Financial Sentiment Across News Sources

FINAL REPOT

Liel Ziv - 212541353
Roee Levi - 207496852
Adir Havakuk - 208636597

Problem Description

Application-Centered
Build a system that extracts company names and sentiment from financial news articles to support investment decision-making by investors, analysts, and financial systems

Motivation:

- Financial sentiment impacts stock movements.
- Analysts and investors need automatic tools to interpret large volumes of financial news.
- Sentiment from news provides an alternative signal beyond fundamentals or price charts.

Why It's Challenging

- Articles often contain mixed sentiment across different entities (e.g., positive for one company, negative for another)
- Language in financial news is subtle, vague, and context-sensitive.
- General sentiment models are not trained on financial texts and perform poorly in this domain

Problem Objectives

Explore a Subset of Financial Sentiment Analysis
Focused on extracting and classifying company- and sector-level sentiment from financial news articles.

Develop an Accurate Two-Step Sentiment Pipeline
Step 1: Identify company mentions and ticker symbols using an enhanced NER module (PretrainedFinancialNER)
Step 2: Assign sentiment labels using :

- VADER - lexicon-based baseline (zero-shot, rule-based) - baseline
- FinBERT (pretrained) - zero-shot transformer model - baseline
- FinBERT + LoRA - fine-tuned on gold-labeled data
- DeBERTa + LoRA - alternative fine-tuned model for comparison

Compare Solution Alternatives
Evaluate different sentiment models and aggregation strategies (clause-level vs. full-article)

- Assess the impact of fine-tuning vs. using zero-shot models

Integrate Multiple Knowledge Sources
Incorporate external knowledge from CSVs (ticker list, sectors, aliases) into the NER system

- Aggregate ticker sentiment into sector-level insights

Evaluate Robustness and Performance

- Measure model performance using Macro F1, accuracy, and sector-level agreement
- Examine how label quality, article length, and clause segmentation affect results

Formal Task Specification

Input
- Full-text financial news articles (2020–2025)
- Articles may reference multiple companies and sectors

Output
- Overall article sentiment: {positive, neutral, negative}
- Ticker-level sentiment: {tickers (sentiment, confidence)}
- Sector-level sentiment: {sector - aggregated sentiment}

Evaluation Metrics
- Classification Accuracy, Macro-F1 (overall and per class)
- Average Confidence
- Sector-level sentiment agreement

High-Level Workflow
- **Data Collection**: Collect full financial articles (2020–2025) via EODHD API
- **Labeling**: Create a 3000 articles gold-standard dev set via LLM-assisted and manual annotation.
- **Preprocessing**: Cleaned HTML, removed duplicates, split into clauses, extracted metadata
- **Entity Recognition**: pretrained financial NER model- Extract tickers and sectors - Enhance detection using external CSVs
- **Modeling:**
  - Baseline models: VADER, FinBERT (zero-shot)
  - Fine-tune FinBERT and DeBERTa with LoRA on gold-standard data
- **Aggregation**: Aggregate clause predictions into ticker, sector, and article-level sentiment.
- **Evaluation**: accuracy, macro F1, confidence, and sector-level agreement on the gold set.

Input:
Full article (headline + body)

Output:
.JSON format with:
- overall_sentiment
- tickers[]: symbol, sector, sentiment, confidence, relative weight
- sectors[]: name, sentiment, confidence, relative weight

Example explanation:

The model identifies strong positive sentiment toward **AAPL** and **MSFT**, making **Information Technology** the dominant sector and driving the article's overall **positive** classification.


```
{
  "overall_sentiment": "Positive",
  "sectors": [
    {
      "name": "Information Technology",
      "sentiment": "Positive",
      "relative_weight": 0.65,
      "confidence": 0.90
    },
    {
      "name": "Energy",
      "sentiment": "Neutral",
      "relative_weight": 0.35,
      "confidence": 0.75
    }
  ],
  "tickers": [
    {
      "symbol": "AAPL",
      "sector": "Information Technology",
      "sentiment": "Positive",
      "relative_weight": 0.40,
      "confidence": 0.88
    },
    {
      "symbol": "MSFT",
      "sector": "Information Technology",
      "sentiment": "Positive",
      "relative_weight": 0.25,
      "confidence": 0.92
    },
    {
      "symbol": "XOM",
      "sector": "Energy",
      "sentiment": "Neutral",
      "relative_weight": 0.35,
      "confidence": 0.75
    }
  ]
}
```

