Exploration of Different AI Methods for Stock Movement and Value Prediction during Earnings Season

Ilkyu Lee, Lingfei Yang, and Zhi (Sophia) Zhou
Stanford University

(Dated: December 14, 2019)

I. INTRODUCTION

The problem of stock prediction has been an active area of research for many years due to both practical reasons of optimizing financial gains and academic endeavors to understand the underlying behavior of the market in relation to internal (e.g. competitions among companies within a specific commodity) and external (e.g. global interactions such as wars) processes. Specifically, stock market behavior is most interesting during the earning-announcement season and the stock volume and volatility spikes during that period.

In this project, we want to focus on exploring two aspects of stock price movement during earning seasons. A momentum strategy is a bet on past returns predicting the cross section of future returns, typically implemented by buying winners and selling past losers. However, stocks tend to experience extreme abnormal return reversals around the earnings announcements dates, which can be explained by the fact that market makers demand greater compensation for providing liquidity ahead of anticipated information events.

Thus, in the first part of our research, we will try to utilize machine learning algorithms to develop a trading strategy to long-short assets and trade short-term reversals that occur around quarterly earnings announcement dates. The second part of the project will focus on utilizing natural language processing with textual data such as earnings-related documents, the 10-K, and news articles to predict the nature of stock movement. Ultimately, we seek to understand the effectiveness of different tools released during earnings season in predicting stock movement.

II. PRIOR WORK

Models of liquidity provision [2] indicate that market makers demand compensation for incurring inventory risks and adverse selection. It suggests that greater anticipated volatility and/or adverse selection risks associated with information events should lead to increased reversals. Lauren Cohen et al. [1] propose a model to longing changers and shorting non-changers given that language change in the public reports can drive a future profit of firms as they tend to repeat what they most recently reported. Especially, changing in litigation is a strong signal for future positive return. Marko Pozenel et al. [3] proposed a novel approach for stock trend prediction and financial success by combining NLP and Japanese candlesticks.

III. MODEL

A. Prediction with historical data mining

This part of study involves data mining and pattern prediction from historical data, and therefore highly relies on the quality of the historical data. Thus, data selection, extraction and manipulation play an important role in order to obtain high quality data in this case.

CRSP Daily stock data and Compustat quarterly fundamental data were obtained from the CRSP database. Our stock universe includes NYSE, AMEX, and NASDAQ. We used share codes ‘10’and ‘11’ to include common stock and exclude foreign, ADRs, REITs, and Closed End Funds. The sample period was from January 1996 to December 2017.

The stock and fundamental data tables were merged into a master table using the link table. The Compustat data was left-joined by GVKEY with the link table, which is the firm identifier. Certain link types were filtered out. Only LC and LU link types were used, since LD indicates a duplicate; LX is not applicable to CRSP data; LS has varying VGKEY for the same PERMNO; NR/NU links are unavailable; and LO has no entries in link table. Also, only primary links were used, which were P or C in the linkprim category. More filtering was done to ensure that the date in the Compustat was betweent the link start and link end date so there was only one unique PERMCO per GVKEY.

In terms of the portfolio sorting and forming, we followed the following steps:

1. Sort stocks into quintiles by size using the last period log market capitalization. Take only the largest quintile for further calculation.
2. Sort stocks in the largest quintile resulted from above step into quintiles using three-day cumulative returns from pre-announcement period (t-2 to t-4).
3. Form the long-short portfolio by buy(selling) firms in the lowest (highest) quintile.

Each stock for every period within the portfolio is equally-weighted.

B. NLP with Earnings Reports

In this section, the earnings conference call transcript files were modified such that the date of the call occurred on the third line for automated time scraping, and in the Q & A section, the text of the questioners were removed so that only the information from the representatives of the companies is retained. Then, the file was stripped of punctuation and put into lowercase and lemmatized. Then, certain words were removed using the Stopwords corpus.

