Capstone Project Report - LoRA Optimization for Sentiment Analysis

1. Student Information

Student Name: Akhila Joseph
Student ID: 223026978
Date Submitted: 16-05-2025

2. Project Introduction

Title of the Project: Sentiment Analysis of Drug Reviews using LoRA Optimization

Objective:

The objective of this project was to implement and optimize Low-Rank Adaptation (LoRA) in the Flan-T5 model for sentiment classification of drug reviews. By leveraging LoRA, the model's training parameters were significantly reduced, allowing for more efficient fine-tuning without compromising predictive accuracy. The project also aimed to enhance the model's ability to detect nuanced sentiment in user reviews by experimenting with various hyperparameter configurations and target modules.

Significance and Relevance:

The pharmaceutical domain is characterized by a wealth of patient reviews and feedback on medication effectiveness and side effects. Accurate sentiment analysis of such data can provide critical insights for healthcare providers, researchers, and policymakers. However, training large models for sentiment classification can be computationally intensive. This project addresses this challenge by incorporating LoRA, a parameter-efficient technique that facilitates effective fine-tuning with minimal computational resources.

3. Environment Setup

- Development Platform: Google Colab
- GPU Availability: Yes (Tesla T4)
- Python Version: 3.10
- Libraries and Dependencies:
  - transformers (v4.30) - for model architecture and training
  - peft (v0.3.0) - for implementing LoRA
  - datasets (v2.12.1) - for data management and processing
  - scikit-learn (v1.2.2) - for evaluation metrics
  - gradio (v3.1.4) - for building an interactive user interface

4. Implementation Approach and Key Methods:

- The LoRA implementation leveraged the peft library, specifically using LoraConfig to define low-rank parameters and get_peft_model() to integrate these into the Flan-T5 model.
- Specific target modules chosen for LoRA fine-tuning included "q", "v", and "o", representing query, value, and output projections in the attention mechanism.

5. LLM Setup

- Model Name: google/flan-t5-base
- Model Architecture: T5 Transformer with Encoder-Decoder architecture

```
[ ]  # Load model and tokenizer
     model_name = "google/flan-t5-base"
     tokenizer = AutoTokenizer.from_pretrained(model_name)
     model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
```

- Provider: Hugging Face
- Model Purpose: Sentiment classification (positive, neutral, negative)
- Training Strategy:
  - Base model training with original dataset.
  - LoRA fine-tuning with targeted modules and hyperparameter tuning.
  - Model evaluation using accuracy, precision, recall, F1-score, and confusion matrix.

6. Dataset Description

- Dataset Name: Drug Review Dataset [3]
- Source: DrugLib.com (local processing and storage)
- Number of Samples: 1,000+:
  - Positive: 648 samples
  - Negative: 234 samples
  - Neutral: 125 samples
- Features:
  - benefitsReview: Description of medication benefits
  - sideEffectsReview: Description of side effects experienced
  - commentsReview: Overall user comments
  - rating: Numerical rating (1-10)
- Preprocessing Steps:
  - Combined review fields (benefits, side effects, comments) into a unified input text.
  - Applied sentiment mapping based on rating [1]:
    - Ratings ≥ 7 → positive
    - Ratings ≤ 4 → negative
    - Ratings 5-6 → neutral

```python
# Updated sentiment mapping
def map_rating_to_sentiment(rating):
    if rating >= 7:
        return "positive"
    elif rating <= 4:
        return "negative"
    else:
        return "neutral"

train_df["sentiment"] = train_df["rating"].apply(map_rating_to_sentiment)
test_df["sentiment"] = test_df["rating"].apply(map_rating_to_sentiment)
```

- Post-Oversampling Distribution:
  - After oversampling, each class was adjusted to 800 samples, creating a more balanced training dataset.
  - The class distribution can be visualized using a bar chart for clarity:

