Deciphering Market Sentiments:
A Comparative Study of Sentiment Analysis Models on Financial News

Name: Chia-Mei Liu
UNI: cl4424
Email: cl4424@columbia.edu

1. Abstract

This paper examines the effectiveness of various sentiment analysis models in interpreting sentiments from financial news headlines, with a focus on numerical sensitivity and contextual understanding. Through comparative analysis, we assess the performance of traditional, transformer-based, and specialized financial sentiment analysis tools against general-purpose models. Our findings reveal significant differences in accuracy, reliability, and the tendency for overconfidence across models. Transformer models show promise in surpassing traditional methods, although specialized financial tools offer nuanced advantages in specific contexts. The study underscores the importance of model selection based on the task's complexity and the need for balancing speed and precision in real-time financial sentiment analysis.

Keywords: sentiment analysis, natural language processing, financial news, machine learning, transformers, large language models

2. Introduction

Sentiment analysis in finance is not just about reading the news—it's about capturing the pulse of the market. The way investors feel about news events can drastically sway stock prices, making the ability to quickly and accurately gauge sentiment from financial texts a game-changer for traders and analysts.

Traditionally, sentiment analysis relied on simple word-count methods, but these early techniques often struggled to interpret the complex and nuanced language used in finance. With the advent of machine learning and, more recently, transformer-based models, the ability of computers to understand written sentiments has advanced significantly. These newer models can consider the context and can handle the intricate patterns of financial communication.

This paper examines various sentiment analysis tools, ranging from basic lexicon-based systems to cutting-edge transformer models, including those specifically designed for financial contexts. We assess their performance on financial news to see which models are the most adept at deciphering subtle cues and extracting sentiment from both the numbers and the narrative.

We conduct a series of tests comparing the effectiveness of these tools. The aim is to explore their precision, how they deal with complex financial language, and their speed. This introduction outlines the landscape of sentiment analysis within the financial sector and sets the stage for a deeper investigation into the capabilities of each tool.

2.1. What Readers is Expected to Learn from the Paper?

- How proficiently do various sentiment analysis models identify and interpret sentiment within financial news headlines, specifically regarding their sensitivity to numerical data and their ability to grasp contextual nuances?

- What level of reliability do these models exhibit in their sentiment predictions for financial news, and is there any overconfidence in their assessments?
- Do the latest transformer models demonstrate a significant improvement over traditional sentiment analysis technique?
- How do specialized financial sentiment analysis tools compare to general-purpose models when applied to the same financial texts?

3. Background

Sentiment analysis, which is also known as opinion mining or emotion AI, is a specialized area within the broader field of Natural Language Processing (NLP). Its primary focus is on the extraction and classification of affective states and subjective information from written language. This involves determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions, and emotions expressed by the writer.

The application of sentiment analysis to financial texts is not just a technical challenge but a necessity driven by the nature of markets. The financial market is a complex system that often reacts swiftly to the trader sentiment. As the economist John Maynard Keynes once said, markets can be influenced by waves of sentiment, which, while not always based on logical analysis, are nonetheless legitimate influencers in the absence of solid data. Investors and analysts alike may exhibit herd behavior, driven by a collective mood that can shift from fear to greed, and vice versa, in a matter of hours or days [1].

In light of this, sentiment analysis becomes a powerful tool for gauging the mood of the market. By applying sentiment analysis to news articles, social media feeds, and financial reports, analysts can glean insights into the collective consciousness of the market. This can lead to more informed decisions, as understanding sentiment provides an additional layer of data that goes beyond traditional financial indicators. It can reveal the hidden dynamics that might be at play, offering a glimpse into how sentiment-driven movements could manifest in market trends and volatility.

4. Literature Review

4.1. Traditional Machine Learning Models

The beginning of sentiment analysis as an automated process can be traced back to the application of traditional machine learning (ML) methods around late 1990s. Initially, these techniques mainly focused on statistical models and were applied to datasets with the assumption that the sentiment could be deduced from the presence or absence of certain words. Key algorithms during this phase included Naive Bayes, Support Vector Machines (SVM), and decision trees, which were often utilized for their efficiency and ease of understanding.

Pang and others were among the pioneers to apply machine learning for sentiment classification, demonstrating that standard machine learning techniques could be adapted for sentiment analysis [2]. However, these traditional models encountered challenges, particularly in dealing with the nuances of language such as sarcasm, negation, and context-dependent meanings. As pointed out by Turney one of the major limitations was the reliance on bag-of-words features that ignored the order of words, making it difficult for the model to capture the sentiment accurately in complex sentences [3].

To mitigate these issues, feature selection techniques and dimensionality reduction methods were often employed to improve the performance of ML models. For instance, using term frequency-inverse document frequency (TF-IDF) weighting instead of raw counts helped to emphasize more informative words and downplay common but less informative ones [4]. Moreover, researchers began exploring the incorporation of n-grams, which consider sequences of n words, to capture some local context and word order, leading to modest improvements in performance [5].

4.2. Lexicon-based Approaches

To address some of the limitations of traditional ML, lexicon-based approaches were introduced. These approaches relied on pre-compiled lists of words that had been manually annotated with their sentiment polarity.

VADER (Valence Aware Dictionary and Sentiment Reasoner), a lexicon and rule-based sentiment analysis tool specifically attuned to sentiments expressed in social media, uses a combination of a sentiment lexicon along with five heuristic rules for understanding the context of a sentence. These rules account for factors such as punctuation, capitalization, degree modifiers, and the conjunction 'but' to interpret the sentiment in a nuanced manner [6].

TextBlob, a Python library, combines a standard lexicon of sentiment-polarized words with a machine learning method, namely a Naive Bayes classifier. This classifier is trained on a dataset of movie reviews with labels for sentiment polarity. TextBlob therefore benefits from both the lexicon-based approach and the predictive power of a statistical classifier, allowing it to capture both the polarity of individual words and some context through its probabilistic model.

Despite the progress made with lexicon-based approaches, they are inherently limited by the static nature of their lexicons. They often fail to capture sentiment that emerges from the combination of words or from domain-specific jargon. Moreover, lexicons may become outdated as language evolves, and they might not transfer well across different domains or languages. Additionally, these methods typically do not account for the sentence structure or the overall document context, which can lead to misinterpretation of sentiments when the meaning is highly dependent on the context or when the sentiment is expressed in a subtle or complex manner.

In research conducted by Bonta and others, VADER demonstrated superior performance in classifying movie reviews from Rotten Tomatoes, with an accuracy of 77.0% and an F1 score of 81.6%. In comparison, TextBlob achieved an accuracy of 74.0% and an F1 score of 79.37%. These results indicate that VADER exhibits a higher efficacy than TextBlob in sentiment classification tasks within the context of film reviews [7], which could serve as a proxy for financial news.

4.3. Sequential Neural Network

The advent of deep learning marked a significant leap in sentiment analysis with the introduction of Long Short-Term Memory networks (LSTMs), a type of recurrent neural network. LSTMs excel at processing sequential data, thanks to their unique architecture consisting of gates that regulate the flow of information. These gates—the input, output, and forget gates—enable LSTMs to maintain context over long text sequences, crucial for accurate sentiment analysis.

Introduced by Hochreiter & Schmidhuber in 1997 [8], LSTMs overcame the vanishing gradient problem that plagued earlier RNNs, allowing for more effective learning of dependencies within text. This made them particularly suited for sentiment analysis, as they could remember the sentiment conveyed earlier in a sentence and how it's affected by words that follow, like negations or intensifiers.