Secondly, given a cleaned text, a feature extractor function $\phi(\mathbf{X})$ was used to quantify the properties of the text. The base feature extractor used was the identity extractor, which took each word in the text and counted its frequency, leading to a unigram model setup. A bigram model, wherein the frequency of a pair of consecutive words was used as features, was also used as an initial test. Then, each feature vector was connected to a label y of "-1" or "1", depending on whether the stock price increased or decreased from n business days before and after the conference call date. In this manner, the task at hand was binary classification, wherein we define a score function:

$$\text{score}_i = \mathbf{w} \cdot \phi(\mathbf{X}_i), \tag{1}$$

where $\mathbf{X}_i$ represents the text of a single, cleaned earnings call transcript, $\phi(x)$ is the feature extractor function, and $\mathbf{w}$ is the vector of weights that linearly combine the features. In order to find the optimized weights for the prediction problem, we minimized the hinge-loss function given by

$$\mathcal{L}(\mathbf{X}_i, y_i, \mathbf{w}) = \max\{0, 1 - \mathbf{w} \cdot \phi(\mathbf{X}_i) y_i\}. \tag{2}$$

In addition, a Multi-Layer Perception neural network was also used for the prediction. One hidden layer was used, with varying number of neurons in the layer. The activation function used was the ReLu function, and the final output of the neural network was then squashed to a value between 0 and 1 via the logistic function, and the prediction was made to be "-1" if the value is below 0.5, and "1" if the value is greater than or equal to 0.5.

C. NLP with 10-K Documents

In addition to NLP with earnings call transcript files, sentiment analysis with 10-K was explored in this part in order to seek earning surprises in the portfolio construction. 10-K is a comprehensive report filed annually by public companies about their financial performance with more detailed disclosure, compared to other regular financial reports. Specifically, it entails four parts: Business Overview, Markets or Financial Conditions, Governance, and Full Financial Statements. The model aimed to identify a trading strategy based on language changes in the 10-K documents.

The main steps involved:

1. Text Processing
2. Feature Extraction
3. Modelling in analyzing the similarities of word sentiment across time for stocks
4. Identifying the relationship between word sentiment and stock return surprises.

As to the text processing, to obtain as many 10-K reports as possible, we grabbed all possible reports in htm format from SEC website by reading url based on their unique CIK first, as 10-K file in txt format could be unavailable for earlier years. Then, we ruled out non-10K reports.

The sample chosen is SP100 whose sample period was available from January 1998 to September 2019. Among the list, some representative large cap corporates included Amazon (ticker: AMZN) and Boeing (BA), etc.

During 10-K text data clean-up, we removed html tags and lower-cased all the text. In addition, we used nltk corpus packages. Stopwords corpus was used for removing words such as I, you, which, etc. Besides, wordnet corpus was used for lemmatizing.

In terms of feature extraction, We created feature extraction by using Loughran and McDonald sentiment word master dictionary, which entails:

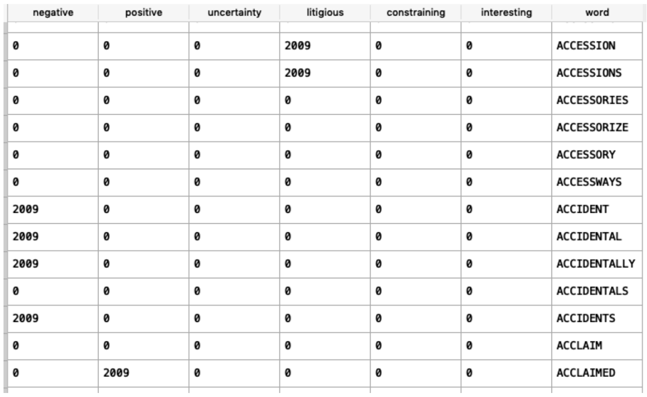
1. Negative
2. Positive
3. Uncertainty
4. Litigious
5. Constraining
6. Superfluous
7. Interesting
8. Modal.

Some categories including "Superfluous" and "Modal" were not considered, but we utilized the remaining categories as other categories were more meaningful in the financial information. Particularly, "Litigious" is more meaningful as companies are required to disclose "significant pending lawsuits or other legal proceedings".

In the process, we started from 86486 words from Loughran and McDonald sentiment word dictionary,removed unused information where the word category count is 0 for that word, and dropped duplicated words. As a result, the bag of words was shortlisted to 2720 words to be prepared for further analysis.

Additionally, several similarity metrics were computed such as Jaccard similarity over time and cos similarity. We implemented it between each tick in time and calculated the similarities for each neighboring bag of words.

