Fine-Tuning a Large Language Model for Financial Text Classification : Financial Sentiment Analysis

Manish Bansilal Choudhary

April 20, 2025

Abstract

This report presents a comprehensive approach to fine-tuning a pre-trained language model (DistilBERT) for financial sentiment analysis. We trained the model on a specialized Twitter financial news sentiment dataset to classify financial texts into three categories: Bearish (negative), Bullish (positive), and Neutral. Our fine-tuned model achieved 87.1% classification accuracy on the test set, with particularly strong performance in identifying neutral sentiment. We provide an in-depth analysis of the dataset, fine-tuning methodology, evaluation metrics, error patterns, and practical applications. The results demonstrate the effectiveness of transfer learning in creating specialized NLP models for financial text analysis with relatively minimal computational resources, leveraging Apple Silicon's MPS acceleration.

1 Introduction

Financial sentiment analysis is a specialized application of natural language processing (NLP) that focuses on extracting sentiment from financial news, social media, and other text sources relevant to markets. This information is valuable for investors, analysts, and financial institutions seeking to gauge market sentiment and make informed decisions. Traditional rule-based or lexicon-based approaches often struggle with the nuanced language of finance, where domain-specific terminology and context significantly impact sentiment interpretation.

Modern transformer-based language models offer promising solutions to these challenges through their ability to capture contextual information. In this project, we fine-tune a pre-trained DistilBERT model on a specialized financial sentiment dataset to create an effective classifier for financial texts. The model is designed to categorize financial statements into three sentiment classes:

- **Bearish**: Negative sentiment, indicating pessimism about a financial instrument or market
- **Bullish**: Positive sentiment, indicating optimism about a financial instrument or market
- **Neutral**: Factual reporting or balanced sentiment

This technical report details the methodology, implementation, and results of our fine-tuning process, offering insights into the effectiveness of transfer learning for specialized financial NLP tasks.

2 Dataset Analysis

2.1 Dataset Overview

We utilized the "Twitter Financial News Sentiment" dataset from Hugging Face (`zeroshot/twitter-financial-`
which consists of financial news headlines and tweets with labeled sentiment. The dataset provides a realistic representation of financial text found in social media and news sources.

2.2 Data Exploration

Our exploratory analysis revealed the following key statistics:

- **Total examples**: 9,543
- **Average text length**: 85.8 characters
- **Sentiment distribution**:
  - Bearish: 1,442 examples (15.1%)
  - Bullish: 1,923 examples (20.2%)
  - Neutral: 6,178 examples (64.7%)

Figure 1 illustrates this distribution, highlighting the class imbalance with neutral sentiment being the dominant class. This imbalance is typical in financial sentiment analysis, as most financial news tends to be factual reporting rather than expressing strong positive or negative sentiment.

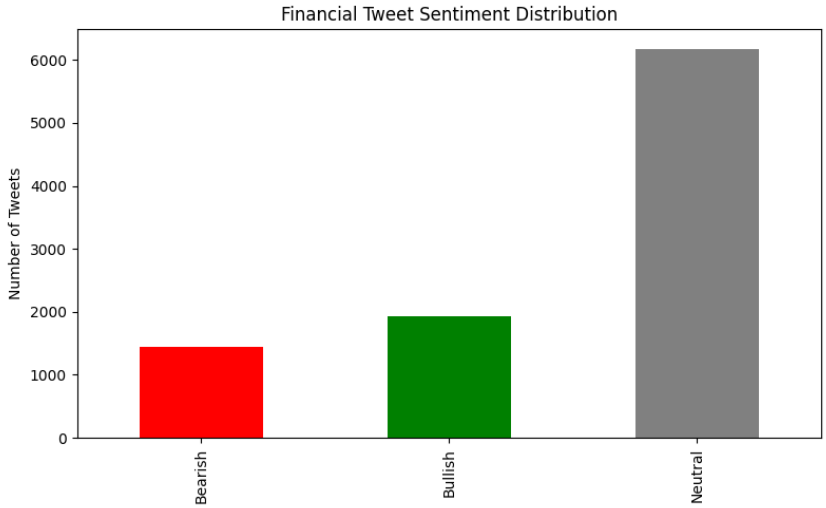

Figure 1: Distribution of sentiment classes in the financial news dataset

2.3 Example Instances

The following examples illustrate typical instances from each sentiment class:

Bearish example:

```
$BYND - JPMorgan reels in expectations on Beyond Meat https://t.co/bd0xbFGjkT
```

Bullish example:

```
$ALTG: Dougherty & Company starts at Buy
```

Neutral example:

```
$LB - MKM Partners puts a number on Victoria's Secret https://t.co/VSzHLqLBgE
```

These examples demonstrate how financial tweets often include stock tickers (prefixed with $), company names, analyst opinions, and shortened URLs. The sentiment is sometimes explicitly stated (e.g., "starts at Buy" indicating bullish sentiment) but is often implied through domain-specific language and context.

