A picture containing application

Description automatically generated

**Sentiment Analysis of Financial Transcripts**

by

**Brian Higgins**

**DRAFT\_2**

Research Supervisor: Dr Josephine Griffith

**MSc in Software Engineering and Database Technologies**

School of Computer Science,

College of Science and Engineering

University of Galway

Date: September 2024

DECLARATION

I, Brian Higgins, do hereby declare that this thesis entitled ‘Sentiment Analysis of Financial Transcripts’ 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.*

# **Table of Contents**

[**Table of Contents** iv](#_Toc174971127)

[**List of Figures** vi](#_Toc174971128)

[**List of tables** viii](#_Toc174971129)

[**Glossary** ix](#_Toc174971130)

[Abstract 1](#_Toc174971131)

[**1 Introduction** 2](#_Toc174971132)

[1.1 Research topic and motivation. 2](#_Toc174971133)

[1.2 Background 3](#_Toc174971134)

[1.3 Earnings Conference Calls 4](#_Toc174971135)

[1.4 Sentiment Analysis 7](#_Toc174971136)

[1.5 Language models. 9](#_Toc174971137)

[1.6 BERT 10](#_Toc174971138)

[1.7 FinBERT 12](#_Toc174971139)

[1.8 Thesis structure 14](#_Toc174971140)

[**2** **Literature Review** 15](#_Toc174971141)

[**3 Data** 17](#_Toc174971142)

[3.1 Data description 17](#_Toc174971143)

[3.2 Available Transcripts 18](#_Toc174971144)

[3.3 Individual transcripts 20](#_Toc174971145)

[3.4 Data Composition 23](#_Toc174971146)

[4 Testing the FinBERT model 24](#_Toc174971147)

[**5 Financial Data Extraction and Processing Pipeline** 25](#_Toc174971148)

[5.1 Stage1\_module\_1 Web scraping the list of available transcripts 29](#_Toc174971149)

[5.2 Stage1\_module\_2 Web scraping the set of S&P 500 transcripts 31](#_Toc174971150)

[5.3 Stage1\_module\_3 Extracting the Q&A section. 33](#_Toc174971151)

[5.4 Stage1\_module\_4 Extracting questions and answers separately 36](#_Toc174971152)

[5.5 Stage2\_module\_1 Sentiment analysis using FinBERT 37](#_Toc174971153)

[5.6 Stage3\_module\_1 combine the FinBERT output files 39](#_Toc174971154)

[5.7 Stage3\_module\_2 compute tone and stock price change 40](#_Toc174971155)

[Stage 1 Web scraping the raw data and preparing it for input to the sntiment analysis model **Error! Bookmark not defined.**](#_Toc174971156)

[**6 Results** 42](#_Toc174971157)

[6.1 Results file 42](#_Toc174971158)

[6.2 Tone results 44](#_Toc174971159)

[6.3 Analysis 48](#_Toc174971160)

[**7 Conclusions and Future Work** 59](#_Toc174971161)

[**8 References** 60](#_Toc174971162)

[**9 Appendices** 62](#_Toc174971163)

[9.1 Appendix 1 : Code – Implementation of FinBERT 62](#_Toc174971164)

# **List of Figures**

[Figure 3.1 JSON object holding transcript information 19](#_Toc174799483)

[Figure 3.2 Extract from JSON object holding individual transcript 21](#_Toc174799484)

[Figure 3.3 Sample from raw data csv file 33](#_Toc174799485)

[Figure 3.4 Sample from file containing the extracted MD&A and the Q&A sections 35](#_Toc174799486)

[Figure 3.5 Data composition by financial quarter 23](#_Toc174799487)

[Figure 3.6 Data composition by GICS sector 23](#_Toc174799488)

[Figure 5.1 Financial Data Sentiment Stock Correlation Pipeline 30](#_Toc175515100)

[Figure 5.4 Sample from raw data csv file 37](#_Toc175515101)

[Figure 5.6 Sample from file containing the extracted MD&A and the Q&A sections 38](#_Toc175515102)

[Figure 5.7 Output file from sentiment analysis processing 42](#_Toc175515103)

[Figure 6.1 Extract from results file 43](#_Toc175321447)

[Figure 6.2 Explanation of fields in the results file 44](#_Toc175321448)

[Figure 6.3 Pearson correlation coefficient (r) formula 48](#_Toc175321449)

[Figure 6. 4 Correlation matrix: questions tone and stock price changes 50](#_Toc175321450)

[Figure 6.5 Scatter plot of questions tone vs 1-day stock price change 51](#_Toc175321451)

[Figure 6.6 Scatter plot of questions tone vs 2-day stock price change 51](#_Toc175321452)

[Figure 6.7 Scatter plot of questions tone vs 5-day stock price change 52](#_Toc175321453)

[Figure 6.8 Correlation matrix: Answers tone and stock price movements 53](#_Toc175321454)

[Figure 6.9 Scatter plot of answers tone vs 1-day stock price movement 54](#_Toc175321455)

[Figure 6.10 Scatter plot of answers tone vs 2-day stock price movement 54](#_Toc175321456)

[Figure 6.11 Scatter plot of answers tone vs 5-day stock price movement 55](#_Toc175321457)

[Figure 6.12 Correlation matrix: Q&A section tone and stock price changes 56](#_Toc175321458)

[Figure 6.13 Scatter plot of Q&A section tone vs 1-day stock price change 57](#_Toc175321459)

[Figure 6.14 Scatter plot of Q&A section tone vs 2-day stock price change 57](#_Toc175321460)

[Figure 6.15 Scatter plot of Q&A section tone vs 5-day price change 58](#_Toc175321461)

# **List of tables**

[Table 4. 1 FinBERT accuracy check 24](#_Toc174884684)

[Table 6.1 Tone of transcripts sections 45](#_Toc175321523)

[Table 6.2 Average tone by quarter 46](#_Toc175321524)

[Table 6.3 Average tone by GICS sector 47](#_Toc175321525)

[Table 6.4 Correlation of Tone and price change 49](#_Toc175321526)

[Table 6.5 Correlation of questions tone and stock price changes 50](#_Toc175321527)

[Table 6.6 Correlation of answers tone and stock price changes 53](#_Toc175321528)

[Table 6.7 Corelation of Q&A section tone and stock price changes 56](#_Toc175321529)

# **Glossary**

API

(Application Programming Interface) 18

BERT

(Bidirectional Encoder Representations from Transformers) 2

BoW

(Bag of Words) 8

ECC

(Earnings Conference Calls) 1

EPS

(earnings per share) 2

FinBERT

(BERT for Financial NLP ) 2

GICS

(Global Industry Classification Standard) 25

HTML

(Hypertext Markup Language) 23

NLP

(Natural Language Processing) 2

S&P 500

(Standard & Poor's 500) 7

SEC

(Securities and Exchange Commission) 6

SVM

(Support Vector Machines) 8

URL

(uniform resource locator) 18

# 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 to discuss the latest financial results with analysts and investors and answer their questions directly. Transcripts of ECC are made available on company websites following the calls. Stock market analysts and investors carefully study ECC in attempts to discover new information 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 previously disclosed in the earlier earnings press release. Information is also contained in the sentiment or linguist tone of the participants. Studies have found that the sentiment of earnings calls may 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 transformer-based language model specifically designed for financial sentiment analysis. The research will be implemented in Python version 3.9.8. Selenium WebDriver will be used to automatically control the browser during web scraping. Loading of the model will be carried out using the transformers package from Hugging Face. The Python Torch library will be used for tensor computation and model operation. Data manipulation and analysis will be carried out using the Python Pandas library.

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

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 Natural Language Processing (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 NLP include the introduction of transformer deep learning models with improved sentiment classification accuracy. This thesis will employ FinBERT (BERT for Financial NLP ), a pretrained transformer-based NLP model, to analyse the sentiment of earnings conference call transcripts. FinBERT is a language model based on the BERT (Bidirectional Encoder Representations from Transformers) and is specifically designed for NLP tasks in the financial domain. The sentiment results will be used in conjunction with stock market data to investigate if a correlation is evident between the sentiment and 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 changes within a period of one to five days.