In the last step, we computed stock return on the first trading date of August of each year as 10-K documents



negative	positive	uncertainty	litigious	constraining	interesting	word
0	0	0	2009	0	0	ACCESSION
0	0	0	2009	0	0	ACCESSIONS
0	0	0	0	0	0	ACCESSORIES
0	0	0	0	0	0	ACCESSORIZE
0	0	0	0	0	0	ACCESSORY
0	0	0	0	0	0	ACCESSWAYS
2009	0	0	0	0	0	ACCIDENT
2009	0	0	0	0	0	ACCIDENTAL
2009	0	0	0	0	0	ACCIDENTALLY
0	0	0	0	0	0	ACCIDENTALS
2009	0	0	0	0	0	ACCIDENTS
0	0	0	0	0	0	ACCLAIM
0	2009	0	0	0	0	ACCLAIMED

FIG. 1. Loughran McDonald Sentiment Word Lists Example

are usually released in earlier months of the year. The source we chose was Quotemedia where corporate actions such as dividend and stock split have been adjusted.

Then, we calculated alpha of each firm and ranked it by quintile. Sharpe ratio was also assessed by word list to identify the association with the stock movement.

IV. ALGORITHMS

A. Prediction with historical data mining

In this investigation, Jensen's alpha and beta were first estimated when the model was exposed to basic CAPM and three-factor Fama & French model. The empirical models we examined were:

CAPM :

$$R_t - R_f = \alpha + \beta_{mkt}(Mkt_t - R_f) + \epsilon_t, \tag{3}$$

Fama & French 3 factor :

$$\begin{aligned} R_t - R_f = \alpha &+ \beta_{mkt}(Mkt_t - R_f) \\ &+\beta_{smb} SMB_t \\ &+\beta_{hml} HML_t + \epsilon_t. \end{aligned} \tag{4}$$

B. NLP with Earnings Reports

The optimization of the classifier was done by applying stochastic gradient descent for each training sample via the following equation:

$$\mathbf{w} \longleftarrow \begin{cases} \mathbf{w} + \eta\phi(\mathbf{X}_i)y_i, & \mathbf{w}\cdot\phi(\mathbf{X}_i)y_i < 1 \\ \mathbf{w}, & \mathbf{w}\cdot\phi(\mathbf{X}_i)y_i \geq 1 \end{cases} \tag{5}$$

where η represents the tunable hyperparameter that dictates the speed of the descent. After performing the update for each training sample, the entire process was repeated a certain number of times represented by the variable N, which is another tunable hyperparameter. Error was measured by finding the quotient of the number of wrongly classified samples over the total number of samples.

In regards to the neural network, stochastic gradient descent was used to update the weights via backpropagation. The loss function that was minimized via stochastic gradient descent is the Cross-Entropy, given by

$$\mathcal{L}(\hat{y}, y, W) = -y \ln \hat{y} - (1-y)\ln(1-\hat{y}) + \alpha||W||_2^2, \tag{6}$$

where W is the matrix of weights, y is the true label, and $\hat{y}$ is the predicted label.

C. NLP with 10-K Documents

The Jaccard similarity compares members for two sets to determine which members are shared and which are distinct. It's a measure of similarity for the two sets of data, with a range from 0 to 100 percentage. The higher the percentage, the more similar the two populations. The Jaccard similarity index $J(U,V)$ is given by:

$$J(U,V) = \frac{|U \cap V|}{|U \cup V|}. \tag{7}$$

Similarly, Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity. In its equation form, the Cosine similarity $C(U,V)$ is given by:

$$C(U,V) = \frac{|U * V|}{|U| * |V|}. \tag{8}$$

V. RESULTS

A. Prediction with historical data mining

To quantitatively capture the pattern of the reversal, a correlation heatmap was utilized over the pre-announcement window ($t-4$ to $t-2$) and the announcement-window, which is shown in the figure below. The figure shows the general negative correlation between the announcement window and the pre-announcement window. It also reveals that $t-1$ is subject to negative correlation with t and $t+1$. This shows there is some variation of the actual timings of the events of reversal. The reversal can sometimes begin around $t-1$ or t. The positive correlation between $t+1$ and t shows evidence that the reversal continues for some time after the announcement.

