Earnings Conference Call Transcripts - Sentiment Analysis



Brian Higgins

School of Computer Science

University of Galway

*Supervisor*

Dr Josephine Griffith

In partial fulfilment of the requirements for the degree of MSc *in Software Engineering and Database Technologies*

August 2024

DRAFT 2

DECLARATION

I, Brian Higgins, do hereby declare that this thesis entitled ‘Earnings Conference Calls – Sentiment Analysis’ is a bonafide record of research work done by me for the award of MSc in Software Engineering and Database Technologies from the University of Galway. It has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Acknowledgements*

*I wish to express my eternal gratitude to my wife Teresa and daughter Catherine for their constant encouragement, patience and support.*

*I would like to thank my supervisor Dr Josephine Griffith for all her time, support and guidance in completing this thesis.*

Abstract

Quarterly and annual Earnings Conference Calls (ECC) are webcasts or conference calls held by publicly traded companies to present and discuss their latest financial results. Participants include company senior executives, institutional investors and financial analysts. Members of the general public may attend in listen mode. ECCs are significant events in the financial year of companies as they offer the first opportunity for companies to discuss the latest financial results with analysts and investors and to answer their questions directly. Transcripts of ECCs are made available on company websites following the calls. ECC are carefully studied by stock market analysts and investors in attempts to discover new information and which could assist in making investment decisions. Information disclosed in Earnings Conference Calls is not limited to technical and fundamental company data, much of which will have been disclosed in the earlier earnings press release. Information is also contained in the sentiment or linguist tone of the management and analysts. Studies have found that the sentiment of earnings calls may have an impact on stock returns with positive and negative correlations observed. This thesis examines if the sentiment expressed in the questions and answers section of ECC transcripts can be related to subsequent stock price changes. ECC transcripts will be web scraped from a single global financial data provider. The sentiment of these transcripts will be extracted using FinBERT, a deep learning language model specifically designed for financial text analysis. This will be implemented in Python version 3.9.8. Selenium WebDriver2 will be used to automatically control the browser. Extensive use of the Python library Pandas3 will be made to manipulate and analyse the data.

**Keywords:** earnings conference calls, sentiment analysis, FinBERT.

# Contents

[1 Introduction 9](#_Toc173595319)

[1.1 Research topic and motivation. 9](#_Toc173595320)

[1.2 Background 10](#_Toc173595321)

[1.3 Earnings Conference Calls 11](#_Toc173595322)

[1.4 FinBERT 13](#_Toc173595323)

[1.5 Thesis structure 16](#_Toc173595324)

[2 Literature Review 16](#_Toc173595325)

[3 Data 19](#_Toc173595326)

[3.1 Data description 19](#_Toc173595327)

[3.2 Available Transcripts 20](#_Toc173595328)

[3.3 Individual transcripts 23](#_Toc173595329)

[4 Verifying the accuracy of the model. 25](#_Toc173595330)

[5 Financial Data Extraction and Processing Pipeline (FDEP) 27](#_Toc173595331)

[5.1 Stage 1 – Raw data 28](#_Toc173595332)

[5.1.1 module 1.1 Web scraping the list of available transcripts 28](#_Toc173595333)

[5.1.2 module 1.2 Web scraping the S&P 500 transcripts 29](#_Toc173595334)

[5.1.3 module 1.3.a splitting the transcripts and extracting the Q&A. 31](#_Toc173595335)

[5.1.3 module 1.3.b extracting cleaned questions and answers separately for input to FinBERT 32](#_Toc173595336)

[5.2 Stage 2 input to FinBERT 33](#_Toc173595337)

[5.2.1 module 2.1 Input to FinBERT 33](#_Toc173595338)

[5.3 Stage 3 Results 34](#_Toc173595339)

[5.3.1 module 3.1 combine FinBERT output files 34](#_Toc173595340)

[5.3.2 module 3.2 update tone and stock price 35](#_Toc173595341)

[Stage 1 Web scraping the raw data and preparing it for input to the sntiment analysis model 36](#_Toc173595342)

[Results and Analysis 38](#_Toc173595343)

[Findings 38](#_Toc173595344)

[Analysis 44](#_Toc173595345)

# List of Figures

[Figure 3.1 JSON object holding transcript data 21](#_Toc173581104)

[Figure 7. 1 Financial Data Extraction and Processing Pipeline (FDEP) 27](#_Toc173591542)

[Figure 7. 2 Web scraping the list of available transcripts 30](#_Toc173591543)

[Figure 7. 3 Splitting the raw transcripts into MD&A and Q&A sections. 32](#_Toc173591544)

[Figure 8. 1 correlation matrix : questions only 41](#_Toc173319859)

[Figure 8. 2 scatter plot of tone vs price change 1day: questions only 42](#_Toc173319860)

[Figure 8. 3 scatter plot of tone vs price change 2day: questions only 42](#_Toc173319861)

[Figure 8. 4 scatter plot of tone vs price change 5day: questions only 43](#_Toc173319862)

[Figure 8. 5 correlation matrix : answers only 44](#_Toc173319863)

[Figure 8. 6 scatter plot of tone vs price change 1day: answers only 45](#_Toc173319864)

[Figure 8. 7 scatter plot of tone vs price change 2day: answers only 46](#_Toc173319865)

[Figure 8. 8 scatter plot of tone vs price change 5day: answers only 47](#_Toc173319866)

[Figure 8. 9 tone and price change correlation matrix. overall Q&A section 48](#_Toc173319867)

[Figure 8. 10 scatter plot of tone vs price change 1day: overall Q&A section 48](#_Toc173319868)

[Figure 8. 11 scatter plot of tone vs price change 2day: overall Q&A section 49](#_Toc173319869)

[Figure 8. 12 scatter plot of tone vs price change 5day: overall Q&A section 50](#_Toc173319870)

# List of tables

[Tbl 8. 1 Tone of transcripts sections. 38](#_Toc173319914)

[Tbl 8. 2 Correlation of Tone and price change 40](#_Toc173319915)

[Tbl 8. 3 Correlation of tone and price change: questions only 40](#_Toc173319916)

[Tbl 8. 4 tone and price change correlation. Answers only 44](#_Toc173319917)

[Tbl 8. 5 corelation tone and price change Q&A section 48](#_Toc173319918)

# Acronyms

CHAPTER 1

# 1 Introduction

## 1.1 Research topic and motivation.

Investors and stock market analysts’ study both quantitative and qualitative data when considering investment decisions. Quantitative data includes for example, earnings per share (EPS), profit and loss, share price. Qualitive or unstructured data includes annual and quarterly reports, news reports, social media posts and earnings conference call transcripts. The study of sentiment of earnings conference call transcripts and its relationship to stock market movements is an active area of research. Use of advanced NLP models to capture obscure market signals from long financial documents can offer useful insights to investors and analysts. Medya et al show that the semantic characteristics of earnings call transcripts can play a role in the prediction of stock price movements. (Medya *et al.*, 2022). A study by Price et al (Price *et al.*, 2012) found that the linguist tone of conference calls holds incremental information additional to earnings press releases and has use in the prediction of abnormal stock returns and trading volume. The extraction of sentiment of conference call transcripts can be carried out using various methods including lexicon or dictionary-based methods, machine learning methods and deep learning techniques. Advances in Natural Language Processing (NLP) include the introduction of transformer deep learning models with improved sentiment classification accuracy. This thesis will employ FinBERT, a pretrained transformer-based model, to extract the sentiment from earnings conference call transcripts and investigate if a correlation is evident with subsequent stock price movements.

The goal of this research is:

* Extract the sentiment of individual questions and answers from the Q&A section of earnings conference call transcripts using deep learning NLP model.
* Examine if a statistical relationship is evident between the sentiment and subsequent stock price movements within a period of one to five days.

1.2 Background Understanding the factors affecting stock market movements is of interest to investors and analysts. Methods currently employed for stock market prediction are unreliable. As a result, there is much research in this area. One of these research areas is the degree to which the stock market builds-in the information contained in the sentiment of earnings conference calls.

Two main theoretical hypotheses define market behaviour: the efficient market hypothesis (EMH) (Malkiel & Fama, 1970) and the adaptive market hypothesis (AMH) (Lo, 2005). The notion that markets are random and not predictable is firmly established in the random walk theory Bollen, Mao, and Zeng (2011). ..EMH …(need to describe the EMH…)…. *Behavioural Finance gives another view*… *Behavioural finance investigates stock market movements based on the phycology of* investors ….principles (Picasso, Merello, Ma, Oneto, & Cambria, 2019). T, . *The AMH is based on the belief that investors learn from their mstakes, and* .. people are motivated by self-interest….

*Stock prices movements are analysed mainly by two methods: Technical and fundamental. Technical analysis consists of modelling of historic stock price data in order to make predictions.*

*Fundamental analysis involves examining a range of factors both internal and external to a company that may affect its share price. Internal factors include the company’s financial ratios…*The fundamental aspects are economic data, financial performance, political and social behaviors, the business environment, and the firm’s financial ratios (Beyaz, Tekiner, Zeng, & Keane, 2018).We point out market capitalization (MC), earnings per share (EPS), the price/sales ratio (P/S), and the debt/equity ratio (D/E) as some of the notable financial ratios.

1*https://seekingalpha.com/*

2 https://www.selenium.dev/documentation/webdriver/

3 https://pandas.pydata.org/

4 https://huggingface.co/ProsusAI/finbert

## 1.3 Earnings Conference Calls

Earnings Conference Calls (ECC) are quarterly conference calls/webcasts hosted by publicly traded companies to present and discuss the latest quarterly or annual earnings reports and outlook with financial analysts and investors. Companies usually issue quarterly earnings press releases within four to six weeks following the end of the previous financial quarter. Press releases which disclose details of the company performance are followed by earnings conference calls which usually take place on the same day or the next day. Participants are typically the company senior management, normally the CEO and CFO, who present the latest financial reports and who will later take questions, financial analysts, institutional investors and financial journalists who may question the management in regard to the company performance and plans. The financial analysts will normally be specialists who cover the particular company or the industrial sector in which the company operates. Members of the general public can attend, normally in listen mode only. Earnings Conference Calls can be freely accessed via the companies’ websites, usually on the ‘Investor Relations’ page. They typically last between 30-60 minutes. Notification of the date and time is posted a few days in advance of the earnings press release. Transcripts and audio recordings are normally made available on company websites following the conference calls. An example can be found on the Microsoft Inc website (microsoft.com).

Reporting requirements: 10Q and 10k reports.

All companies listed on a US stock exchange are legally required to publish detailed financial information regarding the current performance of the company on a quarterly basis (Securities Act of 1933). At the end of each of the three financial quarters Earnings Reports must be filed with the U.S. Securities and Exchange Commission (SEC). The three quarterly Earnings Reports are known a 10Q reports and contain unaudited financial statements and review of the quarter’s operations. The fourth quarter report is the annual and more detailed Earnings report which is filed following the end of the fiscal year. The annual Earnings report is known as the 10K report and contains audited financial statements, annual review of markets, operations and company organisation and history, its products and services. 10Q Reports must be filed with the SEC within 45 days of the quarter end. 10K reports must be filed within the SEC 60 days of the end of the fiscal year.

