Towards Earnings Call and Stock Price Movement

Zhiqiang Ma
S&P Global
New York, NY, USA
zhiqiang.ma
@spglobal.com

Grace Bang
S&P Global
New York, NY, USA
grace.bang
@spglobal.com

Chong Wang
S&P Global
New York, NY, USA
chong.wang
@spglobal.com

Xiaomo Liu
S&P Global
New York, NY, USA
xiaomo.liu
@spglobal.com

ABSTRACT

Earnings calls are hosted by management of public companies to discuss the company's financial performance with analysts and investors. Information disclosed during an earnings call is an essential source of data for analysts and investors to make investment decisions. Thus, we leverage earnings call transcripts to predict future stock price dynamics. We propose to model the language in transcripts using a deep learning framework, where an attention mechanism is applied to encode the text data into vectors for the discriminative network classifier to predict stock price movements. Our empirical experiments show that the proposed model is superior to the traditional machine learning baselines and earnings call information can boost the stock price prediction performance.

CCS CONCEPTS

• **Computing methodologies → Supervised learning**; **Natural language processing**;

KEYWORDS

stock price movement prediction; earnings call; deep learning

1 INTRODUCTION

With \$74 trillion in assets under management in the US alone[1], understanding the mechanism of stock market movements is of great interest to financial analysts and researchers. As such, there has been significant research in modeling stock market movements using statistical and, more recently, machine learning models in the past few decades. However, it may not be sensible to directly predict future stock prices given the possibility that they follow a random walk [13]. Thus researchers have proposed to predict the directional movements of stocks and their volatility levels [9, 22, 26]. In this study, we explore company's earnings call transcripts data and investigate using the information embedded in earnings call transcripts to address the task of predicting the movements of stocks by leveraging the recent advancements in natural language processing (NLP).

Stock markets demonstrate notably higher levels of volatility, trading volume, and spreads prior to earnings announcements given the uncertainty in company performance [7]. Such movements can be costly to the investors as they can result in higher trading fees, missed buying opportunities, or overall position losses. Thus, the ability to accurately identify directional movements in stock prices and hold positions accordingly based on earnings releases can be hugely beneficial to investors by potentially minimizing their losses and generating higher returns on invested assets.

Stock market prices are driven by a number of factors including news, market sentiment, and company financial performance. Predicting stock price movements based on market sentiment from the news and social media have been studied previously [6, 9, 26]. However, earnings calls, which occur when companies report on and explain their financial results, have not been extensively studied for predicting stock price movements.

Earnings call are conference calls hosted by the companies and occur between the senior executives of publicly traded companies and call participants such as investors and equity analysts. Generally, the earnings calls are comprised of two components: 1) Presentation of recent financial performance by senior company executives and 2) Question and Answer (Q&A) session between company management and market participants. Earnings calls are comprised of tremendous insights regarding current operations and outlook of companies, which could affect confidence and attitude of investors towards companies and therefore result in stock price movements. The first part of the earnings call – Presentation – is typically scripted and rehearsed, particularly in the face of bad news. However, the question and answer portion of the call incorporates unscripted and dynamic interactions between the market participants and management thus allowing for a more authentic assessment of a company. Thus, we focus on the Answer section in this work and discuss our findings regarding Presentation data in Section 6.

In this paper, we propose a deep learning network to predict the stock price movement, in which sentences from the Answer section of a transcript are represented as vectors by aggregating word embeddings and an attention mechanism is used to capture their contributions to predictions. Discrete industry categories of companies are also considered in the work by encoding them into learnable vector presentations. We compare the proposed method with several classical machine learning algorithms to assess its effectiveness. We review several related work and present our researching and findings in the reset of this paper.

2 RELATED WORK

Stock price movement predictions have traditionally been considered a time series prediction problem [26]. Existing approaches tackle this problem by discovering trading patterns in the historical market data to predict future movements. Statisticians usually use time series analysis techniques like exponential smoothing, autoregressive (AR), and autoregressive integrated moving average (ARIMA) to predict prices or price movements. Computer science researchers have also showed great interest in this topic and have applied machine learning prediction models [17, 19] to solving this task. Recurrent neural networks (RNN) and especially its variants such as LSTM [8], which were developed to process sequential

[1]https://www.bloomberg.com/graphics/2019-asset-management-in-decline

signals, have been widely adopted to model time series stock data [3, 16]. In contrast to these statistical methods, RNN is not subject to the stationarity requirement on the stock time series data and is able to capture the dependency of stock prices at different time instances.

