Corporate Earnings Call Sentiment Analysis & Financial Time-Series Forecasting:

Assessing if Sentiment Analysis Improves Time Series Prediction

Jay Vaidya[1] and Ruchi Kumar[2]

Northwestern University School of Professional Studies

633 Clark St, Evanston, IL, 60208

June 5, 2022

[1] Contact email: jvaidya@journeymcap.com
[2] Contact email: ruchikumar12@yahoo.com

Abstract

Accurately predicting stock market outcomes is a challenge given that security prices reflect all publicly available information. Deep learning techniques applied to stock market analysis typically rely on historical macroeconomic indicators, security prices, or proprietary datasets, unavailable for general public consumption, to forecast future prices ranging from microseconds to years into the future. Resource-constrained individuals or firms are at a disadvantage in this arena given the prohibitive cost of obtaining large, cleaned datasets suitable for use in deep-learning models. Open-source software and transfer-learning techniques, however, can help level the playing field. Applying cutting-edge Natural Language Processing (NLP) and transfer-learning techniques to a database of public-company earnings call transcripts, using self-annotated sentiment scores, we demonstrate that predictive accuracy can be improved in a timely and cost-efficient manner. Sentiment analysis of management answers are paired with standard machine-learning and deep-learning time-series methods to improve predictive power of whole-market performance, as represented by the Standard and Poor's 500 (S&P 500) index. By fine-tuning a pretrained masked-language model - Bidirectional Encoder Representation from Transformers (BERT) - on a self-annotated dataset, we further demonstrate that modest datasets can be used to enhance performance of open-source models (such as BERT) trained on much larger corpora.

Keywords: NLP, machine learning, sentiment analysis, finance, time-series, small sample, BERT, fine-tuning

Introduction

Efficient, well-functioning markets necessitate that security prices reflect all publicly available information. This fundamental principle is known as the Efficient Market Hypothesis,

which was set forth by Eugene Fama in his seminal 1970 paper, *Efficient Capital Markets: A Review of Theory and Empirical Work* (Fama, 1970). In an effort to outperform the market – as represented by an index such as the S&P 500 – professional and retail investors alike assess quantitative and qualitative data in hopes of deriving unique insights not currently reflected in the prices of a given security and exploiting those insights for a profit. By definition, then, smaller firms, or individuals, seeking to achieve consistently above-average performance over time are at a significant disadvantage to more well-resourced firms who can access proprietary research and data streams that provide greater, or quicker, access to material information.

More recently, investment institutions have begun adopting – with gusto – deep learning technologies for the purpose of developing predictive models capable of, once again, providing enhanced insights into the future performance of markets and their constituents. These models employ the use of fundamental macro- and micro-economic factors, the price history of the securities in question and any number of other related features that could be used to find predictable patterns of behavior qualifying as actionable intelligence for the purpose of deriving a profit.

The problem we sought to address in our research is if, and how, smaller firms, or individuals, can leverage opensource software, deep-learning methodologies and transfer-learning techniques to achieve improved predictive accuracy in stock market time-series forecasting, thereby narrowing the performance gap with more well-resourced firms. Specifically, in this research project, we set out to test whether Transformers architecture, as represented by the pre-trained BERT model, can be fine-tuned on hand-annotated datasets of 1,000 datapoints to perform sentiment analysis on management responses in publicly available

corporate earnings call transcripts, providing a rich feature set upon which to train predictive models.

Literature Review

In a comprehensive 2019 survey of deep learning methodologies as applied to time series forecasting, Sezer, Gudelek and Ozbyoglu asserted that "One rising trend, not only for financial time series forecasting, but for all intelligent decision support systems, is the human-computer interaction and NLP research. Within that field, text mining and financial sentiment analysis areas are of particular importance to financial time series forecasting" (Sezer, 2019). The authors went on further to predict the rise in the use of NLP, semantics and text mining-based hybrid models would constitute an increasingly common application of deep learning technology to time series forecasting (Sezer, 2019).

Recent work in the fields of deep learning and time-series forecasting have borne out the prediction of Sezer, Gudelek and Ozbyoglu, and our work is a continuation of prior efforts to explore the benefits of natural language processing methods for financial sentiment analysis and time series forecasting. The starting point for our research was to review the most relevant recent studies to understand the current state of this intellectual terrain and how our efforts could be directed to augment and add to the body of knowledge in a meaningful manner.

Previous work has focused primarily on sentiment analysis as it relates to speeches of members of powerful financial governance bodies, such as global central bankers who maintain responsibility for financial stability and governance of their respective territories. Most recently, Petropoulos and Siakoulis analyzed central banker speeches using NLP techniques combined with machine learning to find important signals in the text. By developing dictionaries of terminology, sourced from the available text documents, the researchers developed signals that

were translated into a sentiment index to forecast future financial market behavior. The NLP techniques utilized included tokenization (mono, bi and trigrams), term frequency-inverse document frequency (TF-IDF) and Latent Dirichlet Allocation (LDA). A variety of machine learning techniques were used, including random forests, XGBoost, SVM and DNNs, providing an overview of many AI techniques in finance. The researchers concluded that boosted tree-based models, developed using the sentiment indicators as input features, offered the most consistently accurate and robust forecasts. In their own words, they assert that "…using ML models can provide increased forecasting accuracy in future financial market disruptions" (Petropoulos, 2021).

In addition to explicit sentiment analysis being used as a feature for time-series forecasting, we also note that studies using topic-based models have been undertaken, wherein unstructured textual data is used as input for ML and DL time-series forecast models. In 2021, Livnat and Singh demonstrated that combining structured and unstructured feature sets can improve short-term time series forecasts and improve the classification of future structural breaks (i.e. significant upward or downward movements) in financial markets. Livnat and Singh applied supervised machine learning techniques to predict stock market returns of day t+2 using data through day t, inclusive of historical price information up to 22 days prior. Rather than predicting returns, the authors attempted to discern a large positive, large negative or other ("zero") returns. They incorporated three inputs: prior returns and volumes, analyst revisions of earnings forecasts and scoring of unstructured text news articles about a firm. Using software provided by Amenity Analytics, the authors demonstrated that the use of hybrid structured/unstructured feature-set models can result in statistically significant improvements in predictive accuracy of short-term

security movements in excess of a 3 percent accuracy improvement above random classifications based on class occurrence probabilities (Livnat, 2021).

We note two key research choices illuminated by our literature review of NLP as applied to time-series forecasting: 1) the decision to use high-frequency user-generated text, from sources such as micro-blogging Web sites including Twitter, or less frequent and slightly more structured text from sources such as earnings reports, news wire releases, analyst reports, etc.; 2) the decision to use structured versus unstructured text analysis techniques to develop features for forecast model inputs. Within the subset of structured text analysis approaches, Jacobs and Hoste in their 2021 paper, *Fine-Grained Implicit Sentiment in Financial News: Uncovering Hidden Bulls and Bears*, make a further distinction between course-grained and fine-grained sentiment analysis. As articulated by the authors, coarse-grained approaches entail the application of sentiment analysis to an entire document by taking into account all expressions of sentiment in the document, regardless of the target of the expression (Jacobs, 2021). Fine-grained sentiment analysis, on the other hand, requires the scrutiny of token-level data to infer target-based implicit sentiment analysis, which at times can differ markedly from the coarse-grained approach. To carry out their study as to the efficacy of such a fine-grained SA approach, the authors generated their own annotated dataset, each observation of which contained three parts: 1) a polar span – containing the implicit or explicit sentiment being voiced; 2) the target span – containing the sentiment target; 3) polarity – expressing positive, negative or neutral sentiment towards the target. Somewhat surprisingly, the authors reported the following conclusions, "Regarding the fine-grained triplet experiments, the large performance gap of our dataset compared to the explicit sentiment review dataset of Wu et al. [15] showed that the current state-of-the-art model in explicit sentiment is not sufficient for our SENTiVENT dataset" (Jacobs, 2021). In other

words, this research indicates that the potential benefits of fine-grained annotation for financial sentiment analysis may be muted, at best, meaning that more coarse-grained annotated training datasets may be sufficient to provide less well-resourced groups with comparable SA results to their more well-resourced peers.

As it pertains to the research objectives of this report, the goal of determining the efficacy of transfer-learning and fine-tuning pretrained models on modest-sized datasets requires that we understand the prior work undertaken with respect to such endeavors. One such research report was published by Ezen-Can in 2020 and detailed a comparative study of a Long Short-Term Memory architecture with that of a pretrained BERT model based on the Transformers architecture, for the purpose of intent classification in text, as it pertained to chatbot development. Ezen-Can used a dataset containing approximately 23,000 datapoints, varying the size of the training, validation and test sets, and compared the classification accuracy of the two model architecture types across each configuration. Ezen-Can's results consistently showed outperformance of the simpler LSTM architecture, reaching a peak test-set accuracy of 70 percent as opposed to roughly 67 percent for the BERT model. Ezen-Can tentatively asserted that overfitting may result in degraded performance for the BERT models, in the instance of small-dataset training (Ezen-Can, 2020).