7. Challenges and Mitigations

- Class Imbalance:
  - The dataset initially had an underrepresented neutral class. Oversampling increased the neutral class size but led to overfitting in certain configurations.
  - Mitigation: Implemented oversampling and conducted extensive hyperparameter tuning to balance class representation.
- Memory Constraints:
  - Training the model with larger datasets and higher rank (r=16) led to GPU memory overload.
  - Mitigation: Applied torch.cuda.empty_cache() and adjusted batch sizes to manage memory usage effectively.

```
[ ]  #clear Memory cache
     import torch
     torch.cuda.empty_cache()
     print("GPU memory cleared.")
```

```
GPU memory cleared.
```

- Neutral Sentiment Misclassification:
  - The model struggled to differentiate between neutral and slightly positive/negative reviews.
  - Mitigation: Further analysis and targeted augmentation of neutral samples were considered.

8. Model Versions and Experimentation

To iteratively improve performance and address specific weaknesses (such as class imbalance and neutral sentiment misclassification), multiple model versions were developed and evaluated. Each version introduced changes in data strategy, model configuration, or fine-tuning approach.

a. Baseline Evaluation with Zero-Shot Inference

Before fine-tuning, I evaluated the zero-shot performance of the google/flan-t5-base model on raw drug review inputs using prompt-based classification.

Prompts such as:

```
prompt = f"Classify the sentiment as positive, negative, or neutral:\n{text}"
```

were passed directly to the model.

While the model handled clearly positive or negative cases well, it showed inconsistencies in neutral predictions. This provided a baseline understanding of the model's out-of-the-box capabilities and justified the need for fine-tuning using LoRA.

```
Review 1
• Benefits: It reduced my anxiety within a few days.
  ↳ Sentiment: positive
• Side Effects: But I constantly felt sleepy and had trouble concentrating.
  ↳ Sentiment: negative
• Comments: I'm not sure if I'll continue using it long-term.
  ↳ Sentiment: negative

Predicted Overall Sentiment: negative

Review 2
• Benefits: The medication significantly improved my focus and energy.
  ↳ Sentiment: positive
• Side Effects: Minor dry mouth, but it went away after a week.
  ↳ Sentiment: negative
• Comments: I'm really happy with the results.
  ↳ Sentiment: positive

Predicted Overall Sentiment: positive
```

b. Hyperparameter Tuning Trials (Fine-Tuned Base Model):

As part of early experimentation, three different configurations were tested to identify the most effective learning rate, batch size, and number of epochs for fine-tuning the Flan-T5 base model:

a. Trial 1: lr = 5e-5, batch_size = 8, epochs = 3

- Training loss dropped from 7.65 → 0.14
- Accuracy: 71.90%
- Macro F1-score: 54.60%
- Strong performance on positive (F1: 83.11%) and negative (F1: 69.05%)
- Very weak on neutral (F1: 11.65%), indicating underfitting for the minority class

b. Trial 2: lr = 3e-4, batch_size = 8, epochs = 4

- Training loss dropped from 2.04 → 0.06
- Accuracy: 75.87% (highest across all trials)
- Macro F1-score: 55.76%
- Good balance of training efficiency and generalization
- Slight improvement in neutral recall, though still low (6.4%)

c. Trial 3: lr = 1e-4, batch_size = 4, epochs = 5

- Training loss plateaued around 0.09
- Accuracy: 69.81%
- Macro F1-score: 55.96% (slightly better than Trial 2)
- Balanced precision/recall for neutral (17.6%), but lower accuracy

- Trial 2 was chosen as the best-performing setup due to its highest accuracy (75.87%), fast convergence, and strong results for positive/negative classe
- This configuration was later used as a baseline for LoRA fine-tuning.

```
Final Hyperparameter Tuning Results:
        lr  batch_size  epochs  accuracy  precision    recall  f1_score
2  0.00010           4       5  0.698113   0.549454  0.587355  0.559604
1  0.00030           8       4  0.758689   0.556676  0.591332  0.557601
0  0.00005           8       3  0.718967   0.535185  0.573904  0.546029
```

c. LoRA Fine-Tuned Model (312 Samples per Class):

Following the baseline model evaluation and hyperparameter tuning, LoRA (Low-Rank Adaptation) was applied to improve efficiency during fine-tuning. A balanced dataset with 312 samples per sentiment class was used in this initial experiment.

