Trade the Event: Corporate Events Detection for News-Based Event-Driven Trading

Zhihan Zhou   **Liqian Ma**   **Han Liu**
Department of Computer Science, Northwestern University
{zhihanzhou, liqianma}@u.northwestern.edu
hanliu@northwestern.edu

Abstract

In this paper, we introduce an event-driven trading strategy that predicts stock movements by detecting corporate events from news articles. Unlike existing models that utilize textual features (e.g., bag-of-words) and sentiments to directly make stock predictions, we consider corporate events as the driving force behind stock movements and aim to profit from the temporary stock mispricing that may occur when corporate events take place. The core of the proposed strategy is a bi-level event detection model. The low-level event detector identifies events' existences from each token, while the high-level event detector incorporates the entire article's representation and the low-level detected results to discover events at the article-level. We also develop an elaborately-annotated dataset ***EDT*** for corporate event detection and news-based stock prediction benchmark. EDT includes 9721 news articles with token-level event labels as well as 303893 news articles with minute-level timestamps and comprehensive stock price labels. Experiments on EDT indicate that the proposed strategy outperforms all the baselines in winning rate, excess returns over the market, and the average return on each transaction. [1]

1 Introduction

By shaping investors' perceptions and assessments of companies, financial news has significant impacts on the stock market (Engle and Ng, 1993; Tetlock, 2007). News-based stock prediction models are thus developed to automatically discover signals of stock market movements from the countless news articles that generated every moment. (Kalyani et al., 2016; Shah et al., 2018; Mohan et al., 2019).

[1]Code and the EDT dataset is available at https://github.com/Zhihan1996/TradeTheEvent

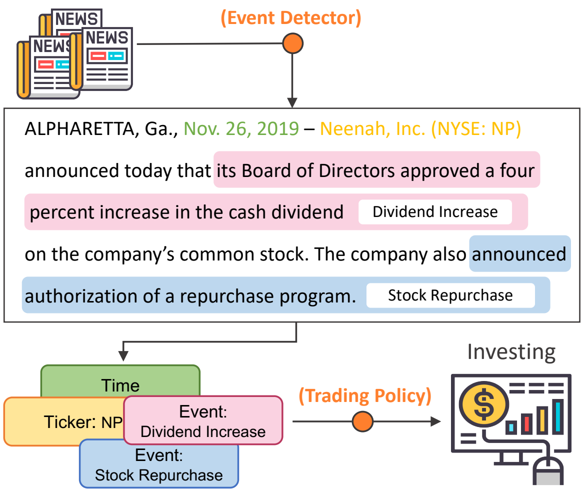
(Event Detector)
NEWS
ALPHARETTA, Ga., Nov. 26, 2019 – Neenah, Inc. (NYSE: NP) announced today that its Board of Directors approved a four percent increase in the cash dividend on the company's common stock. The company also announced authorization of a repurchase program.
Dividend Increase
Stock Repurchase
Time
Ticker: NP
Event: Dividend Increase
Event: Stock Repurchase
(Trading Policy)
Investing

Figure 1: Overview of the event-driven trading strategy

Previous studies mainly rely on textual features and sentiment analysis to forecast the stock movements (Fung et al., 2003; Liu, 2018; Huynh et al., 2017). Both of them, however, often face the problem of poor explainability and low signal-to-noise ratio. Textual feature-based methods often formulate the stock prediction as a text classification problem by directly predicting the rise and fall of stocks based on the extracted features. These models fail to make reasonable trading decisions since they omit the reasons behind stock price changes. Sentiment-based methods avoid this problem by regarding the news articles' sentiments as the indicator of stock movement. However, news sentiment is subjective, which can be greatly affected by the author's standpoint and writing style.

Unlike textual features and sentiments, corporate events are objective facts that impact how investors perceive and assess the related companies. Thus, we resort to corporate events to make more convincing and explainable stock predictions. Jacobs et al. (2018) achieves corporate events detection by splitting a news article into sentences and detecting events on each of them with multi-label sentence classification. This method, however, discards the global contextual information of the entire article and fails to indicate the evidence of events' existences. We believe that detecting events at a smaller granularity (e.g., at the token-level) is beneficial on both model training and application sides. During training, explicitly assigning a label to each token gives the model specific guidelines of what to identify. On the application side, each detected event is supported by one or more subsequences of the original article, allowing users to easily distinguish the predicted results.

However, singly detecting events from the token-level may result in a lack of macro understandings of the entire article. To tackle this, we introduce a bi-level event detection model, in which a low-level detector identifies the subsequences that describe specific events by classifying each token. And a high-level detector takes the predicted results from low-level and integrates them with the input article's global contextual information to predict the probabilities of each event's existence.

Another problem with existing models is that they ignore the timeliness of news articles. Most of them utilize news articles to predict the related securities' rise/fall on the following trading day(s). However, stock prices are very likely to change immediately in response to noteworthy news. Thus, the stock movement in the following trading day(s) may not accurately reflect a news article's influence. To tackle this, we make stock predictions as soon as a news article is published and perform tradings at that moment with the proposed trading policies.