They provide detailed financial information regarding the company performance for the most recent quarter including balance sheet, sales, cash flow, earnings per share and include a comparison with the same quarter the previous year.

Quarterly company reports to the SEC in the USA are mandatory, but earnings conference calls are not, however most of the larger listed companies conduct earnings conference calls.

ECC are normally conducted in two distinct sections. Section one, the management discussion and analysis section (MD&A), is a prepared and scripted presentation by management of the financial results, plans and outlook for the company. The second section is the Q&A section. In the QA section analysts question management on the latest quarter results and future prospects and plans. This section is unscripted as analysts present their own questions. Brockman et al found the Q&A section to be more informative than the MD&A section in prediction of future returns (Brockman, Li and Price, 2015).

Earnings Seasons Many companies in the US align their financial year with the calendar year. This leads to a tendency for earnings reports to be released in clusters within specific date ranges. These date ranges are called ‘Earnings Seasons’. Mid-April to Mid-May 1’st QTR, Mid July to Mid Aug -2’nd QTR, Mid Oct – Mid Nov – 3’rd Qtr, Mid Jan – Mid Feb -4’th Qtr. It is often the case that hundreds of companies could be releasing earnings reports and hosting Earnings Calls on the same day (forbes.com, 2023).

Earnings Call Schedule. Companies announce in advance the exact date when earnings will be released. At the same time the date and time of the related earnings call/webcast is announced with details of how to attend, usually via the company website.

Earnings Press releases are issued a few weeks after quarter or year-end in advance of the SEC filings. The Earnings Press release contains a summary of the financial performance for the quarter or year and includes ley metrics such as Earnings Per Share (EPS), Revenue, Net Income and Sales.

Earnings Conference Calls are usually held on the same day or next day following the press releases. There is no legal requirement to hold earnings calls conferences, but most companies hold them in the interests of transparency and investors relations. In the period considered by this thesis there are records of conference calls by 499 companies that make up the S&P 500 index. The exception being Berkshire Hathaway Inc (BRK), , which does not hold earnings conference calls. At the time of the earnings press releases, a few weeks into the next quarter, campnaies have some idea of how the current quarter is progressing and often will incorporate projections for the coming quarter known as forward guidance.

## 1.4 Sentiment Analysis

Determining the sentiment of ECC is usefull to investors and analysts as it can give an indication of the puture stock price direction.

Sentiment or tone can be extracted by carrying out sentiment analysis of the transcripts. Sentiment analysis involves classifying the text into predefined categories based on the emotions or opinions expressed in it.

There are numerous techniques available to carry out sentiment analysis.

Traditional techniques use rule based lexicon approaches which rely on predefined lists of words classified as positive neagtive or neutral. Examples are SentiWordNet and VADER

Machine learning models such as Naïve Bayes…………

Later techniques employ word embedding models that make use of neural networks to learn vector representations of words in a continuous vector space, and can capture semantic (how words are used and their intended meaning) relationships. Examples include Word2Vec, **GloVe (Global Vectors for Word Representation) and FastText:.** They can be used used to create word embeddings that can be fed into a language model for sentiment analysis.  **Recurrent Neural Networks (RNN) A type of RNN that can capture long-term dependencies in sequential data, effective for handling context in sentences.**

**o GRU (Gated Recurrent Units): A variant of LSTM, simpler in architecture, often faster to train while providing competitive performance.**

3. Convolutional Neural Networks (CNNs):

o Used primarily for image processing, CNNs have been adapted for text by applying convolutional filters over word embeddings to capture local features, like key phrases, for sentiment analysis.

Transformer based language models.

The transformer archicture enables the handling of long-range dependencies in NLP more effectively than previous RNN and LSTM models. The architecture relies entirely on a mechanism called self-attention and eliminates the need for recurrent layers,

A mechanism called ‘self-attention’ eliminates the need for recurrent layers and leads to faster and more parallelizable training. The concept was introduced by Vaswani et al in a 2017 conference paper ‘Attention is All You Need’ (Vaswani *et al.*, 2023).

Pre-trained language models such as BERT, GPT and LSTM models can used for this purpose. Thay have the ability to generate numerical representations of words that represent their meaning in context. In this way language models can be used to extract syntactic and semantic features necessary for understanding sentiment.

## 1.5 Language models.

In Natural Language Processing (NLP) language models are mathematical formulations that quantify the likelihood of sequences of words in order to understand, interpret and generate natural human language. The goal of language models is to enable computers to understand human language as it is spoken naturally. Natural language is constantly evolving and can be ambigious. This creates cahllenges for computers in understanding human language. Areas of difficulty include sarcasm, humour, inflection and tone.

Language models use probabilities and various statistical measures to represent language patterns and predict the likelihood of a sequence of words, generate text, or understand the context of words in a given text. language models in NLP are frameworks that use statistical and computational methods to understand and predict language patterns.

They are implemented using algorithms and computational techniques that process large amounts of text data to learn these patterns. They are trained on large datasets or corpora of text to learn the structure and distribution of language. They are trained on large datasets of text and use statistical and neural network techniques to predict the next word in a sentence, translate languages, answer questions, and perform other language-related tasks.

This learning process creates a model of the language based on observed data. Once trained, the model can generalize to new, unseen text, making predictions or generating coherent text based on its learned understanding. They represent a structured, learned abstraction of language, similar to how models in other scientific fields represent real-world phenomena.

**Key Concepts in Language Models**

1. **Training Data**: Large corpora of text used to teach the model about language patterns, grammar, and facts.
2. **Architecture**: The structure of the neural network, such as Transformer models, which are widely used today.
3. **Pretraining and Fine-tuning**: Pretraining involves learning general language patterns, while fine-tuning adapts the model to specific tasks.

## 1.6 BERT

BERT (Bidirectional Encoder Representations from Transformers) is one of a class of transformer based large language models or LLMs. It is an open-source architecture for natural language processing (NLP) introduced by Google in 2018. It is designed to help computers understand naturally spoken human language. In order to understand the meaning of words BERT uses surrounding text to determine the context and deal with ambiguity. As a bidirectional model, BERT can consider both preceding and succeeding words in a sentence, providing a more comprehensive understanding of context. This bidirectional nature is particularly useful for tasks that require understanding the relationship between words and phrases across the entire sentence.

BERT's ability to consider both preceding and following context, coupled with its bidirectional nature and self-attention mechanism, allows it to generate rich and contextually aware embeddings for tokens. This capability makes BERT particularly effective in understanding nuanced meanings and relationships in natural language.

BERT splits or tokenizes the input text into smaller, more manageable pieces, which can be whole words or sub words (parts of words) in a specific process called WordPiece tokenization. Each token (word and sub-word) is mapped to a unique high-dimensional (768 dimensions) numerical vector representation (embedding). BERT can accept a maximum token length of 512.

Embeddings are numerical representations of words and sub-words that computers can process. These embeddings capture semantic and syntactic properties of the tokens and provide BERT with the ability to understand and process the input text effectively. The initial token embeddings are augmented with positional embeddings which capture the token position in a sentence or sequence, and segment embeddings which distinguish between sentences. The token embeddings are processed by transformer encoder layers which apply a process called self-attention which takes into account the context in which a token appears. The result is a final embedding which is dynamically context aware or context dependent. meaning it considers the surrounding tokens in the sequence. The same word can have different representations based on its context within a sentence, allowing for more accurate and nuanced understanding.

**Before BERT**: Word2Vec, GloVe, and FastText were popular methods for representing words as vectors in a continuous vector space. However, these models produced static word embeddings, meaning a word had the same representation regardless of its context.

It employs two key innovations in language modelling. First the transformer architecture (T) is used for modelling long term dependencies and second it uses Masked Language Modelling (MLM) in which a random portion of the tokens or words are masked and the model predicts them, leading to bi-directionality (B). There are two versions of BERT, Bert Large which uses an embedding dimension of 1024, and BERT base which uses 768 dimensions. This enables BERT to understand the context and nuances of language.

**Seq2Seq Models with Attention**

**Before BERT**: Seq2Seq models with attention mechanisms were widely used for tasks like machine translation and text summarization. While effective, these models typically required significant amounts of data and compute resources for training.

**With BERT**: BERT's pretraining on large corpora allows for effective transfer learning, where it can be fine-tuned on specific tasks with relatively smaller amounts of labeled data. This improves performance across a wide range of tasks, including those traditionally handled by Seq2Seq models.

Transformers.

BERT makes use of a transformer architecture.

Transformers are a type of neural network designed to process sequential data.

The transformer is the part of the model that gives BERT its increased capacity for understanding context and ambiguity in language. The transformer processes any given word in relation to all other words in a sentence, rather than processing them one at a time. By looking at all surrounding words, the transformer enables BERT to understand the full context of the word and therefore better understand searcher intent.

**Masked language modeling**

Word embedding models require large data sets of [structured data](https://www.techtarget.com/whatis/definition/structured-data). While they are adept at many general NLP tasks, they fail at the context-heavy, predictive nature of question answering because all words are in some sense fixed to a vector or meaning.

BERT uses an MLM method to keep the word in focus from seeing itself, or having a fixed meaning independent of its context. BERT is forced to identify the masked word based on context alone. In BERT, words are defined by their surroundings, not by a prefixed identity.

**Self-attention mechanisms**

BERT also relies on a self-attention mechanism that captures and understands relationships among words in a sentence. The bidirectional transformers at the center of BERT's design make this possible. This is significant because often, a word may change meaning as a sentence develops. Each word added augments the overall meaning of the word the NLP algorithm is focusing on. The more words that are present in each sentence or phrase, the more ambiguous the word in focus becomes. BERT accounts for the augmented meaning by reading bidirectionally, accounting for the effect of all other words in a sentence on the focus word and eliminating the left-to-right momentum that biases words towards a certain meaning as a sentence progresses.

This is contrasted against the traditional method of language processing, known as word embedding. This approach was used in models such as GloVe and word2vec. It would map every single word to a [vector](https://www.techtarget.com/whatis/definition/vector), which represented only one dimension of that word's meaning

“The animal did’nt cross the street because it was too wide”

For example, in the image above, BERT is determining which prior word in the sentence the word "it" refers to, and then using the self-attention mechanism to weigh the options. The word with the highest calculated score is deemed the correct association. In this example, "it" refers to "animal", not "street". If this phrase was a search query, the results would reflect this subtler, more precise understanding BERT reached. BERT examines individual words in context to determine the meaning of ambiguous language.

Next sentence prediction

NSP is a training technique that teaches BERT to predict whether a certain sentence follows a previous sentence to test its knowledge of relationships between sentences. Specifically, BERT is given both sentence pairs that are correctly paired and pairs that are wrongly paired so it gets better at understanding the difference. Over time, BERT gets better at predicting next sentences accurately. Typically, both NSP and MLM techniques are used simultaneously.