The issue of applying BERT fine-tuning to limited datasets, noted above, however, is addressed in a number of subsequent studies that come to different conclusions, noting the importance of model hyperparameter specifications in ensuring optimal model performance. Specifically, in their 2021 paper, *Revisiting Few-Sample BERT Fine-Tuning*, Zhang et al. note that most BERT fine-tuning studies propose replacing the final output layer of the model with a new task-specific layer and then retraining the entire model; this approach introduces new

sources of randomness with respect to the initialization of new parameters and can also lead to a degradation in intermediate features learned by the model. The authors go on to recommend a number of potential remedies, such as regularization techniques including pre-trained weight decay, layer-wise learning-rate decay and transfer learning via intermediate tasks on larger, public datasets as means for achieving better results on small datasets (Zhang, 2021).

Finally, having established, to our satisfaction, the feasibility of our research objectives, we explored the techniques and methodologies recommended for the production of self-annotated datasets for fine-tuning our model and, ultimately, engaging in sentiment analysis of our corpus. Given the limited resources available to our team (comprised of two researchers), we sought to determine the reliability of manually annotated datasets produced by a limited number of parties. Research by Kiritchenko and Mohammed indicates that reliable sentiment annotation agreement and rankings can be achieved with as few as two or three independent annotations per data entry (Kiritchenko, 2016). Furthermore, a sentiment annotation scheme developed by Batanovic, Cvetanovic and Nikolic, which has been adapted for use in our research, reported intra-group sentiment agreement for polarity direction of 96.6 percent; this same piece reported four- and six-group sentiment classification agreement of 95.5 and 92.2, respectively (Batanovic, 2020). These findings and methodological setups will be further discussed in the methodology section of this report.

Data

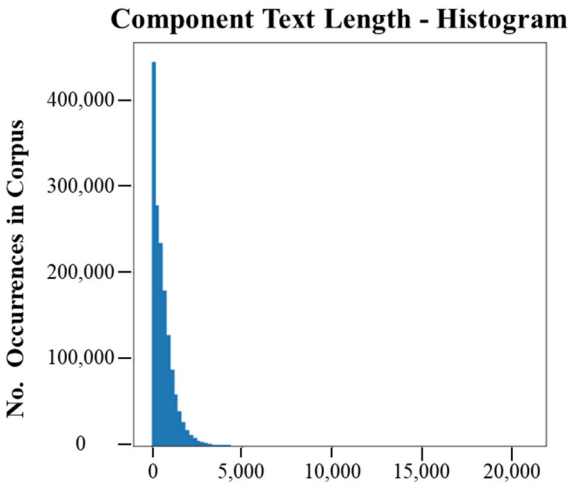
Two distinct datasets were procured in order to generate the feature sets required of the research conducted in this report. The first dataset was comprised of the split- and dividend-adjusted time-series of the S&P 500 index (as represented by the ETF, Ticker: SPY). The second dataset comprised the corpus of text documents to be used for sentiment analysis. The text

corpus dataset was sourced through the Wharton Research Data Services (WRDS) database, which consolidates a broad array of data for use by researchers in many disciplines, and the S&P 500 time-series data was sourced from Yahoo! Finance.

For the text corpus, we have selected to use quarterly earnings report transcript data for U.S. publicly listed companies that are sourced from CapitalIQ, a contributor to the WRDS database. CapitalIQ is a Standard & Poor's business which delivers comprehensive fundamental and quantitative research to investment firms. One benefit of using WRDS-based data is that the organization conducts rigorous data review and validation so users may confidently conduct research and build models. With respect to the portion of the transcripts used to build our corpus, we focused on the question-and-answer portion of the earnings call transcripts. Specifically, the corpus was built using only the answer portion of these dialogues, as the goal of the research is to use NLP techniques to intuit the sentiment being conveyed by senior management staffs and use the sentiment classifications as feature inputs for time-series forecasts. Each question and each response are assigned a unique identifier in the database managed by CapitalIQ that is timestamped, recorded and numbered in order of the unfolding of the conversation.

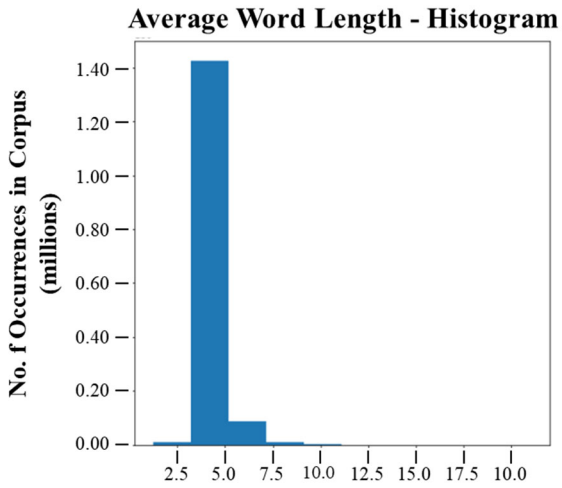
The dataset of all earnings call transcript entries for U.S. domiciled companies was merged, using the unique company identifiers, with another CapitalIQ dataset containing industry information on each organization in the dataset – permitting for segmentation by industry later in our analysis. The beginning text corpus dataset contained 2.17 million entries and 63 columns, or data series, per entry. After removing duplicate entries – identified as those with the same unique 'transcriptid', 'componentorder' and 'componenttext' entries – a total of 1.54 million records remained. These 1.54 million records were comprised of the following: 54,247 unique transcript id's; 2,304 unique company id's; and 49 unique industries. These

entries spanned across the period from January 17, 2006, to December 10, 2021. Below are presented several of the key findings from the initial EDA performed on this dataset:



Source: WRDS – Capital IQ Transcripts Database

(Figure 1)

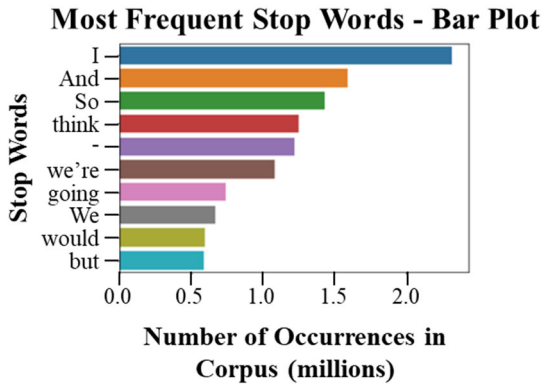


Source: WRDS – Capital IQ Transcripts Database

(Figure 2)

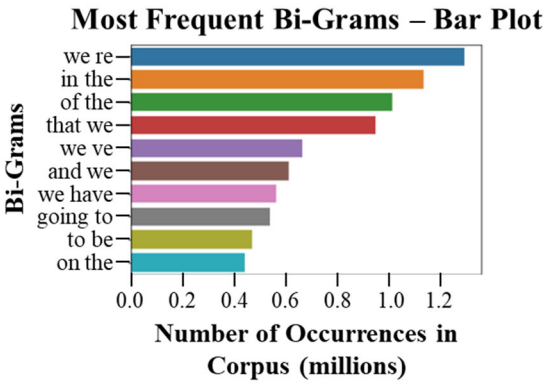
Review of the distribution of the number of tokens per entry, or length, shows that the minimum ‘componenttext’ length is 2 tokens, the maximum is 20,912 and the median is 458. Additionally, as a proxy for the complexity of the language being employed in the transcripts, we plotted a histogram of the average word length for each ‘componenttext’ entry in the dataset; as can be seen, average word length ranges between 3 and 5 characters, a crude, though potentially insightful, indication that complex language isn’t frequently employed, though corporate and industry-specific jargon is assuredly prevalent throughout the dataset.

The next step in our analysis of the text corpus was to remove ‘stop words’ – or those most common words that add little value to textual analysis – using the NLTK package. Having removed the stop words, we then plotted the most commonly occurring uni-, bi- and tri-gram occurrences throughout the dataset. Additionally, we sought to understand the distribution of entries, by industry, and generated a list of occurrences by industry to determine relative distributions.



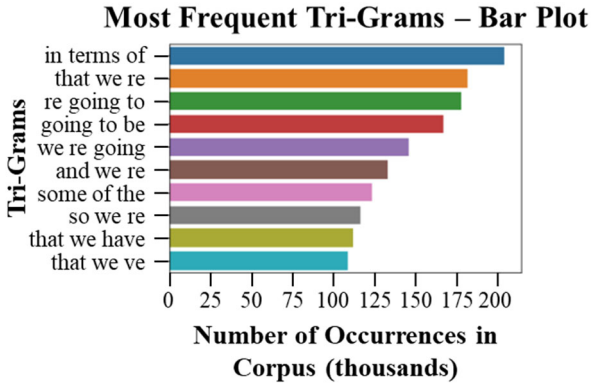
Source: WRDS – Capital IQ Transcripts Database

(Figure 3)



Source: WRDS – Capital IQ Transcripts Database

(Figure 4)



Source: WRDS – Capital IQ Transcripts Database

(Figure 5)

A cursory analysis of the most common uni-, bi- and tri-grams displays several interesting patterns. First, the top words shown in the uni-grams are disproportionately representative of qualifying terms, or those used in conversation to hedge a response, such as 'so', 'think', 'would' and 'but'. The top bi-gram and tri-grams, while also containing qualifying terms, however, also demonstrate verbiage representative of shifts in conversation to clarify historical occurrences – such as 'in terms of' and 'that we have' – to a forward shift – such as 'we're going to' and 'going to be'. Although these terms in and of themselves may not offer strong predictive value, together they indicate that earnings transcripts frequently contain discussions of historical organizational performance as well as forward-looking statements that

could be indicative of management outlooks into the future, which is critical to the success or failure of this research project.