Another important research branch on this topic concentrates on leveraging external information outside of market data, e.g., events, news, macroeconomic environment, business operations, and geopolitical status, as drivers of stock price movements. For example, Equifax's stock price plummeted more than 15 percent immediately following news reports that it had suffered a massive data breach scandal. [5, 6] proposed to extract structured events from news and then use deep neural networks to model the impact of the events on the stock movement. Hu et al. [9] developed a hierarchical attention based neural network – HAN – studying the dependency and influence of the recent online news on stock markets. As social media began reporting breaking news, researchers found social media posts can serve as input along with historical stock data [20, 26]. Financial filings (10-K) summarizing companies' business performance contain sentiment signals from management, which can be used to forecast stock return volatility [24]. Bag-of-words features, TFIDF and LOG1P, were adopted in their work. Researchers name this type of prediction fundamental analysis, and since our work adopted earnings call as the input, it falls in this category as well.

As mentioned in the Section 1, earnings call transcripts have unique properties and provide crucial information of companies. There is tremendous potential for exploration of this dataset as limited previous work has studied earnings call transcripts in stock return volatility prediction to evaluate companies' financial risk [22, 25]. In [22], Theil et al. introduced a neural network PRoFET to predict the stock return volatility, where it considers textual features from earnings call transcripts and financial features including past volatility, market volatility, book-to-market, etc. To create the textual feature, each section (presentation, questions, and answers) is represented as a vector by applying a Bi-LSTM with attention mechanism [1] on the tokens. Financial features pass through a deep feedforward network to calculate the financial vector. The final prediction is returned by summing these two vectors and feeding to another hidden layer.

In NLP tasks, word representation is always a critical component. Pre-trained word vectors and embeddings have been widely adopted in various state-of-the-art NLP architectures and achieved great success. The work like word2vec [14] and GloVe [18] represents words as high dimensional real-valued vectors, and their vector arithmetic operations can reflect the semantic relationship of the words. In this work, we adopted the pre-trained GloVe embeddings to save computing time.

3 PROBLEM STATEMENT

Assuming that there is a set of stocks $\Theta = \{S_1, S_2, \cdots, S_n\}$ of n public companies. For a stock S_c, there exists a series of earnings call transcript $\Gamma_c = \{\boldsymbol{T}_{d_1}, \boldsymbol{T}_{d_2}, \cdots, \boldsymbol{T}_{d_m}\}$, which were held on dates $d_1, d_2, \cdots, d_m$ respectively. The goal is to predict the movement of the stock S_c on date $d_i + 1$ given the earnings call $\boldsymbol{T}_{d_i}$ occurred on date d_i. The movement y is a binary value, 0 (down) or 1 (up). The stock price in the market moves constantly in a trading day. To formally define y, here we adopt the closing price, i.e. $y = \mathbb{1}(p_{d_i+1} > p_{d_i})$, where p_{d_i} and p_{d_i+1} are the closing prices on date d_i and d_i+1.

We aim to learn a prediction function f, which takes features $\boldsymbol{E}$ extracted from an earnings call transcript $\boldsymbol{T}$ and industry categorization $\boldsymbol{I}$ of the company as input, to predict the stock price movement y of the day after the earnings call.

4 MODEL OVERVIEW

To solve the problem, we utilize two features to build the prediction function: 1) Answer section textual feature and 2) company industry type feature. In this section, we firstly propose a deep neural network structure designed to represent the textual feature. Hereafter, we introduce the industry type embedding. The final prediction is generated via a discriminative network by feeding in the combined features.

4.1 Earnings Call Representation

A Q&A section consists of multiple rounds of communications between market participants and company management executives. We only use Answer sections from managements with the assumption that the answers are a more realistic representation of the feedback interested by investors. In the case where a response provided by managements does not answer a specific question, market participants typically follow up with clarifying questions to which they then receive required answers.