## 1.7 FinBERT

FinBERT is a pre-trained open-source Natural Language Processing model (NLP) designed specifically for financial sentiment analysis. The language used in the financial domain has its own characteristics particular to that domain and contains specialized financial language not usually found is normal everyday language. General purpose sentiment analysis models do not perform well when attempting to classsify financial text. FinBERT is based on BERT. It is constructed by further training BERT on the unlabelled 1.1 million word Reuters news corpus. This is followed by fine tuning on a human annotated 4,485 sentence labelled financial text dataset.

The FinBERT model used in this research is produced by the technology company ProsusAi.

It is available on the Hugging Face model hub55. The model was developed by Araci (Araci, 2019) to address the problem of financial sentiment analysis. Araci used the pretrained BERT language model and provided it with further training on a large financial corpus

. The existing BERT model is given futher training on a large financial text corpus, the Reuters TRC2. The resulting model is fine-tuned on a labelled specific financial text dataset, the Financial PhraseBank. The Financial PhraseBank is a dataset consisting of 4845 financial industry phrases which have been hand labelled as positive, negative or neutral by human experts Malo et al (Malo *et al.*, 2014). This is a state if the art method to produce models for language understanding in specialised or specific domains.

The model was loaded on Kaggle to avail of the GPU accellerators available there.

55https://huggingface.co/ProsusAI/finbert?text=Food+companies+doing+bad+due+to+the+global+markets+downturn+due+to+covid

## 1.5 Thesis structure

This thesis consists of six chatpers.

The introduction and background is given in chapter 1. It presents the motivation to analyse ECC followed by a description of ECC and the transcripts. This followed by summary of sentiment analysis techniques and an outline of language models. A description FinBERT, the particular language model used along with a summary of the BERT model on which it is based.

Chapter 2 reviews the literature relating to the study of ECC. Chapter 3 provides a description of the data and how it is obtained and pre-processed. Chapter 4 describes the process of verification of the FinBERT model being used. Chapter 5 describes in detail the processing steps or the Financial Data Extraction and Processing pipeline. Chapter 6 discusses the results. Chapter 7 presents the analysis.

Chapter 2

# 2 Literature Review

Price et al (Price *et al.*, 2012) found that the tone of language used in earnings calls can be correlated to the subsequent abnormal stock returns by comparing the actual returns to the expected returns. They examined earnings call transcripts and quantified the textual tone of the MD&A section. The relationship between the tone and subsequent stock returns was investigated. Tone was quantified using the Loughran-McDonald financial sentiment dictionary and the Harvard IV-4 psychosocial dictionary. A positive tone was found to be correlated to positive abnormal stock returns. A negative tone was found to be correlated to negative abnormal stock returns. The conference paper ‘Forecasting Earnings Surprises from Conference Call Transcripts’ by Koval et al describes the development of a deep learning model to predict earnings surprises based on the sentiment of earnings conferenc call transcripts. Earnings surprises are a measure of the difference between the actual Earnings Per Share (EPS) and the analysts’ consensus estimates. The sentiment of Earnings Call transcripts was extracted using a number of different traditional models and hierarchical transformer models. They link each transcript to the subsequent earnings report to determine if an earnings surprise occurred. The results show that it is possible to predict earnings surprises with what they call ‘reasonable accuracy’ using only the text of earnings conference call transcripts. The authors acknowledge certain limitations of the work. Nagendra et al (Bv *et al.*, 2023) in their conference paper ‘Deploying NLP Techniques for Earnings Call Transcripts for Financial Analysis: A Reverse Phenomenon Paradigm’ observe what they term the ’Inverse Effect’ where stock price movements are in the opposite direction to the sentiment of the corresponding earnings conference call transcripts. The results show that there are significant instances of positive sentiment followed by negative stock price movement and vice versa. They term this as the ‘Inverse Effect’. The sentiment of earnings conference call transcripts was extractwed using the generic VADER (Valenc Aware Dictionary and Sentiment Reader) model which is a lexicon rule-based sentiment analysis model which generates word sentiment scores for the categories positive, neagative, neutral and compound. VADER is an open-sourced package within the Natural Language Toolkit (NLTK). One of the limitations of lexicon methods is their inability to capture context particularly in specialised domains such the financial text area. The study uses a small dataset of earnings confereence call transcripts from three large software covering a period of ten years.

A review of the performance of sentiment analysis models when applied to the financial domain was carried out by Mishev et al (Mishev *et al.*, 2020). They found the best performing lexical models achieved an accuracy of 0.652. This is in contrast to the accuracy of the FinBERT transformer based model which they found to be 0.890. Overall transformer based modes perform best in financial sentiment classification. The best performing transformer model was found to be BART-Large with an accuracy of 0.947 which the authors note is comparable that of a human expert.

Yamamoto et al (Yamamoto *et al.*, 2022) examined the tone of the management section of ECCs. They found that the tone of the management during ECCs provides a pointer to the future performance of the company and can be used to enhance investment strategies. The authors find that incorporation of the sentiment analysis into the area of quantitative financial analysis has the potential to improve investiment strategies. The language used in earnings press releases were examined by Davis et al (Davis, Piger and Sedor, 2012) to determine if it could be used to predict future performance. A text analysis software package called ‘Diction’ was used to count optimistic and pessimistic words. They found that measures of net optimistic language used in earnings press releases could be used to predict future performane.

.( In sharp contrast to the optimism revealed by MANAGER TONE, ANALYST TONE is either negative (mean = –0.01) or neutral (median = 0.00)( Differences in Conference Call Tones: Managers vs. Analysts Paul Brockman Pg 9))

Chapter 3

# 3 Data

## 3.1 Data description

The data consists of earnings conference call transcripts and historical stock price data. These are transcripts of quarterly earnings conference calls hosted by companie with analysts and investors. Transcripts were web scraped from the Seeking Alpha website2.Historical stock price data for each of the 500 companies that make up the S&P 500 was obtained from YahooFinance.com3 .

A total of 5126 transcripts covering the period November 2021 to February 2024 were obtained. This covers financial quarters 2021\_03 to 2023\_04. Of these 3956 are earnings conference call transcripts. The remaining 1,170 are transcripts from other events. For example, transcripts of company presentations at financial conferences. These were discarded from the analysis.

Seeking Alpha is a privately owned crowd-sourced financial data content provider. It publishes news, analysis and research on financial markets. Published Earnings Conference Call Transcripts are gathered globally by Seeking Alpha from company websites and made available on their website. Access is either by free access with viewing limited of one transcript per day or via a number paid plans with varying degrees of access. For this research the data was web scraped directly from the website.

This research focusses on S&P 500 companies. The reason for this was twofold. The S&P 500 index is considered to be the numerical indicator or barometer of the U.S. economy. It is made up of 500 of the largest companies listed on the NYSE, Nasdaq, or CBOE. (Investopedia, 2023) and covers the.. Secondly it was necessary to restrict the amount of data in anticipation of lengthy processing time for long documents.

*2 https://seekingalpha.com/*

*3 https://finance.yahoo.com/quote/AAPL/*

## 3.2 Available Transcripts

In order to get a list of taranscripts that are available on the Seeking Alpha website an API (Application Programming Interface) endpoint can be searched. Details of the set of transcripts available on the Seeking Alpha website can be found at this endpoint4. API endpoints are the final locations from where information is sent and received by the server. A website can have one or multiple endpoints. This particular endpoint is the Seeking Alpha URL holding lists of transcript details.

The data is presented as a list of 50 JSON objects per page (JavaScript Object Notation). Pages 1 to 1000 are available.

*4 https://seekingalpha.com/api/v3/articles?filter[category]=earnings%3A%3Aearnings-call-transcripts&filter[since]=0&filter[until]=0&include=author%2CprimaryTickers%2CsecondaryTickers&isMounting=true&page[size]=50&page[number]=1*

Each of the 50 JSON objects per page contains detailed information regarding an individual transcript. JSON objects are data structures which store information in key-value pairs. The JSON data is structured to provide a comprehensive set of details about each transcript, including metadata (such as publish date and title), related entities (such as author and tickers), and links to the full content.

Attributes include a unique seven-digit transcript id, company name, ticker symbol, date, type of transcript and fiscal period and further metadata relevant to each transcript. The JSON data structure simplifies access, analysis and extraction of various pieces of the data.

Fig xxx is an example of a JSON object from the Seeking Alpha API endpoint URL. It relates to transcript id:4619723. This is the Texas Instruments Incorporated (TXN) Q2 2023 Earnings Call Transcript published on 2023-07-25 at 18:47:06 local time with an offset from GMT of minus 4 Hrs.

{"id":"4619723","type":"transcript","attributes":{"publishOn":"2023-07-25T18:47:06-04:00","isLockedPro":false,"commentCount":0,"gettyImageUrl":null,"videoPreviewUrl":null,"videoDuration":null,"themes":{},"title":"Texas Instruments Incorporated (TXN) Q2 2023 Earnings Call Transcript","isPaywalled":false},"relationships":{"author":{"data":{"id":"44211","type":"author"}},"sentiments":{"data":[]},"primaryTickers":{"data":[{"id":"1051","type":"tag"}]},"secondaryTickers":{"data":[]},"otherTags":{"data":[{"id":"96991","type":"tag"},{"id":"49","type":"tag"},{"id":"586376","type":"tag"},{"id":"326","type":"tag"}]}},"links":{"self":"/article/4619723-texas-instruments-incorporated-txn-q2-2023-earnings-call-transcript"}}.

Figure 3.1 JSON object holding transcript data

The JSON object shows attributes of the object and their values. In the example, the unique transcript ID can be identified: key “id” and its value *"4619723".*

In summary the JSON object above contains the following information:

Transcript ID: 4619723

Type: transcript

Publish Date: 2023-07-25T18:47:06-04:00

Is Locked Pro: False

Comment Count: 0

Title: Texas Instruments Incorporated (TXN) Q2 2023 Earnings Call Transcript

Is Paywalled: False i.e. Not behind a paywall – content is accessable without payment.

Author ID: 44211

Primary Tickers: ['1051'] - ID used in the ‘primary’ stock market area.

Other Tags: ['96991', '49', '586376', '326']

Self Link: /article/4619723-texas-instruments-incorporated-txn-q2-2023-earnings-call-transcript

(link to the transcript.)

The elements of interest for this thesis are*:*

*{*

*"id": "4619723",*

*"type": "transcript",*

*"attributes": {*

*"publishOn": "2023-07-25T18:47:06-04:00",*

*.*

*.*

*.*

*"title": "Texas Instruments Incorporated (TXN) Q2 2023 Earnings Call Transcript",*

*}'''*

The JSON object is in the form of a string which can be read with the Python’s JSON module. The information in the object can be output as a Python dictionary which is a data structure storing the key-value pair data. Once this is available the data can be manipulated using ‘Pandas’, a Python data analysis and manipulation library.