Corpus Entries by Industry (Top 20): Jan. 2006 thru Dec. 2021		
Industry Sector	**No. Records**	**% Total**
Information Technology	146,126	9.95%
Holding Company	122,484	8.34%
Property & Real Estate	121,406	8.26%
Capital Goods	108,145	7.36%
Energy	107,448	7.31%
Health Care	94,616	6.44%
Retailing	90,454	6.16%
Consumer Products	80,592	5.49%
Media & Entertainment	54,096	3.68%
Transportation	47,712	3.25%
Commercial & Professional Services	45,948	3.13%
Chemicals	45,700	3.11%
Automobiles & Components	36,114	2.46%
Finance Company	34,048	2.32%
Electric	29,950	2.04%
Aerospace & Defense	29,024	1.98%
Midstream Energy Companies	28,940	1.97%
Metals & Mining	27,007	1.84%
Hotels & Gaming	26,250	1.79%
Building Materials	23,690	1.61%
Top 20 Total	1,299,750	88.48%

Source: WRDS – Capital IQ Transcripts Database

(Table 1)

Looking at the distribution of data by industry, the decision was made to use only the top 20 industries for this research project; specifically, the top 20 industries represent approximately 88 percent of the total text corpus, and also provide input from a wide diversity of industries. Given that each industry, over each pre-specified period of time, will be assigned a sentiment score to be used as a feature input for our time-series prediction model, we chose not to use under-represented industries that may offer skewed feature input scores due a significantly lower number of constituents and hence lower overall diversity within the industry group. The final text corpus dataset, comprised of only the top 20 industries, contained 1.30 million entries.

With respect to the time-series data sourced for the S&P 500 index, retrieved from Yahoo! Finance, we collected the data at the daily interval. Daily data permits for aggregation at varying time-steps, as required of our research; but, for the purpose of this study, we chose to reindex the data at a monthly interval in order to preserve computational resources that could be over-taxed by the creation of our sentiment-ratings feature set at a shorter interval (such as weekly). As an added benefit, monthly intervals best reflect the cadence of receipt of earnings call transcripts, which most typically occur following quarterly periods, with varied start and end dates. From the time-series data, we calculated the total return of the index, at our chosen interval, using the simple return of the adjusted closing prices as of the final day of the period and the final day of the first day preceding the period (i.e., the opening price of the period in question). While the time-series data was originally sourced from the onset of the establishment of the SPY exchange-traded-fund (ETF), which tracks the S&P 500, in January 1993, thru the end of December 2021, we ultimately used 11 years of data for our analysis to ensure the quantity of sentiment data, per industry, per time-step, was of sufficient depth to provide quality signals upon which to train our models. Our final train, validation and test datasets spanned the period of January 2011 thru December 2021.

Methods

The methodology employed for determining the efficacy of using earnings call sentiment analysis to improve time-series forecasts of the S&P 500 entails two distinct, yet complementary, phases. In the first phase, a randomized subset of the un-annotated text corpus is selected for annotation and assignment of sentiment ratings. Those ratings are then used to fine-tune a pre-trained NLP, transformer-based, model that will be used to predict the sentiment ratings of the remainder of the dataset. Once sentiment ratings have been assigned to the remaining dataset, the

sentiment ratings are aggregated at the industry-level – recall that there are 20 distinct industries represented – and then segmented into distinct monthly time periods. For each period interval, each industry segment will be assigned a single sentiment classification rating of positive (1), negative (-1) or neutral (0). For a given time-step interval, a single rating class must be assigned to each industry segment. To accomplish this task, we've developed a proprietary aggregate rating method that we have dubbed a 'negative sentiment tripwire,' which will be explained below in further detail.

Once the sentiment data and the S&P 500 returns data have been prepared, two forms of models are be applied to the resultant feature set, time-series classification transformer-based neural network models, as advocated by Li et al. in their paper *Incorporating Transformers and Attention Networks for Stock Movement Prediction*, and Bagging Tree Models (Li et al, 2022). The goal of the model applications to the feature sets is to forecast whether the returns in the next period will be positive (meaning greater than or equal to zero) or negative. The primary advantage of employing tree-based bagging models as one model type is that feature selection approaches can also be applied, wherein feature categories can be randomly permutated and the model refit to determine the impact, thereby highlighting which features – or clusters of features – generate the highest predictive value for the model.

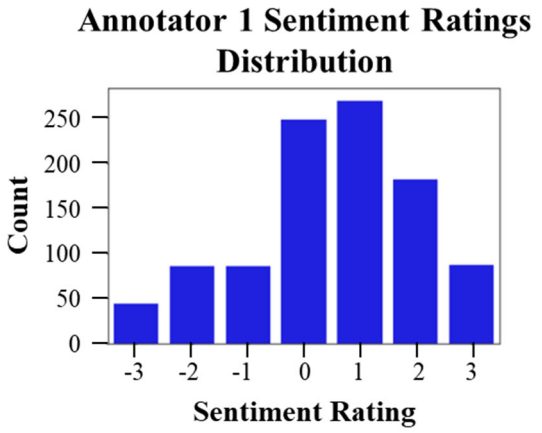
In order to prepare the text corpus dataset for use in the transformer-based sentiment classification model, we needed to self-annotate a subset of the records. To do so, we created and followed an "Annotation Guidelines" document (See Appendix A for details). This document was intended to guide the annotators towards consistent ratings to help maximize the usefulness of a relatively small self-annotated dataset. 1,000 datapoints were randomly selected, 50 from each of the 20 industry categories, for the annotation dataset. These 1,000 datapoints were then

randomly shuffled to avoid groupings of similar, or sequential, text distorting annotator ratings. The annotations ranged from the integers of -3 to +3, for a total of 7 distinct classes (See Appendix A for details).

There were a handful of challenges that had to be carefully addressed, and where appropriate, mitigated, in assigning categories to the dataset. To begin, we had to determine which entity, or foci, was the target of our sentiment analyses; stated differently, was the purpose of the sentiment rating to gauge an executive's perception of the macroeconomic outlook of the economy as a whole, or was the focus specific to the executive's firm. Executives frequently commented on both topics, as in many instances they are deeply intertwined, though sometimes opposing, as in the statement "The economic backdrop is becoming cloudier, but we remain excited about our prospects in the next quarter." For the purpose of this research, we determined that the focus, when possible, would be on the outlook with respect to the firm in question, and only in the instances when no mention of company-specific sentiments were expressed, but macroeconomic sentiments were expressed, would macroeconomic considerations take precedence over company-specific considerations.

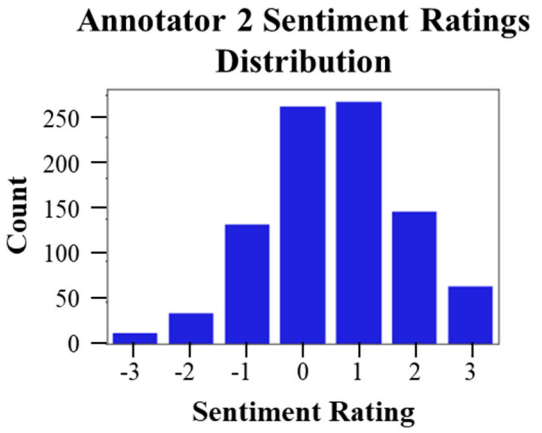
Furthermore, in many instances, multiple sentiments were expressed in one entry – a finding we believe is particularly endemic to corporate leadership, whose words are carefully parsed by teams of analysts, and are hence conscientious of discussing overtly negative topics or outcomes. Accordingly, these same executives often attempt to offer a positive 'spin' in many occasions that would not seemingly warrant such positivity. For example, while acknowledging underperformance in one business segment, an executive might also describe outperformance in another segment, which the executive believes can compensate for, or mitigate the impacts of underperformance elsewhere. Or, while describing historical underperformance, an executive

may express significant optimism for future performance. Also, given that executives are eager to present their companies in the best light, it was expected that the sentiment would skew toward a higher prevalence of positive sentiment ratings. This assumption was borne out by the distribution of annotations generated by the two researchers in this research project:



Source: Annotator 1 Hand-Annotated Dataset

(Figure 6)



Source: Annotator 2 Hand-Annotated Dataset

(Figure 7)

Additionally, decisions bearing on how mixed-sentiment entries, neutral sentiment entries, and other ambiguities were handled are explained further in Appendix A to this document.

Discrepancies between the annotators were resolved as per the Annotation Guidelines document (See Appendix A for details). Of the 1,000 annotation rankings, only 168 were two or more categorical rankings apart. Of those 168, 116 were only two categories apart. Of the remaining 52 annotations, 40 were three apart, 12 were four apart and none were five or six apart. For rankings that were two or fewer categories apart, the categories were averaged and rounded up (down) to the next whole number (+1.5 becomes 2, and -1.5 becomes -2). Rounding to the extremes of the sentiment ranking scale was decided upon in order to try to force the model to lean towards assigning positive and negative rankings, and away from the neutral rankings. For the 52 annotations that were three or more categories apart, the authors reviewed

each one as a group and decided on a new category, based on a re-reading of the entries and the Annotation Guidelines.