Prior art

Source / Title	FinBERT -ZS (Mansouri et al., 2023)	Financial Longformer (Bhandari 2024)	SectorSent (Zhang et al., 2024)
Task solved	Classification task: from financial news headlines to overall sentiment labels (zero-shot)	Classification task: from long financial news articles (up to 4k tokens) to sentiment labels	Per-sector classification task: from SEC 10-K filings to per-sector sentiment scores
Approach / Model	Zero-shot FinBERT-base classifier, headline only	Longformer-base + GRU attention on body	Rule-based ticker→sector + RoBERTa-large per-sentence sentiment
Data	FiQA-2018 + Reuters test set	2 M English news (2010-2023)	85 k SEC 10-K filings
Metrics	Macro-F1	Macro-F1, AUROC	Per-sector AUROC
Results	71 % F1 overall sentiment (no sector granularity)	Handles 4 k-token docs; 68 % F1; 0.83 AUROC	AUC 0.81 (Tech); fragile regex, no calibration

Data Preparation & Description

Source dataset

- Source: EODHD API
- Size: 100k financial news articles (2020 - 2025)
- Format: .Parquet fields include article body, date, tickers, and metadata

Preprocessing Steps

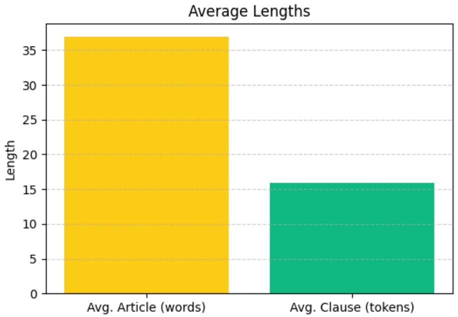
- Removed HTML tags, symbols, duplicates, and low-quality articles
- Split into sentences and semantic clauses using NLTK + custom regex
- Extracted metadata: publication date, tickers, source
- Saved in structured .jsonl format for pipeline processing

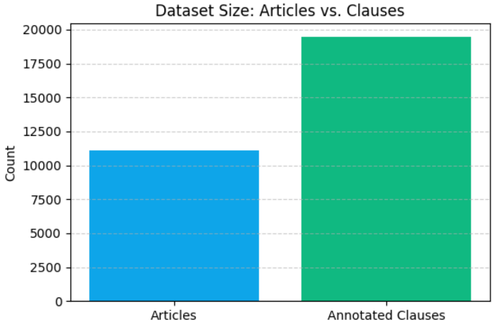
Labeling Process

- LLM-assisted annotation using GPT-4 for sentiment per article and ticker
- Manual validation and correction on a 3000-article subset to form a gold-standard dev set.
- Sentiment classes: {Positive, Neutral, Negative}
- Labels include sentiment class, confidence score, and relative weight

Data Properties / EDA

Data Source file: final_gold_standard_9000.jsonl
Total Articles: 11,127 financial news articles extracted from 100k financial news articles
Input Feature Length
Average article: 35.18 words
Average clause: 15.69 tokens
Total annotated clauses: 14,737

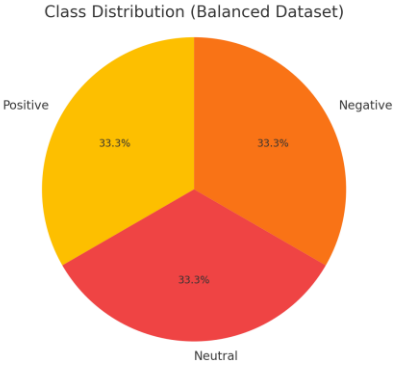


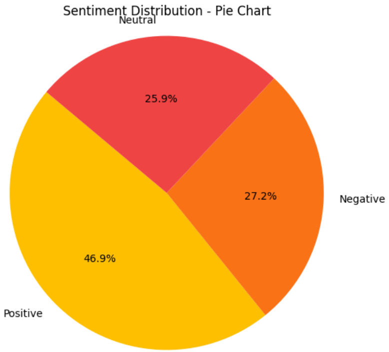


Class Distribution (Before Balancing)
Positive: 46.9% (~5,219)
Negative: 27.2% (~3,027)
Neutral: 25.9% (~2,881)

