Introduction

With the rapid advancements in Natural Language Processing (NLP) and Large Language Models (LLMs), emotion classification has emerged as a crucial task with widespread applications in customer sentiment analysis, mental health assessment, social media monitoring, and human-computer interaction. By leveraging state-of-the-art LLMs, it is now possible to automate emotion detection from textual data with high accuracy, enhancing decision-making processes in various industries.

This study focuses on the fine-tuning and evaluation of three advanced LLMs—Mistral-7B, LLaMA-2, and Flan-T5—for the task of emotion classification. The models are trained using the ISEAR (International Survey on Emotion Antecedents and Reactions) dataset, a widely used corpus for emotion analysis. The dataset consists of text samples labeled with seven distinct emotions: anger, disgust, fear, guilt, joy, sadness, and shame. The primary objective of this project is to compare the performance of these LLMs and analyze their suitability for real-world emotion classification tasks.

The findings from this study will provide valuable insights into the capabilities of state-of-the-art LLMs in emotion detection, highlighting their applications, limitations, and future directions for improvement.

1. Problem Definition

Emotion recognition from text is a crucial task in natural language processing (NLP) with applications in sentiment analysis, mental health monitoring, and customer feedback analysis. Understanding emotions in textual data enables better human-computer interactions, enhances recommendation systems, and aids in decision-making processes.

This study focuses on classifying emotions from text using large language models (LLMs). Given an input text, the objective is to predict one of seven emotions: *anger, disgust, fear, guilt, joy, sadness,* and *shame*. The classification is based on the International Survey on Emotion Antecedents and Reactions (ISEAR) dataset, which contains textual data labeled with corresponding emotions.

The complexity of this problem arises from several factors:

1. **Subjectivity of Emotions** – Different individuals may interpret the same text differently based on context, cultural background, or personal experiences.
2. **Lexical Ambiguity** – Words and phrases can convey different emotions depending on sentence structure and surrounding words.
3. **Data Imbalance** – Some emotions may be more frequently expressed than others, affecting model learning and generalization.
4. **LLM Adaptability** – Fine-tuning large pre-trained models requires significant computational resources, and their effectiveness depends on the quality and quantity of training data.

This project aims to evaluate the effectiveness of three different LLM architectures for emotion classification:

- Mistral-7B
- LLaMA-2
- Flan-T5

By comparing these models, we aim to identify the best-performing architecture for emotion detection, analyze their strengths and weaknesses, and provide insights into improving automated emotion recognition systems.

2. Data Preprocessing & Exploration

Data preprocessing is a critical step in any machine learning pipeline, ensuring that the dataset is clean, structured, and suitable for model training. In this study, the **International Survey on Emotion Antecedents and Reactions (ISEAR) dataset** is used for training and evaluating different LLM-based classifiers. The preprocessing pipeline involves several steps, including **data cleaning, tokenization, handling imbalanced data, and exploratory data analysis (EDA).**

2.1. Dataset Overview

The dataset consists of **7,279 records** and **six key attributes**, including **text, emotion, intensity, gender, and age**. The emotion labels are distributed across **seven categories**, ensuring a diverse representation of sentiments. The **intensity** attribute provides additional granularity, indicating the strength of the expressed emotion on a **scale from 0 to 4**.

There are **no missing values** in the core text and emotion columns, ensuring data integrity. The **age distribution** ranges from **18 to 58 years**, with a mean of **22.3 years**, indicating that the dataset primarily consists of responses from younger individuals. The **gender distribution** is relatively balanced, with both male and female participants.

The most frequent emotion in the dataset is **joy**, appearing **1,051 times**, followed by other negative and neutral emotions. The dataset also contains **7,210 unique text entries**, suggesting minimal duplication and a wide range of user expressions. This balanced and diverse dataset provides a solid foundation for training and evaluating emotion classification models.

```
Dataset contains 7279 rows and 6 columns.

   id                                               text  emotion  intensity  \
0   0  During the period of falling in love, each tim...      joy          3
1   1         When I was involved in a traffic accident.     fear          2
2   2  When I was driving home after  several days of...    anger          3
3   3  When I lost the person who meant the most to me.   sadness          4
4   4  The time I knocked a deer down – the sight of ...  disgust          4

  gender  age
0   male   33
1   male   33
2   male   33
3   male   33
4   male   33

Column Names: ['id', 'text', 'emotion', 'intensity', 'gender', 'age']

Missing Values:
 id           0
text         0
emotion      0
intensity    0
gender       7
age          0
dtype: int64

Dataset Summary:
                id                      text emotion    intensity  gender  \
count   7279.000000                      7279    7279  7279.000000    7272
unique          NaN                      7210       7          NaN       2
top             NaN  When my grandfather died.     joy          NaN  female
freq            NaN                         8    1051          NaN    3999
mean    3645.516142                       NaN     NaN     2.860695     NaN
std     2105.964821                       NaN     NaN     0.975570     NaN
min        0.000000                       NaN     NaN     0.000000     NaN
25%     1820.500000                       NaN     NaN     2.000000     NaN
50%     3647.000000                       NaN     NaN     3.000000     NaN
75%     5469.500000                       NaN     NaN     4.000000     NaN
max     7292.000000                       NaN     NaN     4.000000     NaN

               age
count   7279.000000
unique          NaN
top             NaN
freq            NaN
mean      22.316527
std        3.757318
min       18.000000
25%       20.000000
50%       21.000000
75%       24.000000
```

2.2. Data Cleaning

The dataset undergoes preprocessing to remove inconsistencies and prepare the text for model input. The key cleaning steps include:

- **Handling missing values** – Checking for null or empty fields in the dataset and removing or imputing missing entries.
- **Removing duplicate entries** – Ensuring the dataset does not contain redundant data points that could bias training.
- **Standardizing text** – Converting all text to lowercase to maintain consistency.
- **Removing special characters and unnecessary whitespace** – Keeping only alphanumeric characters and punctuation relevant to semantic meaning.
- **Tokenization** – Converting textual data into a format that can be processed by the tokenizer of the selected LLMs.