## 1.2 Background

Attempting to predict stock market movements if a difficult task and as yet there is no satisfactory method. As a result, there is much research in this area. Understanding the factors affecting stock market movements is of great interest to investors and analysts.

Two main stock market behaviour theories. The Efficient Market Hypothesis (EMH) and the Adaptive Market Hypothesis (AMH). The EMH (Fama, 1970) proposes that investors always act rationally and that the stock market is “efficient”. Efficient in this sense means that all available information is already reflected in stock prices. It is therefore not possible for investors to consistently outperform the market through technical or fundamental analysis. The AMH (Lo, 2004) asserts that investors do not always act rationally and that the stock market is not always efficient in immediately absorbing relevant information. It adapts and evolves over time being influenced by the actions of the market participants and external factors. It views the market as a system where rational and irrational behaviours compete and the EMH holds at some periods but not always.

Behavioural finance is an area of study which examines how investment decisions are influenced by phycological factors, biases, rationality, irrationality and emotions. It aligns with the AMH as it proposes that investor behaviour is not always rational which can lead to an inefficient market in periods of uncertainty or volatility. (Picasso *et al.*, 2019).

According to EMH all publicly available information, including sentiment from Earnings Conference Calls should be reflected immediately in stock prices. If the sentiment of ECC can be linked to subsequent stock price movements this would challenge the EMH as it might indicate inefficiencies in the market.

The AMH allows room for the market response to sentiment to vary and adapt over time and under different market conditions. In stable conditions the market response to sentiment may not be as noticeable as the response in periods of high volatility.

An active area of stock market research is the degree to which the market builds-in the sentiment contained in ECC.

## 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. Company 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 (Chief Executive Officer) and CFO (Chief Financial Officer), 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 website on the investor relations page[[1]](#footnote-1).

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 Q&A 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 (Standard & Poor's 500). The exception being the company Berkshire Hathaway Inc (BRK), which has no record of earnings conference calls. The S&P 500 index is a stock market index which tracks the performance of 500 of the largest US based public companies listed on US stock markets. It is used as a barometer of the US economy by investors and financial analysts due to the wide scope of industries which make it up.

## 1.4 Sentiment Analysis

Determining the sentiment of ECC is useful to investors and analysts as it may give an indication of the future stock price movements. 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 methods use rule-based lexicon approaches which rely on predefined lists of words classified as positive negative or neutral. Examples are SentiWordNet and VADER sentiment lexicons. Machine learning methods include the Naïve Bayes probabilistic model based on Bayes’ theorem. Support Vector Machines (SVM) construct hyperplanes in a high-dimensional space to classify data in a supervised learning algorithm. Decision tree methods such as Random Forrest. Logistic regression is another technique which predicts the probability of an input belonging to a particular class. Another method for sentiment analysis of a given text is the Bag of Words (BoW) method which counts the frequency of pre-classified words in a text. This method is often combined the classifiers referred to above. Term Frequency-Inverse Document Frequency (TF-IDF), which is an enhancement of BoW is frequently used in sentiment analysis. The algorithm weighs the relative importance of terms in a text to identify the significant words or phrases and can categorise the text as positive, negative or neutral.

Deep learning techniques employ word embedding text representation. Examples include Word2Vec which make use of neural networks to learn vector representations of words in a continuous vector space allowing it to capture semantic relationships or the connection between words based on their meaning. FastText, extends Word2Vec by considering sub word information, enabling it to better handle out-of-vocabulary words. GloVe (Global Vectors for Word Representation) captures global statistical information of a corpus to produce word embeddings. The resulting word embeddings that can be fed into a language model for sentiment analysis.

LSTM (Long Short-Term Memory) modes are a type of recurrent neural network (RNN) that can capture long-term dependencies in sequential data such as text. Their variants GRU (Gated Recurrent Units) have a simpler architecture and faster training times.  LSTMs Long Short-Term Memory is a type of RNNs Recurrent Neural Network that can detain long-term dependencies in sequential data. LSTMs are able to process and analyse sequential data, such as time series, text, and speech.

Transformer based language models.

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 ambiguous. This creates challenges for computers in understanding human language. Areas of difficulty include sarcasm, humour, inflection and tone.

Language models use statistical and computational methods 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. They represent a structured, learned abstraction of language, similar to how models in other scientific fields represent real-world phenomena. Transformer based language models are used extensively in sentiment analysis and classification. Their improved accuracy and robustness result in their replacing traditional rule based and machine learning methods in many applications. 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.

## 1.6 BERT

An overview the BERT (Bidirectional Encoder Representations from Transformers) transformer based language model will be given here as it is the language model on which FinBERT is based.

BERT is a transformer-based language model. It is an open-source architecture for Natural Language Processing (NLP) introduced by Google in 2018 designed to help computers understand naturally spoken human language. Transformers are a type of neural network designed to process sequential data. The Transformer architecture, introduced in the paper "Attention is All You Need" by Vaswani et al. (2017) (Vaswani *et al.*, 2023) enables NLP models to handle long-range dependencies in text more effectively than previous models like RNNs and LSTMs. 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.

The architecture relies on a mechanism called self-attention and eliminates the need for recurrent layers, allowing for faster and more parallelizable training. The self-attention mechanism captures and understands relationships among words in a sentence. The bidirectional transformers at the center of BERT's design make this possible. Words may change meaning as sentences develop. 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 mechanism enables BERT to consider both preceding and succeeding words in a sentence, enabling it to generate contextually aware embeddings for tokens providing a more comprehensive understanding of context and the relationship between words and phrases across an entire sentence. determine the context and deal with ambiguity.

By looking at all surrounding words, the transformer enables BERT to understand the full context of the word resulting in a better understanding of nuanced meanings and relationships in natural language.

Earlier text representation models such as Word2Vec, GloVe, and FastText represent words as vectors in a continuous vector space. They map each word to a vector, producing static word embeddings which represent only one dimension of that word's meaning, resulting in a word having the same representation regardless of its context.

Pretraining BERT

**To be completed (BH 19 Aug)**

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 process a maximum token length of 512.

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.

There are two versions of BERT, Bert Large which uses an embedding dimension of 1024, and BERT base which uses 768 dimensions.

## 1.7 FinBERT

FinBERT (BERT for financial domain is a pre-trained open-source Natural Language Processing model (NLP) designed specifically for financial sentiment analysis. It is based on the BERT language model. The particular FinBERT model used in this research is produced by the technology company ProsusAi. It is available on the Hugging Face model hub[[2]](#footnote-2). The model was developed by Araci (Araci, 2019) to address the problem of financial sentiment analysis.

The problem of financial sentiment analysis arises from the fact that language used in the financial domain has its own characteristics not usually found is normal everyday language. General purpose sentiment analysis models do not perform well when attempting sentiment analysis of financial text.

The FinBERT model was constructed using the pre-trained BERT model by providing it with additional training on a purely financial corpus, the Reuters TRC2. This additional training exposed the model to the language used in the financial industry and adapted it to this domain. The model was then fine-tuned with labelled data for financial sentiment classification. Fine-tuning was carried out using the dataset of the Financial PhraseBank by Malo et al (Malo *et al.*, 2014).

This is a dataset of 4,845 english sentences randomly selected from financial news. Each sentence has been classified as positive, negative or neutral by 16 financial industry experts according to their opinion of its likely effect on the relevant company stock price. The dataset is available at the researchgate website[[3]](#footnote-3)

Limitations of the FinBERT model.

The Financial PhraseBank classifications of the phrases and sentences are not reflective of 100% agreement by the annotators in all cases. There is inter-annotator disagreement in regard to the accepted classifications. The accepted sentiment is the majority vote in the case of each phrase. The highest rate of disagreement occurs between positive and neutral classifications. The developers of FinBERT report an accuracy of 97% when applying the model to the 100% annotator agreement sub-set of phrases. They report that 73% of the incorrect classifications occur between positive and neutral classifications, and 5% misclassifications between negative and positive.

## 1.8 Thesis structure

This thesis consists of eight chapters.

