The Voice of the Market: Multi-Source Supervised Sentiment Scoring for Real-Time Financial Forecasting

Sanggyu Sean Choi

MInf Project (Part 2) Report
Master of Informatics
School of Informatics
University of Edinburgh

2025

Abstract

Amid the information overload and the AI-driven transformation of the modern financial landscape, real-time sentiment processing is no longer optional—it is essential for contemporary investment strategies. This thesis proposes an end-to-end, real-time automated sentiment score prediction system that draws from heterogeneous sources to support stock return and volatility forecasting. The sentiment outputs encapsulate multiple layers of market perception, including the internal stance of the firm, the tone of its leadership, and evaluations by buy-side experts. Generated in real time, these sentiment signals support not only risk management and fundamental analysis, but also high-frequency trading strategies triggered immediately upon a document's release. The system consists of three core models—the Financial Text Retrieval Model (FTRM), the Parallel Data Construction Model (PDCM), and the Sentiment Score Prediction Model (SSPM)—which are seamlessly integrated and orchestrated via the Airflow framework within a Dockerised cloud computing environment.

Keywords Sentiment Analysis, Fundamental Analysis, Data Orchestration, Machine Learning, Return, Volatility, ETL, Real-Time Computing, Big data

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics Committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Sanggyu Sean Choi*)

Acknowledgements

I sincerely appreciate the invaluable support and guidance provided by my supervisor, Professor Tiejun Ma, and my co-supervisor, Dr. Luis Felipe Costa Sperb, throughout this project. I also thank Hao Zhou and Waylon Li, his PhD student, for their support.

Many thanks and love to my family(Hyungmook Choi, Jeongseun Lee, and Jiwon Choi) and my grandparents(Jongmook Lee, Chunja Kwon, and Boksoon Lee) for your love.

Foremost, Thank you, God, for your immeasurable love.

"The fear of the Lord is the beginning of wisdom,
and the knowledge of the Holy One is understanding."

– Proverbs 9:10

Table of Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Project Objective . . . 2
1.2.1 Minf1 Project Summary . . . 2
1.2.2 Minf2 Project Objective . . . 2
1.3 Contribution . . . 4
1.4 Outline of the Project . . . 6

2 Related Works **7**
2.1 Data Processing Framework . . . 7
2.1.1 Apache Kafka . . . 7
2.1.2 Apache Flink . . . 7
2.1.3 Apache Spark . . . 8
2.2 Financial Data Retrieval . . . 8
2.2.1 Regulatory Filing . . . 8
2.2.2 Financial News Feeds . . . 8
2.2.3 Analysis Reports . . . 9
2.2.4 Social Media . . . 9
2.3 Financial Textual Sentiment Analysis (Sentiment Scoring) . . . 9
2.3.1 Lexicon-Based Approach . . . 10
2.3.2 Machine Learning-Based Approach in Finance . . . 10
2.4 Sentiment-Based Market Prediction Model . . . 12
2.4.1 Numerical Sentiment Scores as a Feature in Prediction Models 12
2.4.2 Trading on Sentiment Signals . . . 12
2.4.3 Deep Learning Models Integrating Text . . . 12
2.4.4 Sentiment-Driven Asset Allocation and Risk Management . . 12

3 Data **13**
3.1 SEC Filings . . . 13
3.2 Buy-Side Experts Analysis Reports . . . 14
3.3 Earning Call Transcripts . . . 14

4 Real-Time System Architecture **16**

5 Financial Text Retrieval Model **18**
5.1 Dynamic S&P 500 List . . . 18

5.2 SEC Filings . . . 19
5.3 Buy-Side Experts Analysis Reports . . . 20
5.4 Earning Call Transcripts . . . 21

6 Parallel Data Construction Model **22**
6.1 Conversion to Parquet . . . 24
6.2 Parallel Document-Term-Metric Construction . . . 25
6.3 Multi-Stage Concatenation . . . 25

7 Sentiment Score Prediction Model **26**
7.1 Methodology . . . 26
7.1.1 Notations . . . 26
7.1.2 Model Assumptions . . . 27
7.1.3 Model . . . 28
7.1.4 Scoring New Filing . . . 31
7.2 Automatic Labeling . . . 31
7.2.1 Stock Return y_i or R_i . . . 31
7.2.2 Stock Volatility y_i or V_i . . . 32
7.3 Kalman Filter . . . 33

8 Experiment Results **35**
8.1 System Performance . . . 35
8.2 Sentiment Scores Analysis . . . 36
8.2.1 Descriptive Statistics . . . 36
8.2.2 Sentiment Scores Correlation Analysis . . . 36
8.3 System Adaptability . . . 37

9 Conclusions **40**
9.1 Conclusions . . . 40
9.2 Limitations and Future Works . . . 40

Bibliography **41**

A Textual Sentiment Analysis Level **52**
A.0.1 Document Level . . . 52
A.0.2 Sentence Level . . . 52
A.0.3 Phrase Level . . . 52
A.0.4 Aspect Level . . . 53

B 10-K Report Form **54**

C 10-Q Report Form **56**

D Financial Text Retrieval Model Example **57**
D.1 10-K Filing (e.g Nvidia 2024-02-21) . . . 57
D.2 10-Q Filing (e.g Nvidia 2024-08-28) . . . 58
D.3 Earning Call Transcripts (e.g Nvidia 2024-08-28) . . . 58
D.4 Expert's Analyst Report (e.g Nvidia 2025-02-03) . . . 59

E Experimental Setup **60**
E.1 Preprocessing 60
E.2 Hyper-parameter Setting 60
E.3 Baseline 62

F Pearson Correlation **63**
F.1 Definition 63

G Descriptive Analysis **64**

H Sentiment Score Prediction Model Continuous Results **65**
H.1 the U.S market 65
H.2 Nvidia 66
H.3 Tesla 67
H.4 Apple 69

I Earning Call Frequency Heatmap **72**
I.1 Nvidia 72
I.2 Tesla 73
I.3 Apple 73

Chapter 1

Introduction

1.1 Motivation

In the era of innovation and turbulence, a shifting industrial paradigm driven by Artificial Intelligence during the Fourth Industrial Revolution, it has been five years since I invested in the market as a retail investor and hard-working college student with a 7.5% compound annual growth rate. The 7.5% revenue with less effort due to college studying is not remarkable, but solid and sound. However, the investment journey wasn't that easy. Nvidia and Bitcoin are well known for not only being notoriously volatile assets but also susceptible to their sentiment or market sentiment [45, 84]. For instance, Nvidia's valuation fell almost $200 billion dollars after their earnings report failed to exceed Wall Street's lofty expectations[43]. Due to their intrinsic volatile features nowadays, our portfolio returns are directly and negatively affected without any system to hedge these volatile assets. If we could quickly capture market sentiments, we would hedge the risks of such assets. Investing in a time of upheaval highlights the importance of real-time market analysis more than ever.

In today's modern financial market, information digitised from news, social media, and regulatory institutions travels instantaneously in a various format, from text, images, and vocals to even brain signals; thus, real-time market analysis has become a critical tool for capitalise on these types of data [54, 79, 111, 98, 117]. Real-time market sentiments have become a strong market indicator that contains predictive signals. High-quality research and real-world practice have shown the predictive power of real-time sentiment on returns and volatility. The consensus from research and real-world practice says that a combination of real-time sentiment analysis to investment leads to better-informed decisions and often more profitable investments/trades [100, 119, 78, 140, 18]. Even [91] emphasised the importance of fast sentiment-driven decision-making, highlighting the necessity of real-time systems to analyse a market or firm sentiment on time.

Additionally, there is a myriad of sentiments (from news, social media, forums, earnings calls, analyst reports, and regulatory filing), and different sentiments reflect various participant groups: retail sentiments, institutional sentiment, and media sentiment. Aggregating these sentiments will give a holistic view of the market/firm mood, preventing

investors from having a biased slice. Recent research showed that the comprehensive sentiment data produced more powerful predictive signals. [19] discovered that combining sentiment analysis from various sources, including financial news, SEC filings, social media, and earring calls, increased the prediction of intraday returns for the S&P 500 index. [47] also showed that multi-source sentiment data from news articles, X(Twitter), and forums improved stock price movement prediction.

Therefore, in the modern era of turbulence, having an unbiased and real-time sentiment metric on the market and firm has become more critical than ever. In this paper, we will develop an **automated** system for generating **real-time** sentiment metrics of the US market **at scales** for successful investment.

1.2 Project Objective

1.2.1 Minf1 Project Summary

The main contribution of our Minf1 project was to develop an automated system to generate sentiment metrics, comprising two main modules: the 10-K Filing Extraction Model and the Sentiment Score Prediction Model (SSPM). The 10-K Filing Extraction Model systematically fetches 10-K filings from the Electronic Data Gathering, Analysis, and Retrieval(EDGAR), including the Item 1A risk factor disclosure section. The extracted filings and their risk factor sections are fed into the SSPM to predict the market sentiment. The SSPM predicts a sentiment score, given the data, which provides a sentiment metric for the market. In the previous study, the SSPM generated three types of predicted sentiment metrics across three levels of stakeholders: the technology sector, the portfolio sector, and individual company (i.e., Nvidia). When predicting a sentiment score, we utilised return and volatility as a label for training; thus, we generated both return-predictive sentiment signals and volatility-predictive sentiment signals. In addition, we critically assessed two sentiment metrics through qualitative analysis (ie, the most impactful words for return and volatility) and quantitative analysis (ie, correlation analysis), which helped investors to understand the market and its firm deeply. In our empirical experiment, we found that our 10-K centred model outperforms our risk-factor-centred model at the technology sector level in distilling return/volatility-predictive signals. At the portfolio level, both models excelled within the context. Finally, at the company level, the risk model did well in showing a trend, whereas both models extracted meaningful words. Moreover, a preliminary automated system was implemented on Airflow.

1.2.2 Minf2 Project Objective

As illustrated above, we developed an automated system for generating sentiment metrics but within a limited functionalities with a small dataset (that is, 10-K filings of 100 NASDAQ firms ranging from 2006 to 2023). Thus, in this Minf2 project, we aim to achieve the following main goals:

- Developing an **automated** system for generating **real-time** sentiment metrics of the US market **at scales**. Specifically, we will generate three types of the S&P

500 sentiment metrics from three datasets: SEC filings (i.e., 10-K and 10-Q), expert analysis reports (i.e. buy-side-focused reports) and earning call transcripts. These sentiment indicators that reflect various perspectives will strongly support not only analyzing the market fundamentally but also predicting the stock market.

The following are the sub-steps for the main goal:

- Extracting data: 10K and 10Q SEC filings, financial experts' analysis reports, and earning call transcripts. 500 firms from the S&P 500 will be collected as a form of these three types of data. The window ranges from 2006 to 2025.

Firstly, we need to extract data at scales. Extraction processing should be seamless, systemic, and efficient for predicting real-time sentiments at scales. Our pipeline will fetch SEC filings from the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system. The number of SEC filings we expect to collect is around 38,000. Experts' analysis reports and earning call transcripts will be collected from Seeking Alpha, the biggest US-based crowdsourced investment platform. The estimated number of experts' reports we will collect is 150,000, and transcripts are 38,000. By using three different datasets, we aim to predict sentiments reflected by various perspectives. That is, SEC filings represent a company's official perspective on its business and its surrounding factors. Analysis reports indicate a buy-side financial expert's opinion on a firm or a market. Earning call transcripts reflect a firm's leadership perspective.

- Data preprocessing and constructing document-term metrics.

After the collection of data at scales, our data should be preprocessed, and subsequently, our system will construct document-term metrics for feeding into our sentiment prediction model. Since the data we will handle is massive (i.e., it's around 1 billion tokens), our preprocessing and construction process should be the most efficient for our research goal. Thus, we will implement a distributed system working in parallel for such processes.

- Generating sentiment scores to reflect multiple perspectives on the US market or a US firm.

In this research, we aim to generate sentiment metrics to measure the market's sentiment to support stock prediction. The sentiment prediction model referenced [67]. Recently, [67] proposed a model to generate a strong sentiment indicator to predict price reactions to news articles. [67] showed that their sentiments predicted by their supervised sentiment model trained on news articles indicate a powerful return-predictive signal. By leveraging their methodology to predict sentiment scores on the market, our research will innovatively improve this area. [67] focused only on return-predictive signal generation within limited news articles. Although they still have discovered meaningful sentiment scores from financial news articles, no prior studies have employed other datasets for this approach. The previous Minf1 project has focused on only 10-K filings from NASDAQ to gauge technology sector sentiments, but this time, we will immensely differentiate our dataset to show sentiment metrics from several perspectives, focusing on the US market. Furthermore, [67] only generated a return-predictive signal. However, due to the nature of [67]'s methodology, we can generate several predictive signals. In this research, we will generate a volatility-predictive signal as well as a return-predictive

signal in terms of sentiment.

- Building a real-time automated sentiment pipeline on the Airflow framework.

In the turbulent macroeconomic environment of the Trump administration era, the importance of rapid market analysis is increasing. Moreover, financial markets are inherently uncharted, like the ocean, with information flooding in every moment. But there is a lot of noise in the information. Under these circumstances, we need to equip a tool which precisely and insightfully analyses a market or a firm by filtering out noise. From [67] and our previous research, we have shown the usefulness of the model. Now that we have the model, it's time to build a real-time system to generate sentiment indicators to stay ahead of the market. We will build this system on the Airflow framework with the distributed system mentioned above.

- Evaluating sentiment metrics in the context of supporting market movement prediction.

After achieving the above goals, we will evaluate our generated metrics quantitatively via Pearson correlation.

1.3 Contribution

We have completed the primary goal, including all sub-goals that we set in the previous project objective section.

Our contributions are shown as follows:

- Developed a real-time automated system for generating sentiment metrics of the US market at scales, comprising three main models: Financial Text Retrieval Model (FTRM), Parallel Data Construction Model (PDCM), and Sentiment Score Prediction Model (SSPM).
  - Implemented the Financial Text Retrieval Model, which automatically retrieves SEC filings (10-K and 10-Q), earning call transcripts, and financial expert's analysis reports. Each dataset was collected via API, but since each dataset has a different API, we handled different protocols, respectively. For SEC filings, unlike the previous project using an RSS feed-based approach, the model algorithm was updated to leverage the new SEC's XBRL data API. A total of 25,925 SEC filings for 10-K and 10-Q were retrieved. For earning call transcripts and financial expert analysis reports, we built algorithms to work in parallel, retrieving 260,000 for reports and 38,000 for transcripts. After retrieval, all data are fed into the PDCM.
  - Implemented the Parallel Data Construction Model, which automatically preprocess and construct a document-term-matrix for the subsequent SSPM. Since we handled an enormous amount of data(approx 1.5 billion tokens in total), this model is built highly efficiently for real-time purposes, highlighting its ability to process a single filing in 17 seconds, an article in just 0.036 seconds, and a transcript in 0.05 seconds. After all data are ready, they are

fed into the SSPM.

  - Implemented the Sentiment Score Prediction Model to predict sentiment scores, reflecting three perspectives: a firm's official tone, its leadership tone, and financial experts' tone. Referenced in [67]. Our model demonstrated results across two levels of stakeholders to maximise the model's applicability.
    * Formulated six sentiment metrics across these levels:
      · **Market Level (i.e., the US market)**
      · Generated market sentiment metrics labelled with return and volatility, reflecting three perspectives.
      · **Company Level (e.g., Nvidia, Tesla, and Apple)**
      · Established company-specific sentiment metrics labelled with return and volatility, reflecting three perspectives

- Collected financial data through FTRM (Type: the S&P 500, Windows: 2006 to 2025).
  - the S&P 500 Historical constituents from Wharton Research Data Services (WRDS).
  - the S&P 500 Daily Index and Price Data from MacroTrends.
  - 25,925 10-K and 10-Q SEC filings from EDGAR.
  - 260,000 financial expert analysis reports from Seeking Alpha.
  - 38,000 earning call transcripts from Ninja API
  - the S&P 500 historical price data from 2012 to 2024 from Seeking Alpha.
  - Earning call calendar from Seeking Alpha.
- Implemented the S&P 500 dynamic filtering algorithm to remain active firms per year when constructing a document-term matrix. This technique is essential for correctness - many academic systems overlook merge/spinoff, leading to incorrect analysis.
- Optimised training, which is relevant to SSPM.
  - When calculating co-occurrence frequencies of terms over all documents, the previous model calculated the co-occurrence frequencies term by term, which is significantly time-consuming due to the very high dimensionality. Thus, we vectorised co-occurrence calculation through matrix multiplication, computing the sum of term occurrences in one operation. This alteration massively increased the training efficiency.
  - Also, we used a Spare data structure in pandas where it stores only the non-zero elements, improving the memory usage. On top of that, we used

a SciPy sparse matrix for matrix multiplication, reducing computation time further.

  - With this improvement compared to the previous training, we made our training 5 times faster.

- Generated return-predictive sentiment signals and volatility-predictive sentiment signals in a real-time scenario, reflecting three viewpoints: a firm's official tone, its management tone, and buy-side focused financial experts' tone. No prior study has developed the system with multiple sources.
- Assessed the performance of the SSPM for three datasets with descriptive statistics and correlation analysis
- Connected to the Airflow server for real-time automatic update of new data.
- Assessed the real-time performance for our whole system regarding running time.
- Suggested a practical strategy for high-frequency stock trading under our developed real-time system with earnings call transcript publication estimation via its calendar data.

1.4 Outline of the Project

The thesis structure is outlined below:

- **Chapter 1 - Introduction** provided motivation for this study, including study objectives and contributions
- **Chapter 2 - Related Works** explained related works about data processing framework, financial data retrieval, financial textual sentiment analysis, and sentiment-based market prediction model.
- **Chapter 3 - Data** explained the data we utilised in the system.
- **Chapter 4 - Real-Time System Architecture** showed overview of our real-time system.
- **Chapter 5 - Financial Text Retrieval Model** described the retrieval algorithms for each type of text - SEC filings, earnings calls transcripts, and buy-side reports
- **Chapter 6 - Parallel Data Construction Model** detailed a distributed preprocessing system for real-time demands
- **Chapter 7 - Sentiment Score Prediction Model** showed a comprehensive explanation of the sentiment prediction model.
- **Chapter 8 - Experiment Results** critically validates the values of sentiment metrics drawn from heterogeneous sources through descriptive statistics and correlation analysis, highlighting the system's adaptability.
- **Chapter 9 - Conclusions** summarised the system's advantages and disadvantages with future works.

Chapter 2

Related Works

2.1 Data Processing Framework

In the field of financial technology, efficient data processing is crucial for real-time financial systems, mainly when dealing with large-scale text data. Indeed, in modern finance applications, continuous real-time data integration and processing have become mandatory [123]. There are three major data processing frameworks that are actively used in academia and the finance industry: Apache Kafka, Apache Flink, and Apache Spark.

2.1.1 Apache Kafka

Apache Kafka is a distributed event streaming platform which is the backbone of real-time architectures. As a central data hub, Kafka decouples data producers and data consumers, allowing asynchronous communications between services. In the finance industry, Kafka is used for real-time transaction processing as well as streaming live market data feeds (e.g., stock ticks and FX rates) to traders and algorithms from exchange sources. Also, in risk management, risk factor data(e.g., market moves) is streamed via Kafka lively to risk calculator[123].

2.1.2 Apache Flink

Apache Flink is a distributed stream processing framework [59]. It enables high-performance, low-latency computation over continuous, unbounded data streams, making it ideal for real-time data processing and analysis. In the financial sector, Flink is used for time-sensitive analytics and decision-making, where both speed and precision are important. Also, risk assessment operations and the trading process can be an example use case of Flink. When streaming trade data to a Flink platform, it can continuously aggregate trades over time windows, calculate indicators, and trigger alerts - all in real-time. Flink can handle millions of events per second with millisecond-level latencies. Flink's exactly-once processing guarantees that a model will be prevented from double-counting or losing events during failures. These features make it a popular framework for mission-critical financial pipelines [59].

2.1.3 Apache Spark

Apache Spark is a distributed computing system for large-scale data processing. It is widely used for SQL, streaming, machine learning, and graph processing while being able to run on various platforms. Spark performs computations in memory, allowing significantly faster large-scale data preprocessing. It can process a vast amount of data per unit of time and integrates seamlessly with Spark's other components. Spark is widely used in finance: streaming analytics and anomaly detection, monitoring real-time training activity, detecting credit card fraud, or real-time risk management[59].

In this research, we mainly used Apache Spark for large-scale data processing. Since the size of our dataset was massive, Spark was necessary for such-scale data processing to build a real-time sentiment-generating system. However, Kafka and Flink wouldn't add much value as the data is not continuously streaming. The retrieval schedule is driven by report releases, not real-time events. Thus, we instead used Airflow to fetch the textual data on schedule, followed by being fed into the system, returning sentiment metrics in real-time.

2.2 Financial Data Retrieval

There are myriad types of textual financial data. Financial analysts and researchers retrieve data from a variety of sources with different techniques. Key sources contain regulatory filings, news feeds, analyst reports, and social media.