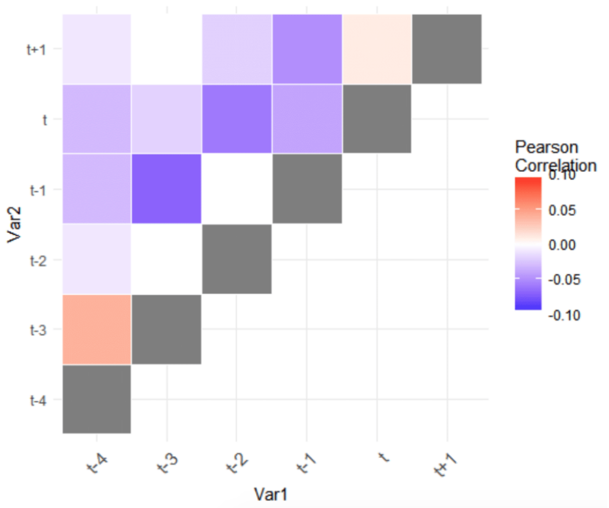
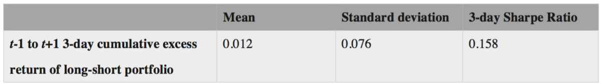


FIG. 2. Correlation Heatmap

Table1 below presents the mean, standard deviation and Sharpe ratio of the announcement-window excess return ($t-1$ to $t+1$) of the long-short portfolio sorted using pre-announcement return ($t-4$ to $t-2$).



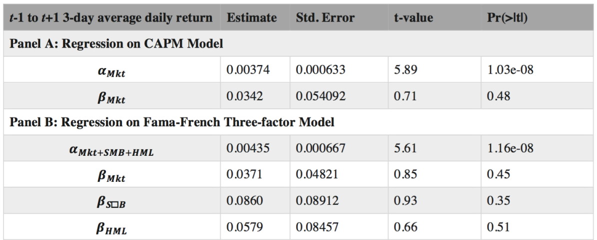
	Mean	Standard deviation	3-day Sharpe Ratio
t-1 to *t*+1 3-day cumulative excess return of long-short portfolio	0.012	0.076	0.158

FIG. 3. Return Statistics

Table2 below presents the strategy's performance in terms of exposures to CAPM and three-factor Fama and French model. Excess return of the strategy is regressed against excess market return and three factors return (Mkt-Rf, SMB, HML) respectively.

As we can see that the two regression results have similar size alphas of around 0.0035, with both being statistically significant according to t-value and P-value. Considering the fact that regression is ran against daily market excess returns, a daily abnormal return of 0.0035 is considerably high. In other words, our strategy is delivering an excess return that is not explained by either CAPM or FF-3.

In terms of factor loadings, CAPM model has a market beta of 0.033, indicating a fairly low correlation between the movement of market return and our strategy return. In addition, the market beta is not statistically significant according to both t-value and P-value. Similarly, for FF-3 model, betas on Mkt, SMB and HML are all positive but below 0.1. And t-stats suggest all of them are statistically insignificant, which demonstrates the fact that this mean reversal strategy cannot be explained by Mkt, SMB or HML factors.



t-1 to *t*+1 3-day average daily return	Estimate	Std. Error	t-value	Pr(>|t|)
Panel A: Regression on CAPM Model				
α_{Mkt}	0.00374	0.000633	5.89	1.03e-08
β_{Mkt}	0.0342	0.054092	0.71	0.48
Panel B: Regression on Fama-French Three-factor Model				
$\alpha_{Mkt+SMB+HML}$	0.00435	0.000667	5.61	1.16e-08
β_{Mkt}	0.0371	0.04821	0.85	0.45
$\beta_{S\square B}$	0.0860	0.08912	0.93	0.35
β_{HML}	0.0579	0.08457	0.66	0.51

FIG. 4. Regression Results

The Figure 2 below plots the average announcement-window return for each calendar year in the sample as a result of long-short strategy. The average returns are positive for all calendar years except for the year 2009. We find that reversal strategy works best in times of market stress when the market has fallen and thus the measures of volatility are high, such as the collapse of tech bubble around 2000 and the financial crisis in 2008.