Chapter 2 provides the introduction and background for the research along with the research goals. The motivation to analyse ECC transcripts is given followed by a description of the structure and conduct ECC and the resulting transcripts. A short summary of sentiment analysis techniques is followed by an outline of language models focussing on BERT, which is the basis of the FinBERT model used in this research to extract the sentiment of the ECC transcripts. A description the FinBERT model is then given.

Chapter 2 reviews the literature relating to the study of ECC transcripts and provides the context for the interest and study ECC transcripts.

Chapter 3 gives a description of the transcript data. Methods to locate and web scrape the ECC transcripts and stock price data are described along with pre-processing the data including extracting the different sections.

Chapter 4 describes an exercise to verify the operation and accuracy of the particular FinBERT model being used.

Chapter 5 provides a detailed description of the Financial Data Extraction and Processing pipeline. This is the data pipeline or series of process steps from raw data sourcing and collection, through pre-processing, input to the sentiment analysis model and collating the output results.

Chapter 6 presents the sentiment analysis of results.

Chapter 7 presents the correlation analysis of sentiment with stock price changes.

Chapter 8 discusses the conclusions and provides suggestions for future research.

# **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 conference 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 extracted 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, negative, 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 conference 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 investment 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 performance.

*To be completed(BH 19 Aug)*

# **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 companies with analysts and investors. Transcripts were web scraped from the Seeking Alpha website[[4]](#footnote-4).Historical stock price data for each of the 500 companies that make up the S&P 500 was obtained from YahooFinance.com[[5]](#footnote-5).

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 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. There are two reasons for this. 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 main US stock exchanges.(Investopedia, 2023). Secondly it was necessary to restrict the amount of data in anticipation of lengthy processing time for long documents.

## 3.2 Available Transcripts

Transcripts data was web scraped from The Seeking Alpha website which is a global source of financial markets data. To obtain a list of all transcripts 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 endpoint[[6]](#footnote-6). 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 (uniform resource locator) holding lists of transcript details.

The URL shows 50 JSON (JavaScript Object Notation) objects per page. Pages 1 to 1000 were available.

JSON objects are data structures which store information in key-value pairs.

Each of the JSON objects contains detailed information regarding an individual transcript. 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.

Figure 3.1 is an example of a JSON object (which is identifiable by the opening and closing brackets {}) 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 information

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

In summary it 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 data structure which will store the key-value pair data. Once this is available the data can be manipulated using the Python ‘Pandas’ data analysis and manipulation library.

Stage1\_module\_1 of the Financial Data Extraction and Processing Pipeline web scrapes this API endpoint and creates a list of available transcripts. The S&P 500 company transcripts are flagged. The pipeline is described in Chapter 5.

## 3.3 Individual transcripts

The transcripts were web scraped using the ID values in the transcript list to locate the individual S&P 500 transcripts.

Individual complete transcripts relating to each earnings conference call can be found at the following Seeking Alpha API endpoint[[7]](#footnote-7). Individual transcripts are contained in JSON objects at this URL. A web scraper was developed to navigate to this URL and extract the transcripts. The transcripts list was opened and ID values flagged as S&P 5000 were iterated through and placed in turn in the ID field of the URL.

An example is shown in Figure 3.2 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 used to locate the relevant sections for extraction.



Figure 3.2 Extract from JSON object holding individual transcript

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

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 Data Composition

Data composition by financial quarter

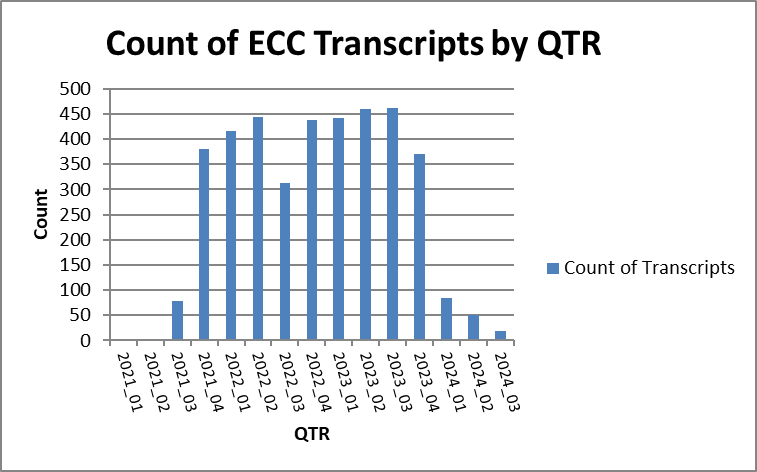


Figure 3.3 Data composition by financial quarter

Figure 3.4 Data composition by GICS sector

The Global Industry Classification Standard (GICS)[[8]](#footnote-8) is an industry classification scheme for assigning public companies to the specific economic sector and industry group that most closely 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. One use of the GICS is the comparision of companies within the same sector.

# 4 The FinBERT model

## 4.1 Testing the FinBERT model

Before using the FinBERT model its operation and accuracy was checked. This was carried out by inputting the Financial PhraseBank dataset, on which the model was fine-tuned, and comparing the model predictions. An accuracy of between 80 and 90% is expected if the model is performing as designed.

Table 4.1 shows the test results. The accuracy of 88.9% indicates that the model is performing satisfactorily allowing for model limitations, generalizations, human annotation errors and data variability.



Table 4.1 FinBERT accuracy check

## 4.2 Operation of the model

The FinBERT model was implemented in Python. The following packages were used.

1. Transformers. The transformers library by Hugging Face is used to load (1) the FinBERT model ‘BERTForSequenceClassification’ used for sequence classification tasks. (2) the BERTTokenizer, for tokenizing text inputs according to the BERT model.

2. Torch. The Python library used for deep learning. This is the main package for tensor computation and model operation.

3. Pandas. A data analysis and manipulation library. It provides datastructures such as DataFrames for working with structured data and is the main package for reading and maipulating data.

4. CSV. The Python module for reading and writing CSV files.

ECC transcripts are long documents and cannot be input to FinBERT as a complete unit. The maximum text length that can be input to a BERT model is that which will be tokenised into a maximum 510 tokens plus the special start and end tokens inserted by the model. In order to input long documents to BERT models two approaches are often used. Summarise the text or split the text into 510 token lengths. The second method, splitting the text into 510 token lengths, is used here.

# **5** Pipeline

The entire data processing from raw data collection to results correlation analysis is carried out in four main stages contained in the Financial Data Sentiment Stock Correlation Pipeline. Each stage is made up of one or more individual modules designed to carry out specific tasks. The structure of the pipeline is given in figure 5.1