2.2.1 Regulatory Filing

SEC offers an EDGAR database to provide corporate disclosures (10-K, 10-Q, etc). EDGAR provides REST APIs for real-time filings of JSON format with sub-second latency. Global filing aggregators such as Refinitiv and FaceSet deploy unified APIs to fetch filings across dozens of markets (U.S., Canada, and others) through key-based authentication and bulk endpoints [141]. Third-party data providers such as QuoteMedia and LSEG Filings API also offer systemic retrieval services [3]. Web scrapping can fetch regulatory filings, too[97, 70]. [97, 70, 141, 3] used regulatory filing for their sentiment analysis via the APIs mentioned above.

2.2.2 Financial News Feeds

News wires like Reuters, Bloomberg, and Dow Jones provide real-time headlines and articles for market-moving events. Researchers can use Bloomberg's Terminal API (BLPAPI) or Refinitiv's News APL to fetch news feeds programmatically[76]. For instance, Refinitiv's Machine Readable News service delivers low-latency, structured news articles from Reuters, allowing automated parsing of headlines and metadata[12]. Dow Jones's Factiva Content API provides news articles, images, and sentiment metadata[38]. In practice, financial institutions and researchers subscribe to these data providers to ingest news in real-time.

2.2.3 Analysis Reports

Academic and corporate users often use analyst reports for their sentiment analysis. Many APIs and licensed databases offer their content to users. For instance, platforms like Refinitiv, AlphaSense, and S&P Capital IQ offer both buy-side and sell-side research reports to entitled clients via their API[9, 77]. Analysts estimate that data can also be collected via financial data APIs such as Financial Modeling Prep or Yahoo Finance API [125].

2.2.4 Social Media

Sentiment from social media has become an alternative data source. Social Messenger X and StockTwits are the major social media data providers, offering APIs to query posts by keyword or ticker. X's API v2 allows streaming or searching tweets via Python libraries like Tweepy [136]. Compared with X official API, [29] used the Twint library for sentiment analysis where it has no restriction in the official X API and offers useful functions to filter out data. Although this real-time social data is noisy and unstructured, retail investor sentiment trends can be revealed in tweets. Research finds that there is a positive correlation between market movements and tweet sentiment, supporting the next-day stock prediction[29]. Moreover, StockTwits is another central platform for investor messages, offering an open API that returns messages in JSON. [126] used this platform to collect over 2 million messages to extract market sentiments.

2.3 Financial Textual Sentiment Analysis (Sentiment Scoring)

In finance, textual sentiment analysis has received much attention due to its broad range of applications. Massive volumes of textual financial data have been employed to analyse, explain, and derive insights via sentiment analysis. One popular application is to support stock market movement prediction along with portfolio risk management [105, 108, 85, 50, 127, 110, 101]. In this context, two kinds of sentiments have been studied: investor sentiment and textual sentiment. Investor sentiment indicates an investor's subjective opinion on companies or markets, whereas textual sentiment, or text-based sentiment, refers to the sentiment of the text itself. In some studies [67, 63], for example, the term positive or negative 'tone' in a firm disclosure indicates sentiment. Note that investor sentiment and textual sentiment have fundamental differences. Investor sentiment reflects investors' subjective judgements and behavioural tendencies. In contrast, while textual sentiment may retain aspects of investor sentiment, it also provides a broader, more objective view of the states within firms, institutions, and markets [68]. Various textual data have been used in textual sentiment analysis, such as news articles, social media data, or corporate disclosures. News articles and social media data are employed to examine the short-term impact of sentiment on market variables such as returns, volatility, and trading volume [108, 127]. Corporate disclosure typically aims to identify the link between sentiment and firm performance [52, 25]. This study integrates both investor sentiment and textual sentiment to help stock market

prediction with multiple sentiment metrics. Offering both types allows investors and researchers to evaluate investment decisions from complementary perspectives.

2.3.1 Lexicon-Based Approach

The lexicon-based approach is one of the two main approaches for sentiment analysis. It relies on a predefined list of words, each assigned a sentiment score with a polarity value like +1, -1, or 0, representing positive, negative, or neutral sentiment or intensity of sentiment. Overall document sentiment is then aggregated by an algorithm. This approach is efficient at the sentence, phrase, or aspect level [65, 56]. However, a significant limitation is domain sensitivity, as word sentiment varies by context. For instance, while 'debt' often carries negative connotations, 'debt financing' indicates natural or even positivity in a financial context.

2.3.1.1 Corpus-Based Approach

The corpus-based approach begins with a list of seed words with pre-known sentiment orientation and utilises syntactic or co-occurrence patterns to identify additional words with their sentiment in a large corpus. This approach relies on linguistic rules or conventions involving connectives (e.g., AND, OR, BUT). For instance, adjectives linked by a conjunction (such as "satisfiable AND controllable") typically share the same sentiment orientation. The main advantage of the corpus-based method is simplicity, but a large dataset is required to detect the polarity of words [53].

2.3.1.2 Dictionary-Based Approach

Moreover, many prior studies have relied on predefined sentiment dictionaries to extract textual sentiment. These predefined dictionaries are developed using a dictionary-based approach. This method uses an algorithm where a computer programme maps text and categorises words, phrases, or sentences into predefined categories [74]. Several major predefined sentiment dictionaries are commonly utilised in financial text analysis. The first is the General Inquirer (GI) or Harvard IV-4 dictionary (HIV_4). The second is the DICTION dictionary. Both have been widely applied in financial analysis [36, 37, 42, 55, 32]. However, as general English linguistic dictionaries, the GI/Harvard and DICTION often produce inaccuracies in financial contexts. To mitigate this issue, Loughran and McDonald developed the LM dictionary, tailored specifically for the finance domain. Designed for sentiment analysis in 10-K filings, the LM dictionary serves as our benchmark for evaluating estimated sentiments. Numerous researchers have used the LM dictionary in their work [34, 58, 40, 26, 81].

2.3.2 Machine Learning-Based Approach in Finance

With advancements in computing power and AI, sentiment analysis in finance has adopted AI-based approaches. Classical machine learning techniques - such as Support Vector Machine (SVM), Na¨ıve Bayes, Decision Trees, or artificial neural networks (ANN) - have been used to address high dimensionality challenges [57, 74, 75, 109, 124]. For example, [124] analysed sentiments from social media textual data(e.g. StockTwits)

using various machine-learning binary classifiers, discovering that the SVM classifier outperformed Decision Trees and Na¨ıve Bayes. However, traditional ML approaches struggle to capture complex features and contextual nuances in sentences, making deep-learning techniques more suitable for these tasks. Deep learning enables a better understanding of sequential information, intricate feature extraction, and positional identification within text [112].

Deep learning, a subset of machine learning, utilises multi-layered neural networks to extract complex features [60]. In textual sentiment analysis, deep learning is used to generate learning patterns and learning sentence contextual information [137]. Many prior studies investigated the benefits of deep learning models, such as recurrent neural networks (RNN) [116, 115], convolutional neural networks (CNN) [71, 138, 64], and attention mechanisms [112, 134] for financial sentiment analysis. Transformer architectures such as BERT or RoBERTA have recently performed well in financial sentiment analysis [87]. However, transformers come with significant drawbacks: long inference times, high computational costs, and extensive data dependency [102]. Additionally, transformers are often seen as a "black box" because their complex internal mechanisms are difficult to interpret [33]. This black box feature can hinder investors from fully understanding the sentiments generated.

In contrast, [67] proposed an innovative and effective sentiment model with strong performance. This model, designed for lexicon learning, aims to identify the relationship between sentiments in firm-specific news and stock returns. The model offers five key advantages: (1) it is a transparent and straightforward supervised lexicon learning model, utilising only basic econometric techniques, such as correlation analysis and maximum likelihood estimation, making it a fully "white-box" approach; (2) it is highly resource-efficient and capable of processing millions of documents on a laptop within minutes; (3) the model is highly scalable, supporting a wide range of financial text data without relying on pre-existing sentiment dictionaries, unlike traditional lexicon-based approaches; (4) it eliminates the need for manual labelling, reduces labour costs, and uses automated labelling based on minimal, sound assumptions; (5) it shows strong predictive performance in stock return prediction.

In this paper, we gained access to the model's other four benefits by leveraging its scalability. In other words, unlike the model using news articles to generate return-predictive sentiment signals, we used three informative types of financial text(i.e., SEC filings, earnings call transcripts, and financial experts' reports) to produce volatility-predictive sentiment signals as well as return one instead. No prior study applied the model's methodology to such a wide range of financial text data. Also, even in this methodology, they did not use volatility to train sentiment signals. Moreover, we built an automatic system to return sentiment scoring in real time for various textual datasets. No prior research has developed this real-time system, highly leveraging the powerful model's applicability to support stock prediction and other financial applications.

2.4 Sentiment-Based Market Prediction Model

2.4.1 Numerical Sentiment Scores as a Feature in Prediction Models

A prediction model can leverage sentiment scores as input features. [139] found that sentiment features extracted from social media enhance the accuracy of stock forecasts. In forecasting consumer spending research, [96] used sentiment scores to predict future consumer spending from social media data.

2.4.2 Trading on Sentiment Signals

Some strategies directly trade off sentiment alterations. In other words, a sentiment-aware trading strategy might go long on stocks with positive sentiment news and short on those with negative sentiment, which is called a long-short sentiment portfolio. [15] found that a GPT-based sentiment model attained a Sharpe ratio of 3.05, which is substantially higher than a comparable strategy using the Lough McDonald dictionary-based sentiment lexicon. Also, [67], which used our system's sentiment prediction model, achieved an annualised Sharpe ratio of 4.21 with news sentiments, which is higher than [15]. Additionally, market practitioners echo these findings, discovering that news sentiment provides alpha across many markets. For instance, a RavenPack case study found that stocks recommended by daily news sentiment scores yielded annualised returns above 20% in Asia-Pacific portfolios [8].

2.4.3 Deep Learning Models Integrating Text

Certain advanced models embed textual analysis directly into the forecasting process rather than using a two-step method (get sentiment score, then feed to model). For instance, hybrid neural networks, such as CNN-LSTM, have been proposed with an attention mechanism focusing on crucial textual information to forecast stock trends [132]. Such architecture allows the model to "decide" which parts of the text are essential to predict sentiments(via attention weights) rather than depending on a pre-generated sentiment score. This approach showed relatively higher accuracy on stock prediction, but there are challenges with the need for extensive training data and the risk of overfitting.

2.4.4 Sentiment-Driven Asset Allocation and Risk Management

Sentiment score can serve as an early warning risk indicator. Portfolio managers monitor sentiment indices to assist asset allocation decisions [88]. For example, they can reduce exposure to a market if news sentiment turns abruptly negative. [10] found that there is a positive correlation between public sentiments and financial market trends, impacting optimal asset allocation. Additionally, [51] represented the value of news sentiment signals as a risk warning indicator in detecting "slow-burning" risks globally with the COVID-19 pandemic and the global financial crisis examples. With the aid of sentiment metrics, risk can be managed with holistic views and instant but deep insights.

Chapter 3

Data

3.1 SEC Filings

SEC filings are a rich deposit of reliable information about a firm. These are official documents which should be submitted to the Securities and Exchange Commission (SEC). In this study, we focused on SEC filings, especially for 10K and 10Q from S&P 500 lists given a window. To facilitate clearer communication, we will refer to 10-K and 10-Q as SEC filings from now on, except for specific mentions. SEC filings offer a comprehensive overview of the business of a firm, financial performance, and possible risks over the fiscal year. 10-K, for instance, is released annually with the richest information about a firm for the fiscal year. They contain audited financial statements, a firm's business model, product (or service), strategies, performance, and potential risks. You can find a detailed description of the 10-K structure in Appendix A. On the other hand, the 10-Q is a quarterly filing and offers less comprehensive information than the 10-K, providing a snapshot of the firm's financial performance over a three-month period. They cover financial statements, management's discussion and analysis of the firm's operations, and updates on risks or legal issues. The 10-Q structure can be found in Appendix C.

10-K and 10-Q are commonly written by a firm, and as such, they represent a firm's perspective about its business, finance, market, and events related to the firm. Over the years, several studies have shown that SEC filings provide predictive signals for stock prices. Also, a shred of evidence the [17] supported that the market responded to SEC filings in a statistically significant way. [69, 135] found that the level of complexity of SEC filings had a positive correlation with stock price volatility. Moreover, [21] showed via state-of-the-art AI methodologies that there was a positive correlation between sentiments of SEC filings and stock price volatility. As many studies revealed, the analysis of SEC filings will provide a predictive signal from a firm's point of view.

3.2 Buy-Side Experts Analysis Reports

A financial analysis report is a document written by a financial professional who thoroughly studies a firm's business, performance, and financial health, using various financial techniques and metrics with their knowledge[114] . It offers insights into its strengths, weaknesses, potential risks, prospects, and the market it belongs to. This type of report is written for an informed decision-making process [2]. In this study, we have focused on a buy-side report written by professionals (e.g., hedge fund managers, portfolio managers, analysts) and individual investors managing their own money. We could collect many buy-side reports from over 500 firms in the S&P 500 list from the Seeking Alpha platform. Seeking Alpha(SA) is one of the biggest crowdsourced investment platforms in the United States. Contributors on that platform publish articles on stocks, markets, and investment strategies, which are generally reviewed by a review team and are subject to editorial alterations. The purpose of the review is to improve the quality of the article, not to hinder the authors' voices. Moreover, SA improves the quality of articles by requiring authors to disclose their identity and their holdings on the stocks they analyse. Since 2011, SA offered payment to contributors based on page view counts they receive, leading to improve the quality of the articles [4].

Several studies have found that a buy-side report can include predictive signals for the market [4, 92, 89, 27]. This is because they contain a detailed analysis of the market, company performance, and economic factors. [92] found that an analyst's analysis on the market has an impact on security price. [4] study specifically used Seeking Alpha articles for their research. They found that SA articles' sentiment tones predict future stock returns, helping retail investors make more informed investment decisions. [89] showed that buy-side traders' behaviours predict stock price movements via deep neural networks. [27] integrated social media indicators with buy-side-relevant momentum signals to predict small-cap stock price movements, outperforming benchmarks. However, a buy-side report doesn't always have an impact on supporting market prediction as analysts have, at best, a limited understanding of the market to predict accurate target price forecasts [72]. Despite that fact, it's worthwhile to analyse the sentiment tone of the buy-side reports to support market prediction.

3.3 Earning Call Transcripts

An earnings call transcript is a written record of a public firm's earnings conference call to discuss the company's financial performance, operational updates, and future prospects with analysts and investors. An earnings call is usually preceded by an earnings report, such as a 10-Q report and a 10-K report. They discuss important aspects of the reports, especially focusing on the management discussion and analysis (MD&A) section. Information from earning calls is used by analysts to do fundamental analysis on the firm. Moreover, unlike SEC filings, this call contains a direct communication, like a Q&A between participants(e.g., analysts and investors) and the company leadership, offering rich insights. However, Q&A could produce unfavourable results, so a firm tries to establish a regular cadence to avoid negative speculation [1].

Recently, many studies showed the predictive signals of earning call transcripts via text analysis [39, 66, 73]. [39, 73] found a significant positive correlation between the linguistic attributes of earning call conferences and the market's assessment of firms' future performance. [66] discovered that their Alexandria technology LLMs trained on earning call transcripts for the S&P 500 stocks outperformed other models, such as FinBERT and the Loughran McDonald dictionary, on trading performance and fund management. Likewise, the earning call transcripts are a rich deposit of information of a firm, reflecting the predictive signals, and as such, we researched the transcripts to see the sentiment tones of a firm for supporting stock predictions.

Chapter 4

Real-Time System Architecture

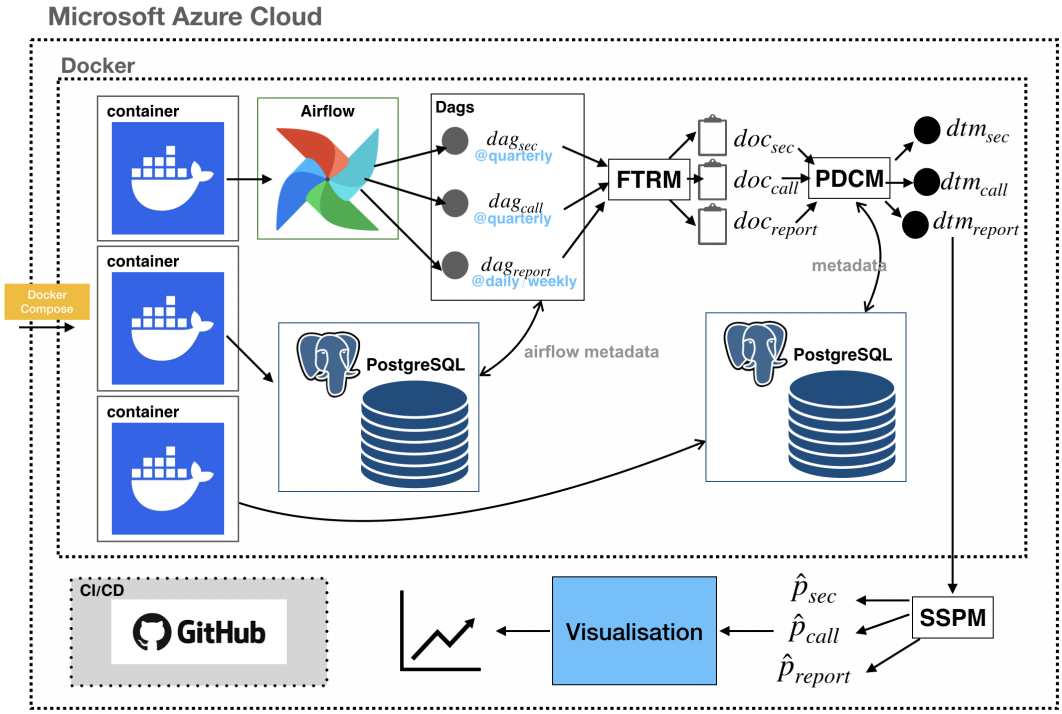

Figure 4.1: Real-Time System Architecture

In this research, we developed an end-to-end automated system to generate sentiment metrics in real-time. This system is comprised of three models: Financial Text Retrieval Model (FTRM), Parallel Data Construction Model (PDCM), and Sentiment Score Prediction Model (SSPM). Three models are seamlessly connected under the Airflow data orchestration framework, providing optimal sentiment analysis immediately upon the arrival of new documents arrivals. In other words, our end-to-end system efficiently and rapidly handles everything from collecting the latest data to processing and analysis in one seamless flow. Additionally, this system can handle a wide variety of text data types, improving the scalability of the system. In this research, three types of text data

are used: regulatory SEC filings, earnings call transcripts, and buy-side experts' reports.

Figure 4.1 illustrates the system architecture, which is deployed on Microsoft Azure Cloud. Leveraging cloud infrastructure ensures scalability and maintainability as data volume and complexity increase. In addition to the three text types used in this research, the system can easily scale to handle additional sources, such as annual reports or social media. Cloud dynamic provisioning can efficiently manage resource demands such as memory, GPU, and storage.

This system is containerised using Docker, providing scalability, modularity, and operational robustness. Thus, when there is a demand for this system to be upgraded, this Dockerised system will facilitate upgrades with more collaboration engaged. This system consists of three containers: one for Airflow and two for PostgreSQL databases. The Airflow container orchestrates the end-to-end pipeline, from collecting data via FTRM to generating sentiment metrics through SSPM. For each document type, we generate both market-level and firm-specific sentiment scores, resulting in six sentiment metrics across the three sources. Also, as each metric is trained for both return and volatility, resulting in a total of twelve sentiment metrics. Each metric is managed by a dedicated DAG; when a new disclosure is published, the corresponding DAG is triggered, initiating real-time sentiment prediction. PostgreSQL acts as the backend for Airflow, managing metadata, DAG schedules, and workflow execution logs.

DAG scheduling should be carefully tailored to the characteristics of each data type, as financial text publications are often irregular and lack standardisation. While SEC filings and earnings call transcripts are usually released quarterly after the market closes, U.S. firms do not follow a unified schedule [24]. Furthermore, the publication timing of buy-side reports is even more variable, affected by significant market events, shifts in a firm's fundamentals, or earnings announcements [121]. Some buy-side reports are also produced internally by investment firms for their investment decision-making, so the publication date may not be publicly available [130]. As a result, to boost the efficacy of the real-time system, a custom DAG scheduling should be required for each data type. In this research, we suggested a tailored DAG scheduling time to the data we employed for a high-frequency stock trading strategy in the section below.

Once scheduled, the FTRM extracts the latest documents through sources-specific APIs: EDGAR for SEC filings, Ninja API for earnings call transcripts, and Seeking Alpha for buy-side reports. FTRM is designed to support various financial text sources, ensuring extensibility. Subsequently, the retrieved data is fed into PDCM for data preprocessing and document-term-metric construction. PDCM tracks a data flow trace, storing its metadata in a PostgreSQL database to rapidly and efficiently update the new documentation. The processed data is then forwarded to the SSPM. It predicts a sentiment score. The sentiment metrics generated can be used variously for both fundamental and technical analyses, depending on the specific objective. We handled continuous integration and continuous delivery (CI/CD) with GitHub to improve development efficiency and development reliability.