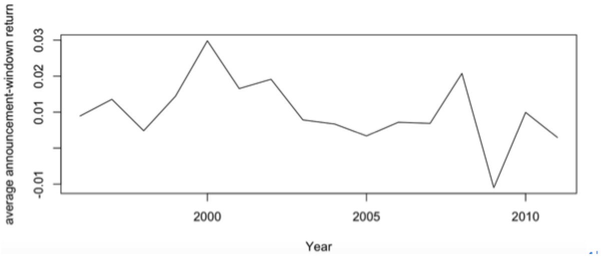
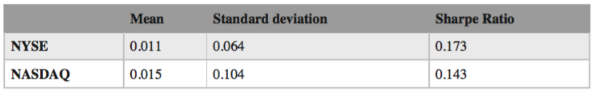


FIG. 5. Average Announcement-window Return

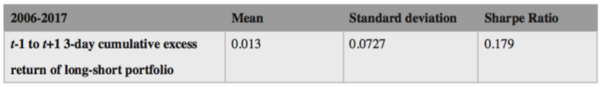
A few additional exercises are implemented to test the robustness of this strategy. First, we separate the sample by the firm's primary listing platform and results suggest that average announcement reversal is 1.1% among NYSE-listed firms and 1.5% among Nasdaq-listed firms. The finding suggests that information asymmetry and inventory risks are larger among Nasdaq firms, which have relatively smaller sizes.



	Mean	Standard deviation	Sharpe Ratio
NYSE	0.011	0.064	0.173
NASDAQ	0.015	0.104	0.143

FIG. 6. Average Announcement-window Return

Second, we repeat the exercise for a different time period from 2006 to 2017, and find the average return due to announcement reversal is 1.3%, which is consistent with what we find using sample data from 1996-2011. This finding demonstrates the fact that this strategy is robust and insensitive to sample data period.



2006-2017	Mean	Standard deviation	Sharpe Ratio
t-1 to *t*+1 3-day cumulative excess return of long-short portfolio	0.013	0.0727	0.179

FIG. 7. Average Announcement-window Return

B. NLP with Earnings Reports

The total data size was 80, consisting of conference call transcripts of Microsoft and Visa from 2009 to 2019. The data was split randomly into training and test sets such that 60% of the data was in the training set, and the remaining was in the test set. First, we do hyper-parameter (η) tuning, with n (the number of business days to use for label extraction) set to 1. The iteration number N was set to 60, as it was seen that convergence generally happens before that number. Per iteration, we measured the percentage of wrong classifications of the predictor and used that as our error metric. 10 runs were done, and the average % errors per iteration was recorded. In Fig 8, we plot the results in relation to the tuning of hyper-parameter η.

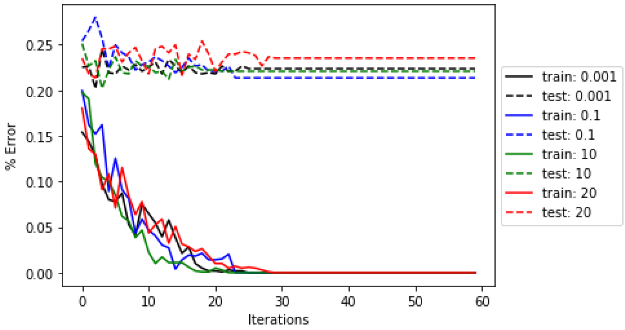


FIG. 8. Classification Error vs. Iterations for Different Values of η. The numbers in the legend represent *eta* values used.

From this figure, we note that tuning η did not make a considerably huge difference among the runs; regardless we choose $\eta = 0.1$ for the following runs. For the unigram model, we test how the model does in predicting stock movement within n business days of the earnings call date. The results are shown in Fig 9. As one can see, the unigram model performs the best in predicting stock movement given 2-month range with the earnings call date at the middle. This could be reflected by the fact that the information received during the earnings call can take some time for the to-do's mentioned in the call to occur and for the spread of information to be internalized by the audience. However, we note that the increase in accuracy is still minimal (10% increase).

Next, we change the unigram to an n-gram model, where the feature extraction is done using n consecutive words in the text. The results are shown in Figs 10 and 11. In the legend, the number after "train" or "test" represent the number of consecutive words used during the feature extraction.