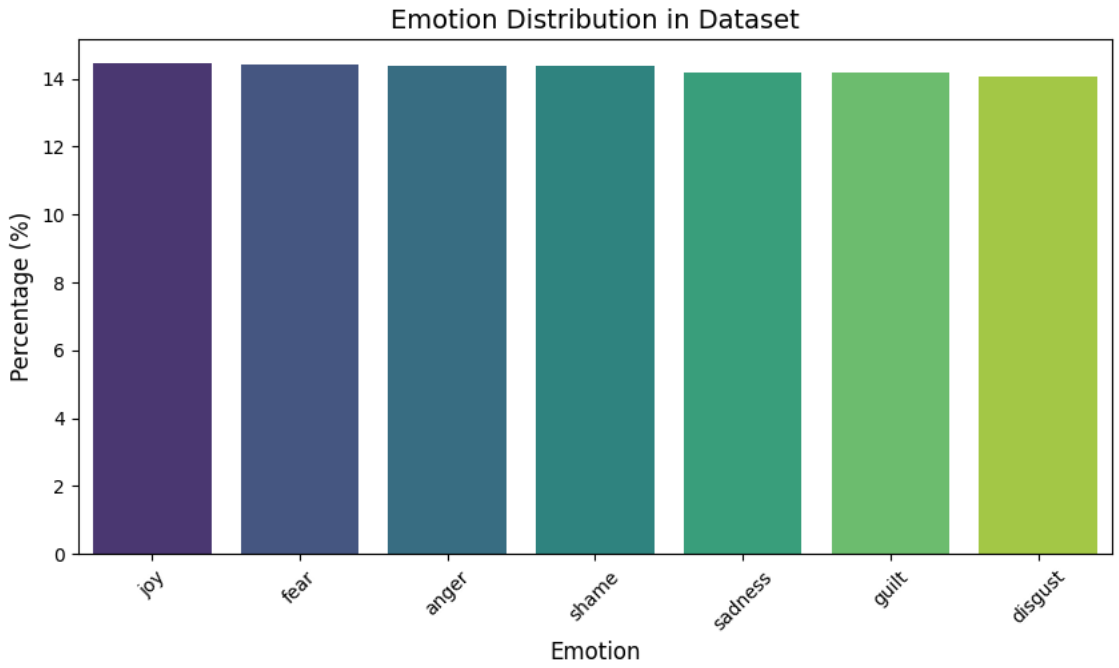
```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
                                                text  \
0  During the period of falling in love, each tim...
1         When I was involved in a traffic accident.
2  When I was driving home after  several days of...
3  When I lost the person who meant the most to me.
4  The time I knocked a deer down – the sight of ...

                                          clean_text
0  period falling love time met especially met lo...
1                          involved traffic accident
2  driving home several day hard work motorist ah...
3                              lost person meant
4  time knocked deer sight animal injury helpless...
```

2.3. Exploratory Data Analysis (EDA)

Before training, a thorough analysis of the dataset is conducted to understand the distribution and characteristics of the text data.

2.3.1. Emotion Distribution Analysis

A **bar chart** is plotted to show the distribution of different emotions in the dataset. This helps in identifying any imbalances in class representation.



The dataset shows a **balanced distribution** of emotions, ensuring fair representation across *joy, fear, anger, shame, sadness, guilt, and disgust*. This helps in training models without bias toward any particular emotion.

Equal representation improves **consistent model performance**, but subtle emotions like *shame* and *guilt* may still be harder to distinguish. **Misclassification analysis** is necessary to understand overlaps, especially between similar emotions.

A well-balanced dataset like this is ideal for **emotion analysis in mental health, customer feedback, and social media tracking**, making AI models more reliable in real-world applications.

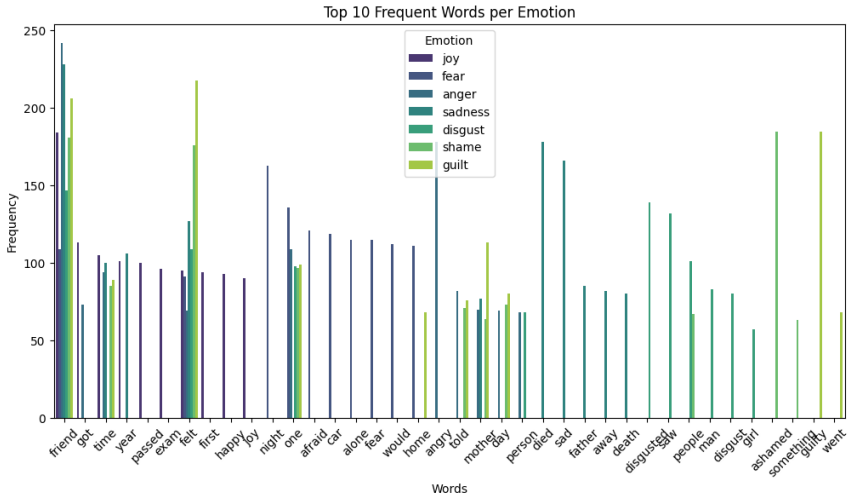
2.3.2. Word Frequency Analysis

To understand the most frequently used words in each emotion category:

- **Word Cloud** is generated for each emotion, highlighting common terms associated with specific sentiments.

The **word cloud visualizations** highlight key terms associated with each emotion, revealing common themes in emotional expressions. **Joy** is linked to achievements and positive events with words like *happy, exam, passed*. **Fear** is driven by uncertainty and danger, reflected in terms like *afraid, night, dark*. **Anger** often stems from conflicts, as seen in *angry, friend, time*. **Sadness** is frequently tied to loss, with words such as *friend, died, mother*. **Disgust** is associated with unpleasant experiences, indicated by *people, saw, disgusted*. **Shame** emerges from self-perception and judgment, shown by *ashamed, told, felt*, while **Guilt** is linked to moral dilemmas, with terms like *guilty, mother, felt*. These insights reinforce the strong relationship between words and emotions, aiding emotion classification models.

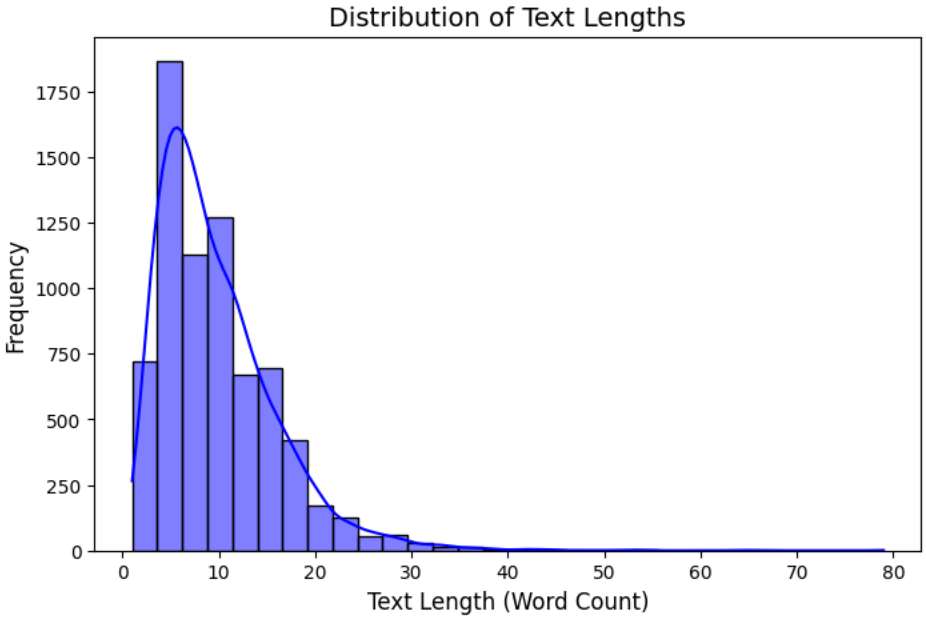
- **Term Frequency Analysis** is conducted to identify key terms that strongly correlate with certain emotions.



The **term frequency analysis** reveals the most common words associated with each emotion. The word **"friend"** appears frequently across multiple emotions, indicating its strong impact on emotional experiences. Words like **"happy"** and **"exam"** are dominant in **joy**, while **"afraid"**, **"alone"**, and **"night"** characterize **fear**. **Anger** is often linked to **"angry"**, **"mother"**, and **"time"**, reflecting interpersonal conflicts. **Sadness** is associated with **"died"**, **"father"**, and **"death"**, emphasizing loss. **Disgust** includes terms like **"disgusted"**, **"people"**, and **"mean"**, highlighting unpleasant interactions. **Shame** and **guilt** are driven by words such as **"ashamed"**, **"something"**, and **"guilty"**, reflecting personal regret. This analysis helps in understanding how words strongly correlate with specific emotions, enhancing text-based emotion classification.