Chapter 5

Financial Text Retrieval Model

5.1 Dynamic S&P 500 List

The S&P 500 index is one of a family of S&P U.S. equity indices designed to measure the country's market performance. The S&P 500 index is composed of 500 large-cap U.S. stocks. Still, over time, due to various market and corporate events, companies can enter or exit the index based on altering eligibility criteria. The S&P 500 index is rebalanced annually. Thus, the S&P 500 is a dynamic list, reflecting alterations such as additions and deletions of companies across years because of bankruptcies, acquisitions, mergers, or significant market cap alterations [49]. Due to that, a researcher should deal with this characteristic of dynamism, reflecting the corresponding years of their research scope. In this research, our windows are ranged from 2006 to 2024, and as such, we have collected a list of firms from 2006 to 2024, as the Item 1A risk factor section has been included since 2006. However, each year would have a different list of firms due to the dynamism of the S&P 500. Since there are no perfect the S&P 500 lists to reflect dynamic alterations over time, we dynamically handled these lists over time.

To do that, we first collected a union of a firm list of given window ranges from Wharton Research Data Services (WRDS). WRDS offers detailed historical information on S&P 500 constituents over multiple years. We filtered firms active during each year in the specified range. To explain a thorough filtering process, the dataset consists of various columns, especially 'permanent number', 'starting date', and 'ending date'. The permanent number is a unique identifier assigned to each security by the Center for Research in Security Prices (CRSP). It remains constant over time, even if the firm's ticker symbol or name alters. Also, this union dataset from the WRDS contains the starting and ending dates of the stock's availability. More specifically, the starting date refers to the date when the stock was first included in the dataset, whereas the ending date indicates the date when the stock was removed from the dataset. With the aid of the permanent number, the starting date, and the ending date, we generated a list of firms that were active each year. Furthermore, there is a case where a few tickers have the same permanent number. This is because a firm can merge and alter its ticker.

For instance, Elevance Health, Inc. is a primary American health insurance provider

listed in the current S&P 500. Before having the name Elevance Health, Inc.(its ticker is ELV), it had different tickers, such as WLP, ATH, and ANTM, with the same permanent number(i.e., 89179). WellPoint, Inc.(WLP) merged with Anthem, Inc.(ATH). After merging, they rebranded to Anthem, Inc.(ATHM) as a major U.S. health insurance provider. Then, it changed its corporate name to Elevance Health, Inc.(ELV) to reflect expanded healthcare services. To handle its alteration in the formation of a dynamic list reflecting active firms, we employed the 'name end date' given by CRSP. The name end date represents a date when the company name ceased to be effective. For each year, we only pick up a ticker active on that year by using this metadata. In summary, to determine which firms were active each year from the dynamic S&P 500, we first retrieved a union of companies' lists of given ranges from WRDS. Then, we identified the earliest and latest years in the dataset. And, for each year in this range, we check which companies were active by seeing if their start date was before or during that year and their ending date was after or during that year. We removed unnecessary or duplicate tickers for the case of the corporation's evolution. This process returns an annual list of active firms. With the exact yearly list of active firms, we processed all types of data of S&P 500; SEC filings, financial expert's reports, and earning call transcripts to precisely predict sentiment scores for given periods.

5.2 SEC Filings

We collected 10-K and 10-Q filings from the S&P 500 list ranging from 2006 to 2025 via our Financial Text Retrieval Model (FTRM). For the collection result, we retrieved 7,539 10-K filings and 18,386 10-Q filings given the windows. All SEC filings are publicly available from SEC's database system, known as the Electronic Data Gathering, Analysis, and Retrieval(EDGAR). To retrieve a big data of SEC filings efficiently and effectively, we designed retrieval algorithms with the modification of the previous work [107].

The SEC retrieval algorithm is designed to efficiently collect specific regulatory documents from EDGAR for a given list of firms, identified by their CIK numbers. The process starts by loading a metadata containing firms name, CIK numbers, and ticker symbols, which are then mapped into a dictionary for easy reference. Using this metadata to fetch filings of a specified document type(e.g., either 10-K or 10-Q) within a customisable data range, the algorithm sends a request to the database. To adhere to SEC's rate limits - allowing only 10 requests per second -, a throttling mechanism is incorporated, which controls request flows between the client and the server as needed. For each firm, metadata such as filing URLs, types, dates, and CIKs are retrieved by querying the EDGAR database via an RSS feed. The algorithm then filters outcomes to match the desired document type and date ranges. It then downloads the full textual content of these filings and saves them to HTML files. However, recently, the EDGAR significantly has changed their Application Programming Interfaces (APIs) to host RESTful data API delivering JSON-formatted to a client. Thus, only SEC filings until 2011 can be retrieved via RSS feed. But since we need to collect SEC filings ranging to the most recent 2025, the algorithm is modified in this research for that usage.

Instead of an RSS feed-based approach, the updated algorithm leverages the SEC's XBRL data API, offering a more modern and structured approach for collection. The new version targets the XBRL API endpoint to fetch filings, especially for focusing on the "AccountsPayableCurrent" concept for a given CIK. This alteration narrows the scope to structured JSON data with still filters for a specified document type and date range. The algorithm now then handles the JSON response to extract filing's metadata such as, form type, date, and accession number, constructing direct URLs to HTML filings. Overall, the update algorithm achieves a modernised retrieval system by considering XBRL data precision and direct HTML access. Thus, we could collect SEC filings from 2011 to the most recent date.

5.3 Buy-Side Experts Analysis Reports

In this research, our objective was to collect as much information as possible from financial experts to reflect their sentiments on our sentiment score. The analysis reports are a financial expert's analysis of a firm of S&P 500, and the range of the collection is from January 2006 to December 2024. Eventually, we retrieved 260,000 analysis reports systemically and efficiently.

To achieve the real-time automatic generation of sentiment index, a systemic and efficient approach is required. The expert analysis reports are collected through API calls. API returns JSON file format, semi-structure data, to provide various meta data which facilitates the development of extraction algorithms. The API requires a ticker parameter such that we can collect a firm-based analysis report by ticker. Since the data we aimed to collect is big, we should approach the process systemically. We should collect all reports of S&P 500 firms from 2006 to 2024. Each firm has a different number of reports each year, and an analysis report corresponds to an ID assigned by the data provider. Firstly, we retrieved all IDs of a report. The ID dataset is structured such that each firm is associated with a sequence of years, indicating the years for which ID is recorded. Each firm is listed in a row, followed by its corresponding years of data. After collecting all IDs, we collected all reports corresponding to IDs through API calls under the same dataset format. As mentioned before, each firm has a different number of reports each year. Also, some companies with a large market cap or are leaders in an industry have more reports than any other firms. These discrepancies can cause unnecessary API request calls. To avoid that, we statistically reduce the unnecessary calls by calculating the sample distribution of the number of reports for each firm. With the 20 per cent sample distribution of the S&P 500 ranging from 2006 to 2024, we could assume that 14% of the population group does not have reports, 75% have more than 1 report and less than 40 reports each year, and 10% have more than 40 reports and less than 80 reports. Through our assumption, we have collected 260K reports, and compared to the population distribution, we avoided unnecessary API calls, leading to saving in our budget. The population group shows 17% of the population lacks reports, 81% have 1-40 reports per year, and only 2% have 40-80 reports per year, which is closely aligned with our assumption.

Retrieving 260K reports was hugely time-consuming at first. As mentioned above,

we should collect the IDs first and then retrieve a report corresponding to an ID. We estimated the initial brute force approach would take 4160 hours to collect only 260K ids. It means it would take around 8000 hours to collect all reports. However, with the newly developed algorithm we built, we could immensely reduce the collection time to 120 times faster than the first brute force. The techniques we employed are asynchronous I/O, concurrency limiting, batch processing, rate limiting with backoff retries, caching via file checks, and reusing a single session. Asynchronous I/O requests ensure a machine (i.e., mainly CPU) is not sitting idle between network calls. Concurrency limits prevent overload on both the client and server. Batch processing structures your workflow in manageable chunks. Through empirical experiments, 50 concurrency limit and 50 batch chunks showed the best performance in this case. Rate limiting with backoff retries keeps you within the provider's limits and allows graceful recovery from transient issues. Caching avoids redundant API calls. Single-session reuse reduces connection overhead. We could handle large-scale API data collection more efficiently and reliably with these techniques.

5.4 Earning Call Transcripts

Furthermore, we collected transcripts of the S&P 500 from 2006 to 2024. We retrieved earnings call transcripts and transcripts of investor days, conferences, special events, annual meetings, and pre-IPO events. We used the same approach and algorithm for analysis reports, and eventually, we collected 43K transcript data.

Chapter 6

Parallel Data Construction Model

Our previous model was time-consuming in handling a large amount of data, as it did not consider a large scale of data. That is, it handled 10,000 large textual data points. But, in this research, we scaled up our data points 30 times larger than the previous to generate a sentiment metric. As mentioned above, we have collected a large amount of medium-to-large textual data through the suggested Financial Text Retrieval Model. The collected data will be fed into the Sentiment Score Prediction Model (SSPM), which requires a term-document matrix to predict a sentiment after pre-processing. However, pre-processing and building a document-term matrix with a large amount of medium-large textual data(in our case, around 300K medium-large documents equivalent to 1.3 billion tokens) will not only be computationally expensive but also time-consuming, especially for generating a real-time sentiment metric representing the US market with a different perspective.

Furthermore, we aim for the whole system to build a real-time sentiment score on the market. Finance data is released every day, every moment, flooding the market, making it crucial to analyse them faster than the market. Due to our increased scalability as well as a new functionality, an efficient data construction model is necessary. Thus, we proposed the Parallel Data Construction Model (PDCM) to facilitate data preparation for generating a real-time sentiment prediction score on the market. The model consists of three processes: Conversion to Parquet, Parallel Document-Term-Matrix Construction, and Multi-Stage Concatenation. The PDCM overview can be found in Figure 6.1.

In this research, three types of data are used: SEC filings, financial expert's analysis reports, and earning call transcripts. These three types of data include a firm of the S&P 500 ranging from 2006 to 2025. SEC filings represent a firm's structured, regulatory perspective on their financial performance and risks, whereas Earning Call transcripts indicate the financial results and business outlook like SEC filings, but contain real-time interpretation of the results with a management's sentiment. Thus, extracting sentiments from both SEC filings and transcripts will offer complementary but also various perspectives on a firm or a market. In addition, a financial expert analysis is used for sentiment generation. Their reports represent the perspective of a financial expert based on their extensive knowledge and experience in a market or a firm.

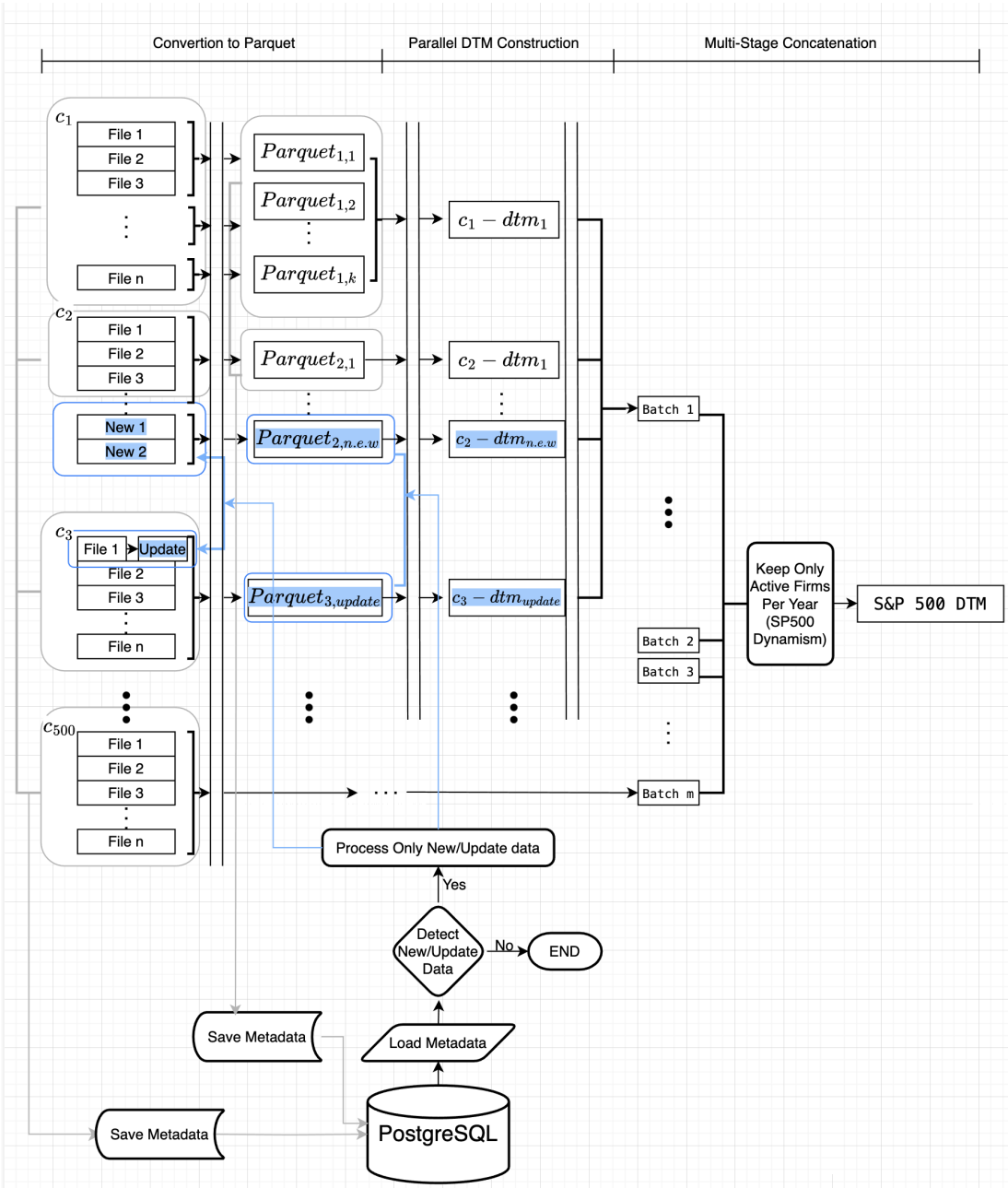

Figure 6.1: Parallel Data Construction Model

Three types of data are being retrieved in the form of JSON or HTML files at the first stage. Unlike JSON files, which are ready to be pre-processed as semi-formatted data, HTML files should remove unnecessary layout elements such as headers, footers, and titles, although HTML has regularity in its structure. Thus, removing layout elements of HTML is a prerequisite. By removing them with text manipulation, the HTML files only contain the necessary body of the text. Then, we convert the HTML files to TXT files for easier data handling after removing them. The HTML parsing and text manipulation are conducted in parallel, distributing a workload to each core of the CPU. Since this procedure is inherently sequential and involves a lot of string operations, a highly performing CPU is necessary. When we have JSON files or TXT files, we are

ready to use our PDCM to construct a document for your sentiment model.

6.1 Conversion to Parquet

As mentioned above, the PDCM procedure is comprised of three processes. Firstly, we aggregate our input files (i.e., either JSON or TXT format) into a Parquet Dataframe. This is typically done to consolidate data from multiple sources into a single, unified dataset. Our input files contain hundreds of thousands of large- and medium-sized documents from around 500 firms from the S&P 500. Due to the large amount of documents to handle, we partitioned the data by company. But still, each firm contains many data points as it has at least hundreds to at most thousands of documents. That is, each firm includes approximately 100,000 sequences of words at most. Due to this reason, we achieved the efficiency of the first step with the aid of parallel processing, Apache Spark, and Apache Parquet. We first processed a collection of documents in parallel. PDCM assigns an aggregation task to worker threads. The number of worker threads is equal to the minimum of available CPU cores and the number of firms. That is, each worker is assigned a task of a few firms, making PDCM aggregate a bunch of documents in parallel. Each document of a firm is stored in a Spark data frame in parallel. Then, we stored them in Apache Parquet format.

A Spark DataFrame is a distributed data structure with named columns, designed to help operations such as aggregation, filtering, and grouping. It is fully compatible with Spark SQL for structured data processing [16]. It is distributed across a cluster, allowing for parallel processing of document aggregation. This parallelism significantly speeds up aggregation processing tasks. Spark uses lazy evaluation, meaning it does not immediately transform the data frame. Instead, they optimise the execution plan by recording a lineage of transformations, enabling fault tolerance. Subsequently, their execution plan is more optimised and automatic with the aid of the Catalyst optimiser of Spark, allowing the most efficient execution strategy for the aggregation [113].

Furthermore, our selected Apache Parquet data format synergises with a Spark data frame. Parquet is a column-oriented data file format designed for big data processing [94]. A columnar storage format of Parquet stores data by columns rather than by rows, allowing more efficient data compression, encoding, and retrieval because we aggregate the same types of data stored together. The parquet performs faster and more efficiently in I/O operations, data retrieval, and memory storage to handle big data. Hence, by using Spark DataFrames and Parquet data format, each document in a firm is processed and put into a Spark DataFrame in a parallel way. Subsequently, they are efficiently encoded, compressed and aggregated into a unified Parquet format with Spark's efficient execution plan. Moreover, this first procedure processes an efficient document aggregation and identifies new, deleted, or updated documents to maintain up-to-date data. Whenever PDCM aggregates documents, it detects changes in PostgreSQL metadata and applies the updates to the aggregation process. When data is fed into the PCDM at the initial stage, the metadata of the data is stored in PostgreSQL to track a trace of the data flow. The metadata consists of a hash value of the data, insertion time, data path, and deletion status. With the metadata for each input data, PDCM can keep

up-to-date data by identifying new, deleted, or updated documents.

6.2 Parallel Document-Term-Metric Construction

The second process is to construct the document-term-metric of each firm in a parallel way. We will call the document-term-metric DTM for this report. This is the main and the most time-consuming process as it should build a document-term metric from hundreds of thousands of documents of the S&P 500. Most importantly, Apache Spark distributes construction tasks to Spark workers. That is, the workloads to construct the DTM of each firm are distributed across multiple nodes in the Spark cluster. DTMs of each firm are then built in parallel, significantly speeding up the processing time. We used an RDD (Resilient Distributed Dataset) for this parallelism. It is a distributed collection of objects being processed in parallel across a cluster. RDD is fault-tolerant, meaning that Spark can remember the lost data using the lineage even if there exists a node's failure during processing [113]. This fault tolerance makes the pipeline resilient. Also, as RDDs are designed to handle large datasets, they are suitable for our case where the PDCM constructs around 500 DTM, each building with approximately 100,000 sentences at most.

Furthermore, our inside DTM construction process itself inherently involves multiple sequential steps. These steps include initialising the database connection, fetching existing metadata, identifying new or changed files, processing the data, updating metadata, and handling errors. Each step is designed to optimise the processing workflow. Our DTM files are also stored in Parquet, facilitating I/O operations for the next step with compressed storage for memory management. Note that this process handles massive data points. Thus, it should not build a DTM every time either a new file is uploaded or a file is altered. Like the first process, PDCM can update a DTM with only a new or changed file by tracing the history of data metadata in the PostgreSQL Database.

6.3 Multi-Stage Concatenation

The final stage was called multi-stage concatenation. To predict the US market sentiment, we should aggregate all DTM of each firm, which was a massive amount of documents due to the number of sentences, causing a memory issue. That is, the data we handled was substantial. Our model handled 25,925 SEC filings (i.e., approx 0.7 billion words), 260,000 expert reports (i.e., approx 0.4 billion tokens), and 38,000 earnings call transcripts (i.e., approx 0.4 billion tokens). To combat that issue, we designed a multi-stage merging process by using a batch. The method splits a file path to merge into batches, reads each batch, and combines and writes an intermediate Parquet file. Until the procedure, the model did not consider the character of the dynamic S&P 500. As mentioned above in Chapter 5, the final DTM data only includes an active firm each year through the S&P 500 dynamism filter layer. The filtered files were again stored in intermediate Parquet files. Finally, we read all intermediate Parquet files to create a final merged file, which was the S&P 500 DTM used for predicting the US market sentiment.

Chapter 7

Sentiment Score Prediction Model

In this section, we introduced the sentiment score prediction model (SSPD) in detail. The SSPD is the supervised learning model, and it predicts a sentiment score of a document. In this research, we presented a model that improves the original model[67] to generate various sentiment indexes. Section 7.1 demonstrated the basic skeleton of the SSPD to show how the supervised model predicts a sentiment score. In Section 7.2, we described how we define labels for the prediction. Specifically, return and volatility were used as labels. In Section 7.3, we introduced the Kalman filter and its working process to indicate the macroscopic sentiment trend by removing noise.

7.1 Methodology

7.1.1 Notations

In this research, we handled three types of documents: SEC filings, earnings call transcripts, and financial experts' analysis reports. All three types of documents share the same notation below.

We consider n as a set of documents and V as a vocabulary which contains m words. The i = 1,. . . ,n indicates the document's index. It indicates to both the document's publication date and the firm to which it was associated with. The word counts are recorded in a vector $d_i \in R_+^m$, where $d_{i,j}$ denotes the frequency of word j in the document i. $D \in R_+^{n \times m}$ is defined as a document-term matrix with $D = [d_1, ..., d_n]$, indicating word counts in each document. We define d_i as the i-th row of matrix D, where the column indices are listed in the set S. It is a subset of the vocabulary(i.e., $S \subseteq V$). The submatrix corresponding to the i-th document, restricted to the columns in S, is denoted as $D_{[S],i}$. Similarly, $d_{[S],i}$ represents the word count vector for the i-th document, considering only the terms in S.