Sentence Embedding: Given an earnings call transcript $\boldsymbol{T}$, we extract the answer sequence $\boldsymbol{A} = [l_1, l_2, \cdots, l_N]$ and $\boldsymbol{A} \in \boldsymbol{T}$, l_i denoting a sentence that comes from splitting the Answer section. We treat one sentence as a feature atom, and transform each sentence to a dense vector. To achieve that, we process each token o of a sentence l to a distributed representation vector $\boldsymbol{e}_o$ by leveraging a pre-trained embedding layer. The sentence vector $\boldsymbol{v}_l$ is constructed by concatenating two vectors obtained from average pooling and max pooling the token vectors across all the tokens of the sentence. To reduce computing complexity, we do not allow the word embedding layer to be trainable or fine-tuned. Another popular approach to representing sentences is to employ RNN to encode a whole sentence to a hidden state vector from the last recurrent unit [22]. Sentence encoders [2, 4] may be used here too. We leave them for the future exploration.

Sentence Attention: Undoubtedly, some sentences convey more information while others do not for the task of predicting stock price movements. We leverage the idea of the attention mechanism introduced in the machine translation domain [1] to learn the weights of the sentences, where the weights quantify the contributions of the sentences to the final outcome. Given an answer sequence $\boldsymbol{A}$ consisting of N sentences and sentences transformed to embedding vector $\boldsymbol{v}$s, the attention weights $\boldsymbol{\alpha} \in \mathbb{R}^{1 \times N}$ are defined as normalized scores over all the sentences by a softmax function as shown below,

$$\begin{aligned} \alpha_l &= \text{softmax}(\text{score}(\boldsymbol{v}_l)), \\ \text{score}(\boldsymbol{v}_l) &= \boldsymbol{u}^T \boldsymbol{v}_l + b, \end{aligned} \tag{1}$$

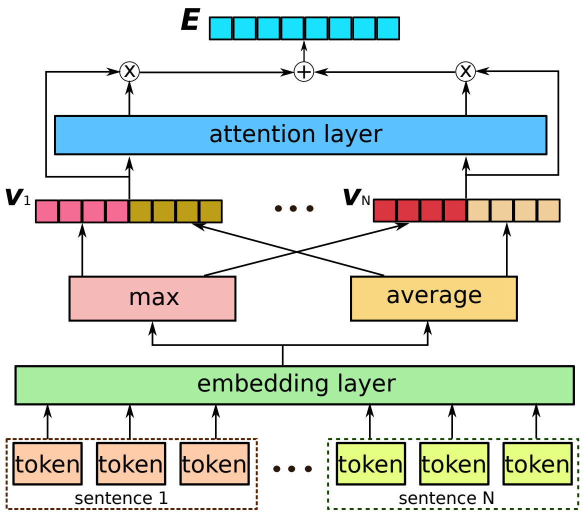

Figure 1: Neural network structure for learning textual feature vectors. The input is tokens from sentences of an Answer section, and the output E is a vector representation of the input.

where $\boldsymbol{u}$ is a learnable vector parameter and b is a learnable bias parameter. The score function may be replaced with others depending on the specific task. Refer to [1, 12, 23] for other score function options. By aggregating the sentence vectors weighted on the attention parameter, the earnings call answer sequence can be transformed to

$$E = \sum_{l}^{N} \alpha_l \boldsymbol{v}_l. \tag{2}$$

Figure 1 demonstrates our network structure introduced above.

4.2 Industry Embedding

Company stock prices usually follow the trend of the industry sector in which it belongs. The sector category and company sector definition vary in terms of standards. We select the Global Industry Classification Standard (GISC) definition in our study. GISC consists of 11 industry sector categories, such as 'energy', 'financials', and 'health care'. The industry sector is a categorical indicator. In machine learning, categorical data are usually transformed by one-hot encoding or ordinal encoding, while we create an embedding layer to transform the categorical values into vector presentation $\boldsymbol{I}$, which is learnable during the network training phase.

4.3 Discriminative Network Structure

With the feature representations $\boldsymbol{E}$ and $\boldsymbol{I}$ built above as input, the final binary classification result is computed by a discriminative network. The feed forward discriminative network consists of multiple hidden layers — batch normalization layer [10], dropout layer [21], ReLU activation layer [15], and linear layer. Figure 2 illustrates the complete neural network structure including the discriminative network.