4.4. General-Purpose Transformers

The transformer model architecture has revolutionized the field of NLP, including sentiment analysis, by offering a novel way to handle sequential data without the need for recurrent networks. Transformer models use self-attention mechanisms to weigh the importance of each word in a sentence, regardless of their position, enabling parallel processing and capturing context more effectively.

BERT (Bidirectional Encoder Representations from Transformers) was introduced by Devlin et al. in 2018 [9]. as a groundbreaking model that learns language representations by considering both left and right context in all layers. BERT is pre-trained on a large corpus of text through two tasks: masked language modeling and next sentence prediction. Among its various iterations, the 'bert-base-multilingual-uncased-sentiment' model is specifically fine-tuned for sentiment analysis for product reviews across six languages or for further refinement on analogous tasks. In an empirical assessment by Arghya Sahoo, this model demonstrated its efficacy with an accuracy rate of 81.23% and an F1-Score of 80.00%, attesting to its capability in sentiment analysis on Twitter data [10].

Unlike BERT, which is bidirectional, GPT (Generative Pretrained Transformer) is an autoregressive model that predicts the probability of a sequence of words, one word at a time, in a left-to-right fashion. This allows GPT to generate coherent and contextually relevant text sequences, which can be leveraged in tasks like sentiment analysis by understanding the progression of thoughts and feelings in a piece of text [11].

RoBERTa (A Robustly Optimized BERT Pretraining Approach), on the other hand, builds upon BERT's foundations with key changes in the pre-training procedure that significantly improve its performance. RoBERTa trains on much more data for a longer period of time, removes the next sentence prediction objective, and uses dynamic masking for the masked language model [12], which increases the efficiency of the BERT algorithm by 0.023. The optimizations allow RoBERTa to achieve better performance on sentiment analysis tasks, among others [13].

ChatGPT builds upon the GPT architecture. It's fine-tuned with both supervised and reinforcement learning techniques on a diverse range of internet text. ChatGPT is adept at understanding and generating human-like text, making it particularly useful for conversational sentiment analysis. It can follow the sentiment in a conversation, recognizing shifts in tone and responding appropriately [14]. As a general-purpose transformer model, ChatGPT impressively outperformed the finance-domain transformer like FinBERT. GPT-P4 outperformed FinBERT with an accuracy of 78.4% and an F1 of 55.6%, beating FinBERT’s accuracy of 56.1% and F1-score of 55.6%, in an experiment conducted by Georgious Fatouros and others. This study, which examined multiple GPT prompts on forex-related news headlines, exhibited approximately a 35% enhanced performance in sentiment classification and a 36% higher correlation with market returns [15].

BART (Bidirectional a, nd Auto-Regressive Transformers) is another model that combines both bidirectional and auto-regressive transformers. BART is pre-trained by corrupting text with an arbitrary noising function and learning to reconstruct the original text. It is versatile for a number of generation tasks, including summarization and translation, and its robust pre-training makes it effective for sentiment analysis as well [16]. Kostadin Mishev and others conducted an experiment in 2020, and it turned out that Facebook’s BART, fine-tuned and adapted to sentiment analysis in finance, outperformed all the other NLP transformers such as BERT, FinBERT, and RoBERTa when applied to finance data, achieving the best accuracy of 94.7% [13].

In a comparative experiment conducted by Aysun Bozanta and others, which involved comparing different models based on news articles about various stocks including Apple, Amazon, Boeing, and others, RoBERTa achieved the highest scores, with an average accuracy of 85.8% and an F1-score of 85.4%. This performance beat BERT's accuracy and F1-score of 85.2% and far exceeded LSTM's scores, which were 77.2% for both accuracy and F1-score [17].

4.5. Finance-Focuses Transformers

In specialized fields such as finance, the jargon and subtleties of language present unique challenges for sentiment analysis. Models like FinBERT, BloombergGPT, and FinGPT were developed to address the nuances of financial language.

In 2019, Araci demonstrated FinBERT, which represents a specialized adaptation of the BERT model that is pre-trained on financial. It's specifically designed to grasp the intricacies of financial news, reports, and texts, making it adept at analyzing sentiment within the financial domain [18]. Compared to other pre-trained versions of BERT, FinBERT model has achieved a 15% improvement in accuracy in text classification tasks specifically applied to financial texts [19]. In a comparative study by Allen Huang, FinBERT demonstrated superior performance over the standard BERT model, achieving an accuracy of 88.2% compared to BERT's 85.0%, and an F1 score of 87.8% versus 84.2% for BERT. These results were obtained from an experiment that applied both models to the same financial sentiment analysis task, which included a dataset representative of the kind of textual data encountered in financial markets [20].

Bloomberg, the leader of financial market data provider, introduced BloombergGPT in 2023. Trained on the largest domain-specific dataset to date, including over 700 billion tokens from Bloomberg, BloombergGPT leverages a BLOOM-style architecture, features a 50 billion parameter architecture designed to excel in financial analysis and predictions. BloombergGPT is tested on various aspect-specific sentiments, including equity news, social media, and earnings call transcripts. In these evaluations, it attained an F1 score of 62.47, outperforming other advanced models such as GPT-NeoX and BLOOM [21].

Presented as an open-source LLM, FinGPT is data-centric framework which emphasizes the importance of high-quality data acquisition, cleaning, and preprocessing. It features a multi-layered architecture, including a Data Source Layer for real-time market coverage, a Data Engineering Layer optimized for NLP processing of financial data, an LLMs Layer for applying fine-tuning methodologies to adapt to the dynamic financial landscape, and an Application Layer demonstrating FinGPT's practical applications in finance. This technical approach ensures FinGPT not only advances financial analysis precision but also fosters innovation in open finance practices [22]. Upon evaluation by researchers, FinGPT demonstrated a marked superiority over FinBERT in terms of sentiment analysis, posting an Accuracy of 0.88 against 0.725 and an F1 Score of 0.841 compared to 0.668. These results of were derived from a validation set comprising 2,388 financial tweets, including both numerical sensitivity test and contextual understanding [23].

4.6. Finance-Specific Distilled Models

The computational demands of transformer models led to the development of distilled versions, such as DistilRoberta. These models utilize knowledge distillation, where a smaller "student" model learns to mimic a larger "teacher" model, achieving near-parity in performance while significantly reducing size and increasing speed.

Sanh et al. (2019) demonstrated that DistilBERT retains 97% of BERT's performance on benchmark tasks, despite being 40% smaller and 60% faster. This efficiency is crucial in finance, where real-time data processing for market analysis and sentiment assessment is needed. Distilled models thus offer a practical solution for financial applications demanding rapid and accurate NLP capabilities without the high computational costs of larger models [24]. For sentiment analysis, the researchers have developed distilRoberta-financial-sentiment transformer-based model provided by hugging face. Trained on the Financial Phrase Bank dataset, it's a fine-tuned model of distilroberta-base with accuracy of 98.23% [25].

5. Experiment

5.1. Overview

This study aims to evaluate the performance of various machine learning models in analyzing stock sentiment derived from financial news, with a special focus on examining the differences in numerical sensitivity and contextual understanding capabilities.