We assigned a label y_i to document i, representing the corresponding time series variable- either return or volatility- on the document's publication date. In this research, we presume that each document has a sentiment score, represented as $p_i \in [0, 1]$.

7.1.2 Model Assumptions

7.1.2.1 Assumption 1

We assumed that the vocabulary V was composed of two distinct subsets: S, containing sentiment-charged words, and N, representing natural words, such that $V = S \cup N$ and $S \cap N = \emptyset$. Moreover, we posited that sentiment-charged words influenced the tone of the document, whereas neutral words did not, i.e., $d_{[S],i} \not\perp\!\!\!\perp p_i$ and $d_{[N],i} \perp\!\!\!\perp p_i$. Additionally, we assumed that the sentiment word count was independent of the neural word count, i.e., $d_{[S],i} \perp\!\!\!\perp d_{[N],i}$, implying that the model excluded neutral words.

7.1.2.2 Assumption 2

We assumed that the sentiment-charged word counts $d_{[S],i}$ followed a mixture multinomial distribution:

$$d_{[S],i} \sim \text{Multinomial}\left(s_i, p_i O^+ + (1 - p_i) O^-\right) \tag{7.1}$$

,where s_i represents the total count of sentiment-charged words in the i-th document, defined as: $s_i = \sum_{w \in S} d_{w,i}$, which determines the scale of the multinomial distribution.

We then modelled O as a non-negative matrix in $\in R_+^{|S| \times 2}$, capturing the probabilities of individual word frequencies using a mixture model incorporating both positive and negative sentiment topics.

- O_+ is a vector of $|S|$ non-negative elements with a unit ℓ^1-norm, forming a probability distribution across words:

$$\sum^{|S|} o_{+,w} = 1 \tag{7.2}$$

  ,where $o_{+,w} \in O_+$ and $w \in |S|$.
  It represents the positive sentiment topic, describing the expected word distribution in the document with the strongest positive sentiment ($p_i = 1$).

- O_- follows the same structure but corresponds to the negative sentiment topics, indicating word distribution in the document with the strongest negative sentiment ($p_i = 0$).

The sentiment score p_i, where $0 < p_i < 1$, acts as a mixture coefficient between these two sentiment topics, controlling the blend of positive and negative word distribution in a given document.

7.1.2.3 Assumption 3

The final assumption is that the sentiment score(p_i) fully encapsulates the relevant information(d_i) within a document that influences the dependent variables(y_i). This was expressed as formally:

$$y_i | p_i \perp\!\!\!\perp d_i \tag{7.3}$$

This assumption implied that the sentiment score p_i served as a sufficient summary of the document's content to support a return prediction or volatility. Thus, $\hat{p}$ refers to a predicted sentiment score, representing a feature in predictive models for financial outcomes such as return and volatility.

7.1.3 Model

The model, built on these three core assumptions, followed a three-step process to predict the sentiment score($\hat{p}$) of a document. First, we identified and extracted the set of sentiment-charged words, denoted as $\hat{S}_n$. Next, we estimated the probabilistic distribution matrix for positive and negative topic parameters across words, represented as $\hat{O} = [\hat{O}_+, \hat{O}_-]$. Finally, we predicted the sentiment score($\hat{p}$) of the document via the penalised maximum likelihood estimation. A detailed explanation of each step is provided in Section 7.1.3 to Section 7.1.4, and an overview of the model is illustrated in Figure 7.1.

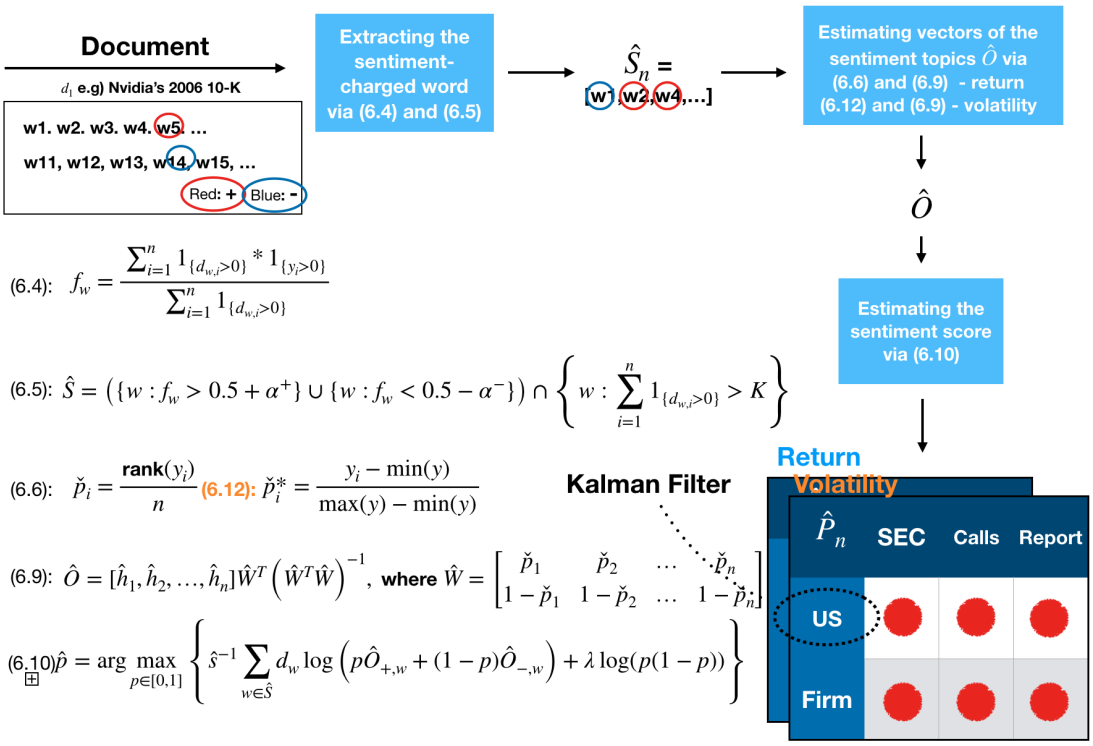

Figure 7.1: A Sentiment Score Prediction Model

7.1.3.1 Extracting Sentiment-Charged Words

In three types of documents, including SEC filings, earnings call transcripts, and analysis reports, sentiment-neutral words typically dominated word count and total frequency. If neutral words were not filtered out, this prevalence could introduce noise and make extracting sentiment-charged words computationally challenging. To combat such an

issue, our model excluded sentiment-neutral words and focused solely on the subset of sentiment-charged words, calculating topic parameters exclusively for this subset. To facilitate this process, we defined the indicator function $1_{\{a\}}$, which takes the value of 1 when the condition a is met and zero otherwise. Specifically, $1_{\{d_{i,w}>0\}}$ signified the presence of word w in the i-th document, while $1_{\{y_i>0\}}$ indicated that the dependent variables(y_i) associated with the i-th document was positive. For each word $w \in V$, we calculated its frequency in a document with positive y_i relative to its overall frequency across all documents using the following equation:

$$f_w = \frac{\sum_{i=1}^{n} 1_{\{d_{w,i}>0\}} * 1_{\{y_i>0\}}}{\sum_{i=1}^{n} 1_{\{d_{w,i}>0\}}} \tag{7.4}$$

This measure captured the relationship between a word's sentiment tone and the dependent variable(y_i, either return or volatility). For a given word w, if its occurrence in documents was more frequently associated with positive y_i (e.g., positive return or low volatility) rather than negative ones, it was classified as having a positive sentiment and vice versa to negative classification.

To refine this classification, we evaluated f_w against predefined thresholds. If f_w was close to 0.5, this suggested that the word was sentiment-neutral and belonged to the set N. To distinguish sentiment-charged words, we introduced threshold parameters α_+ and α_-, both within the interval (0, 0.5], to exclude sentiment-neutral terms. A word was classified as positive if $f_w > 0.5 + \alpha_+$ and as negative $f_w < 0.5 + \alpha_+$.

Additionally, we introduced an occurrence threshold $k \in N$, which specified the minimum frequency count of a word across all documents. This threshold was used to filter out rare words that could introduce noise. For example, if a word appeared only once in all documents, its f_w value would be either 0 or 1, making it an unreliable sentiment indicator. To combat that issue, we required that a word appear more than k times to be included in our analysis: $\sum_{i=1}^{n} 1_{\{d_{w,i}>0\}} > k$. Applying these criteria, we defined our extracted set of sentiment-charged words, $\hat{S}$, as follows:

$$\hat{S} = \left(\{w : f_w > 0.5 + \alpha^+\} \cup \{w : f_w < 0.5 - \alpha^-\}\right) \cap \left\{ w : \sum_{i=1}^{n} 1_{\{d_{w,i}>0\}} > k \right\} \tag{7.5}$$

7.1.3.2 Estimating Probabilistic Distribution of Topic Parameters

In the process, our objective was to estimate the topic parameters for each document. We defined $\hat{O}_i = [\hat{O_i+}, \hat{O_-}]$ as a 1 x 2 vector, where each element represented the estimated probabilistic distribution of either the positive or negative sentiment topic for a given document. To derive the overall topic distribution $\hat{O} = [\hat{O_+}, \hat{O_-}]$, we connected the sentiment expressed in a document to stock returns or volatility. This approach was based on the assumption that stock returns or volatility at the publication date of a document reflected its sentiment. Since direct sentiment scores for documents were not available, we estimated a sentiment proxy. In this research, we have two types of sentiment proxies: return and volatility.

- Return sentiment proxy, defined by:

$$\check{p}_i = \frac{\text{rank}(y_i)}{n}, \tag{7.6}$$

, where $\check{p}_i$ indicates the estimated return sentiment proxy for the i-th document, determined by ranking y_i in ascending order and normalising by n, the total number of documents. In Section 7.2.1, further details of how to calculate return y_i can be found.

- Volatility sentiment proxy, defined by:

$$\check{p}_i^* = \frac{y_i - \min(y)}{\max(y) - \min(y)}, \tag{7.7}$$

, where $\check{p}_i^*$ is the estimated volatility sentiment proxy, restricted between 0 and 1. This is a normalised volatility. Since volatility is inherently different from return, volatility operation requires alterations in the model process. Volatility-based estimation processes as well as how to calculate volatility y_i are explained in detail in Section 7.2.2)

Using these estimated sentiment proxies, we constructed a matrix $\hat{W}$, which contains two rows for each document. The first low captured the estimated positive sentiment proxy $\check{p}_i$, while the second row represented negative sentiment proxy $1 - \check{p}_i$. This structure was expressed in the following equation:

$$\hat{W} = \begin{bmatrix} \check{p}_1 & \check{p}_2 & \dots & \check{p}_n \\ 1 - \check{p}_1 & 1 - \check{p}_2 & \dots & 1 - \check{p}_n \end{bmatrix}, \tag{7.8}$$

This matrix offered a structured representation of the sentiment proxies across all documents, helping further analysis and model fitting.

Subsequently, we adjusted the counts of sentiment-charged words.

$$\hat{h}_i = \frac{d_{[\hat{S}],i}}{\hat{s}_i}, \text{ where } \hat{s}_i = \sum_{w \in \hat{S}} d_{w,i}, \tag{7.9}$$

, where $\hat{h}_i$ was the adjusted counts of sentiment-charged words. $\hat{h}_i$ was defined by dividing $d_{[\hat{S}],i}$, each word frequency within the subset $[\hat{S}]$ of the i-th document, by $\hat{s}_i$, the total number of sentiment-charged words for the document. Then, we stored these relative term counts in a matrix $\hat{H} = [\hat{h}_1, ..., \hat{h}_n]$.

Through having $\hat{H}$ and $\hat{W}$, we aimed to estimate a matrix $\hat{O}$, consisting of parameters that indicate the probability distribution of positive and negative sentiments. To achieve this, we estimated $\hat{O}$ through a linear regression, where the predicted outcome was given by matrix $\hat{H}$, and the predictor was matrix $\hat{W}$. The estimation followed the equation:

$$\hat{O} = \hat{H}\hat{W}^T(\hat{W}\hat{W}^T)^{-1}. \tag{7.10}$$

In the final step, we ensured that the estimated values corresponded to a valid probability distribution. This involved correcting any negative entries by setting them to zero and

then re-normalising each column to ensure their sums equaled one. The adjustment produced a refined matrix; however, for simplicity, we continued to denote it as $\hat{O}$. The first and second columns of this matrix were labeled $\hat{O_+}$ and $\hat{O_-}$, respectively, corresponding to the estimated distributions of positive and negative sentiment topics.

7.1.4 Scoring New Filing

Now that we have all necessary estimators- $\hat{S}$, $\hat{O}$, and $d_{[S]}$- to predict the best probable sentiment score $\hat{p}$. $d_{[S]}$ is drawn from the mixed multinomial distribution at Assumption 7.1.2.2. $\hat{p}$ is inferred via Maximum Likelihood Estimation(MLE):

$$\hat{p} = \underset{p\in[0,1]}{\arg\max} \left\{ \hat{s}^{-1} \sum_{w\in\hat{S}} d_{i,w} \log(p\hat{O}_{+,w} + (1-p)\hat{O}_{-,w}) + \lambda \log(p(1-p)) \right\}, \quad (7.11)$$

where:

- $\hat{s}$ is the total count of sentimet-charged words from the set $\hat{S}$ in the new document.
- $d_{i,w}, \hat{O_{+,w}}$, and $\hat{O_{-,w}}$ represent the w-th elements of their respective vectors.
- λ is a positive constant used to adjust the model.

In the MLE equation, the term $\lambda \log(p(1-p))$ serves as a penalty to prevent overfitting, particularly in cases where documents contain very few sentiment-charged words. For example, if a disclosure includes only a small number of negative sentiment words and no positive words, the model might incorrectly assign it a strong negative sentiment. The penalty term serves as a regularisation factor, pushing the sentiment score toward 0.5, representing a neutral sentiment. In other words, this mechanism encourages the model to be more conservative when evaluating new documents, preventing extreme sentiment classification based on limited data.

7.2 Automatic Labeling

In this section, we explained the types of dependent variables y_i and how they are calculated. Note that we can not attain direct sentiment scores from documents, so we estimated a sentiment proxy using the dependent variables y_i as a label. In this research, we used stock returns and volatility as a label to estimate a sentiment proxy, which was eventually used to predict each document's sentiment score. Through this approach, we can hugely reduce the time for manually labelling a document's sentiment as the labelling mechanism works automatically.

7.2.1 Stock Return y_i or R_i

When using stock returns as a label, we considered the publication date i over the period from the open return at i to the close return at i. The return y_i is calculated before the content of the i-th document is reflected on the stock price. Since most of the three

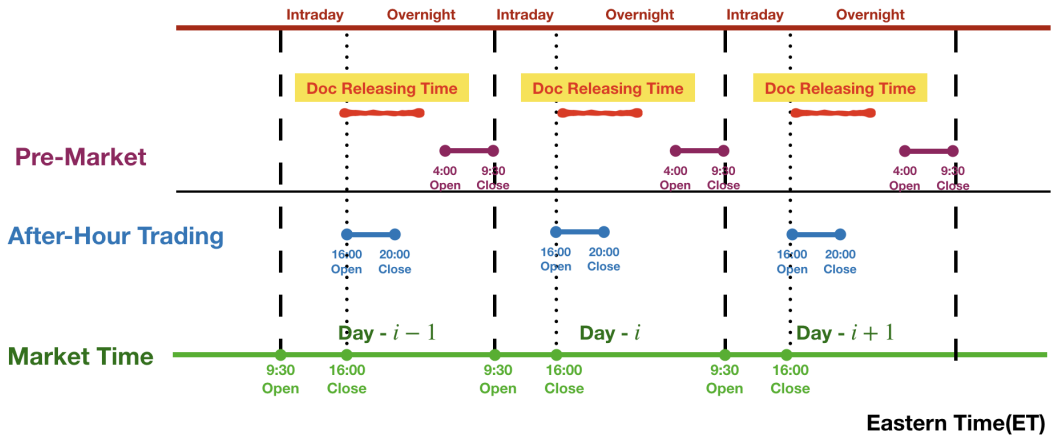

Figure 7.2: Document Publication Timeline

Note that most of the documents (e.g., SEC filings and earnings call transcripts) are usually published after the market closes. Expert reports are typically released during the market. But, dependent variables are calculated based on the document's issue date (i.e., i-th day). Here, we assumed that most of the documents are released after the market closes.

types of documents - SEC filings, earning call transcripts, and financial experts' reports - are often released after market close at i (i.e., at 16:00 U.S Stock Market, Eastern Time), stock return y_i should be calculated with past stock prices to reflect predictive signals. That is, the return should not be trained with future stock prices. Thus, we used open-close log returns R_i:

$$R_i = \log\left(\frac{P_{(i)c}}{P_{(i)o}}\right), \tag{7.12}$$

where P represents the stock price at a given time. Here, i_O and i_C denote the market open and close times on day i , respectively. By using the open-close return, we ensured a symmetric alignment with the document's publication date, providing a more stable measure of market reaction.

7.2.2 Stock Volatility y_i or V_i

The sentiment prediction model is noted for its universal adaptability as proposed in [67]. However, the model's fundamental assumptions and methodologies may not be suitable for all forecasting tasks, especially when using volatility as the target variable instead of returns. While a return-based model aligns naturally with a binary classification framework (i.e., positive sentiment for profit or negative sentiment for loss), volatility does not inherently fit into such a structure. Therefore, modifications are necessary to incorporate volatility as a label in the model. To address this, we introduced a threshold θ, classifying values above θ as high volatility and those below it as low volatility. Consequently, in Equation 7.4, we replaced 0 with θ, and in Equation 7.5, we substituted

0.5 with a quantile q. Note that there is no universally correct threshold or quantile; however, our choice of hyperparameters will influence the model's performance.

Volatility is inherently asymmetric, which means that getting a ranking of the volatility like Equation 7.6 causes significant information loss when estimating sentiment scores. To tackle that, normalizing volatility is suggested as it retains the variable's asymmetric characteristics while improving its usability in sentiment prediction. The normalised volatility is defined above at Equation 7.7. However, while normalizing volatility ensures symmetry around zero, it fails to effectively capture market movement outliers, resulting in suboptimal predictive performance. We incorporated volatility measurements to enhance robustness over multiple days, specifically averaging volatility over three days. We selected the intra-daily range, as it serves as a less noisy measure and reduces distortion [11, 95]. The intra-daily log range is defined as:

$$RG_i = \max_{\tau} \log P_\tau - \min_{\tau} \log P_\tau, \quad \tau \in [t_o, t_c]. \tag{7.13}$$

The volatility proxy $\tilde{V}_i$ was then computed like this:

$$\tilde{V}_i = \frac{RG_i^2}{4\log(2)}, \tag{7.14}$$

where the intra-daily log range was squared and adjusted using the factor$\frac{1}{4\log(2)}$. This adjustment factor compensates for potential biases that may arise in the data generation process, particularly assuming that the process follows Brownian motion with drift [93]. Subsequently, we computed the three-day average of the volatility to establish a more reliable labeling approach:

$$V_i = \frac{1}{3}\left(\tilde{V}_{i-2} + \tilde{V}_{i-1} + \tilde{V}_i\right), \tag{7.15}$$

which is the common approach to calculating multi-day aggregation of daily volatility proxies [31].

7.3 Kalman Filter

In this study, we derived sentiment metrics for both the US market and individual firms (e.g., Nvidia) using a massive amount of documents over a specific period. However, estimating the market's time-varying sentiment measures tends to be highly noisy. We applied the Kalman filter to smooth the noisy signals, obtaining more stable sentiment features in the time series. Precisely, to refine sentiment trends across documents at the market level, we adapted the Kalman filter methodology based on [22] to fit our context. Following this adaptation, our approach to smoothing sentiment data proceeds as follows:

$$\mu_{i+1} = \mu_i + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma_\eta^2) \tag{7.16}$$

$$v_i = \mu_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2) \tag{7.17}$$

We first structured our approach within the Kalman filter framework by defining two key variables: the observed sentiment score
v_t and the unobserved sentiment score μ_t. μ_t was extracted from sentiment scores based on publication dates using Equation 7.16 and Equation 7.17. Here, Equation 7.16 corresponds to the prediction model, while Equation 7.17 indicates the correction model.

In this framework, the prediction model updates the unobserved state based on prior estimates. In contrast, the correction model adjusts those estimates by incorporating the observed sentiment scores influenced by the latent variable. The Kalman filter operates iteratively, alternating between prediction and correction to refine state estimates. A more comprehensive explanation of this Kalman filter model can be found in [35]. The Kalman filter estimates the unobserved sentiment score at time i, given all observations available up to and including time i.

To implement the Kalman filter in our context, sentiment scores were first converted to time series data, as multiple firm-level sentiment scores were associated with the same publication date. To address this, we aggregated all sentiment scores published on a given day using the following formula:

$$\bar{p}_i = \frac{1}{|A_i|} \sum_{i \in A_i} \hat{p}_i, \tag{7.18}$$

where A_i referred to the set of documents published on day i. Next, we substituted v_t with $\bar{p}_t$ in Equation 7.17, and after applying the Kalman filter, we obtained the filtered market-level sentiment score $\tilde{p}_t$. This approach effectively treated all documents published on the same day as equally contributing to the aggregated sentiment score.