With the annotations completed, the sentiment-analysis model was trained on the 1,000 datapoint sentiment-rated text corpus using the transformer-based BERT-base-uncased pretrained masked-language-model (MLM) provided by the organization Hugging Face. According to the Hugging Face documentation, the pretrained BERT model "…was trained on 4 cloud TPUs in Pod configuration (16 TPU chips total) for one million steps with a batch size of 256. The sequence length was limited to 128 tokens for 90% of the steps and 512 for the remaining 10%" (bert-base-uncased, 2022). In order to fine-tune the model on the downstream task of sentiment analysis, two layers were appended to the base pre-trained BERT model: the first layer aggregates the features of the last layer of the transformer model and passes them to a layer of neurons of equal size – 768; the second, and final, layer is comprised of the number of output categories being sought by the model – in this case, three for negative, neutral and positive sentiments – and fed into a softmax, or sigmoid, function in order to determine the predicted class of the example.

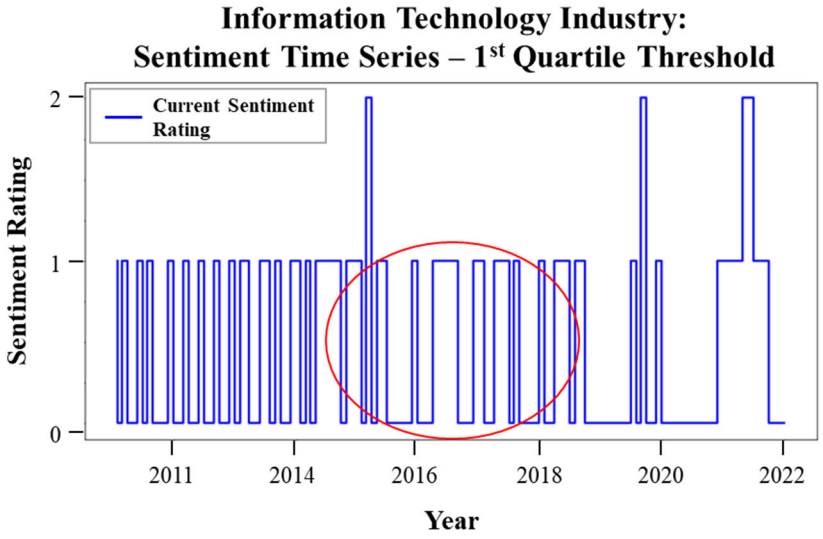
Among the most critical aspects of training the transformer-based NLP sentiment analysis model is the avoidance of severe overfitting during fine-tuning. Specifically, pre-trained transformer-based models such as BERT can suffer from a phenomenon known as 'catastrophic forgetting' when the models are asked to adapt to a new task (Araci, 2019). To compensate, it is generally recommended that no greater than four epochs be undertaken during training and validation; we adhered to this advice in our work.

The fine-tuned sentiment analysis model was next used to make predictions on the remaining, unseen text corpus dataset, and the data was then subdivided into the 20 respective

industries, each of which was condensed into a single point-estimate of the prevailing sentiment within that industry over the preceding month. To condense all of the datapoints contained in a given time-step, for a particular industry, into one rating, we developed and applied a proprietary method known as the 'negative sentiment tripwire.' Specifically, the tripwire functioned by comparing the number of negative sentiment entries in a given time-step and industry to a preset threshold; if the prevalence of negative sentiment entries in the current time-step exceeded the threshold, the rating for the current time-step and industry would be assigned the 'negative' or zero class. If the negative sentiment tripwire was not triggered in the current time-step, however, a sentiment rating was assigned by averaging all of the sentiment ratings in the period and rounding to the nearest class – the 'neutral' class was assigned to one and the 'positive' class was assigned to two.

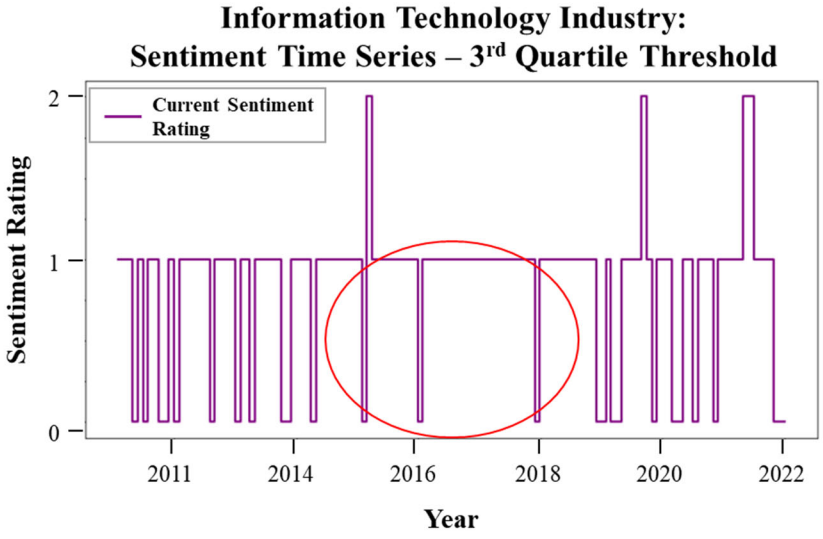
We tested both static and dynamic tripwire thresholds in our research, developing four distinct tripwire thresholds, which if surpassed would trigger a negative sentiment rating, in order to test how the sensitivity of the tripwire impacted the feature sets being generated and thus the signal being transmitted and interpreted by our downstream prediction models. The negative sentiment tripwires we employed were the following: 1) a static, hard threshold of five negative sentiment readings in the current time step; 2) the dynamic threshold calculated as the first quartile percentage prevalence of negative sentiment readings across the training set for a given industry; 3) the same as the previous, but the second quartile; 4) the same as the previous, but the third quartile. The intuition behind employing dynamic thresholds is that negative sentiment assignation for a particular industry and time step should be understood as searching for a rise in negative sentiment entries as compared to some 'normal' level – in this case, the quartile thresholds define the levels of normality, above which an abnormal level of negative sentiment is

said to be detected. The outcome of applying the four negative sentiment tripwire approaches, elaborated above, was the production of 4 distinct feature sets (one for each threshold approach), two of which are presented below to highlight how the feature sets changed across thresholds:



Source: Proprietary sentiment rating method applied to WRDS – Capital IQ Transcripts Database

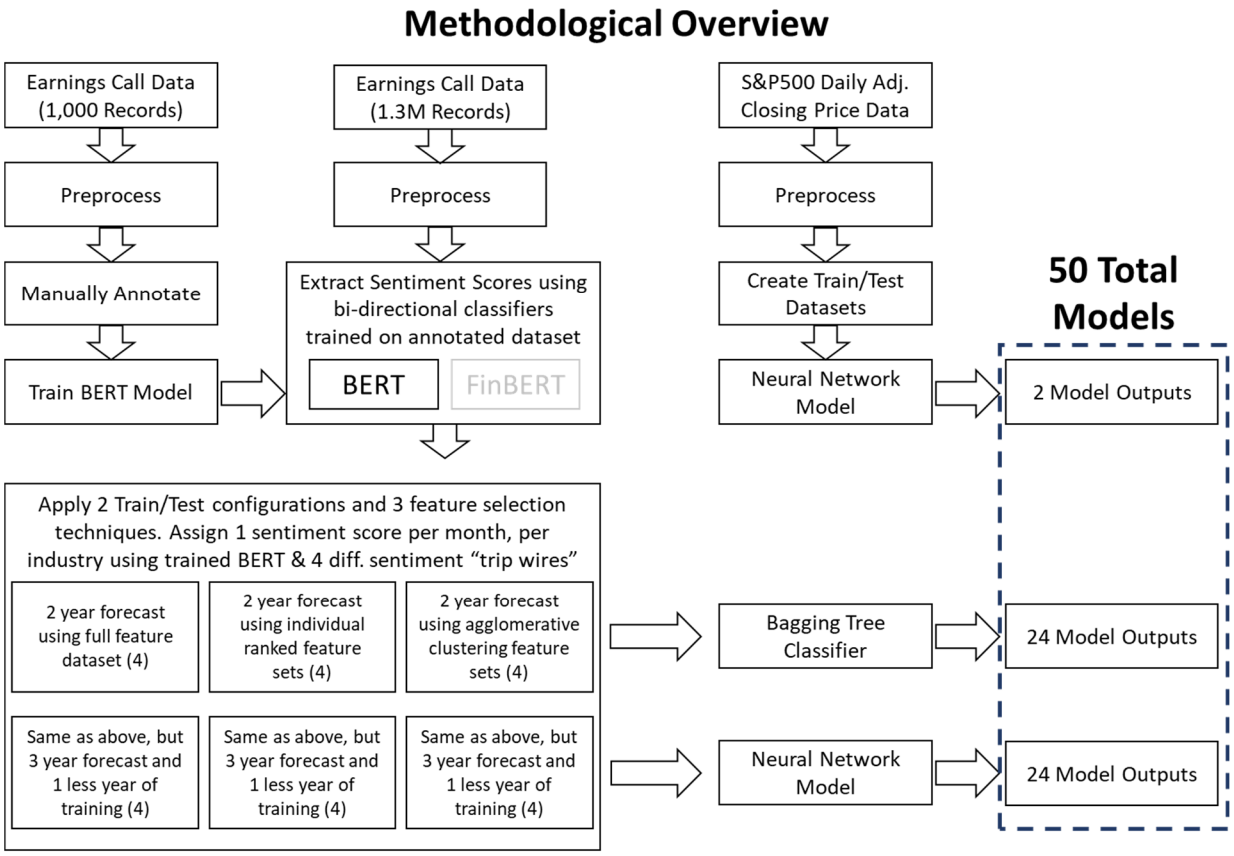
(Figure 8)