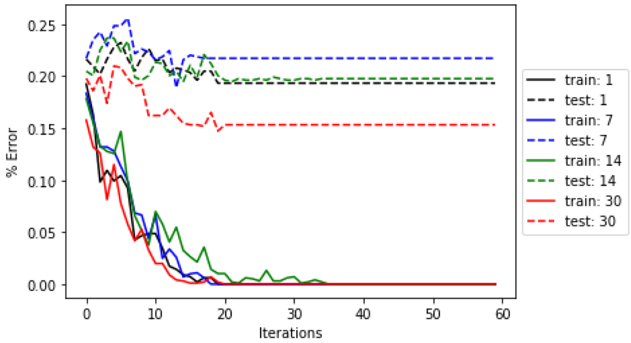


FIG. 9. Classification Error vs. Iterations for Varying Days Before and After Earnings Call

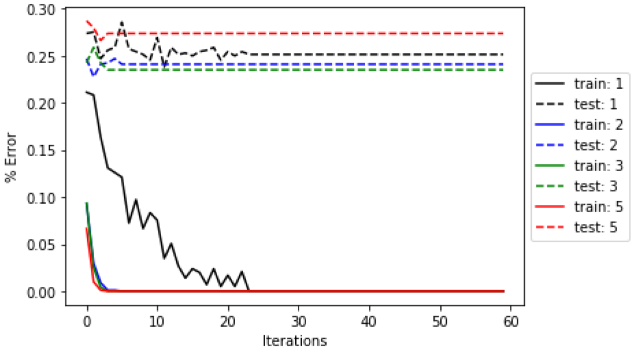


FIG. 10. Classification Error vs. Iterations for Hinge-Loss Minimizer using N-gram Feature Extractor and $n = 5$ Days

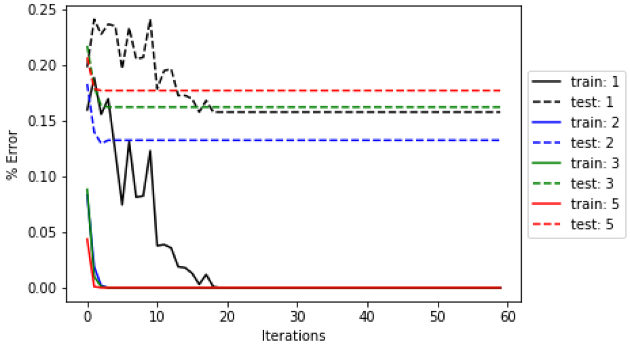


FIG. 11. Classification Error vs. Iterations for Hinge-Loss Minimizer using N-gram Feature Extractor and $n = 30$ Days

Firstly, we note that convergence is faster for n-grams model compared to unigram, due to the fact that a pairing of an n number of words adds extra intelligence to the model via correlations and thus faster in understanding (correctly or incorrectly) the text. Secondly, in terms of performance, the n-grams model does not perform better than the unigram model in the case where we are trying to predict in a 10-day window (Fig 10).

In comparison, we see that the bigram model performs much better than the unigram model when trying to predict in a 2-month window (Fig 10). This could again be due to the fact that there is some time delay from when the earnings call occurs and when the public fully understands its information and when the company starts actualizing statements in the earnings call. It is interesting that the bigram model performs better than the trigram or pentagram model. This could be due to the fact that the bigram is able to capture simple sentiments such as "not good" and "going well", while the trigram or pentagram model may involve company-specific terms (products, etc.) that can cause the feature space to be too specific and large.

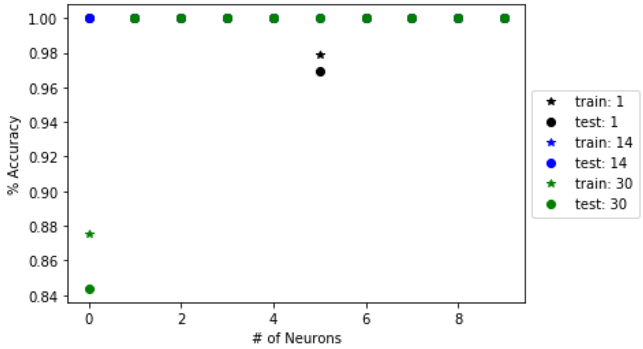


FIG. 12. Classification Accuracy vs. Number of Neurons for MLP neural network with varying n Days. Number in legend represents n.