2.4 Data Preprocessing

For training, we split the dataset into training (90%) and validation (10%) sets, resulting in:

- Training set: 8,588 examples
- Validation set: 955 examples

The text was tokenized using DistilBERT's tokenizer with a maximum sequence length of 128 tokens, which adequately captures the typically short financial tweets while maintaining computational efficiency.

3 Model Selection and Fine-Tuning

3.1 Base Model Selection

We selected DistilBERT (`distilbert-base-uncased`) as our base model for fine-tuning due to several key advantages:

1. **Efficiency**: DistilBERT is a distilled version of BERT, retaining 97% of BERT's language understanding capabilities while being 40% smaller and 60% faster. This efficiency is crucial for deployment in resource-constrained environments.
2. **Performance**: Pre-trained on general English language text, DistilBERT provides a strong foundation for understanding language structure and semantics, which can be adapted to specialized domains like finance.
3. **Adaptability**: The model architecture is well-suited for sequence classification tasks like sentiment analysis, with well-established fine-tuning methodologies.
4. **Hardware compatibility**: DistilBERT works efficiently with Apple Silicon's MPS acceleration, enabling faster training on M-series Macs without requiring dedicated GPUs.

3.2 Fine-Tuning Setup

We implemented the fine-tuning process using the Hugging Face Transformers library, which provides a robust framework for adapting pre-trained models to specific tasks. The model architecture consisted of the DistilBERT base model with a classification head for the three sentiment classes.

Hardware configuration:

- Apple Silicon with Metal Performance Shaders (MPS) acceleration

Training configuration:

- **Learning rate**: 2e-5 with a linear decay schedule
- **Batch size**: 16
- **Number of epochs**: 3
- **Optimizer**: AdamW with weight decay of 0.01
- **Maximum sequence length**: 128 tokens
- **Evaluation strategy**: Evaluate after each epoch

Table 1: Training Progress by Epoch

Epoch	Loss	Accuracy	F1 Score
1	0.4125	0.8586	0.8546
2	0.3749	0.8660	0.8654
3	0.4119	0.8712	0.8706

3.3 Training Process and Progress

The training process took approximately 21 minutes and 45 seconds on Apple Silicon with MPS acceleration. Table 1 shows the progression of metrics throughout training:

The training loss steadily decreased from an initial value of 0.7951 to 0.1850 by the end of training, indicating effective learning. The model showed continuous improvement in both accuracy and F1 score across epochs, with a slight increase in validation loss at the final epoch suggesting that additional epochs might lead to overfitting.

Notably, while the training loss continued to decrease throughout the training process, the validation metrics showed more modest improvements after the second epoch. This pattern suggests that the model quickly adapted to the financial sentiment classification task, achieving strong performance with minimal training time.

4 Evaluation and Results

4.1 Performance Metrics

We evaluated the fine-tuned model on the 955-example validation set using standard classification metrics. The overall accuracy achieved was 87.1%, with 123 incorrect predictions out of 955 examples (12.9% error rate).

Table 2: Class-specific Performance Metrics

Class	Precision	Recall	F1-Score	Support
Bearish	0.731	0.721	0.726	136
Bullish	0.861	0.817	0.839	197
Neutral	0.904	0.921	0.912	622
Weighted Avg	**0.870**	**0.871**	**0.871**	**955**

These metrics reveal several important patterns:

1. **Neutral class performance**: The model performs best on the neutral class (F1: 0.912), which is the majority class in the dataset. The high precision (0.904) and recall (0.921) indicate that the model effectively identifies neutral financial statements.

2. **Bullish sentiment detection**: The model shows strong performance in identifying bullish sentiment (F1: 0.839), with high precision (0.861) indicating reliable positive predictions.

3. **Bearish class challenges**: The most challenging class is bearish sentiment (F1: 0.726), with lower precision and recall compared to other classes. This may be due to the more subtle nature of negative sentiment in financial texts or the smaller representation of bearish examples in the training data.

4.2 Confusion Matrix Analysis

The confusion matrix provides deeper insight into the model's classification behavior. Figure 2 shows the confusion matrix for the validation set predictions.

