ cảnh từ cả hai phía - trái và phải. Đây chính là điểm khác biệt cốt lõi so với mô hình GPT, vốn chỉ xử lý văn bản theo chiều từ trái sang phải.

- *Pre-train*: giai đoạn tiền huấn luyện, tức là huấn luyện mô hình trên một dữ liệu lớn tổng quát trước khi thực hiện một tác vụ cụ thể..
- *Deep*: BERT gồm nhiều tầng Transformer encoder được xếp chồng lên nhau.

Ví dụ, trong câu “Tôi thích ăn **phở** nóng”:

- GPT sẽ chỉ nhìn từ trái sang phải và đoán từ tiếp theo, cụ thể "Tôi thích ăn __".
- Ngược lại, BERT đồng thời nhìn cả hai phía: “Tôi thích ăn __ nóng”, từ đó hiểu rằng chỗ trống cần điền là tên một món ăn có tính chất nóng.

Nhờ vào quá trình *pre-trained* trên dữ liệu lớn, mô hình BERT có khả năng học được rất nhiều kiến thức ngôn ngữ tổng quát và sau đó mô hình có thể fine-tune trên dữ liệu cụ thể của từng bài toán cần giải quyết (*phân loại cảm xúc*) chỉ bằng cách **thêm một tầng đầu ra phù hợp với bài toán.** Điều này giúp tạo ra mô hình đạt kết quả tốt nhất cho nhiều tác vụ khác nhau mà không cần phải thiết kế một mô hình riêng biệt cho từng tác vụ.

BERT được huấn luyện trên hai tập dữ liệu lớn:

- **Google BooksCorpus** (800 triệu từ)
- **English Wikipedia** (2,500 triệu từ)

Trong đồ án này, nhóm em lựa chọn distilBERT - là một phiên bản nhỏ gọn hơn của BERT với tốc độ nhanh hơn tới 60% trong khi vẫn giữ được hiệu suất đạt 97% của mô hình BERT. Nhóm sẽ tiếp cận theo hướng **Fine-Tuning Transformers**, tức là dùng bộ dữ liệu của chúng em để điều chỉnh toàn bộ các tham số đã được pre-trained giúp mô hình thực hiện tác vụ phân lớp cụ thể.

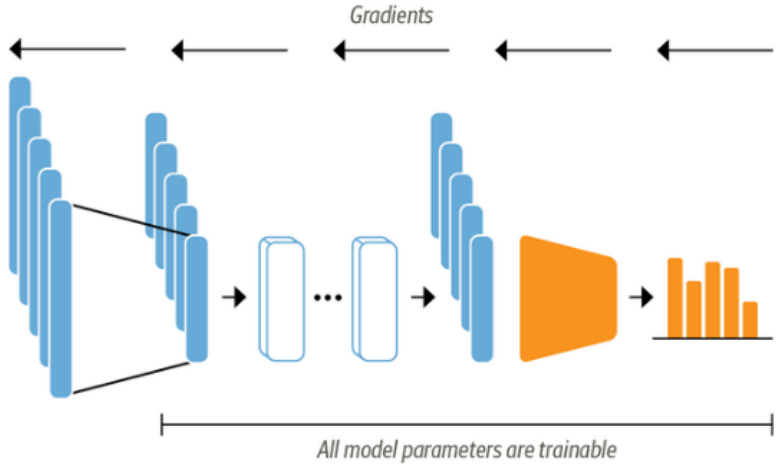


Figure 2-6. When using the fine-tuning approach the whole DistilBERT model is trained along with the classification head

Load mô hình đã được pre-trained

```
import torch
model_ckpt = "distilbert-base-uncased"
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

```
from transformers import AutoModelForSequenceClassification
num_labels = 6

model = (AutoModelForSequenceClassification
        .from_pretrained(model_ckpt, num_labels=num_labels)
        .to(device))
```

Load tokenizer của mô hình distilBERT (WordPiece) và tokenize bộ dữ dữ liệu

