INFO 7375
PROMPT ENGINEERING AND AI

Report
Fine-Tuning FinBERT for Financial Sentiment Analysis

Fine-Tuning Large Language Models for Financial Sentiment Analysis: A Comprehensive Study

Executive Summary

This report presents a comprehensive fine-tuning study of Large Language Models (LLMs) for financial sentiment analysis, addressing the critical need for domain-specific natural language processing in financial markets. The Assignment implements a systematic approach to fine-tuning two distinct pre-trained models—DistilBERT (general-purpose) and FinBERT (domain-specific)—on a curated financial news dataset containing 4,837 samples across three sentiment classes: negative, neutral, and positive.The study demonstrates significant performance improvements through fine-tuning, with FinBERT achieving 88.98% accuracy (an 84.02% improvement over baseline) and DistilBERT reaching 85.67% accuracy (a 30.17% improvement). Through systematic hyperparameter optimization across three distinct configurations, comprehensive error analysis, and the development of a production-ready inference pipeline, this work establishes a robust framework for financial sentiment analysis that can be deployed in real-world trading and risk management applications.

1. Dataset Preparation and Preprocessing

1.1 Dataset Selection and Characteristics

The Assignment utilizes a financial sentiment analysis dataset sourced from Kaggle, containing 4,846 financial news headlines and articles with manually annotated sentiment labels. The dataset exhibits a realistic class distribution reflective of financial news: 59.4% neutral, 28.2% positive, and 12.5% negative sentiment, mirroring the typical balance of financial reporting where most news is factual rather than emotionally charged.

Key Dataset Statistics:

- Total samples: 4,837 (after preprocessing)
- Average text length: 23.1 words, 128 characters
- Text length range: 3-150 words (filtered for quality)
- Class distribution: Imbalanced but representative of real-world financial news

1.2 Comprehensive Data Preprocessing

The preprocessing pipeline implements domain-specific considerations for financial text:Text Cleaning Strategy:

- Preserved financial symbols ($, %, numbers) critical for sentiment analysis
- Normalized whitespace while maintaining numerical precision
- Removed texts shorter than 3 words or longer than 150 words
- Eliminated exact duplicates and duplicate texts (keeping first occurrence)

Quality Control Measures:

- Removed 9 samples (0.2%) during preprocessing
- Verified no missing values in text or sentiment columns
- Applied lowercase normalization to sentiment labels
- Reset indices for clean dataset structure

1.3 Data Splitting and Label Encoding

The dataset was strategically split using stratified sampling to maintain class distribution:

- Training set: 3,387 samples (70%)
- Validation set: 724 samples (15%)
- Test set: 726 samples (15%)

Label encoding mapped sentiment classes to numerical values: negative=0, neutral=1, positive=2, with reverse mapping preserved for inference. This encoding strategy maintains alphabetical ordering while providing clear class boundaries for the classification task.

2. Model Selection and Architecture

2.1 Comparative Model Strategy

The study implements a dual-model approach to evaluate the impact of domain-specific pre-training:DistilBERT (General-Purpose Model):

- Architecture: DistilBERT-base-uncased
- Parameters: 66 million
- Pre-training: General English text (Wikipedia, Books)
- Rationale: Establishes baseline performance for general-purpose models

FinBERT (Domain-Specific Model):

- Architecture: ProsusAI/finbert
- Parameters: 110 million
- Pre-training: Financial domain text (earnings reports, financial news)
- Rationale: Tests hypothesis that domain-specific pre-training provides superior foundation

2.2 Model Architecture Configuration

Both models were configured for sequence classification with:

- 3 output classes (negative, neutral, positive)
- Custom label mappings (id2label, label2id)
- Compatible tokenization (max_length=128 for financial text)
- GPU optimization with mixed precision training (FP16)

The architecture selection demonstrates thoughtful consideration of computational efficiency (DistilBERT) versus domain expertise (FinBERT), enabling direct comparison of general versus specialized pre-training approaches.

3. Fine-Tuning Implementation and Training Infrastructure

3.1 Training Environment Configuration

The fine-tuning setup implements production-grade training infrastructure:Hardware Optimization:

- GPU: Tesla T4 (15.83 GB memory)
- Mixed precision training (FP16) for memory efficiency

- Batch processing with optimized data loaders
- Device-agnostic code with automatic GPU detection

Training Configuration:

- Learning rate: 2e-5 (standard BERT fine-tuning)
- Batch size: 16 (optimized for GPU memory)
- Epochs: 3 (with early stopping)
- Weight decay: 0.01 (regularization)
- Warmup steps: 500 (learning rate scheduling)

3.2 Advanced Training Features

Callback Implementation:

- Early stopping with patience=3 to prevent overfitting
- Best model checkpointing based on F1 score
- Comprehensive metrics tracking (accuracy, precision, recall, F1)
- TensorBoard logging for training visualization

Reproducibility Measures:

- Fixed random seed (42) across all experiments
- Version pinning for all dependencies
- Complete configuration saving to JSON files
- Systematic experiment tracking with run names

3.3 Training Results and Performance

DistilBERT Fine-Tuning:

- Training time: 123.17 seconds
- Final training loss: 0.5781
- Samples per second: 82.50
- Total steps: 636

FinBERT Fine-Tuning:

- Training time: 378.78 seconds (Config 3)
- Final training loss: 0.4907
- Total steps: 1000
- Superior convergence due to domain-specific pre-training

4. Hyperparameter Optimization Strategy

4.1 Systematic Configuration Testing

The study implements a comprehensive hyperparameter optimization across three distinct configurations:Configuration 1 (Standard):

- Learning rate: 2e-5

- Batch size: 16
- Epochs: 3
- Results: 88.98% accuracy, 0.8891 F1 score

Configuration 2 (Conservative):

- Learning rate: 1e-5 (50% reduction)
- Batch size: 16
- Epochs: 5 (67% increase)
- Results: 88.84% accuracy, 0.8885 F1 score

Configuration 3 (Aggressive):

- Learning rate: 5e-5 (150% increase)
- Batch size: 8 (50% reduction)
- Epochs: 4
- Results: 86.36% accuracy, 0.8624 F1 score

4.2 Optimization Analysis and Insights

Learning Rate Impact:Configuration 1's learning rate (2e-5) proved optimal, demonstrating the importance of balanced learning speed. The conservative approach (Config 2) showed minimal improvement, while the aggressive approach (Config 3) led to performance degradation, indicating the critical nature of learning rate selection.Training Duration Analysis:The 3-epoch configuration (Config 1) achieved the best performance, suggesting that financial sentiment analysis benefits from focused, efficient training rather than extended epochs. This finding has practical implications for computational cost and deployment timelines.Batch Size Considerations:Larger batch sizes (16) consistently outperformed smaller batches (8), likely due to more stable gradient estimates and better convergence in the financial domain.

5. Model Evaluation and Performance Analysis

5.1 Comprehensive Evaluation Metrics

The evaluation framework implements multiple metrics to provide holistic performance assessment:FinBERT (Best Configuration):

- Test Accuracy: 88.98%
- F1 Score: 0.8891
- Precision: 0.8905
- Recall: 0.8898

DistilBERT:

- Test Accuracy: 85.67%
- F1 Score: 0.8546
- Precision: 0.8569
- Recall: 0.8567

5.2 Baseline Comparison Analysis

FinBERT Performance Improvement:

- Baseline accuracy: 4.96% (random performance due to label mismatch)
- Fine-tuned accuracy: 88.98%
- Improvement: +84.02 percentage points

DistilBERT Performance Improvement:

- Baseline accuracy: 55.51%
- Fine-tuned accuracy: 85.67%
- Improvement: +30.17 percentage points

The dramatic improvement in FinBERT performance, despite poor baseline alignment, demonstrates the effectiveness of fine-tuning in adapting pre-trained models to specific tasks, even when initial label mappings are suboptimal.

5.3 Per-Class Performance Analysis

Detailed Classification Report (FinBERT):

- Negative: Precision=0.8119, Recall=0.9011, F1=0.8542
- Neutral: Precision=0.8675, Recall=0.9118, F1=0.8891
- Positive: Precision=0.8547, Recall=0.7206, F1=0.7819

The analysis reveals that the model performs best on neutral sentiment (most common class) and struggles slightly with positive sentiment classification, likely due to the nuanced nature of positive financial news and the class imbalance in the dataset.

6. Error Analysis and Model Limitations

6.1 Systematic Error Pattern Analysis

The error analysis identified several critical patterns in model failures:Confusion Matrix Analysis:

- Most common error: Neutral→Positive misclassification
- Second most common: Positive→Neutral misclassification
- Least common: Negative sentiment errors (model excels at identifying negative sentiment)

Confidence Calibration:

- Correct predictions: Average confidence 85.2%
- Incorrect predictions: Average confidence 68.7%
- Confidence gap: 16.5% difference indicates good calibration

6.2 Linguistic Feature Impact Analysis

Negation Handling:The model struggles with negated statements, particularly in financial contexts where "not profitable" versus "profitable" requires careful semantic understanding.Contrasting Statements:Sentences containing "but," "however," or "despite" show increased error rates, as the model must balance competing sentiment signals within single statements.Financial Terminology:The model demonstrates strong performance on standard financial terms but occasionally misinterprets technical jargon or company-specific language.