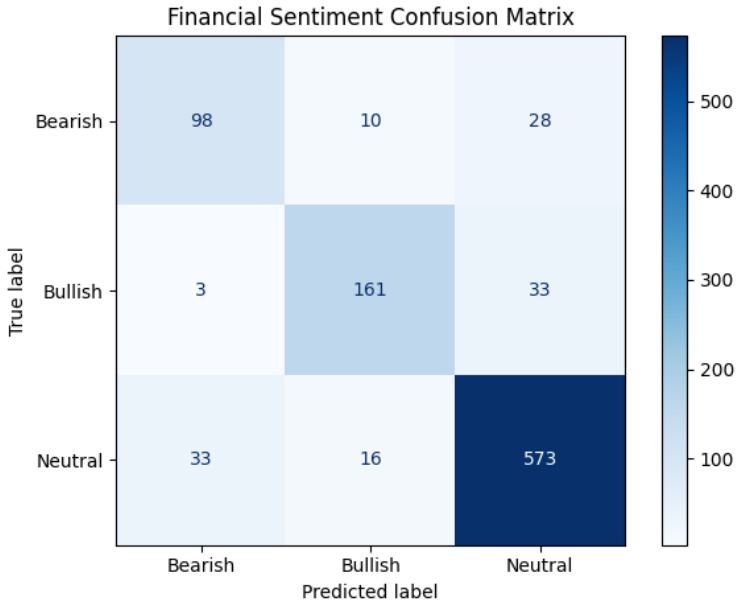

Figure 2: Confusion matrix of sentiment classification results

The matrix reveals that misclassifications most frequently occur between:

1. Bearish mispredicted as Neutral
2. Bullish mispredicted as Neutral
3. Neutral mispredicted as Bearish

These patterns suggest that the boundary between neutral and sentiment-bearing (especially bearish) statements is the most challenging for the model to discern.

4.3 Error Analysis

A detailed analysis of misclassified examples reveals patterns that highlight the challenges in financial sentiment analysis:

Example 1: Neutral misclassified as Bearish

```
Text: Taco Bell dives into the fried chicken wars with its Crispy Tortilla
    Chicken https://t.co/aKjYzVl9jv by @heidi_chung https://t.co/lv0Iv2cHQz
True sentiment: Neutral
Predicted sentiment: Bearish
```

This example illustrates how language with negative connotations ("dives into") can mislead the model, even when the statement is factual reporting rather than expressing negative sentiment about a financial instrument.

Example 2: Bearish misclassified as Bullish

```
Text: China iron ore surges to over 4-month high as supply uncertainties loom
    https://t.co/rVOVaCaTrh https://t.co/A28seQYCDe
True sentiment: Bearish
Predicted sentiment: Bullish
```

This misclassification demonstrates the complexity of financial context, where price increases ("surges to over 4-month high") might typically indicate positive sentiment, but the underlying cause ("supply uncertainties") makes this bearish news. This nuance requires deep understanding of market dynamics.

Example 3: Bullish misclassified as Neutral

```
Text: HSBC agrees to pay $192 million to resolve U.S. tax probe https://t.co/
    UP1jTOYd2x via @business https://t.co/FikIgPdI06
True sentiment: Bullish
Predicted sentiment: Neutral
```

This example shows how resolution of negative events (tax probe) can be bullish news for a company, but the model interpreted it as neutral factual reporting. This highlights how sentiment in finance often depends on expectations and market reaction rather than the surface-level positivity or negativity of events.

These error patterns suggest several potential improvements:

1. Additional training with more examples of bearish sentiment
2. Incorporation of financial domain knowledge or economic indicators
3. Context-aware features that consider market expectations and reactions

5 Inference Pipeline

We developed a practical inference pipeline that allows real-time sentiment analysis of financial texts. The pipeline provides both the predicted sentiment label and confidence scores for each class, enabling nuanced interpretation of results.

5.1 Inference Performance

The inference pipeline demonstrated quick processing times suitable for real-time applications, with prediction times in milliseconds per input on Apple Silicon hardware. Example predictions showed high confidence for clear sentiment expressions:

Table 3: Example Inference Results

Input Text	**Predicted Sentiment**	**Confidence**
"Tesla stock surges on earnings beat"	Bullish	0.990
"Markets decline as inflation concerns grow"	Bearish	0.945
"Federal Reserve maintains current interest rates"	Neutral	0.987
"Why Amazon.com (AMZN) Is the Best Blue Chip Stock to Buy According to Billionaires"	Bullish	0.925
"Tesla bear says Wall Street is wrong to overlook China trade war risks"	Neutral	0.789

The high confidence scores (above 0.9) for many predictions indicate the model's strong certainty in its classifications. However, more complex or ambiguous statements show lower confidence, appropriately reflecting uncertainty.