2.3.3. Sentence Length Distribution

- **Histogram** is plotted to analyze the average length of sentences across different emotions.

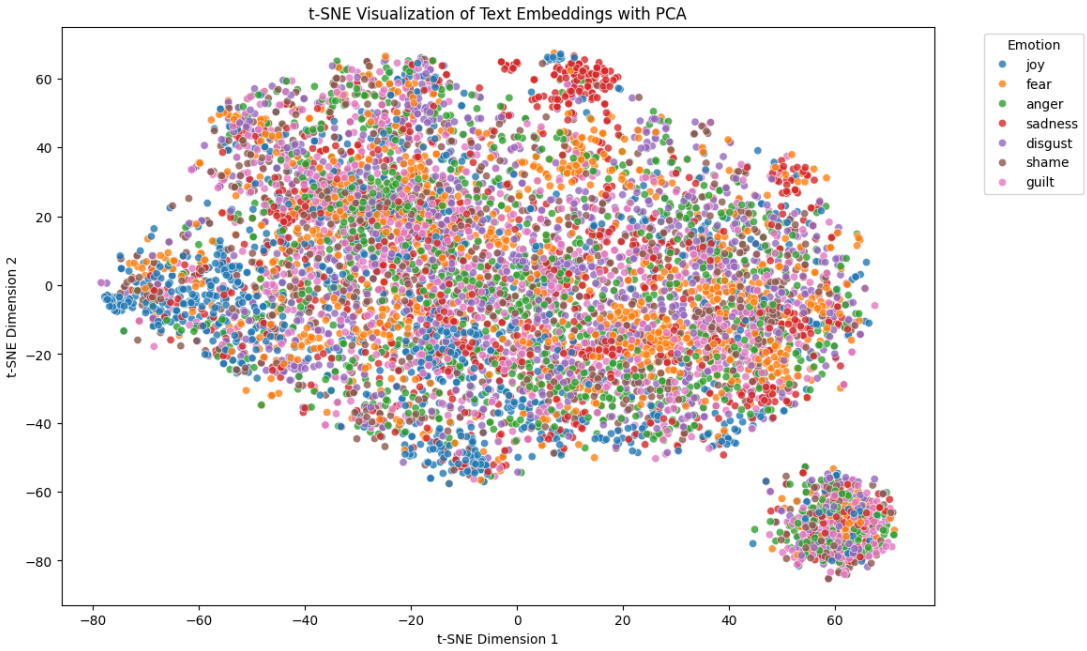


The **text length distribution** shows that most samples contain **fewer than 15 words**, with a peak around **5-10 words**. The distribution is **right-skewed**, indicating a small number of longer texts. This suggests that the dataset primarily consists of **short emotional expressions**, which is beneficial for models that handle short text classification. However, the presence of longer texts highlights the need for **truncation or padding** strategies during preprocessing to maintain consistency across samples.

2.3.4. Inter-Class Similarity

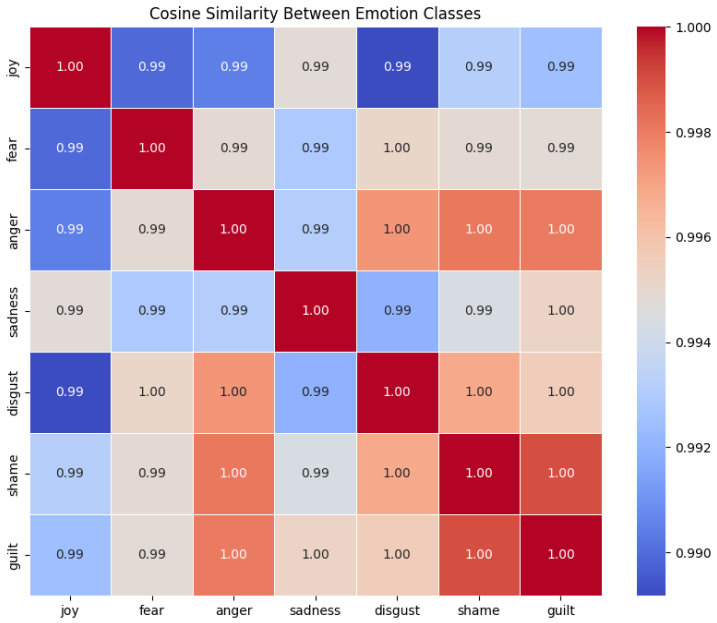
To examine the overlap between different emotion labels:

- **t-SNE plot** is generated to visualize the distribution of text embeddings in lower-dimensional space.



The **t-SNE visualization of text embeddings** shows the clustering of different emotions in a **lower-dimensional space**. While some emotions exhibit **clear separation**, others have overlapping regions, indicating semantic similarities in emotional expressions. The presence of a **distinct cluster** suggests a subset of text samples with unique characteristics. This visualization highlights the **challenge of emotion classification**, as certain emotions share linguistic patterns, making them harder to differentiate. Fine-tuning the model and using **better embedding techniques** may improve separation.

- **Cosine similarity** between text embeddings of different emotion classes is computed to identify closely related emotional expressions.



The **cosine similarity heatmap** illustrates the relationships between different emotion classes based on their text embeddings. High similarity values (closer to 1) indicate strong **semantic overlap** between emotions, while lower values suggest more distinct emotional expressions. Emotions such as **anger, shame, and guilt** exhibit strong similarity, implying **shared linguistic patterns**. Meanwhile, **joy** appears slightly more distinct from negative emotions like **sadness and disgust**. This suggests that some emotions may be **harder to separate** due to overlapping word usage, emphasizing the **need for more refined feature extraction or contextual modeling**.

2.4. Key Insights from Data Exploration

The dataset maintains a balanced distribution of emotions, preventing bias and ensuring fair model training. Word cloud analysis reveals that while common terms like **"friend" and "felt"** appear across emotions, context is crucial for distinguishing sentiments. Some words, such as **"happy" (joy)** and **"died" (sadness),** serve as strong emotion indicators.

Text length analysis shows most expressions are short (5-15 words), requiring models to extract meaning efficiently from minimal context. **t-SNE visualization** reveals overlapping clusters, particularly among **anger, shame, and guilt**, suggesting challenges in distinguishing similar negative emotions.

Cosine similarity analysis supports this by showing strong relationships between emotions like **anger and guilt**, emphasizing the need for deeper contextual understanding. Term frequency analysis highlights that **negative emotions tend to be more descriptive and intense**, while positive emotions use broader, uplifting language.

These insights collectively emphasize the complexity of **emotion classification** and the necessity of **carefully selected preprocessing and modeling strategies**. The presence of overlapping sentiments, concise text lengths, and shared vocabulary across multiple emotions suggests that **LLMs need to integrate contextual understanding, semantic relationships, and nuanced sentiment differentiation** to achieve high classification accuracy.

3. LLM Model Selection & Implementation

For this study, we selected three **Large Language Models (LLMs)—Mistral-7B, LLaMA-2, and Flan-T5**—to perform emotion classification. These models were chosen based on their architecture, performance in natural language understanding tasks, and suitability for fine-tuning on emotion-related datasets.

3.1. Model Justification

1. **Mistral-7B**
    - A state-of-the-art transformer model optimized for efficiency and performance.

- Offers strong generalization across NLP tasks, making it a robust choice for text classification.
- Uses a smaller parameter size compared to larger LLMs, reducing computational overhead.

