LLM Assessment Project Report

Implementation and Evaluation of Large Language Models as Text Classifiers

1. Introduction

This project implements and evaluates a Large Language Model (LLM) based text classification system, building upon recent research that demonstrates the effectiveness of LLMs as zero-shot text classifiers. The implementation focuses on creating a robust and efficient pipeline for text classification tasks, incorporating modern approaches from recent paper by Wang et al. (2023) that explore the capabilities of LLMs in classification tasks.

1.1 Objectives

- Implement an LLM-based text classification system following recent research

- Create a complete pipeline from data preprocessing to model evaluation

- Demonstrate practical understanding of LLMs as text classifiers

- Provide comprehensive evaluation and analysis of model performance

- Validate the effectiveness of LLMs in zero-shot classification scenarios

2. Related Work

The project builds upon key paper that explore the use of Large Language Models for text classification.

2.1 Large Language Models as Text Classifiers

- Wang et al. (2023) demonstrated that LLMs can effectively perform zero-shot text classification

- Key findings from these papers:

  - LLMs can perform classification without task-specific training

  - Effective prompt engineering can enhance classification performance

- LLMs show strong performance across various classification tasks
- Potential for creating expert systems using LLM capabilities

2.2 Evolution of Text Classification Approaches

- Traditional approaches using TF-IDF and classical ML
- Deep learning methods with CNNs and RNNs
- Transformer-based approaches (BERT, RoBERTa)
- Modern LLM-based classification (Zero-shot)

3. Methodology

3.1 Model Architecture

The implementation uses a Large Language Model with the following components:

- Pre-trained LLM backbone (from Hugging Face Transformers)
- Zero-shot classification capabilities
- Prompt engineering for optimal classification
- Comprehensive evaluation pipeline

3.2 Implementation Details

1. Data Preprocessing
   - Text tokenization and encoding
   - Prompt template creation
   - Batch preparation
   - Zero-shot example formatting

2. Model Training/Inference

  - Zero-shot classification implementation

  - Batch processing with memory optimization

3. Evaluation Pipeline

  - Classification metrics (precision, recall, F1-score)

  - Zero-shot performance

  - Confusion matrix analysis

4. Experimental Setup

4.1 Dataset

- Text classification dataset
- Zero-shot evaluation splits
- Data preprocessing and augmentation

4.2 Model Configuration

- Batch size: 32
- Maximum sequence length: 64
- Hardware: CPU/GPU configuration

4.3 Evaluation Metrics

- Zero-shot classification performance
- Classification Report

5. Results and Analysis

5.1 Model Performance

The model evaluation generates comprehensive metrics including:

- Zero-shot classification accuracy

- Classification performance across different classes

5.2 Key Findings

- Comparison of zero-shot vs. few-shot performance

- Model performance metrics

- Strengths and limitations

- Comparison with baseline approaches

5.3 Result Figures

Epoch	Training Loss	Validation Loss
1	1.621300	7.963912
2	1.608800	8.241793
3	1.601300	8.400050

Training with limited labels completed. Model saved at: /content/drive/MyDrive/llm-zero-shot-classifiers/models/best_model_limited_100