- LoRA Configuration:
  - r = 8, alpha = 16, dropout = 0.1
  - Target modules: "q" and "v" (query and value projections)

```
base_model = T5ForConditionalGeneration.from_pretrained("google/flan-t5-base")
lora_config = LoraConfig(
    r=8,
    lora_alpha=16,
    lora_dropout=0.05,
    bias="none",
    target_modules=["q", "v"],
    task_type=TaskType.SEQ_2_SEQ_LM
)
```

- Training Efficiency:
  - LoRA significantly reduced the number of trainable parameters, improving memory efficiency while preserving performance.
  - Trained in fewer steps with lower computational cost compared to full fine-tuning.

```
Classification Report:
              precision    recall  f1-score   support

    negative     0.5556    0.9188    0.6924       234
     neutral     0.2692    0.1120    0.1582       125
    positive     0.9173    0.8040    0.8569       648

    accuracy                         0.7448      1007
   macro avg     0.5807    0.6116    0.5692      1007
weighted avg     0.7528    0.7448    0.7320      1007


Confusion Matrix:
[[521  32  95]
 [ 34  14  77]
 [ 13   6 215]]

Evaluation Metrics Summary:
Accuracy  : 0.7448
Precision : 0.5807
Recall    : 0.6116
F1 Score  : 0.5692
```

  - Excellent recall on negative class due to LoRA's focused adaptation
  - Model still biased toward positive/negative; neutral detection remained weak
  - Provided a strong foundation for further improvements using data oversampling and advanced LoRA configurations

d. LoRA Fine-Tuned Model (800 Samples per Class):

To address the limitations seen in neutral class detection, the training data was oversampled to 800 samples per class. This created a fully balanced dataset for fine-tuning, helping the model better generalize across all sentiment categories.

- LoRA Configuration:
  - r = 8, alpha = 16, dropout = 0.1

- Target modules: "q" and "v" (consistent with the 312-sample run)

```
# LoRA config
lora_config = LoraConfig(
    r=8,
    lora_alpha=16,
    lora_dropout=0.05,
    bias="none",
    target_modules=["q", "v"],
    task_type=TaskType.SEQ_2_SEQ_LM
)
```

- Training Objective:
  - Improve model fairness and class balance, especially for the underrepresented neutral class
  - Use LoRA's parameter efficiency to scale training without increasing model size

```
Classification Report:
              precision    recall  f1-score   support

    negative     0.6506    0.8675    0.7436       234
     neutral     0.2533    0.3040    0.2764       125
    positive     0.9339    0.7855    0.8533       648

    accuracy                         0.7448      1007
   macro avg     0.6126    0.6523    0.6244      1007
weighted avg     0.7836    0.7448    0.7562      1007

Confusion Matrix:
[[509  89  50]
 [ 28  38  59]
 [  8  23 203]]

Accuracy  : 0.7448
Precision : 0.6126
Recall    : 0.6523
F1 Score  : 0.6244
```

  - Oversampling helped the model generalize better across all classes
  - Neutral predictions improved significantly but remained the weakest class
  - LoRA proved scalable and effective even with a larger dataset

e. best LoRA model configuration:

After testing several combinations of hyperparameters, the best-performing configuration for LoRA fine-tuning was identified based on both accuracy and macro F1-score.