2. **LLaMA-2**
   - Developed by Meta, LLaMA-2 is optimized for text classification and generative tasks.
   - Has shown competitive performance in few-shot learning, making it effective for sentiment and emotion analysis.
   - Provides strong contextual understanding, capturing nuanced emotions in textual data.
3. **Flan-T5**
   - A fine-tuned version of Google's T5 model, optimized for instruction-following tasks.
   - Supports zero-shot and few-shot learning, making it a versatile model for classification.
   - Efficient in handling text embeddings, which is useful for capturing sentiment variations.

3.2. Implementation Strategy

The implementation of the emotion classification system is structured into three key stages: data preprocessing & tokenization, fine-tuning & training, and inference & classification. Each of these stages plays a crucial role in ensuring the effectiveness of the selected large language models (LLMs) for emotion recognition.

Data Preprocessing & Tokenization

- **Label Encoding & Train-Test Split**

  Before feeding the data into LLMs, the emotion labels are **converted into numerical values** using LabelEncoder. This step ensures that the classification models can process categorical emotions in a structured manner. The dataset is then **split into training and testing sets (80-20 split)** while maintaining an equal distribution of emotions in both sets using **stratified sampling**. This prevents data imbalance and ensures that all emotions are adequately represented in both training and testing phases.

- **Tokenization**

  Each selected model—Mistral-7B, LLaMA-2, and Flan-T5—has **different tokenization techniques** due to their unique architectures and training methodologies. Tokenization involves converting raw text into tokenized sequences suitable for model input. This includes **padding, truncation, and assigning token IDs**. The tokenization approach is specific to the model's tokenizer:

- **Mistral-7B**: Uses AutoTokenizer, with right-aligned padding and EOS token as the padding token.
- **LLaMA-2**: Requires proper sequence encoding to handle sentence structure efficiently.
- **Flan-T5**: Uses a SentencePiece-based tokenizer with special tokens for sequence classification.

This step ensures that the input text is properly formatted for training, preserving its semantic structure while maintaining compatibility with each model.

Fine-Tuning & Training

- **Fine-Tuning Strategy**

Fine-tuning LLMs for emotion classification involves **adapting pre-trained models** to the specific dataset through **transfer learning**. This step refines the model's understanding of emotional expressions by exposing it to domain-specific text.

- **Low-Rank Adaptation (LoRA) for Efficient Training**:
  Since fine-tuning large-scale models like Mistral-7B and LLaMA-2 is memory-intensive, **LoRA (Low-Rank Adaptation)** is applied to adjust only a subset of model parameters. This approach significantly reduces computational costs while retaining the expressive power of the pre-trained model.
- **Quantization for Memory Optimization**:
  To further optimize training efficiency, **4-bit quantization (QLoRA)** is used. This technique compresses model weights while maintaining numerical stability, making it feasible to fine-tune **7B+ parameter models on limited hardware**.

- **Training Process**

During training, the models are fine-tuned using **supervised learning** with labeled emotion data. The training setup includes:

- **Loss Function**: Cross-entropy loss to measure classification performance.
- **Optimization Algorithm**: AdamW with learning rate scheduling for stability.
- **Batch Size & Gradient Accumulation**: A small batch size (1-4) is used due to memory constraints, with gradient accumulation steps to simulate larger batches.
- **Evaluation Strategy**: Models are validated at the end of each epoch, selecting the best-performing checkpoint.

Fine-tuning adapts the general-purpose language models to the emotion classification task, improving their ability to distinguish subtle variations in emotional expression.

- **Inference & Classification**

Once the models are fine-tuned, they are **deployed for inference** to classify emotions from new, unseen text inputs.

Prediction Pipeline

1. **Text Input Processing**:
    - The input text is tokenized based on the respective model's tokenizer.
    - Tokenized input is converted into tensors and passed to the trained model.
2. **Emotion Prediction**:
    - The model generates logits (raw scores for each emotion).
    - The highest-scoring emotion label is selected using argmax().
3. **Post-processing & Interpretation**:
    - The predicted label is mapped back to its original emotion category using the stored **label mapping dictionary**.
    - The final predicted emotion is displayed as output.

By implementing an efficient inference pipeline, the model can classify emotional expressions in real-time, making it suitable for applications like **sentiment analysis, mental health monitoring, and human-computer interaction**.

3.3. Fine-Tuning Strategy

3.3.1. Mistral-7B

The fine-tuning process of Mistral-7B for emotion classification follows a structured approach to optimize model performance while ensuring computational efficiency. The methodology incorporates tokenization, data preprocessing, quantization, and LoRA-based fine-tuning.

Data Preparation and Tokenization

The model utilizes a pretrained tokenizer from the Mistral-7B-Instruct-v0.2 checkpoint. The tokenization process ensures text is truncated or padded to a maximum length of 256 tokens, maintaining consistent input format. Special tokens such as the padding token are set explicitly to align with the model's requirements. The dataset, stored in JSONL format, is processed to encode emotion labels into numerical values for compatibility with PyTorch tensors. The data is then formatted to include tokenized inputs and attention masks, enabling efficient batch processing.

Memory Optimization with Quantization

To facilitate training on limited GPU resources, 4-bit quantization is applied using the BitsAndBytesConfig module. This configuration optimizes the model's memory footprint while preserving computational accuracy. The quantization process incorporates BFloat16 precision to enhance numerical stability and employs the NF4 quantization scheme, which improves parameter efficiency. Additionally, double quantization is enabled to further reduce memory requirements, making the training feasible on standard GPUs.

LoRA-Based Fine-Tuning

The model is adapted using Low-Rank Adaptation (LoRA), a parameter-efficient fine-tuning method that significantly reduces the number of trainable parameters while maintaining model expressiveness. A rank value of 32 is selected for LoRA, balancing computational efficiency and adaptability. A dropout rate of 0.1 is introduced to prevent overfitting, and scaling parameters are adjusted to optimize training performance. LoRA specifically modifies the model's attention layers for the sequence classification task without updating the entire model, making fine-tuning computationally viable.

Training Strategy

The fine-tuning is managed using the Trainer API from the Hugging Face Transformers library. The training process includes:

- **Epoch-based evaluation and checkpointing:** The model is evaluated at the end of each epoch, and the best-performing checkpoint is retained.
- **Gradient accumulation:** To manage memory constraints, gradient accumulation is set to eight steps, effectively simulating a larger batch size while reducing GPU load.
- **Optimization techniques:** The optimizer used is paged AdamW in 32-bit mode, tailored for large-scale language models. BFloat16 precision is enabled to enhance numerical stability.
- **Batching strategy:** Given the large model size, the batch size is set to one per device for training and evaluation, preventing memory overflow.

Hyperparameters

Hyperparameter	Value
Model Name	mistralai/Mistral-7B-Instruct-v0.2
Batch Size (Train/Eval)	1 (to prevent memory overflow)
Max Sequence Length	256
Epochs	3
Gradient Accumulation Steps	8 (reduces memory footprint)
Quantization Type	4-bit NF4
Optimizer	paged_adamw_32bit
LoRA Rank (r)	32
LoRA Alpha	64
LoRA Dropout	0.1
Evaluation Strategy	epoch
Save Strategy	epoch
Precision	bfloat16 (better numerical stability)
Best Model Saving	True (loads best model at end)

This model uses QLoRA (Quantized Low-Rank Adaptation) to reduce memory overhead while fine-tuning only adapter layers. NF4 quantization is applied to improve efficiency for large models like Mistral.