## 5.1 Pipeline Structure

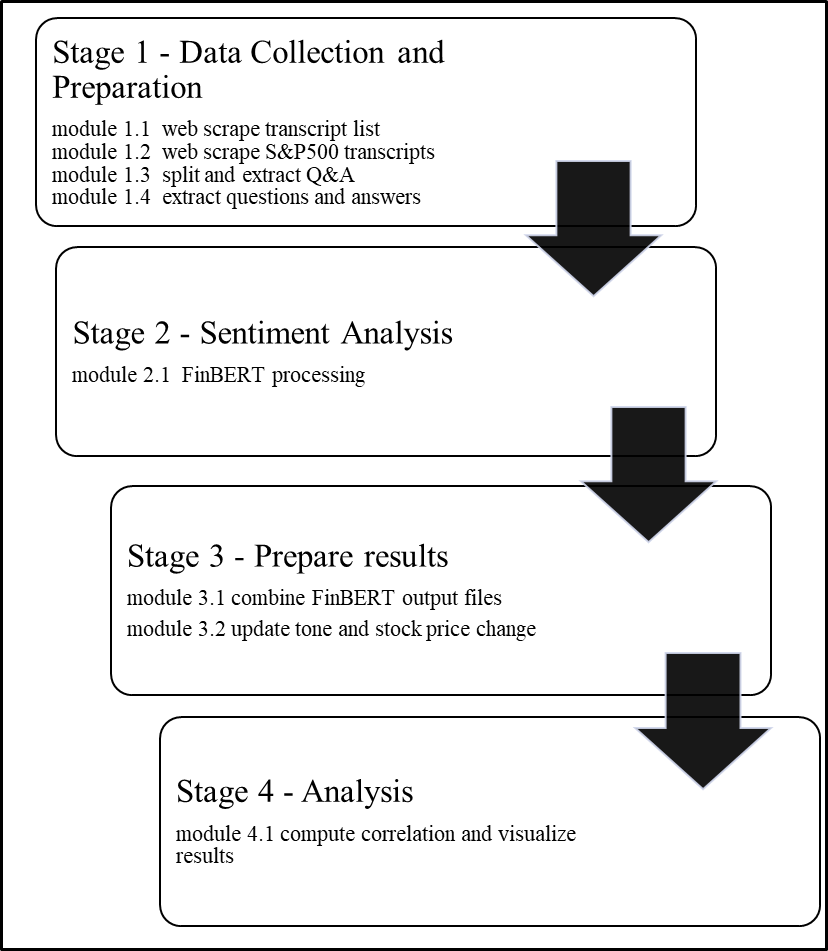


Figure 5.1 Financial Data Sentiment Stock Correlation Pipeline

Four main stages make up the pipeline with each stage containing various modules to carry out specific tasks.

An overview of the purpose of each of the modules is as follows. Detailed descriptions of each stage and module will follow.



Stage1

module 1.1

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

module 1.2

Use this list to web scrape S&P 500 companies transcripts and write them to a series of csv files.

Stage2

module 2.1

Read each of these files and extract the MD&A section and the Q&A section and from each transcript.

module 2.2

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.

Stage3

module 3.1

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

Stage4

module 4.1

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.

module 4.2

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.

Stage5

The results were computed and analysed.

## 5.2 Pipeline modules and files overview

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: web scraping the raw transcript data and preparing it for analysis by FinBERT.

A list of available transcripts on the Seeking Alpha site is first compiled which contains their unique IDs, ticker symbol and date. The transcripts of S&P 500 companies are then flagged. The seeking alpha website was again accessed to webscrape those transcripts flagged as S&P 500.

All of these complete transcripts were webscraped in HTML format and written to a series of csv files. Each of the transcripts were then split into two sections The MD&A section and the Q&A section. The Q&A section was then split into each individual question and answer.

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.

## 5.2 module 1.1 Web scraping transcript list

Purpose

This module web scrapes information relating to transcripts available on the Seeking Alpha website and creates a list consisting of transcript IDs, ticker symbols, company names, ECC dates and financial quarters. The records relating to S&P 500 companies are then flagged.

Method

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[[9]](#footnote-9) 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.

This thesis focusses on S&P 500 companies. 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.com[[10]](#footnote-10) 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 transcript details, with S&P 500 transcripts flagged, was available.

## 5.2 module 1.2 Web scraping S&P 500 transcripts

Purpose

Webscrape the individual S&P 500 transcripts form the Seeking Alpha site using the list created in Stage1\_module\_1

Method  In order to locate and save individual transcripts a second web scraper was developed. This web scraper was designed to access 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 uses Selenium WebDriver to navigate to the URL of each transcript in turn by iterating over each ID which had been flagged 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. To avoid overloading the website the WebDriver disconnected from the website for a random time delay period of between seven and nine minutes before reconnecting between each batch of fifteen transcripts. 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 this module a series of csv files containing the complete transcripts of S&P 500 companies covering the period Qtr3 2021 to Qtr1 2024 was available.

The series of csv files contained all of the web scraped transcripts in HTML format.

An extract from one of these files is shown in figure 5.4. The transcript is in column ‘transcript\_text’. It is spread over a number of rows. The date of the ECC, the company name and ticker symbol are shown as is the Seeking Alpha provided description of the transcript in the ‘text’ column.



Figure 5.4 Sample from raw data csv file

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

Purpose

This module splits the wescraped 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).

Method

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 also extracted 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 of the file with the split transcripts is shown in Figure 5.5



Figure 5.6 Sample from file containing the extracted MD&A and the Q&A sections

Method

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.

## 5.4 module 1.4 Extracting questions and answers

Purpose

extract the questions and answers separetly in plain text from file *'transcripts\_Qs\_and\_As.csv'* created in module 1.3

Method

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'.*

## 5.5 module 2.1 Applying FinBERT

-

Purpose

Apply the FinBERT model to cassify the sentiment of each question and answer. Input file 'Transcripts\_Qs\_and\_As\_Split.csv'.

Method

Input of the transcripts to FinBERT.

The questions and answers extracted from the Q&A sections of the ECC transcripts were contained in a series of csv files. Each file contained the individual questions and answers on separate rows relation to a number (~100?) of transcripts. These files were input to the FinBERT model which classified the sentiment each question and answer as 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 input file was returned with four new columns, sentiment, count of +ve, count of -ve and count of neutral.

To be completed (BH 19 Aug)

## 5.6 module 3.1 Combine FinBERT output files

Purpose

Combine the set of output files from FinBERT.

Method

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 figure 5.7

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.



Figure 5.7 Output file from sentiment analysis processing

A number of columns were added to the file as follows

**To be completed (BH 19 Aug)**

## 5.7 module 3.2 Compute tone, stock price change

Purpose

Compute tone and stock price changes in respect of each transcript.

Method

A measure for the (linguist) tone of each question and answer was defined following the method of Brockman et al (Brockman, Li and Price, 2015). 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 change.**

Stock price data is contained in folder ‘Stock\_Data\_Files’. This folder contains a file with historical stock price data for each of the S&P 500 companies. The historical stock data was webscraped from YahooFinance.com[[11]](#footnote-11).

The files are named for the ticker symbol of each company, 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 conjunction with the results file.

Based on the ticker and date fields the average close price over the 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 centred on the ECC date. The same process was repeated for the two-day and one-day 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.

## 5.8 module 4.1 Results correlation and visualization

# **6 Sentiment Analysis Results**

## 6.1 Results file

Stage 3 of the pipeline aggregrates the output files from the model and carries out the tone and price change computations. This process produces a file containing the results: ‘updated\_combined\_file\_with\_compatible\_date .csv’. An extract form this file is shown in Fig 6.1



Figure 6.1 Extract from results file

Explanation of the fields in the results file.

|  |  |
| --- | --- |
| **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 6.2 Explanation of fields in the results file

## 6.2 Tone

Overall tone



Table 6.1 Tone of transcripts sections

The overall tone of the ECC transcript sections is given in Table 6.1 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**

Average tone by quarter.



Table 6.2 Average tone by quarter

The average tone by financial quarter is shown in Table 6.2

**Average tone by GICS sector**



Table 6.3 Average tone by GICS sector

Processing on Kaggle.

To be done (BH 19 Aug)

# 7 Cprrelation 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 6.3. 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 6.3 Pearson correlation coefficient (r) formula

A Pearson correlation function ‘pearsonr’ is available in Python library ‘Scipy’. This function 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.



Table 6.4 Correlation of Tone and price change

Table 6.4 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 Brockman et al (Brockman, Li and Price, 2015) .