The models selected for our comparative analysis include:

- TextBlob (NaiveBayesAnalyzer): a model trained on a movie review corpus using nltk [26]
- VADER [6]
- distilRoberta-financial-sentiment transformer-based model ("DistilRoBERTa"): a fine-tuned version of distilroberta-base on the financial_phrasebank dataset [27]
- FinBERT Tone ("FinBERT"): the FinBERT model fine-tuned on 10,000 manually annotated (positive, negative, neutral) sentences from analyst reports [20]
- BART Large Mnli ("BART") : a fine-tuned on the Multi-Genre Natural Language Inference (MNLI) dataset and is well suited for inference tasks like classification [28]
- bert-base-multilingual-uncased model ("BERT"): a bert-base-multilingual-uncased model finetuned for sentiment analysis on product reviews [29]
- ChatGPT4 [30]

These models were selected based on two primary criteria: their immediate availability and ease of access for use without incurring additional costs—for instance, unlike BloombergGPT, which requires a subscription to Bloomberg Terminal, and their readiness for application without requiring further training, unlike models such as LSTM. I also chose the most sophisticated versions of general-purpose transformers designed for sentiment analysis, for example, FinBERT Tone over FinBERT Base.

The inclusion of TextBlob and VADER alongside transformer models has two reasons. Firstly, it sets a baseline model and performance benchmark, demonstrating how much transformer models outperform traditional sentiment analysis techniques in handling the specific challenges of financial news sentiment analysis. Secondly, it underscores the evolution of sentiment analysis tools, from simple lexicon-based methods to advanced neural network architectures, reflecting on the technological advancements and their implications for accuracy and reliability in financial sentiment analysis.

5.2. Methodology and Data

My experiment utilizes a Twitter financial news sentiment validation dataset, accessible via Hugging Face [31], which comprises 2,390 samples annotated with three sentiment labels: Bearish, Bullish, or Neutral. These labels were respectively mapped to Negative, Positive, and Neutral for our analysis. For our analysis, I selectively cleaned 100 samples to ensure no duplicates and reasonable distribution of different sentiment, then curated into two distinct subsets as below [23].:

- Numerical Sensitivity Dataset (Numerical): Composed of 50 samples that include at least two numerical values linked to financial indicators but without explicit sentiment indication words ('raise,' 'fall,' 'increase,' 'decrease'). This subset aims to test the models' capability to infer sentiment from numerical information.

- Contextual Understanding Dataset (Contextual): This subset consists of 50 randomly selected samples from the Twitter Val dataset, which lack clear contextual information necessary for sentiment prediction. It is designed to evaluate the models' ability to understand and analyze sentiment without explicit contextual cues.

	Total	Positive	Neutral	Negative
Numerical Sensitivity	50	23	10	17
Contextual Understanding	50	10	32	8

Table 1: Distribution of Numerical and Contextual Dataset

To facilitate the analysis without access to ChatGPT API, I utilized the following prompt for ChatGPT4, applying a similar approach to interact with other models directly:

```
Please act as a financial expert to determine the sentiment of the below financial news as negative, neutral, or positive, and provide your confidence score in the following format for each headline, no need to provide rationale:

news headline: <news_headline>
sentiment: <sentiment>
confidence score: <confidence_score> (up to .4f)
```

Figure 2: Prompt for ChatGPT4

I used Google Colab to code and run the experiment on a T4 GPU with 16GB of memory, with the exception of the ChatGPT4 component. The code details and analysis result are available in appendix.

```
from transformers import pipeline

def distilroberta(titles, excel_path):
  MODEL = "mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis"
  pipe = pipeline("text-classification", model=MODEL)
  excel_result = []

  start_time = time.time()

  for title in numerical_titles:
    # print(title)
    # print(pipe(title))
    # print("------------------------")
    data = pipe(title)
    excel_result.append((title, data[0]['label'], data[0]['score']))

  end_time = time.time()
  duration = end_time - start_time

  write_to_excel(excel_result, "DistilRoBERTa", excel_path)

  return duration
```

Figure 3: Code example for DistilRoBERTa

5.3. Evaluation and Analysis

5.3.1. Performance Metrics

The performance metrics for my experiment include accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correct predictions overall. Precision and recall are calculated for each of the three different sentiments individually. The F1-score is computed as a weighted average of the precision and recall for each class, taking into account the number of true instances for each sentiment.

5.3.2. Numerical Sensitivity

News Headline	Correct	TextB.	VADER	DistilR.	FinB.	BART	BERT	ChatG.
Madison Square Garden Q2 EPS $3.93 vs. $3.42	Positive	Negative	Neutral	Positive	Neutral	Positive	Negative	Positive
$XLU - XLU Weekly: The Relief Rally From 43.44s Stalls At 58.09s.	Negative	Positive	Positive	Neutral	Neutral	Positive	Negative	Neutral
Pre-tax loss totaled euro 0.3 million, compared to a loss of euro 2.2 million in the first quarter of 2005	Positive	Positive	Negative	Positive	Neutral	Negative	Negative	Positive

Table 3: Example result for the numerical sensitivity dataset

	TextBlob	VADER	DistilRoBERTa	FinBERT	BART	BERT	ChatGPT4
Confidence	0.710	0.144	0.980	0.940	0.765	0.450	0.861
Accuracy	0.500	0.460	0.780	0.580	0.580	0.340	0.740
Precision	0.588	0.612	0.860	0.757	0.567	0.300	0.724
Recall	0.500	0.460	0.780	0.580	0.580	0.340	0.740
F1	0.541	0.525	0.818	0.657	0.573	0.319	0.732

Table 4: Performance metrics for the numerical sensitivity dataset
Notes: The 'Confidence' score is an average, 'Precision' and 'Recall' are weighted averages

The results of the analysis indicate that DistilRoBERTa is the most effective model, achieving the highest scores across performance metrics and demonstrating strong confidence in its predictions. Notably, ChatGPT4, despite not being specifically designed for financial sentiment analysis, surpasses other transformer models in overall performance. FinBERT, which is tailored for financial data, exhibits high confidence levels but does not correspondingly excel in accuracy. Generally, transformer-based models demonstrate superior capability over lexicon-based methods such as TextBlob and VADER. Intriguingly, BERT registers the lowest performance among the transformer models, which highlights the relevance and potential utility of lexicon-based models in certain scenarios. Given its robust performance, DistilRoBERTa emerges as a particularly dependable choice for tasks necessitating precise sentiment classification.

5.3.3. Contextual Understanding

News Headline	Correct	TextB.	VADER	DistilR.	FinB.	BART	BERT	ChatG.
Business tax payments will hit a record low this year	Negative	Positive	Negative	Negative	Positive	Negative	Negative	Neutral
$MESA - Mesa Air EPS beats by $0.01, misses on revenue	Neutral	Positive	Negative	Negative	Negative	Positive	Negative	Neutral
Should You Buy Harmony Gold Mining Co. (HMY)?	Neutral	Negative	Positive	Neutral	Neutral	Positive	Negative	Neutral

Table 5: Example result for the contextual understanding dataset

	TextBlob	VADER	DistilRoBERTa	FinBERT	BART	BERT	ChatGPT4
Confidence	0.799	0.201	0.967	0.956	0.764	0.438	0.734
Accuracy	0.200	0.400	0.500	0.520	0.320	0.180	0.500
Precision	0.697	0.499	0.600	0.639	0.434	0.301	0.540
Recall	0.200	0.400	0.500	0.520	0.320	0.180	0.500
F1	0.311	0.444	0.545	0.573	0.368	0.225	0.519

Table 6: Performance metrics for the contextual understanding dataset
Notes: The 'Confidence' score is an average, 'Precision' and 'Recall' are weighted averages

In the contextual dataset, all models tend to show reduced accuracy and F1 scores compared to the numerical sensitivity dataset, yet their confidence scores are relatively stable. Among the models, DistilRoBERTa, FinBERT, and ChatGPT-4 emerge as the more capable ones for contextual understanding. However, despite DistilRoBERTa and FinBERT having higher confidence scores, their accuracies hover around the 0.5 mark, which could indicate potential overestimation of their predictive reliability. TextBlob's performance is characterized by high confidence but significantly lower scores in other metrics, indicating a probable overconfidence in its contextual comprehension abilities. Both BART and BERT exhibit subpar performance across all evaluated metrics, suggesting that they may struggle with the intricacies of contextual sentiment analysis.

