Exploring the Relationship Between Financial Earnings Call Sentiments and Stock Returns

Nikita Tcherevik and Colin Daniel

Department of Statistics and Data Science, Carnegie Mellon University

36-668: Special Topics - Text Analysis

Professor David Brown

12-12-2022

Executive Summary

Purpose of the Research Study

The purpose of this study is to highlight potential relationships between the positive and negative sentiment scores in earnings calls and stock returns in the month leading up to a call and the month after. We aim to gather useful insight in the process of analyzing the recent financial performance of a company.

Data Collection and Methodology

Our data is collected from a number of sources with financial data, namely Alpaca for stock prices, Yfinance for company metadata, and The Motley Fool for earnings call transcripts. Data for 478 companies was scraped from the sources and were consolidated into a csv file for exploratory data analysis and modeling. We utilized the finBERT model to calculate sentiment scores based on the earnings call transcripts from The Motley Fool. We built four multilevel linear regression models, two for predicting the monthly returns a month after the earnings call based on positive or negative sentiment, as well as two models for predicting positive or negative sentiment, based on the monthly returns one month prior to the earnings call.

Overview of Findings

We saw that the coefficient for negative sentiment is a significant predictor of monthly returns one month after the earnings call, while the positive sentiment score coefficient was not significant in its respective model. A similar relationship is seen in the other two models predicting sentiment based on returns. The coefficients are only significant in the model used to predict negative sentiment and not significant in the model used to predict positive sentiment. Overall, there is evidently a relationship between negative sentiment and stock returns.

Next Steps

While there is useful insight about the association between sentiment and stock returns in this study, we have the following recommendations for eliminating limitations in further research:

1. Improve computing power to handle a larger dataset of companies.
2. Increase the number of secondary features in the model to account for omitted variable bias and limit noise.
3. Explore a smaller subset of the earnings call transcripts themselves, namely from the question and answer section only to prevent diluting true company sentiment.

Introduction

This study aims to uncover any potential relationships or associations between the overall negative and positive sentiments in the language of an earnings call and the stock returns. Earnings reports have long been used as important tools for investors and stock analysts as a way to determine the financial health and performance of a company to make informed decisions over time. Results of these reports can have massive implications for the company in the long and short term: poor performance potentially resulting in widespread sell-off of the stock, strong performance resulting in subsequent growth and success (Tuovila 2022). This report will attempt to answer the following questions:

1. Is there an association between sentiment in an earnings call and its stock returns in the month following?

2. Is there an association between the monthly returns of a company leading up to the earnings call and the sentiment portrayed during the call?

Prior to conducting our analysis, we had the following hypotheses respectively for each question:

1. There is a positive association between sentiment of a company and its stock returns in the month following an earnings call. We expect to see stronger returns for companies that have more positive sentiment, and weaker returns for companies with more negative sentiment.

2. Similarly, we also expect to see a positive association between the sentiment of a call and the monthly returns leading up to the call.

In order to answer these questions, we will be utilizing over 400 different earnings call transcripts and calculating sentiment scores for them, along with meta-data corresponding to company characteristics and their returns. We believe that this data is relevant as it is focused directly on the variables of interest stated in our questions.

Data

Data Compilation

Our dataset for the modeling portion of this report is stored in a CSV file that we compiled with data gathered from various sources, outlined in the subsequent sections. Overall, our file consists of 476 rows, each representative of one publicly traded company. The metadata for these companies comes from Q2 and Q3 of 2022. Relevant features in this file are: *posPer, negPer,* *mb_returns, mf_returns, Sector, beta* and *mktcap*. These variables were extracted and merged in a Python module, writing the data frame to a csv file for further analysis. The features were pulled from the following sources.

The Motley Fool Earnings Reports

For gathering earnings calls, the data was pulled from publicly available transcripts on The Motley Fool (The Motley Fool, 2022). The scraping process was done using a Python module and the BeautifulSoup package to easily parse through the desired text (Richardson, 2020). We were forced to clean some of the text to make sure that the transcript was representative of the earnings call only. For example, we had to remove an advertisement for The Motley Fool that was at the end of the transcript in each of the companies' reports. The composition of the corpus is shown below:

Corpus	Total	Tokens
Earnings Call Transcrips	**480**	**4077724**

Table 1: Composition of the Corpus

Afterward, we conducted sentiment analysis on the data, which left us with two main variables of interest:

posPer: The percent of total sentences that were deemed positive within the earnings call transcript.

negPer: The percent of total sentences that were deemed negative within the earnings call transcript.

