WINTER IN DATA SCIENCE 2025

Market Mood and Moves: Sentiment-Driven Stock Prediction

MID TERM REPORT

Introduction

The Market Mood and Moves project aims to study the relationship between financial market movements and the underlying sentiment expressed in news and textual data. Financial markets are not driven purely by numerical fundamentals; investor psychology, expectations and reactions to news play a significant role in shaping short-term and medium-term price movements.

Note- This report outlines the **theoretical understanding, design choices, and learning outcomes** from Weeks 1 and 2. All implementation details and code files are provided separately in the GitHub repository.

Week 1: Market Data and Text Processing Foundations

1 Understanding Financial Market Data

The first step involved understanding the structure and relevance of financial market data. Equity price data typically consists of Open, High, Low, Close, and Volume (OHLCV) values, which capture different aspects of intraday and interday price behaviour.

Key concepts studied include:

- **Returns vs. prices**: Returns are more suitable for statistical analysis as they are stationary compared to raw prices.
- **Market efficiency and information flow**: News and information are reflected in prices with varying delays depending on market conditions.
- **Time alignment issues**: Financial data is time-indexed, making synchronization with textual data a non-trivial task.

2 Motivation for Textual Analysis in Finance

Modern financial markets are highly sensitive to information disseminated through news articles, corporate announcements, analyst reports, social media, online discussions, etc.

While numerical indicators capture what happened in the market, textual data often explains why it happened. This motivates the use of NLP techniques to extract sentiment and contextual signals from unstructured text.

Independent reading focused on:

- Behavioural finance theories
- The role of sentiment in asset pricing
- Differences between general sentiment and finance-specific sentiment

This clarified why generic sentiment tools may be insufficient for financial applications.

3 Text Preprocessing and Linguistic Normalization

Raw text data is noisy and unsuitable for direct analysis. Therefore, a detailed study of text preprocessing techniques is needed.

Key preprocessing steps include:

- **Tokenization**: Breaking text into individual words or tokens.
- **Stop word removal**: Eliminating common words that carry little semantic meaning.
- **Lowercasing and normalization**: Ensuring uniform representation.
- **Lemmatization**: Reducing words to their base or dictionary form.

Understanding these steps from a linguistic perspective was essential to appreciate how preprocessing choices influence downstream sentiment analysis. The preprocessing pipeline was tested on sample financial headlines to validate correctness.

4 Lexicon-Based Sentiment Analysis (VADER)

Lexicon-based sentiment analysis was explored as a baseline approach. VADER (Valence Aware Dictionary and sEntiment Reasoner) assigns sentiment scores based on predefined word dictionaries and heuristic rules.

Key learnings:

- VADER is lightweight and interpretable.
- It performs well on short, informal text.
- However, it lacks domain-specific understanding of financial terminology.

For example, words like “liability” or “volatile” may carry neutral or technical meanings in finance but are treated negatively in general-purpose lexicons. This highlighted the need for more specialized models.

Week 2: Contextual Language Models and Financial NLP

1. Why Static Word Embeddings Fail for the Word "Bank" and the Need for BERT

Traditional static word embeddings such as Word2Vec or GloVe assign a single fixed vector to each word, irrespective of the context in which the word appears. This approach fails for words with multiple meanings.

For example, the word "bank" can appear in different contexts:

- "He deposited money in the bank." → Here, bank refers to a financial institution
- "The river bank was flooded." → Here, banks refers to a geographical landform

In static embeddings, both meanings share the same vector representation, leading to semantic ambiguity. As a result, downstream tasks like sentiment analysis or financial text classification suffer from misinterpretation.

Importance of BERT Architecture:

BERT (Bidirectional Encoder Representations from Transformers) overcomes this limitation by generating contextual embeddings. Instead of assigning a fixed vector to a word, it reads the entire sentence bidirectionally and generates word representations conditioned on surrounding words.