Source: Proprietary sentiment rating method applied to WRDS – Capital IQ Transcripts Database

(Figure 9)

In addition to producing four distinct feature sets, we also tested our models on two different train/test configurations: 1) 9 years of training data and 2 years of test data; 2) and 8 years of training data and 3 years of test data. Finally, we also tested our models on three different feature-selection approaches: 1) full-feature sets wherein all 20 industry features are fed into the models; 2) individual feature-selection wherein only those individual features positively adding to log-loss were included; 3) agglomerative feature-selection wherein only those clusters of features positively adding to log-loss were included. Together this resulted in 24 feature combinations (4 feature sets x 2 train/test configurations x 3 feature selection approaches), which were tested on two distinct model types – tree-based bagging models and transformer-based NN's for time-series classification – for a total of 48 total model-feature configurations trained and tested. As a final step, we generated two transformer-based NN models, one for each of our two train/test configurations, that were trained using a rolling window of six months of lagged S&P 500 returns as a benchmark for the performance of our sentiment-trained predictive models.



(Figure 10)

Results

As an outcome of our two-step methodology – feature generation using fine-tuned BERT models and then time-series prediction using the generated features – there are two important sets of results upon which to report. First, we'll detail the results of our fine-tuning of pre-trained masked-language-models for sentiment analysis; second, we'll detail the results of our time-series prediction and classification of the S&P 500's next month returns using the generated sentiment-rated feature sets.

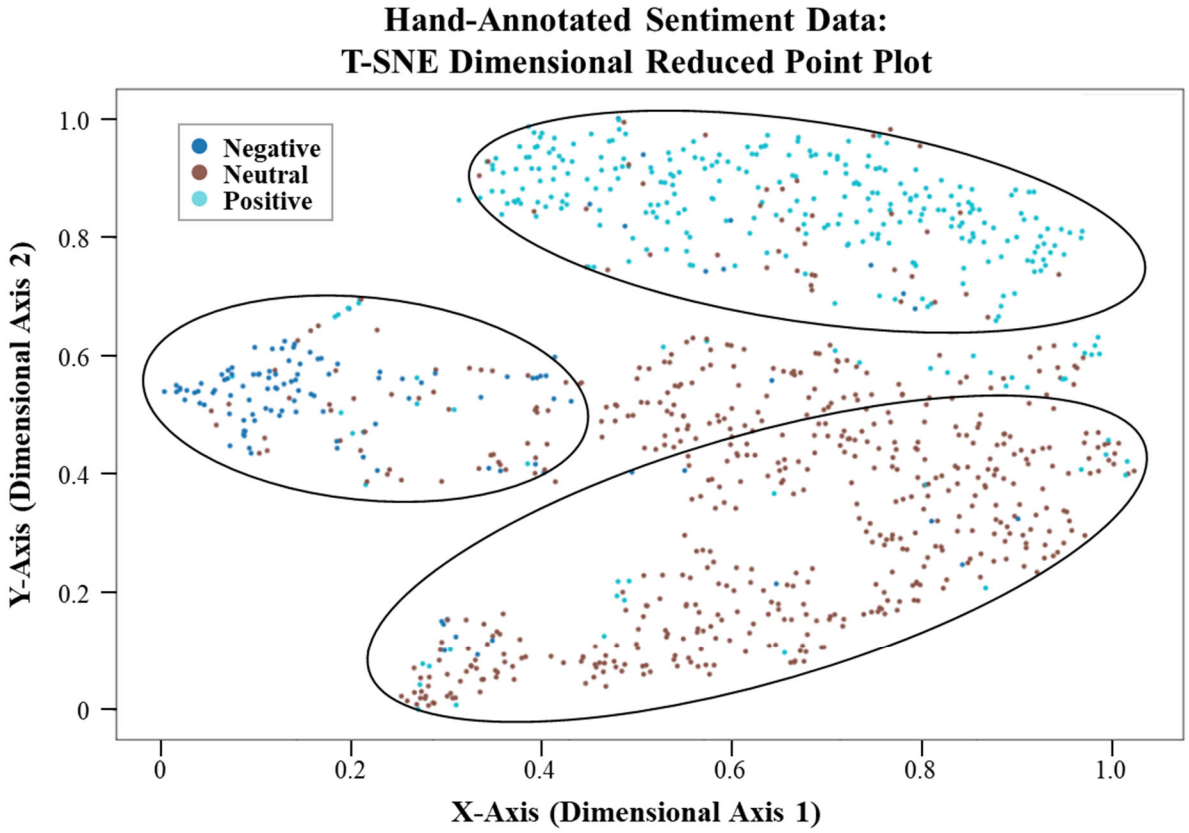
The first-stage sentiment-rating results were generated using a pre-trained BERT model, provided by HuggingFace, and the PyTorch library applied to a local RTX 3070 Nvidia GPU. The pre-trained BERT tokenizer was also used, and the maximum number of tokens per entry

was set to 128. Using this setup, the first trial we undertook was on an unfrozen BERT model with every layer able to be trained, without restriction; we generated an output layer with seven neurons reflecting the seven integer sentiment categories used during annotation (-3 to +3). This initial trial achieved a maximum validation accuracy of 41.8 percent.

For the second trial, we mapped the seven sentiment categories to only three (positive, negative or neutral); categories -1, 0 and +1 were classified as neutral, with the poles of the remaining integers being mapped as positive, or negative sentiments, respectively. Retesting this model setup using the unfrozen pre-trained BERT model produced a validation accuracy of 71.6 percent and a test set accuracy of 69.0 percent. This result compares to the class prevalence of approximately 55 percent for the neutral category, indicating the model had learned to discriminate classes more accurately than simply assigning the most-prevalent category to each observation being predicted.

In order to validate this assertion, we applied the t-distributed stochastic neighbor embedding (t-SNE) methodology to engage in dimensionality reduction of the 768 dimensions of the final global pooling layer of the fine-tuned BERT sentiment classification model. By passing each of the 1,000 hand-annotated sentiment rated datapoints through the fine-tuned model, recording the results of the final global pooling layer dimensional mapping and then using t-SNE to reduce the mappings to two dimensions, we can visualize the unfurled manifold of the dataset and visually scrutinize the efficacy of the sentiment-rating model. As can be seen in the image below, the sentiment classification model is able to discern the difference between negative, neutral and positive sentiments quite well. Perhaps unsurprisingly, the negative sentiment datapoints (dark blue), which are the least-frequently occurring in the dataset, are the most muddied in terms of clear separability from the other two classes. We believe this points to a

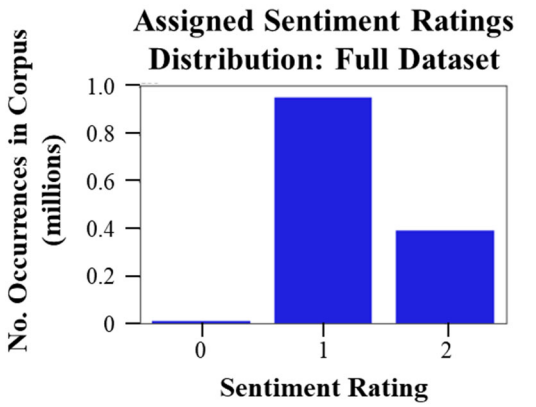
phenomenon we detected during our work - namely, that corporate executives are loathe to make unabashedly negative statements, using qualifying terms and shifting topics to more positive sentiments as a way to obfuscate when negative sentiments have been expressed during an earnings call.



Source: Fine-tuned BERT model trained on 1,000 datapoint hand-annotated sentiment rating samples from WRDS – Capital IQ Transcripts Database

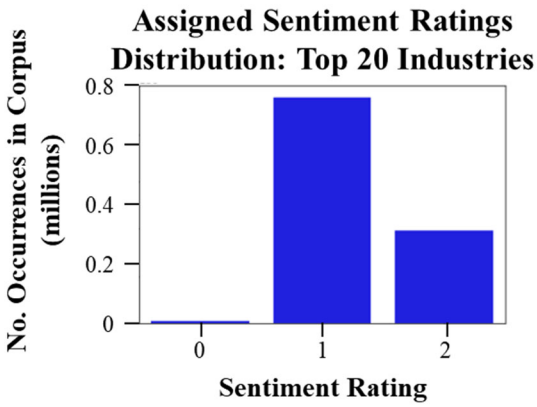
(Figure 11)

Satisfied that the fine-tuned sentiment rating model was producing reasonable classifications, we then used the model to make sentiment predictions on the full text corpus dataset containing approximately 1.3 million entries spanning the period of 2011 thru 2021, the results of which can be seen below:



Sentiment ratings for BERT model required mapping to positive figures: 0 is 'negative'; 1 is 'neutral'; and 2 is 'positive'. Respective prevalence are as follows: 6,232; 956,629; 392,768.