Alpaca Stock Data

We used the Alpaca API to pull relevant closing stock prices on the day one month prior to the earnings call, the day of the earnings call and one month after the report for each of the 476 companies (Alpaca, 2022). Using these stock prices, we calculated the two following variables for our analysis:

mb_returns: The monthly returns of the stock price month after the earnings call. Given by the formula: *(Price on day of call - Price month before call)/(Price month before call) X 100.* Note that the unit of this variable is percentage.

mf_returns: The monthly returns of the stock price a month after the earnings call. Given by the formula: *[(Price After a Month - Price on day of call)/(Price on day of call)] X 100*. Note that the unit of this variable is percentage.

Yfinance Metadata

We used the Yfinance library to access the Yahoo Finance API to extract metadata for the 476 stocks (Aroussi, 2022). For our dataset specifically, the variables include:

Sector: 11 levels consisting of Healthcare, Financial Services, Technology, Real Estate, etc.
Beta: Measure of the systematic risk of the security compared to the overall market.
Market Cap: Total value of the company, calculated by multiplying the total shares of the stock by the current stock price. The unit of measure for this variable is in dollars.

These secondary variables are used to help alleviate the noise that is present in the model.

Exploratory Data Analysis

Below, we conducted basic exploratory data analysis on the distributions of our main factors in the dataset. The four figures are histograms of the percentage of negative sentences in the calls, percentage of positive sentences in the calls, returns one month after the earnings call and the returns from one month before the earnings call, respectively.

Figure 1: Distribution of Negative Sentences

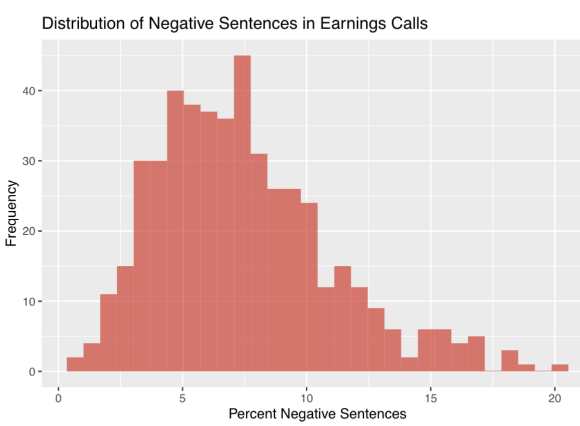
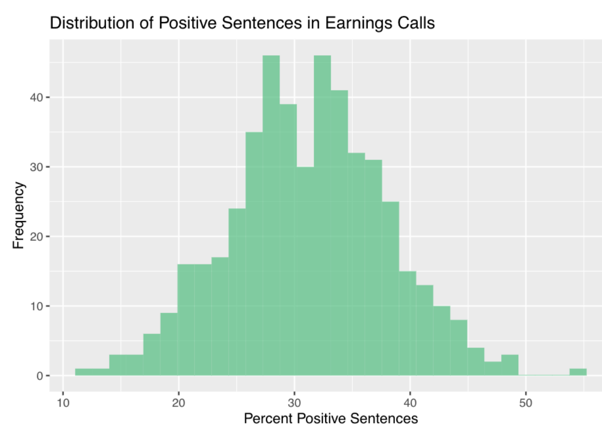
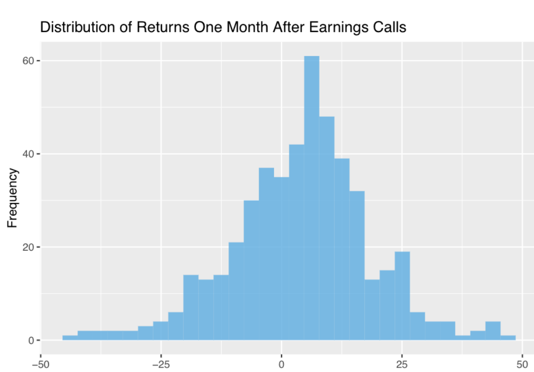


Figure 2: Distribution of Positive Sentences





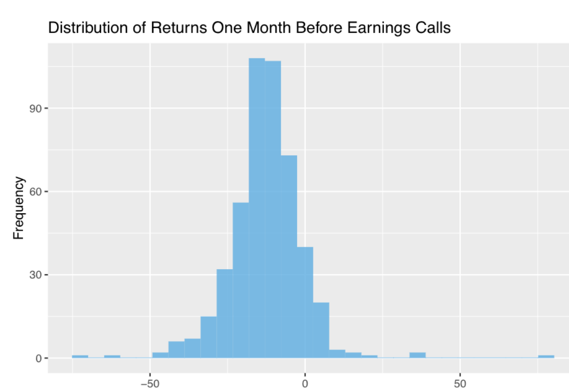


Figure 3: Distribution of Returns One Month After **Figure 4:** Distribution of Returns One Month Prior

Figure 1 above seems to be slightly skewed to the right, while figure 2 appears to be distributed approximately normal. As for Figure 3, the returns one month after the call seem to be skewed ever so slightly to the left, while figure 4 appears to be somewhat normal, with potential outliers. Some of these outliers can be attributed to reverse stock-splits as was the case with NLY, or just periods of really strong growth or decline, such as ISEE and PANW (Kiesche, 2022). Since reverse stock-splits are not truly representative of the growth of the stock, we opted to remove NLY from our data. To limit complexity, we opted to keep the distributions the same and not pursue a Box-Cox transformation for any of the variables. This will keep coefficient interpretation simple.