Findings
- The dataset was imbalanced, with nearly 50% Positive samples
- This risks biased predictions and lower fairness
- We chose to balance all classes before training

Balancing Strategy
Applied undersampling to reduce the size of the Positive and Negative classes





Class Distribution (After Balancing)
Number of Articles: 7359
Positive: 2453
Negative: 2453
Neutral: 2453
Total balanced samples: 8,637

Models and Processing Pipelines

Models Used

Models	Purpose	Method	Notes
FinBERT + LoRA	Fine-tuned FinBERT for financial sentiment	Transformer + LoRA	Fine-tuned on 3K gold articles with LoRA adapters
DeBERTa +LoRA	Fine-tuned DeBERTa for financial sentiment	Transformer + LoRA	Fine-tuned on 3K gold articles with LoRA adapters
FinBERT	Pretrained financial sentiment model	Transformer (frozen)	No fine-tuning; used as zero-shot baseline
VADER	Baseline rule-based sentiment scorer	Zero-shot lexicon-based	No training; used for comparison against BERT models

Training Configuration

Model: finbert, deberta-fin
Data split: Train: 70% Val: 15% Test: 15%
Method: lora (Low-Rank Adaptation)
Epochs: 8 (finbert)
Epochs: 30 (deberta)
Learning rate: 2e-4 (finbert)
Learning rate: 0.0001 (deberta)
Batch size: 32
LoRA rank: 8
LoRA alpha: 64
Optimizer: AdamW
Loss Function: Cross-entropy
Platform: Google Colab Pro
GPU: Tesla T4 (16GB)

Metrics

Metrics Used at Each Step

Task type: Multi-class classification - Positive, Neutral, Negative

Main metrics used:

- Accuracy - overall correctness
- Precision, Recall, F1-score - per class
- Macro avg - equal weight per class
- Weighted avg - weighted by class support
- Confusion Matrix - to visualize misclassifications

How metrics were computed:

During training: on validation set after each epoch

During evaluation: on held-out test set

Why these metrics:

Accuracy alone is not sufficient due to class imbalance

Macro F1 gives better view of performance across all classes, especially Neutral

Confusion matrix helped identify frequent misclassifications

Code Organization

GitHub Repository:
https://github.com/Roee104/FinancialSentimentAnalysis.git

Data Files:
financial_news_2020_2025_100k.parquet – Raw article corpus
final_dataset.jsonl – LLM-assisted + manually reviewed annotation
raw_articles.jsonl - unlabeled financial news scraped from APIs
sprocessed_articles_standard.jsonl / processed_articles_optimized.jsonl – Clause-level outputs from FinBERT
master_ticker_list.csv, ticker_sector.csv - mapping company names

Key Code Modules (in '/core/', '/scripts/', '/utils/'):
core/text_processor.py - cleans, splits, and normalizes articles
core/ner.py - ticker extraction (replaced with `PretrainedFinancialNER`)
core/sentiment.py - sentiment classification using FinBERT + LoRA
core/aggregator.py - aggregates sentiment from ticker → sector
utils/helpers.py - loading/saving JSONL, CSV utilities

Code Organization

Execution Scripts:

‘scripts/run_all_lora_tasks.py’- runs full pipeline (NER → Sentiment → Aggregation → Evaluation)
‘scripts/run_finbert_lora.py’ - trains LoRA adapters on gold-labeled set
‘scripts/run_finetuned_pipelines.py’ - applies the fine-tuned model to new articles
‘scripts/run_experiments.py’ - runs evaluation loop across configurations
scripts/evaluate_on_test.py – computes metrics, prints classification report, plots confusion

Results & Evaluation Files

‘data/processed_articles_finetuned_finbert.jsonl’ — full sentiment predictions
‘data/evaluation_report.csv’ — per-ticker / sector / article metrics
‘outputs/metrics_summary.json’ — aggregated performance
‘outputs/graphs/*.png’ — training curves, sector agreement plots, etc.