While this method enhances the stability of sentiment estimation at the market level, it also introduces a trade-off. Specifically, it reduces the granularity of firm-specific sentiment variations by aggregating into a collective metric from individuals.

Chapter 8

Experiment Results

8.1 System Performance

The real-time sentiment metric generation system was an end-to-end system deployed within the Microsoft Azure cloud, from document collection to sentiment score predictions. The models comprising the system were interconnected seamlessly. The system was executed based on the user-scheduled time. The current system was built on a limited computing specification, where the CPU operated with only four virtualised cores at a base frequency of 2.45 GHz, and the RAM was equipped with 15 GB. Moreover, the environment provided only a virtual GPU without dedicated hardware acceleration. However, as the architecture was mainly CPU-bounded, the existing GPU specification was sufficient for the current goal.

System performance was limited by its hardware constraints in Table 8.1. Nevertheless, it consistently met the requirements for real-time operations. Moreover, since the entire system was containerised via Docker and deployed in a cloud-native environment, it remained highly scalable. It can be effortlessly upgraded to satisfy future performance demands or rising data volumes. For example, the system could perform significantly better with more powerful hardware specifications. At PDCM, the most time-consuming process, SEC Filing took 17 seconds, transcripts took 0.05 seconds, and the article took 0.0036 seconds per document. These tests were conducted on a system with six physical CPU cores at 3.3 GHz and 16 GB of DDR4 RAM.

Seconds	**SEC Filing**	**Transcript**	**Report**
FTRM	0.449	0.720	0.460
PDCM	22.185	31.241	27.413
SSPM	4.231	6.426	5.13
Total	**26.915**	**38.387**	**33.003**

Table 8.1: Real-Time System Running Time Performance
Note: This table represents the average time across 10 tests per document.

8.2 Sentiment Scores Analysis

Through the Sentiment Score Prediction Model (SSPM), our system generates continuous sentiment metrics labeled with return and volatility, reflecting three points of view: the firm's official perspective, the leadership perspective and the aggregated perspective of buy-side experts. In this analysis, conventional evaluation metrics like classification F1 score were not applicable for assessing the performance. Rather than categorising text as positive or negative, SSPM estimates probabilistic sentiment scores ($\tilde{p}^{RET}$, $\tilde{p}^{VOL}$) that are used as features for downstream financial forecasting models. Thus, the true value of the sentiment scores lies in their predictive power. Building upon the [67] framework, which our model extends, we focus on assessing whether the scores convey economically significant signals. Hence, we evaluated the models using descriptive statistics and correlation with actual returns. For robust benchmarking, we utilised Loughran and McDonald's (LM) dictionary-based sentiment model, a method widely adopted in financial text analysis [80]. However, its inherent negativity bias may understate forward-looking optimism. Baseline details are in Appendix E.3

8.2.1 Descriptive Statistics

Table 8.2: Market Level (U.S. Market)

Source	Var	Mean	SD	Max/Min
SEC Filing	$\tilde{p}^{RET}$	0.49	0.01	0.53/0.45
	$\tilde{p}^{VOL}$	0.50	0.03	0.58/0.38
Transcript	$\tilde{p}^{RET}$	0.48	0.01	0.52/0.44
	$\tilde{p}^{VOL}$	0.47	0.03	0.60/0.36
Buy-Side	$\tilde{p}^{RET}$	0.49	0.01	0.51/0.47
	$\tilde{p}^{VOL}$	0.48	0.01	0.52/0.45

Table 8.3: Company Level (NVIDIA)

Source	Var	Mean	SD	Max/Min
SEC Filing	$\tilde{p}^{RET}$	0.49	0.19	0.80/0.19
	$\tilde{p}^{VOL}$	0.44	0.37	0.91/0.09
Transcript	$\tilde{p}^{RET}$	0.49	0.08	0.62/0.39
	$\tilde{p}^{VOL}$	0.53	0.19	0.88/0.28
Buy-Side	$\tilde{p}^{RET}$	0.47	0.05	0.61/0.37
	$\tilde{p}^{VOL}$	0.51	0.10	0.78/0.14

The descriptive statistics in Table 8.2, 8.3 showed the mean, standard deviation, and maximum and minimum range for each signal type across both market and firm (e.g., Nvidia) levels. These descriptive statistics indicate that our SSPM is responsive to different types of texts, companies, and market periods, producing well-distributed sentiment metrics without being static or overly clustered scores around neutral values. These show that our model adjusts its sensitivity at different levels (market vs. company), capturing meaningful variations. While the previous Minf1 research [28] where 10K filings' technology sector (i.e., NASDAQ 100) sentiment showed high variability with SD = 0.24-0.28, the model showed the U.S market (i.e., S&P 500) had low standard deviation with SD = 0.01-0.03. It indicates market sentiment operates as a stable macro-level indicator. In particular, the model captured pronounced variations in company-specific tone, highlighting its potential for robust predictive applications. Additional firms' descriptive analysis is found in the Appendix F.

8.2.2 Sentiment Scores Correlation Analysis

The sentiment correlation analysis showed diverse insights from the stakeholders' viewpoints at the market level at Table 8.4. Sentiment scores for both return and volatility extracted from SEC filings and buy-side reports indicated positive and statistically

Table 8.4: U.S Market Tone Metrics

SEC Filing	$\tilde{p}$-RET	$\tilde{p}$-VOL	$\tilde{p}$-LM	S&P 500 Index
$\tilde{p}$-RET	1			
$\tilde{p}$-VOL	0.196**	1		
$\tilde{p}$-LM	-0.187**	-0.093**	1	
S&P 500 Index	**0.204****	-0.08**	-0.536**	1
Transcripts	$\tilde{p}$-RET	$\tilde{p}$-VOL	$\tilde{p}$-LM	S&P 500 Index
$\tilde{p}$-RET	1			
$\tilde{p}$-VOL	**-0.127****	1		
$\tilde{p}$-LM	0.105**	-0.398**	1	
S&P 500 Index	0.07**	.	.	1
Buy Side	$\tilde{p}$-RET	$\tilde{p}$-VOL	$\tilde{p}$-LM	S&P 500 Index
$\tilde{p}$-RET	1			
$\tilde{p}$-VOL	0.202**	1		
$\tilde{p}$-LM	-0.309**	-0.321**	1	
S&P 500 Index	**-0.486****	-0.034*	0.590**	1

Note: $p-value^{*} < 0.05$, $p-value^{**} < 0.005$

Table 8.5: Nvidia Tone Metrics

SEC Filing	$\tilde{p}$-RET	$\tilde{p}$-VOL	$\tilde{p}$-LM	NVDA Stock
$\tilde{p}$-RET	1			
$\tilde{p}$-VOL	**-0.522****	1		
$\tilde{p}$-LM	0.741**	-0.637**	1	
NVDA Stock	.	0.406**	.	1
Transcripts	$\tilde{p}$-RET	$\tilde{p}$-VOL	$\tilde{p}$-LM	NVDA Stock
$\tilde{p}$-RET	1			
$\tilde{p}$-VOL	**-0.706****	1		
$\tilde{p}$-LM	-0.484**	.	1	
NVDA Stock	.	.	.	1
Buy Side	$\tilde{p}$-RET	$\tilde{p}$-VOL	$\tilde{p}$-LM	NVDA Stock
$\tilde{p}$-RET	1			
$\tilde{p}$-VOL	-0.164**	.		
$\tilde{p}$-LM	-0.203**	.	1	
NVDA Stock	**-0.415****	.	0.233**	1

Note: $p-value^{*} < 0.05$, $p-value^{**} < 0.005$

significant correlations, suggesting that periods of optimism are related to heightened market volatility, whether expressed by firms or financial experts. However, sentiment scores from earnings calls represented an inverse relationship between return and volatility, suggesting that leadership's optimism may alleviate perceived market risk. Notably, SEC-trained return sentiment was positively correlated with S&P 500 index, indicating that broader market performance is aligned with firm-level optimism. However, buy-side optimism had a negative correlation with the index. It implies potential mismatches between actual market dynamics and analyst expectations.

At the firm level, at Table 8.5, stronger and more interpretable correlations appeared. For Nvidia, a consistent negative correlation was revealed between return and volatility sentiment across all perspectives: firm filings, leadership communications, and buy-side reports. It supports the interpretation that Nvidia's risk tends to be alleviated when stakeholders concurrently share optimism. Furthermore, the negative correlation between buy-side return sentiment and Nvidia's stock price suggests analysts' optimism may not reliably predict short-term performance, potentially reflecting over-optimistic forecasting or delayed market reactions. More cases like Tesla and Apple are in Appendix H.3, and H.4.

8.3 System Adaptability

Our system generates real-time sentiment metrics that capture diverse market perspectives, including the firm's official, leadership, and buy-side analysts' views. These metrics apply to various use cases, especially for algorithmic and high-frequency trading. Instant sentiment scores enable traders to respond dynamically to company-specific disclosures, buying on positive signals and selling on negative ones. Prior study found that algorithmic traders applied the informativeness of news sentiment scores into the trading algorithm [6]. Upon the arrival of financial documents, our system automatically retrieves, processes, and predicts sentiment scores in real time, supporting rapid, data-driven trading decisions.

Here, we propose a practical application of our real-time system in high-frequency stock

trading. The model developed in [67], which our SSPM directly built upon, found that sentiment can deliver a superior return prediction signal, achieving an annual Sharpe ratio of 4.21, particularly for small-cap and volatile stocks. Leveraging its short-lived predictive power, we adapt our system for intraday trading [67].

We first identify volatile stocks among the S&P 500 and estimate the expected publication timing of each document. Then we configure a tailored DAG schedule to extract real-time sentiment scores with the estimation. Following [67], we construct a zero-net-investment portfolio - going long stocks with positive SSPM sentiment and short on those with negative sentiment, allocating equal capital to both sides. Each position is held for one trading day.

This setup indicates short-term return predictability in a high-frequency stock trading context: if the sentiment model is accurate, stocks with positive sentiment should rise, while those with negative sentiment should fall the next day, generating profit from the price differential. For instance, if SSPM signals positive tone for Tesla and negative tone for Nvidia, a $1 long for Tesla and a $1 short in Nvidia can yield a return. If Tesla increases 1% and Nvidia drops 2%, the strategy yields a 3% return on zero net capital. Performance is evaluated by daily alpha and Sharpe ratio.

To implement this strategy effectively, an accurate estimation of document release timing is essential, as sentiment efficacy weakens quickly after publication [23, 83, 67]. For example, earnings call calendars offer both publication date and the time, allowing us to estimate the release timing pattern. Over the past 20 years, Nvidia held its earnings call on Wednesday with 75% probability, typically between 10:00 and 13:00 with 65% probability. Since transcripts are typically released a few hours after the call, we can leverage this probabilistic pattern to design a firm-specific DAG schedule that extracts sentiment scores ahead of the broader market. Appendix I represents the earnings call frequency heatmaps for Nvidia, Tesla, and Apple.

In addition to short-term trading, our system also supports mid-long-term investment. Many quantitative asset managers embody sentiment signals into their models. For instance, BlackRock and other leading firms leverage sentiment metrics for market timing, portfolio rebalancing, and asset returns forecasting [5]. Our system produces historical sentiment data that supports both fundamental and technical analysis. Sentiment scores from SEC filings and earnings call offer clues of managerial tone shifting and their risk perception over time, contributing to a broader investment study[14]. Also, time-series sentiment scores may act as momentum indicators, flagging emerging bullish or bearish sentiment trends [67].

Real-time sentiment analysis is also essential for risk management. Sentiment of new information can be an early warning risk indicator for volatility spikes or crises. According to the Mixture of Distributions Hypothesis (MDH), volatility is driven by the rate of information arrivals[86]. Recent research [7] found that GPT-4-based sentiment scoring significantly outperformed traditional news feeds like RavenPack in predicting intraday volatility. From that observation, it is reasonable to assume that one or all of the three sentiment scores by our system can affect intraday volatility. In practice, a risk manager might monitor a dashboard of sentiments from SEC filings, earnings calls, and buy-side reports, as well as news sentiment.

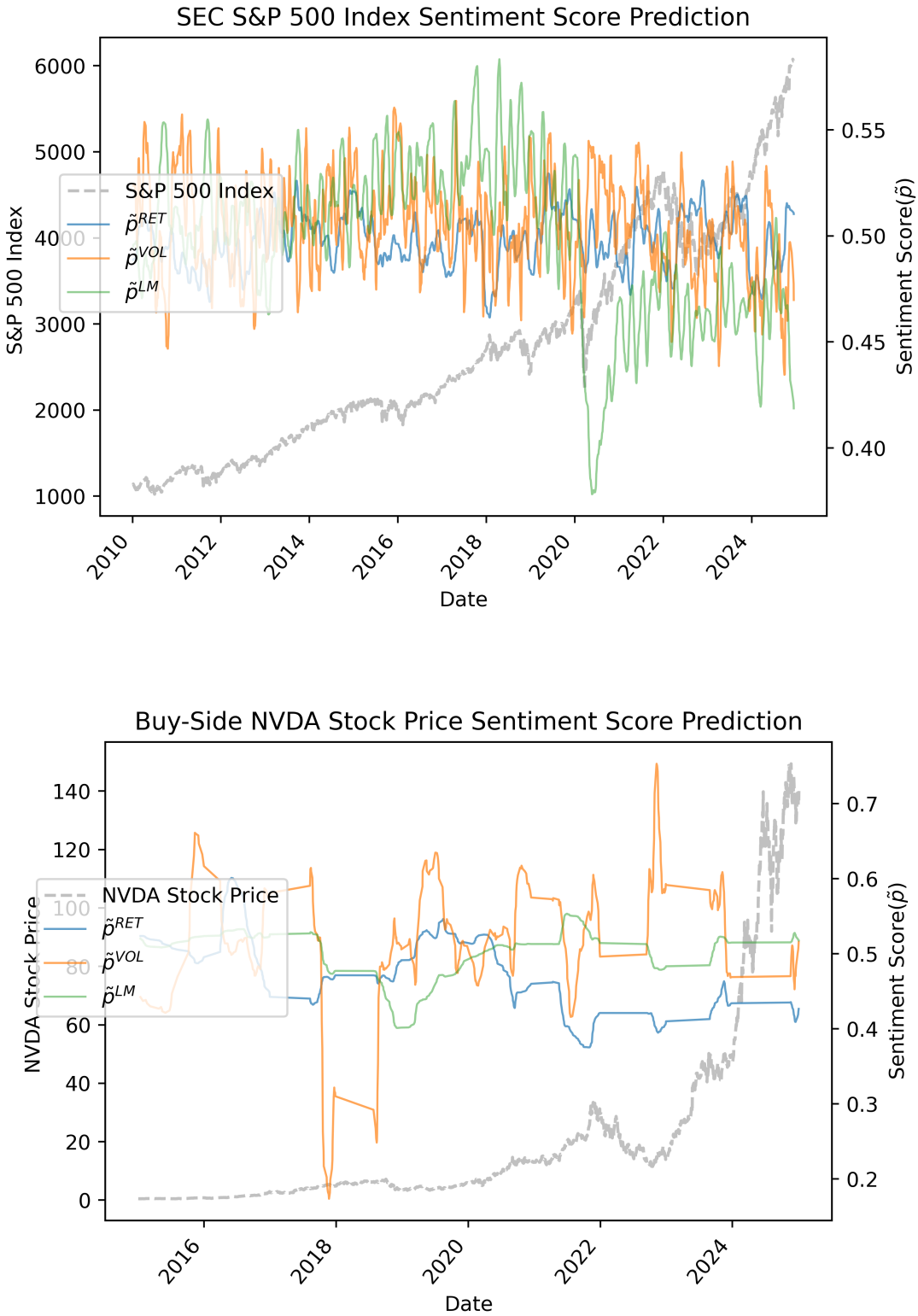

Figure 8.1: Sentiment score predictions: (Top) SEC filings and the S&P 500 Index; (Bottom) Buy-Side reports and NVDA stock price

Chapter 9

Conclusions

9.1 Conclusions

In this research, we developed a real-time automated system to generate sentiment scores with multiple sources - SEC filings, earnings calls, and buy-side analyst reports. These sentiment scores reflect the perspectives of a firm, its leadership, and buy-side analysts, supporting risk management and fundamental analysis. One intriguing finding is that Nvidia's perceived risk is negatively correlated when Nvidia, its leadership, and buy-side analysts shared concurrent optimism on their business. Furthermore, our system provides real-time sentiment metrics from multiple sources, generating multiple sentiment metrics in less than 30 seconds per document under a constrained hardware system. Since the system is Dockerised on Cloud computing, it can provide significantly faster performance than now with minimal efforts, and can be scaled flexibly to meet future demands.

9.2 Limitations and Future Works

We validated the predictive power of return and volatility-oriented sentiment scores from heterogeneous sources through correlation analysis and descriptive statistics. The strong and statistically significant correlations are a proxy for the predictive value of our sentiment scores. In addition, our prediction model is grounded in well-established mathematical theory, and the sentiment metrics based on the returns it generates have already demonstrated strong predictive power, as validated through the portfolio simulation test in [67]. However, portfolio simulation for volatility-based sentiment score remains untested, which future work may pursue to enhance its predictive validity.

In this research, we generated three types of sentiment metrics to reflect various perspectives on the market. These values help to provide a deeper understanding of a market. A recent study [122] proposed a cross-source sentiment synthesis approach, which increased the predictive power of sentiments on the Bitcoin market. We suggest researching our three sentiments fusion model to strengthen the predictive performance of our model further.

Bibliography

[1] Earnings call. Investopedia.

[2] Financial analysis - qlik.

[3] Quotemedia data feed solutions. https://quotemedia.com/solutions/data-feed. Accessed: 2025-03-22.

[4] Wisdom of crowds: The value of stock opinions transmitted through social media.

[5] Artificial intelligence in asset management. Technical report, CFA Institute, 2020.

[6] We show that news analytics can be used for directional trading on company stocks. Technical report, Federal Reserve Board, 2020.

[7] News sentiment and stock return predictability: Evidence from firms in the dow jones index. Technical report, Macquarie University, 2024.

[8] News sentiment in apac markets. Technical report, RavenPack, 2024.

[9] Qingyao Ai, Hongning Wang, et al. News impact analysis with graph neural networks. *Communications of the ACM*, 2024.

[10] Fadi Al-Turjman et al. Deep learning-based sentiment analysis for financial text data. *Cognitive Computation*, 12:1026–1040, 2020.

[11] Sassan Alizadeh, Michael W. Brandt, and Francis X. Diebold. Range-based estimation of stochastic volatility models. *The Journal of Finance*, 57(3):1047–1091, 2002.

[12] Amazon Web Services. Enriching real-time news streams with the refinitiv data library, aws services, and amazon sagemaker. https://aws.amazon.com/ko/blogs/machine-learning/enriching-real-time-news-streams-with-the-refinitiv-data-library-aws-services-and-amazon-sag Accessed: 2025-03-22.

[13] Analytics Vidhya. Stemming vs lemmatization in nlp: Must know differences, 2022. Accessed: 22-03-2024.

[14] Anonymous. Application of sentiment analysis for predicting stock market movement. *WSEAS Transactions on Business and Economics*, 2013.

[15] Anonymous. Foundation models for financial regulation and policy analysis, 2024. arXiv preprint.

[16] Apache Software Foundation. Spark sql, dataframes and datasets guide, 2025. Accessed: 2025-04-06.

[17] Sharad Asthana and Steven Balsam. The effect of edgar on the market reaction to 10-k filings. *Journal of Accounting and Public Policy*, 20(4-5):349–372, 2001.

[18] Ahmet Faruk Aysan, Efe Muğaloğlu, Ali Yavuz Polat, et al. Whether and when did bitcoin sentiment matter for investors? before and during the covid-19 pandemic. *Financial Innovation*, 9(1):124, 2023.

[19] Aurelio F Bariviera. Anomalous volatility, market efficiency, and calendar effects in the cryptocurrency market. *Physica A: Statistical Mechanics and its Applications*, 483:82–90, 2017.

[20] Piyush Bhatia, Yangfeng Ji, and Jacob Eisenstein. Better document-level sentiment analysis from rst discourse parsing. *arXiv preprint*, 2015.

[21] Sander Blomme and Julie Dedeyne. Predicting the effect of 10-k, 10-q and 8-k company reports on abnormal stock returns using finbert nlp methods. Technical report, University of Ghent, 2020.

[22] Svetlana Borovkova and Philipp Lammers. Sector news sentiment indices. *Available at SSRN 3080318*, 2017.

[23] Yi Cai, Zhenpeng Tang, and Ying Chen. Can real-time investor sentiment help predict the high-frequency stock returns? evidence from a mixed-frequency-rolling decomposition forecasting method. *The North American Journal of Economics and Finance*, 72:102147, 2024.

[24] Calcbench. The timing of the earnings press release and the market reaction, 2023. Accessed: 2025-04-02.

[25] S. Che et al. Anticipating corporate financial performance from ceo letters utilizing sentiment analysis. *Mathematical Problems in Engineering*, 2020, 2020.