Questions: tone and price change correlation



Table 6.5 Correlation of questions tone and stock price changes

Correlation matrix: questions tone and stock price changes

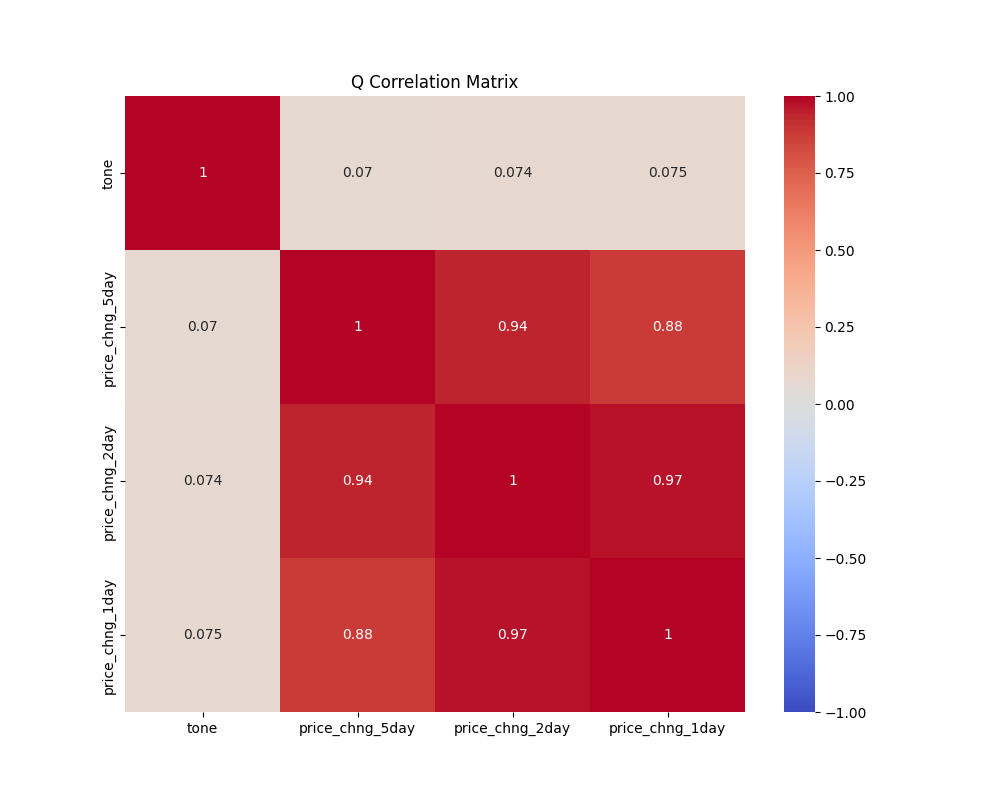


Figure 6. 4 Correlation matrix: questions tone and stock price changes

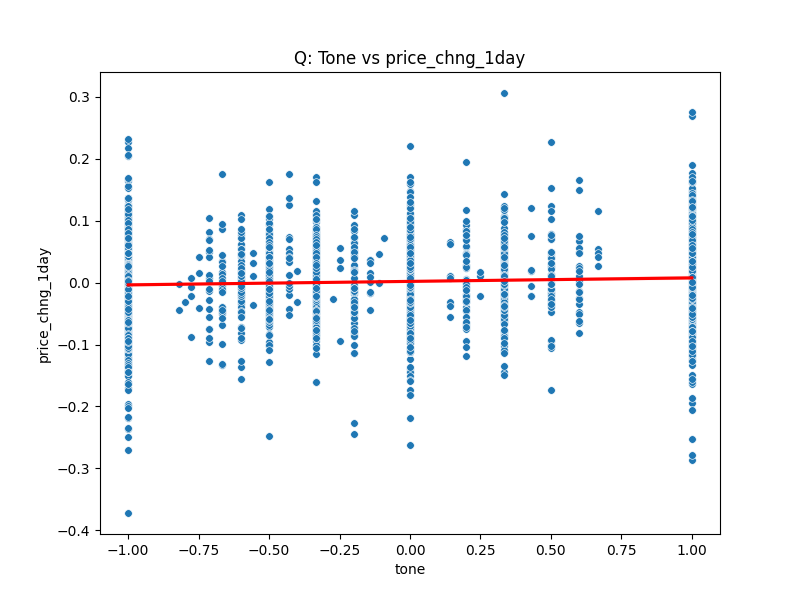


Figure 6.5 Scatter plot of questions tone vs 1-day stock price change

The trendline is shown in red.