Integration with Pipeline Subtasks:

All code components are modular and used within the unified pipeline defined in ‘run_all_lora_tasks.py’, following the steps:

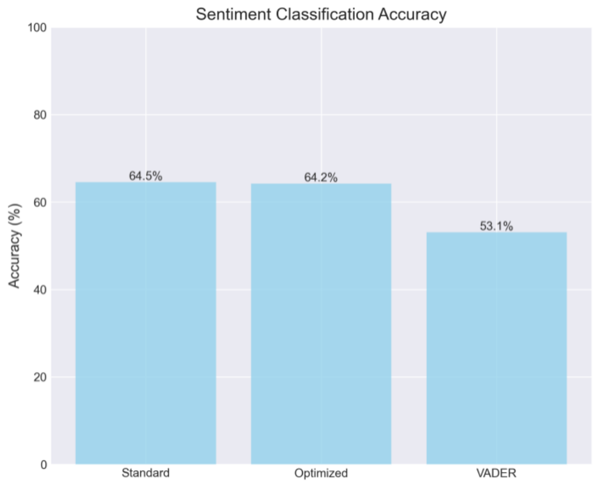
Load article → Extract entities → Classify sentiment → Aggregate → Evaluate

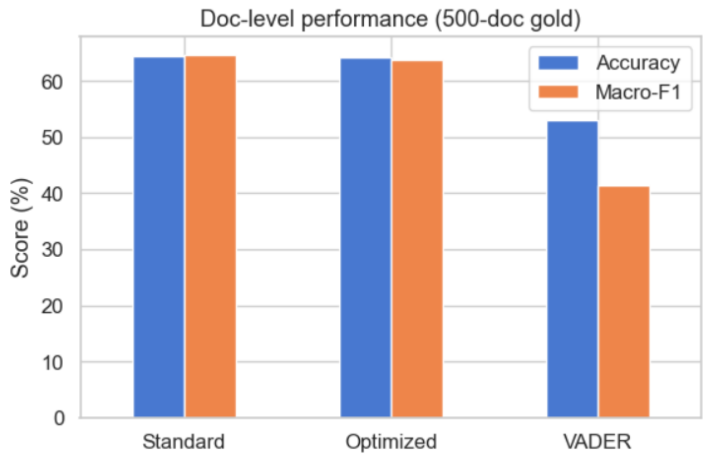
Baseline Results

VADER
Lexicon-based, no financial adaptation
Fails to handle subtle or multi-entity context
Accuracy: 53.1%
Macro-F1: 41.0%
Strong bias toward Positive, poor detection of Negative

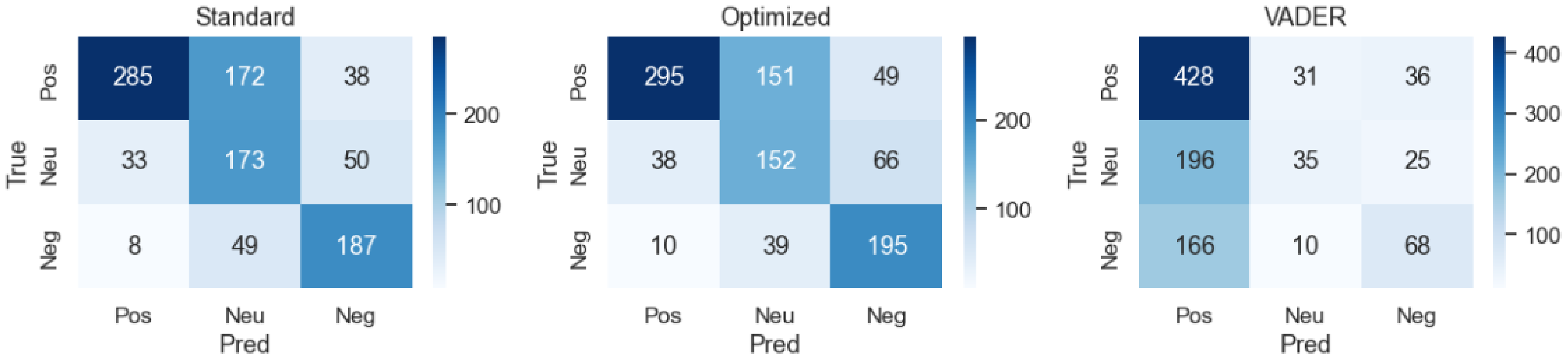
FinBERT – Standard
Pretrained transformer on financial sentiment
Applied to clause-level chunks
Accuracy: 64.5%
Macro-F1: 64.1%
Performs well across all classes, especially Negative