[26] Hailiang Chen, Prabuddha De, Yu Jeffrey Hu, and Byoung-Hyoun Hwang. Wisdom of crowds: The value of stock opinions transmitted through social media. *Review of Financial Studies (RFS)*, Forthcoming. Available at SSRN: https://ssrn.com/abstract=1807265 or http://dx.doi.org/10.2139/ssrn.1807265.

[27] Min Choi, Hye Jin Lee, Soh Hyung Park, Sung Whan Jeon, and Sungzoon Cho. Stock price momentum modeling using social media data. *Expert Systems with Applications*, 237(Part C):121589, 2024.

[28] Sanggyu Sean Choi. Generation of supervised sentiment metrics for return and volatility prediction from 10-k filings. https://www.inf.ed.ac.uk, 2025. Master's thesis, University of Edinburgh, School of Informatics.

[29] Woojin Chung et al. Sentiment analysis of financial news articles using transformer models. *Journal of Financial Data Science*, 2023.

[30] Sergio Consoli, Luca Barbaglia, and Sebastiano Manzan. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowledge-Based Systems*, 247:108781, 2022.

[31] Fulvio Corsi. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009.

[32] Angela K. Davis, Weili Ge, Dawn Matsumoto, and Jenny Li Zhang. The effect of manager-specific optimism on the tone of earnings conference calls, January 13 2014. CAAA Annual Conference 2012, Available at SSRN: https://ssrn.com/abstract=1982259 or http://dx.doi.org/10.2139/ssrn.1982259.

[33] J.E. Dobson. On reading and interpreting black box deep neural networks. *International Journal of Digital Humanities*, 5:431–449, 2023.

[34] J.S. Doran, D.R. Peterson, and S.M. Price. Earnings conference call content and stock price: The case of reits. *Journal of Real Estate Finance and Economics*, 45(2):402–434, 2010.

[35] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

[36] J. Engelberg. Costly information processing: Evidence from information announcements. In *AFA 2009 San Francisco Meetings Paper*, 2008. Available at SSRN: http://ssrn.com/abstract=1107998.

[37] J. Engelberg, A.V. Reed, and M.C. Ringgenber. How are shorts informed? short sellers, news, and information processing. *Journal of Financial Economics*, 105(2):260–278, 2012.

[38] Lily H. Fang and Joel Peress. Media coverage and the cross-section of stock returns. *The Journal of Finance*, 64(5):2023–2052, 2009.

[39] Xingyuan Fei, Hongkang Xu, and Jiarui Zhang. Linguistic attributes and trade credit: Evidence from textual analysis of earnings conference calls. *Journal of Corporate Accounting and Finance*, 2022. Forthcoming.

[40] Nicky J. Ferguson, Dennis Philip, Herbert Lam, and Jie Michael Guo. Media content and stock returns: The predictive power of press, May 27 2015. Available at SSRN: https://ssrn.com/abstract=2611046.

[41] Alessio Ferrari and Andrea Esuli. An nlp approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, 26(3):559–598, 2019.

[42] S.P. Ferris, G.Q. Hao, and M. Liao. The effect of issuer conservatism on ipo pricing and performance. *Review of Finance*, 7(3):993–1027, 2013.

[43] Financial Times. Tech stocks under pressure amid valuation concerns. https:

//www.ft.com/content/4bd3ab1b-984f-44e7-9b40-43d0802d5672, 2024. Accessed: 2025-03-23.

[44] Lucie Flek. Returning the n to nlp: towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838, 2020.

[45] Forbes. Tesla, amd, and nvidia show higher volatility than bitcoin in october 2024. https://www.forbes.com/sites/digital-assets/2024/11/02/tesla-amd-and-nvidia-show-higher-volatility-than-bitcoin-in-october-2024/, 2024. Accessed: 2025-03-23.

[46] Karen I. Fredriksen-Goldsen and Hyun-Jun Kim. The science of conducting research with lgbt older adults—an introduction to aging with pride: National health, aging, and sexuality/gender study (nhas). 2017.

[47] Lin Gao, Wei Zhang, Yan Yang, et al. Multi-source sentiment analysis for stock price prediction with ensemble learning. *Neural Computing and Applications*, 34:7777–7793, 2022.

[48] Diego Garcia. Sentiment during recessions. *The Journal of Finance*, 68(3):1267–1300, 2013.

[49] SP Global. S&p u.s. indices methodology, 2023. Accessed: 2025-02-17.

[50] Alexander Glodd and Diana Hristova. Extraction of forward-looking financial information for stock price prediction from annual reports using nlp techniques. In *Proceedings of the 56th Hawaii International Conference on System Sciences*, page 10, 2023. Rights: Attribution-NonCommercial-NoDerivatives 4.0 International.

[51] London Stock Exchange Group. Sentiment as a risk indicator: When does it work best?, 2024.

[52] Y. Guo and L. Zhou. Textual tone in corporate financial disclosures: a survey of the literature. *International Journal of Disclosure and Governance*, 17:101–110, 2020. Received October 25, 2019; Published June 27, 2020; Issue Date September 1, 2020.

[53] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain, 1997. Association for Computational Linguistics.

[54] Fei He, Christopher J. Neely, and Jian Wang. Investment decisions based on eeg emotion recognition. *SSRN Electronic Journal*, 2023.

[55] Elaine Henry and Andrew J. Leone. Measuring qualitative information in capital markets research, April 2009. Available at SSRN: https://ssrn.com/abstract=1470807 or http://dx.doi.org/10.2139/ssrn.1470807.

[56] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 168–177, New York, New York, USA, 2004. ACM Press.

[57] Allen H. Huang, Amy Zang, and Rong Zheng. Evidence on the information content of text in analyst reports. *Accounting Review*, April 2014. Forthcoming.

[58] Xuan Huang, Siew Hong Teoh, and Yinglei Zhang. Tone management, August 21 2013. The Accounting Review, Forthcoming, Available at SSRN: https://ssrn.com/abstract=1960376 or http://dx.doi.org/10.2139/ssrn.1960376.

[59] IAEME. Utilization and operational complexity: Apache flink performance analysis. 2021.

[60] IBM. Deep learning. https://www.ibm.com/topics/deep-learning. Accessed: 2024-04-01.

[61] Nitin Indurkhya and Fred J. Damerau. *Handbook of Natural Language Processing*. CRC Press, 2010.

[62] Tejashri Inadarchand Jain and Dipak Nemade. Recognizing contextual polarity in phrase-level sentiment analysis. *International Journal of Computer Applications*, 7(5):12, 2010.

[63] Narasimhan Jegadeesh and Di Wu. Word power: A new approach for content analysis. *Journal of Financial Economics*, 110:712–729, 2013.

[64] R. Johnson and T. Zhang. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, volume 1, pages 562–570, 2017.

[65] Agata Jurek, Maurice D. Mulvenna, and Yaxin Bi. Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4:1–13, 2015.

[66] Christopher Kantos, Dan Joldzic, Gautam Mitra, and Kieu Thi Hoang. Comparative analysis of nlp approaches for earnings calls.

[67] Zheng Ke, Bryan T. Kelly, and Dacheng Xiu. Predicting returns with text data. *University of Chicago, Becker Friedman Institute for Economics Working Paper*, (2019-69), September 2020. Yale ICF Working Paper No. 2019-10, Chicago Booth Research Paper No. 20-37.

[68] Colm Kearney and Sha Liu. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, 33:171–185, 2014.

[69] C. Kim, K. Wang, and L. Zhang. Readability of 10-k reports and stock price crash risk. *Contemporary Accounting Research*, 36:1184–1216, 2019.

[70] Hyeongjun Kim, Ha Young Jang, Myunggyu Kim, and Sungsoo Lim. Analysis of global finance using web scraping and topic modeling. *Journal of Intelligence and Information Systems*, 2022.

[71] Y. Kim. Convolutional neural networks for sentence classification. http://arxiv.org/abs/1408.5882, 2014.

[72] S.P. Kothari, Eric So, and Rodrigo Verdi. Analysts' forecasts and asset pricing: A survey. *Annual Review of Financial Economics*, 8:197–219, 2016.

[73] Ross Koval, Nicholas Andrews, and Xifeng Yan. Forecasting earnings surprises from conference call transcripts. Technical report, University of California, Santa Barbara and Johns Hopkins University and AJO Vista, 2023.

[74] F. LI. The information content of forward-looking statements in corporate filings—a naïve bayesian machine learning approach. *Journal of Accounting Research*, 48:1049–1102, 2010.

[75] Nan Li, Xun Liang, Xinli Li, Chao Wang, and Desheng Dash Wu. Network environment and financial risk using machine learning and sentiment analysis. *Human and Ecological Risk Assessment: An International Journal*, 15(2):227–252, 2009.

[76] Xiaodong Li, Haifeng Chen, Yifan Wang, and Yu Zheng. Financial news impact analysis using machine learning approaches. *Procedia Computer Science*, 122:336–343, 2017.

[77] Zonglin Li, Zheyi Pan, et al. Finnewslm: A large language model for financial news understanding, 2024.

[78] Jiaqi Liu, Hao He, Haoming Xu, et al. Financial news analytics: An explainable news-driven trading system using large language models. *Information Processing Management*, 61(3):103657, 2024.

[79] Mingyu Lo et al. Brain activity of professional investors signals future stock performance. https://www.rsm.nl/discovery/2024/brain-activity-signals-stock-market-success/, 2024. Rotterdam School of Management Discovery.

[80] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

[81] Tim Loughran and Bill McDonald. Ipo first-day returns, offer price revisions, volatility, and form s-1 language. *Journal of Financial Economics*, 109:307–326, 2013.

[82] Bin Lu, Myle Ott, Claire Cardie, and Benjamin K. Tsou. Multi-aspect sentiment analysis with topic models. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 81–88. IEEE, 2011.

[83] Beibei Luo. Research on the impact of investor sentiment on stock returns based on uncertainty. *Highlights in Business, Economics and Management*, 44:181–188, November 2024.

[84] Pelin Ece Mandaci and Ekrem C Cagli. Bitcoin volatility forecasting with a hybrid artificial neural network. *Physica A: Statistical Mechanics and its Applications*, 453:95–102, 2016.

[85] Junaid Maqbool, Preeti Aggarwal, Ravreet Kaur, Ajay Mittal, and Ishfaq Ali Ganaie. Stock prediction by integrating sentiment scores of financial news and mlp-regressor: A machine learning approach. *Procedia Computer Science*, 218:1067–1078, 2023.

[86] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[87] K. Mishev, A. Gjorgjevikj, I. Vodenska, L.T. Chitkushev, and D. Trajanov. Evaluation of sentiment analysis in finance: From lexicons to transformers. *IEEE Access*, 8:131662–131682, 2020.

[88] Moody's Analytics. The power of news sentiment in modern financial analysis, 2023. Accessed: 2025-04-06.

[89] Haejung Na and Soonho Kim. Predicting stock prices based on informed traders' activities using deep neural networks. *Economics Letters*, 204:109917, 2021.

[90] Keane Ong, Wihan van der Heever, Ranjan Satapathy, Erik Cambria, and Gianmarco Mengaldo. Finxabsa: Explainable finance through aspect-based sentiment analysis. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 773–782, 2023.

[91] Outside Insight. Can news and social data facilitate better trading decisions? https://outsideinsight.com/insights/can-news-and-social-data-facilitate-better-trading-decisions/, 2020. Accessed: 2025-03-23.

[92] C.W. Park and E.K. Stice. Analyst forecasting ability and the stock price reaction to forecast revisions. *Review of Accounting Studies*, 5:259–272, 2000.

[93] Michael Parkinson. The extreme value method for estimating the variance of the rate of return. *The Journal of Business*, 53(1):61–65, 1980.

[94] Apache Parquet. Apache parquet: Concepts and architecture. *Apache Software Foundation*, 2024. Accessed: Feb. 6, 2025.

[95] Andrew J. Patton. Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256, 2011.

[96] Viktor Pekar and Jane Binner. Forecasting consumer spending from purchase intentions expressed on social media. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–101, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

[97] Joel Peress and Lily H. FANG. Media and finance. *Oxford Research Encyclopedia of Economics and Finance*, 2019.

[98] Yu Qin and Yi Yang. What you say and how you say it matters: Predicting financial risk using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 390–401, Florence, Italy, 2019. Association for Computational Linguistics.

[99] Gang Rao, Wenbin Huang, Zhen Feng, and Quan Cong. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing*, 308:49–57, 2018.

[100] RavenPack. News Sentiment in APAC Markets. https://www.ravenpack.com/research/news-sentiment-apac, 2023. Accessed: 2025-03-23.

[101] R. Ren, D. D. Wu, and T. Liu. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*, 13(1):760–770, Mar 2019.

[102] M. Rizinski, H. Peshov, K. Mishev, M. Jovanovik, and D. Trajanov. Sentiment analysis in finance: From transformers back to explainable lexicons (xlex). *IEEE Access*, 12:7170–7198, 2024.

[103] Danielle Saunders. *Domain adaptation for neural machine translation*. PhD thesis, University of Cambridge, 2021.

[104] Karo Schouten and Flavius Frasincar. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2015.

[105] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems*, 27(2):12, Mar 2009.

[106] Securities and Exchange Commission. Form 10-k. https://www.sec.gov/files/form10-k.pdf, 2024. Annual Report Pursuant to Section 13 or 15(d) of The Securities Exchange Act of 1934, MB Number: 3235-0063, Expires: December 31, 2026, Estimated average burden hours per response: 2,249.36.

[107] Jun Sha. A data pipeline framework for automated extraction of risk factors from sec-10k filings, 2023.

[108] Dev Shah, Haruna Isah, and Farhana Zulkernine. Predicting the effects of news sentiments on the stock market. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4705–4708. IEEE, 2018.

[109] S.M. Shuhidan, S.R. Hamidi, S. Kazemian, S.M. Shuhidan, and M.A. Ismail. Sentiment analysis for financial news headlines using machine learning algorithm. In *Proceedings of the 7th International Conference on Kansei Engineering and Emotion Research 2018. KEER 2018. Advances in Intelligent Systems and Computing*, volume 739, Singapore, 2018. Springer.

[110] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič. Predictive sentiment analysis of tweets: A stock market application. In *Proc. Int. Workshop Hum.-Comput. Interact. Knowl. Discovery Complex Unstructured Big Data*, pages 77–88, 2013.

[111] Ryan J. Smith and Others. Left frontal eeg power responds to stock price changes. *Frontiers in Neuroscience*, 15:664354, 2021.

[112] S. Sohangir, D. Wang, A. Pomeranets, and T.M. Khoshgoftaar. Big data: Deep learning for financial sentiment analysis. *Journal of Big Data*, 5(1):3, Dec 2018.

[113] Apache Spark. Apache spark: Sql programming guide. *Apache Software Foundation*, 2024. Accessed: Feb. 6, 2025.

[114] Investopedia Staff. Financial analysis: Definition, importance, types, and examples, 2023. Accessed: 2025-04-06.

[115] K. Sheng Tai, R. Socher, and C.D. Manning. Improved semantic representations from tree-structured long short-term memory networks. http://arxiv.org/abs/1503.00075, 2015.

[116] D. Tang, B. Qin, and T. Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, 2015.

[117] Khalid Tatane, Mohamed Sahib, and Zaki T. From time series to images: Revolutionizing stock market predictions with convolutional deep neural networks. *International Journal of Advanced Computer Science and Applications*, 15:474, 01 2024.

[118] Muath Tubishat, Norita Idris, and Mohammad A. M. Abushariah. Implicit aspect extraction in sentiment analysis: Review, taxonomy, opportunities, and open challenges. *Information Processing & Management*, 54:545–563, 2018.

[119] Panagiotis Tzouvanas, Renatas Kizys, and Michael Donadelli. News and stock returns: A study on sentiment analysis of news articles. *Expert Systems with Applications*, 42(24):9603–9615, 2015.

[120] U.S. Securities and Exchange Commission. Form 10-Q: Quarterly Report Pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934. https://www.sec.gov/files/form10-q.pdf, 2024. Accessed: 2025-04-04.

[121] U.S. Securities and Exchange Commission. Analyzing analyst recommendations, n.d. Accessed: 2025-04-02.

[122] Anh Tuan Vo, Quang Hieu Bui, Minh Nguyen, Quan Tran, and Thanh Tam Le. Multisource financial sentiment analysis for detecting bitcoin price change indications using deep learning. *Neural Computing and Applications*, 2022.

[123] Kai Waehner. Apache kafka in financial services: Open banking, api, finserv, payment fraud, middleware, messaging, and transactions. 2021.

[124] Gang Wang, Tianyi Wang, Bolun Wang, Divya Sambasivan, Zengbin Zhang, Haitao Zheng, Ben Zhao, and Santa Barbara. Crowds on wall street: Extracting value from collaborative investing platforms. 03 2015.

[125] Jiawei Wang, Linpeng Huang, et al. Explainable financial event detection based on news articles using hybrid deep learning model. *Procedia Computer Science*, 217:546–553, 2023.

[126] Peng Wang, Wei Li, and Liang Wang. Financial news prediction using deep learning and nlp techniques. In *Proceedings of the 24th International Conference on World Wide Web Companion*, 2015.

[127] Z. Wang, Z. Hu, F. Li, et al. Learning-based stock trending prediction by incorporating technical indicators and social media sentiment. *Cognitive Computation*, 15:1092–1102, 2023. Received September 15, 2022; Accepted February 9, 2023; Published March 9, 2023; Issue Date May 1, 2023.

[128] Wiki. Form 10-k. https://en.wikipedia.org/wiki/Form_10-K, 2024. Wikipedia page.

[129] Wikipedia. Pearson correlation coefficient — Wikipedia, the free encyclopedia, 2024. [Online; accessed 24-March-2024].

[130] Wikipedia contributors. Buy-side analyst, n.d. Accessed: 2025-04-02.

[131] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433, 2009.

[132] Yao Xue, Bo Wang, et al. A sentiment-driven stock prediction model with lstm and attention mechanism, 2022. arXiv preprint.

[133] Bishan Yang and Claire Cardie. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 325–335, 2014.

[134] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[135] H. You and Xj Zhang. Financial reporting complexity and investor underreaction to 10-k information. *Review of Accounting Studies*, 14:559–586, 2009.

[136] Jingyuan Zhang, Jiawei Zhang, et al. News impact analysis using deep learning and natural language processing techniques. *Frontiers in Artificial Intelligence*, 5:836809, 2022.

[137] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(e1253), 2018.

[138] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 649–657, 2015.

[139] Xiaodong Zhang, Hauke Fuehres, and Peter A Gloor. Sentiment analysis for chinese stock market prediction using deep neural networks. *Expert Systems with Applications*, 42(21):10567–10574, 2015.

[140] Yumo Zhang, Hauke Fuehres, and Peter Gloor. Sentiment-aware stock market prediction: A deep learning method. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5288–5295. IEEE, 2019.

[141] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities, 2020.

Appendix A

Textual Sentiment Analysis Level

A.0.1 Document Level

Document-level analysis is performed on an entire document, which gains a single polarity. This approach is not usually used often. The chapters of a book are classified as positive, negative, or neutral. To classify the document, both supervised and unsupervised learning approaches are employed [20]. Cross-domain and cross-language sentiment analysis are the most significant problems at this level of analysis [103]. Domain-specific document sentiment analysis has been performed with remarkable accuracy, but this sentiment is highly sensitive to the domain.

A.0.2 Sentence Level

The sentence-level analysis is focused on each sentence in a document. The analysed individual sentences are assigned the corresponding polarity. The main objective is to determine whether each sentence's tone is positive, negative, or neutral. This approach is particularly beneficial when a document consists of a diverse array of sentiments [133]. Individual sentence polarity will be calculated independently through the same methodologies as the document level. It requires greater training data and processing resources. Subsequently, the document's sentiment is found by aggregating the polarity of each sentence. Sentiment-level analysis focuses on finding subject sentences, as this analysis is related to subjective classification [99]. This sentiment-level sentiment analysis is important in the task of having some sentences, such as conditional sentences or ambiguous statements [41].

A.0.3 Phrase Level

Phrase-level analysis is focused on opinion words, which can be mined at the phrase level. Each opinion word is then classified. Individual phrases may include multiple aspects or single aspects. This analysis has recently become a popular topic among researchers. A word is the fundamental building block of language, and its polarity is closely associated with the subjectivity of the sentence or document where it occurs. Sentences that contain adjectives are highly likely to be subjective [46]. Furthermore,

an adjective indicates the demographic features of individuals, such as gender and age, along with desires, social status, personality, and other psychological and social traits. Thus, a word is a foundation for text sentiment analysis [44].

A.0.4 Aspect Level

Aspect level analysis performs fine-grained analysis as it needs to find multiple aspects within a sentence where several aspects exist. Multiple aspects in the sentence were assigned polarity, and then these polarities were aggregated to calculate the sentiment of the whole sentence [104, 82]. For instance, consider the following sentence: "Increased tariffs could decrease our profit margins." The risk evaluation is on tariffs, which is an aspect of the firm's profit, and the evaluation is negative. Hence, this level of task helps to figure out directly whether a firm is profitable or not [61]. According to [118], a variety of aspect extraction techniques were introduced from different points of view.