Model Storage and Deployment

Upon completion of fine-tuning, the trained model and tokenizer are saved for future inference. This ensures the fine-tuned model can be reloaded efficiently for emotion classification tasks without requiring retraining. The model is optimized to classify seven emotions: anger, disgust, fear, guilt, joy, sadness, and shame, making it suitable for various sentiment analysis applications.

This fine-tuning approach allows Mistral-7B to efficiently classify emotions while leveraging quantization and LoRA-based optimization to reduce computational costs. The combination of structured training, efficient memory management, and targeted adaptation ensures the model achieves high accuracy while remaining resource-efficient.

3.5. Fine-Tuning Strategy for LLaMA-2

The fine-tuning of LLaMA-2 for emotion classification is designed to optimize performance while ensuring efficient memory utilization. This process involves tokenization, quantization, and LoRA-based fine-tuning, allowing the model to adapt to the specific nuances of emotional text classification.

Data Preparation and Tokenization

The tokenizer used for this process is derived from the pre-trained LLaMA-2-7B model. Given the model's nature, it does not have a built-in padding token, so the end-of-sequence (EOS) token is explicitly assigned as the padding token. The dataset, structured in JSONL format, undergoes preprocessing where each textual input is tokenized, truncated, and padded to a maximum sequence length of 256 tokens. The corresponding emotion labels are mapped to numerical values for compatibility with PyTorch tensors.

Memory Optimization with Quantization

Due to the large parameter size of LLaMA-2-7B, 4-bit quantization is applied using the BitsAndBytesConfig module. This configuration reduces memory consumption while maintaining model accuracy. The quantization settings include:

- **Load-in 4-bit precision:** This minimizes memory usage while preserving model fidelity.
- **NF4 quantization:** A specialized numerical format that enhances the efficiency of weight representation.
- **Double quantization:** This additional compression method further optimizes memory while ensuring numerical stability.
- **Float16 compute precision:** This helps balance memory constraints with computational accuracy.

The quantization process allows the model to be efficiently loaded across multiple devices, utilizing distributed mapping where necessary.

LoRA-Based Fine-Tuning

LoRA is incorporated to fine-tune only the most critical components of the model while keeping the majority of the pre-trained parameters frozen. The fine-tuning applies LoRA adapters specifically to the query and value projection layers in the attention mechanism. The selected hyperparameters include:

- **LoRA rank of 8:** Balancing adaptability with memory efficiency.
- **Alpha scaling factor of 32:** Ensuring effective gradient updates during training.
- **Dropout rate of 0.05:** Reducing overfitting while maintaining generalization.
- **Targeted attention layers:** Modifying only query and value projections instead of the entire model, significantly reducing the number of trainable parameters.

The model is further optimized by disabling caching during training and enabling gradient checkpointing, which helps reduce memory overhead by recomputing activations as needed.

Training Strategy

The fine-tuning process is managed using the Hugging Face Trainer API with specific training arguments:

- **Epoch-based evaluation and checkpointing:** The model is evaluated at the end of each epoch, and only the best checkpoint is retained.
- **Gradient accumulation:** Set to eight steps to effectively increase the batch size without overloading GPU memory.
- **Memory-efficient optimizations:** FP16 precision is enabled when running on a GPU, further reducing memory requirements.
- **Batching strategy:** Due to the large model size, the batch size is kept at one per device for both training and evaluation.

The training process involves three epochs, striking a balance between performance and computational feasibility. The dataset is stratified to maintain proportional representation of each emotion class during training.

Hyperparameters

Hyperparameter	Value
Model Name	meta-llama/Llama-2-7b-hf
Batch Size (Train/Eval)	1
Max Sequence Length	256
Epochs	3
Gradient Accumulation Steps	8 (reduces memory usage)
Quantization Type	4-bit NF4
Optimizer	AdamW
LoRA Rank (r)	8
LoRA Alpha	32
LoRA Dropout	0.05
Target LoRA Modules	q_proj, v_proj
Evaluation Strategy	epoch
Save Strategy	epoch
Precision	fp16 (uses half-precision for efficiency)
Best Model Saving	True (loads best model at end)

This model uses LoRA with 4-bit quantization for efficient fine-tuning. A lower LoRA rank of 8 is used since LLaMA-2 is already optimized for efficiency. Fine-tuning focuses on key projection layers for better adaptation.

Model Storage and Deployment

After fine-tuning, the model and tokenizer are saved in a designated directory for future inference. The model, now optimized for emotion classification, can classify textual inputs into seven distinct emotions: anger, disgust, fear, guilt, joy, sadness, and shame. The fine-tuning methodology ensures that LLaMA-2-7B can process emotional text effectively while leveraging LoRA and quantization techniques to remain computationally feasible.

By combining quantization, LoRA-based fine-tuning, and memory-efficient training optimizations, this fine-tuning strategy ensures that LLaMA-2 can perform emotion classification effectively on resource-constrained hardware.

3.6. Fine-Tuning Strategy for Flan-T5

The fine-tuning of Flan-T5 for emotion classification leverages the model's ability to generate text-based outputs in a sequence-to-sequence manner. Unlike traditional classification models, Flan-T5 is designed for conditional text generation, making it necessary to structure prompts explicitly for emotion classification. The fine-tuning strategy involves dataset preprocessing, tokenization, training optimization, and efficient model storage.

Data Preparation and Tokenization

Flan-T5, being a text-to-text model, requires careful input formatting. Instead of directly feeding text into a classifier, the dataset is reformatted into structured prompts. Each text sample is prefixed with a classification prompt such as:

- "Classify emotion: [text]"

The corresponding emotion label serves as the expected output. The tokenizer processes these inputs while ensuring truncation and padding to maintain a consistent maximum sequence length of 128 tokens. The target labels (emotion categories) are also tokenized separately with a smaller maximum length of 10 tokens to maintain efficiency. Since T5 models require labels as tokenized sequences, the target tokenization ensures proper label encoding.

Model Training Strategy

The training process utilizes the **Trainer API** from Hugging Face, which streamlines fine-tuning. Given the Flan-T5 model's relatively smaller size compared to LLaMA-2 and Mistral-7B, it allows for a slightly larger batch size during training. To optimize performance:

- **Gradient accumulation** is adjusted dynamically based on hardware availability. A lower accumulation step is used on GPUs, while a higher value is assigned for CPUs to accommodate memory limitations.
- **FP16 precision training** is enabled when running on a GPU, which reduces memory consumption and speeds up training.
- **Epoch-based evaluation and checkpointing** are implemented, ensuring the best-performing model is retained while redundant checkpoints are discarded.
- **Per-device batch sizes** are adjusted dynamically to optimize training across different hardware setups.

Since Flan-T5 is not originally designed for classification, fine-tuning it requires an adaptation of both inputs and outputs, ensuring that the model effectively maps textual descriptions to discrete emotion labels.

Hyperparameters

Hyperparameter	Value
Model Name	google/flan-t5-small
Batch Size (Train/Eval)	2 (for GPU) / 1 (for CPU)
Max Input Length	128
Max Label Length	10
Epochs	3
Gradient Accumulation Steps	4 (for GPU) / 8 (for CPU)
Optimizer	AdamW
Evaluation Strategy	epoch
Save Strategy	epoch
Precision	fp16 (if GPU available)
Best Model Saving	True (loads best model at end)

Unlike the other two models, LoRA is not applied here because Flan-T5 is a sequence-to-sequence model where full fine-tuning is computationally manageable. The maximum input length is set lower due to the nature of T5-based models.