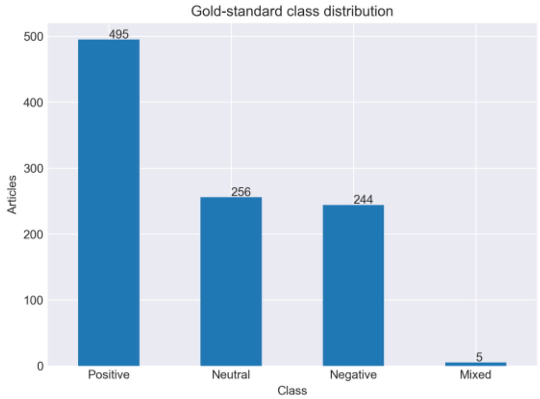
FinBERT – Optimized (Fine-Tuned)
Fine-tuned on 3K LLM-assisted + manually labeled articles
Slight improvement in Neutral & Negative class separation
Accuracy: 64.2%
Macro-F1: 63.2%
More stable predictions across ambiguous clauses





Baseline Results





Main Results and conclusion

Performance Comparison

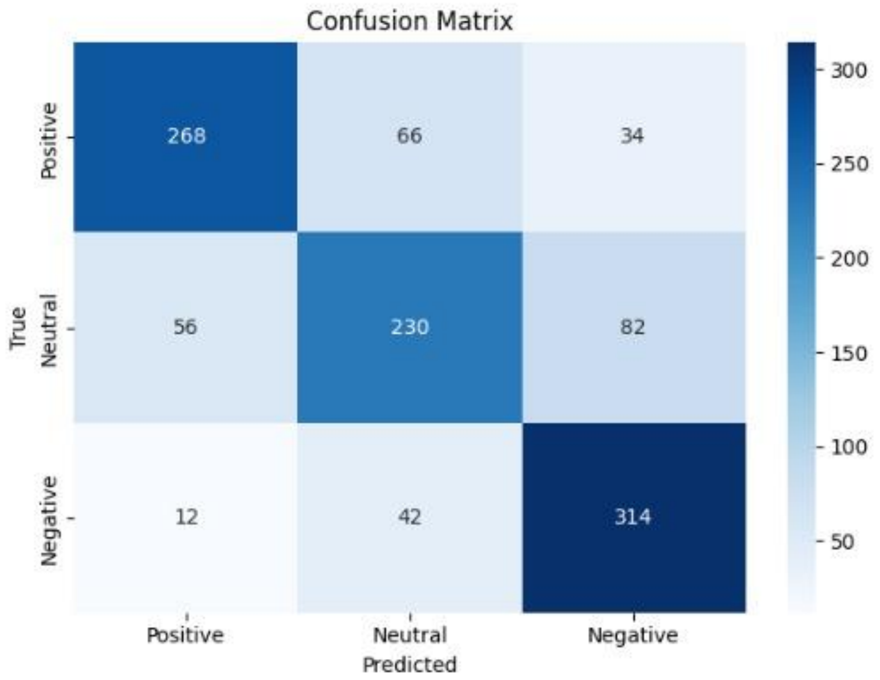
Model	Macro-F1	Accuracy
VADER Baseline	42%~	53.1%
FinBERT Baseline (Optimized)	64%~	64.5%
FinBERT Baseline (Optimized)	63%~	64.2%
FinBERT (LoRA tuned)	73.3%	73.6%
DeBERTa (LoRA tuned)	71.8%	72.2%

Models Results

Finbert result

Classification Report:

	precision	recall	f1-score	support
Positive	0.7976	0.7283	0.7614	368
Neutral	0.6805	0.6250	0.6516	368
Negative	0.7302	0.8533	0.7870	368
accuracy			0.7355	1104
macro avg	0.7361	0.7355	0.7333	1104
weighted avg	0.7361	0.7355	0.7333	1104