In Fig 12, we plot the percent accuracy of classification of the MLP neural network with respect to different days before and after the earnings call date. We can see that even with just one layer, the accuracy of the predictor is extremely high at a few number of neurons in the layer. However, we note that the accuracy is 100% for most of the trials run for the plot. This perfect accuracy can be due to a multitude of reasons; it is difficult to tell whether overfitting has occurred or whether the test set was simply too small. However, it is clear that the neural network performs better than a simple linear classifier, which is to be expected due to the non-linear nature and complexity of text comprehension.

C. NLP with 10-K Documents

Both Jaccard similarity and cosine similarity showed a similar pattern (Fig 13). Below figure shows the cosine similarity result of BA's sentiment as an example. There is a significant change in "litigious" (red line), while "interesting" is not volatile (brown line). It hints that such word list may not be a strong factor for stock movement.

Similarity scores are used, and associated with forecasts of stock returns. Alpha is calculated based on sample size 10 (see table below). It seems to indicate positive earning surprises for "negative", "positive", "uncertainty" and "litigious", while "constraining" and "interesting" display the opposite. However, under small sample size, the result may be biased. Such bias can also be viewed through the bar charts of cumulative return by quintile and word category under sample size 5 vs 10.

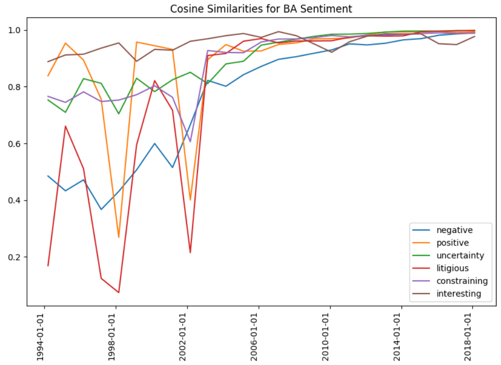


FIG. 13. Cos Similarities for BA Sentiment

Sentiment	α
negative	3.05
positive	3.04
uncertainty	0.57
litigious	1.64
constraining	-2.38
interesting	-2.34

TABLE I. Alpha Signal Based on Sentiment.

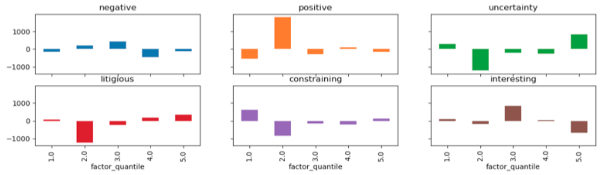


FIG. 14. Sample Size = 5

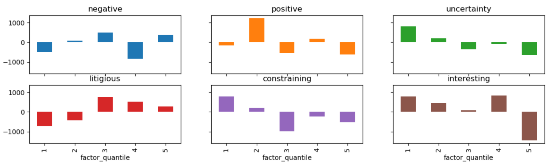


FIG. 15. Sample Size = 10

From the bar charts above, we notice that the most stable word list is "litigious" (red bars) where lower quintile has lower cumulative return no matter what sample size is. The most unstable word list is "uncertainty" (green bar). Others are in-between. In general, the NLP analysis on 10-K showed that the significant change in category ”Litigious” can generate earning surprises, while other categories do not have stable results to show strong signal when sample size changes.

The following figure shows that we can differentiate cumulative returns by word list. It is a result of trading strategy by longing changers and shorting non-changers in terms of word sentiment. Litigious word list can bring alpha on average, but it shows it is faded in current 3 years. From the graph, strategy on positive and negative word lists can generate alphas as well on average. However, since we saw that they are not stable by changing samples, we may need expand sample size to verify the effect. Other word lists do not show that they are strong strategies.