Module\_1 of the data collection and analysis pipeline webscrapes the Seeking Alpha API endpoint and creates a list of available transcripts with S&P 500 company transcripts highlighted. Details of the pipeline are presented in Chapter 7.

## 3.3 Individual transcripts

Individual complete transcripts relating to each earnings conference call can be found at another Seeking Alpha API endpoint6 . Individual transcripts can be found in JSON objects at this URL. A web scraper was developed to navigate to this URL. The transcripts list< from module 1.1 was opened and ID values flagged as S&P 5000 were iterated through and placed in the ID value location in the URL

The transcript presents as a JSON object including the tarnscript text and a range of meta data in HTML format.

An example is shown here with sections of the Texas Instruments Inc earnings conference call transcript JSON object located between the brackets{}. The portions of text highlighted in yellow are those the used locate the relevant sections for extraction later in the process.



The transcripts were in HTML format when web scraped. They were automatically written to a series of csv files by the web-scraper..

Earnings call transcripts are long documents with word counts ranging between 6000 and 12000 words. They were written to the csv files in chunks of 8,000 characters to a row. This method resulted in each transcript occupying a number of rows of the csv file. Typically, a transcript of 11,000 words would have approx 80k characters including spaces. Such a transcript in its raw scraped html state occupies approximately ten rows of the csv file when HTML Tags are included.

5 <https://stockanalysis.com/list/sp-500-stocks/>

6 https://seekingalpha.com/api/v3/articles/4635802?include=author%2CprimaryTickers%2CsecondaryTickers%2CotherTags%2Cpresentations%2Cpresentations.slides%2Cauthor.authorResearch%2Cauthor.userBioTags%2Cco\_authors%2CpromotedService%2Csentiments

The length of the transcripts made them unsuitable to place a complete transcript on a single csv file row. They were split into 8000 charatcer long chunks with each chunk being placed on a single row. On average each transcript occupied eight csv file rows. This series of csv files contained the raw transcripts in HTML format.

An example of some rows of one of the csv files is shown below:



Two distinct sections were next extracted from each transcript. The MD&A (Management Discussion and Analysis) section and the Q&A section. Of interest in this research is the Q&A section. This section was extracted in HTML format for later splitting into individual questions and answers. The MD&A was extracted in plain text but was not used.

The individual questions and answers were then extracted from each transcript.

The two sections were written to the csv file in row pairs with the ‘Q&A’ written over a number of rows and identified as ‘Q&A Session’ in the column ‘Call\_Section’. The MD&A was written to the rows immediately below in the same manner and identified as ‘company\_statement’. The one ID in the ‘ID’ column identified all the rows which formed part of the same transcript. Kimbrough(Kimbrough, 2005) in his analysis of earnings calls contends that The MD&A is a prepared and scripted reiteration of the earlier Earnings Press Release. He splits the analysis of Earnings Calls into two sections, the MD&A and the Q&A. Similarly Price (Price *et al.*, 2012) in their analysis split transcripts into what they call INTRO and Q&A sections.

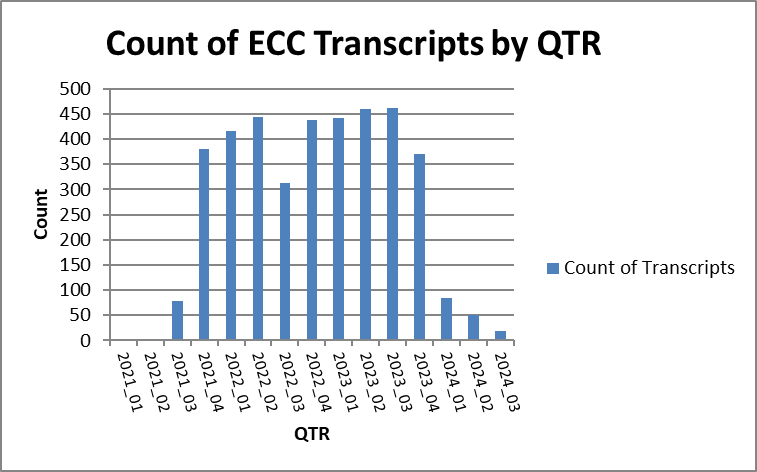
A sample is shown below:



A number of the transcripts were found to relate to events other than quarterly earnings calls and were excluded from the analysis. For example, transcripts of company presentations at financial conferences. The excluded number amounted to 1552. This left 4256 transcripts for analysis.

## 3.4 Profile of the data.

Spread of the data in time. Fig XXX



Spread of the data accross the GICS sectors. Fig xxx

The Global Industry Classification Standard (GICS) is an industry classification scheme for assigning public companies to the specific economic sector and industry group that best defines its business operations. It is made up of 11 sectors, 25 industry groups, 74 industries and 163 sub-industries and is the basis of the S&P 500 index.

https://www.investopedia.com/terms/g/gics.asp)

Spread of the data ccross the S&P 500.

Input of the transcripts to FinBERT.

The transcripts wer input to the model one file at a time. This was done to ensure the Kaggle server did not time out while processing. The steps in the model are as follows.

To be completed

Chapter 4

# 4 Verifying the accuracy of the model.

Before using the FinBERT model it was decided to check its operation and accuracy. The operation and accuracy of the FinBERT model was checked using the Financial Phrase Bank as input and comparing the model predictions of sentiment to the human annotated values on which the model had been fine tuned. The FinBERT model was fine-tuned with the Financial Phrasebank. Using this data as input the model should exhibit an accuracy of between 80 and 90%. Tbl xxx shows the test results. The accuracy of xxx indicates that the model is performing well allowing for model limitations, generalizations, human annotation errors and data variability.

xx https://huggingface.co/ProsusAI/finbert

Operation of the model.

ECC transcripts are long documents and cannot be input to FinBERT as a complete unit. The maximum token length that can be input to a BERT model is 512 tokens. This includes 510 text tokens plus start and end tokens inserted by the model. In order to input long documents to FinBERT two approaches are often used. Summarise the text or split the text into 512 length tokens using a sliding window method. The second method is used here.

Chapter 5

# 5 Financial Data Extraction and Processing Pipeline

The processing of the data can be described in three main stages shown in Fig 7.1

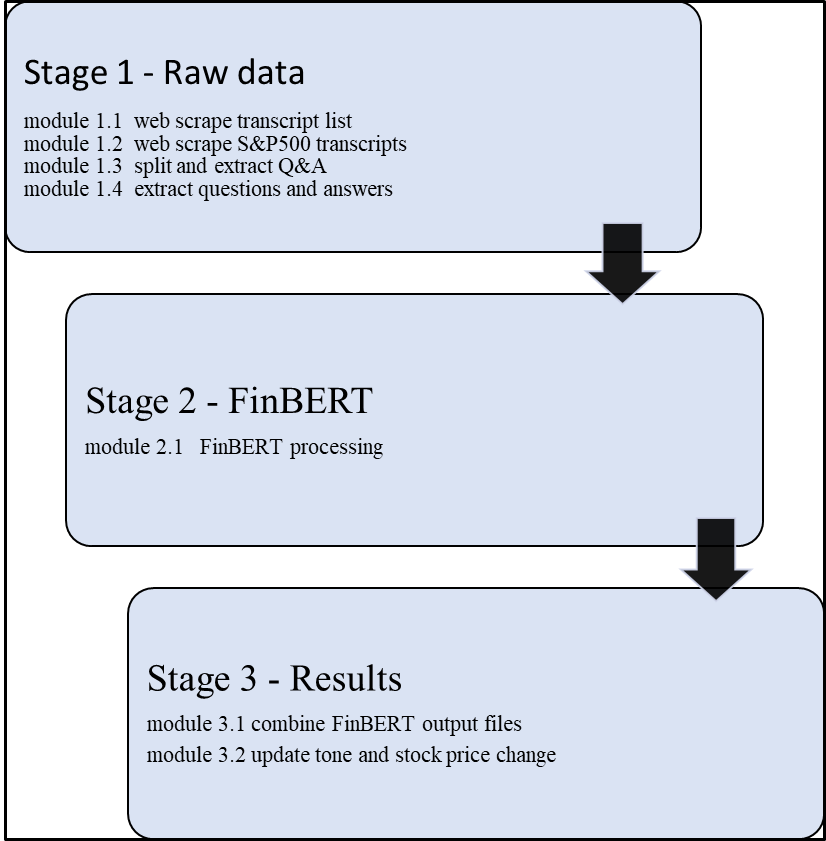


Figure 7. 1 Financial Data Extraction and Processing Pipeline

Web scraping and processing of data was integrated into the Financial Data Extraction and Processing Pipeline. Fig 7.1.

The purpose of the pipeline is as follows.

1. Web scrape a list ECC transcripts from the Seeking Alpha wesite.

2.Extract the Question and Answers section in each transcript and split it into separate questions and answers which are then written to a series of csv files.

3.Input this series of csv filesto the FinBERT model in order to to classify the sentiment of each question and answer.

4.The output files are then recombined, aggregated and grouped into one file. This combined file contains the sum for each transcript of the sentiment classifications in respect of all of the questions and answers.

5. A number of fields were then added to this file, Tone, average 1-day, 2-day and 5-day stock price change cnetered on the ECC date. The resulting combined and updated file contained the results of the processing.

6.The results were analysed to find if a correlation is evident between the tone of the questions and answers and the stock price changes.

Three stages, Stage 1, Stage 2 and Stage 3, together make up the pipeline. Each stage contains a number of modules designed to carry out the specific processing tasks referred to above.

Stage\_1 is concerned with web scraping the raw transcript data and preparing it for FinBERT input. It consists of modules 1.1 to 1.3.

Stage\_2 is the FinBERT processing stage. It consists of one module, module 2.1

Stage\_3 is the FinBERT output processing and results stage. It consists of modules 3.1 and 3.2. A detailed description of specific modules in each stage is now given.

## 5.1 Stage 1 – Raw data

### 5.1.1 module 1.1 Web scraping the list of available transcripts



Figure 7. 2 Web scraping the list of available transcripts

This module web scrapes data relating to all transcripts available on the Seeking Alpha website and creates a list with S&P 500 transcripts flagged.

Transcripts which are available on the Seeking Alpha website can be found at a Seeking Alpha API endpoint. This is the URL where a list of JSON objects containing transcripts data is located. The endpoint is paged from 1 to 1000 with nominally 50 JSON objects per page. A web scraping tool was developed to access the endpoint and iterate through pages 1 to 1000 extracting relevant transcript data from each JSON object in turn. The data extracted consisted of the transcript ID, Stock market ticker symbol, financial quarter, description of transcript and the date of the conference call. This data was written automatically to file ‘transcripts\_list\_data.csv’. Selenium WebDriver was used automate the browser and navigate to the API endpoint. The browser was configured to iterate through 150 pages per run. A random time delay of between 4 and 9 seconds was applied to ensure the page loaded and to prevent detection as an automated tool. A further random time delay of between 4 and 9 seconds was applied between page pulls to further reduce the chance of blocking by the website. BeautifulSoup and JSON were used to parse the HTML content and extract the data.