```
from transformers import AutoTokenizer
model_ckpt = "distilbert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_ckpt)

def tokenize(batch):
  tokenized = tokenizer(batch["text"], padding=True, truncation=True)
  tokenized["label"] = batch["label"]
  tokenized["text"] = batch["text"]
  return tokenized

emotions_encoded = emotions.map(tokenize, batched=True, batch_size=None
```

Chọn các độ đo

```
from sklearn.metrics import accuracy_score, f1_score

def compute_metrics(pred):
  labels = pred.label_ids
  preds = pred.predictions.argmax(-1)
  f1 = f1_score(labels, preds, average="weighted")
  acc = accuracy_score(labels, preds)
  return {"accuracy": acc, "f1": f1}
```

Chọn các siêu tham số

```
from transformers import Trainer, TrainingArguments

batch_size = 64
logging_steps = len(emotions_encoded["train"]) // batch_size
model_name = f"{model_ckpt}-finetuned-emotion-2"
training_args = TrainingArguments(output_dir=model_name,
                                  num_train_epochs=10,
                                  learning_rate=2e-5,
                                  per_device_train_batch_size=batch_size,
                                  per_device_eval_batch_size=batch_size,
                                  weight_decay=0.01,
                                  evaluation_strategy="epoch",
                                  disable_tqdm=False,
                                  logging_steps=logging_steps,
                                  push_to_hub=True,
                                  log_level="error")
```

Train mô hình

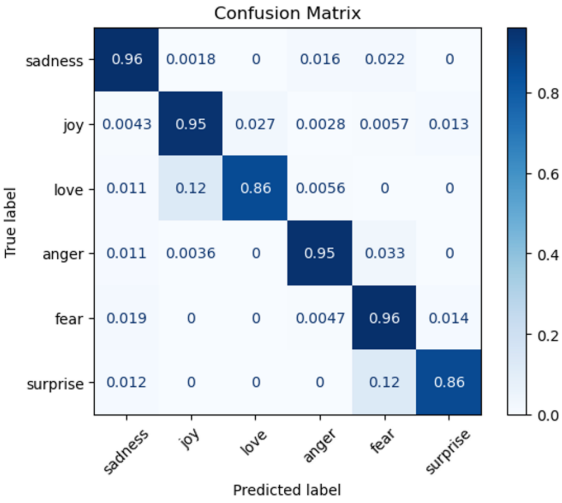
```
from transformers import Trainer
trainer = Trainer(model=model, args=training_args,
                  compute_metrics=compute_metrics,
                  train_dataset=emotions_encoded["train"],
                  eval_dataset=emotions_encoded["validation"],