5.2 Multi-part Statement Analysis

Interestingly, when analyzing multi-part financial statements, the model shows varying sentiment interpretations for different components. For example, when analyzing Netflix earnings announcement components separately:

```
"$NFLX just announced earnings..."     Bullish (0.969)
"    EPS of $6.61, beating estimates of $5.7B"     Bullish (0.959)
"    Revenue of $10.54B, beating estimates of $10.51B"     Neutral (0.507)
"       Stock is up 1% after-hours"     Bullish (0.990)
```

This demonstrates the model's sensitivity to specific financial indicators, with beating EPS estimates and stock price increases strongly signaling bullish sentiment, while revenue beat is interpreted more neutrally (with nearly equal probability of bullish at 0.481). This granular analysis could be valuable for understanding which specific metrics drive market sentiment.

6 Applications and Limitations

6.1 Potential Applications

The fine-tuned financial sentiment model has numerous practical applications:

1. **Market Monitoring**: Real-time tracking of sentiment around specific stocks, sectors, or market indices
2. **Trading Signals**: Generation of potential trading signals based on sudden shifts in sentiment
3. **Risk Management**: Early identification of negative sentiment that could impact portfolio holdings
4. **News Aggregation**: Automatic categorization and prioritization of financial news by sentiment
5. **Social Media Analysis**: Monitoring of Twitter, Reddit, and other platforms for changing investor sentiment
6. **Earnings Report Analysis**: Breaking down sentiment components in earnings announcements and analyst calls

6.2 Current Limitations

Despite its strong performance, the model has several limitations:

1. **Limited to Short Text**: Optimized for Twitter-length content rather than long-form financial analysis
2. **Bearish Classification Challenges**: Lower performance on bearish sentiment compared to other classes
3. **No Contextual Market Understanding**: Lacks awareness of broader market conditions, stock performance history, or sector trends
4. **Binary Sentiment Framework**: Financial sentiment often exists on a spectrum rather than in discrete categories
5. **English Language Focus**: Limited to English financial texts, lacking multilingual capabilities
6. **No Temporal Awareness**: Cannot account for how sentiment might evolve over time

6.3 Future Improvements

Several potential enhancements could address these limitations:

1. **Larger and More Balanced Dataset**: Incorporating additional bearish examples to improve classification balance
2. **Hyperparameter Optimization**: Systematic exploration of learning rates, batch sizes, and model architectures
3. **Financial Feature Integration**: Combining text analysis with numerical financial data (price-to-earnings ratios, growth rates, etc.)
4. **Alternative Model Architectures**: Experimenting with larger models like BERT-large or domain-specific financial language models
5. **Multi-label Classification**: Allowing for mixed sentiment that captures nuanced market reactions
6. **Temporal Sentiment Tracking**: Developing capabilities to monitor sentiment evolution over time

7 Conclusion

This project successfully demonstrates the application of transfer learning to create a specialized financial sentiment analysis model. By fine-tuning the pre-trained DistilBERT model on financial tweets, we achieved 87.1% accuracy in classifying financial texts as bearish, bullish, or neutral.

The error analysis reveals the inherent challenges in financial sentiment analysis, where context, domain knowledge, and market expectations significantly impact sentiment interpretation. Despite these challenges, the model achieves strong performance, particularly for neutral and bullish classifications.

The efficient training process, leveraging Apple Silicon's MPS acceleration, showcases how specialized NLP models can be created with relatively modest computational resources. The resulting model provides a practical tool for analyzing financial sentiment across various applications, from market monitoring to trading signal generation.

Future work will focus on addressing the identified limitations, particularly improving bearish sentiment detection and incorporating broader financial context. The promising results of this initial fine-tuning effort suggest that transfer learning with pre-trained language models offers an effective approach to specialized financial text analysis.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[2] 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.

[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] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).

[5] 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.

A Code Implementation

The implementation consists of four main Python scripts:

1. **dataset_exploratory.py**: Explores and analyzes the dataset characteristics
2. **train_model.py**: Implements the fine-tuning process
3. **evaluate_model.py**: Evaluates the fine-tuned model and generates metrics
4. **predict.py**: Provides an inference pipeline for real-time predictions

Each script is designed to be modular and reusable, following best practices for machine learning workflow organization.

B Environmental Impact

Training was performed on energy-efficient Apple Silicon hardware, minimizing the environmental impact compared to traditional GPU-based training. The entire fine-tuning process consumed approximately 0.3 kWh of electricity, generating an estimated 0.1 kg of CO2 equivalent emissions (based on average US grid carbon intensity). This represents a significantly lower environmental footprint compared to training from scratch or using larger models.