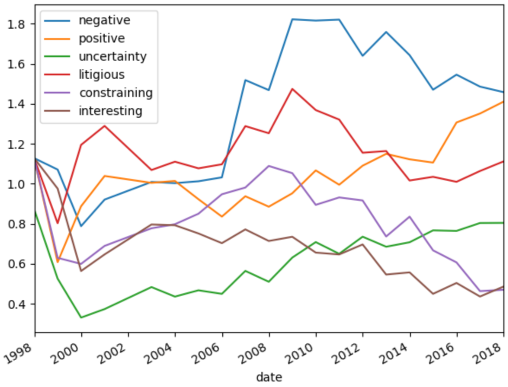


FIG. 16. Alpha Factor based on Sentiment

VI. CONCLUSION AND FUTURE DIRECTIONS

A. Prediction with historical data mining

Short term reversals around earnings announcements can be explained by market maker activity charging a premium for risk they undertake. We find that short term reversals around earnings announcements is a viable trading strategy. Abnormal returns over the CAPM model and Fama French Three-factor model are observed in this strategy. There is very little correlation with our strategy to the market. The strategy performs well under market instability and downturn. The performance is robust, working on both NYSE and NASDAQ exchanges.

This is not a strategy using a self-financing portfolio; it is dynamic and trade sizes relative to the full portfolio need to be further optimized. This information is necessary to accurately annualize returns. A future area to explore is peer group interactions during earnings announcements. Since this has be shown to increase reversal magnitude, only trading these events or weighting these trades heavier could yield more returns. Also, some research paper indicates that holding period returns on average were slightly higher during t to t+2 vs t -1 to t + 1. This holding period will need further investigation.

B. NLP with Earnings Reports

Stock movement prediction was tested using the earnings call transcripts to assess their validity and importance in stock movement. From using a simple linear classifier, we were able to determine that using an n-gram model with $n > 1$ performed better than a unigram model. Moreover, predicting stock movement over a greater range of time was more successful than immediate predictions, which is in line with the idea that time needs to pass for the information in the report to be understood and have its actions seen. Nevertheless, the error was not able to decrease after a certain threshold due to the inherent non-linear nature of the problem. This was visible from the neural network results in comparison to the linear model results.

Future directions involve increasing the data size and isolating certain features that are most correlated to stock movement. Moreover, other data can be used to bolster the model's understanding of a given company's potential to fail or succeed. While the earnings call transcript provide great information from the company itself regarding its results, the wording of the statements can have a bias, that including other textual data such as news could work towards reducing the bias. However, it may not necessarily be the best tactic to reduce bias, as the underlying nature of stock movement involves illogical, human-emotion driven aspects that cater towards how a given company is perceived rather than how it actually is objectively doing. Thus, understanding the importance of bias can be another area of further research.

C. NLP with 10-K Documents

Stock movement is also explored through changes of 10-K documents over time by word list from Loughran and McDonald sentiment dictionary for 10 stocks from sP100. The results are not stable except ”Litigious”. It indicates stocks with significant change in litigious words in the 10-K tend to under-perform in the following year. It can be understandable as it is a negative sign for required disclosure lawsuits. However, other word sentiment category does not show strong indication.

Such sentiment factor can be complemented to the previous FFM 3-factor model and CAPM model detailed in the section A. So is the earning transcript factor detailed in section B that can be added to the factor models explored in section A. Other future work can be performed by increasing data size including whole universe of SP100 stocks and 10-Q documents to verify whether other word sentiment can also be constructed as a return factor under other ways. For example, current negative and positive words are separately assessed and the results are not stable. We can explore another possibility to combine negative and positive sentiments by scaling and normalizing to verify whether it can be as a earnings factor.

VII. LINK TO CODES AND DATA

The zip files to the code and data are in the following link:

`https://drive.google.com/open?id=1itj-Lznd5fSHTiiyOD3_HIm8LACc1WpM`

[1] Lauren Cohen, Christopher J. Malloy, and Quoc Nguyen. Lazy Prices. *Academic Research Colloquium for Financial Planning and Related Disciplines*, 03 2019.

[2] Stefan Nagel. Evaporating Liquidity. *The Review of Financial Studies*, 25(7):2005–2039, 06 2012.

[3] Marko Poženel and Dejan Lavbič. Discovering language of stocks. *Frontiers in Artificial Intelligence and Applications - Databases and Information Systems X*, pages 243 – 258, 2019.