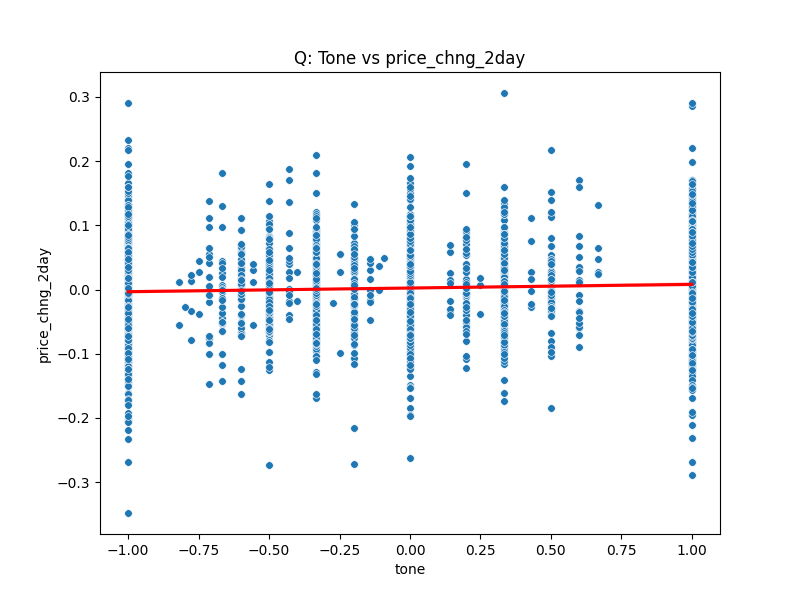


Figure 6.6 Scatter plot of questions tone vs 2-day stock price change

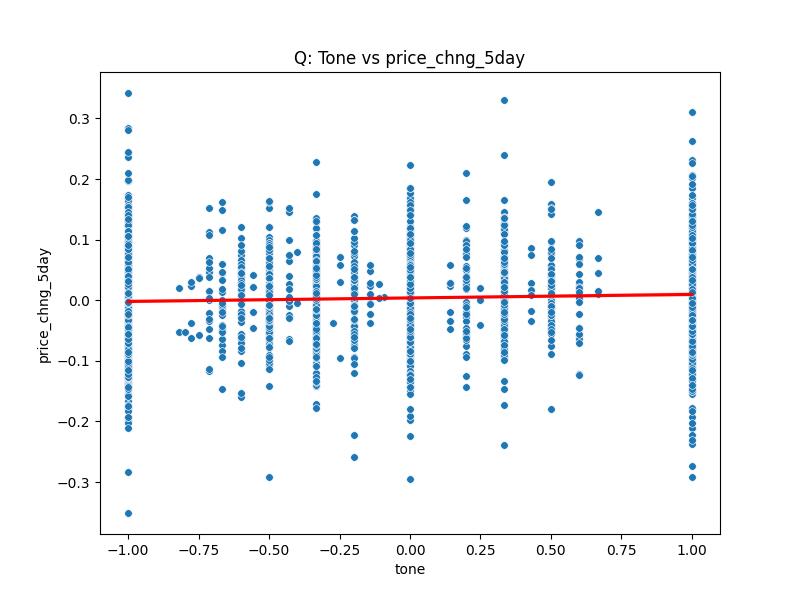


Figure 6.7 Scatter plot of questions tone vs 5-day stock price change

Answers: tone and price change correlation



Table 6.6 Correlation of answers tone and stock price changes

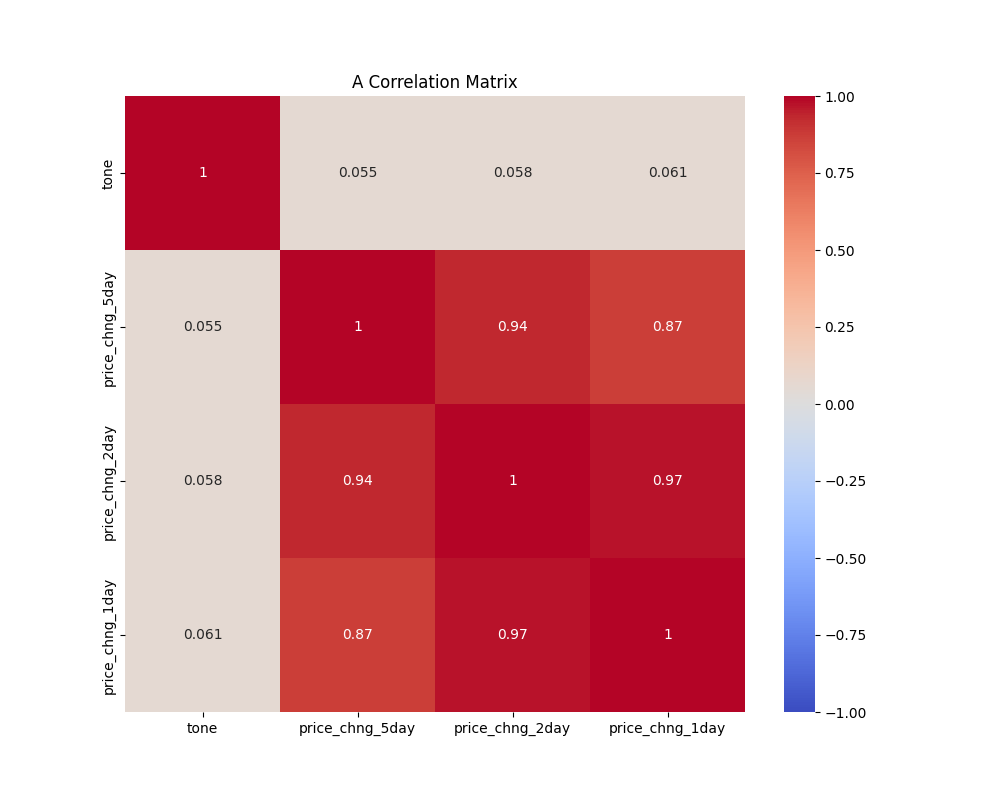


Figure 6.8 Correlation matrix: Answers tone and stock price movements

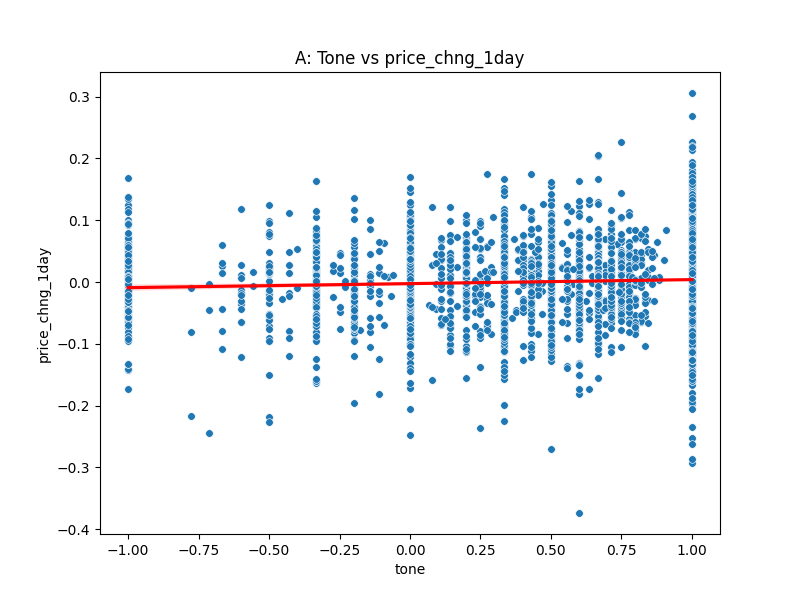


Figure 6.9 Scatter plot of answers tone vs 1-day stock price movement

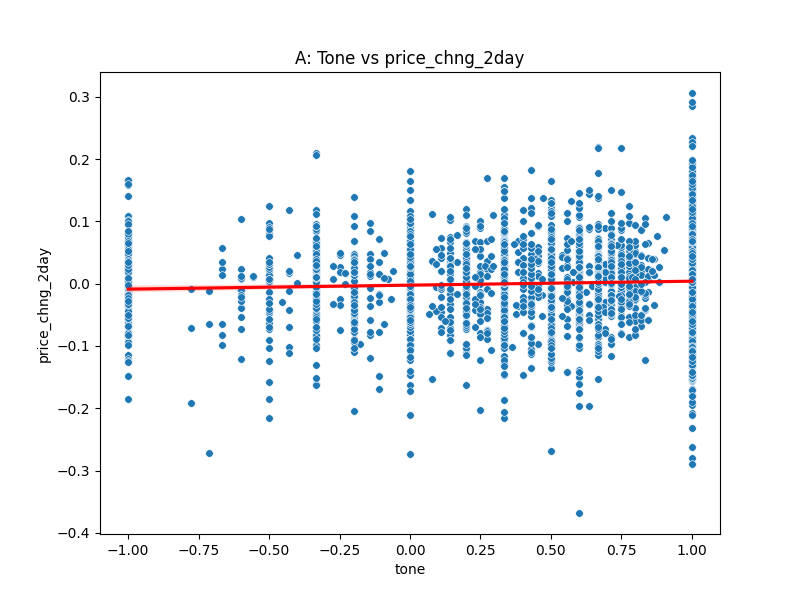


Figure 6.10 Scatter plot of answers tone vs 2-day stock price movement

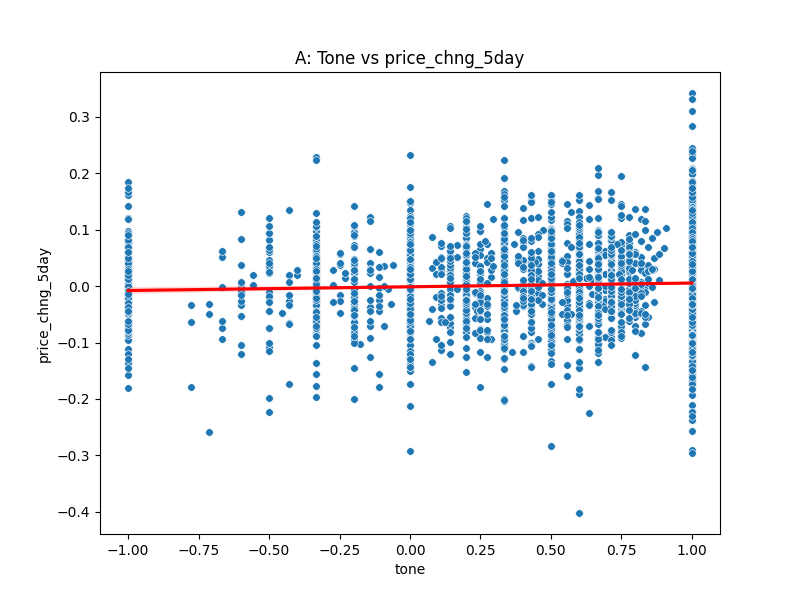


Figure 6.11 Scatter plot of answers tone vs 5-day stock price movement

Q&A section overall: tone and price change correlation



Table 6.7 Corelation of Q&A section tone and stock price changes

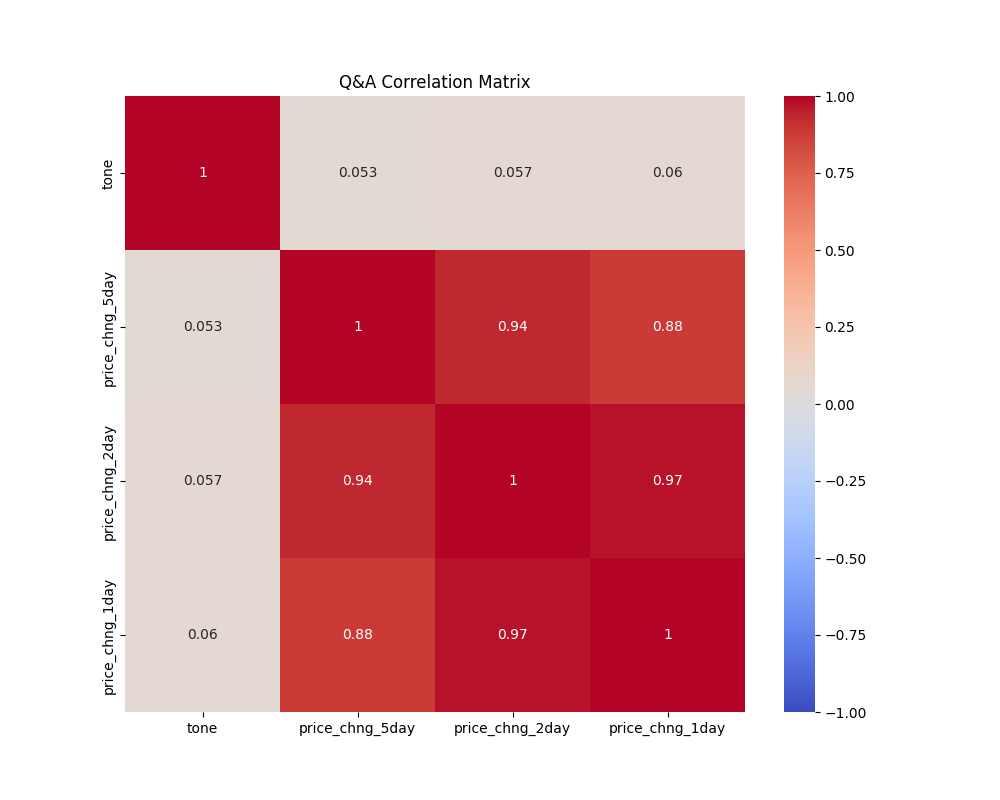


Figure 6.12 Correlation matrix: Q&A section tone and stock price changes

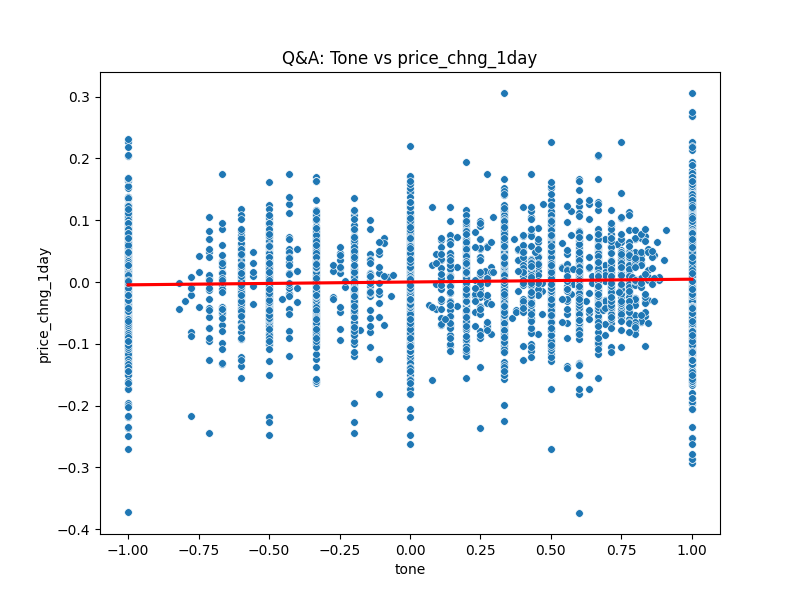


Figure 6.13 Scatter plot of Q&A section tone vs 1-day stock price change

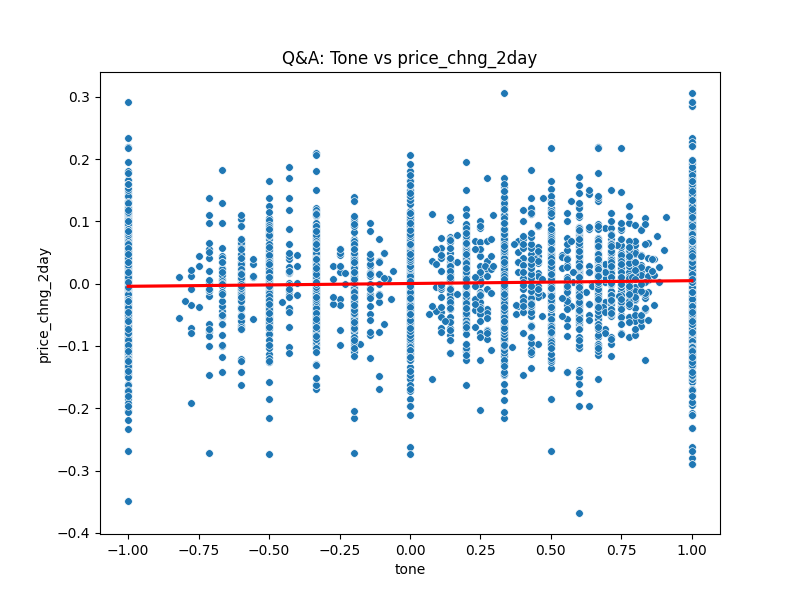


Figure 6.14 Scatter plot of Q&A section tone vs 2-day stock price change

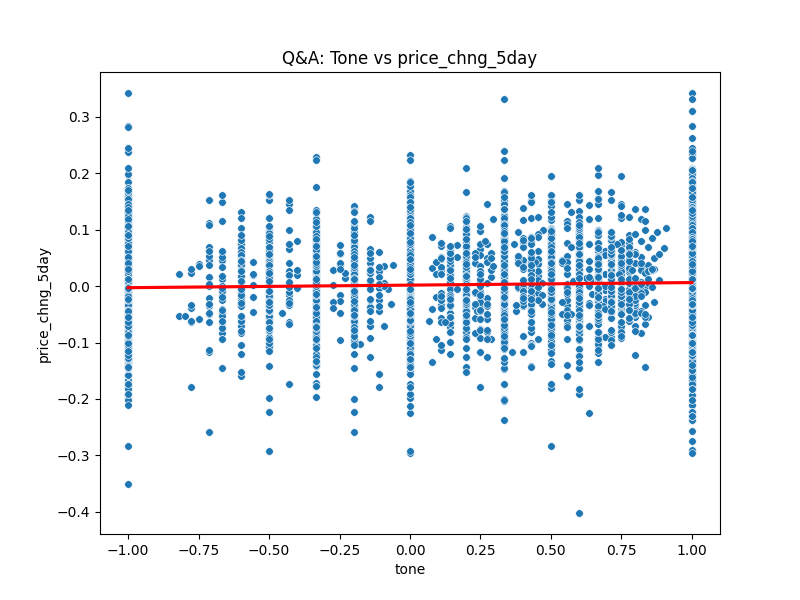


Figure 6.15 Scatter plot of Q&A section tone vs 5-day price change

# **8 Conclusions and Future Work**

The results indicate a weak positive correlation between the tone of the Q&A section of earnings conferences and subsequent stock price changes over a five-day period. The strongest correlation is between the tone of the analysts and investors (questions) and the next day price changes. The tone of analysts and investors was found to be negative. This is in contrast to the positive tone of the company (answers). This may indicate that the questioners were not in full agreement with the company view of its own performance.

The data set used in this research was small. A dataset covering a period of the order of ten years with ten thousand transcripts would provide a more accurate

*(To be completed)*

# **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.

Fama, E.F. (1970) ‘Efficient Capital Markets: A Review of Theory and Empirical Work’, *The Journal of Finance*, 25(2), p. 383. Available at: https://doi.org/10.2307/2325486.

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.

Lo, A.W. (2004) ‘The Adaptive Markets Hypothesis’, *The Journal of Portfolio Management*, 30(5), pp. 15–29. Available at: https://doi.org/10.3905/jpm.2004.442611.

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.

Picasso, A. *et al.* (2019) ‘Technical analysis and sentiment embeddings for market trend prediction’, *Expert Systems with Applications*, 135, pp. 60–70. Available at: https://doi.org/10.1016/j.eswa.2019.06.014.

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.

