**Can Machine Learning Read CEOs Better Than Humans:**

A Natural Language Processing Analysis of the Textual Characteristics of Quarterly Earnings Calls and Medium-Term Stock Returns

Adam Watts, William White

Northwestern University, 1845 Sheridan Road, Evanston, IL, 60208

**Abstract**

As Quarterly Earnings Calls have become an increasingly important tool for public company CEO’s to communicate information, they have become a target of intense investor scrutiny. The content of these calls causes dramatic swings in stock prices after, and even during the call. Traditionally, humans have done this analysis, parsing the exchanged language in search of any information that changes the stock’s value. As Natural Language Processing (NLP), a machine learning technique designed to analyze text, has become increasingly relevant in financial literature and practice, investors have begun to use NLP to analyze these calls in real time. I investigated if NLP tools could uncover latent information in the text of the calls that would be gradually incorporated into the stock price. Using a basket of NLP techniques, including sentiment and personality trait analysis, to predict 20-day returns, I was able to generate 18.3% annualized returns with 13% of alpha, or “edge” to the market, on an out-of-sample set. This shows that the NLP scores reflect latent patterns in these calls that the market is missing. Traditional investors have cautioned that machine learning may have an ability to handle large, quantitative datasets but lack a necessary understanding of the human, behavioral forces at work in the market. However, these results, along with similar NLP research, suggest that machine learning and Natural Language Processing may already be better at interpreting the human, behavioral forces in the market than even sophisticated Wall Street investors.

Table of Contents

[Intro 3](#_Toc11245639)

[Literature Review 4](#_Toc11245640)

[Data Collection 5](#_Toc11245641)

[Analysis 5](#_Toc11245642)

[Natural Language Processing 5](#_Toc11245643)

[Regression Analysis 6](#_Toc11245644)

[Backtesting 10](#_Toc11245645)

[Portfolio Selection Techniques 10](#_Toc11245646)

[Selections of Best Linear Expected Return Model 10](#_Toc11245647)

[Comparison of Backtest Results for Backwards-Selected Model with Equal Weighting 11](#_Toc11245648)

[Works Cited 15](#_Toc11245649)

[Appendix A: Python Script Pseudocode for Cleaning Transcripts 16](#_Toc11245650)

[Appendix B: Long-Only Out-of-Sample Backtest Results with Backwards Linear Model and Equal Weighting 18](#_Toc11245651)

[Appendix C: Link to Github Repository 22](#_Toc11245652)

## Intro

As quarterly earnings calls, hereafter referred to as QECs, have become an integral tool of the modern executive and company, I sought to determine whether there are textual cues embedded in the interactions among CEOs and other participants in the call using an approach called natural language processing. Natural language processing, NLP, is a machine learning algorithm trained to detect sentiment and textual characteristics, understanding the meaning and context of different words and phrases. In some cases, training these algorithms produces dictionaries of words and phrases with associated traits such as positivity or negativity. I built on the work of others applying machine learning, especially NLP, to uncovering signals for future, abnormal stock returns. I used regressions, along with other data science techniques, to tie the textual characteristics of QECs to the stock returns of companies. I then constructed a portfolio of companies based on their textual scores to determine if this strategy produces high returns and quality portfolio benchmark statistics. This analysis determined that interaction between CEO’s and other earnings calls participants provide actionable insight into the future stock performance of a company.

Companies hold QECs four times a year to discuss financial results for the past quarter, as well as offering guidance for the future. These calls are normally led by the CEO and a handful of other company executives, with a large group of equity analysts independently evaluating the performance and financial health of a company. These calls are separated into prepared remarks and a question and answer section; I am interested in the Q&A section as it will provide better insight into the executive’s feelings and personality more so than prepared remarks, which may have been written by someone else. QECs have increasingly become tools for companies to manage investor expectations, and thus, they have become increasingly scrutinized for clues to future company performance. Recent financial literature is incorporating NLP, a machine learning technique designed to analyze the tone and meaning of words within text documents

## Literature Review

Malhotra et. al. (2017) establishes a clear link between the extraversion of a CEO and the likelihood of that company acquiring other companies. This shows the feasibility of measuring personality traits using NLP tools, and the basis for connecting the personality of an executive to the traits of the company as a whole. Majumder et. al. (2017) provides an open-source tool for the analysis of personality traits from transcribed text, which will allow me to apply their analytical techniques to my research question. Price et al. (2011) establishes that the text of QECs can be used to predict abnormal stock returns, using a NLP named General Inquirer to determine positivity and negativity scores. In Price et al. (2011), they utilize both a generic dictionary for their NLP, as well as a finance-specific dictionary because Henry (2008) contends that different disciplines inherently have different contextual meanings for words and phrases, specifically in the case of finance.