Identifying transcripts of S&P 500 companies. This thesis focusses on S&P 500 companies. These are the only companies of interest here. The JSON data does not indicate if a company is a member of the S&P 500 or not. In order to flag the transcript IDs of these companies a list of S&P 500 ticker symbols was obtained from a stock data website stockanalysis.com5 and written to file ‘S&P 500 Index Stocks List.csv’.

This file was used to update a new field ‘S&P500\_Company’ created in the file ‘transcripts\_list\_data.csv’ above. The field was updated to ‘yes’ in the case of the ticker of an ‘S&P500\_Company’ otherwise ‘no’. Following completion of the above steps a file containing a list of approximately 50k transcripts details and with the S&P 500 transcripts marked was available. At the conclusion of this module,. module 1.1, a list of transcript IDs and associated data was available. In addition the S&P 500 transcripts on the list were flagged.

### 5.1.2 module 1.2 Web scraping the S&P 500 transcripts

 In order to locate and save the individual transcripts a second web scraper was designed. This web scraper accessed a different Seeking Alpha API endpoint6. This endpoint is the URL of each individual transcript. Each page on this endpoint holds a single transcript with its associated meta data. The URL for each individual transcript contains the unique transcript ID which can be used to inentify a particular transcript. This provides a means to access any transcript by inserting the relevant ID into the URL. In this way and by using the transcript list from module\_1 it was possible to locate each S&P 500 transcript by its unique ID and web scrape it to a csv file. The web scraper accessed the URL of each transcript in turn by iterating over each ID which had been labelled as ‘S&P500\_Company’ in the transcripts list and placing it at the appropriate position in the URL. This ensured that only S&P500 company transcripts were fetched. The transcripts were written to file ‘transcriptscraped\_test\_data.csv’. The full set of transcripts was written to a series of these csv files ‘transcriptscraped\_test\_data.csv’ part 1 to part 10.

Transcripts are long documents, typically between 8000 and 12000 words. Attempting to place a complete transcript on one csv file row is not feasable. To overcome this the web scraping tool was designed to write the transcripts to the csv files in chunks of 8000 characters per row. In this way each transcript chunk fitted comfortably on a single row. This method resulted in each entire transcript occupying a number of rows of the csv file. Each transcript was identified by its unique ID which was placed in the ID column as well as the ticker, date, quarter and title in adjacent columns on each row the transcript occupied. The transcripts were in HTM format. Selenium was used to automate the browser and navigate to the relevant URLs. The process was carried out in batches of 150 with Selenium configured to close the WebDriver and disconnect from the website before reopening it after a random period of between seven and nine minutes between batches to avoid being blocked. This web scraping process was set to run automatically.

After completing module\_1.2 a series of csv files containing the complete transcripts of S&P 500 companies covering the period Qtr3 2021 to Qtr1 2024 was available.

### 5.1.3 module 1.3 Extracting the Q&A section.

This module splits the transcripts into MD&A (Management Discussion and Analysis)+ Q&A combined and Q&A only sections and writes the exttracted HTML format to csv files *'transcripts.csv'* (MD&A and Q&A combined) and‘*transcripts\_Qs\_and\_As.csv'* ( Q&A only).



Figure 7. 3 Splitting the raw transcripts into MD&A and Q&A sections.

This module reads the csv file *'transcriptscraped\_test\_data.csv'* created in module 1.2 which contains the raw HTML web scraped transcripts. The HTML was extracted and parsed using BeautifulSoup. The two main sections of the transcripts are the MD&A (Management Discussion and Analysis or The Company Statement) and the Q&A (Question and Answer section). Extracting the MD&A. By inspecting the HTML, it was be seen that MD&A lies between the tags

*'<strong>Company Participants</strong>' or '<strong>Corporate Participants</strong>'*

and *'<strong>Question-and-Answer Session</strong>'.* BeautifulSoup4 was used to search for these tags and extract the text between them labelling it as ‘*company\_statement’.*

Extracting the Q&A section. The Q&A section was found by inspection to lie between the tags *'<strong>Question-and-Answer Session</strong>' and ‘twitContent'.*

This section was extracted and labelled it as ‘*q\_and\_a’.*

Section ‘*company\_statement’ and* section ‘*q\_and\_a’* were both written in the extracted HTML format to file *'transcripts.csv'* in chunks of 8,000 characters per row.

The Q&A section was written separately to file *'transcripts\_Qs\_and\_As.csv'* The text was placed in column ‘transcript\_text’in chunks of 8000 characters per rowby function ‘*write\_q\_and\_a\_html\_to\_csv()’*. This is the section of the transcripts that will be analysed.

### 5.1.4 module 1.4 extracting questions and answers separately.

Purpose: extract the questions and answers separetly in plain text.

In this module the input file *'transcripts\_Qs\_and\_As.csv'* created in module\_3A is read and the questions and answers extracted separately in plain text and written to file *'Transcripts\_Qs\_and\_As\_Split.csv'.* The following columns were added to the file: 'Company\_Name', 'Ticker', 'GICS Sector', 'Text', 'QUARTER', 'QTR', 'day\_date\_formatted'.

These were updated with values from a previously created file based on the ‘ID’ 'transcripts\_all\_analysed\_level1\_grouped\_modified\_with\_sector\_updated.csv'.

Questions and answers can be found in the HTML by locating the ‘p’ tags ‘strong’, ‘span’, ‘class’ and ‘question’ or ‘answer’. The extracted questions and answers were appended to a list ‘qa\_list’. Lines withn less than ten words or containing the text ‘Next question’ were removed from this list. The data frames containing the questions and answers and the new columns were merged and written to output file *'Transcripts\_Qs\_and\_As\_Split.csv'.*

## 5.2 Stage 2 input to FinBERT

The FinBERT model.

**Packages Used**

1. **Transformers**: A library by Hugging Face that provides pre-trained models and tools for natural language processing (NLP). It is used to load the FinBERT model and tokenizer.
   * BertForSequenceClassification: The FinBERT model for sequence classification tasks.
   * BertTokenizer: The tokenizer for tokenizing text inputs according to the BERT model.
2. **Torch**: A Python library used for deep learning. It provides data structures and functions for tensor computation and model training.
   * torch: The main package for tensor computation and model operations.
3. **Pandas**: A data manipulation and analysis library. It provides data structures like DataFrames for working with structured data.
   * pandas: The main package for reading and manipulating data.
4. **CSV**: A module in Python for reading and writing CSV files.
   * csv: Used for writing processed data to a CSV file.

Steps:

Step 1 • Purpose: A dictionary mapping sentiment class indices to sentiment labels.

Step 2 • Purpose: Load the FinBERT tokenizer and model for sequence classification. FinBERT is a BERT-based model fine-tuned on financial text.

Step 3 • Purpose: Tokenize the input text and return the token IDs, attention mask, and total length. The tokenizer converts text into tokens that the model can understand.

Step 4 • Purpose: Split the input text into chunks (of length 510 tokens), apply the model to each chunk, and compute class probabilities. The function appends the [CLS] and [SEP] tokens to each chunk, as required by BERT-based models.

Step 5 • Purpose: Compute the mean probabilities across all chunks to get an overall sentiment for the entire input text. This function ensures that gradients are not computed, saving memory.

Step 6 • Purpose: The main script block processes the data from the input CSV file and writes the results to a new CSV file. It reads the data, performs sentiment analysis using the defined functions, and writes the results along with sentiment counts to the output file.

**Summary**

* **Software Packages**: The script uses transformers for loading the FinBERT model and tokenizer, torch for tensor operations and model inference, pandas for data manipulation, and csv for writing output data.
* **Functionality**: The script tokenizes input text, splits it into manageable chunks, predicts sentiment probabilities using FinBERT, computes mean probabilities across chunks, and writes the results to a CSV file. The sentiments are classified as positive, negative, or neutral.

### 5.2.1 module 2.1 Input to FinBERT

Purpose: Input the series of questions and answer files *'Transcripts\_Qs\_and\_As\_Split.csv'* to the model.



Explanation.

The questions and answers extracted from the Q&A sections of the ECC transcripts were contained in a series of csv files. Each question and answer appeared on separate rows. These files were input to the FinBERT model which classified each question and answer with a sentiment of positive, negative or neutral and provided a count of the 512 token length sections of each question and answer classified in each category. In practice all questions and answers were less that this length. The output consisted of the input file with the four new columns, sentiment, count of +ve, count of -ve and count of neutral.

## 5.3 Stage 3 - Results

### 5.3.1 module 3.1 combine FinBERT output files

Purpose: Combine the separate output files from FinBERT into one combined file.

The series of output files from the FinBERT model were aggregated and grouped on the ID and ticker fields. This produced a combined file with the complete set of transcripts, in each case split into questions (Q) and answers (A) on two separate rows for each ID. An extract from the file shown in Fig xxx.

The columns ‘sum of +ve’, ‘sum of -ve’ and ‘sum of neutral’ show the number of questions and answers that were classified into each of the sentiment categories by the model.



Fig XXX

The following columns were then added to the file.

### 5.3.2 module 3.2 update tone and stock price

Updating stock price data and tone in respect of each transcript.

Purpose: calculate tone and the change in average stock price

Following the method of Brockman et al (Brockman, Li and Price, 2015) a measure for linguistic tone was defined. Linguist Tone (Tone) is defined as the difference between the sum of positive and negative sentiment counts divided by the sum of the two.

*Tone = (sum of Positive – sum of Negative) / (sum of Positive + sum of Negative)*

This definition provides a measure of relative positivity and is bounded between –1 and +1.

The results file was updated with a new column ‘tone’ calculated as above for each transcript.

Stock price data is contained in folder ‘Stock\_Data\_Files’. This folder contains a file with stock price data for each of the S&P 500 companies. The files are named for the ticker symbol, for example, the stock price data for Apple Inc is contained in file AAPL.csv and so on. Module\_7 reads these files individually in conjunstion with the results file. Based on the ticker and date fields the average close price over five days immediately preceding the ECC date and the average close price for the five days immediately following the ECC date is calculated for each transcript. The difference between these two figures is the five-day price change for the stock centered on the ECC date. The same process was repeated for the two day and one price changes. This module provided three data points in respect of each ECC transcript. Tone, 5-Day price change, 2-Day price change and 1-Day price change.

https://stockanalysis.com/list/sp-500-stocks/

Stage 1 Web scraping the raw data and preparing it for input to the sntiment analysis model.

module 1.1 Webscraping the transcripts list.

module 1.2 Web scraping the individual S&P 500 transcripts.

module 1.3 Splitting the transcripts and ectracting the Q&A section.