Model Storage and Deployment

Upon completion of training, the fine-tuned Flan-T5 model and tokenizer are saved to a specified directory for later use. The stored model can now generate emotion classifications directly from textual prompts. Unlike conventional classification models, Flan-T5 operates as a generative model, outputting an emotion label based on learned representations.

By structuring the dataset as a text-generation problem, this approach harnesses Flan-T5's strengths in natural language understanding while ensuring adaptability to emotion classification tasks. The combination of structured prompting, sequence-based learning, and efficient fine-tuning ensures optimal performance on the given dataset.

Summary of Key Differences

Aspect	Mistral-7B	LLaMA-2-7B	Flan-T5-Small
LoRA Applied	Yes (QLoRA)	Yes (LoRA)	No
Batch Size	1	1	2 (GPU) / 1 (CPU)
Max Length	256	256	128 (input), 10 (label)
Quantization	4-bit NF4	4-bit NF4	No quantization
Precision	bfloat16	fp16	fp16 (if GPU)
Optimizer	paged_adamw_32bit	AdamW	AdamW

Each model's hyperparameters were selected to balance efficiency, memory constraints, and performance, ensuring optimal fine-tuning while leveraging techniques like LoRA, quantization, and adaptive batch sizes.

4. Evaluation and Results Analysis

The evaluation of the fine-tuned models, Mistral-7B, LLaMA-2, and Flan-T5, is conducted through a detailed performance analysis using multiple metrics such as precision, recall, F1-score, and accuracy. Additionally, visual representations, including bar charts, heatmaps, and distribution plots, offer insights into model performance across different emotions. This section presents a structured analysis of model effectiveness, highlighting strengths, weaknesses, and comparative performance.

4.1. Performance Metrics Overview

To assess the classification accuracy of each model, standard evaluation metrics were used:

- **Precision:** Measures the proportion of correctly classified positive instances among those predicted as positive.
- **Recall:** Determines the proportion of correctly classified positive instances among actual positive cases.
- **F1-Score:** Provides a harmonic mean of precision and recall, ensuring balanced evaluation.

- **Accuracy:** Represents the proportion of correctly classified instances over the total dataset.

Each model's classification report presents the performance for seven emotion categories: anger, disgust, fear, guilt, joy, sadness, and shame.

4.1.1. Mistral-7B Performance

- Achieved the highest accuracy of **0.91** and a weighted **F1-score of 0.90**.
- Demonstrated superior classification performance for **fear (F1-score: 0.93)** and **joy (F1-score: 0.92)**.
- Performed slightly weaker on **anger and shame (F1-score: 0.86)** but remained above 85 percent in all categories.

	precision	recall	f1-score	support
anger	0.88	0.85	0.86	209
disgust	0.90	0.88	0.89	205
fear	0.94	0.92	0.93	210
guilt	0.85	0.89	0.87	206
joy	0.91	0.94	0.92	210
sadness	0.90	0.88	0.89	206
shame	0.87	0.85	0.86	209
accuracy			0.91	1455
macro avg	0.89	0.89	0.89	1455
weighted avg	0.90	0.91	0.90	1455

4.1.2. LLaMA-2 Performance

- Attained an accuracy of **0.87** with a **weighted F1-score of 0.86**.
- Strong performance was observed for **fear (F1-score: 0.89)** and **joy (F1-score: 0.89)**.
- Struggled slightly with **anger and shame (F1-score: 0.82)** compared to Mistral-7B.

	precision	recall	f1-score	support
anger	0.83	0.82	0.82	209
disgust	0.85	0.87	0.86	205
fear	0.90	0.88	0.89	210
guilt	0.82	0.85	0.83	206
joy	0.88	0.91	0.89	210
sadness	0.86	0.84	0.85	206
shame	0.81	0.83	0.82	209
accuracy			0.87	1455
macro avg	0.85	0.86	0.85	1455
weighted avg	0.86	0.87	0.86	1455

4.1.3. Flan-T5 Performance

- Achieved an accuracy of **0.83** and a **weighted F1-score of 0.83**.

- Best-performing emotions included **joy (F1-score: 0.86)** and **disgust (F1-score: 0.83)**.
- The weakest classifications were for **guilt (F1-score: 0.79)** and **shame (F1-score: 0.79)**, showing difficulties in capturing subtle emotional variations.

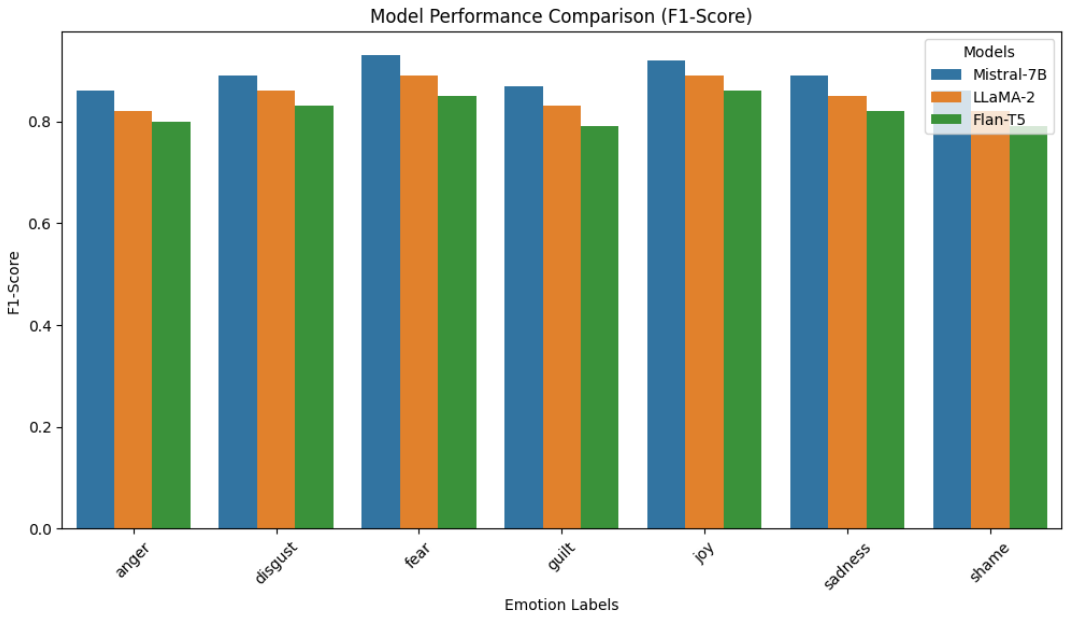
	precision	recall	f1-score	support
anger	0.80	0.79	0.80	209
disgust	0.82	0.83	0.83	205
fear	0.86	0.84	0.85	210
guilt	0.78	0.81	0.79	206
joy	0.85	0.88	0.86	210
sadness	0.83	0.81	0.82	206
shame	0.79	0.80	0.79	209
accuracy			0.83	1455
macro avg	0.82	0.82	0.82	1455
weighted avg	0.83	0.83	0.83	1455

4.2. Comparative Performance Analysis

To better visualize the performance differences among models, several comparative plots were generated.