Within our exploratory data analysis, we also found that some companies had missing data for their beta values and market caps. To address this, we removed companies with these missing values from our dataset, as no proper substitutions can be made to include them in our regressions. This left us with 427 observations.

Methods

Sentiment Analysis

In order to conduct the sentiment analysis, we used finBERT in Python, which is a Bidirectional Encoder Representations from Transformers model with a focus on financial text. We used this model for one main reason, the specific focus on financial literature in its training (Huang, 2022). In our early analysis, it was evident that conventional sentiment analysis through Syuzhet in R was not really applicable, as positive and negative sentiment in financial reports are represented in different words than in normal conversation. On the other hand, finBERT is trained on earnings call transcripts and analyst reports, so it is more suited for the task.

finBERT iterates through each sentence of a given list and classifies the sentence under three different labels: positive, negative and neutral. In order to utilize this model, we iterated through each earnings call transcript and separated it into sentences for preparation in model ingestion. Once the data is prepared, we feed the separated call into the model to get a total count of each positive, negative and neutral label. From there we use the percentages of positive and negative sentences as a proxy for the overall sentiment of an earnings call.

Relationships between Sentiments and Returns

To evaluate the relationships between sentiment and the returns of a stock, we will be using regression analysis techniques, and then analyzing the coefficients and their statistical significance. These regressions should help quantify a relationship between the returns of a stock and the sentiment of its earnings calls and let us deduce whether an association exists between the different metrics.

In order to analyze the relationships between sentiment and returns a month after the call, we will be utilizing multi-level linear regressions from the lme4 package in R, grouped by the sectors of each respective company (Bates, 2022). The general hierarchical model formula is shown below, note that $x_{sentiment}$ represents either positive or negative sentiment, depending on the specific model.

$$i = 1 \dots N \ companies, \ j = 1 \dots J \ sectors$$

$$returns_i = \alpha_{0,j[x[i]]} + \beta_1 x_{sentiment[i]} + \beta_2 x_{mktcap[i]} + \beta_3 x_{beta[i]} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

$$\alpha_{0,j[x[i]]} = \beta_0 + \eta_j, \eta_j \sim N(0, \tau^2)$$

We will be utilizing a similar model for the relationship between sentiment and the returns leading up to an earnings call. The key differences are that the sentiment scores will be the dependent variable, and the returns leading up to the call are the prediction variables, and there are no secondary variables included. The models will have the following structure:

$$i = 1 \dots N \ companies, \ j = 1 \dots J \ sectors$$

$$sentiment_i = \alpha_{0,j[x[i]]} + \beta_1 x_{returns[i]} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

$$\alpha_{0,j[x[i]]} = \beta_0 + \eta_j, \eta_j \sim N(0, \tau^2)$$

Ultimately, we decided to use hierarchical models for our analysis because we noted that the means of returns and sentiment differed significantly across different sectors. This regression method should help capture some of that variation between the groups. Furthermore, note the secondary variables used in the regression for forward returns but not backward returns. The reasoning for this is to account for omitted variable bias in returns, as some of the variables might play a role in a stock’s returns. For example, the Fama-French model suggests that market capitalization plays a role in a stock’s returns, and beta is a factor related to how correlated a stock’s price is with market movements (Hayes, 2022). There is no reasoning to suggest that these same factors play a role in the sentiment of an earnings call, and our exploratory data analysis in the technical appendix supports this notion. The exclusion of these variables doesn’t mean that there aren't omitted variables in the second set of models, just that the necessary omitted variables haven’t been identified yet.

Results

Relationship between forward stock returns and sentiment

Sentiment	Coefficient	SE	T.Value	lower.CI	upper.CI
Positive Sentiment	0.1436	0.1009	1.787	-0.0542	0.3414
Negative Sentiment	-0.4166	0.1969	-4.533	-0.8025	-0.0307

Table 2: Coefficients of sentiment scores when regressing on returns in month after earnings call. Measures of uncertainty are also included

In Table 2 above, the results of the two regressions run on the returns in the month after are shown. From the coefficients, it is evident that for every additional percent of sentences scored as a positive sentiment, the monthly returns are expected to increase 0.1436 in percentage points, on average. Similarly, for every additional percent of sentences scored as a negative sentiment, the monthly returns were expected to decrease 0.4166 in percentage points, on average. However, note that from the T-Values and confidence intervals, only the negative sentiment coefficient is statistically significant.