- Multiple configurations were tested, varying r (rank), alpha (scaling factor), and dropout values.
- The best configuration identified:
  - r=16 - Moderate rank, balancing parameter reduction and representation power.
  - alpha=64 - Increased scaling to enhance feature adaptation.
  - dropout=0.1 - Prevents overfitting, maintains generalization.
- Performance: Achieved an accuracy of 76.37% and a macro F1-score of 66.30%.

```
LoRA Hyperparameter Tuning Summary:
   trial   r  alpha  dropout  avg_loss
3      4  16     64     0.10    0.3164
2      3  16     32     0.05    0.3418
1      2   8     32     0.10    0.3622
0      1   8     16     0.05    0.3813
```

f. Advanced LoRA Configuration:

- Implemented targeted LoRA on specific attention submodules:
  - "q" (Query) - Responsible for context query generation.
  - "v" (Value) - Represents the attention values in each layer.
  - "o" (Output) - Integrates query and value representations.
- Result: Over 99% reduction in trainable parameters, significantly decreasing memory usage while maintaining classification accuracy.

9. Model Comparison and Evaluation:

- Models were systematically evaluated across key metrics:

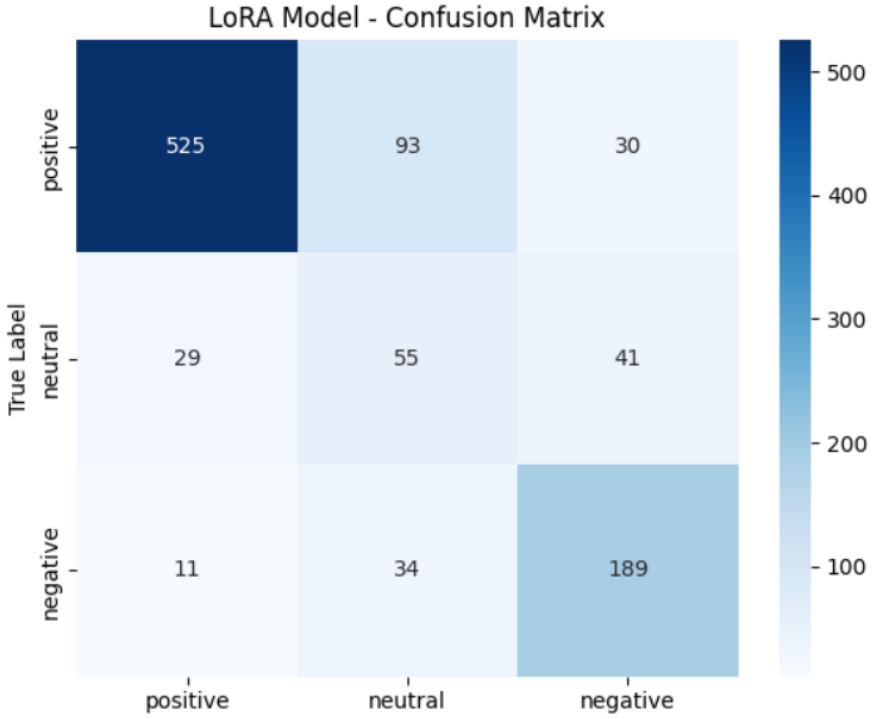
LoRA Variants and Base Model Evaluation

	Stage	accuracy	precision	recall	f1_score	weighted_f1
0	Base Model	0.724926	0.574428	0.608032	0.584581	0.731643
1	LoRA with 300 labels	0.744786	0.580680	0.611605	0.569177	0.731956
2	LoRA with 800 labels	0.744786	0.612640	0.652338	0.624421	0.756198
3	LoRA with Best Config	0.763654	0.652775	0.685959	0.663037	0.779309
4	LoRA with Target Modules	0.755467	0.613623	0.632589	0.621664	0.757763