5.3.4. Overall Performance

	TextBlob	VADER	DistilRoBERTa	FinBERT	BART	BERT	ChatGPT4
Confidence	0.755	0.172	0.974	0.948	0.764	0.444	0.798
Accuracy	0.350	0.430	0.640	0.550	0.450	0.260	0.620
Precision	0.643	0.556	0.730	0.698	0.500	0.300	0.632
Recall	0.350	0.430	0.640	0.550	0.450	0.260	0.620
F1	0.426	0.485	0.682	0.615	0.471	0.272	0.625

Table 7: Overall performance metrics
Notes: The 'Confidence' score is an average, 'Precision' and 'Recall' are weighted averages

DistilRoBERTa distinguishes itself as the premier model in our analysis, demonstrating superior performance across all metrics coupled with the highest confidence score, thus establishing its efficacy and dependability for sentiment analysis. ChatGPT4 also demonstrates robustness, maintaining a consistent and reliable performance in both numerical sensitivity and contextual understanding, backed by a solid confidence score. In contrast, FinBERT, despite being tailored for financial sentiment analysis and showing high self-confidence, fails to deliver outstanding results, which raises questions about its practical effectiveness. Interestingly, the anticipated superiority of transformer-based models over lexicon-based methods is not fully realized in our study. BART and BERT show comparable performance to TextBlob and VADER, highlighting that lexicon-based methods still hold significant value and applicability in specific sentiment analysis scenarios.

5.4. Limitations

This study is subject to several limitations that must be acknowledged. First, the selection of models was constrained to those that were readily accessible without additional costs or extensive training, which limits representativity. Secondly, the dataset utilized was sourced from a single one and focused exclusively on Twitter financial news. This presents a limitation in terms of data diversity and may not capture the full spectrum of linguistic nuances present in other financial news sources or social media platforms. Another limitation arises from the dataset size. Although the original dataset contains 2,390 samples, only 100 were selected manually for analysis due to resource constraints. This sample size may not be representative of the larger population of financial news headlines, which could affect the generalizability of the results. Additionally, the methodology employed for evaluating sentiment—using a prompt for models like ChatGPT4—may not perfectly align with other models which are accessed through codes.

6. Conclusion

In this study, I sought to assess the performance of several leading machine learning models in the sentiment analysis of financial news, focusing on their ability to parse numerical data and grasp contextual information. My findings revealed that DistilRoBERTa excelled, demonstrating the highest performance metrics and consistent confidence, establishing itself as the most reliable for high-stakes financial sentiment classification. ChatGPT4 also proved its mettle, showing impressive adaptability and a balanced performance, underscoring its aptitude beyond its primary design. Conversely, FinBERT's performance, while confident, did not align with its accuracy, prompting a reconsideration of its practical utility. The study also illuminated the unexpected competitiveness of lexicon-based methods, with TextBlob and VADER holding their ground against some transformer-based models. This suggests that despite the evolution of complex neural network architectures, traditional sentiment analysis techniques maintain their significance in certain applications. The investigation underscored the necessity of selecting sentiment analysis tools tailored to the specific needs of the task, considering the trade-offs between sophistication, accessibility, and performance.

6.1. What I Learned from this Literature Study and Experiment?

Starting my first research project has been an amazing experience. Choosing a topic that I'm interested in was a challenging at first because I didn't know much about sentiment analysis, NLP, or machine learning. However, as I read more than 20 papers, I began to understand what doing research in computer science is all about. This work has strengthened my confidence about stepping into an area I didn't know before. Constructing Venn diagrams was particularly educational for me, it provides a remarkable approach to visually organize my thoughts from scratch, which in turn facilitated the writing process of this paper. Moreover, my recent fascination with ChatGPT's capabilities led to an anticipation that transformer models would undoubtedly outperform traditional methods. It was surprising to discover that this is not always the scenario in sentiment analysis. This insight has greatly enriched my appreciation for the complexity and nuance of machine learning methodologies

7. Reference

[1] Keynes, John Maynard. "The general theory of employment." *The quarterly journal of economics* 51.2 (1937): 209-223.
https://academic.oup.com/qje/article/51/2/209/1939387

[2] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques." *arXiv preprint cs/0205070* (2002).
https://arxiv.org/abs/cs/0205070

[3] Turney, Peter D. "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews." *arXiv preprint cs/0212032* (2002).
https://arxiv.org/abs/cs/0212032

[4] Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge: Cambridge University Press, 2008.
https://www.cis.uni-muenchen.de/~hs/teach/14s/ir/pdf/19web.pdf

[5] Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing*. 3rd ed., 2021. https://web.stanford.edu/~jurafsky/slp3/

[6] Hutto, Clayton, and Eric Gilbert. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." *Proceedings of the international AAAI conference on web and social media*. Vol. 8. No. 1. 2014.
https://ojs.aaai.org/index.php/icwsm/article/view/14550

[7] Bonta, Venkateswarlu, Nandhini Kumaresh, and N. Janardhan. "A comprehensive study on lexicon based approaches for sentiment analysis." *Asian Journal of Computer Science and Technology* 8.S2 (2019): 1-6.
https://www.researchgate.net/profile/Janardhan-Naulegari/publication/333602124_A_Comprehensive_Study_on_Lexicon_Based_Approaches_for_Sentiment_Analysis/links/5d13452ca6fdcc2462a688ed/A-Comprehensive-Study-on-Lexicon-Based-Approaches-for-Sentiment-Analysis.pdf

[8] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
https://ieeexplore.ieee.org/abstract/document/6795963

[9] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
https://arxiv.org/abs/1810.04805

[10] Sahoo, Arghya, et al. "Comparative Analysis of BERT Models for Sentiment Analysis on Twitter Data." *2023 9th International Conference on Smart Computing and Communications (ICSCC)*. IEEE, 2023.
https://ieeexplore.ieee.org/abstract/document/10335061

[11] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018). https://www.mikecaptain.com/resources/pdf/GPT-1.pdf

[12] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).
https://arxiv.org/abs/1907.11692

[13] Mishev, Kostadin, et al. "Evaluation of sentiment analysis in finance: from lexicons to transformers." *IEEE access* 8 (2020): 131662-131682.
https://ieeexplore.ieee.org/abstract/document/9142175

[14] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html?utm_medium=email&utm_source=transaction

[15] Fatouros, Georgios, et al. "Transforming sentiment analysis in the financial domain with ChatGPT." *Machine Learning with Applications* 14 (2023): 100508.
https://www.sciencedirect.com/science/article/pii/S2666827023000610

[16] Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).
https://arxiv.org/abs/1910.13461

[17] Bozanta, Aysun, et al. "Sentiment analysis of stocktwits using transformer models." *2021 20th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2021.
https://ieeexplore.ieee.org/abstract/document/9680251