While we are not including the diagnostic plots in the contents of this report for brevity, note that they have been satisfied. In particular, the marginal and conditional residuals are centered around 0. While some skew exists in the distributions of the residuals and the variances of means, it is not significant enough to disqualify the model. Please refer to the technical appendix in order to see how to generate these diagnostic plots.

Relationship between backward stock returns and sentiment

Sentiment	Coefficient	SE	T.Value	lower.CI	upper.CI
Positive Sentiment	0.0489	0.0274	1.787	-0.004800	0.1026
Negative Sentiment	-0.0616	0.0136	-4.533	-0.088256	-0.0349

Table 3: Coefficients of returns leading up to earning call when regressing on positive and negative sentiment scores. Measures of uncertainty are also included

In Table 3 above, the results of the two regressions run on the sentiment scores of earnings calls are shown above. From the results, it is clear that for every 1% increase in the returns of a stock leading up to an earnings call, the percent of positively-scored sentences is expected to increase 0.0489 percentage points on average. Similarly, note that for every 1% increase in the monthly returns of a stock, the percent of negatively-scored sentences is expected to decrease 0.0616 percentage points on average. Note that related to the means of percent of positive sentences and negative scores, which are 30.87% and 7.58% respectively, these changes are marginal.

However, when considering the wide range of returns, this can still account for some significant variation in the sentiment presented in earnings calls. In these regression models, note that the coefficient in the model for positive sentiment is not statistically significant, but the coefficient in the model for negative sentiment is significant.

When checking the diagnostics of these models, we found that they were satisfactory. The marginal residuals and conditional residuals are centered around 0. Furthermore, the distribution of the residuals and variance of means are satisfactory. Please refer to the technical appendix for how to reproduce these plots.

Discussion

Associations between relative sentiment and month forward returns

To summarize our findings, by examining the coefficients of hierarchical linear regressions, we find that there is no significant association between how positive a company is during its earnings call, and the returns that its stock experiences in the subsequent month. On the other hand, we do find that there is a significant association between the negativity of an earnings call, and the returns in the month after. Notice, given average growth rates of a stock, these changes are quite significant. As a point of comparison, between the years of 1980 and 2019 the S&P 500 has an average monthly return of 0.67% (Gunnars, 2019). This partially confirms our hypothesis that we listed in the introduction of this report.

After some thought, these results do make sense. Executives are expected to be positive during earnings calls, so it isn't out of the ordinary to see a lot of positive statements in the transcript. Therefore, investors probably don't put a lot of stock into positivity. On the other hand, when negativity arises during the call, it can be seen as a red flag to investors. The subsequent transactions in the market would cause a fall in the price of a stock.

Associations between relative sentiment and month backward returns

To summarize the results of the models used to address this question, we find that there is no statistically significant relationship between the monthly returns leading up to an earnings call and the positive sentiment of a company's earnings call. However, we do see that a significant relationship exists between the returns and the negative sentiment of an earnings call. It is important to note that while a statistically significant relationship exists, the effect that monthly returns leading up to the earnings call is marginal, as for each 1% decrease in returns, the negative sentiment percentage is expected to only increase by 0.0616. Once again, this only partially confirms our hypothesis.

From our understanding, positive sentiment doesn't have a relationship with the returns leading up to it because executives have a predetermined amount of positive things that they will say about the company, and furthermore there is only so much they can say about the growth of a stock. On the other hand, the small magnitude of change can be attributed to the fact that executives will address the current negative returns of a company stock, but only sparingly. Other negative scores can be attributed to future projections and other matters not directly related to the stock

Limitations of Data

While this study has provided some insight on the hypotheses laid out in this research report, it is important to note various limitations in the study. Firstly, computation power in the time frame of this assignment. If we had more time, we could run the model on as many companies as we wanted, however, the finBERT sentiment analysis takes roughly 30-45 seconds per company, thus constraining us to less than 500 companies, so it would not take days to run the analysis. Building off of this, we are confident that a more accurate sentiment score could be obtained. Our sentiment scores do not capture the magnitude of the negativity or positivity in the sentence, but merely the percentage of negative or positive sentences. Finding a way to capture the severity might help us get a more accurate prediction on the return. We also think that our analysis is limited by the amount of secondary features in the model. We believe having more metadata for each of the stocks would help to reduce more of the noise and uncertainty in the models. In reality, there are effectively infinite variables that go into stock movement and returns, so it is difficult to have enough metadata to truly eliminate omitted variable bias.