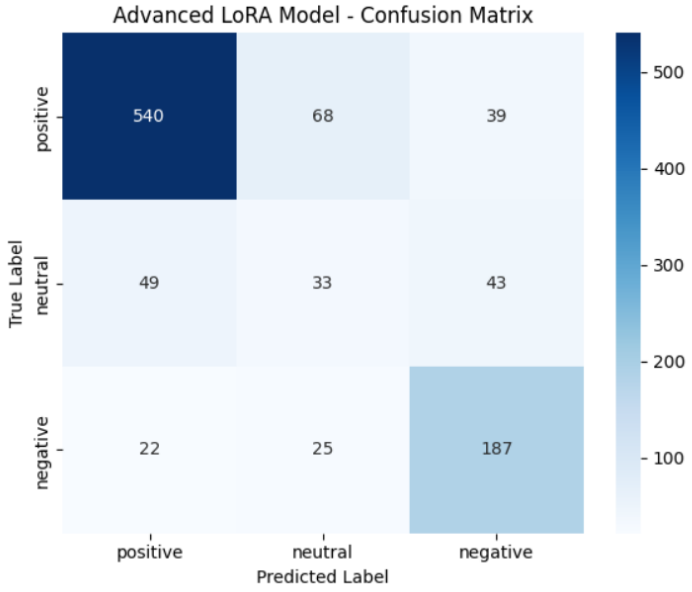
7. Benchmarking & Evaluation

- Evaluation Metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix
- Why These Metrics?
  - Accuracy: Measures overall correct predictions.
  - Precision: Evaluates false positives (relevant in neutral class misclassification).
  - Recall: Assesses sensitivity, especially in detecting negative reviews.
  - F1-score: Balances precision and recall, crucial for imbalanced datasets.

- Results Interpretation:
  - The best LoRA configuration demonstrated consistent performance in positive and negative classes but showed a slight reduction in neutral class accuracy due to limited sample diversity.



  - Advanced LoRA targeting ("q", "v", "o") provided comparable accuracy but with significantly reduced training parameters, highlighting its efficiency for real-world deployment.



10. Evaluation Insights and Recommendations

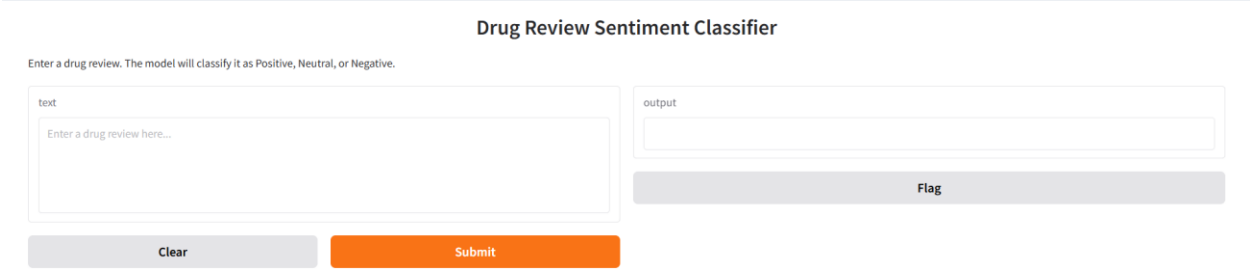
- Positive Class Performance:
  - Consistently high F1-scores across all configurations indicate strong model learning for clearly positive reviews.
- Neutral Class Analysis:
  - Despite being oversampled, the neutral class F1-score remained relatively low, suggesting that more nuanced examples may be necessary to enhance model differentiation.
  - Recommendations: Introduce targeted data augmentation for neutral samples and include additional context features (e.g., review length, sentiment intensity).
- Impact of Target Module Selection:
  - Focusing LoRA on "q", "v", and "o" layers achieved over 99% parameter reduction while maintaining competitive accuracy.
  - Future Considerations: Evaluate the impact of incorporating "k" and attention bias parameters for further optimization.