[18] Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." *arXiv preprint arXiv:1908.10063* (2019).
https://arxiv.org/abs/1908.10063

[19] Yang, Yi, Mark Christopher Siy Uy, and Allen Huang. "Finbert: A pretrained language model for financial communications." *arXiv preprint arXiv:2006.08097* (2020).
https://arxiv.org/abs/2006.08097

[20] Huang, Allen H., Hui Wang, and Yi Yang. "FinBERT: A large language model for extracting information from financial text." *Contemporary Accounting Research* 40.2 (2023): 806-841.
https://onlinelibrary.wiley.com/doi/full/10.1111/1911-3846.12832

[21] Wu, Shijie, et al. "Bloomberggpt: A large language model for finance." *arXiv preprint arXiv:2303.17564* (2023).
https://arxiv.org/abs/2303.17564

[22] Yang, Hongyang, Xiao-Yang Liu, and Christina Dan Wang. "FinGPT: Open-Source Financial Large Language Models." *arXiv preprint arXiv:2306.06031* (2023).
https://arxiv.org/abs/2306.06031

[23] Zhang, Boyu, Hongyang Yang, and Xiao-Yang Liu. "Instruct-FinGPT: Financial Sentiment Analysis by Instruction Tuning of General-Purpose Large Language Models." *arXiv preprint arXiv:2306.12659* (2023).
https://arxiv.org/abs/2306.12659

[24] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).
https://arxiv.org/abs/1910.01108

[25] Maurya, Shubham, and B. Tech. "A Hybrid Nowcasting Approach for Stock Price Prediction." (2022).
https://www.researchgate.net/profile/Shubham-Maurya-4/publication/370025734_A_Hybrid_Nowcasting_Approach_for_Stock_Price_Prediction/links/6439d2bea08d9a67a49b18ae/A-Hybrid-Nowcasting-Approach-for-Stock-Price-Prediction.pdf

[26] TextBlob: Simplified Text Processing. Version 0.18.0. GitHub (2024).
https://github.com/sloria/TextBlob

[27] DistilRoberta-financial-sentiment. Version Unknown. Hugging Face (2021).
https://huggingface.co/mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis

[28] BART-Large-MNLI. Hugging Face (2020).
https://huggingface.co/facebook/bart-large-mnli

[29] bert-base-multilingual-uncased-sentiment Hugging Face (2020).
https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment

[30] ChatGPT-4. OpenAI (2023.)
https://openai.com/gpt-4

[31] Twitter Financial News Sentiment Dataset. Version 1.0.0. Hugging Face (2022)
https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment

8. Appendix

GitHub repository, which includes the codes, dataset, analysis details:
https://github.com/madiliu/stock_sentiment_analysis

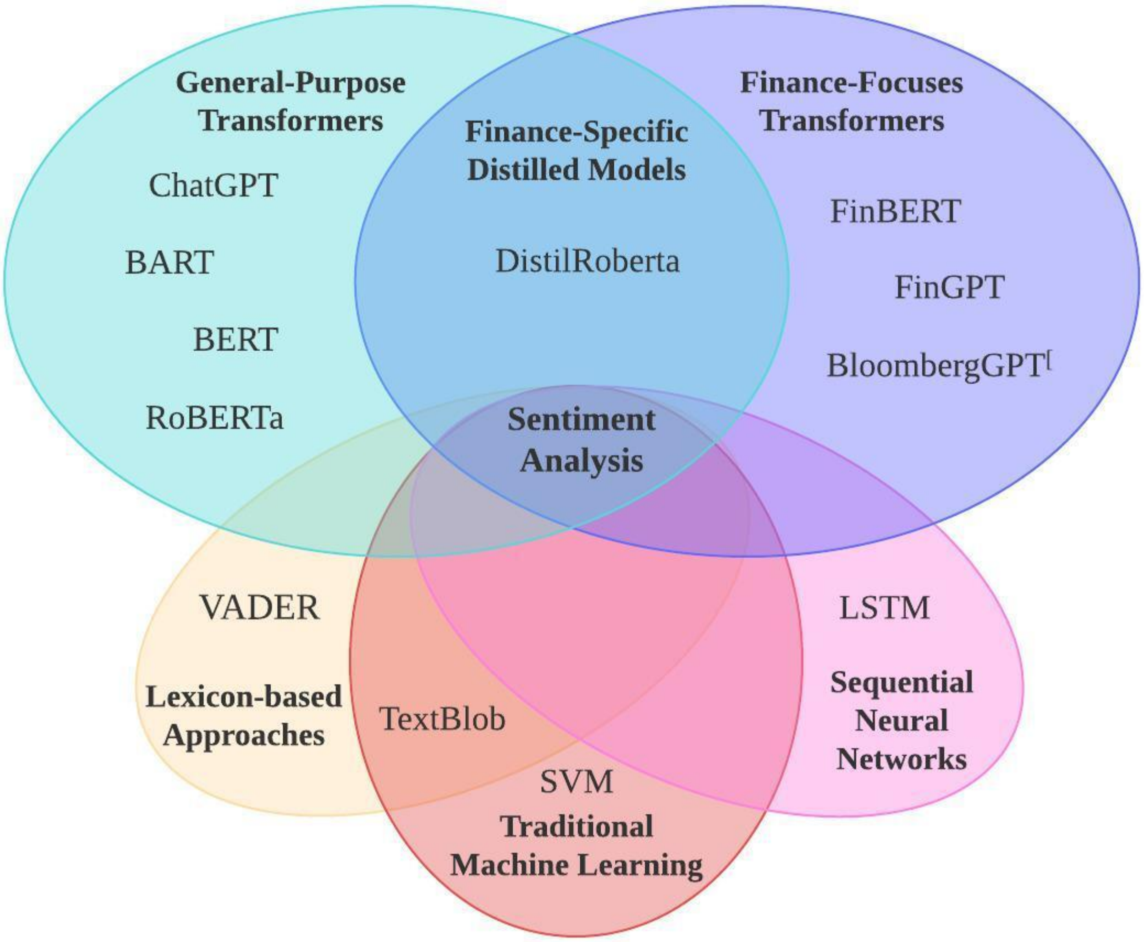

Figure 1: Venn Diagram
Note: I have updated the Venn Diagram by categorizing VADER and TextBlob under the newly added lexicon-based approaches and by incorporating SVM into the section for traditional machine learning

Figure 2: Codes

Set up

Write to excel

```
import time
import os
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')

numerical_excel_path = '/content/drive/My Drive/Colab Notebooks/numerical_vTest.xlsx'
contextual_excel_path = '/content/drive/My Drive/Colab Notebooks/contextual_vTest.xlsx'


def write_to_excel(result, sheet_name, excel_path):
    # Define the column names
    df = pd.DataFrame(result)
    df.columns = ['title', 'sentiment', 'confidence']

    with pd.ExcelWriter(excel_path, mode='a', engine='openpyxl', if_sheet_exists='new') as writer:
        df.to_excel(writer, sheet_name=sheet_name, index=False)
```

```
!ls -l {numerical_excel_path}
```

```
!unzip -t {numerical_excel_path}
```

