Text Classification Project Report

AG News Dataset Classification Using DistilRoBERTa

---

1. Project Overview

Objective

To develop and train a text classification model that can automatically categorize news articles into 4 different categories using the AG News dataset.

Dataset Used

- **Dataset:** AG News Dataset
- **Source:** Hugging Face Datasets
- **Number of Classes:** 4 categories
    - World News
    - Sports
    - Business
    - Science/Technology
- **Training Samples:** 120,000 articles
- **Test Samples:** 7,600 articles

---

2. Technical Implementation

Model Architecture

- **Base Model:** DistilRoBERTa-base
- **Model Type:** Transformer-based sequence classification
- **Framework:** Hugging Face Transformers
- **Computing Platform:** CUDA-enabled GPU

Data Preprocessing

- **Tokenization:** AutoTokenizer from DistilRoBERTa
- **Maximum Sequence Length:** 128 tokens
- **Padding:** Applied to maximum length
- **Truncation:** Enabled for longer texts

Training Configuration

- **Number of Epochs:** 3
- **Training Batch Size:** 16

- **Evaluation Batch Size:** 64
- **Learning Rate:** 2e-5
- **Weight Decay:** 0.01
- **Warmup Ratio:** 0.1
- **Mixed Precision:** FP16 enabled
- **Optimizer:** AdamW with linear learning rate scheduling

---

3. Model Performance Results

Key Metrics Achieved

Based on the model evaluation, the following performance metrics were obtained:

Overall Accuracy: 95% (0.95)

Weighted Precision: 94.96% (0.9496)

Weighted Recall: 94.93% (0.9493)

Weighted F1-Score: 94.94% (0.9494)

Macro F1-Score: 94.94% (0.9494)

Micro F1-Score: 94.93% (0.9493)

4. Technical Achievements

Model Training Success

- Successfully fine-tuned DistilRoBERTa on AG News dataset
- Completed 3 epochs of training with stable convergence
- Implemented comprehensive evaluation metrics
- Generated confusion matrix for detailed performance analysis

Model Deployment Preparation

- Saved trained model in two formats:
    1. Hugging Face format (complete model + tokenizer)
    2. PyTorch state dictionary (.pth file)
- Model ready for inference and deployment
- Total checkpoint saved at step 22,500

Code Implementation Features

- **Reproducibility:** Set random seed (42) for consistent results
- **Memory Optimization:** Used gradient accumulation and mixed precision

- **Monitoring:** Implemented logging every 50 steps
- **Best Model Selection:** Automatic saving of best performing checkpoint
- **Evaluation:** Comprehensive metrics including confusion matrix

---

5. Tools and Technologies Used

Libraries and Frameworks

- **PyTorch:** Deep learning framework
- **Transformers:** Hugging Face library for pre-trained models
- **Datasets:** Hugging Face datasets library
- **Scikit-learn:** Machine learning metrics and evaluation
- **NumPy & Pandas:** Data manipulation
- **Matplotlib:** Visualization for confusion matrix

Development Environment

- **Platform:** Google Colab with CUDA-enabled environment
- **Python Version:** 3.11.13
- **Hardware:** GPU acceleration enabled
- **Storage:** Local checkpoint saving with cloud download capability

---

6. Project Workflow

Step 1: Environment Setup

- Installed required libraries using pip
- Configured CUDA device for GPU acceleration
- Set random seeds for reproducible results

Step 2: Data Loading and Preprocessing

- Loaded AG News dataset from Hugging Face
- Applied tokenization with DistilRoBERTa tokenizer
- Set appropriate padding and truncation parameters

Step 3: Model Configuration

- Initialized DistilRoBERTa model for sequence classification
- Configured training arguments for optimal performance
- Set up comprehensive evaluation metrics

Step 4: Training Process

- Executed 3-epoch training with evaluation after each epoch
- Monitored training progress with logging
- Automatically saved best performing model

Step 5: Model Evaluation

- Generated detailed classification report
- Created confusion matrix visualization
- Analyzed sample predictions with confidence scores

Step 6: Model Export

- Saved complete model and tokenizer
- Exported PyTorch state dictionary
- Prepared model for deployment

---

7. Key Learning Outcomes

Technical Skills Developed

- Hands-on experience with transformer-based models
- Understanding of fine-tuning pre-trained language models
- Implementation of comprehensive model evaluation
- Experience with modern NLP libraries and frameworks

Best Practices Applied

- Proper train/validation/test split usage
- Comprehensive metrics evaluation beyond accuracy
- Model checkpointing and version control
- Reproducible research practices with seed setting

---

8. Future Improvements

Potential Enhancements

1. **Hyperparameter Tuning:** Experiment with different learning rates and batch sizes
2. **Model Comparison:** Test other pre-trained models like BERT, RoBERTa-large
3. **Data Augmentation:** Apply text augmentation techniques
4. **Cross-Validation:** Implement k-fold cross-validation for robust evaluation

5. **Deployment:** Create REST API or web interface for model inference

Additional Metrics

- Implement per-class precision-recall curves
- Add ROC curves for multi-class classification
- Calculate computational efficiency metrics (training time, inference speed)

---

9. Conclusion

This project successfully demonstrated the implementation of a state-of-the-art text classification system using DistilRoBERTa. The model achieved strong performance on the AG News dataset, effectively categorizing news articles into four distinct categories. The comprehensive evaluation approach and proper model saving procedures ensure the work is both scientifically rigorous and practically deployable.

The project provided valuable hands-on experience with modern NLP techniques and established a solid foundation for future text classification tasks.