Thus, the embedding for "bank" dynamically changes based on context, allowing accurate semantic understanding which is a critical requirement in financial text analysis, where subtle wording can change interpretation by a lot.

2. Components of BERT Input Embedding

Each token fed into BERT is represented as the sum of three embeddings:

[Token Embedding] + [Segment Embedding] + [Position Embedding]

(a) Token Embeddings represent the actual tokens obtained after WordPiece tokenization

For example: "unbelievable" → "un", "believable"

(b) Segment (Sentence) Embeddings are used to distinguish between Sentence A and Sentence B. It is important for tasks like Next Sentence Prediction

- Tokens in Sentence A → Segment ID 0
- Tokens in Sentence B → Segment ID 1

(c) Positional Embeddings

- Since Transformers lack inherent word order awareness, positional embeddings encode the position of each token in the sequence and also, preserve syntactic structure and word order

The combination of these three embeddings allows BERT to capture semantic meaning, sentence relationships and word order simultaneously.

3. The 80–10–10 Masking Rule in BERT

BERT is trained using Masked Language Modeling (MLM), where 15% of tokens in a sentence are selected for prediction.

Among these selected tokens, 80% are replaced with the [MASK] token, 10% are replaced with a random word and 10% remain unchanged

Why this rule matters?

- It prevents the model from becoming overly dependent on the [MASK] token
- It improves robustness during real-world inference where [MASK] is absent
- It encourages learning deeper contextual representations

This strategy helps BERT generalize better across unseen text.

4. FinBERT: Domain-Specific Adaptation for Financial Text

4.1 Three-Stage Training Pipeline of FinBERT

FinBERT extends BERT by adapting it specifically for the financial domain using a three-stage pipeline:

Stage 1: General Language Pre-training- Here, model is initialized using BERT-base. It is then trained on large general-domain corpora (Wikipedia + BookCorpus) where it learns grammar, syntax and general language structure.

Stage 2: Domain-Adaptive Pre-training- The model is further pre-trained on financial text corpora which helps the model understand financial terminology, jargon and writing style.

For example: “bullish outlook”, “earnings surprise”, “credit downgrade”

Stage 3: Task-Specific Fine-tuning- Model is fine-tuned for specific tasks such as financial sentiment analysis and market news classification for which it uses labelled financial datasets

5. Domain Adaptation in NLP

Domain adaptation refers to the process of transferring a pre-trained language model to a specific domain by exposing it to domain-relevant data.

In the context of financial NLP, general BERT understands everyday language but lacks understanding of financial semantics. Domain adaptation bridges this gap by aligning word

meanings with financial usage and reducing semantic mismatch between training and target data. This step is crucial because financial language differs significantly from general English in tone, vocabulary and structure.

6. Why Specific Datasets Are Used in FinBERT

6.1 Why TRC2-Financial for Pre-training?

TRC2-Financial is a large-scale corpus consisting of financial news articles, market reports and corporate disclosures.

It is ideal for domain-adaptive pre-training because it provides unlabeled but domain-rich data, covers diverse financial contexts and helps the model internalize financial semantics at scale

This stage improves the model's understanding of financial language before any supervised learning.

6.2 Why Financial PhraseBank for Fine-tuning?

Financial PhraseBank is a manually annotated dataset where sentences are labeled as one from positive, negative or neutral.

It is used for fine-tuning because it provides high-quality labelled data, focuses on sentence-level financial sentiment and enables supervised learning for precise classification.

Fine-tuning on this dataset aligns the model's representations with sentiment-oriented financial tasks, such as analyzing market mood from news headlines.

Learning Outcomes and Reflections

Across Weeks 1 and 2, the primary learning outcomes were:

- Developed a foundational understanding of financial market data
- Gained clarity on why sentiment matters in finance
- Learned the strengths and weaknesses of different sentiment analysis approaches
- Understood the importance of domain-specific NLP models
- Appreciated real-world challenges such as data alignment, noise and model assumptions