(Figure 12)



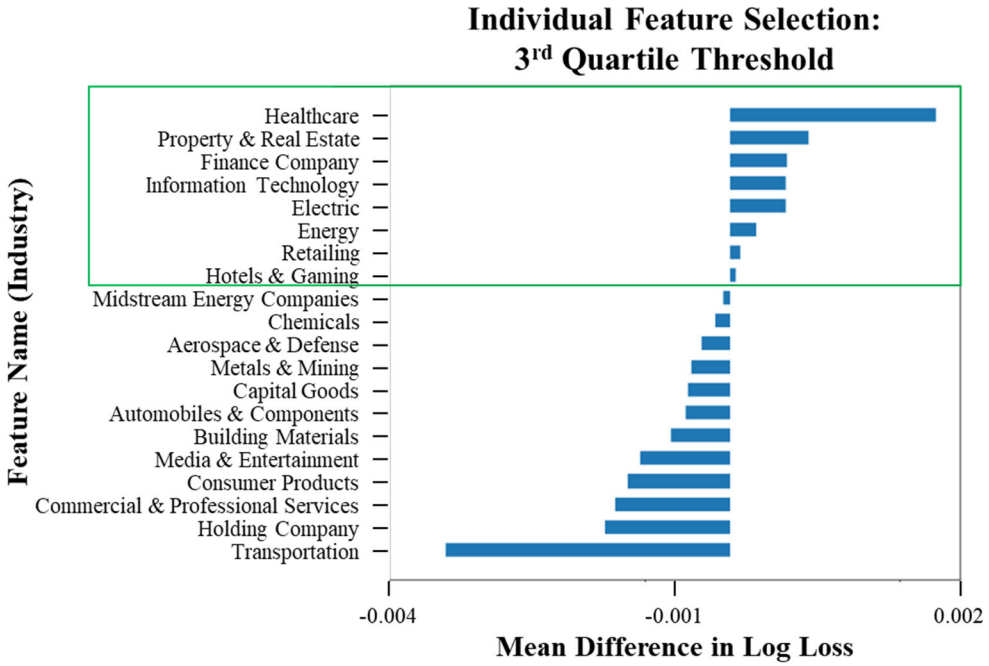
Sentiment ratings for BERT model required mapping to positive figures: 0 is 'negative'; 1 is 'neutral'; and 2 is 'positive'. Respective prevalence are as follows: 4,660; 701,177; 293,926.

(Figure 13)

Using the proprietary negative sentiment-tripwire approach, discussed in the "Methods" section of this report, we then condensed the sentiment ratings of the top-20 industries into a single rating, per month, over our scrutiny period – resulting in 20 sentiment rating classifications (or, features) for each of the 132 monthly time steps.

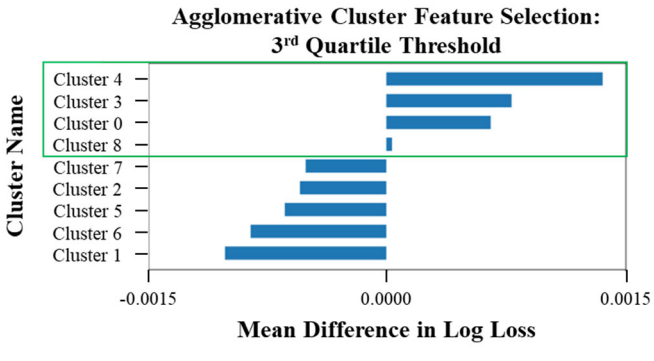
Before undertaking predictions, we engaged in feature selection. Using single- and clustered-feature approaches, we applied random permutations to our feature sets, one feature, or cluster, at a time, and tested how those random permutations impacted the log loss of our predictions - in doing so we could uncover those features that added (detracted) the most value to our predictions. By removing features deemed to detract from predictive accuracy from our feature sets, we scrutinized how de-noised feature sets compared to full 20-feature predictive performance. The feature sets produced using the single- and clustered-feature selection approaches, on the third-quartile negative sentiment tripwire feature sets, consistently produced the best model results, achieving a maximum 70.8 percent test-set accuracy on our 'version 1'

(trained on 2011-‘19 data and tested on 2020-’21) dataset and 75.0 percent test-set accuracy on our ‘version 2’ (trained on 2011-‘18 data and tested on 2019-’21) dataset.



Observations within a feature (i.e., data column) are randomly permutated and then all features are fit on a bagging classifier using 10-fold cross validation. This process is repeated for each feature, saving the results of every permutation. Mean difference is calculated as the mean change in log loss over the permutations of each feature. Positive values indicate improvement, while negative values indicate degraded predictive performance.

(Figure 14)



(Figure 15)

Cluster ‘0’ Components: Information Technology, Consumer Products, Aerospace & Defense

Cluster ‘1’ Components: Transportation, Hotels & Gaming

Cluster ‘2’ Components: Retailing, Capital Goods

Cluster ‘3’ Components: Property & Real Estate, Metals & Mining

Cluster ‘4’ Components: Electric

Cluster ‘5’ Components: Media & Entertainment, Chemicals

Cluster ‘6’ Components: Holding Company, Health Car, Midstream Energy Companies, Finance Company

Cluster ‘7’ Components: Commercial & Professional Services, Building Materials

Cluster ‘8’ Components: Energy, Automobiles & Components

A selection of our best results achieved using both bagging tree classifiers and time-series classification transformer-based NN models are presented here (See Appendix B For Full Results Tables):

Version 1 Models - Tested On 2 years of Data (2020 - 2021)							
Feature Set Employed	**Model Type**	**Overall Accuracy**	**Neg. Pred. Correct**	**Total Neg. Preds.**	**Pos. Pred. Correct**	**Total Pos. Preds.**	**Total Test Preds.**
Baseline Train Set Data (S&P500 Actual Returns)		100.00%	N/A	31	N/A	77	108
Baseline Test Set Data (S&P 500 Actual Returns)		100.00%	N/A	8	N/A	16	24
6 Months Lagged S&P 500 Returns	Neural Net	62.50%	1	3	14	21	24
Individually Ranked Features - 3rd Quartile Threshold	Bagging Tree	70.83%	3	5	14	19	24
Cluster Ranked Features - 2nd Quartile Threshold	Bagging Tree	70.83%	2	3	15	21	24
Cluster Ranked Features - 3rd Quartile Threshold	Bagging Tree	70.83%	4	7	13	17	24
Individually Ranked Features - Hard Threshold	Neural Net	75.00%	2	2	16	22	24

(Table 2)

Version 2 Models - Tested On 3 years of Data (2019 - 2021)							
Feature Set Employed	**Model Type**	**Overall Accuracy**	**Neg. Pred. Correct**	**Total Neg. Preds.**	**Pos. Pred. Correct**	**Total Pos. Preds.**	**Total Test Preds.**
Baseline Train Set Data (S&P500 Actual Returns)		100.00%	N/A	28	N/A	68	96
Baseline Test Set Data (S&P 500 Actual Returns)		100.00%	N/A	11	N/A	25	36
6 Months Lagged S&P 500 Returns	Neural Net	61.11%	1	5	21	31	36
Individually Ranked Features - 2nd Quartile Threshold	Neural Net	66.67%	2	5	22	31	36
Cluster Ranked Features - 3rd Quartile Threshold	Bagging Tree	75.00%	4	6	23	30	36

(Table 3)

Of the 50 total model configurations tested in our research, the bagging tree classifier models demonstrated the greatest consistency in predictive performance, sensitivity to the less prevalent prediction class (‘0’ or negative next month’s return) and the least likelihood of overfitting the data. Six of the 24 NN models fit on our feature sets generated zero ‘negative’ predictions for next month’s returns – simply defaulting to predicting the most prevalent class (‘positive’ next month’s return), despite the application of dropout rates as high as 50% during the training process to mitigate overfitting. Correspondingly, none of the 24 bagging tree classifier models fit in our research resulted in zero ‘negative’ predictions for next month’s returns, indicating a more robust mapping of the feature-space achieved by the bagging models, as a whole.

Analysis and Interpretation

There are three key findings from the results presented above. First, we find that tree-based models consistently outperform neural networks in forecasting financial returns for datasets with a limited number of datapoints. The bagging tree models used in this analysis generated robust performance while minimizing model overfitting. Specifically, for NN models, the best overall 'version 1' train/test dataset configuration achieved prediction accuracy of 75.0 percent with precision of 100.0 percent, but recall of only 25.0 percent. On the other hand, the best bagging tree classifier achieved overall prediction accuracy of 70.8 percent with precision of 57.1 percent and recall of 50.0 percent. For the purposes of binary classification of future returns, we consider the greater sensitivity of bagging tree classifiers to future negative returns to be a substantial positive. The results were similar in the 'version 2' train/test dataset configuration, as well. The resampling with replacement used by bagging models is particularly beneficial for smaller datasets, such as our train/test time-series data which contained 132 total datapoints.

The second key finding is with respect to the need to carefully calibrate the negative sentiment 'tripwire' used to assign a negative sentiment rating to a given industry and time-step for a 'tipping point.' We believe we've demonstrated quantifiable benefits to using sentiment analysis, aggregated by industry, for time series forecasting, and a key determinant in generating the feature sets used for prediction was the calibration of a negative sentiment prevalence threshold using quartile demarcations. Notably, the higher quartile thresholds – i.e., those which indicate a higher prevalence of negative sentiment entries in a given time-step, as compared to central tendency measures for a given industry – provided higher accuracy, precision and recall across model types and train/test dataset configurations, when viewed as a whole.