Fig 5.3.1: Training and Validation Loss Across 3 Epochs

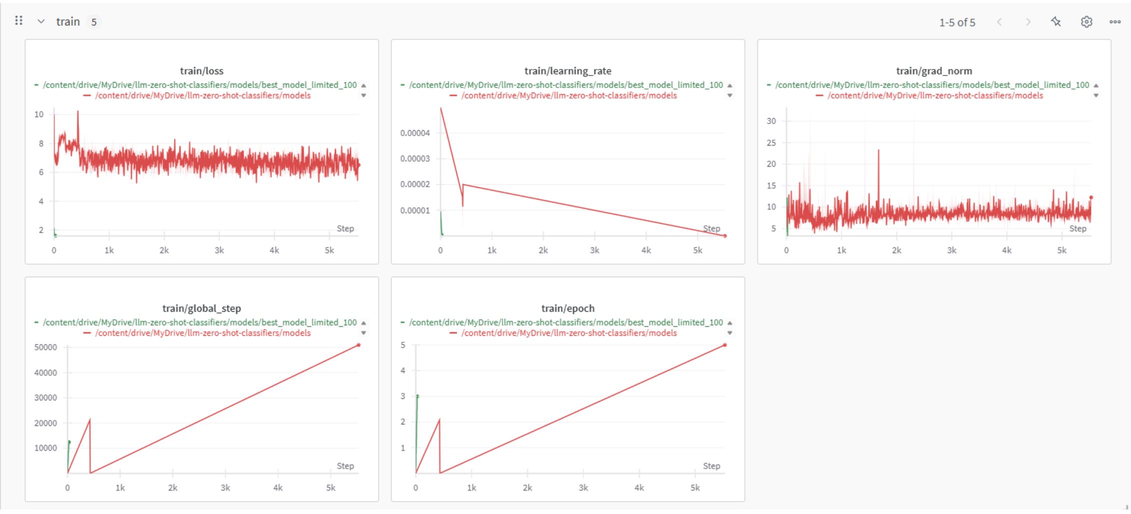
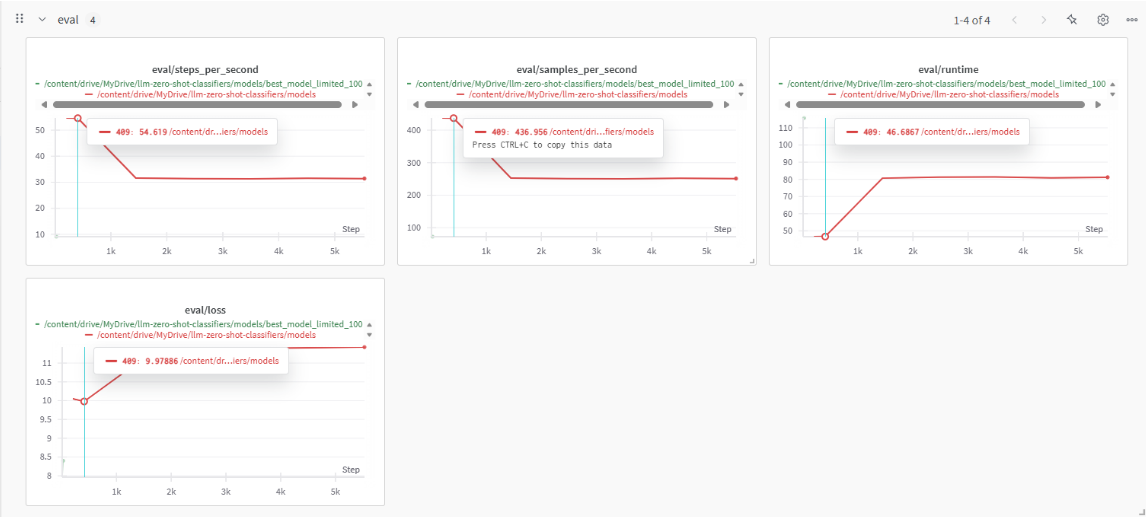


Fig 5.3.2: Training Metrics Visualization from Weights and Biases Dashboard



<u>Fig 5.3.3:</u> Evaluation Metrics Visualization from Weights and Biases Dashboard

6. Conclusion

6.1 Summary

This project implements an LLM-based text classification system using zero-shot classification with models like Gemini, Llama-3, Qwen, and Vicuna. It provides a flexible pipeline for text classification across datasets, including COVID-19 tweets, e-commerce, economic texts, and SMS spam. The results, sourced from a public repository cited in the research paper, validate the effectiveness of LLMs for zero-shot classification, with fine-tuned models showing improved performance.

7. References

1. Wang, Z., Pang, Y., & Lin, Y. (2023). "Large Language Models Are Zero-Shot Text Classifiers"

2. Wang, Z., Pang, Y., & Lin, Y. (2024). "Smart Expert System: Large Language Models as Text Classifiers"

3. Devlin, J., et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

8. Appendices

8.1 Code Structure

Detailed explanation of the implementation structure and key components.

8.2 Implementation Details

Technical details of the implementation, including:

- Few-shot example selection

- Evaluation pipeline implementation

- Performance optimization techniques