9. UI Integration

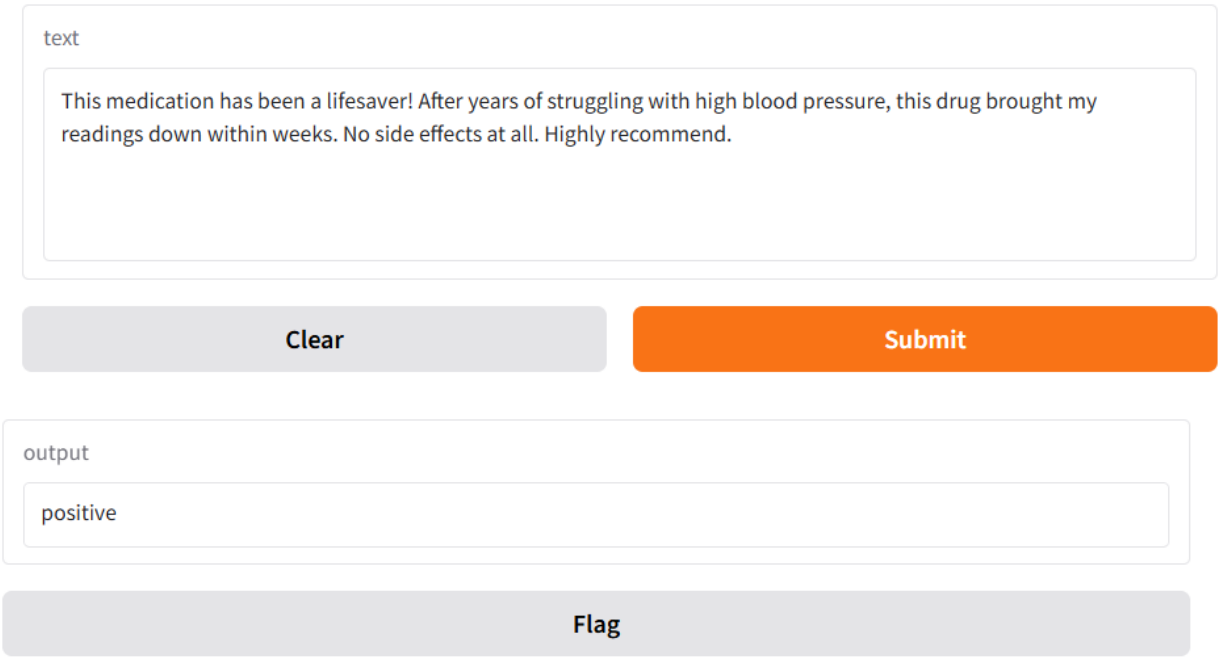
- Tool Used: Gradio
- Interface Functionality:
  - Accepts user input as drug review text.
  - Processes input through the best LoRA model (r=16, alpha=64, dropout=0.1).
  - Outputs sentiment prediction as positive, neutral, or negative along with confidence scores.
  - Provides a simple, accessible interface for real-time interaction with the model.
- Implementation Details:
  - User inputs are processed through a prompt template: "Classify the sentiment of this review as positive, neutral, or negative:\n{text}".
  - Predictions are generated using the generate() method of the LoRA-enhanced Flan-T5 model.
  - Confidence scores are calculated by softmax activation over the model outputs.

Link to the presentation on UI:
https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=afb727ed-b876-45ff-8fc2-b2e00012222c