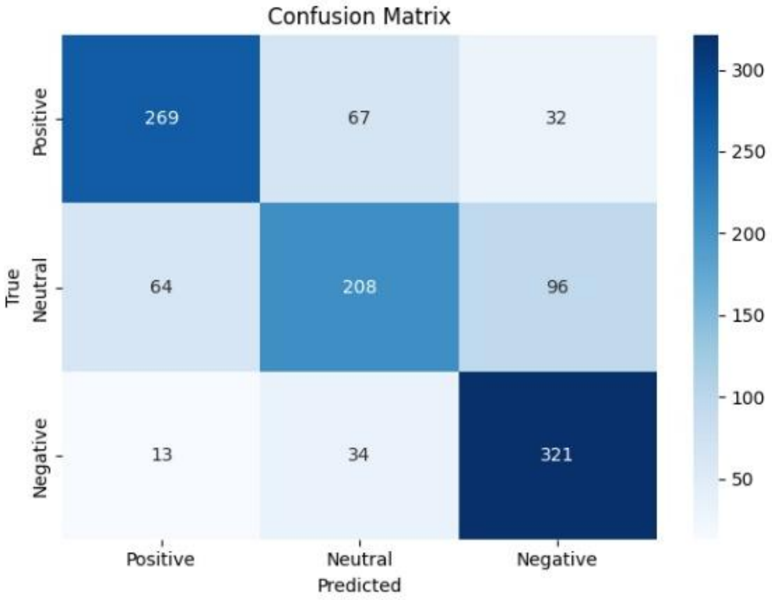


Models Results

DeBerta results

Classification Report:

	precision	recall	f1-score	support
Positive	0.7775	0.7310	0.7535	368
Neutral	0.6731	0.5652	0.6145	368
Negative	0.7149	0.8723	0.7858	368
accuracy			0.7228	1104
macro avg	0.7218	0.7228	0.7179	1104
weighted avg	0.7218	0.7228	0.7179	1104



Models conclusion

Key Findings

- FinBERT, fine-tuned with LoRA and balanced data, achieved the highest performance - 73.6% accuracy and 73.3% macro-F1, consistently outperforming all baselines
- DeBERTa achieved 72.2% accuracy, with slightly lower macro-F1, yet maintained strong performance across all classes.
- FinBERT Baseline models (Standard / Optimized) reached ~64% accuracy but lacked deep contextual reasoning.
- VADER, a rule-based approach, struggled especially in distinguishing Neutral and Negative, with overall accuracy of just 53.1%.

Impact of Methods

- LoRA fine-tuning allowed efficient adaptation with limited compute, while undersampling mitigated class imbalance effectively.
- Confusion matrices show a clear reduction in misclassifications for Negative and Positive classes with transformer models.
- Results highlight the superiority of contextual language models in capturing sentiment nuances in financial texts.

Conclusion

- Our objective - to enhance sentiment classification in financial news - was clearly achieved
- Fine-tuned transformer models outperform both traditional baselines and rule-based systems, offering state-of-the-art results
- These findings support continued investment in task-specific fine-tuning for financial NLP applications

Challenges, Results & Conclusion

Project Scope & Limitations

- Initial Goal: Predict sentiment at three levels - article, ticker, and sector, including rationales.
- Limitation: This scope proved too ambitious due to limited time and data constraints

<u>Final Scope:</u>

Focused on predicting overall article sentiment→ Positive / Neutral / Negative with confidence scores

Data Challenges

- 100k.parquet dataset: All samples labeled neutral → unusable for training
- Label inconsistencies: Many gold samples used "mixed" or non-standard fields
- Imbalanced dataset: Positive class overrepresented in early gold sets
- Heuristic filters for Negative/Neutral added noise - many irrelevant or mislabeled articles

Modeling Issues

LoRA fine-tuning initially failed - models predicted only Neutral.

Suspected causes:

- Class imbalance
- Broken masking/loss
- Misconfigured labels

Span mapping & rationale detection were dropped due to time constraints.

Challenges, Results & Conclusion

Impact of Noise & Fixes

Early failures linked to:

- Label noise (mixed labels)
- Class imbalance
- Poor configuration

Once data was cleaned and rebalanced, models improved significantly

Conclusion

We did not meet the original objective due to complexity and data limitations

We delivered a system that accurately predicts overall sentiment for financial articles

Our results show that domain-specific fine-tuning (FinBERT + LoRA) enables reliable financial sentiment classification - suitable for real-world use

Models conclusion

Processing Pipeline

Data collection
EODHD API(2020-2025)
Full financial articles
External CSVs for tickers, sectors, aliases

↓

Preprocessing
Clean HTML, remove duplicates
Split into clauses, extract metadata
Structured .jsonl output

↓

Entity & Sentiment Extraction
NER using PretrainedFinancialNER + alias matching
Sentiment classification using FinBERT / DeBERTa + LoRA

↓

Confidence-Weighted Aggregation
Clause → Ticker → Sector
Aggregated with confidence weights