This collection of literature provides a basis for the study of CEO interactions with the use of NLP tools. I will go beyond the work of Price et al. (2011) to not just analyze the positivity and negativity of the CEO’s words, but I will analyze the interactions of the sentiment of the CEO’s words with those of analysts on the call and the personality traits that the CEO’s transcribed speech demonstrates.

I will utilize the work of many before in the realm of portfolio optimization to ensure that any trading strategy based on our analysis performs at its highest potential. I will build on the classic work of Markowitz (1952), establish Modern Portfolio Theory, integrating the newer ideas of risk-parity. The popularization of risk-parity is often credited to Bridgewater Investments and Ray Dalio. This weighting will be done using a GARCH model to estimate volatility as done by French et. al. (1987).

## Data Collection

Transcripts were collected from the website SeekingAlpha, with the PDF’s of the transcript downloaded from the website and then converted into text files using an online tool. 1,500 transcripts were collected from SeekingAlpha, with roughly equal representation across sectors and time (spanning 2011-2018). Of this 1,097 only were able to be used in the analysis. This elimination of transcripts was done for a number of reasons. Firstly, our analysis was not set up in a way to accommodate calls without CEO’s present on the call. In addition, the NLP tools required minimum word counts from the CEO, leading transcripts where CEO’s spoke under 100 words to be eliminated. Lastly some were not usable simply because of formatting issues with the PDF that corrupted the files. These undoubtedly introduce biases, especially the first two reasons, into the study, and further research could incorporate calls with little or no CEO involvement into this analysis.

The transcripts were cleaned, separated into the Q&A section, and then into the text of the CEO, the analysts, and company executives (not the CEO). This was done using a python script described further in Appendix A. These pieces of text were then fed into the analytical techniques described in the NLP Analysis section of this paper.

Analysis

### Natural Language Processing

Natural Language Processing is a Machine Learning technique that looks to analyze text rather than numerical data. Generally, it involves assigning characteristics to certain combinations of words. There are myriad techniques that evaluate everything from positive or negative sentiment to personality traits such as openness or neuroticism. In this research I utilized a generic sentiment analysis tool, a finance-specific sentiment analysis tool, and an IBM Watson personality trait analysis tool.

I utilized the Python NLTK Vader sentiment scores for the use of this analysis. The Vader Lexicon in python is a generic, rules-based sentiment analysis tool. It was originally developed for use on social media, and therefore, while it does not provide a finance-specific application, it provides a generic idea of the sentiment in each call.

Language is often context specific, and finance often has a vocabulary of its own with different meanings for different combinations of words. In her paper, Henry (2008) details dictionaries of positive and negative words specifically for financial contexts. I applied simple counting mechanisms to determine the number of positive and negative words in the CEO’s text. This was then translated into a score as shown in the formula below.

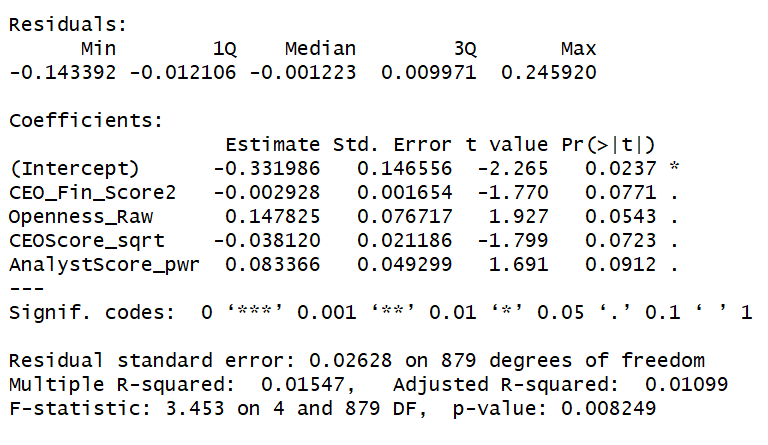
In addition to the sentiment scores, I looked to also utilize the IBM Watson tool that evaluated the Big 5 Personality Types scores. These traits are Openness, Neuroticism, Extraversion, Conscientiousness, and Agreeableness. The IBM Watson NLP tool assigns each block of CEO text a score from 0-1 based on a comparison to IBM’s entire database of text. This score introduced the limit for CEO’s having greater than 100 words spoken, as the scoring is not effective for any block of text shorter than 100 words.

### Regression Analysis

Some of the scores formed skewed distributions, and I looked to transform these into normal distributions. I utilized the ladder of power transformation laid out in Tukey (1977). These transformations should allow for superior performance of the regression tools as opposed to if I had simply used the raw, skewed data. In the regression results, variables have been renamed with the appropriate transformation (i.e. CEO\_Score\_sqrt).