The data collection and processing is made up of seven modules. The pipeline is shown in Figure 7.1. Modules 1 and 2 are concerned with web scraping the raw data. Modules 3 and 4 process the raw data and produces files that contain the Q&A section of each transcript split into individual questions and answers. Some additional fields are also added to the files for later processing. Module\_4 is the FinBERT processing module. Module\_5 combines the series of FinBERT output files. Module\_6 iterates over all of the FinBERT output files. They are concatenated and grouped producing a single output file ‘updated\_combined\_file\_with\_compatible\_date .csv ‘ Module\_6 updates this file by the addition of felds for tone and stock price change data are added and updated. Module\_7 carries out the analysis and Pearson correlations on the tone and price change data.

The results of data collection and the computed measures, tone and price change values, are contained in file ‘updated\_combined\_file\_with\_compatible\_date .csv’.

Module\_8 presents the results of the correlation between Tone of the questions and answers and the corresponding stock price changes over one day, two days and five days following the conference calls.

Chapter 6

# Results and Analysis

## Results

The values of the tone and price change calculations are contained in file ‘updated\_combined\_file\_with\_compatible\_date .csv ‘ This is the final output file from module 3.2.



Figure xxx : Extract from file ‘updated\_combined\_file\_with\_compatible\_date .csv ‘

Explanation of the fields.

|  |  |
| --- | --- |
| **File: updated\_combined\_file\_with\_compatible\_date .csv** | |
| ID | Unique transcript ID |
| Call\_Section | Identifies the row as Questions (Q) or Answers (A) |
| GICS Sector | GICS sector the company belongs to.(Global Industry Classification Standard) |
| QTR | Financial quarter |
| day\_date | Date of transcript |
| Company\_AName | Company Name |
| Ticker | Company stock market ticker symbol |
| Text | Description of transcript |
| sum of +ve | sum of the Qs /As classed as positive sentiment |
| sum of -ve | sum of the Qs /As classed as negative sentiment |
| sum of neutral | sum of the Qs /As classed as neutral sentiment |
| tone | tone calculated as (sum of positive - sum of negative)/ (sum of positive + sum of negative) |
| price\_chng\_5day | difference in avg close price for the 5 days preceding the date of the ECC and the 5 days post the ECC date. |
| price\_chng\_2day | difference in avg close price for the 2 days preceding the date of the ECC and the 2 days post the ECC date. |
| price\_chng\_1day | difference in close price between the day before and day after the ECC date. |
| date\_compatible | a date field constructed to ensure date compatibility with stock data file dates. |

Figure XXX. fields in file ‘updated\_combined\_file\_with\_compatible\_date .csv ‘

Findings

Overall tone



Table XXX overall tone of the Q&A sections

The overall tone of the Q&A section and tone of the answers was found to be positive. The tone of the questions was found to be negative at -0.197. This result is in line with Brockman et al who found the tone of questions to be

Average tone by quarter.



Table XXX Average tone by quarter

The average tone by financial quarter is shown in Table XXX

Average tone by GICS sector



Table XXX Average tone by GICS sector

Processing on Kaggle.

The files were input to the model using an account created on Kaggle xxxxx. Kaggle, a subsiriary of Google, is an online community of data scientists and machine learning engineers. It provides users with access to and use of use of AI models such as FinBERT leveraging the power of its processers. A account was created with Kaggel and the FinBERT model was accessed. Processing of the transcripts through the model requires in practice the use of a GPU. Various GPU options are available on Kaggle. The Nvidia Tesla P100 GPU was chosen. Processing time for the entire Q&A transcript data set using this GPU was approximately eleven hours.

The fields ‘sum of +ve’, ‘sum\_of\_-ve and sum of neutral contain the sum of the classifications of questions and answers by the model in the case of each transcript. The results file was produced as follows. The transcripts dataset was input to FinBERT in a series of csv files with questions and answers accommodated separately on each row in sequence. The model classified each row as positive, negative or neutral, and output the file with four additional fields: Sentiment, count of positive, count of negative and count of neutral. Access to a fast GPU is needed to process the input files. Kaggle was used making use of the Nvidia Tesla P100 GPU. The size of file that could be input had to be small enough to ensure it was processed to completion. The Kaggel accelerator had a tendency to drop out randomly if the processing took longer than around 15 minutes at a time. In order to get around this the input dataset was split into a manageable size files and input to the model in stages. All of the output files were then combined and summed on each unique ID and its associated question and answer.

Tone

Overall tone



Tbl 8. 1 Tone of transcripts sections.

Table 8.1shows the values of tone obtained for the overall Q&A section and for the Questions and the Answers separetly. The tone of answers was found to be positive with a value of +0.534. Questions had a negative tone of -0.197. Overall the Q&A section had a positive tone value of +0.288.

This is in line with the results of Brockman et al (Brockman, Li and Price, 2015) who measured the sentiment of conference call transcripts based on a lexicon method using the Loughran and McDonald (2011) dictionary. Brockman et al found that the most optimistic tone during an entire conference call was found to be the MD&A. During the Q&A session, the tone of answers were found to be significantly more positive than the tone of questions.

Average tone

## Analysis

Correlation.

In order to examine if a relationship exists between the tone of the Q&A section and subsequent stock price movements a statistical measure of the linear relationship between the tone and price change can be calculated. The Pearson correlation coefficient (r) is one such measure. This is a numerical measure of strength and direction of the linear correlation or statistical relationship between two sets of variables. The correlation coefficient ranges from +1 and -1. Well correlated datasets have a correlation index of near +1 or -1 if inversely correlated. Uncorrelated datasets have a correlation index close to zero. A formula for the Pearson correlation coefficient is shown in Figure xxx. The numerator is the sum of the product of the data point differences from the mean value. The denominator is the square root of the product of sums of squared differences.(Berman, 2016)



Figure XXX Pearson correlation coefficient

A Pearson correlation function is available in Python library ‘Scipy’. One of the functions available in Scipy is ‘pearsonr’ which was used to compute the Pearson correlation coefficient for tone and price change (5-day, 2-day and 1-day).

The ‘pearsonr’ function also computes the P-value. The P-value is a measure of the statistical significance of the correlation coefficient or the probability of two unrelated data sets producing the computed correlation index. A lower P-value indicates a lower probability of two uncorrelated data lists producing the computed correlation coefficient.

Correlation of tone with stock price change.

The results file contains the fields ‘price\_chng\_1day’, ‘price\_chng\_2day’ and ‘price\_chng\_5day’. The values are the difference in the mean stock market close price of the stock over the five/two/one days immediately prior to the ECC (Earnings Conference Call) date and the five/two/one days immediately following the ECC date. The Pearson correlation of tone and price change was calculated using the Pearson correlation package in Python.



Tbl 8. 2 Correlation of Tone and price change

Table 8.2 shows the correlation matrix for tone and price change. The highest Pearson correlation was found for questions and the one-day price change. The value of 0.0749 indicates a weak positive relationship between the tone of questions and the following day stock price change. The associated scatter plots show this weak positive correlation. This is in line with the results of Blockman et al.