                  tokenizer=tokenizer)
trainer.train();
```

[2500/2500 6:49:27, Epoch 10/10]

Kết quả huấn luyện

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.822400	0.269757	0.914500	0.914733
2	0.212000	0.161553	0.932500	0.932164
3	0.134500	0.152339	0.937000	0.937498
4	0.107200	0.151918	0.942000	0.942276
5	0.085000	0.144389	0.938500	0.938645
6	0.071100	0.153147	0.939000	0.939225
7	0.060000	0.172611	0.936000	0.935312
8	0.051200	0.167168	0.940000	0.939924
9	0.040400	0.176795	0.937500	0.937588
10	0.039000	0.177495	0.940500	0.940392

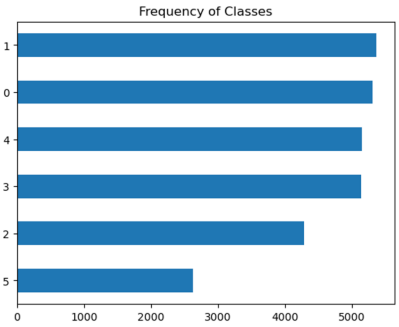
Đánh giá mô hình trên validation set

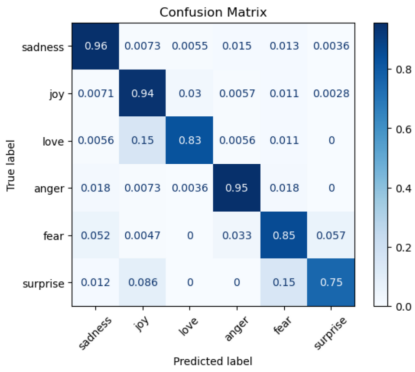


Predicted label

	precision	recall	f1-score	support
sadness	0.98	0.96	0.97	550
joy	0.97	0.95	0.96	704
love	0.89	0.86	0.87	178
anger	0.95	0.95	0.95	275
fear	0.85	0.96	0.90	212
surprise	0.85	0.86	0.86	81
accuracy			0.94	2000
macro avg	0.92	0.92	0.92	2000
weighted avg	0.94	0.94	0.94	2000

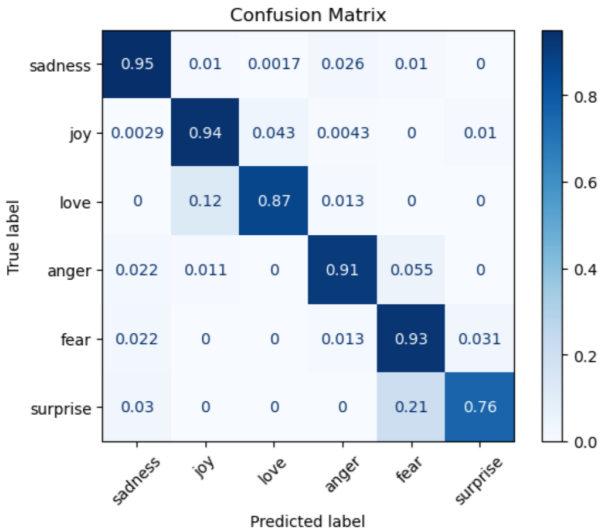
- Ta thấy kết quả rất tốt. Mô hình đạt accuracy lên đến 94%. Nhưng các nhãn love và surprise có kết quả dưới 90%. Nguyên nhân của việc này là do bộ dữ liệu bị lệch rất nhiều ở hai nhãn này.
- Nhóm đã thử xử lý dữ liệu mất cân bằng này bằng cách, dịch các câu ban đầu sang các ngôn ngữ khác rồi dịch ngược lại tiếng Anh. Nhưng kết quả cho thấy cũng không tốt hơn. Cụ thể như sau:





	precision	recall	f1-score	support
sadness	0.96	0.96	0.96	550
joy	0.94	0.94	0.94	704
love	0.85	0.83	0.84	178
anger	0.93	0.95	0.94	275
fear	0.84	0.85	0.85	212
surprise	0.79	0.75	0.77	81
accuracy			0.92	2000
macro avg	0.89	0.88	0.88	2000
weighted avg	0.92	0.92	0.92	2000

Đánh giá mô hình trên test set



	precision	recall	f1-score	support
sadness	0.97	0.97	0.97	581
joy	0.94	0.94	0.94	695
love	0.81	0.82	0.82	159
anger	0.92	0.93	0.93	275
fear	0.87	0.92	0.89	224
surprise	0.84	0.65	0.74	66
accuracy			0.93	2000
