Emotion-Aware Sentiment Analysis Chatbot using GoEmotions, DistilBERT, and OpenRouter

Overview

This project is a comprehensive implementation of an emotion-aware sentiment analysis chatbot built using DistilBERT fine-tuned on the GoEmotions dataset, and enhanced with OpenRouter's LLM (Groq) for empathetic and contextual responses. The chatbot not only detects nuanced emotional cues from user input but also engages in intelligent, context-aware conversations, offering human-like interactions based on real-time emotion detection.

The model is capable of classifying input text into one of 28 fine-grained emotions (as defined in the GoEmotions dataset), and it uses that emotional insight to prompt a conversational AI via OpenRouter, allowing the chatbot to respond with emotional intelligence.

Objectives

- Fine-tune a lightweight transformer model (DistilBERT) on GoEmotions for high-accuracy emotion detection.
- Build an interactive, real-time chatbot interface that responds empathetically based on detected emotions.
- Integrate OpenRouter's powerful conversational models to generate human-like replies.
- Enable checkpointing and efficient retraining using Google Drive for persistence.
- Create a presentable, deployable Gradio UI interface for testing and showcasing.

Technologies Used

- Transformers (Hugging Face)
- Datasets (Hugging Face)
- Torch / PyTorch
- Gradio
- OpenRouter / Groq
- Google Colab + Google Drive
- Scikit-learn

Dataset: GoEmotions

- Source: Google’s GoEmotions dataset (via Hugging Face)
- Total emotions: 28 emotions + 1 neutral class
- Examples per emotion: ~58k carefully curated Reddit comments
- Format: Each entry contains a piece of text and one or more associated emotion labels.

Model Architecture and Training

Model Used: distilbert-base-uncased

Modified For: Sequence classification (28 emotion classes)

Preprocessing

Each text is tokenized into input IDs and attention masks. Truncation ensures max length doesn’t exceed model input limits. Labels are simplified for single-label classification.

Dataset Split

- Training Set: 80% of the dataset

- Evaluation Set: 20%

Training Configuration

- Learning Rate: 2e-5
- Epochs: 40
- Batch Size: 16
- Weight Decay: 0.01
- Evaluation Strategy: Epoch-wise
- Checkpointing enabled
- Gradient accumulation: 2
- Mixed-precision training (fp16): Enabled
- Optimizer: adamw_torch

Metric

- Metric Used: Accuracy

Emotion Detection Pipeline

After training, the model is converted into a pipeline for direct use for real-time detection of emotions.

OpenRouter Integration (Groq)

Using the openai Python library (configured to point to https://openrouter.ai/api/v1), the system sends prompts to conversational LLMs like Mixtral.

User Interaction (Command Line or UI)

- Terminal Chat Loop: Interacts via CLI
- Gradio Interface: Clean UI allows user-friendly interaction

Checkpointing and Persistence

All training progress and checkpoints are saved to Google Drive. Training can resume from the last checkpoint.

Results & Performance

- Achieved stable training accuracy after 40 epochs.
- Real-world sentence predictions are accurate.
- Chatbot responds empathetically and conversationally.

Key Features

- Emotion detection using a fine-tuned transformer
- Conversational response via OpenRouter
- Real-time predictions
- Portable, GPU-compatible, and cloud-deployable
- Suitable for integration with mental health support tools

Future Improvements

- Multi-label classification support
- Speech integration
- Deployment on Hugging Face Spaces or Streamlit Cloud
- Add chat history memory
- Emotion trend analytics

License

MIT License

Acknowledgments

- Hugging Face
- Google (GoEmotions)
- OpenRouter
- PyTorch
- Gradio