# **Appendices**

## Appendix 1 : Code – Implementation of FinBERT

*# Implementation of FinBERT  
"""  
Description  
This module is run on Kaggle making use of the GPU P100 accelerator  
Input 'Transcripts\_Questions\_Answers\_Test\_cleaned\_7.csv'  
the file contains the plain text questions and answers extracted  
from Earnings Conference Call transcripts.  
output file: 'earnings\_calls\_Kaggle\_Test\_q\_a.csv' The output consists of the input returned with  
each question and answer (row) given sentiment classification scores.   
Each Q and A can be inspected   
Processor: (Kaggle) GPU P100 accelerator.  
Total processing time for entire input is approx 11 hours  
  
"""***from** transformers **import** BertForSequenceClassification, BertTokenizer  
**import** torch  
**import** pandas **as** pd  
**import** csv  
  
classes = {0:**'positive'**, 1:**'negative'**, 2:**'neutral'**}  
  
tokenizer = BertTokenizer.from\_pretrained(**'ProsusAI/finbert'**)  
  
model = BertForSequenceClassification.from\_pretrained(**'ProsusAI/finbert'**)  
  
  
**def** text\_processing(text):  
 txt = text  
 tokens = tokenizer.encode\_plus(txt, add\_special\_tokens=**False**)  
 input\_ids, token\_type\_ids, attention\_mask = tokens[**'input\_ids'**], tokens[**'token\_type\_ids'**], tokens[**'attention\_mask'**]  
 total\_len = len(tokens[**'input\_ids'**])  
 **return** input\_ids, attention\_mask, total\_len  
 tokens.keys()  
*# tokens.keys()***def** chunk\_text\_to\_window\_size\_and\_predict\_proba(input\_ids, attention\_mask,  
 total\_len):  
 *"""  
 This function splits the given input text into chunks of a specified window length,  
 applies transformer model to each chunk and computes probabilities of each class for each chunk.  
 The computed probabilities are then appended to a list.  
  
 Args:  
 input\_ids (List[int]): List of token ids representing the input text.  
 attention\_mask (List[int]): List of attention masks corresponding to input\_ids.  
 total\_len (int): Total length of the input\_ids.  
  
 Returns:  
 proba\_list (List[torch.Tensor]): List of probability tensors for each chunk.  
 """* proba\_list = []  
  
 start = 0  
 window\_length = 510  
  
 loop = **True  
 while** loop:  
 end = start + window\_length  
 *# If the end index exceeds total length, set the flag to False and adjust the end index* **if** end >= total\_len:  
 loop = **False** end = total\_len  
  
 *# 1 => Define the text chunk* input\_ids\_chunk = input\_ids[start: end]  
 attention\_mask\_chunk = attention\_mask[start: end]  
  
 *# 2 => Append [CLS] and [SEP]* input\_ids\_chunk = [101] + input\_ids\_chunk + [102]  
 attention\_mask\_chunk = [1] + attention\_mask\_chunk + [1]  
  
 *# 3 Convert regular python list to Pytorch Tensor* input\_dict = {  
 **'input\_ids'**: torch.Tensor([input\_ids\_chunk]).long(),  
 **'attention\_mask'**: torch.Tensor([attention\_mask\_chunk]).int()  
 }  
  
 outputs = model(\*\*input\_dict)  
 probabilities = torch.nn.functional.softmax(outputs[0], dim=-1)  
 proba\_list.append(probabilities)  
 start = end  
  
 **return** proba\_list  
  
 *# proba\_list = chunk\_text\_to\_window\_size\_and\_predict\_proba(input\_ids, attention\_mask, total\_len)  
 # print("This is the 'proba' list:", proba\_list)***def** get\_mean\_from\_proba(proba\_list):  
 *"""  
 This function computes the mean probabilities of class predictions over all the chunks.  
  
 Args:  
 proba\_list (List[torch.Tensor]): List of probability tensors for each chunk.  
  
 Returns:  
 mean (torch.Tensor): Mean of the probabilities across all chunks.  
 """  
  
 # Ensures that gradients are not computed, saving memory* **with** torch.no\_grad():  
 *# Stack the list of tensors into a single tensor* stacks = torch.stack(proba\_list)  
  
 *# Resize the tensor to match the dimensions needed for mean computation* stacks = stacks.resize(stacks.shape[0], stacks.shape[2])  
 *# print("This is 'stacks':", stacks) #BH Tue 16Apr 00.27  
  
 # Compute the mean along the zeroth dimension (i.e., the chunk dimension)* mean = stacks.mean(dim=0)  
  
 **return** mean  
  
*# mean = get\_mean\_from\_proba(proba\_list)  
# tensor([0.0767, 0.1188, 0.8045])  
  
# torch.argmax(mean).item()  
# mean*output\_filename = **'./earnings\_calls\_Kaggle\_Test\_q\_a.csv'**df = pd.read\_csv(  
 **'/kaggle/input/transcripts-questions-answers-test-cleaned-7-csv/Transcripts\_Questions\_Answers\_Test\_cleaned\_7.csv'**,  
 encoding=**'utf-8'**)  
**with** open(output\_filename, **'w'**, newline=**''**, encoding=**'utf-8'**) **as** csvfile:  
 csv\_writer = csv.writer(csvfile)  
  
 csv\_writer.writerow(  
 [**'ID'**, **'Company\_AName'**, **'Ticker'**, **'Text'**, **'Call\_Section'**,  
 **'Transcript\_Text'**, **'sentiment'**, **'count of +ve'**, **'count of -ve'**,  
 **'count of neutral'**])  
  
 *# Iterate over each row in the DataFrame* **for** i, row **in** df.iterrows():  
 *# Extract relevant information from the current row* ID = row[**'ID'**]  
 Company\_AName = row[**'Company\_AName'**]  
 Ticker = row[**'Ticker'**]  
 Text = row[**'Text'**]  
 Call\_Section = row[**'Call\_Section'**]  
 Transcript\_Text = row[**'Transcript\_Text'**]  
 *# sentiment = row['sentiment']  
  
 # Perform sentiment analysis on the text to get FinBERT sentiment  
 # input\_ids, attention\_mask, total\_len = text\_processing(text)* input\_ids, attention\_mask, total\_len = text\_processing(Transcript\_Text)  
  
 proba\_list = chunk\_text\_to\_window\_size\_and\_predict\_proba(input\_ids,  
 attention\_mask,  
 total\_len)  
 mean = get\_mean\_from\_proba(proba\_list)  
 result\_class = classes[torch.argmax(mean).item()]  
 *# print("This is the 'proba\_list':",proba\_list)   
 # Count of probability classes per line.  
  
 # Initialize counters* count\_positive = 0  
 count\_negative = 0  
 count\_neutral = 0  
  
 *# Count the occurrences of each class* **for** prob **in** proba\_list:  
 pred\_class = torch.argmax(prob).item()  
 **if** pred\_class == 0:  
 count\_positive += 1  
 **elif** pred\_class == 1:  
 count\_negative += 1  
 **elif** pred\_class == 2:  
 count\_neutral += 1  
  
 *# Write the processed row to the output CSV file* csv\_writer.writerow(  
 [ID, Company\_AName, Ticker, Text, Call\_Section, Transcript\_Text,  
 result\_class, count\_positive, count\_negative, count\_neutral])  
  
# End

1. https://www.microsoft.com/en-us/investor [↑](#footnote-ref-1)
2. https://huggingface.co/ProsusAI/finbert? [↑](#footnote-ref-2)
3. https://www.researchgate.net/publication/251231364\_FinancialPhraseBank-v10 [↑](#footnote-ref-3)
4. https://seekingalpha.com/ [↑](#footnote-ref-4)
5. https://finance.yahoo.com/ [↑](#footnote-ref-5)
6. *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* [↑](#footnote-ref-6)
7. https://seekingalpha.com/api/v3/articles/4635802?include=author%2CprimaryTickers%2CsecondaryTickers%2CotherTags%2Cpresentations%2Cpresentations.slides%2Cauthor.authorResearch%2Cauthor.userBioTags%2Cco\_authors%2CpromotedService%2Csentiments [↑](#footnote-ref-7)
8. https://www.investopedia.com/terms/g/gics.asp# [↑](#footnote-ref-8)
9. https://www.selenium.dev/documentation/webdriver/ [↑](#footnote-ref-9)
10. <https://stockanalysis.com/list/sp-500-stocks/> [↑](#footnote-ref-10)
11. https://finance.yahoo.com/ [↑](#footnote-ref-11)