The third, and final, key finding is that feature selection consistently improved model performance. We found that using feature subsets, determined either by individual or agglomerative feature ranking methods, consistently improved model performance over models trained on full-feature datasets (See Appendix B For Full Results Tables). We assert that the feature selection techniques applied served to de-noise the feature sets, providing clearer signals upon which the models could train and make predictions. Of the two feature selection methods applied, agglomerative feature sets outperformed individually ranked feature sets in most trials, supporting the findings of Man and Chan in their whitepaper *Cluster-based Feature Selection* (2021).

In summary, our results confirm that under-resourced competitors can achieve meaningful performance enhancements over NN models trained solely on historical financial time-series data by using publicly available earnings transcripts, modest hand-annotated datasets, and fine-tuning pre-trained language models, thereby improving their competitive stature with better-resourced groups.

Conclusions

Our research provides several distinct contributions to the existing literature on sentiment-analysis as applied to financial time-series forecasting: first, our focus on fine-tuning pre-trained transformer-based models on small hand-annotated datasets is a unique and underrepresented sub-focus with important practical implications; second, although NLP sentiment analyses are steadily gaining prominence in time-series forecasting models, our research indicates that segmenting corporate earnings call sentiment scores by industry, and applying feature selection techniques to determine those industries best suited for inclusion as features in time-series forecast models, is hardly mentioned in the literature and may prove

fertile ground for further inquiry and research. We believe our work makes clear that aggregating sentiment analyses by industry imparts meaningful information that may not be properly synthesized by financial market participants, individually or as a whole, providing clear opportunities to improve predictive performance despite the information being publicly available and highly scrutinized at the individual component level.

Directions for Future Work

In this paper we highlighted several novel ideas that further prior work in the area of sentiment analysis for financial time series prediction. With additional time and resources, however, there are four areas of exploration that we believe could further improve predictive performance and merit investigation:

1. Incorporate S&P 500 historical price data with sentiment feature sets for model training, rather than evaluate them independently, as in this paper. A joint analysis could provide additional insight.
2. Develop feature sets at a shorter interval. Due to computational resource constraints, we evaluated sentiment at a monthly interval. The next logical approach would be to scrutinize the sentiment data and make time-series predictions at a weekly cadence.
3. Increase the number of hand-annotated datapoints used to fine-tune the sentiment classification model. Due to limited time and funds, the authors hand-annotated 1,000 text entries over a period of a week. We would want to explore annotating at least 2,000 datapoints, and up to 5,000 datapoints, to improve the sentiment classification accuracy above the roughly 70 percent achieved.
4. Generate a mirrored approach to negative sentiment threshold ‘tripwires’, but with the positive sentiment data. Given the imbalanced nature of the underlying text corpus

dataset, it was important to vary the thresholds for negative sentiment assignment, which significantly impacted model performance across threshold levels. We theorize that doing so for positive sentiment assignments may also improve performance, and would like to explore further this further.

Data and Code Availability

Original data files, annotated data files and the Jupyter Notebook are available on GitHub at the following link:

https://github.com/ruchi-kumar/Natural-Language-Processing-of-Corporate-Earnings-Calls

Acknowledgments

This paper greatly benefited from the peer-review conducted by Pamela Connors and Joshua Fritz. Their comments and suggestions were thoughtful and the authors appreciate the time and effort they put into it.

References

Araci, Dogu. 2019. "FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models." *arXiv.org*. June 25, 2019. https://doi.org/10.48550/arXiv.1908.10063.

Batanovic, Vuc, Milos Cvetanovic and Bosko Nikolic. 2020. "A versatile framework for resource-limited sentiment articulation, annotation, and analysis of short texts." *PLOS ONE.* November 12, 2020. https://doi.org/10.1371/journal.pone.0242050.

bert-based-uncased. (n.d). 2022. *Hugging Face.* Retrieved May 26, 2022 from https://huggingface.co/bert-base-uncased.

Calomiris, Charles and Harry Mamaysky, 2019. "Truth From Lies: Why Natural Language Processing will Revolutionize Central Bank Accountability and Encourage Systematic Monetary Policy." *Shadow Open Market Committee Meeting.* March 29, 2019. https://www.shadowfed.org/wp-content/uploads/2019/03/CalomirisSOMC-March2019.pdf

Compustat Daily Updates - CapitalIQ. 2022. "Transcripts, 2010 - 2021: Data set." *Wharton Research Data Services*. April 4, 2022. https://wrds-www.wharton.upenn.edu/pages/get-data/compustat-capital-iq-standard-poors/capital-iq/transcripts/

Ezen-Can, Aysu. 2020. "A Comparison of LSTM and BERT for Small Corpus." *arXiv.org*. September 14, 2020. https://doi.org/10.48550/arXiv.2009.05451.

Fama, Eugene F. 1970. "Efficient Capital Markets: A Review of Theory and Empirical Work." *The Journal of Finance* 25, no. 2. 383–417. 1970. https://doi.org/10.2307/2325486.

Fisher, Ingrid E., Margaret R. Garnsey and Mark E Hughes. 2016. "Natural Language Processing in Accounting, Auditing and Finance: A Synthesis of the Literature with a Roadmap for Future Research." *Intell. Sys. Acc. Fin. Mgmt. 23, 157–214 (2016).* March 1, 2016. DOI: 10.1002/isaf.1386.

Gu, Yanlei, Takuya Shibukaya, Yohei Kondo, Shintaro Nagao and Shunsuke Kamijo. 2020. "Prediction of Stock Performance Using Deep Neural Networks." *Applied Sciences 2020, 10, 8142.* November 17, 2020. doi:10.3390/app10228142.

Gutierrez-Fandino, Asier, Miguel Noguer i Alonso, Petter Kolm, and Jordi Armengol-Estape. 2021. "FinEAS: Financial Embedding Analysis of Sentiment." *arXiv.org*. Nov 22, 2021. https://arxiv.org/abs/2111.00526.

Jacobs, Gilles and Hoste, Veronique. 2021. "Fine-Grained Implicit Sentiment in Financial News: Uncovering Hidden Bulls and Bears." *Electronics* 2021, 10, 2554. October 19, 2021. https://doi.org/10.3390/electronics10202554.

Kantos, Chris and Joldzic, Dann. 2022. “Comparative Analysis of NLP Approaches for Earnings Calls.” *https://www.alexandriatechnology.com/.* Accessed April 10. 2022.

Kiritchenko, Svetlana and Mohammad, Saif. 2016. “Capturing Reliable Fine-Grained Sentiment Associations by Crowd-Sourcing and Best-Worst Scaling.” *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* June, 2016. https://aclanthology.org/N16-1095/.

Lane, Hobson. 2022. *Natural Language Processing in Action*. S.l.: O'Reilly Media, 2022.

LeCun, Y., Y. Bengio and G. Hinton. 2015. “Deep learning”. *Nature*, 521, 436–444. May 27, 2015. https://doi.org/10.1038/nature14539.

Li, Yawei, Shuqi Lv, Xinghua Liu and Qiuyue Zhang. 2022. “Incorporating Transformers and Attention Networks for Stock Movement Prediction.” *Hindawi.* February 27, 2022. https://doi.org/10.1155/2022/7739087.

Livnat, Joshua and Singh, Jyoti. 2021. “Machine Learning Algorithms to Classify Future Returns Using Structured and Unstructured Data.” *The Journal of Investing.* April 2021. https://joi.pm-research.com/content/30/3.

Man, Xin and Chan, Ernest P. 2021. “The Best Way to Select Features? Comparing MDA, LIME and SHAP.” *The Journal of Financial Data Science, Winter 2021.* July 14, 2021. https://doi.org/10.3905/jfds.2020.1.047.

Maren, Alianna J. 2020. “Natural Language Processing (NLP): Algorithms Overview.” *YouTube*. February 13, 2020. https://www.youtube.com/watch?v=vhV-RpphFHc&t=62s.

Petropoulos, Anastasios and Vasilis Siakoulis. 2021. “Can Central Bank Speeches Predict Financial Market Turbulence? Evidence from an Adaptive NLP Sentiment Index Analysis Using XGBoost Machine Learning Technique.” *Central Bank Review 21 (2021) 141-153.* December 13, 2021. https://doi.org/10.1016/j.cbrev.2021.12.002.

Sezer, Omer Berat, Mehmet Ugur Gudelek and Ahmet Murat Ozbayoglu. 2019. “Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005 - 2019.” *arXiv.org*. November 29, 2019. https://arxiv.org/abs/1911.13288.

Zhang, Tianyi, Felix Wu, Arzoo Katiyar, Kilian Weinberger and Yoav Artzi. 2021. “Revisiting Few-sample BERT Fine-tuning.” *arXiv.org*. March 11, 2021. https://arxiv.org/abs/2006.05987.

Appendix A – Annotation Guidelines

In this document we present the sentiment annotation scheme developed and applied to "Earnings QA Sentiment," a corpus of answers from the Q&A section of earnings calls transcripts. These transcripts are sourced from CapitalIQ, a Standard & Poor's business which delivers comprehensive qualitative and quantitative research and raw data, primarily to investment firms. We were granted access to the data used to compile our corpus through the Wharton Research Data Services (WRDS) database, which consolidates a broad array of data for use by many academic institutions across a broad array of disciplines.