Next Steps

While there are some notable limitations, this also opens the door for more projects and improvements in the future. As noted, it would be interesting to explore an extension of the finBERT sentiment score calculation to try to capture the severity of the negativity and positivity as opposed to the absolute number of sentences or percentage of sentences that are a particular sentiment. Another area of improvement could be in the corpus selection itself. It may be less representative to pull sentiment from the entire report, where it is flooded with prepared remarks. It is entirely possible that the companies are presenting the information in a way that may be sugarcoating their true health and performance so they do not discourage investors. As a result, our sentiment scores may be a bit understated. To account for this, we could try analyzing only the question and answer section of the earnings call where they are being asked more specific questions about the financials of the business. We believe we would be able to extract more valuable and representative sentiment from this section of the report as opposed to the call as a whole.

References

Alpaca API (2022) from https://alpaca.markets/docs/api-references/market-data-api/

Aroussi, Ran (2022), *Download market data from Yahoo! Finance's API,* from https://pypi.org/project/yfinance/

Bates, Douglas (2022), *Package: lme4,* from https://cran.r-project.org/web/packages/lme4/lme4.pdf

Gunnars, Kris (2019) "What Are the Average Stock Market Returns by Month?" *Stock Analysis*. from https://stockanalysis.com/article/average-monthly-stock-returns/

Hayes, Adam. (2022). "Fama and French Three Factor Model Definition: Formula and Interpretation." *Investopedia*. from https://www.investopedia.com/terms/f/famaandfrenchthreefactormodel.asp

Huang, Allen H., Hui Wang, and Yi Yang. (2022)"FinBERT: A Large Language Model for Extracting Information from Financial Text." *Contemporary Accounting Research* .

Kiesche, Liz. (2022) "Annaly Capital Management to Implement 1-for-4 Reverse Stock Split (NYSE:NLY)." *SeekingAlpha*. from https://seekingalpha.com/news/3881425-annaly-capital-management-to-implement-1-for-4-reverse-stock-split

Richardson, Leonard (2020), *BeautifulSoup Documentation*, from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

The Motley Fool (2022), *Earnings Call Transcripts,* from https://www.fool.com/earnings-call-transcripts/

Tuovila, Alicia, Investopedia (2022), *Guide to Company Earnings*, from https://www.investopedia.com/terms/e/earningsreport.asp#:~:text=Earnings%20reports%20are%20important%20to,sometimes%2C%20for%20the%20entire%20market.

Code Appendix

Here we have the code appendix for our web scraping, sentiment analysis and multi-level modeling. Below, we used the R package reticulate to attempt to make our results reproduceable in an R environment from a Python Module. Our first python module functions properly, however, the second one fails to run, since reticulate has experienced some bugs downloading particular python packages and libraries, namely the alpaca-api the yfinance api and timedelta. However, if your machine is able to install these packages, feel free to run this appendix to create the finalData.csv with your own Alpaca Api or contact ntcherev@andrew.cmu.edu for more details. Notice that the code chunks in the appendix have all the necessary code necessary to replicate this report, so you can also extract the python modules from this appendix and run it in a normal python environment. If you have access to the "finalData.csv" you can copy the code from the "Modeling and Statistical Analysis in R" and on to reproduce our hierarchical model results!

```r
library(reticulate)

py_run_file("download_calls_txt.py")
```

Python Module 1

```python
import pathlib
import requests
from os.path import exists
from bs4 import BeautifulSoup
import lxml.html


listofurls = []
for pageno in range(1, 25):
   url = f'https://www.fool.com/earnings-call-transcripts/?page={pageno}'
   headers={'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36'}
   response = requests.get(url, headers=headers, timeout=100)
   page = response.text
   soup = BeautifulSoup(page, "lxml")
   for a in soup.find(class_="page").find_all(lambda x: x.name == 'a' and
   x.get('class') == ['text-gray-1100']):
       listofurls.append('https://www.fool.com'+str(a['href']))
print(len(listofurls))

for idx, url in enumerate(listofurls):
   path = pathlib.PurePath(url)
   filename = 'out-text-analysis/' + path.name + '.txt'
   if exists(filename):
       print("Hello!")
       continue

   response = requests.get(url, headers=headers, timeout=100)
   page = response.text
   soup = BeautifulSoup(page, features="html.parser")

   for script in soup(["script", "style"]):
```

```
        script.extract()

    text = soup.get_text()

    lines = (line.strip() for line in text.splitlines())

    chunks = (phrase.strip() for line in lines for phrase in line.split("  "))

    text = '\n'.join(chunk for chunk in chunks if chunk)
    textClean = '\n'.join(text.split("\n")[308:])
    textFullClean = textClean.split("All earnings call transcripts")[0]


    with open(filename, "w") as file:
            file.write(str(textFullClean)
```

```
py_run_file("configure_df.py")
```