Questions: tone and price change correlation



Tbl 8. 3 Correlation of tone and price change: questions only

Correlation matrix : questions only

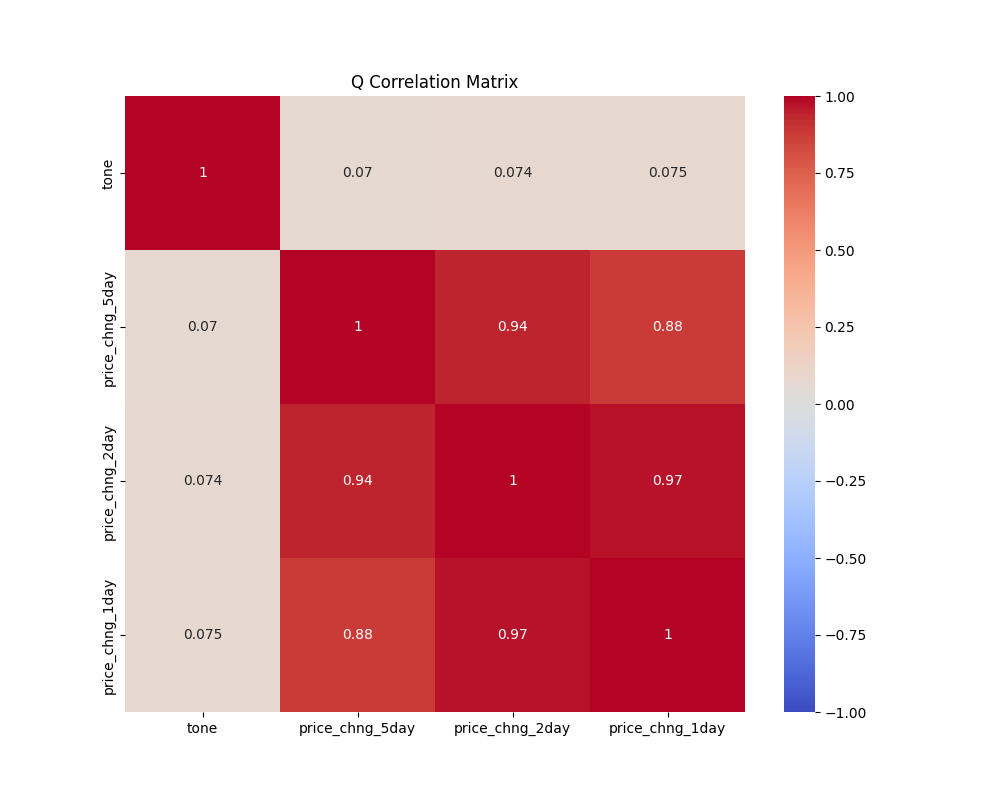


Figure 7.1 2 correlation matrix : questions only

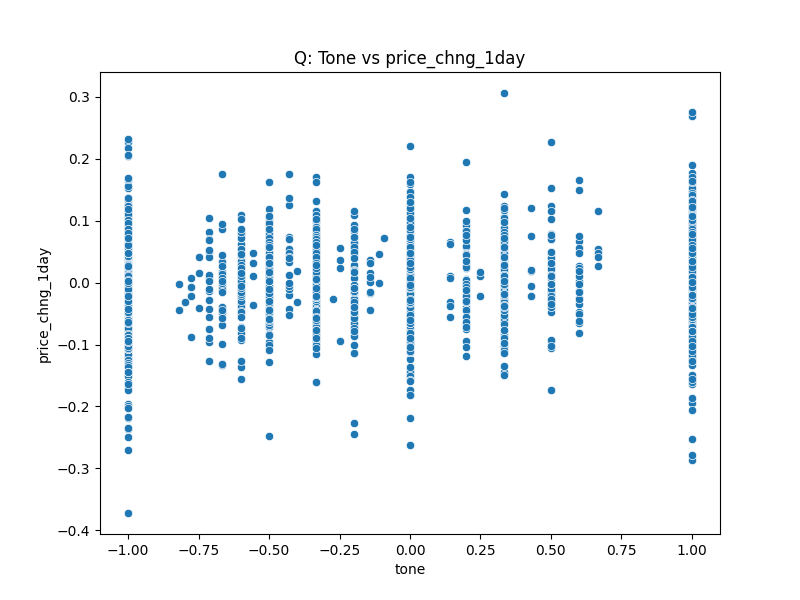


Figure 7.1 3 scatter plot of tone vs price change 1day: questions only

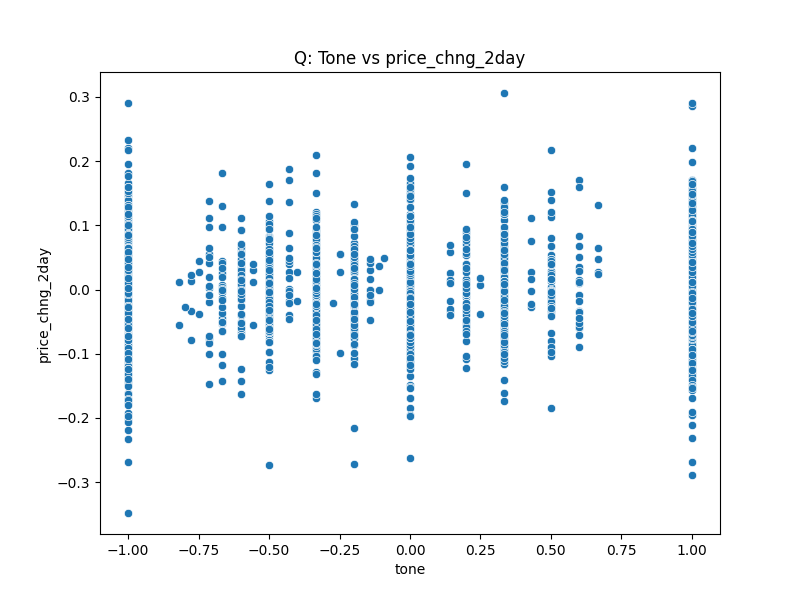


Figure 7.1 4 scatter plot of tone vs price change 2day: questions only

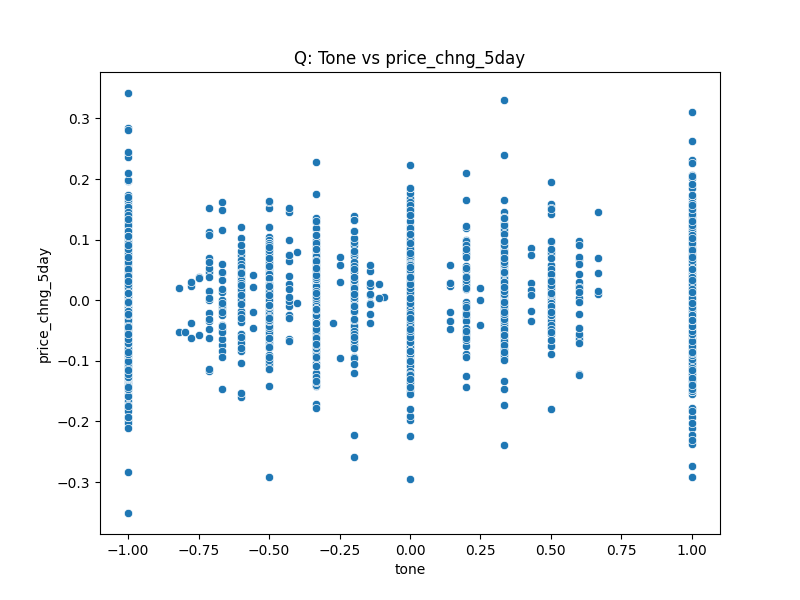


Figure 7.1 5 scatter plot of tone vs price change 5day: questions only



Tbl 8. 4 tone and price change correlation. Answers only

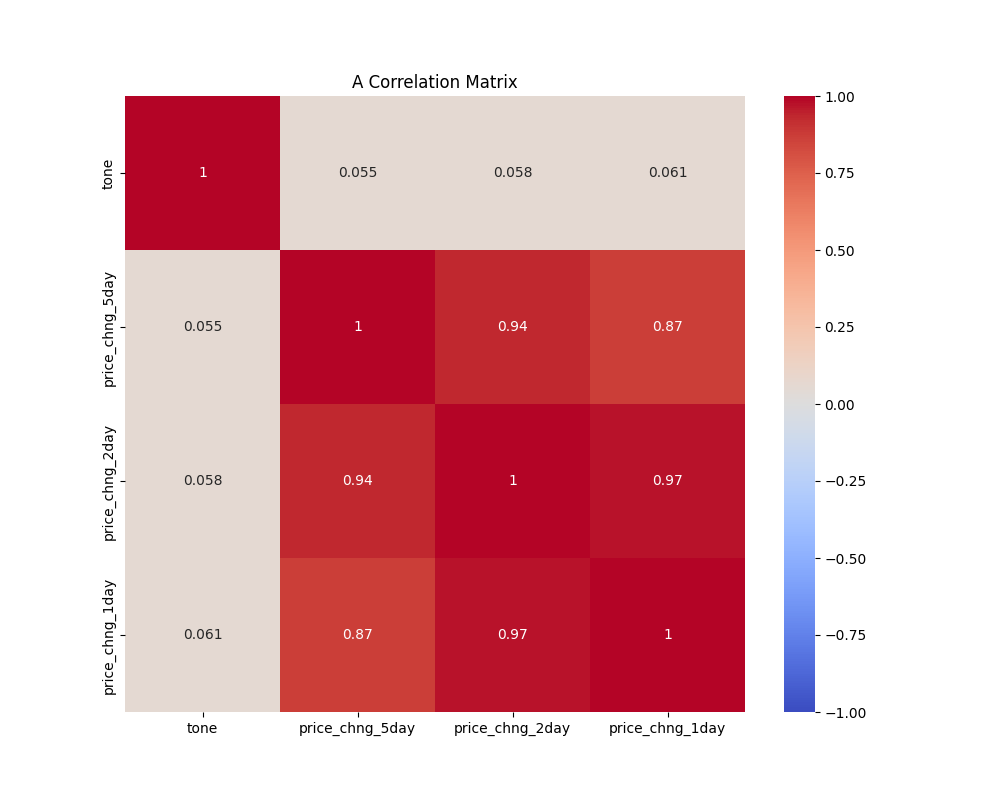


Figure 7.1 6 correlation matrix : answers only

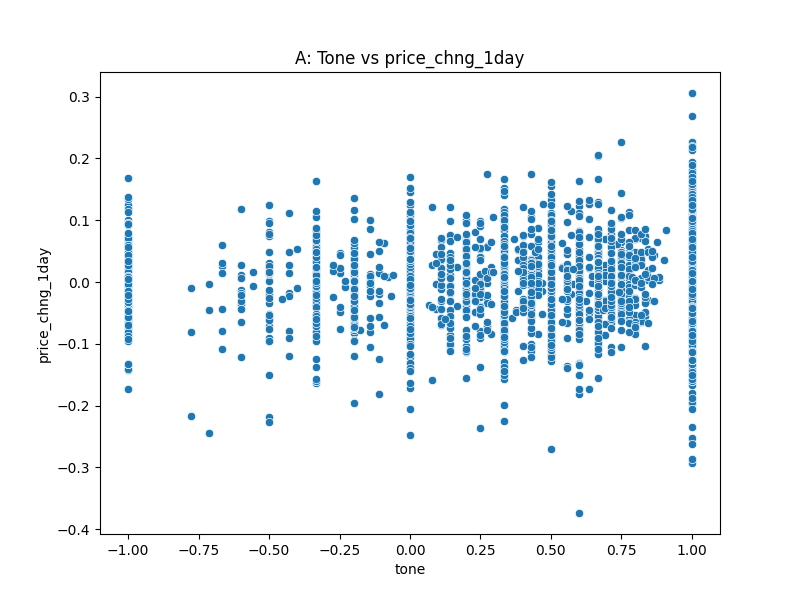


Figure 7.1 7 scatter plot of tone vs price change 1day: answers only

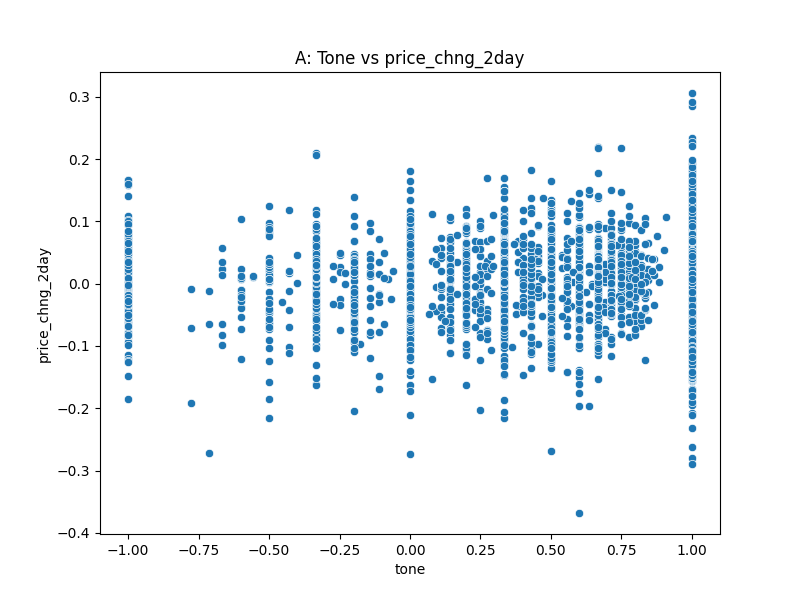


Figure 7.1 8 scatter plot of tone vs price change 2day: answers only

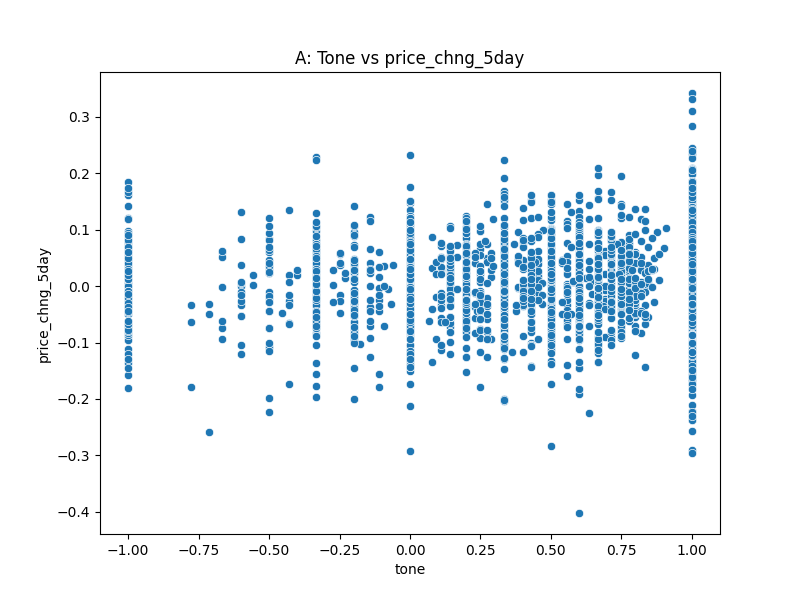


Figure 7.1 9 scatter plot of tone vs price change 5day: answers only



Tbl 8. 5 corelation tone and price change Q&A section

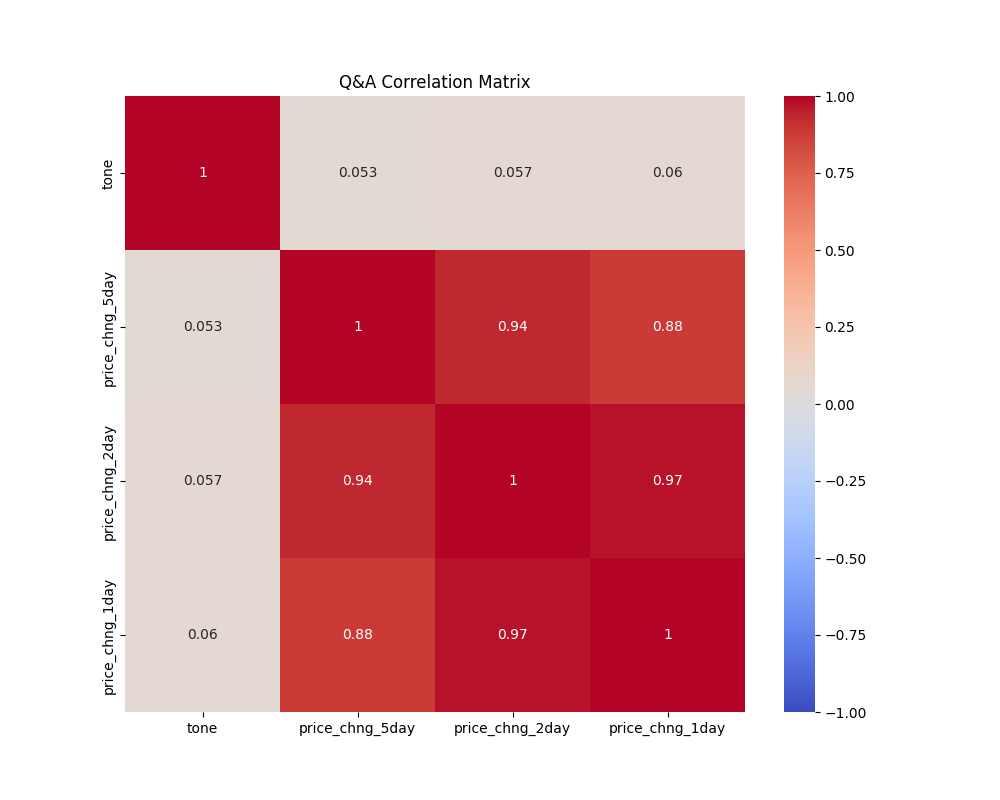


Figure 7.1 10 tone and price change correlation matrix. overall Q&A section

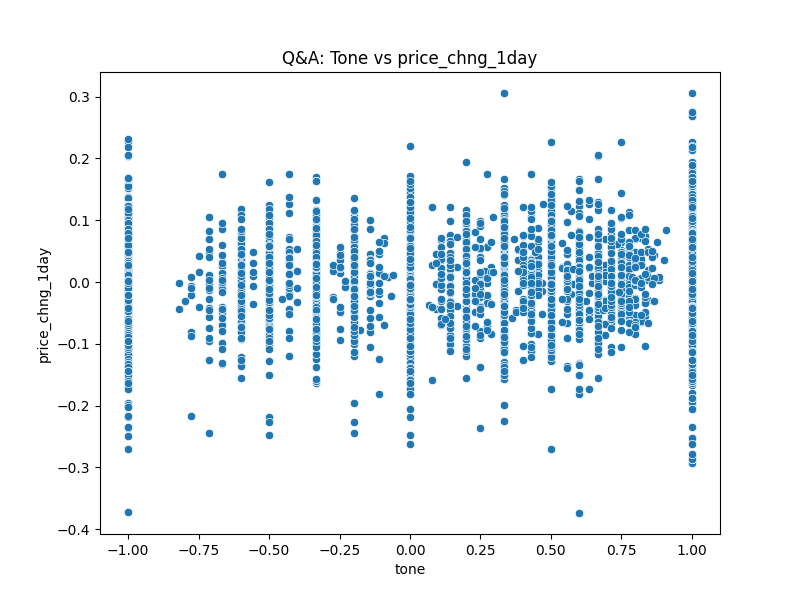


Figure 7.1 11 scatter plot of tone vs price change 1day: overall Q&A section

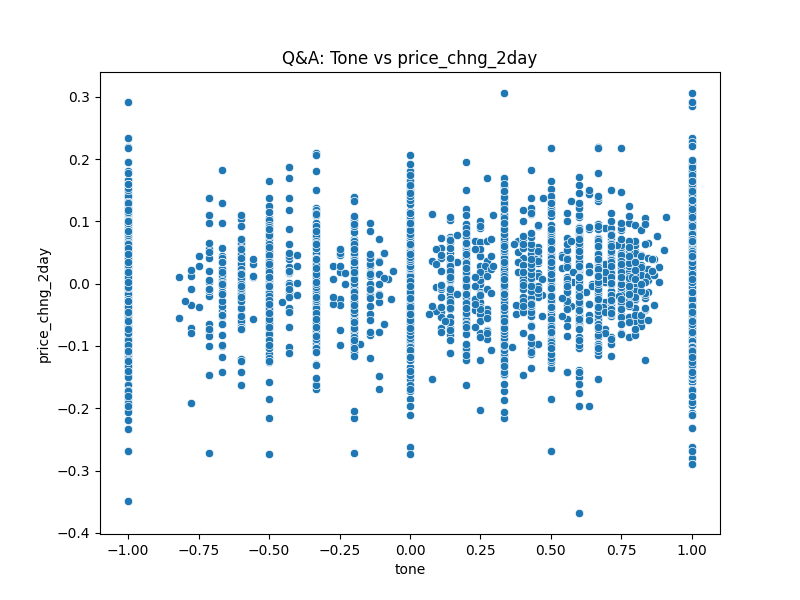


Figure 7.1 12 scatter plot of tone vs price change 2day: overall Q&A section

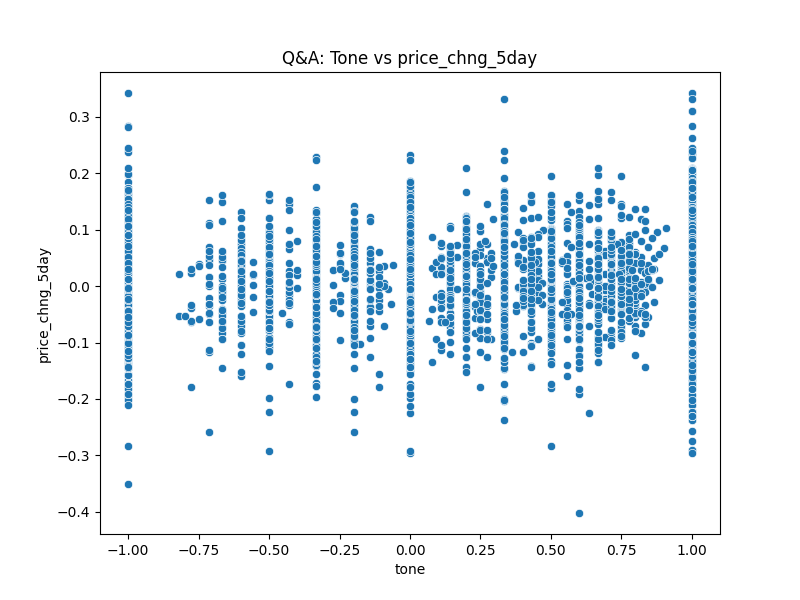


Figure 7.1 13 scatter plot of tone vs price change 5day: overall Q&A section

Chapter 7

# Conclusions and Future Work

# References

Araci, D. (2019) ‘FinBERT: Financial Sentiment Analysis with Pre-trained Language Models’. Available at: https://doi.org/10.48550/ARXIV.1908.10063.

Berman, J.J. (2016) *Data simplification: taming information with open source tools*. Amsterdam: Elsevier ; Morgan Kaufmann.

Brockman, P., Li, X. and Price, S.M. (2015) ‘Differences in Conference Call Tones: Managers vs. Analysts’, *Financial Analysts Journal*, 71(4), pp. 24–42. Available at: https://doi.org/10.2469/faj.v71.n4.1.