6.3 Targeted Improvement Recommendations

Data Augmentation Strategy:

- Increase training examples with negations and contrasting statements
- Add paraphrased versions of existing samples
- Include more diverse financial terminology and company-specific language

Class Balance Optimization:

- Implement class weights during training (positive class weight: 2.0x)
- Apply focal loss to focus on hard examples
- Collect additional positive sentiment samples

Confidence-Based Prediction Refinement:

- Set confidence threshold (65%) for automated decisions
- Flag low-confidence predictions (<60%) for human review
- Implement ensemble methods for uncertain cases

7. Production-Ready Inference Pipeline

7.1 Modular Inference Architecture

The inference pipeline implements a comprehensive, production-ready system:Core Functions:

- predict_sentiment(): Single text prediction with probability distributions
- predict_batch(): Optimized batch processing for high-throughput scenarios
- analyze_financial_sentiment(): Interactive analysis with confidence scoring

Performance Optimization:

- Batch processing: ~50 texts/second throughput
- Single prediction latency: ~50ms average
- Batch prediction latency: ~20ms per text
- Memory-efficient tokenization with truncation and padding

7.2 Deployment Infrastructure

Gradio Web Interface:

- Real-time sentiment analysis with confidence scores
- Batch processing capabilities
- Shareable public URL for demonstration
- User-friendly interface for non-technical stakeholders

Standalone Package:

- inference_pipeline.py: Complete deployment-ready class
- MODEL_README.md: Comprehensive documentation
- requirements.txt: Dependency management
- Version control and reproducibility measures

7.3 Real-World Application Scenarios

Financial News Monitoring:

- Real-time sentiment analysis of news feeds
- Automated alert systems for significant sentiment changes
- Portfolio impact assessment based on news sentiment

Risk Management:

- Early warning systems for market sentiment shifts
- Automated risk scoring based on news analysis
- Integration with existing risk management frameworks

Trading Strategy Enhancement:

- Sentiment-based trading signal generation
- Market timing optimization using sentiment data
- Quantitative strategy backtesting with sentiment factors

8. Ethical Considerations and Limitations

8.1 Model Limitations and Constraints

Language and Domain Constraints:

- English-only processing (limitation for global financial markets)
- Professional financial sources only (excludes social media sentiment)
- Requires periodic retraining for evolving financial terminology

Bias and Fairness Considerations:

- Class imbalance may introduce subtle biases toward neutral predictions
- Potential geographic bias due to English-only training data
- Need for continuous monitoring of prediction fairness across different market conditions

8.2 Deployment Safeguards

Confidence Thresholds:

- Automated decisions only for high-confidence predictions (>65%)
- Human review required for low-confidence cases (<60%)
- Escalation procedures for uncertain or contradictory predictions

Monitoring and Maintenance:

- Regular performance monitoring on new data
- Adversarial testing for robustness
- Continuous learning pipeline for model updates

9. Conclusions and Future Directions

9.1 Key Findings and Contributions

This study demonstrates the significant value of fine-tuning LLMs for financial sentiment analysis, with FinBERT achieving 88.98% accuracy through domain-specific pre-training and

systematic hyperparameter optimization. The comprehensive error analysis provides actionable insights for model improvement, while the production-ready inference pipeline enables real-world deployment.

Technical Contributions:

- Systematic comparison of general versus domain-specific pre-training
- Comprehensive hyperparameter optimization with clear performance insights
- Production-ready inference pipeline with confidence-based decision making
- Detailed error analysis with targeted improvement recommendations

9.2 Practical Implications

The 84.02% improvement in FinBERT performance over baseline demonstrates the critical importance of domain-specific pre-training for financial applications. The systematic approach to hyperparameter optimization provides a replicable framework for similar projects, while the error analysis offers concrete strategies for model enhancement.

Business Value:

- Real-time financial sentiment analysis capability
- Automated risk assessment and trading signal generation
- Scalable deployment architecture for enterprise applications
- Clear confidence thresholds for human-in-the-loop decision making

10. References

1. Malo, P., Sinha, A., Korhonen, P., Wallenius, J., & Takala, P. (2014). Good debt or bad debt: Detecting semantic orientations in economic texts. **Journal of the Association for Information Science and Technology**, 65(4), 782-796.

2. Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. **arXiv preprint arXiv:1908.1006***.

3. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. **arXiv preprint arXiv:1910.01108**.

4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805.**

5. Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-art natural language processing. **Proceedings of EMNLP: System Demonstrations**, 38-45.

6. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. **arXiv preprint arXiv:1801.06146**.