Python Module 2

```
from transformers import BertTokenizer, BertForSequenceClassification
from transformers import pipeline
import pandas as pd
import alpaca_trade_api as tradeapi
import datetime as dt
import datedelta
import os
import yfinance as yf


ENDPOINT = 'https://api.alpaca.markets'
KEY = "API KEY CONTACT NTCHEREV FOR ACCESS"
SECRET_KEY = "API SECRET KEY CONTACT NTCHEREV FOR ACCESS"

api = tradeapi.REST(key_id=KEY, secret_key=SECRET_KEY,
                    base_url=ENDPOINT, api_version='v2')

print("API Connected")

finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone',num_labels=3)
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')
nlp = pipeline("sentiment-analysis", model=finbert, tokenizer=tokenizer)

print("Model Downloaded")

directory = 'out-text-analysis'
iter = 0
format = ' %B %d, %Y.'
invalid_days = set([5, 6, 0])

final_df=pd.DataFrame(columns = ['TIC','cur_price','old_price', 'new_price',
```

```
'old_week_price', 'new_week_price','date', 'posRel', 'negRel', 'pos', 'neg',
'sector', 'industry', 'beta', 'mktcap'] )

for filename in os.listdir(directory):
    print(iter)

    try:
        f = os.path.join(directory, filename)
        a_file = open(f)
        file_contents = a_file.read()

        contents_split = file_contents.splitlines()
        ticker = contents_split[-1].split(' ')[1]

        sentences = ' '.join(contents_split[13:]).split(".")

        date = dt.datetime.strptime(contents_split[8].split('ending')[1], format)
        if date.weekday() in invalid_days:
            date = date + 3*datedelta.DAY
        cur_date = date.strftime("%Y-%m-%d")
        month_back = date - datedelta.MONTH

        if month_back.weekday() in invalid_days:
            month_back = month_back + 3*datedelta.DAY

        week_back = date - datedelta.WEEK
        if week_back.weekday() in invalid_days:
            week_back = week_back + 3*datedelta.DAY

        week_forward = date + datedelta.WEEK
        if week_forward.weekday() in invalid_days:
            week_forward = week_forward+ 3*datedelta.DAY

        month_forward = date + datedelta.MONTH
        if month_forward.weekday() in invalid_days:
            month_forward = month_forward + 3*datedelta.DAY

        month_back_str = month_back.strftime("%Y-%m-%d")
        month_forward_str = month_forward.strftime("%Y-%m-%d")
        week_forward_str = week_forward.strftime("%Y-%m-%d")
        week_back_str = week_back.strftime("%Y-%m-%d")

        week_back_price = api.get_bars(ticker, '1Day', week_back_str,
        week_back_str, adjustment='raw').df.close[0]
        week_forward_price = api.get_bars(ticker, '1Day', week_forward_str,
        week_forward_str, adjustment='raw').df.close[0]
        month_back_price = api.get_bars(ticker, '1Day', month_back_str,
        month_back_str, adjustment='raw').df.close[0]
        month_forward_price = api.get_bars(ticker, '1Day', month_forward_str,
        month_forward_str, adjustment='raw').df.close[0]
        call_price = api.get_bars(ticker, '1Day', cur_date, cur_date,
        adjustment='raw').df.close[0]
```

```python
        print('Prices loaded')

        results = nlp(sentences)

        num = len(results)
        positive = 0

        neutral = 0
        negative = 0
        for sentence in results:
            if sentence['label'] == 'Neutral':
                neutral = neutral + 1
            elif sentence['label'] == 'Positive':
                positive = positive + 1
            else:
                negative = negative + 1

        negative_rel = negative/num
        positive_rel = positive/num

        print("Sentiment worked!")

        try:
            yfinfo = yf.Ticker(ticker)
            sector = yfinfo.info['sector']
            industry = yfinfo.info['industry']
            mktcap = yfinfo.info['marketCap']
            beta = yfinfo.info['beta']
            print('Metadata secured')

        except:
            print('Metadata failed')
            sector = 'NA'
            industry = 'NA'
            mktcap = -9999
            beta = -9999

        final_df.loc[len(final_df.index)] = [ticker, call_price, month_back_price,
                                        month_forward_price, week_back_price,
                                        week_forward_price,cur_date, positive_rel, negative_rel,
                                        positive, negative, sector, industry, beta, mktcap]

        print(ticker + " worked! " + str(iter) + " Done")

    except Exception as e:
        print(e)
        print(str(iter) + " failed!")

    iter += 1

final_df.to_csv('finalData.csv')
```