For the transcript data used to produce our corpus, each record is composed of one person's response, typically that of a corporate executive, in a Q&A session following a quarterly earnings release. The responses are located in a data series labeled "componenttext" in the CapitalIQ "Transcripts" database. Each response is logged with a series of unique identifiers specific to the corporate event taking place, the individual respondent and the chronological order of the response with respect to the order in which it occurred within the transcript. Component texts consisting of fewer than 10 tokens - i.e., words and punctuations - were considered too short for proper sentiment determination and were discarded. We also focused our attention on the top 20 industry sectors, by volume of available responses, resulting in a group of 1,146,361 total responses. From that group, 50 responses were randomly selected from each of the 20 industry sectors, for a total of 1,000 sampled data points in the annotation dataset.

Sentiment rankings are generated based on the annotator's understanding of the current and/or future outlook of the company as implied by the content of a given text component response. The developed sentiment articulation and annotation scheme consists of seven sentiment labels, which are heavily influenced by and modeled after the scheme developed by Batanovic, Cvetanovic and Nikolic (Batanovic, 2020). The developed labels are as follows:

- +3 - An unambiguously, or predominately, positive response regarding the immediate or near-future of the organization
- +2 - For texts that articulate an ambiguous sentiment, or a mixture of sentiments, leaning in a positive direction, for a strictly binary classification requirement
- +1 - For texts that are predominately factual statements, devoid of sentiment, but implying a somewhat positive outlook
- 0 - For texts that are purely factual statements and neutral in tone, implying no distinct sentiment; also used to classify questions erroneously labeled as responses by CapitalIQ
- -1 - For texts that are predominately factual statements, devoid of sentiment, but implying a somewhat negative outlook
- -2 - For texts that articulate an ambiguous sentiment, or a mixture of sentiments, leaning in a negative direction, for a strictly binary classification requirement
- -3 - An unambiguously, or predominately, negative response regarding the immediate or near-future of the organization

Starting from an initial set of annotation rules, entries from the 1,000 datapoint annotation corpus were manually annotated, using the sentiment rating system detailed above, by two annotators working separately. Sentiment annotation is performed in three passes through the dataset: 1) the first pass served for the two annotators to familiarize themselves with the data and be sure to work from a common understanding of the guidelines; 2) in the second pass, each annotator works separately through all of the dataset, assigning ratings based on her/his best judgement of the annotation guidelines; 3) using the each of the annotator's ratings from the second pass, escalation procedures are applied (see below), and the third and final annotation pass then consists of the annotators jointly reviewing the entries raised during the escalation

procedures in order to resolve inconsistencies and apply a final rating for those entries. The methodology applied to combine the annotation rating from both annotators is as follows:

- If ratings from both annotators are in agreement, no change is required.
- If the ratings are two or fewer categories apart, the ratings will be averaged and rounded up (down) to the next whole number (+1.5 becomes 2, and -1.5 becomes -2), in effect, pushing the ratings towards the poles of the rating system.
- If the responses differ by more than two categories, the entries enter an escalation procedure whereby the annotators will manually revisit these data points, together, and determine a final value by reviewing and discussing the entry in context of the annotation guidelines.

Upon completion, the ‘final’ annotation ratings will be unified into a single data series for application to NLP sentiment-analysis model training.

Appendix B – Full Model Results Tables

Bagging Tree Models, Version 1 - Models Tested On 2 years of Data (2020 - 2021)	Overall Accuracy	Neg. Pred. Correct	Total Neg. Preds.	Pos. Pred. Correct	Total Pos. Preds.	Total Test Preds.
Baseline Train Set Data	100.00%	N/A	31	N/A	77	108
Baseline Test Set Data	100.00%	N/A	8	N/A	16	24
Models Include Full Feature Sets						
Hard Threshold - 5 Neg. Scores	50.00%	3	10	9	14	24
1st Quartile Threshold	62.50%	0	1	15	23	24
2nd Quartile Threshold	66.67%	2	4	14	20	24
3rd Quartile Threshold	58.33%	0	2	14	22	24
Models Inlcude Individually Ranked Feature Sets						
Hard Threshold - 5 Neg. Scores	54.17%	5	13	8	11	24
1st Quartile Threshold	66.67%	2	4	14	20	24
2nd Quartile Threshold	62.50%	5	11	10	13	24
3rd Quartile Threshold	70.83%	3	5	14	19	24
Models Include Cluster-Based Feature Sets						
Hard Threshold - 5 Neg. Scores	54.17%	4	11	9	13	24
1st Quartile Threshold	62.50%	0	1	15	23	24
2nd Quartile Threshold	70.83%	2	3	15	21	24
3rd Quartile Threshold	70.83%	4	7	13	17	24

Bagging Tree Models, Version 2 - Models Tested On 3 years of Data (2019 - 2021)	Overall Accuracy	Neg. Pred. Correct	Total Neg. Preds.	Pos. Pred. Correct	Total Pos. Preds.	Total Test Preds.
Baseline Train Set Data	100.00%	N/A	28	N/A	68	96
Baseline Test Set Data	100.00%	N/A	11	N/A	25	36
Models Include Full Feature Sets						
Hard Threshold - 5 Neg. Scores	61.11%	3	9	19	27	36
1st Quartile Threshold	52.87%	1	8	18	28	36
2nd Quartile Threshold	52.87%	3	12	16	24	36
3rd Quartile Threshold	52.87%	1	8	18	28	36
Models Inlcude Individually Ranked Feature Sets						
Hard Threshold - 5 Neg. Scores	61.11%	6	15	16	21	36
1st Quartile Threshold	58.33%	5	14	16	22	36
2nd Quartile Threshold	61.11%	6	15	16	21	36
3rd Quartile Threshold	61.11%	4	11	18	25	36
Models Include Cluster-Based Feature Sets						
Hard Threshold - 5 Neg. Scores	55.56%	3	11	17	25	36
1st Quartile Threshold	50.00%	1	9	17	27	36
2nd Quartile Threshold	47.22%	4	16	13	20	36
3rd Quartile Threshold	75.00%	4	6	23	30	36

Neural Net Models, Version 1 - Models Tested On 2 years of Data (2020 - 2021)						
	Overall Accuracy	**Neg. Pred. Correct**	**Total Neg. Preds.**	**Pos. Pred. Correct**	**Total Pos. Preds.**	**Total Test Preds.**
Baseline Train Set Data	100.00%	N/A	31	N/A	77	108
Baseline Test Set Data	100.00%	N/A	8	N/A	16	24
Model Includes SP500 Lagged Returns						
6 Months Lagged Returns	62.50%	1	3	14	21	24
Models Include Full Feature Sets						
Hard Threshold - 5 Neg. Scores	50.00%	0	4	12	20	24
1st Quartile Threshold	62.50%	4	9	11	15	24
2nd Quartile Threshold	62.50%	1	3	14	21	24
3rd Quartile Threshold	66.67%	0	0	16	24	24
Models Inlcude Individually Ranked Feature Sets						
Hard Threshold - 5 Neg. Scores	75.00%	2	2	16	22	24
1st Quartile Threshold	33.33%	8	24	0	0	24
2nd Quartile Threshold	70.83%	1	1	16	23	24
3rd Quartile Threshold	58.33%	0	2	14	22	24
Models Include Cluster-Based Feature Sets						
Hard Threshold - 5 Neg. Scores	66.67%	0	0	16	24	24
1st Quartile Threshold	58.33%	0	2	14	22	24
2nd Quartile Threshold	58.33%	3	8	11	16	24
3rd Quartile Threshold	66.67%	0	0	16	24	24

Neural Net Models, Version 2 - Models Tested On 3 years of Data (2019 - 2021)						
	Overall Accuracy	**Neg. Pred. Correct**	**Total Neg. Preds.**	**Pos. Pred. Correct**	**Total Pos. Preds.**	**Total Test Preds.**
Baseline Train Set Data	100.00%	N/A	28	N/A	68	96
Baseline Test Set Data	100.00%	N/A	11	N/A	25	36
Model Includes SP500 Lagged Returns						
6 Months Lagged Returns	61.11%	1	5	21	31	36
Models Include Full Feature Sets						
Hard Threshold - 5 Neg. Scores	55.56%	4	13	16	23	36
1st Quartile Threshold	52.78%	4	14	15	22	36
2nd Quartile Threshold	47.22%	2	12	15	24	36
3rd Quartile Threshold	63.89%	0	2	23	34	36
Models Inlcude Individually Ranked Feature Sets						
Hard Threshold - 5 Neg. Scores	69.44%	0	0	25	36	36
1st Quartile Threshold	63.89%	0	2	23	34	36
2nd Quartile Threshold	66.67%	2	5	22	31	36
3rd Quartile Threshold	69.44%	0	0	25	36	36
Models Include Cluster-Based Feature Sets						
Hard Threshold - 5 Neg. Scores	55.56%	4	13	16	23	36
1st Quartile Threshold	33.33%	11	35	1	1	36
2nd Quartile Threshold	50.00%	3	13	15	23	36
3rd Quartile Threshold	69.44%	0	0	25	36	36