Recently, several authors have studied sentiment analysis in a financial context through aspect-based approaches [90, 30]. [90] applied aspect-based sentiment analysis on social media text to calculate sentiment scores for individual aspects in financial literature. [30] proposed the aspect-based sentiment model, which is unsupervised and customised to the financial domains to identify the sentiment associated with topics of interest in individual sentences and assign polarity scores to those topics. In this paper, we concentrated on the phrase-level approach to identify the sentiment-charged words within each document. However, compared to most of the previous approaches using a large lexicon of words labelled with their prior polarity [131, 62], we utilised an innovative polarity system on a term. This innovative polarity system is so scalable that it can be employed in various domains where a relevant resource exists. In our specific domain of finance, the term polarity is determined with various labels, such as return, volatility, S&P 500 price index, and S&P 500 volatility index. Thus, our research on finance offers a broad range of sentiments from different views, which makes our model more useful, innovative, and scalable.

Appendix B

10-K Report Form

Part 1

Item 1: Business - This section describes the business of the company: its main products or services, subsidiaries, and markets in which it operates. It may also contain recent events, competition, regulation, and labour problems.

Item 1A: Risk Factors - This section provides risks and uncertainties, likely external effects, or possible failures that could affect their financial performance. Risk factors are generally enumerated based on their importance.

Item 1B: Unresolved Staff Comments - This section offers an explanation of any issues raised by the SEC staff on the previous reports if these issues have not been resolved afterwards.

Item 2:Properties - This section only lays out the company's significant physical properties, not intellectual or intangible property.

Item 3: Legal Proceedings - This section discloses any significant ongoing lawsuit or other legal proceeding.

Item 4 - [RESERVED]

Part 2

Item 5: Market for Registrant's Common Equity, Related Stockholder Matters and Issuer Purchases of Equity Securities - This section discloses their performance in the stock market, dividends, and repurchases of their own stocks.

Item 6 - [RESERVED]

Item 7: Management's Discussion and Analysis of Financial Condition and Results of Operations (MDA) - This section discusses the firm's management for its financial performance, challenges, chances, and future outlook.

Item 7A: Quantitative and Qualitative Disclosures about Market Risks - This section shows the company's exposure to market risks.

Item 8: Financial Statement and Supplementary Data - This section offers the audited financial statements. It contains the balance sheet, income statement, cash flow statement, and footnotes.

Item 9: Changes in and Disagreements with Accountants on Accounting and Financial Disclosure - Companies address any alterations in or disputes with their accountants over financial reporting.

Item 9A: Controls and Procedures - This section details the company's procedures for disclosure controls and its internal control mechanisms for financial reporting.

Item 9B: Other Information - This section offers any additional information that does not align with the contents of other sections.

Part 3

Item 10: Directors, Executive Officers and Corporate Governance - This section delves into the specifics of the company's leadership, their respective roles, and the practices in place for corporate governance.

Item 11: Executive Compensation - This section addresses compensation policies and programmes, and the compensation of top executives.

Item 12: Security Ownership of Certain Beneficial Owners and Management and Related Stockholder Matters - This section offers major shareholders' ownership of the company's stock as well as insiders.

Item 13: Certain Relationships and Related Transactions, and Director Independence- This section encompasses transactions involving directors, executives, and their affiliates, along with details regarding the independence of directors.

Item 14: Principal Accountant Fees and Services - The section details the fees charged by the company's auditors for their services.

Part 4

Item 15: Exhibits, Financial Statement Schedules - This section contains a list of the financial statements and exhibits.

[106, 128]

Appendix C

10-Q Report Form

Part 1 Covers financial information, including financial statements, management's discussion, and analysis of the company's financial condition

Part 2 Covers other relevant information, such as legal proceedings, changes in internal controls, and unregistered securities sales. [120]

Appendix D

Financial Text Retrieval Model Example

D.1 10-K Filing (e.g Nvidia 2024-02-21)

...

[heading]Our Company[/heading]
NVIDIA Corporation helped awaken the world to the power of computer graphics when it invented the graphics processor unit, or GPU, in 1999. Expertise in programmable GPUs has led to breakthroughs in parallel processing which make supercomputing inexpensive and widely accessible.

...

[heading]Item 7. Management's Discussion and Analysis of Financial Condition and Results of Operations [/heading] The following discussion and analysis of our financial condition and results of operations should be read in conjunction with "Item 1A. Risk Factors", our Consolidated Financial Statements and related Notes thereto, as well as other cautionary statements and risks described elsewhere in this Annual Report on Form 10-K, before deciding to purchase, hold or sell shares of our common stock.

...

[heading]Overview[/heading] [heading]Our Company[/heading]
NVIDIA Corporation helped awaken the world to the power of computer graphics when it invented the graphics processor unit, or GPU, in 1999. Expertise in programmable GPUs has led to breakthroughs in parallel processing which make supercomputing inexpensive and widely accessible. We serve the entertainment and consumer market with our GeForce graphics products, the professional design and visualization market with our Quadro graphics products, the high-performance computing market with our Tesla computing solutions products, and the mobile computing market with our Tegra system-on-a-chip products.

...

D.2 10-Q Filing (e.g Nvidia 2024-08-28)

...

[heading]Item 2. Management's Discussion and Analysis of Financial Condition and Results of Operations[/heading] [heading]Forward-Looking Statements[/heading] [heading]This Quarterly Report on Form 10-Q contains forward-looking statements based on management's beliefs and assumptions and on information currently available to management. In some cases, you can identify forward-looking statements by terms such as "may," "will," "should," "could," "goal," "would," "expect," "plan," "anticipate," "believe," "estimate," "project," "predict," "potential" and similar expressions intended to identify forward-looking statements. These statements involve known and unknown risks, uncertainties and other factors, which may cause our actual results, performance, time frames or achievements to be materially different from any future results, performance, time frames or ...

...

The following discussion and analysis of our financial condition and results of operations should be read in conjunction with the risk factors set forth in Item 1A. "Risk Factors" of our Annual Report on Form 10-K for the fiscal year ended January 28, 2024 and Part II, Item 1A. "Risk Factors" of this Quarterly Report on Form 10-Q and our Condensed Consolidated Financial Statements and related Notes thereto, as well as other cautionary statements and risks described elsewhere in this Quarterly Report on Form 10-Q and our other filings with the SEC, before deciding to purchase, hold, or sell shares of our common stock.

...

D.3 Earning Call Transcripts (e.g Nvidia 2024-08-28)

"Operator: Good afternoon. My name is Abby and I will be your conference operator today. At this time, I would like to welcome everyone to NVIDIA's Second Quarter Earnings Call.

...

Q2 was another record quarter. Revenue of $30 billion was up 15% sequentially and up 122% year-on-year and well above our outlook of $28 billion. Starting with data center, data center revenue of $26.3 billion was a record, up 16% sequentially and up 154% year-on-year, driven by strong demand for NVIDIA Hopper, GPU computing, and our networking platforms.

...

And as Colette mentioned, NVIDIA software will exit the year at a $2 billion run rate. Thank you all for joining us today.: Ladies and gentlemen, this concludes today's call and we thank you for your participation. You may now disconnect.

D.4 Expert's Analyst Report (e.g Nvidia 2025-02-03)

Summary: "Shares of Nvidia soared 171% in 2024 but quieted in the back half of the year.", "NVDA will present at two of the biggest conferences of the year this week and next.", "With its Q4 earnings report not due out until late February, investors may pay close attention to Jensen Huang2̆019s comments on Blackwell chips and how AI demand is trending in the coming days."

Appendix E

Experimental Setup

E.1 Preprocessing

Through the Financial Text Retrieval Model (FTRM) at Chapter 4, we finally collected almost all texts across three types - SEC filings, earnings call transcripts, and buy-side reports - from 2006 to 2025 of the S&P 500. We preprocessed these files before feeding them into our prediction model. Firstly, we split all sentences of a document into words. Secondly, we removed non-alphabetic tokens from the texts, such as numbers, proper nouns, and special characters(i.e. punctuations). For instance, Item 8 of a 10-K filing contains many numbers as a financial statement of a company is included in that section. As numbers were not necessary for textual analysis, we removed them. Thirdly, we removed stop words such as ”is”, ”the”, or ”and”, as they do not contain informative information. In the final step of the preprocessing, we selected lemmatisation, instead of stemming. Although lemmatisation is computationally more expensive than stemming, it tends to capture the more accurate base form of a word through linguistic analysis (i.e. considering Parts-of-speech) [13].

E.2 Hyper-parameter Setting

We set the hyper-parameters equal for the models to ensure a fair comparison between the sentiments with return labels and those with volatility labels for a market level, and a company level.

In Section 7.2.2, we replaced 0 by the θ in Equation 7.4 , and the value 0.5 by quantile q in Equation 7.5 . θ is a threshold, and we used it to define high and low volatility. To figure out the balanced θ and q hyper-parameter for both the the U.S market and a firm(in this paper, Nvidia), we selected the value of θ and q from Figure E.1, Figure E.2 on the training windows. In the technology sector from Figure E.1, we practically set θ as the 65th percentile of the distribution and q as 0.65. It implies we defined the volatility above the 65th quantile as high volatility, whereas the volatility below the 65th quantile is low. Note that we set the 65th percentile and 0.65 for θ and q for both the U.S market volatility. We can see the peaks from the graph, referring to the financial crisis

of 2008 and the COVID-19 pandemic around 2020. In the case of a firm(i.e. Nvidia) from Figure E.2, θ was set as the 65th percentile of the distribution, and q was 0.65. Similar to the U.S market volatility graph, Nvidia experienced high levels of volatility during the 2008 financial crisis, and COVID-19 showed higher volatility movement in general during the training windows.

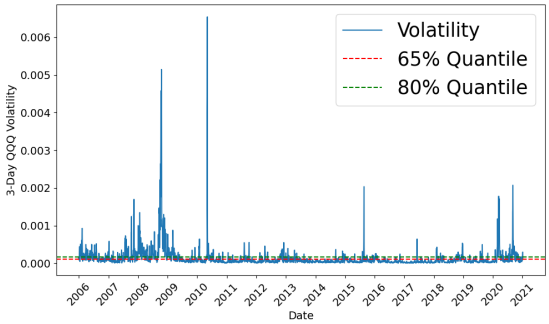

Figure E.1: 3-Day SP500 Volatility

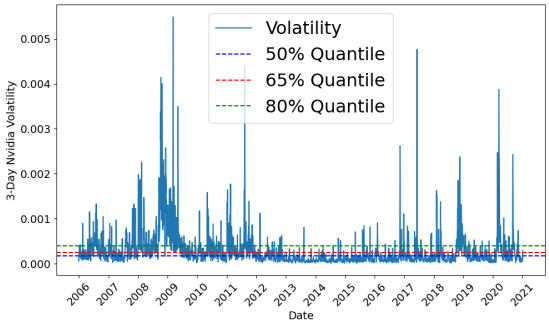

Figure E.2: 3-Day Nvidia Volatility

Furthermore, returns for the U.S market and a firm showed a balanced proportion at the median of three-day returns. Hence, we set 0.5 at Equation 7.5 for both the technology sector in Figure E.3 and a firm return in Figure E.4.

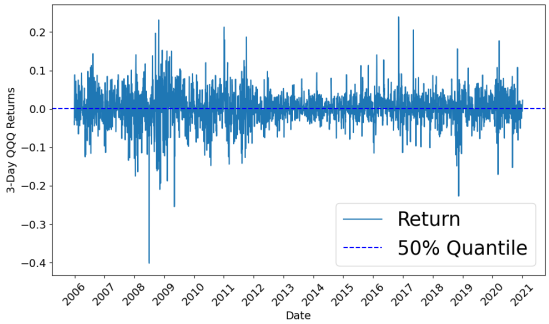

Figure E.3: 3-Day SP500 Return

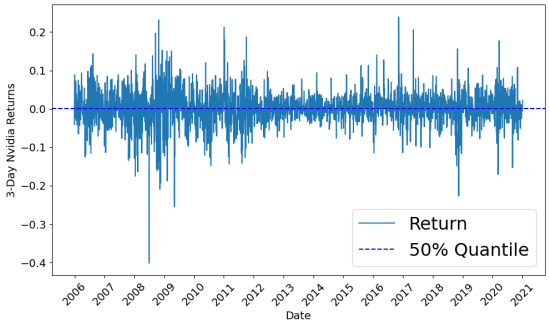

Figure E.4: 3-Day Nvidia Return

In Equation 7.5, we specifically defined α and k. Note that α was a threshold to filter out sentiment-neutral words. We set α^+ and α^- within the interval $(0, 0.5]$ such that each of the positive word sets and the negative word sets includes 100 words. Moreover, note that $k \in N$ was another threshold to relate to the count of word w across all filings such that we used k as a minimum frequency requirement to reduce the influence of rare words. We set k as the 90 percentile quantile of the term frequency distribution. It means we ignored words that appear less than the 90 percentile quantile in a filing.

In Equation 7.10, we also defined λ. It was a positive constant and used in a penalty term to adjust our model. The penalty term was used to prevent the model from overfitting when few sentiment-charged words appear in the document. Without the penalty term, the models can consider the filing that contains few negative words but does not include positive words as a negative filing. However, just because the model contains few negative sentiment-charged words without positive words, that does not mean the filing has a negative tone. To control this phenomenon, we set the penalty coefficient λ to 0.1 in the penalty term.

E.3 Baseline

The purpose of the paper was to predict the sentiment score for both the U.S market and a firm from various financial texts. To achieve that, we could infer the sentiment scores for $\hat{p}^{RET}$ and $\hat{p}^{VOL}$ by labelling with return and volatility, respectively. Then, we introduced a baseline sentiment score to compare our sentiment scores with it.

Our baseline model to calculate baseline sentiment scores was suggested by [48], and used the dictionary created by [80]. Our baseline Loughran and McDonald(LM) sentiment score,$\hat{p}^{LM}$, is computed as

$$\hat{p}_i^{LM} = \left(\sum_{w \in LM^+} d_{i,w} - \sum_{w \in LM^-} d_{i,w} \right) \left(\sum_{w \in V} d_{i,w} \right)^{-1}, \tag{E.1}$$

where LM^+ refers to the positive word lists and LM^- refers to the negative words list of Loughran and McDonald's dictionary. To attain the base sentiment scores, we calculated the difference between the number of positive words and the number of negative words, and then we divided this result by the total number of words in each document.

To explain more of the LM's dictionary for our robust evaluation, the LM dictionary is traditionally used for financial analysis. LM offered an improved textual dictionary for a more accurate financial analysis in financial documents. Existing the financial dictionary of 10-K filings based on the Harvard dictionary misclassified the negative words in a financial context. Three-fourths of negative words in the 10-K filings do not carry a negative connotation in financial reports. To solve this issue, LM suggested three approaches. Firstly, LM created a refined list of words that more accurately reflects negative sentiment in a financial context, by analysing every word that appears in at least 5% of the SEC's 10-K filings. Secondly, LM introduced a term weighting scheme that controls the influence of frequently mentioned words and amplifies the significance of rarer terms, thus mitigating the misclassification of words. Finally, they added five other word classifications(e.g., positive, uncertainty, litigious, strong modal, and weak modal words). They found that these new classifications can be linked to market reactions, volatility in stock returns, unexpected earnings, and trading volumes [80]. In our study, we used only negative and positive categories to adjust the financial dictionary for our model.

Appendix F

Pearson Correlation

F.1 Definition

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}, \tag{F.1}$$

- r refers to the correlation coefficient.
- x_i and y_i refer to the individual sample points for variables x and y, respectively.
- $\bar{x}$ and $\bar{y}$ represent the mean values of the samples for x and y.

[129]

Appendix G

Descriptive Analysis

Table G.1: Tesla

Source	Var	Mean	SD	Max/Min
SEC Filing	$\tilde{p}^{RET}$	0.50	0.18	0.80/0.16
	$\tilde{p}^{VOL}$	0.50	0.33	0.90/0.10
Transcript	$\tilde{p}^{RET}$	0.48	0.04	0.53/0.43
	$\tilde{p}^{VOL}$	0.56	0.11	0.73/0.36
Buy-Side	$\tilde{p}^{RET}$	0.50	0.04	0.63/0.39
	$\tilde{p}^{VOL}$	0.58	0.11	0.76/0.20

Table G.2: Apple

Source	Var	Mean	SD	Max/Min
SEC Filing	$\tilde{p}^{RET}$	0.36	0.28	0.88/0.13
	$\tilde{p}^{VOL}$	0.43	0.34	0.90/0.11
Transcript	$\tilde{p}^{RET}$	0.51	0.05	0.42/0.56
	$\tilde{p}^{VOL}$	0.51	0.24	0.90/0.14
Buy-Side	$\tilde{p}^{RET}$	0.56	0.06	0.67/0.40
	$\tilde{p}^{VOL}$	0.53	0.07	0.74/0.33