With the number of predictors, 17, and a high degree of multicollinearity between the different predictors some variable selection was always likely to be needed. A linear regression run with all the predictors that produced over-fitting and performed poorly on the test set confirmed this fact. To accomplish this variable selection, I utilized three different methods initially, backwards-stepwise selection, forwards-stepwise selection, and LASSO. Backwards-stepwise simply starts with all of the predictors, at each step removing the one that will generate the biggest improvement in AIC, until no more improvements can be made. Forward-stepwise does the opposite, starting from no predictors, moving until no improvements can be made. LASSO regression in general is simply a regression with a penalty term for each variable introduced. In R this involves using the glmnet package to optimize for the best penalty term and then assessing which variables this regression chooses. All of these, excluding LASSO, were also applied with the selection of all one-way, multiplicative interactions (predictor1 \* predictor2). The LASSO model could not be applied, as the number of possible predictors was to high for the model to converge correctly.

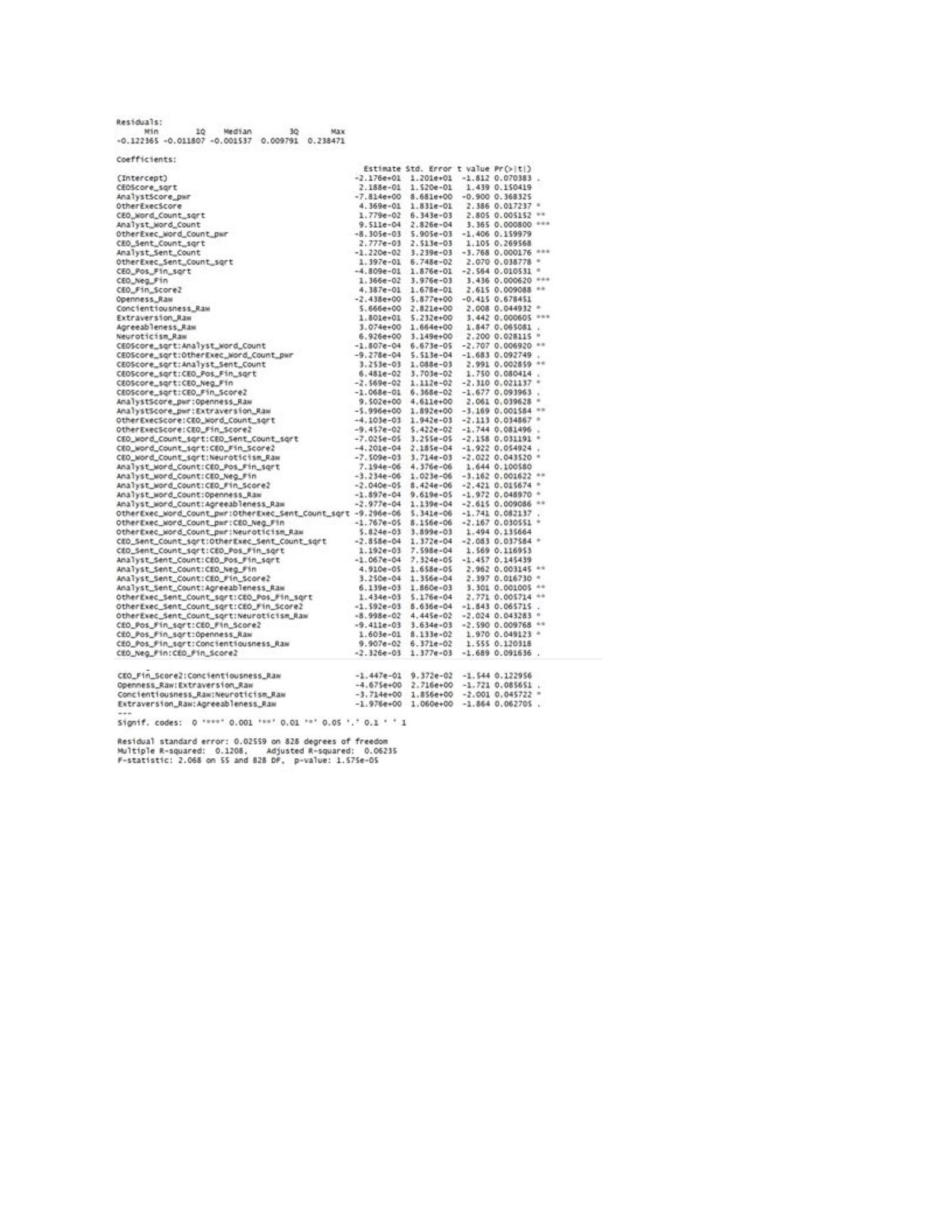
The forward-selected model selected none of the interactions and was limited to a very small number of terms. The R output below describes the model selected.



*Figure 1: Forward-Selected Model with Interactions*

This model provides interpretable results, with CEO positivity being negatively correlated to returns and Analyst positivity and CEO Openness being positively correlated with returns. This model provided small, 1.5% explanatory power for returns. It had a train set MSE of .00069 and a test set MSE of .00055. This suggests it has definitively not over-fitted, selecting a small set of predictive variables from the much larger set, and none of the multiplicative interactions.

