

Sentiment-Driven Trading Strategy using NLP

Internship Project Report

Author: Priyansh Soni

Project: Sentiment Analysis Trading System

Organization: Zetheta (Internship Project)

Tools: Python, Natural Language Processing, Machine Learning, Streamlit

1. Introduction

Financial markets are highly influenced by news, global events, and investor sentiment. News headlines related to companies, economic policies, technological developments, and global financial conditions can significantly impact stock prices.

This project aims to develop a Sentiment-Driven Trading Strategy that analyzes financial news headlines using Natural Language Processing (NLP) techniques. The system processes news headlines, extracts relevant entities, determines the sentiment of the news, classifies events, and generates trading signals.

By combining machine learning methods with quantitative trading techniques, the system attempts to identify potential market opportunities and evaluate them using historical market data through a backtesting framework.

2. Problem Statement

Traditional trading strategies rely mainly on historical price movements and technical indicators. While these methods are useful, they often ignore important information contained in financial news and market sentiment.

Financial news can strongly influence investor behavior and market movements. Therefore, integrating news sentiment analysis into trading strategies can potentially improve trading decisions.

The objective of this project is to build a system that:

- Collects financial news headlines
- Extracts company entities using NLP techniques
- Performs sentiment analysis on news headlines
- Classifies different types of financial news events
- Generates trading signals based on sentiment
- Evaluates the strategy using a backtesting engine

3. System Architecture

The system is designed as a data processing pipeline that transforms raw financial news into trading signals and performance insights.

Figure 1: System Architecture of the Sentiment-Driven Trading Strategy



The architecture integrates NLP modules, machine learning models, and financial backtesting techniques into a single workflow.

4. Data Collection

Financial news headlines are collected using automated scraping techniques from financial news sources. These headlines serve as the primary dataset for sentiment analysis and event classification.

The collected dataset includes multiple stages of processing:

- Raw news headlines
- Processed headlines
- Entity-extracted headlines
- Sentiment-scored headlines

For backtesting the trading strategy, historical stock price data is retrieved using the yFinance API, which provides market data such as daily closing prices.

5. Natural Language Processing Pipeline

The NLP pipeline is responsible for analyzing and interpreting financial news data.

Named Entity Recognition (NER)

Named Entity Recognition is used to identify companies or organizations mentioned in news headlines. This helps the system associate news sentiment with specific companies.

Example headline:

Apple launches new AI-powered chip.

Entity detected:

Company: Apple

Sentiment Analysis

A machine learning model is used to determine the sentiment of financial news headlines. The sentiment classification categorizes headlines into three classes:

- Positive
- Negative
- Neutral

The model uses TF-IDF vectorization to convert text data into numerical features, which are then used by a classification algorithm to predict sentiment.

Event Classification

Financial news often describes different types of events such as:

- Earnings announcements
- Product launches

- Market updates
- Regulatory changes

Event classification helps provide better context to the sentiment and improves the decision-making process of the trading strategy.

6. Trading Strategy

The trading strategy is designed to generate signals based on the sentiment of financial news.

Decision rules:

Positive Sentiment → BUY signal

Negative Sentiment → SELL signal

Neutral Sentiment → HOLD signal

These signals are applied to historical market data to simulate trading activity.

7. Backtesting Engine

The backtesting engine evaluates the trading strategy by simulating trades using historical market data.

The system performs the following tasks:

- Reads generated trading signals
- Retrieves historical price data
- Executes simulated buy and sell trades
- Tracks portfolio value over time

This simulation allows evaluation of the strategy's profitability and stability.

8. Performance Metrics

The performance of the trading strategy is evaluated using several financial metrics.

Total Return

Measures the overall profitability of the strategy during the testing period.

Sharpe Ratio

Evaluates the risk-adjusted return of the strategy by comparing excess returns relative to volatility.

Maximum Drawdown

Measures the largest decline in portfolio value from a peak to a trough.

9. Interactive Dashboard

An interactive dashboard was developed using Streamlit to visualize the results of the trading strategy.

Dashboard features include:

- Portfolio performance charts
- Sentiment distribution graphs
- Trading signal analysis
- Strategy performance metrics

This interface allows users to interactively explore trading results.

10. Conclusion

This project demonstrates how Natural Language Processing and Machine Learning can be applied to financial trading systems.

By analyzing financial news sentiment and integrating it with trading strategies, the system provides insights into how market sentiment can influence trading decisions.

Future improvements may include:

- Real-time financial news feeds
- Deep learning models for sentiment analysis
- Advanced portfolio optimization techniques
- Expansion to multiple stocks and asset classes

11. References

- Python Documentation
- Scikit-learn Documentation
- SpaCy NLP Library
- Streamlit Documentation
- yFinance API
- Financial Sentiment Analysis Research Papers