macro avg	0.89	0.87	0.88	2000
weighted avg	0.93	0.93	0.93	2000

V. Khác biệt giữa mô hình BERT và các mô hình khác

1. Mô hình truyền thống (dựa trên đặc trưng thủ công)

Các mô hình: Naive Bayes, SVM, Logistic Regression (kết hợp với TF-IDF, Bag of Words).

- Ưu điểm:
  - Dễ huấn luyện, nhanh, cần ít tài nguyên.
  - Dễ hiểu và triển khai.
- Nhược điểm:
  - Không hiểu ngữ cảnh (context), không xử lý tốt sự mơ hồ của ngôn ngữ.
  - Phụ thuộc vào đặc trưng thủ công → hiệu quả thấp hơn mô hình học sâu.

2. Mạng nơ-ron truyền thống

Các mô hình: CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), LSTM, GRU.

- Ưu điểm:
  - Biết khai thác chuỗi từ, hiểu được thứ tự từ (đặc biệt RNN, LSTM).
  - Tốt hơn các mô hình truyền thống về xử lý ngôn ngữ.
- Nhược điểm:
  - Không xử lý được mối quan hệ xa giữa các từ (LSTM có thể quên thông tin).
  - Không song song hóa tốt (huấn luyện chậm).
  - Cần huấn luyện từ đầu nếu không có mô hình tiền huấn luyện.

3. Mô hình Word Embedding kết hợp với MLP

Dùng Word2Vec, GloVe để biểu diễn từ → đưa vào mạng neural đơn giản.

- Ưu điểm:
  - Hiểu nghĩa của từ tốt hơn TF-IDF.
  - Nhẹ hơn mô hình transformer.
- Nhược điểm:
  - Vector từ cố định, không theo ngữ cảnh (ví dụ: từ "bank" trong hai câu khác nhau vẫn giống nhau).
  - Không mạnh bằng mô hình ngôn ngữ tiền huấn luyện như BERT.

Vì vậy, nhóm em chọn mô hình **BERT** (cụ thể là **distilBERT**). Vì **BERT** vượt trội về khả năng hiểu ngữ cảnh và cấu trúc ngôn ngữ, phù hợp cho các bài toán phức tạp như phân loại cảm xúc. Các mô hình cũ nhanh hơn, đơn giản hơn nhưng độ chính xác thấp hơn, không đủ mạnh để nắm bắt sự tinh tế trong cảm xúc văn bản.

VI. Xử lý cảm xúc văn bản

1. Tiền xử lý văn bản đầu vào (bao gồm chuẩn hóa và dịch)

a. Chuẩn hóa văn bản (normalize_text)
   - Chuyển toàn bộ văn bản thành chữ thường.
   - Loại bỏ các ký tự không phải là chữ cái, số và khoảng trắng.