The backwards-stepwise model selected a much larger subset of the data, including many multiplicative interactions. The R output below describes the model:



*Figure 2: Backwards-Stepwise Selected Model*

While this model is significantly more difficult to asses in terms of interpretability, it does have a significantly larger 12.1% R2, and even 6.23% when adjusted for number of variables. It produces a .00061 MSE on the training set, and a .00069 MSE on the test set. While the MSE is higher on the test set than the smaller, forward-selected model, it is still reasonably similar to the training set and does not suggest high levels of over-fitting.

From here I looked to evaluate the results not just in terms of the statistical results described above but how a trading strategy based on the expected returns generated would perform in a backtest.

## Backtesting

### Portfolio Selection Techniques

The initial portfolios Ire simply selected by equally Weighting each stock in the portfolio. While this technique may lack the sophistication of later techniques described, it provided a baseline for which models and which trading strategies (long-only vs. long/short) might prove the most promising. In the case of long/short portfolio, the long and short weights were computed separately, so that the total long and short exposures Ire equal.

The second technique attempted was a weighting of each stock by its expected return, as specified by the regression model. At the beginning of each day, the portfolio was adjusted such that each stock within the portfolio had a weighting of its expected return. Again, in the case of the long/short portfolio, the long and short weights were computed separated, to create a market-neutral portfolio where long and short exposures were equal.

### Selections of Best Linear Expected Return Model

While the backwards and forwards selected models demonstrated similar Mean-Squared-Errors, I looked to backtests to see if there was one that outperformed the other in a trading strategy. As it turns out, the backwards selected model vastly outperformed the forwards selected in every backtest performed: both in and out of sample as well as in both long-only strategies and long/short strategies. The fact that the backwards-selected model outperformed the forwards one in the out of sample backtests assuaged the main concern about the backwards-selected model: that it was overfitting the data. These comparisons were made with an equal-weighted portfolio; however, given the backwards model at least doubled the Sharpe Ratio of the forwards model in each of the tests, I feel confident that the other portfolio optimization techniques would not lead to better performance from the forwards model as compared to the backwards one.