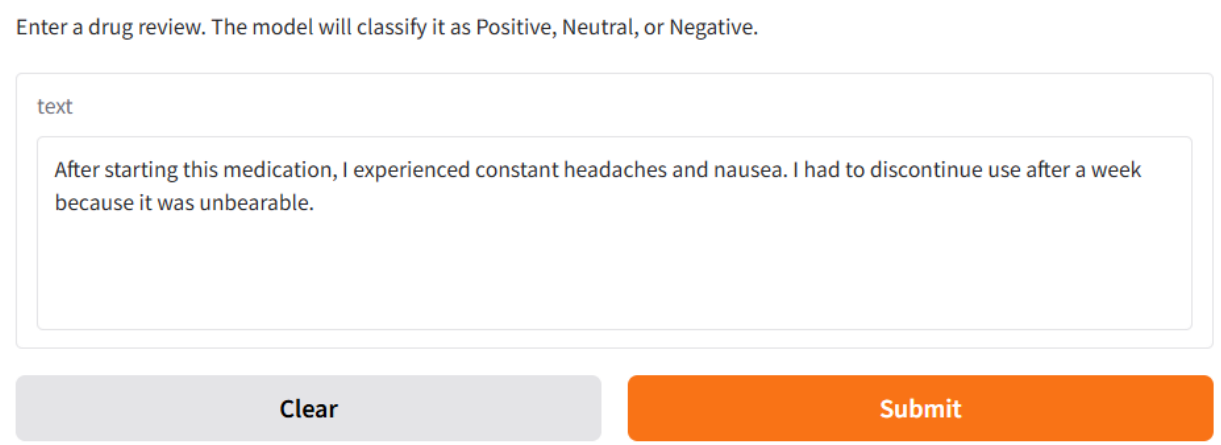
UI Interface:



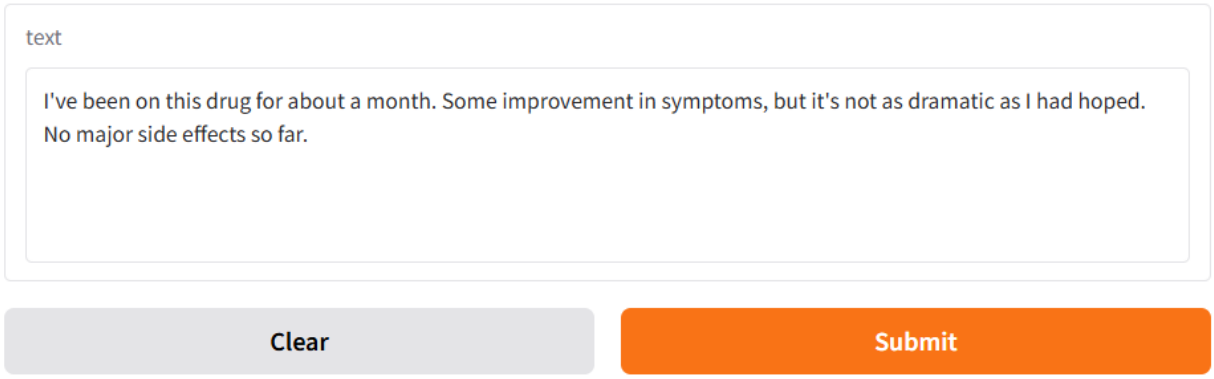
Positive Output Generated by model:



Negative Output Generated by model:



Neutral Output Generated by model:



11. Conclusion and Future Work

- The implementation of LoRA has proven to be highly effective in reducing model parameters while maintaining or even improving sentiment classification accuracy.
- The best configuration (r=16, alpha=64, dropout=0.1) achieved the highest macro F1-score of 66.30%, establishing a strong baseline for future optimization.

12. Future Work and Extensions

- Implement data augmentation strategies to generate more balanced neutral samples, including text paraphrasing and sentiment masking techniques.
- Explore additional LoRA configurations targeting key-value attention layers to capture contextual nuances in reviews.

- Integrate domain-specific embeddings to further refine sentiment classification, particularly for medical or pharmaceutical contexts.
- Deploy the model as a streaming API to allow continuous review processing and sentiment monitoring.

References:

[1]. Gräßer, F., Kallumadi, S., Malberg, H. and Zaunseder, S., 2018. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In: Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, pp.1380–1387. doi:10.1109/BIBM.2018.8621392

[2]. Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W., 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685. Available at:https://arxiv.org/abs/2106.09685

[3]. Kallumadi, S. and Grer, F., 2018. Drug Reviews (Druglib.com) [dataset]. UCI Machine Learning Repository. Available at https://archive.ics.uci.edu/dataset/461/drug+review+dataset+druglib+com [Accessed 23 March 2025]. doi:10.24432/C55G6J