↓

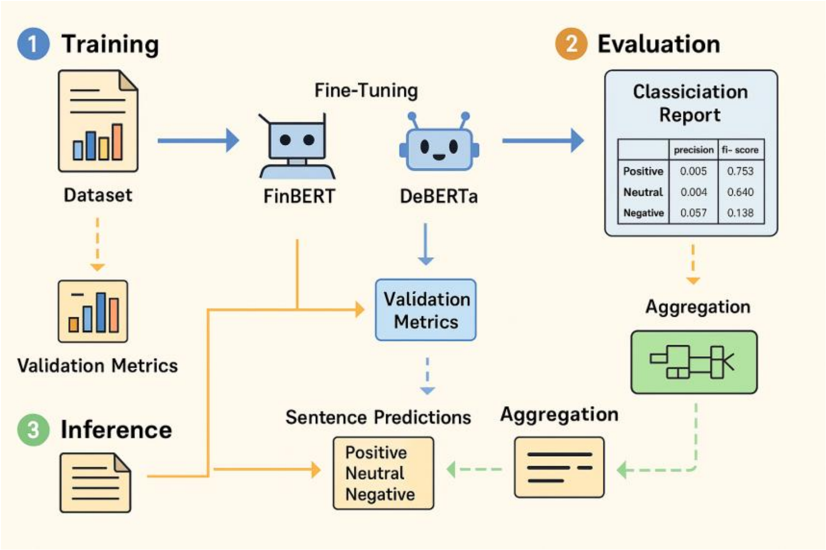
Output
Structured JSONL with ticker-level and sector-level sentiment

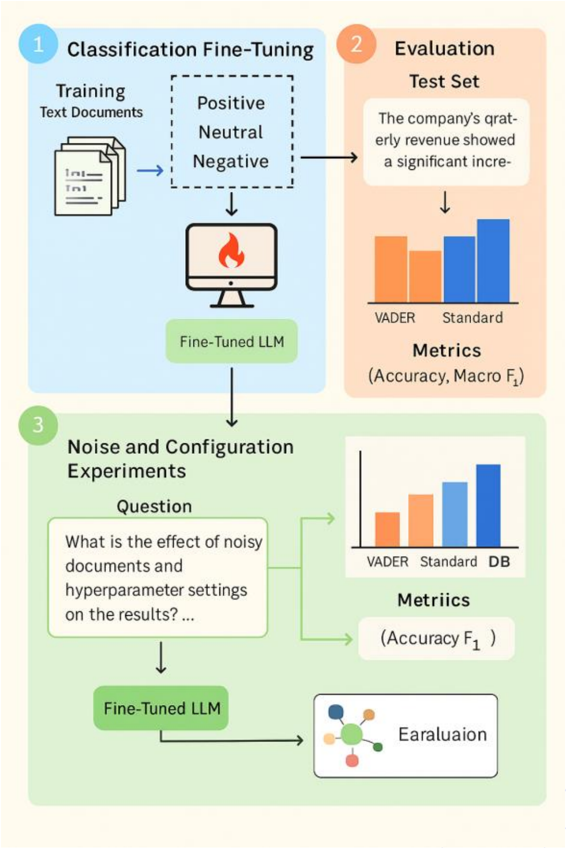
↓

Evaluation
Compare vs. gold standard
Accuracy, macro-F1, recall, confusion-matrix, precision,
Weighted avg, macro avg

Clause-Level Processing

Article
↓
Sentence Split
↓
Semantic Clause


1 Training
2 Evaluation
Dataset
Fine-Tuning
FinBERT
DeBERTa
Classiciation Report
precision
fi- score
Positive 0.005 0.753
Neutral 0.004 0.640
Negative 0.057 0.138
Validation Metrics
Validation Metrics
Aggregation
3 Inference
Sentence Predictions
Aggregation
Positive
Neutral
Negative


1 Classification Fine-Tuning
Training
Text Documents
Positive
Neutral
Negative
Fine-Tuned LLM
2 Evaluation
Test Set
The company's qrat-erly revenue showed a significant incre-
VADER
Standard
Metrics
(Accuracy, Macro F1)
3 Noise and Configuration Experiments
Question
What is the effect of noisy documents and hyperparameter settings on the results? ...
VADER
Standard
DB
Metriics
(Accuracy F1)
Fine-Tuned LLM
Earaluaion