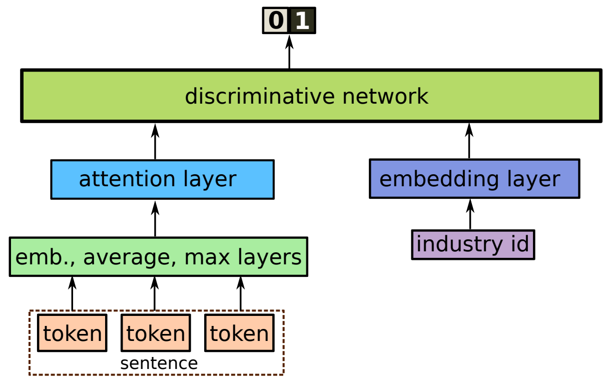

Figure 2: Proposed neural network structure. The input to the discriminative network is a concatenated vector of a textual feature vector and an industry embedding vector.

Answer Section	No. of Sentences	No. of Tokens
Total	3.6 M	21.7 M
Average	2145	1457

Table 1: Statistics of the raw text data (without stop words).

5 EXPERIMENT STUDIES

5.1 Data

We perform experiments by using earnings calls transcripts of S&P 500 companies. We collected 17025 earnings call transcripts over 485 companies[2] from S&P Global Market Intelligence TRANSCRIPTS database. The temporal span of the data is between 2009 and 2019, and the temporal spans for a few companies might be shorter because the companies were added to S&P 500 later than 2009. On average, each company has around 35 transcripts. Every transcript in the TRANSCRIPTS database has been segmented into components in terms of types of the components, such as 'Presentation Operator Message', 'Presentation Section', 'Question', and 'Answer'. We select the 'Answer' components and employed NLTK sentence tokenizer to split sections to sentences. Figure 3 shows the distribution of the number of sentences in 'Answer' components in the dataset. Table 1 shows more statistics of the dataset in terms of sentences and tokens (stop words excluded). As to download the corresponding historical stock data to get the stock price movements $\boldsymbol{y}$, we map company names to their stock tickers and employed Python `pandas_datareader` package with the source set to `yahoo`.

5.2 Experiments

5.2.1 Model Training Settings: For the experiments, we hold out the most recent five earnings call transcripts from each company as the testing dataset (2425 observations in total), and everything else is used as the training and validation dataset. Please note that

[2] S&P 500 index composes 505 stocks from 500 companies. Due to merger and acquisition, ticker changing, shortness of available transcripts, and etc. reasons, 15 companies were not included in the data.

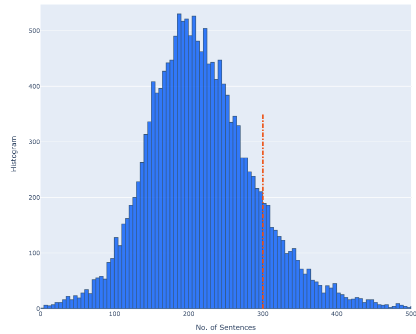

Figure 3: Histogram of the number of sentences in Answer sections. The dashed line in red at 300 at `x − axis` indicates the cut-off on the length of the Answer section.

companies have their earnings call conferences on different dates for every reporting quarter, and it is not reasonable to set a cutoff date to split the dataset.

We split Answer section to sentences and then tokenize sentences to tokens. When transforming tokens to embedding vectors, a vocabulary is constructed, where stop words are ignored and tokens with total frequency less than four times would be disregarded as well. Tokens are transformed to vectors by applying pre-trained GloVe (embedding dimension = 300). As Figure 3 describes with the red cut-off line, when learning the attention scores for the sentences, we set the dimension of $\boldsymbol{E}$ of each transcript to $N = 300$, i.e., transcripts composing more than 300 sentences would be truncated or padded if less than 300 in our implementation. Transcripts with their Answer section lengths shorter than ten sentences are ignored. The model is implemented in Pytorch and experimented on a Nvidia V100 GPU server.