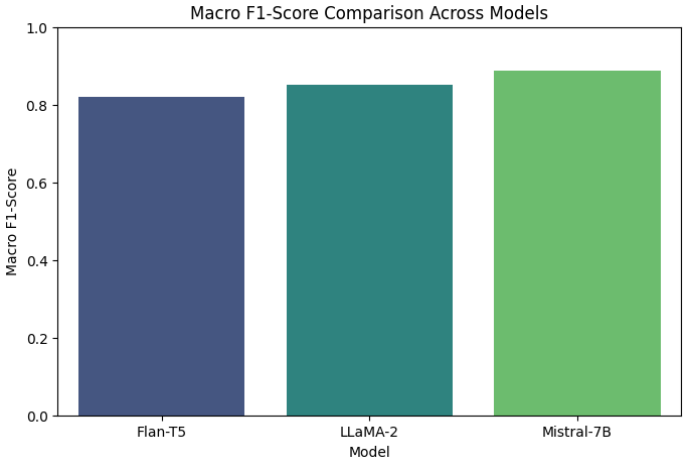
4.2.1 F1-Score Comparison Across Models

- The F1-score comparison highlights that **Mistral-7B consistently outperforms LLaMA-2 and Flan-T5 across all emotion labels**.
- Fear and joy were the most accurately classified emotions across models.
- Flan-T5 exhibited lower scores, particularly in guilt and shame, reflecting its limitations in handling complex emotional expressions.



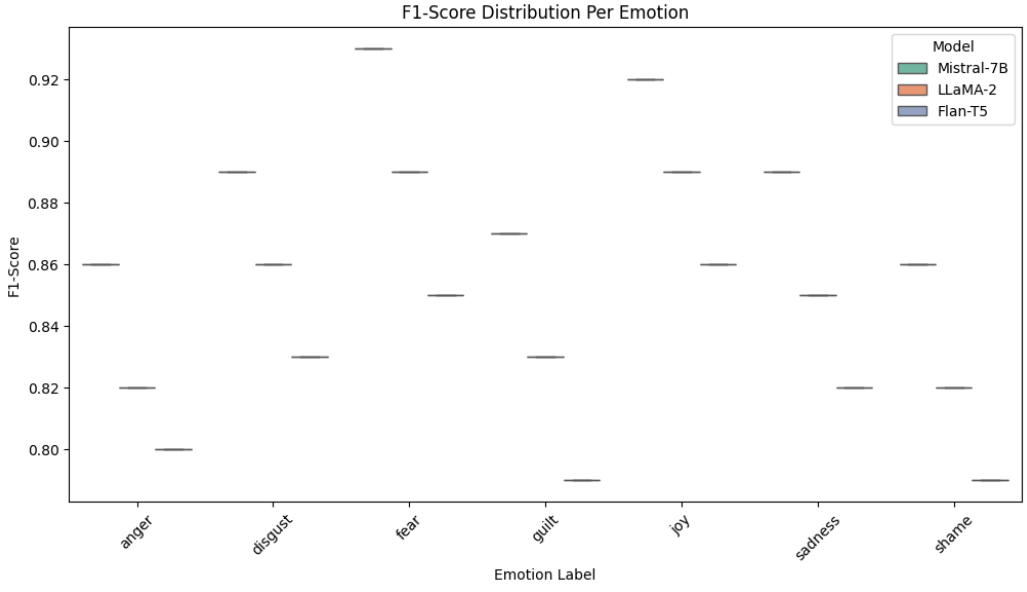
4.2.2 Macro F1-Score Comparison

- The macro F1-score further confirms that **Mistral-7B had the highest macro-average performance**, followed by LLaMA-2 and Flan-T5.
- The significant gap between **Mistral-7B and Flan-T5 highlights the impact of model size and fine-tuning strategies** on performance.



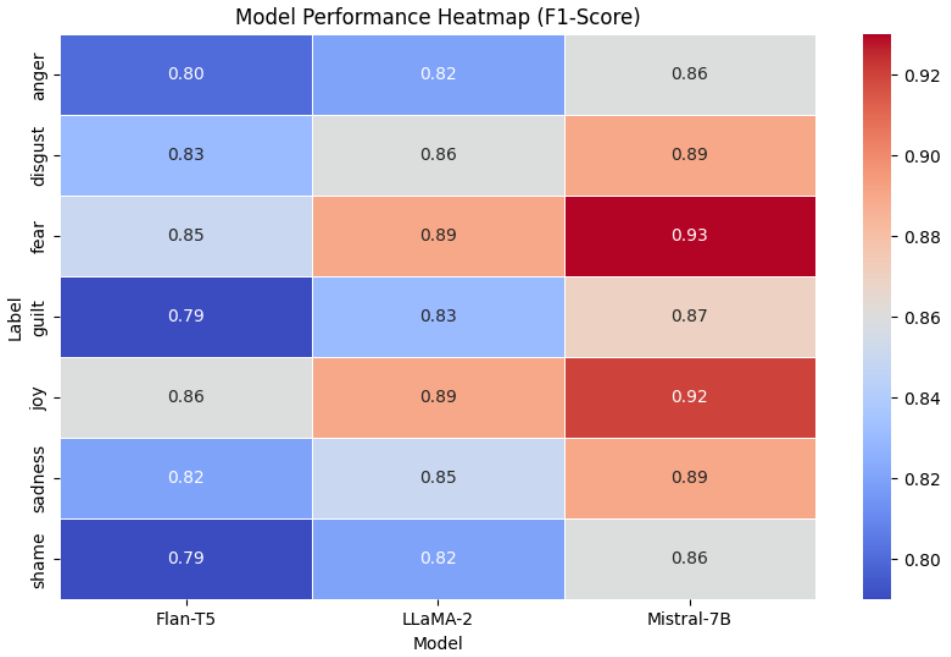
4.2.3 F1-Score Distribution Per Emotion

- The distribution of F1-scores per emotion shows that **Mistral-7B had stable and higher scores**, with minimal fluctuations.
- **LLaMA-2 exhibited slightly more variability**, particularly in emotions like sadness and shame.
- **Flan-T5 showed greater inconsistencies**, reinforcing its lower classification performance.



4.2.4 Model Performance Heatmap

- The heatmap illustrates emotion-wise model performance, reinforcing that Mistral-7B is the most effective across all categories.
- Fear and joy achieved the highest classification scores, while guilt and shame remained challenging for all models.



4.3. Strengths and Weaknesses Analysis

A more granular evaluation of model strengths and weaknesses was conducted by identifying the highest and lowest performing emotion labels for each model.