Modeling and Statistical Analysis in R

```
library(ggplot2)
library(lme4)
library(arm)
library(HLMdiag)
library(leaps)
library(LMERConvenienceFunctions)
```

```
library(cmu.textstat)
library(tidyverse)
library(quanteda)
library(quanteda.textstats)
library(nFactors)
library(ggridges)
library(ggplot2)
```

Read the Data, Set up necessary Variables, Data Exploration

```
sentiment <- read.csv('sentiment-data.csv')
```

```
sentiment$mb_returns <- (sentiment$cur_price - sentiment$old_price)/sentiment$old_price
sentiment$mf_returns <- (sentiment$new_price - sentiment$cur_price)/sentiment$cur_price
sentiment$wb_returns <- (sentiment$cur_price - sentiment$old_week_price)/sentiment$old_week_price
sentiment$wf_returns <- (sentiment$new_week_price - sentiment$cur_price)/sentiment$cur_price
```

```
earlyfiles_list <- list.files("out-text-analysis", full.names = T)

corpus2021 <- earlyfiles_list %>%
  readtext::readtext()%>%
  mutate(text = str_squish(text))

dfm2021 <- corpus(corpus2021) %>%
  tokens(what="fastestword", remove_numbers=TRUE) %>%
  tokens_compound(pattern = phrase(multiword_expressions)) %>%
  dfm()

tokens <- sum(dfm2021@x)
```

```
sentiment.no <- sentiment[sentiment$TIC != 'NLY', ]
```

```
tabl <- data.frame("Corpus" = c("Earnings Call Transcrips"), Total = c(480), Tokens = c(4077724))
kableExtra::kbl(tabl, caption = "Composition of the corpus.", booktabs = T, linesep = "") %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic() %>%
  kableExtra::row_spec(1, bold=T)
```

```
data.copy <- sentiment
data.copy$sector <- factor(data.copy$sector)
contrasts(data.copy$sector) <- contr.sum(11)
lm.check <- lm(negRel ~ sector, data = data.copy)
summary(lm.check)
```

```
lm.check2 <- lm(posRel ~ sector, data = data.copy)
summary(lm.check2)
```

```
lm.check3 <- lm(negRel ~ sector, data = data.copy)
summary(lm.check3)
```

```
sentiment.no$negPer <- sentiment.no$negRel * 100
sentiment.no$posPer <- sentiment.no$posRel * 100
sentiment.no$mb_returns <- sentiment.no$mb_returns * 100
sentiment.no$mf_returns <- sentiment.no$mf_returns * 100

sentiment$negPer <- sentiment$negRel * 100
sentiment$posPer <- sentiment$posRel * 100
sentiment$mb_returns <- sentiment$mb_returns * 100
sentiment$mf_returns <- sentiment$mf_returns * 100


sentiment.no <- sentiment.no[!is.na(sentiment.no$mktcap),]
sentiment.no <- sentiment.no[!is.na(sentiment.no$beta),]
```

```
cor(sentiment.no$posPer, sentiment.no$beta)
cor(sentiment.no$negPer, sentiment.no$mktcap)
cor(sentiment.no$negPer, sentiment.no$beta)
cor(sentiment.no$posPer, sentiment.no$beta)
```

```
mean(sentiment.no$posPer)
mean(sentiment.no$negPer)
sd(sentiment.no$negPer)
```

Data Visualization

```
ggplot(sentiment, aes(x = posPer)) +
  geom_histogram(fill = "#52BE80", alpha = 0.7) +
  labs(x = "Percent Positive Sentences", y = "Frequency",
       title = "Distribution of Positive Sentences in Earnings Calls")
```

```
ggplot(sentiment, aes(x = mf_returns)) +
  geom_histogram(fill = "#5DADE2", alpha = 0.8) +
  labs(x = "Percent Returns", y = "Frequency",
       title = "Distribution of Returns One Month After Earnings Calls")
```

```r
ggplot(sentiment, aes(x = mb_returns)) +
  geom_histogram(fill = "#5DADE2", alpha = 0.8) +
  labs(x = "Percent Returns", y = "Frequency",
       title = "Distribution of Returns One Month Before Earnings Calls")
```

```r
ggplot(sentiment.no, aes(x = negPer)) +
  geom_histogram(fill = "#CB4335", alpha = 0.7) +
  labs(x = "Percent Negative Sentences", y = "Frequency",
       title = "Distribution of Negative Sentences in Earnings Calls")
```