### Comparison of Backtest Results for Backwards-Selected Model with Equal Weighting

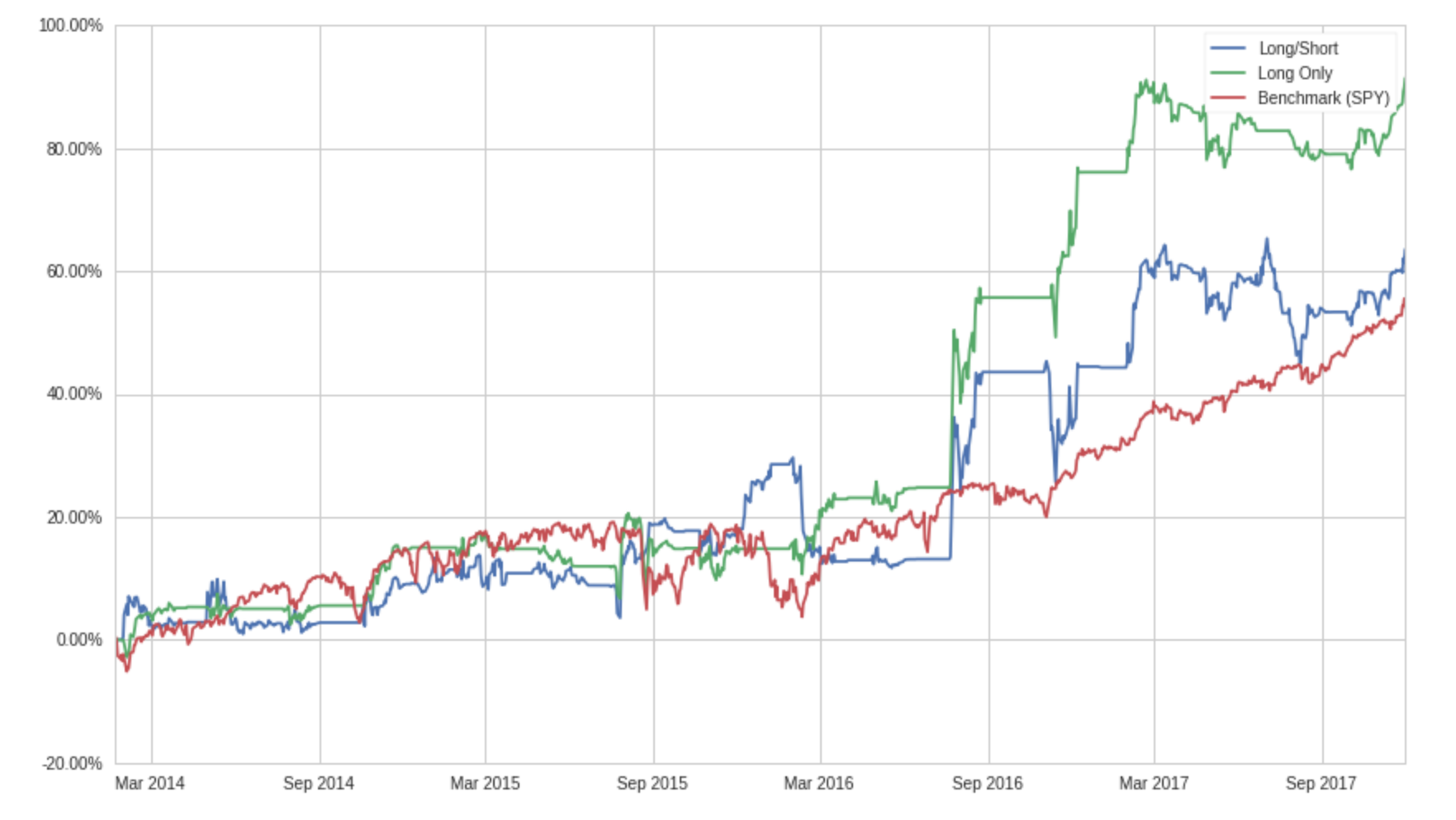
In all four of the backtests, the trading strategy based on the backwards-selected model performed effectively, demonstrating the predictive power the model held both in predicting long and short price movements, both in sample and out of sample. Both the long-only and the long/short strategies perform equally if not better out of sample as opposed to their in sample backtest, which demonstrates that the model is not simply overfitting to the data it is fed, but it is finding consistent patterns that hold even for new data the model sees. For the long-only, the strategy almost doubles the benchmark returns, 91.42% for the strategy and 55.27% for SPY. It generated 13% of alpha, or uncorrelated returns to the market, and it traded with a Sharpe Ratio of 1.22, a measure of risk-adjusted returns. For the long/short, the strategy still returned an extra 8% over the benchmark, totaling 63.54% for the period. It generated 14.85% of alpha and a Sharpe Ratio of .83. Figure 3 below contains a full suite of statistics for each of the four backtests while figures 4 and 5 show graphs of the portfolios’ value with the benchmark for reference both in and out of sample respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Linear Long Only Training Set | Linear Long/Short Training Set | Linear Long Only Test Set | Linear Long/Short Test Set |
| Total Returns | 133.38% | 76.11% | 91.42% | 63.54% |
| Benchmark Returns (SPY) | 137.73% | 137.73% | 55.27% | 55.27% |
| Alpha | 5.86% | 11.33% | 13.04% | 14.85% |
| Beta | 54.72% | -6.07% | 39.30% | -6.07% |
| Sharpe Ratio | 1.06 | 0.78 | 1.22 | 0.83 |
| Sortino Ratio | 1.61 | 1.30 | 2.27 | 1.40 |
| Volatility | 12.50% | 13.40% | 14.55% | 17.01% |
| Benchmark Volatility | 12.45% | 12.45% | 12.16% | 12.16% |

*Figure 3: Backtest Statistics for Backwards Linear Model with Equal Weighting*



*Figure 4: In-Sample Returns for Backwards-Selected Model with Equal Weighting*



*Figure 5: Out-of-Sample Returns for Backwards-Selected Model with Equal Weighting*

**Conclusions:**

While the long/short portfolio does generate alpha and possibly indicate some predictive power on the short side, Sharpe Ratio’s below 1.0 do not indicate a successful trading strategy especially when compared to the long-only portfolio. The nature of the portfolio as market-neutral should, in theory, help decrease the volatility of the strategy, but instead it has increase the volatility of the strategy. In all fairness, the entire backtest period was characterized by positive market returns so the long/short strategy could perform better in bull markets; however, in this test it appears that the model predicts the returns for stocks to the upside significantly better than to the downside. This is, in the end, not terribly surprising, as, in general, it is harder to successfully short stocks than to profit from going long stocks due to the upward-drifting nature of the market over time.

The long-only portfolio proved successful, attaining a much more positive Sharpe Ratio of 1.2 and generating 13% of alpha. This demonstrates the ability of the backwards-selected model to not only closely predict returns, but also to do so in a way that yields a successful trading strategy. This strategy indicates that the NLP scores do have the ability to uncover latent information in the earnings calls that are being incorporated into the stock price over 20 days following each earnings call.

In the future, there is certainly opportunity for more research into this subject. While I was limited by the fact that all of the transcripts were collected by hand, if there was a way to collect transcripts algorithmically, it would be possible to perform a more comprehensive analysis on all modern Quarterly Earnings Calls rather than a randomly selected subset. In addition, there is an opportunity to incorporate into this analysis the cases where CEO’s have little to no involvement in the call. This research stemmed from the question of how CEO’s interact with participants on earnings calls and how that affects stock returns; however, a broader analysis would incorporate how having a non-participatory CEO on a call might also affect the valuation of the stock going forward.