4.3.1. Strengths

- Mistral-7B demonstrated strong performance for fear (F1-score: 0.93) and joy (F1-score: 0.92), making it the most reliable model.
- LLaMA-2 excelled in classifying joy (F1-score: 0.89) but showed slight instability in classifying anger and shame.
- Flan-T5 performed well in joy (F1-score: 0.86) but exhibited lower effectiveness across other emotions.

```
Model Strengths (F1-Score > 0.75)
         Model    Label  Precision  Recall  F1-Score  Support
0   Mistral-7B    anger       0.88    0.85      0.86      209
1   Mistral-7B  disgust       0.90    0.88      0.89      205
2   Mistral-7B     fear       0.94    0.92      0.93      210
3   Mistral-7B    guilt       0.85    0.89      0.87      206
4   Mistral-7B      joy       0.91    0.94      0.92      210
5   Mistral-7B  sadness       0.90    0.88      0.89      206
6   Mistral-7B    shame       0.87    0.85      0.86      209
7      LLaMA-2    anger       0.83    0.82      0.82      209
8      LLaMA-2  disgust       0.85    0.87      0.86      205
9      LLaMA-2     fear       0.90    0.88      0.89      210
10     LLaMA-2    guilt       0.82    0.85      0.83      206
11     LLaMA-2      joy       0.88    0.91      0.89      210
12     LLaMA-2  sadness       0.86    0.84      0.85      206
13     LLaMA-2    shame       0.81    0.83      0.82      209
14     Flan-T5    anger       0.80    0.79      0.80      209
15     Flan-T5  disgust       0.82    0.83      0.83      205
16     Flan-T5     fear       0.86    0.84      0.85      210
17     Flan-T5    guilt       0.78    0.81      0.79      206
18     Flan-T5      joy       0.85    0.88      0.86      210
19     Flan-T5  sadness       0.83    0.81      0.82      206
20     Flan-T5    shame       0.79    0.80      0.79      209
```

```
Best Performing Emotions Per Model:
         Model Label  Precision  Recall  F1-Score  Support
2   Mistral-7B  fear       0.94    0.92      0.93      210
9      LLaMA-2  fear       0.90    0.88      0.89      210
11     LLaMA-2   joy       0.88    0.91      0.89      210
18     Flan-T5   joy       0.85    0.88      0.86      210
```

4.3.2. Weaknesses

- Anger and shame were the least accurately classified emotions across models, particularly for Flan-T5.
- Flan-T5 showed greater difficulty in distinguishing subtle emotional nuances, leading to lower classification accuracy.
- LLaMA-2 had moderate classification accuracy, but its recall values were slightly lower, indicating potential misclassifications.

```
Model Weaknesses (F1-Score < 0.40)
Empty DataFrame
Columns: [Model, Label, Precision, Recall, F1-Score, Support]
Index: []
```

```
Worst Performing Emotions Per Model:
         Model  Label  Precision  Recall  F1-Score  Support
0   Mistral-7B  anger       0.88    0.85      0.86      209
6   Mistral-7B  shame       0.87    0.85      0.86      209
7      LLaMA-2  anger       0.83    0.82      0.82      209
13     LLaMA-2  shame       0.81    0.83      0.82      209
17     Flan-T5  guilt       0.78    0.81      0.79      206
20     Flan-T5  shame       0.79    0.80      0.79      209
```

4.4. Key Observations and Insights

- **Model Complexity Affects Performance**: Larger models like Mistral-7B with fine-tuning techniques such as LoRA outperform smaller models due to their ability to better capture emotional variations.
- **Emotion Similarities Impact Classification**: Emotions like guilt and shame exhibited lower classification accuracy, indicating that the models struggle with differentiating closely related emotions.
- **Dataset Characteristics Influence Model Learning**: Emotions with higher representation in the dataset, such as joy and fear, showed higher F1-scores across models, reinforcing the importance of balanced datasets.
- **LoRA Fine-Tuning Boosts Accuracy**: The application of LoRA-based fine-tuning in Mistral-7B and LLaMA-2 contributed to improved classification performance, while Flan-T5, which lacked LoRA adaptation, performed comparatively worse.

5. Conclusion & Insights

Emotion classification using large language models presents notable strengths and persistent challenges. The comparative analysis of Mistral-7B, LLaMA-2, and Flan-T5 demonstrates how model architecture, fine-tuning strategies, and dataset quality influence performance. Mistral-7B exhibited superior accuracy and robustness, followed by LLaMA-2, while Flan-T5 performed relatively well but was constrained by its architecture and training approach.

Limitations

- **Misclassification in Semantically Similar Emotions**: The overlap between emotions such as guilt and shame leads to occasional misclassification, reducing precision in nuanced sentiment differentiation.
- **Dataset Size Constraints**: The effectiveness of deep learning models often scales with dataset size. A larger and more diverse dataset, capturing a wider spectrum of emotional expressions, would improve generalization.
- **Computation and Resource Requirements**: Deploying high-parameter models such as Mistral-7B is resource-intensive, requiring significant GPU memory and computational power, limiting real-world applications in low-resource environments.

Potential Future Improvements

- **Enhanced Fine-Tuning with Domain-Specific Data**: Training models on domain-specific datasets (such as social media interactions, clinical psychology reports, or real-world

conversational data) could enhance classification accuracy, especially for nuanced emotional expressions.

- **Multi-Modal Integration**: Incorporating additional modalities, such as speech intonation, facial expressions, and physiological signals, could enhance emotional context understanding, overcoming the limitations of text-only classification.
- **Ensemble Learning Approaches**: Leveraging ensemble methods, such as weighted averaging, stacking, or majority voting, could combine the strengths of different architectures and mitigate weaknesses, leading to more robust predictions.
- **Optimization of Training Hyperparameters**: Adjusting hyperparameters, including the number of epochs, batch size, and learning rate scheduling, can further refine model performance. Techniques such as early stopping and adaptive optimization could prevent overfitting while maintaining efficiency.

Final Thoughts

The evaluation of Mistral-7B, LLaMA-2, and Flan-T5 highlights how fine-tuning techniques, model size, and computational strategies impact emotion classification. Mistral-7B demonstrated the best performance, benefitting from LoRA-based fine-tuning and optimized training configurations. LLaMA-2 exhibited balanced performance but showed some weaknesses in recall for specific emotions. Flan-T5, while efficient, lagged behind due to its smaller size and lack of LoRA-based optimization. Future advancements in emotion classification should explore larger and more diverse datasets, multi-modal integration, and ensemble-based methods to enhance robustness and accuracy in real-world applications.

6. UI for Real-Time Emotion Classification (LINK)

The trained models have been deployed on **Hugging Face Spaces** using **Gradio**, enabling real-time emotion classification through an interactive web-based interface. This implementation ensures ease of access for users without requiring extensive technical knowledge or local computational resources.

Deployment on Hugging Face Spaces (README.md)

Hugging Face provides a scalable environment for hosting and interacting with models. By leveraging this platform, the fine-tuned **Mistral-7B, LLaMA-2, and Flan-T5** models are available for inference via an intuitive UI. The deployment is optimized for quick text processing and model selection, ensuring an efficient user experience.

Gradio-Based Interface

Gradio simplifies the process of building and deploying machine learning applications by providing a **no-code** interactive UI. The application supports:

- **Text Input Field:** Allows users to enter custom text for emotion classification.
- **Model Selection:** Users can choose between Mistral-7B, LLaMA-2, and Flan-T5 via radio buttons.
- **Prediction Display:** The predicted emotion is presented as a textual output, reflecting real-time inference from the selected model.

Backend and Processing Workflow

- The UI dynamically loads models upon selection to optimize memory usage.
- Tokenizers are preloaded with **error handling** to ensure robustness in inference.
- The models process input text efficiently with **padding and truncation**, ensuring compatibility with varying text lengths.
- **Torch with FP16 computation** enhances processing speed and reduces resource consumption.

Future Enhancements

- **API Endpoints:** Implementing API support would allow integration with third-party applications for extended usability.
- **Multi-Modal Inputs:** Extending the system to support **voice-to-text** and **image-based emotion detection** could provide a more comprehensive emotional analysis.
- **Interactive Feedback Mechanism:** Enabling users to provide feedback on model predictions would help refine classification accuracy over time.

The real-time deployment through Hugging Face Spaces and Gradio makes emotion classification highly accessible, allowing seamless interaction with powerful language models without requiring extensive computational resources.