Reading news data - Twitter Value

```
numerical_filename = 'Numerical.xlsx'
search_path = '/content/drive/My Drive/Colab Notebooks'

file_path = None

# read numerical
for root, dirs, files in os.walk(search_path):
    if numerical_filename in files:
        file_path = os.path.join(root, numerical_filename)
        break
        numerical_df = pd.read_excel(file_path)

numerical_df = pd.read_excel(file_path)
# numerical_df.head()

numerical_titles = numerical_df.iloc[:, 0]
numerical_titles
```

```
contextual_filename = 'Contextual.xlsx'
search_path = '/content/drive/My Drive/Colab Notebooks'

file_path = None

# read contextual
for root, dirs, files in os.walk(search_path):
    if contextual_filename in files:
        file_path = os.path.join(root, contextual_filename)
        break
    # contextual_df = pd.read_excel(file_path)
contextual_df = pd.read_excel(file_path)

contextual_titles = contextual_df.iloc[:, 0]
contextual_titles
```

Sentiment analysis function for each models

1. TextBlob - NaiveBayesAnalyzer

`NaiveBayesAnalyzer` : uses `nltk` , this model was trained on a movie review corpus

```
!python -m textblob.download_corpora
```

```
from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer

def textblob(titles, excel_path):
  excel_result = []

  start_time = time.time()
  for title in titles:
      blob = TextBlob(title, analyzer=NaiveBayesAnalyzer())
      # print(title)
      # print(blob.sentiment)
      # print("------------------------")
      # Unpack the sentiment into its own tuple
      sentiment_data = (blob.sentiment.classification, blob.sentiment.p_pos, blob.sentiment.p_neg)
      # Extend the original tuple (title, sentiment) with the unpacked sentiment data
      excel_result.append((title,) + sentiment_data)

  end_time = time.time()
  duration = end_time - start_time

  # The rest of your code remains the same
  df = pd.DataFrame(excel_result, columns=['title', 'classification', 'p_pos', 'p_neg'])

  # Now you can write df to the Excel file as before
  with pd.ExcelWriter(excel_path, mode='a', engine='openpyxl', if_sheet_exists='new') as writer:
      df.to_excel(writer, sheet_name="TextBlob")

  return duration
```

2. VADER

```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
from zipfile import BadZipFile

custom_nltk_path = '/content/drive/My Drive/Colab Notebooks/nltk_data'
nltk.data.path.append(custom_nltk_path)

try:
    nltk.download('vader_lexicon', force=True)
except BadZipFile as e:
    print(f"Failed to open: {e.filename}")


def vader(titles, excel_path):

  # Initialize the VADER sentiment intensity analyzer
  sia = SentimentIntensityAnalyzer()

  excel_result = []

  # Iterate through each title and perform sentiment analysis
  start_time = time.time()
  for title in titles:
      # Get the sentiment scores
      sentiment_scores = sia.polarity_scores(title)

      # Determine sentiment type and confidence
      compound_score = sentiment_scores['compound']
      if compound_score >= 0.05:
          generalized_sentiment = 'positive'
      elif compound_score <= -0.05:
          generalized_sentiment = 'negative'
      else:
          generalized_sentiment = 'neutral'

      # Confidence interpretation
      confidence = abs(compound_score)  # The absolute value of the compound score as confidence

      # Print the result in the specified format
      # print(title)
      # print(f"{{'label': '{generalized_sentiment}', 'score': {confidence:.4f}}}")
      # print("------------------------")
      excel_result.append((title, generalized_sentiment, confidence))


  end_time = time.time()
  duration = end_time - start_time

  write_to_excel(excel_result, "VADER", excel_path)

  return duration
```

3. fine-tuned DistilRoBERTa (Transformers!)

This is a transformer model that has been fine-tuned on the financial phrasebank dataset.

```python
from transformers import pipeline

def distilroberta(titles, excel_path):
  MODEL = "mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis"
  pipe = pipeline("text-classification", model=MODEL)
  excel_result = []

  start_time = time.time()

  for title in numerical_titles:
    # print(title)
    # print(pipe(title))
    # print("------------------------")
    data = pipe(title)
    excel_result.append((title, data[0]['label'], data[0]['score']))

  end_time = time.time()
  duration = end_time - start_time

  write_to_excel(excel_result, "DistilRoBERTa", excel_path)

  return duration
```

4. FinBERT - 2022

```python
from transformers import BertTokenizer, BertForSequenceClassification
from transformers import pipeline

HF_TOKEN = "HF_TOKEN"

def finbert(titles, excel_path):
  finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone',num_labels=3)
  tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')

  sentiment_analysis = pipeline("sentiment-analysis", model=finbert, tokenizer=tokenizer)

  titles = [str(title) for title in titles]
  sentiment_results = sentiment_analysis(titles)

  start_time = time.time()
  excel_result = []

  for title in titles:
    sentiment_result = sentiment_analysis(title)
    # print(title)
    # print(sentiment_result)
    # print("------------------------")
    excel_result.append((title, sentiment_result[0]['label'], sentiment_result[0]['score']))

  end_time = time.time()
  duration = end_time - start_time

  write_to_excel(excel_result, "FinBERT", excel_path)

  return duration
```

5. BART Large Mnli

```
!pip install transformers
```

```
from transformers import pipeline

def bart(titles, excel_path):
  # Initialize the zero-shot classification pipeline
  classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")

  # Candidate labels for sentiment analysis
  candidate_labels = ["positive", "negative", "neutral"]

  # Classify the sentiment of each title
  start_time = time.time()
  excel_result = []

  for title in numerical_titles:
      # Perform zero-shot classification
      result = classifier(title, candidate_labels)
      sentiment = result['labels'][0]  # The label with the highest score
      confidence = result['scores'][0]  # The confidence of the top label

      # Print the result
      # print(title)
      # print(f"[{{'label': '{sentiment}', 'score': {confidence:.4f}}}]")
      # print("------------------------")
      excel_result.append((title, sentiment, confidence))

  end_time = time.time()
  duration = end_time - start_time

  write_to_excel(excel_result, "BART Large Mini", excel_path)

  return duration
```

6. BERT

since BERT itself doesn't natively support zero-shot classification in the way models like facebook/bart-large-mnli do, we'll use a model that has been fine-tuned for sequence classification tasks and can infer sentiment directly. One such model is nlptown/bert-base-multilingual-uncased-sentiment, which can classify text into positive, negative, or neutral sentiments and is available on the Hugging Face model hub.

```
from transformers import pipeline

def bert(titles, excel_path):
  # Initialize the sentiment analysis pipeline with a BERT model fine-tuned for sentiment
  # Here we use 'nlptown/bert-base-multilingual-uncased-sentiment' which outputs sentiment scores
  classifier = pipeline("sentiment-analysis", model="nlptown/bert-base-multilingual-uncased-sentiment")

  excel_result = []

  # Classify the sentiment of each title
  start_time = time.time()

  for title in numerical_titles:
      # Perform sentiment analysis
      results = classifier(title)

      # The 'nlptown/bert-base-multilingual-uncased-sentiment' model provides labels and scores directly
      sentiment = results[0]['label']
      # Convert labels provided by this model into a more general form (positive, negative, neutral)
      if sentiment in ["1 star", "2 stars"]:
          generalized_sentiment = "negative"
      elif sentiment in ["4 stars", "5 stars"]:
          generalized_sentiment = "positive"
      else:
          generalized_sentiment = "neutral"
      confidence = results[0]['score']

      # Print the result
      # print(title)
      # print(f"[{{'label': '{generalized_sentiment}', 'score': {confidence:.4f}}}]")
      # print("-------------------------")
      excel_result.append((title, generalized_sentiment, confidence))

  end_time = time.time()
  duration = end_time - start_time

  write_to_excel(excel_result, "BERT", excel_path)

  return duration
```