In addition to the biases that could be removed from this research, as discussed above, there is opportunity for more sophisticated NLP analysis to be applied to this problem. While they produced significant results, most of the tools I utilized were commonplace, out-of-the-box tools, rather than customized algorithms. Training a Word2Vec instance on my corpus of data and obtaining a significantly larger dataset and training a lexicon on that would likely provide more accurate results. An approach that takes into account the sequential nature of the conversation could also yield interesting results: namely, analyzing each paragraph as a response to the one above it rather than as individual blocks.

This research suggests that the human investors listening to these calls are missing latent

information, hidden in the text, that provides information about the value of the stock. The broader interpretation is perhaps that NLP algorithms can read and analyze the text of human conversation better than a human can. This has broad-reaching consequences not just in financial markets, but for the duality between humanity and Machine Learning in general. It has always been thought that, while Machine Learning might have quantitative edges over human’s, it was a long way off before it could understand the nuances of personality and behavior; however, this research, and other similar results, suggest that, in fact, Natural Language Processing algorithms allow computers to uncover patterns in human conversations that even the most sophisticated investors are missing.

## Works Cited

Amir, Eli, and Baruch Lev, 1996, Value-relevance of nonfinancial information: The wireless communications industry, *Journal of Accounting and Economics* 22, 3–30.

Fama, Eugene F., and Kenneth R. French, 1992, The Cross-Section of Expected Stock Returns, *The Journal of Finance* 47, 427–465.

French, Kenneth R., et al. “Expected Stock Returns and Volatility.” *Journal of Financial Economics*, vol. 19, no. 1, 1987, pp. 3–29., doi:10.1016/0304-405x(87)90026-2.

Garleanu, Nicolae B, and Lasse H Pedersen, 2009, Dynamic Trading with Predictable Returns and Transaction Costs. Working Paper, National Bureau of Economic Research.

Henry, Elaine, 2006, Are Investors Influenced by How Earnings Press Releases are Written? SSRN Scholarly Paper, Social Science Research Network, Rochester, NY.

Majumder, N., S. Poria, A. Gelbukh, and E. Cambria, 2017, Deep Learning-Based Document Modeling for Personality Detection from Text, *IEEE Intelligent Systems* 32, 74–79.

Malhotra, Shavin, Taco H. Reus, PengCheng Zhu, and Erik M. Roelofsen, 2018, The Acquisitive Nature of Extraverted CEOs, *Administrative Science Quarterly* 63, 370–408.

Markowitz, Harry. “Portfolio Selection.” *The Journal of Finance*, vol. 7, no. 1, 1952, p. 77., doi:10.2307/2975974.

Price, S. McKay, James Doran, David R. Peterson, and Barbara A. Bliss, 2011, Earnings Conference Calls and Stock Returns: The Incremental Informativeness of Textual Tone. SSRN Scholarly Paper, Social Science Research Network, Rochester, NY.

Tukey, J. W. (1977). Exploratory Data Analysis. Addison-Wesley, Reading, MA.

## Appendix A: Python Script Pseudocode for Cleaning Transcripts

This section describes the code that will be used to obtain the question and answer section of the quarterly earnings call transcripts and then separate that text into what is said by the CEO, other executives, and the analysts. Right after this description is an abridged version of a transcript for reference.

1. The code will first find the date of the call, for organization purposes, by finding a line in before the start of the “Executives” section that contains a piece of text of the format of a full date (i.e. [Month] [Day], [Year] [Time] [AM/PM]).
2. The code will then search for the beginning of the “Executives” section, which is delineated by a line that only contains the word “Executives” then a new-line character.
   1. Within the executives section, each executive is listed as [Name] – [Title], i.e. “Dr. Lisa Su – President and CEO”
   2. The code will split this into two string based on the dash, into [Name] and [Title].
      1. If the Title contains the word “CEO” or “Chief Executive Officer”, it will store that name as the CEO\_Name
      2. Otherwise, the name is simply stored in an array of other executive names
      3. Once the code reaches a line that is not of the form [Name] – [Title] it will stop collecting executive names
         1. It will create an error message if a CEO name has not been found yet
3. Once the code reaches the “Analyst” Section denoted by “Analysts” and a new-line character, it will begin collecting analyst names.
   1. In a similar fashion, the analyst section consists of lines with [Name] – [Company], i.e. “David Wong – Ills Fargo”
      1. This will again, be split in a similar fashion to executives on the dash, into [Name] and [Title]
      2. [Name] will be stored in an array of analyst names.
   2. Similarly, once a line is no longer of the specified format, it will stop looking for analysts.
4. The code will then search the entire transcript for a line that is entitled “Question-And-Answer-Section” followed by a new-line character.
   1. The code will store all of the lines in the transcript after this question and answer session line in a variable called “Interview” and this will be the text used for the rest of the code.
5. Within the interview, blocks of texts are delineated by tags of the format “[Name]\n” where \n is a newline character (the \n only shows up when code reads the text, it will not be visible in example transcript, just shown by the fact that there is nothing after the name)
   1. The code will find all of the tags in the interview, defining the CEO tags as the lines of format [CEO\_Name\n], the other executive tags as [Exec\_Name\n] for all of the executives in the array, and the analyst tags as [Analyst\_Name\n] for all the analyst in the analyst array.
      1. It will then, for each tag, find all of the lines after the tag until the code reaches a new tag.
      2. This block of text will then be assigned to an array of text based on what the tag at the start of the block was, i.e. if a CEO tag, text stored in CEO\_Text, if an Analyst Tag, assign to Analyst\_Text.
6. This results in three different arrays, a CEO\_Text, Other\_Exec\_Text, and Analyst\_Text arrays. These arrays of texts can then be analyzed using the tools described above.