5.2.2 Baselines: In order to assess the performance of our model, we compare its performance with two baseline models below:

- *Mean Reversion (MR)*: MR is a simple trading strategy, which assumes that stock prices would tend to revert toward their moving averages when deviating from them. We calculate 60-day moving average in the experiment.
- *XGBoost*: XGBoost has achieved great success in solving various classification problems in practice. To transform the text data into numeric format, we adopted two feature engineering techniques, `TFIDF` and `LOG1P` defined as below [11]:
  – TFIDF = $\mathrm{TF}(o, \boldsymbol{A}) \times \mathrm{IDF}(o, \boldsymbol{A}) = \mathrm{TC}(o, \boldsymbol{A}) \times \mathrm{IDF}(o, \boldsymbol{A})$
  – LOG1P = $\mathrm{LOG}(1 + \mathrm{TC}(o, \boldsymbol{A}))$
  where $\mathrm{TC}(o, \boldsymbol{A})$) is the count of the token o in earnings call Answer section $\boldsymbol{A}$ and $\mathrm{IDF}(o, \boldsymbol{A}) = \log(|\boldsymbol{\Gamma}|/|\boldsymbol{A} \in \boldsymbol{\Gamma}, o \in \boldsymbol{A}|)$.

5.2.3 Results: In the experiments, we predict the stock price movement of the companies on the day just after their earnings call being released. Table 2 compares the performance of our proposed model and the baseline models. When compare the models, we adopt two evaluation metrics, accuracy and Matthews Correlation Coefficient (MCC), which are also used in the previous work [6, 26]. The definition of MCC is as follows, given true positive (tp), true negative (tn), false positive (fp), and false negative (fn) from the prediction output:

$$\mathrm{MCC} = \frac{\mathrm{tp} \cdot \mathrm{tn} - \mathrm{fp} \cdot \mathrm{fn}}{\sqrt{(\mathrm{tp} + \mathrm{fp})(\mathrm{tp} + \mathrm{fn})(\mathrm{tn} + \mathrm{fp})(\mathrm{tn} + \mathrm{fn})}}.$$

The value of MCC is between −1 and 1, where 1 stands for complete match between predictions and ground truths while −1 means predictions and ground truths are entirely opposite. It can be seen that our model outperforms the baseline models by more than 1% in the prediction accuracy and doubling the MCC measure.

Table 2: Model performance summary.

	Accuracy (%)	MCC
MR	50.80	0.0202
XGBoost (Log1P)	50.89	0.0013
XGBoost (TFIDF)	51.25	0.0154
Our Model	**52.45**	**0.0445**

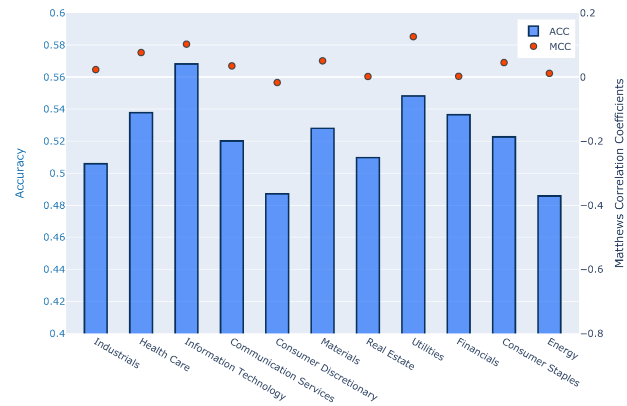

Figure 4: Model performance in terms of industry sectors. The bars represent the accuracy measure and the dots indicate the MCC measure.

Figure 4 is the accuracy and MCC measures in terms of the 11 industry sectors. We can observe that the model performance varies with respect to industries, the highest accuracy (56.8%) occurring in the information technology sector and the lowest accuracy (48.5%) in the energy sector. This result is roughly consistent with our common perception on the stock market – generally stock price movements of high-tech companies are driven by bearish or bullish signals from various sources, while for energy companies their stock performance heavily relies on the crude oil price and macroeconomic factors rather than external news and information.