Conducting sentiment analysis

```python
# 1. Textblob
numerical_textblob_duration = textblob(numerical_titles, numerical_excel_path)
contextual_textblob_duration = textblob(contextual_titles, contextual_excel_path)
print(f"numerical_textblob_duration: {numerical_textblob_duration} seconds")
print(f"contextual_textblob_duration: {contextual_textblob_duration} seconds")

# 2. VADER
numerical_vader_duration = vader(numerical_titles, numerical_excel_path)
contextual_vader_duration = vader(contextual_titles, contextual_excel_path)
print(f"numerical_vader_duration: {numerical_vader_duration} seconds")
print(f"contextual_vader_duration: {contextual_vader_duration} seconds")

# 3. DistilRoBERTa
numerical_distilroberta_duration = distilroberta(numerical_titles, numerical_excel_path)
contextual_distilroberta_duration = distilroberta(contextual_titles, contextual_excel_path)
print(f"numerical_distilroberta_duration: {numerical_distilroberta_duration} seconds")
print(f"contextual_distilroberta_duration: {contextual_distilroberta_duration} seconds")

# 4. FinBERT
numerical_finbert_duration = finbert(numerical_titles, numerical_excel_path)
contextual_finbert_duration = finbert(contextual_titles, contextual_excel_path)
print(f"numerical_finbert_duration: {numerical_finbert_duration} seconds")
print(f"contextual_finbert_duration: {contextual_finbert_duration} seconds")

# 5. BART Large Mini
numerical_bart_duration = bart(numerical_titles, numerical_excel_path)
contextual_bart_duration = bart(contextual_titles, contextual_excel_path)
print(f"numerical_bart_duration: {numerical_bart_duration} seconds")
print(f"contextual_bart_duration: {contextual_bart_duration} seconds")

# 6. BERT
numerical_bert_duration = bert(numerical_titles, numerical_excel_path)
contextual_bert_duration = bert(contextual_titles, contextual_excel_path)
print(f"numerical_bert_duration: {numerical_bert_duration} seconds")
print(f"contextual_bert_duration: {contextual_bert_duration} seconds")
```

Figure 3: Example for conducting sentiment analysis using ChatGPT4

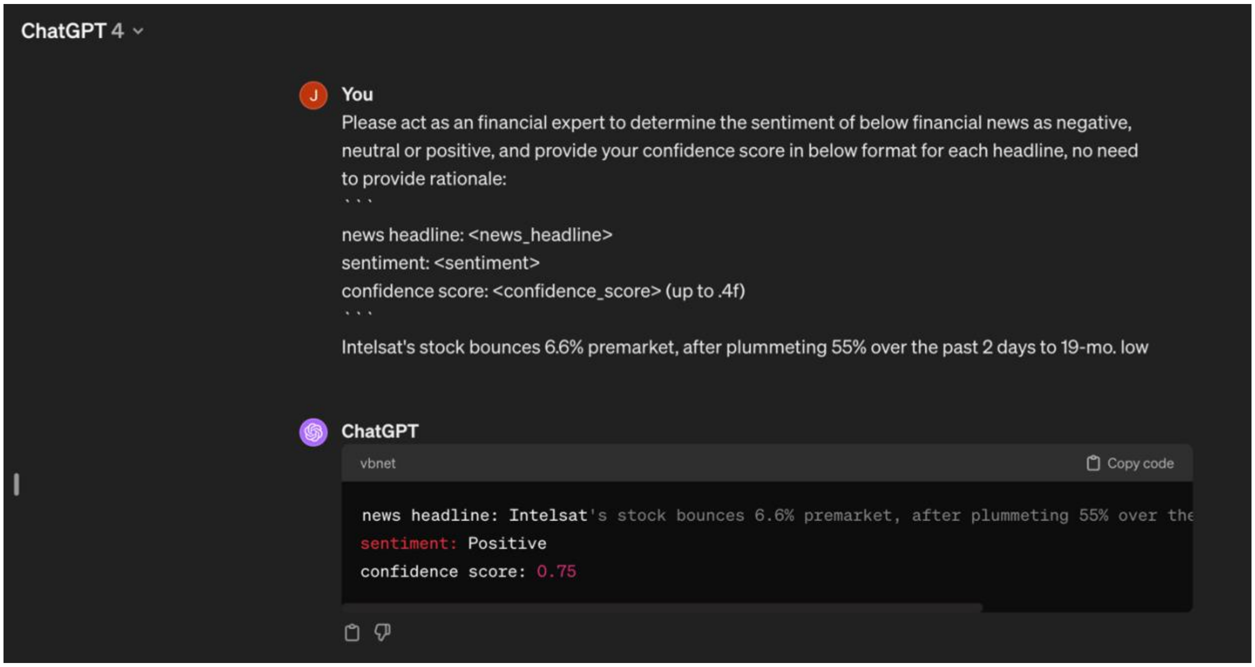

Table 1: Numerical Sensitivity Dataset

No	Title	Sentiment
1	LendingTree price target cut to $350 from $400 at SunTrust Robinson Humphrey	negative
2	NortonLifeLock stock price target cut to $18 from $25 at Deutsche Bank	negative
3	Norwegian Cruise stock price target cut to $48 from $55 at CFRA	negative
4	Novus Therapeutics stock price target cut to $2.50 from $3.75 at Ascendiant Capital	negative
5	Nutrien stock price target cut to $54 vs. $55 at BofA Merrill Lynch	negative
6	Solarworlds stock price target cut to $23 vs. $24 at Instinet	negative
7	Wayfair price target lowered to $110 from $120 at Stifel, buy rating maintained	negative
8	10Y Yield LOD 1.7620%	negative
9	$CAE: CAE temporarily suspends dividend and share repurchase plan; lays off 2,600 of its 10,500 employees; to...	negative
10	BJ's Wholesale lowers 2019 sales outlook to approximately $12.9 bln from $12.9 bln-$13.2 bln	negative
11	Kohl's Q3 EPS 78 cents vs. 98 cents	negative
12	Kohl's Q3 revenue $4.625 bln vs. $4.628 bln; FactSet consensus $4.399 bln	negative
13	U.S. Dollar Index Futures (DX) Technical Analysis ‚Äì Looking for Break into Retracement Zone at 97.630 to 97.472	negative
14	Only 11% of quant funds are ahead of their benchmark YTD, underperforming by 3.2ppt: BofA	negative
15	India's state-run banks report fraud of more than $13 billion in 6 months	negative
16	Dow closes about 102 points, 0.4%, lower	negative
17	$XLU - XLU Weekly: The Relief Rally From 43.44s Stalls At 58.09s.	negative
18	BJ's Wholesale Q3 adj. EPS 41 cents; FactSet consensus 40 cents	neutral
19	Estee Lauder Q2 adj. EPS $2.11; FactSet consensus $1.90	neutral
20	Kellogg 2020 FactSet EPS consensus is $4.03; 2019 currency-neutral adj. EPS was $4.00	neutral
21	A couple claims a 500-pound, $280-million emerald was destroyed when their house burned down in California's worst-,	neutral
22	How a 29-year-old YouTube millionaire making up to $220,000 a month spends his money. (via @CNBCMakeIt),	neutral
23	$SRNE: Sorrento Therapeutics says rejects cash acquisition proposal for between $3.00-5.00/share	neutral
24	$ES equilibrium is 3107.50-3108 Long if either open is above 3108 and goes bid above the open. Look for some resis,	neutral
25	40% YTD Return On 18 Stocks: Andersen Capital Mgmt. (Radio)	neutral
26	$SUMRX: Gapping up/down: ABB +4%, MCHP +3%, NVO +3%, HUM +2% and CMG +1.6% after earnings, RDS.A +2% and TOL +2%...¬†	neutral
27	$SUMRX: Gapping up/down: DOCU +10.5%, ULTA +9%, AOBC +7%, BIG +7%, CLDR +6.5% and CRWD +5% after earnings, TTD +5%.,	neutral
28	Boeing 737 Max Net Orders Rise to -73 YTD in Nov. From -93 Oct. Yes, negative	positive
29	AT&T to raise quarterly dividend by 2% to 52 cents a share, retire $100 million of stock in Q1	positive
30	Madison Square Garden Q2 EPS $3.93 vs. $3.42	positive
31	Dec. gold climbs $2.40, or 0.2%, to settle at $1,474.30/oz	positive
32	Feb. gold climbs $8.90, or 0.6%, to settle at $1,481.20/oz	positive
33	$ECONX: December University of Michigan Consumer Sentiment- Prelim 99.2 vs 96.5	positive