Bv, N. *et al.* (2023) ‘Deploying NLP Techniques for Earnings Call Transcripts for Financial Analysis: A Reverse Phenomenon Paradigm’, in *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, Nepal: IEEE, pp. 368–375. Available at: https://doi.org/10.1109/I-SMAC58438.2023.10290494.

Davis, A.K., Piger, J.M. and Sedor, L.M. (2012) ‘Beyond the Numbers: Measuring the Information Content of Earnings Press Release Language\*’, *Contemporary Accounting Research*, 29(3), pp. 845–868. Available at: https://doi.org/10.1111/j.1911-3846.2011.01130.x.

Kimbrough, M.D. (2005) ‘The Effect of Conference Calls on Analyst and Market Underreaction to Earnings Announcements’, *The Accounting Review*, 80(1), pp. 189–219. Available at: https://doi.org/10.2308/accr.2005.80.1.189.

Malo, P. *et al.* (2014) ‘Good debt or bad debt: Detecting semantic orientations in economic texts’, *Journal of the Association for Information Science and Technology*, 65(4), pp. 782–796. Available at: https://doi.org/10.1002/asi.23062.

Medya, S. *et al.* (2022) ‘An Exploratory Study of Stock Price Movements from Earnings Calls’, in *Companion Proceedings of the Web Conference 2022*. *WWW ’22: The ACM Web Conference 2022*, Virtual Event, Lyon France: ACM, pp. 20–31. Available at: https://doi.org/10.1145/3487553.3524205.

Mishev, K. *et al.* (2020) ‘Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers’, *IEEE Access*, 8, pp. 131662–131682. Available at: https://doi.org/10.1109/ACCESS.2020.3009626.

Price, S.M. *et al.* (2012) ‘Earnings conference calls and stock returns: The incremental informativeness of textual tone’, *Journal of Banking & Finance*, 36(4), pp. 992–1011. Available at: https://doi.org/10.1016/j.jbankfin.2011.10.013.

Vaswani, A. *et al.* (2023) ‘Attention Is All You Need’. arXiv. Available at: http://arxiv.org/abs/1706.03762 (Accessed: 13 August 2024).

Yamamoto, R. *et al.* (2022) ‘Managements’ tone strategies by earnings call transcripts in the global markets’, *Journal of Asset Management*, 23(3), pp. 246–255. Available at: https://doi.org/10.1057/s41260-022-00256-2.

# Appendix 1

Financial Data Extraction and Processing Pipeline (FDEP) step by step details.