   - Rút gọn khoảng trắng dư thừa.

b. Dịch văn bản sang tiếng Anh (translate_to_english)
   - Dùng thư viện langdetect để phát hiện ngôn ngữ.
   - Nếu là tiếng Việt (vi), dùng deep_translator.GoogleTranslator để dịch sang tiếng Anh.
   - Nếu văn bản đã là tiếng Anh, giữ nguyên.
   - Lý do: mô hình phân tích cảm xúc được huấn luyện trên tập dữ liệu tiếng Anh (như BERT-based), nên nếu đầu vào là tiếng Việt thì sẽ được dịch qua tiếng Anh để mang

đi dự đoán kết quả. Ta sẽ dịch trước rồi chuẩn hóa sau vì hàm chuẩn hóa chỉ áp dụng cho tiếng Anh.

c. Xử lý văn bản dài (split_text_into_chunks)
   - Nếu văn bản > 512 tokens (giới hạn của BERT), chia thành nhiều đoạn nhỏ hơn max_tokens (mặc định: 450) với overlap giữa các đoạn (mặc định: 50 tokens).
   - Cố gắng cắt tại dấu câu hoặc từ nối (and, but, however, etc.) để tránh cắt giữa câu.
   - Lý do: BERT không thể xử lý >512 tokens trong một lần dự đoán.

2. Phân tích cảm xúc văn bản (kể cả văn bản dài)

Dự đoán cảm xúc (predict_single_chunk và predict)

- Dùng mô hình BERT (hoặc biến thể tương thích) đã fine-tuned cho bài toán phân tích cảm xúc.
- Tokenize văn bản, dùng mô hình để trả ra logits, tính softmax để thu được xác suất các lớp cảm xúc.
- Với văn bản dài (nhiều đoạn), tính trung bình xác suất từ tất cả các đoạn.

VII. Triển khai mô hình trên web

1. Tổng quan

Ứng dụng web phân tích cảm xúc được xây dựng với kiến trúc client-server, sử dụng:

- Back-end: Python Flask.
- Front-end: HTML, CSS (Tailwind CSS), JavaScript.

2. Kiến trúc hệ thống

- Back-end (Flask):
  - Sử dụng Flask làm web framework.
  - Tích hợp CORS để cho phép cross-origin requests.
  - API endpoint chính: /analyze để xử lý phân tích cảm xúc.
  - Sử dụng mô hình DistilBERT đã được fine-tune cho phân tích cảm xúc.
- Front-end:
  - Giao diện người dùng được xây dựng với HTML5 và Tailwind CSS.
  - JavaScript xử lý tương tác người dùng và gọi API.