Appendix H

Sentiment Score Prediction Model Continuous Results

H.1 the U.S market

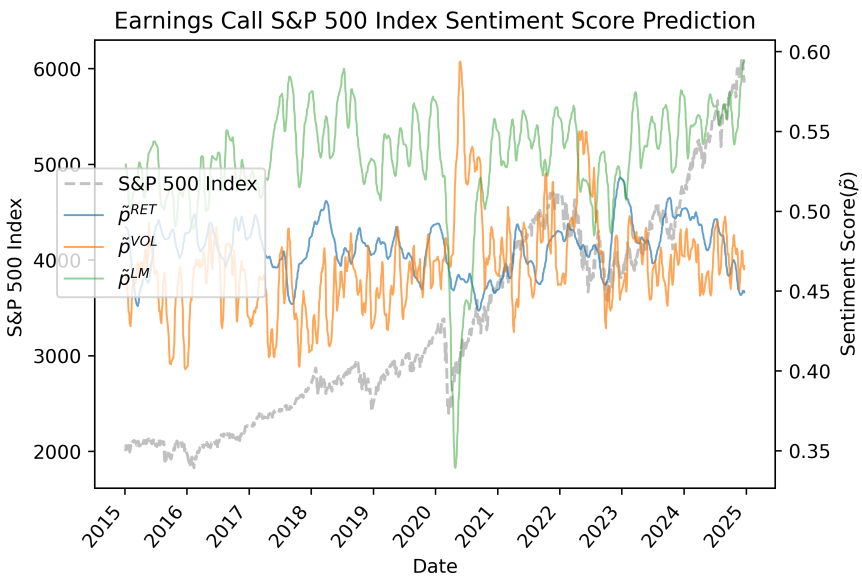

Figure H.1: Earnings Call sentiment score

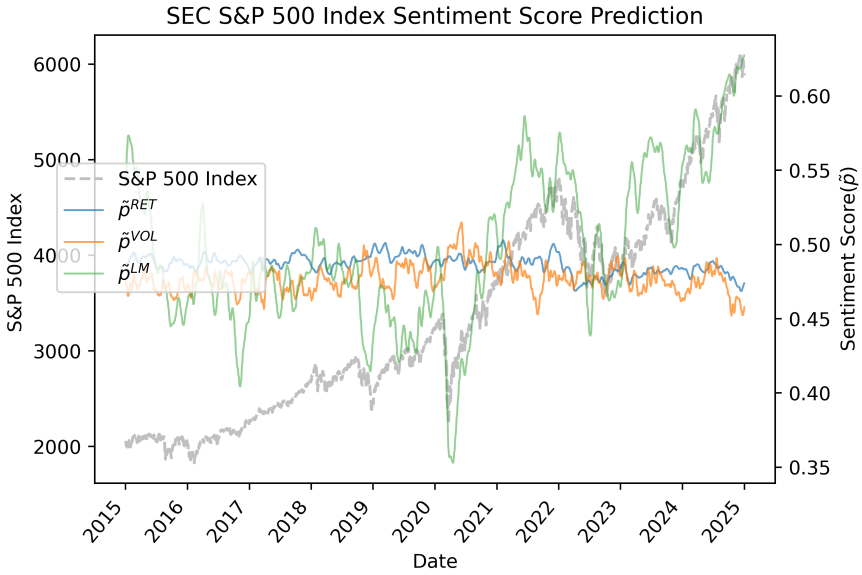

Figure H.2: Buy-Side sentiment score

H.2 Nvidia

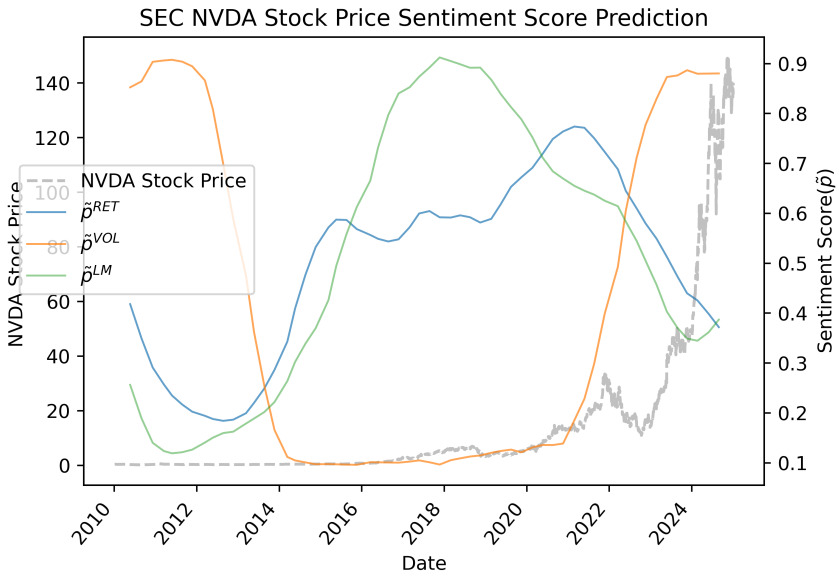

Figure H.3: SEC filing sentiment score

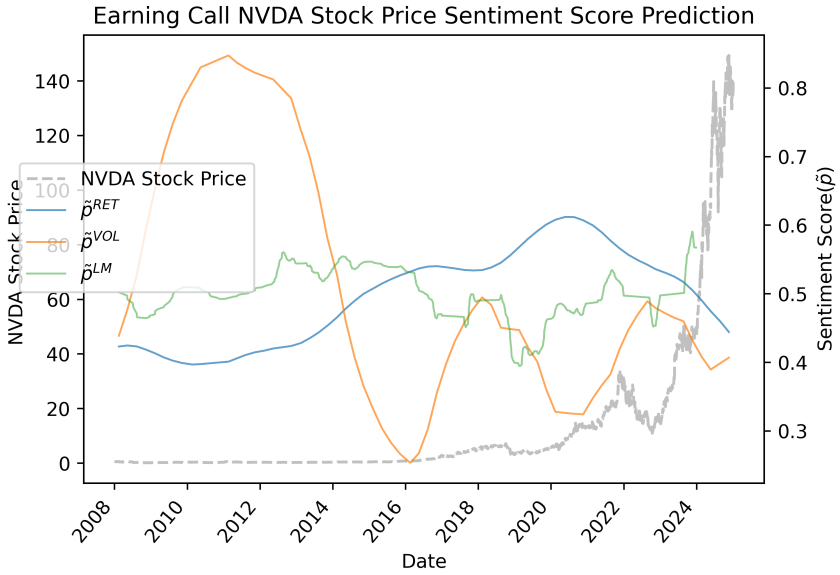

Figure H.4: Earnings Call sentiment score

H.3 Tesla

Table H.1: Tesla Tone Metrics

SEC Filing	$\tilde{p}^{\text{RET}}$	$\tilde{p}^{\text{VOL}}$	$\tilde{p}^{\text{LM}}$	TSLA Stock
$\tilde{p}^{\text{RET}}$	1			
$\tilde{p}^{\text{VOL}}$	0.355**	1		
$\tilde{p}^{\text{LM}}$	-0.215	-0.534**	1	
TSLA Stock	0.480**	**0.729****	-0.123	1
Transcripts	$\tilde{p}^{\text{RET}}$	$\tilde{p}^{\text{VOL}}$	$\tilde{p}^{\text{LM}}$	TSLA Stock
$\tilde{p}^{\text{RET}}$	1			
$\tilde{p}^{\text{VOL}}$	-0.516**	1		
$\tilde{p}^{\text{LM}}$	0.795**	-0.675**	1	
TSLA Stock	0.485**	**-0.863****	0.549**	1
Buy Side	$\tilde{p}^{\text{RET}}$	$\tilde{p}^{\text{VOL}}$	$\tilde{p}^{\text{LM}}$	TSLA Stock
$\tilde{p}^{\text{RET}}$	1			
$\tilde{p}^{\text{VOL}}$	0.225**	1		
$\tilde{p}^{\text{LM}}$	0.193**	0.325**	1	
TSLA Stock	0.498**	**0.709****	0.566**	1

$$Note\text{: } p - value^{**} < 0.005$$

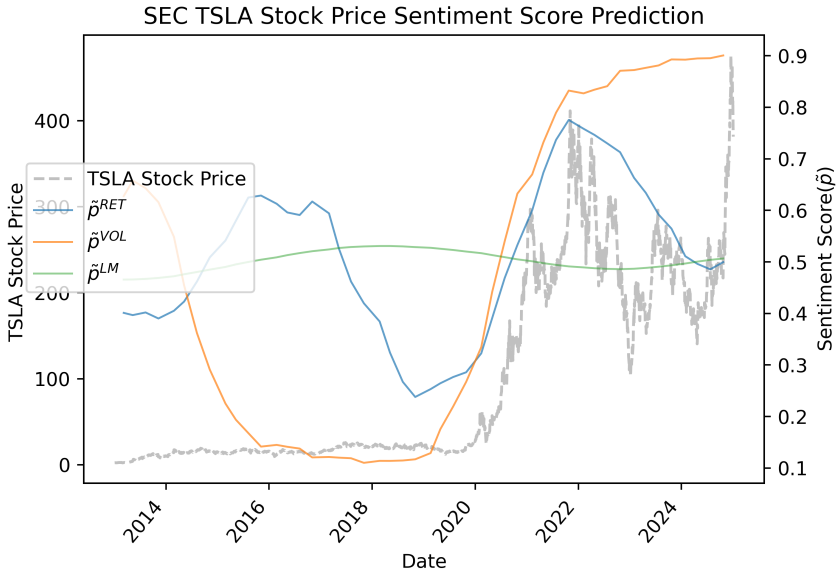

Figure H.5: SEC filing sentiment score

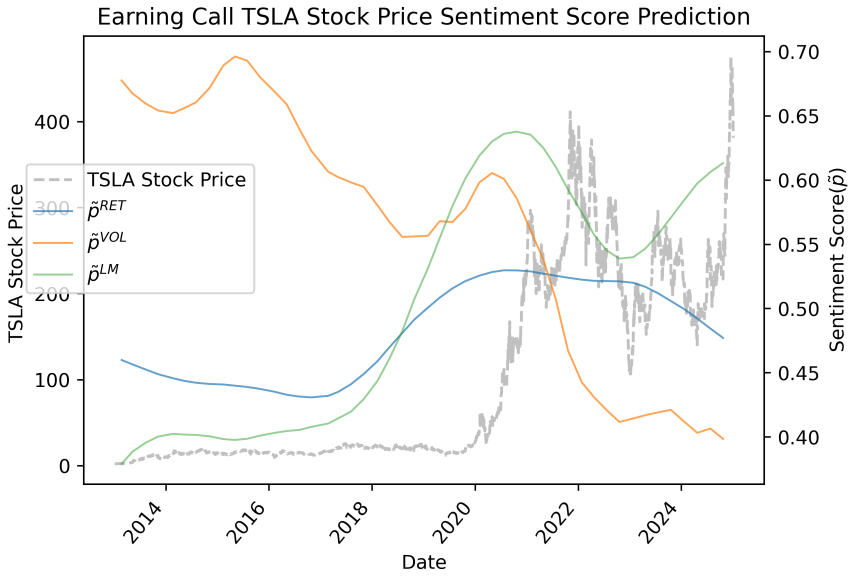

Figure H.6: Earnings Call sentiment score

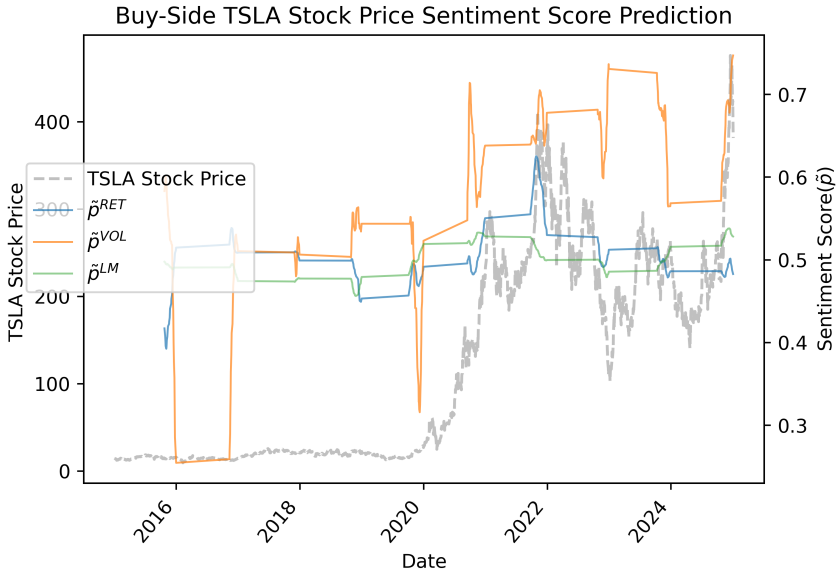

Figure H.7: Buy-Side sentiment score

H.4 Apple

Table H.2: Apple Tone Metrics

SEC Filing	$\tilde{p}^{\text{RET}}$	$\tilde{p}^{\text{VOL}}$	$\tilde{p}^{\text{LM}}$	APPL Stock
$\tilde{p}^{\text{RET}}$	1			
$\tilde{p}^{\text{VOL}}$	-0.348**	1		
$\tilde{p}^{\text{LM}}$	-0.802**	0.645**	1	
APPL Stock	-0.635**	**0.682****	0.874**	1
Transcripts	$\tilde{p}^{\text{RET}}$	$\tilde{p}^{\text{VOL}}$	$\tilde{p}^{\text{LM}}$	APPL Stock
$\tilde{p}^{\text{RET}}$	1			
$\tilde{p}^{\text{VOL}}$	**-0.740****	1		
$\tilde{p}^{\text{LM}}$	0.834**	-0.865**	1	
APPL Stock	0.663**	-0.272*	0.272*	1
Buy Side	$\tilde{p}^{\text{RET}}$	$\tilde{p}^{\text{VOL}}$	$\tilde{p}^{\text{LM}}$	APPL Stock
$\tilde{p}^{\text{RET}}$	1			
$\tilde{p}^{\text{VOL}}$	-0.272**	1		
$\tilde{p}^{\text{LM}}$	0.165**	-0.058	1	
APPL Stock	**0.453****	0.192**	-0.096**	1

$Note: p-value* < 0.05,\ p-value** < 0.005$

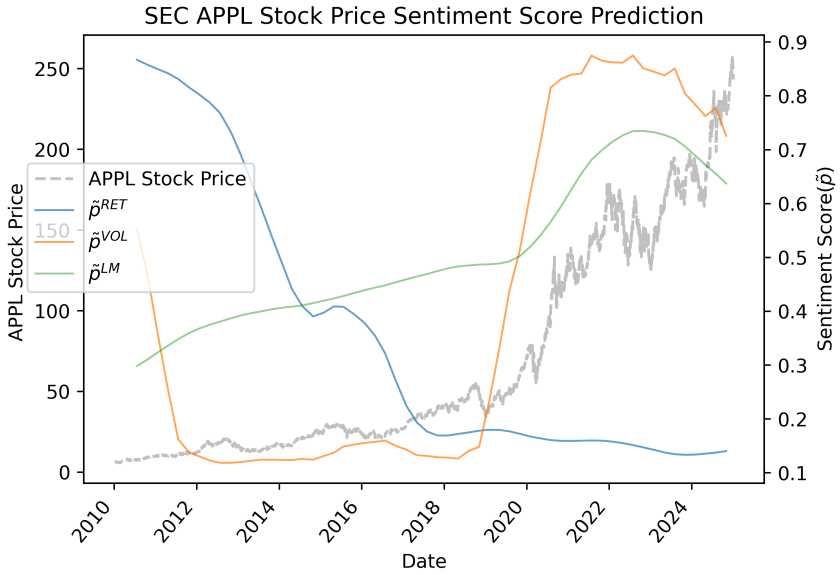

Figure H.8: SEC filing sentiment score

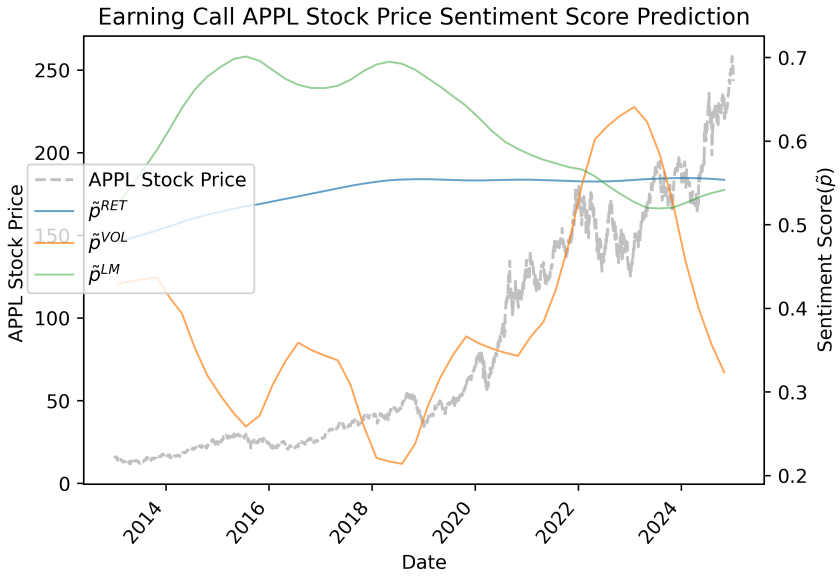

Figure H.9: Earnings Call sentiment score

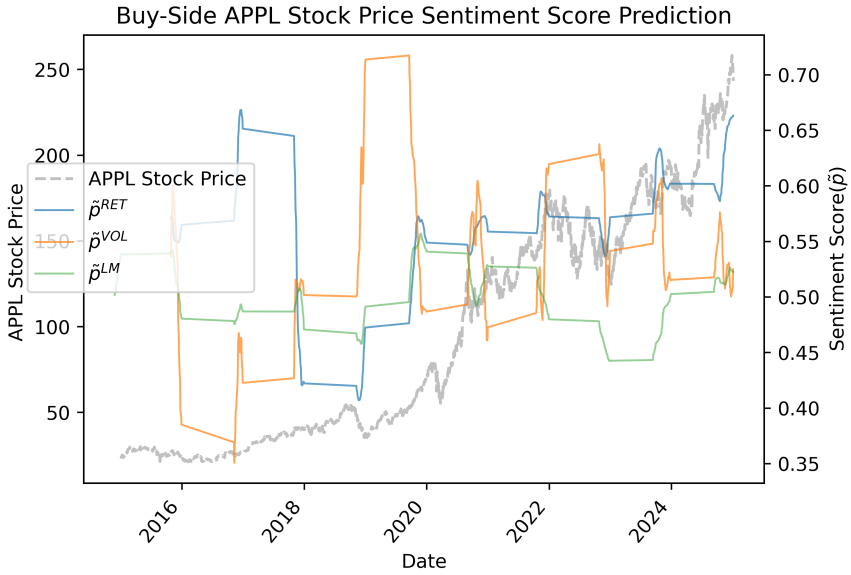

Figure H.10: Buy-Side sentiment score

Appendix I

Earning Call Frequency Heatmap

I.1 Nvidia

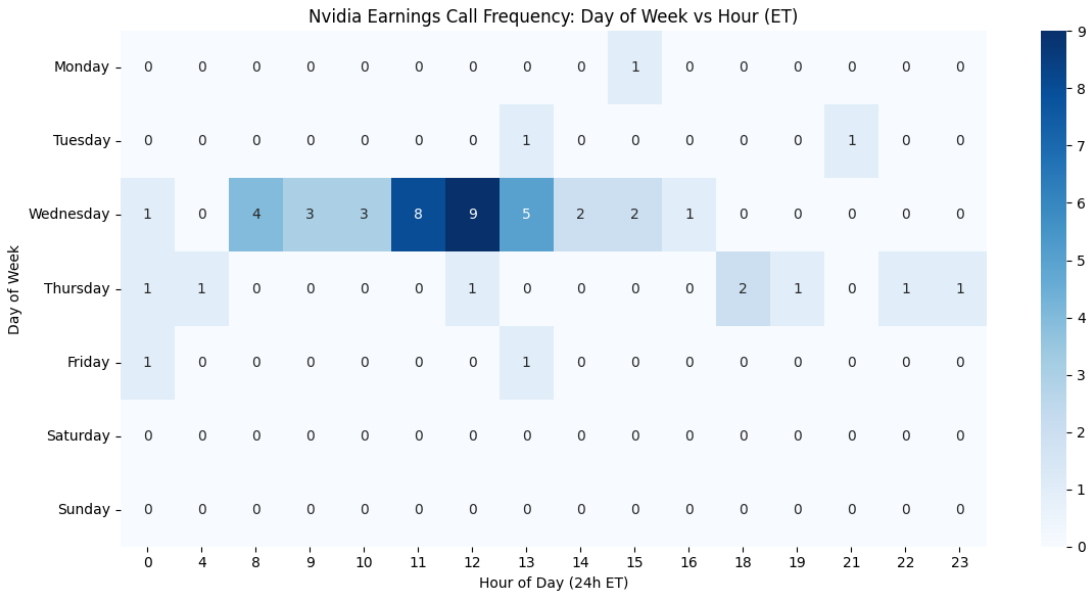

Figure I.1: Nvidia Heatmap

I.2 Tesla

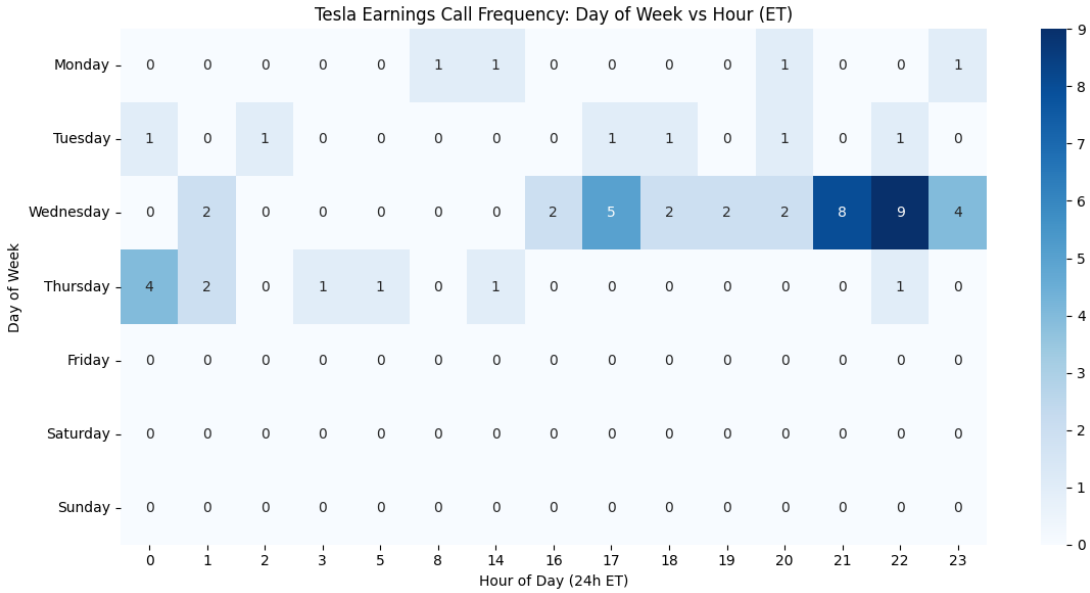

Figure I.2: Tesla Heatmap

I.3 Apple

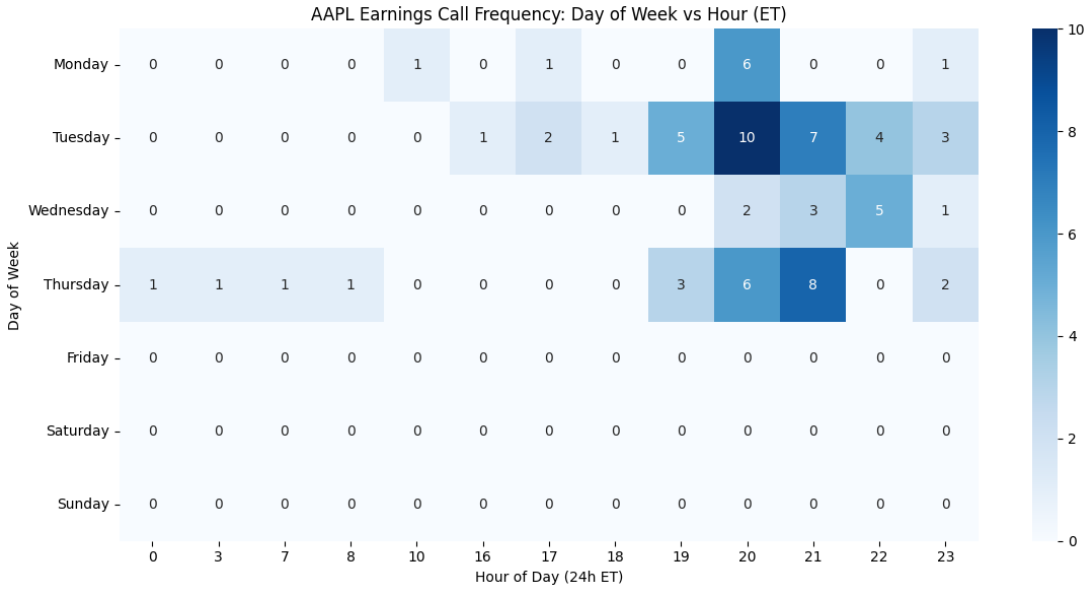

Figure I.3: Apple Heatmap