6 DISCUSSIONS AND CONCLUSIONS

Undoubtedly, stock price movement prediction is a very challenging task. Through our experimentation in this study, we confirm that earnings call transcripts have certain predictive power for future stock price movements. Thus, the inclusion of this dataset in stock price prediction analysis can have predictive impact in the development of such systems for in practice use of stock investment risk analysis. Additionally, we note two other aspects, which are worth more investigation in the future:

- In Section 1, we mention that only Answer sections are included in the model with the management Presentation sections excluded. Our decision on that, besides the heuristics reason mentioned in Section 1, is that the model does not improve by including the Presentation data. The sole consideration of the Presentation text also did not improve the model performance results. Interestingly, this observation is not consistent with the conclusion made by Theil et al. [22], where the Presentation data yields better results in their ablation study for predicting stock volatility, despite the different prediction targets in their work and ours. Future work will be conducted to justify the observation.
- In addition to the fundamental analysis like our work, features originating from technical analysis on the historical stock price data are able to be absorbed into the forecast model. For example, the historical stock time series data can be encoded into another feature vectors by RNN models, which are further used to build global vectors along with the features from the fundamental analysis.

To summarize, we propose leveraging textual information from Answer sections of earnings call transcripts to predict movements of stock prices. To create textual features from transcripts, tokens are transformed into embedding vectors and sentence vectors are built by max pooling and average pooling over the word vectors. An earnings call Answer section is represented as a vector by aggregating its sentence vectors through the attention mechanism. The final prediction is made by a discriminative network which takes the textual feature vectors and learned industry embedding vectors as input. The experiments show that the proposed deep learning model outperforms the classical baseline models, and also prove that the information conveyed in earning calls correlates with stock price movements and therefore can be used in relevant forecasting tasks.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium, 169–174.

[3] L. Cheng, Y. Huang, and M. Wu. 2018. Applied attention-based LSTM neural networks in stock prediction. In *2018 IEEE International Conference on Big Data (Big Data)*. 4716–4718.

[4] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 670–680.

[5] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z. Pan, and Huajun Chen. 2019. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. In *Companion Proceedings of The 2019 World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 678–685.

[6] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep Learning for Event-Driven Stock Prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI '15)*. AAAI Press, 2327–2333.

[7] W.M. Donders, Monique, Roy Kouwenberg, and C. F. Vorst, Ton. 2000. Options and earnings announcements: an empirical study of volatility, trading volume, open interest and liquidity. *European Financial Management* 6, 2 (2000), 149–171.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[9] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to Chaotic Whispers: A Deep Learning Framework for News-Oriented Stock Trend Prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 261–269.

[10] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML '15)*. JMLR.org, 448–456.

[11] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. 2009. Predicting Risk from Financial Reports with Regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Boulder, Colorado, 272–280.

[12] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1412–1421.

[13] Burton Gordon Malkiel. 1999. *A Random Walk down Wall Street: The Time-tested Strategy for Successful Investing*. WW Norton & Company.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[15] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML' 10)*. Madison, WI, USA, 807–814.

[16] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira. 2017. Stock market's price movement prediction with LSTM neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN '17)*. 1419–1426.

[17] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. 2014. Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications* 42, 4 (10 2014), 2162–2172.

[18] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[19] Alaa Sheta, Sara Ahmed, and Hossam Faris. 2015. A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index. *International Journal of Advanced Research in Artificial Intelligence* 4 (07 2015), 55–63.

[20] Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 2013. Exploiting Topic based Twitter Sentiment for Stock Prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, 24–29.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 1929–1958.

[22] Christoph Kilian Theil, Samuel Broscheit, and Heiner Stuckenschmidt. 2019. PRoFET: Predicting the Risk of Firms from Event Transcripts. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI '19)*. International Joint Conferences on Artificial Intelligence Organization, 5211–5217.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS' 17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[24] Chuan-Ju Wang, Ming-Feng Tsai, Tse Liu, and Chin-Ting Chang. 2013. Financial Sentiment Analysis for Risk Prediction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, Nagoya, Japan, 802–808.

[25] William Yang Wang and Zhenhao Hua. 2014. A Semiparametric Gaussian Copula Regression Model for Predicting Financial Risks from Earnings Calls. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland, 1155–1165.

[26] Yumo Xu and Shay B. Cohen. 2018. Stock Movement Prediction from Tweets and Historical Prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1970–1979.