Financial News Sentiment Analysis: From Keywords to FinBERT

Dongjun Park(2022148094)

1 Introduction

This project explores the development of an automated pipeline for classifying sentiment in financial news headlines. Financial markets react instantly to news, making the ability to distinguish between positive, negative, and neutral information crucial for quantitative trading strategies and risk management.

In this work, I implemented a rule-based baseline using a financial lexicon and compared it with a deep learning pipeline utilizing a pre-trained FinBERT model. The results demonstrate how context-aware AI models significantly outperform simple keyword-counting heuristics, achieving an accuracy improvement of approximately 26 percentage points. The project focuses on the inference capabilities of pre-trained models on a single GPU environment.

2 Task Definition

- **Task description:** The objective is to classify financial news headlines into three sentiment categories: **Negative, Neutral, or Positive**.
- **Motivation:** Manually processing the vast volume of daily financial news is infeasible. An automated system that can accurately interpret market sentiment is essential for algorithmic trading. Distinguishing between "market-moving" news and neutral reporting is a key challenge.
- **Input / Output:**
  - *Input*: A text string containing a financial news headline (e.g., "Operating profit rose to EUR 13.1 mn.").
  - *Output*: An integer label representing the sentiment (0: Negative, 1: Neutral, 2: Positive).
- **Success criteria:** The system is considered successful if it achieves significantly higher accuracy than the naive baseline and demonstrates the ability to handle complex linguistic structures (e.g., negations or mixed sentiments like "profit dropped") which typically confuse rule-based systems.

3 Methods

This section includes both the naïve baseline and the improved AI pipeline.

3.1 Naïve Baseline

To establish a performance baseline, I implemented a simple heuristic method based on a financial lexicon (keyword dictionary) without using any machine learning models.

Baseline Implementation

- **Method description:** I constructed a list of common financial positive keywords (e.g., *rise, profit, growth, gain*) and negative keywords (e.g., *loss, fall, decline, drop*). The algorithm counts the occurrences of these keywords in the input sentence.
  - If $Count(Positive) > Count(Negative) \rightarrow$ **Positive**
  - If $Count(Negative) > Count(Positive) \rightarrow$ **Negative**
  - Otherwise $\rightarrow$ **Neutral**
- **Why naïve:** This approach treats a sentence as a "bag of words" and completely ignores syntax and context. It cannot understand negation (e.g., "not a loss") or the relationship between words.
- **Likely failure modes:**
  1. **Mixed Keywords:** In phrases like "profit dropped", the method sees one positive word ("profit") and one negative word ("dropped"), often resulting in a Neutral prediction despite the meaning being clearly negative.
  2. **Morphology:** If the dictionary only contains "fall" but the text uses "fell", the keyword might be missed.

3.2 AI Pipeline

I designed an inference pipeline using **FinBERT**, a BERT model pre-trained specifically on financial texts.

Pipeline Design

- **Models used:** `ProsusAI/finbert` (Hugging Face Transformers). This model is based on the BERT architecture and fine-tuned on the Financial PhraseBank dataset.
- **Pipeline stages:**
  1. **Preprocessing:** Tokenization using the BERT tokenizer (converting text to token IDs, handling padding/truncation).
  2. **Inference:** Passing tokens through the FinBERT model on a GPU (RTX 3090 environment) to obtain logits.
  3. **Post-processing:** Mapping the model's output labels (`positive`, `negative`, `neutral`) to the target integer format (2, 0, 1).
- **Design choices and justification:** I chose FinBERT over a generic BERT model because financial language is highly domain-specific. A generic model might interpret terms like "share" or "bond" in a general sense, whereas FinBERT understands their specific financial implications. I utilized a pre-trained model for inference to maximize performance with limited compute resources.

4 Experiments

4.1 Datasets

I utilized the **Financial PhraseBank** dataset, specifically the `atrost/financial_phrasebank` repository from the Hugging Face Hub. This dataset contains sentences from financial news labeled by domain experts.

- **Source:** Hugging Face Datasets (`atrost/financial_phrasebank`)

- **Total examples:** Approximately 4,840 labeled sentences.
- **Train/Test split:** I utilized a **Test Set of 970 examples** (approx. 20% of the data) to evaluate and compare the performance of the baseline and the AI pipeline.
- **Preprocessing steps:** I applied type-casting to ensure all input data were treated as strings to prevent `ValueError` during tokenization. No extensive text cleaning was performed for the AI pipeline as BERT relies on full sentence context.

4.2 Metrics

- **Metric: Accuracy**
- **Justification:** The task is a multi-class classification problem. Accuracy provides a straightforward measure of how often the model's prediction matches the expert's label, allowing for a direct comparison between the baseline and the AI model.

4.3 Results

The AI pipeline demonstrated a significant performance advantage over the baseline.

Method	Accuracy	Improvement
Naïve Baseline	60.52%	-
AI Pipeline (FinBERT)	**86.49%**	+25.97%

Qualitative Analysis (Success Cases) Below are specific instances where the AI pipeline correctly classified the sentiment while the baseline failed:

1. **Contextual Conflict:**
   - *Input:* "L&T's net **profit** for the whole 2010 **dropped** to EUR 36 million..."
   - *Baseline:* Neutral (Incorrect - canceled out 'profit' and 'dropped').
   - *AI Pipeline:* **Negative** (Correct - understood the profit was decreasing).
2. **Verb Tense:**
   - *Input:* "Net sales **fell** by 5 % from the previous accounting period."
   - *Baseline:* Neutral (Incorrect - missed 'fell').
   - *AI Pipeline:* **Negative** (Correct - handled word variation).
3. **Implicit Sentiment:**
   - *Input:* "Temporary **lay-offs** concern simultaneously at most 80 employees."
   - *Baseline:* Neutral (Incorrect - keyword not in dictionary).
   - *AI Pipeline:* **Negative** (Correct - associated 'lay-offs' with negative sentiment).

5 Reflection and Limitations

The project successfully demonstrated the superiority of deep learning over heuristic methods for context-dependent tasks, achieving an accuracy of 86.49%. However, the implementation process revealed technical challenges in system engineering. I encountered an `OS Error 28 (No space left on device)` due to limited quota on the shared server, which I resolved by redirecting the Hugging Face cache to a local directory. Additionally, resolving version conflicts between `transformers` and the server's `torch` version provided valuable experience in environment management.

Limitations: While Accuracy is a good starting metric, financial datasets are often imbalanced. Using F1-score would have provided better insight. Also, the current pipeline analyzes headlines in isolation; integrating stock price movement (volatility) as a multimodal input would be a logical next step for a real-world trading application.