Modeling

```r
lmer.forward.neg <- lmer(mf_returns ~ 1 + negPer  + beta + mktcap + (1 | sector), data = sentiment.no)
lmer.backward.pos <- lmer(posPer ~ mb_returns + (1 | sector), data = sentiment.no)
lmer.forward.pos <-lmer(mf_returns ~ 1 + posPer  + beta + mktcap + (1 | sector), data = sentiment.no)
lmer.backward.neg <- lmer(negPer ~ mb_returns + (1 | sector), data = sentiment.no)
```

```r
summary(lmer.forward.pos)
```

```r
summary(lmer.backward.pos)
```

```r
summary(lmer.forward.neg)
```

```r
summary(lmer.backward.neg)
```

```r
library(dplyr)
```

```r
tabl <- data.frame(
  Sentiment = c("Positive Sentiment", "Negative Sentiment"),
  Coefficient = c( 0.1436, -0.4166),
  SE = c( 0.1009, 0.1969),
  T.Value = c( 1.787, -4.533),
  lower.CI = c(-0.0542, -0.8025),
  upper.CI = c(0.3414, -0.0307))
kableExtra::kbl(tabl, caption = "Coefficients on Sentiment in Multi-Level Models focused on Returns afte
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

```r
tabl2 <- data.frame(
  Sentiment = c("Positive Sentiment", "Negative Sentiment"),
  Coefficient = c( 0.0489, -0.0616),
  SE = c( 0.0274, 0.0136),
  T.Value = c( 1.787, -4.533),
  lower.CI = c(-0.0048, -0.088256),
  upper.CI = c(0.1026, -0.0349))
kableExtra::kbl(tabl2, caption = "Coefficients on Returns in Multi-Level Models focused on Sentiment", b
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

```r
r.11 <- hlm_resid(lmer.forward.neg,level=1,include.ls=F)
r.11s <- hlm_resid(lmer.forward.neg,level=1,include.ls=F,standardize=T)
r.21 <- hlm_resid(lmer.forward.neg,level="sector",include.ls=F)
r.21s <- hlm_resid(lmer.forward.neg,level="sector",include.ls=F,standardize=T)


ggplot(r.11, aes(x = .mar.fitted, y = .mar.resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.11, aes(x = .fitted, y = .resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.11s, aes(sample = .std.resid))+
  stat_qq()+
  stat_qq_line()

ggplot(r.21s, aes(sample = .std.ranef.intercept))+
  stat_qq()+
  stat_qq_line()
```

```r
r.12 <- hlm_resid(lmer.forward.pos,level=1,include.ls=F)
r.12s <- hlm_resid(lmer.forward.pos,level=1,include.ls=F,standardize=T)
r.22 <- hlm_resid(lmer.forward.pos,level="sector",include.ls=F)
r.22s <- hlm_resid(lmer.forward.pos,level="sector",include.ls=F,standardize=T)


ggplot(r.12, aes(x = .mar.fitted, y = .mar.resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.12, aes(x = .fitted, y = .resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.12s, aes(sample = .std.resid))+
  stat_qq()+
  stat_qq_line()

ggplot(r.22s, aes(sample = .std.ranef.intercept))+
  stat_qq()+
  stat_qq_line()
```

```r
r.13 <- hlm_resid(lmer.backward.neg,level=1,include.ls=F)
r.13s <- hlm_resid(lmer.backward.neg,level=1,include.ls=F,standardize=T)
r.23 <- hlm_resid(lmer.backward.neg,level="sector",include.ls=F)
r.23s <- hlm_resid(lmer.backward.neg,level="sector",include.ls=F,standardize=T)


ggplot(r.13, aes(x = .mar.fitted, y = .mar.resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")
```

```r
ggplot(r.13, aes(x = .fitted, y = .resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.13s, aes(sample = .std.resid))+
  stat_qq()+
  stat_qq_line()

ggplot(r.23s, aes(sample = .std.ranef.intercept))+
  stat_qq()+
  stat_qq_line()
```

```r
r.14 <- hlm_resid(lmer.backward.pos,level=1,include.ls=F)
r.14s <- hlm_resid(lmer.backward.pos,level=1,include.ls=F,standardize=T)
r.24 <- hlm_resid(lmer.backward.pos,level="sector",include.ls=F)
r.24s <- hlm_resid(lmer.backward.pos,level="sector",include.ls=F,standardize=T)


ggplot(r.14, aes(x = .mar.fitted, y = .mar.resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.14, aes(x = .fitted, y = .resid))+
  geom_point()+
  geom_abline(intercept = 0, slope = 0, color = "red")

ggplot(r.14s, aes(sample = .std.resid))+
  stat_qq()+
  stat_qq_line()

ggplot(r.24s, aes(sample = .std.ranef.intercept))+
  stat_qq()+
  stat_qq_line()
```