34	$ECONX: November Nonfarm Payrolls 266K vs 182K	positive
35	$ECONX: October Wholesale Inventories M/M -0.7% vs +0.2%	positive
36	Consumer credit $18.9BN, Exp. $16BN, Last $9.6BN	positive
37	Housing starts climb 3.8% in October to 1.314 million rate	positive
38	Dow up 26 points, or 0.1%; S&P 500 adds 1 point	positive
39	Stock market live updates: Dow futures up 800 points, set to add to 1,600-point Monday rally	positive
40	STOCKS AT THE OPEN: - Dow up 3.90% - Nasdaq up 2.68% - S&P up 2.76%	positive
41	STOCKS SURGE INTO THE CLOSE: - Dow up 7.59% - Nasdaq up 7.35% - S&P up 6.95%	positive
42	$NYMT out 1.82 from $1.60 @here	positive
43	$RH clears prior high of 192 after gapping up over its 50-day on volume. On watch how it acts around 200.	positive
44	9:45 am (it's early) $SPY 272.99 1,520 STOCKS are up from the open vs 1,700 that are down from the open. $IWM 116.87	positive
45	ChartTrader's recent $YMm0 short took off for a +700 pt gain in less than 30 minutes. Learn more and #tradesmarter,	positive
46	Tesla $TSLA with 1400 January 2022 $1280 calls bought to open $33.50 to $36 for around $5M. And with that, I am off...	positive
47	Intelsat's stock bounces 6.6% premarket, after plummeting 55% over the past 2 days to 19-mo. low	positive
48	Sanofi Q4 top-line up 7%; shares up 3% premarket	positive
49	Transocean up 4% on new $91M contract	positive
50	Pre-tax loss totaled euro 0.3 million, compared to a loss of euro 2.2 million in the first quarter of 2005	positive

Table 2: Contextual Understanding Dataset

No	Title	Sentiment
1	Trans Mountain Costs to Increase 70% to $9.5 Billion, CBC Says	negative
2	Dish TV India Limited Just Missed Earnings With A Surprise Loss - Here Are Analysts Latest Forecasts	negative
3	$NSTG - NanoString Q1 top line shy of guidance amid COVID-19	negative
4	Caterpillar Inc.'s 2020 profit outlook trailed analysts' estimates, with the heavy-equipment saying continued econo...	negative
5	Business tax payments will hit a record low this year	negative
6	When Chinese Consumers Stay Home, the World's Retailers Take Hit	negative
7	Amazon files suit, challenging Pentagon‚Äôs $10 billion cloud contract to Microsoft	negative
8	The situation of coated magazine printing paper will continue to be weak	negative
9	$EPR: EPR Properties announces sale of charter school portfolio for approximately $454 mln; lowers FY19 FFO guidance	neutral
10	$ITRN - Margins suffer at Ituran Location in Q3	neutral
11	Alibaba on track to raise $12.9 billion in Hong Kong listing - CNN	neutral
12	Amazon will spend more than $35 billion on shipping costs this year, more than twice what it spent two years ago. T,	neutral
13	Boeing announces additional order for 737 MAX planes	neutral
14	Boeing gets 10 additional orders for 737 MAX 8 planes valued at $1.2 billion from SunExpress	neutral
15	Hedge fund Two Sigma builds private equity power with $1.2bn debut Sightway Capital fund close	neutral
16	AT&T takes on $5.5 billion loan to boost 'financial flexibility'	neutral
17	B. Riley offers $50M of senior notes	neutral
18	Match Group announces $500M senior notes offering	neutral
19	$ERC - Wells Fargo Multi-Sector Income Fund declares $0.09888 dividend	neutral
20	$FRT - Federal REIT FFO misses by $0.02, beats on revenue	neutral
21	$MESA - Mesa Air EPS beats by $0.01, misses on revenue	neutral
22	$MOH - Molina Healthcare EPS beats by $0.07, misses on revenue	neutral
23	Domtar EPS misses by $0.05, revenue in-line	neutral
24	Molina Healthcare EPS beats by $0.07, misses on revenue	neutral
25	Nokia EPS beats by ‚Ç¨0.02, misses on revenue	neutral
26	Philip Morris profit and sales top estimates but tobacco company offers soft guidance	neutral
27	SoftBank's Oyo reveals over $330 million annual loss; revenue surges	neutral
28	Virco Manufacturing EPS beats by $0.14, misses on revenue	neutral
29	Dollar mixed versus major rivals after January jobs report	neutral
30	Coronavirus case tally: 565 deaths, 28,256 cases	neutral
31	Ukraine still hasn't received $35 million of approved military aid: report	neutral
32	Casper prices its IPO at $12 a share, the low end of range	neutral
33	OneWater Marine raised $55.4 mln in the IPO	neutral
34	PPD's stock indicated in early going to open at $30, or 11% above $27 IPO price	neutral
35	$TD $SCHW $AMTD - TD Bank to own 13% Schwab stake after Ameritrade deal	neutral
36	LVMH nears deal to buy Tiffany for $16.3 billion	neutral
37	YTD Performance: Dow Jones: Red 30Y: +9%	neutral

38	Should Community Bankers Trust (NASDAQ:ESXB) Be Disappointed With Their 99% Profit?	neutral
39	Should You Be Excited About Allianz SE's (ETR:ALV) 10% Return On Equity?	neutral
40	Should You Buy Harmony Gold Mining Co. (HMY)?	neutral
41	Liberty Global PLC maintained as buy with $32 price target at Benchmark	positive
42	Boeing : Deliveries 24 Jets in November #Boeing #Stock #MarketScreener	positive
43	Pershing Square Holdings announces new $100M share buyback program	positive
44	USD/JPY Forecast: USD to Consolidate Against JPY - DailyForex.com	positive
45	USD/JPY Weekly Price Forecast ‚Äì Dollar Continues To Test Resistance	positive
46	Global air passenger traffic has grown at 4% YoY in 2019 ytd while air freight volumes contracted 4% over the same period: BofA	positive
47	added $MRK Feb 90/Dec 90.50 Call Diagonals	positive
48	$XOM (+5.8% pre) Exxon cuts full-year capex forecast by 30%, maintains long-term outlook - SA	positive
49	Tesla Tops $900 in Parabolic Surge, Spurring Analyst Downgrade	positive
50	Tesla's stock ticks up after Deutsche Bank lifts price target, which implies an 18% decline	positive