## Appendix B: Long-Only Out-of-Sample Backtest Results with Backwards Linear Model and Equal Weighting

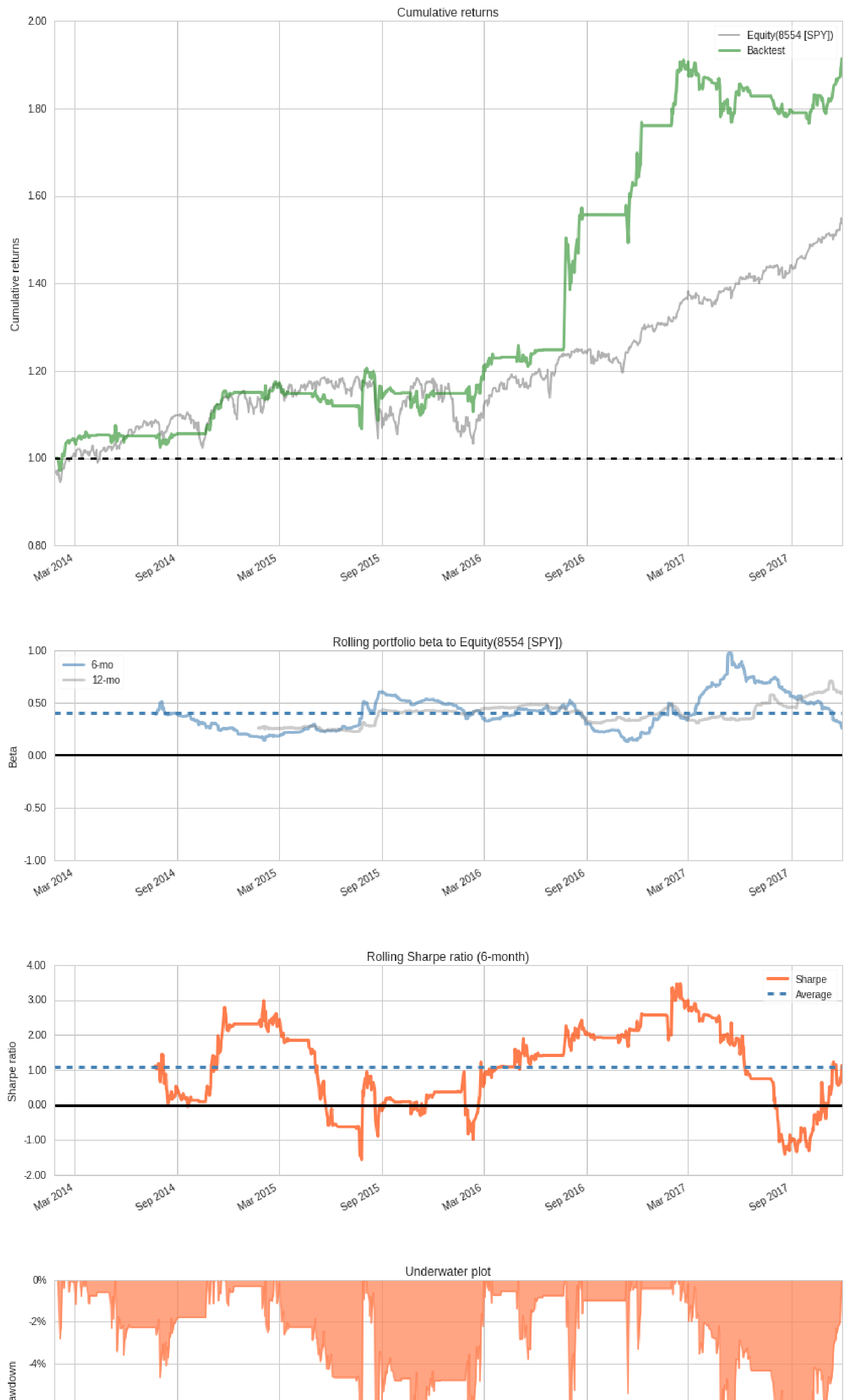
**Start date** 2014-01-23

**End date** 2017-12-01

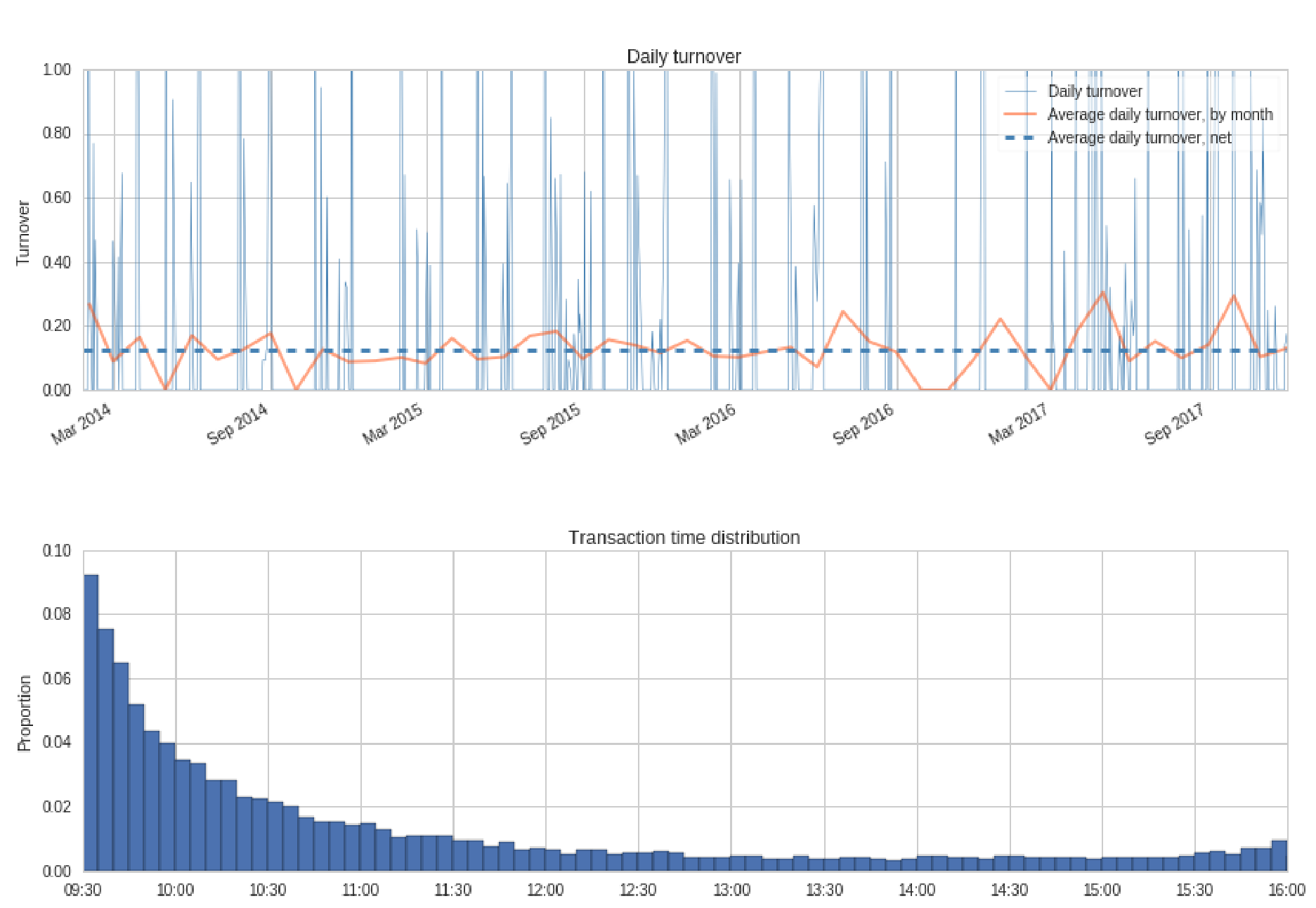
**Total months** 46

**Backtest**

|  |  |
| --- | --- |
| **Annual return** | 18.3% |
| **Cumulative returns** | 91.4% |
| **Annual volatility** | 14.6% |
| **Sharpe ratio** | 1.22 |
| **Calmar ratio** | 1.83 |
| **Stability** | 0.85 |
| **Max drawdown** | -10.0% |
| **Omega ratio** | 1.42 |
| **Sortino ratio** | 2.27 |
| **Skew** | 4.89 |
| **Kurtosis** | 78.84 |
| **Tail ratio** | 1.40 |
| **Daily value at risk** | -1.8% |
| **Gross leverage** | 0.42 |
| **Daily turnover** | 12.4% |
| **Alpha** | 0.13 |
| **Beta** | 0.39 |







## Appendix C: Link to Github Repository

<https://github.com/adamw1623/NLP_QEC->