  - Sử dụng ECharts để hiển thị biểu đồ thống kê.
  - Lưu trữ lịch sử phân tích trong localStorage.

3. Hướng dẫn sử dụng

Chạy chương trình:

1) Cài đặt các thư viện yêu cầu:
   - pip install requirements
2) Chạy lệnh:
   - python model.py
3) Truy cập: http://localhost:5000/ hoặc http://127.0.0.1:5000/

4. Web Version

Trong quá trình phát triển ứng dụng web, nhóm đã xây dựng hai phiên bản giao diện:

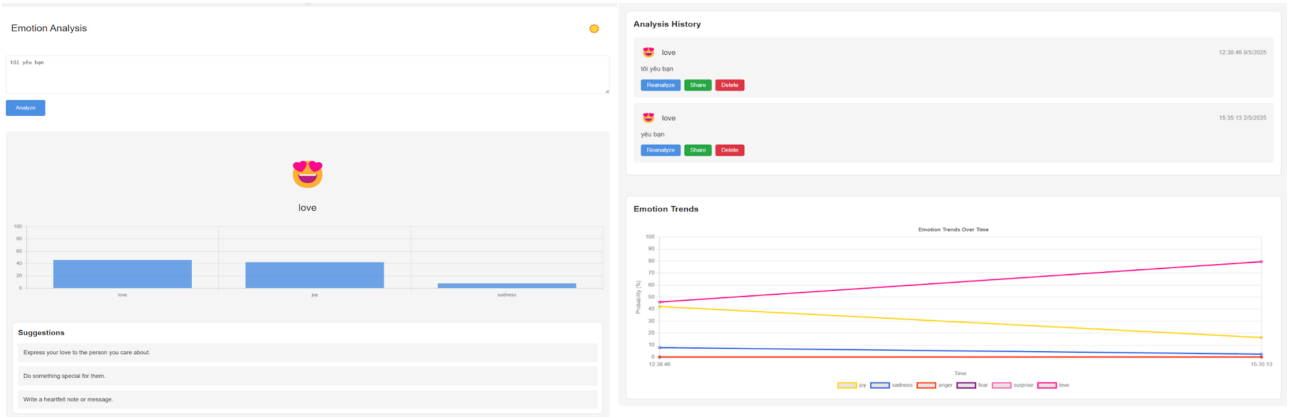
- Phiên bản 1 (Version 1): là phiên bản giao diện đơn giản ban đầu, đóng vai trò làm nền tảng định hướng cho việc phát triển giao diện chính thức.
- Phiên bản 2 (Version 2): là giao diện hoàn chỉnh được phát triển từ phiên bản 1. Phiên bản này là phiên bản cuối cùng của nhóm em chọn.

5. Các tính năng

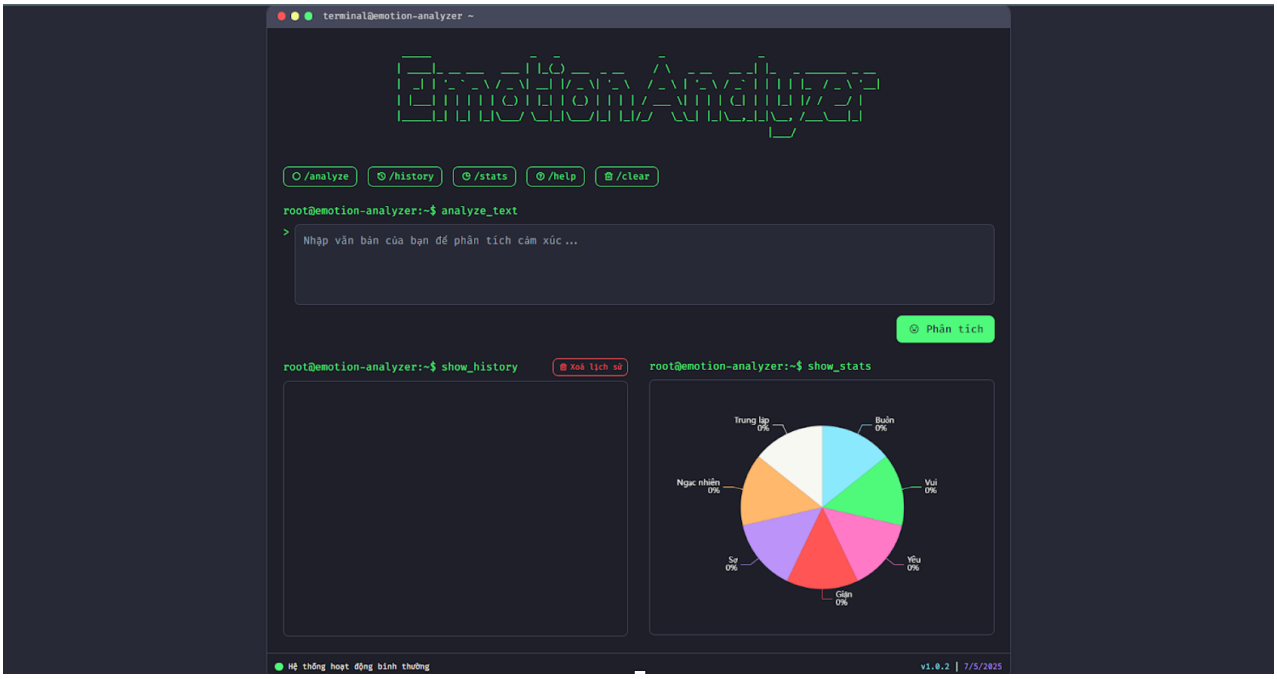
- **Version 1 – Giao diện đơn giản ban đầu**
  - Phân tích cảm xúc từ văn bản nhập vào.
  - Hiển thị lịch sử các văn bản đã nhập.
  - Hiển thị biểu đồ đường thống kê sự thay đổi cảm xúc của văn bản qua. các mốc thời gian.
  - Biểu đồ cột so sánh top 3 xác suất nhiều nhất về cảm xúc có trong câu
  - Xóa, chia sẻ, tải xuống lịch sử phân tích cảm xúc.
  - Chế độ sáng tối giúp cho web thích hợp với nhiều người dùng.
  - Hiển thị các lời khuyên, chia sẻ đối với nhãn cảm xúc.
- **Version 2 – Giao diện hoàn chỉnh, nâng cấp từ Version 1**
  - analyze_text: phân tích cảm xúc từ văn bản nhập vào.
  - show_history: hiển thị lịch sử các văn bản đã nhập.
  - show_stats: hiện thị biểu đồ tròn thống kê tỷ lệ cảm xúc của văn bản.
  - clear: xoá văn bản nhập vào và ẩn kết quả.

6. Giao diện

Version 1



Version 2



VIII. Tham khảo

- Dataset: https://huggingface.co/datasets/dair-ai/emotion
- Natural Language Processing with Transformers, Revised Edition by Lewis Tunstall, Leandro von Werra, Thomas Wolf.
- https://arxiv.org/abs/1706.03762
- https://arxiv.org/abs/1810.04805