Based on the event detection model and trading policies, we construct an event-driven trading strategy that simultaneously detects corporate events from news articles, indicates the subsequences that describe the detected events, and performs trading on the related stocks. By running the strategies against massive historical data in EDT, we demonstrate the superiority of the proposed strategy in terms of excess returns over the market and the average return on each transaction. The experiment results also reveal the timeliness of news and the effectiveness of corporate events in indicating stock movement.

The main contributions of our paper are as summarized follows: (i) We introduce a novel event-driven trading strategy that detects trading signals from arbitrary unlabeled news articles; (ii) We present ***EDT***, an elaborately-annotated dataset with 300000+ news articles for corporate event detection and news-based trading benchmark. (iii) We propose a bi-level event detection model that integrates macro and fine-grained understandings to effectively identify corporate events;

2 Problem Definition

We aim to construct an event-driven trading strategy that automatically detects corporate events from news articles and performs trading accordingly. The proposed strategy consists of two components: *bi-level event detector* and *trading policy*.

The low-level event detector identifies events from each token. We formulate the low-level event detection as a sequence labeling problem. The label set $\mathcal{L} = \{e_1, e_2, ..., e_k, O\}$ consists of k pre-defined events and a special label O that stands for ***Noevent*** [2]. We define an article $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ as a sequence of tokens and define its label sequence as $\boldsymbol{y} = (y_1, y_2, ..., y_n)$. The same event may be mentioned multiple times in an article, and a single article may contains multiple events. If a subsequence $\boldsymbol{x'} = (x_t, x_{t+1}, ..., x_{t+s})$ of $\boldsymbol{x}$ describes the ***event i***, $\{y_j\}_{j=t}^{t+s}$ are labeled as e_i. All the other words are labeled as O. The low-level event detection is defined as follows: given an article $\boldsymbol{x}^* = (x_1, x_2, ..., x_n)$, find its best label sequence $\boldsymbol{y}^* = (y_1, y_2, ..., y_n)$. We say ***event i*** is detected by the low-level detector if $e_i \in \boldsymbol{y}^*$. Based on the low-level detected results, the high-level event detector calculates the probability of each event's existence. We say ***event i*** is detected by the high-level detector if its existence probability is larger than a given threshold. We combine the predictions on both levels as the final prediction. When events are detected, the *trading policy* decides when to buy and sell the related securities.

3 Strategy

This section discusses the proposed strategy by respectively introducing the event detector and the trading policies.

3.1 Event Detector

Before training the model to detect events, we first equip it with prerequisite knowledge of the financial domain by performing a *domain adaptation*.

[2]events that are not included in $\{e_i\}_{i=1}^{k}$ are also considered as *Noevent*

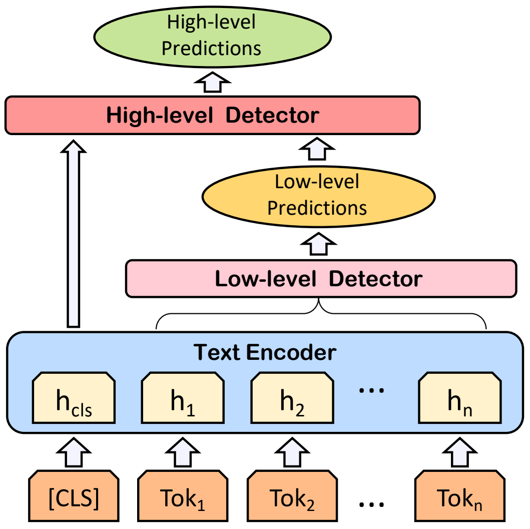

Figure 2: Model overview. The low-level detector identifies corporate events from each token, while the high-level detector summarizes the low-level predictions and the input representation to detect events at the article-level

3.1.1 Domain Adaptation

Since the same event can be expressed with significantly different terms and descriptions, and the same terms can refer to different meanings, understanding the event itself and its related terms is of great importance. We perform domain adaptation by training the model with Masked-Language-Model (MLM) loss on a financial encyclopedia as well as financial news articles. Section 4.3 discusses this corpus in details. During training, 15% tokens of an input sequence are masked, and the model is asked to predict the masked tokens. The prediction is essentially a multi-class classification over the entire vocabulary. We optimize the model with the *Categorical CrossEntropy* loss calculated on all the masked tokens.

3.1.2 Bi-Level Event Detection

As shown in fig. 2, the event detection model takes an article as input and respectively detects events from two levels. Each article is concatenated with a special token [CLS]. The Transformer-based text encoder calculates a series of hidden states for each token. We consider [CLS]'s last hidden state h_{cls} as the representation of the entire article, and h_i as the representations of *token i*.

The low-level detector identifies the subsequences that describe corporate events. For a given label set $\mathcal{L}$ (section 2) with $K+1$ labels (e.g., K pre-defined events and a "*Noevent*"), the detector assigns $K+1$ scores to each token by performing a multi-classes classification based on the token's representation. Each score stands for the confidence of finding a specific event at this position. These scores are concatenated and passed to the high-level detector. During training, we calculate the *Categorical CrossEntropy* loss for each token of an article and average them as the article's low-level loss.

The high-level detector concatenates the low-level prediction as well as the entire article's representation to calculate the probability of each event's existence. We formulate it as a multi-label classification problem. Specifically, we assign K binary labels to each article and represent them with a K-length label vector. For articles without any events, their label vectors are all-zero. If *event i* occurs in an article, the i-th component of its label vector is set to 1. We utilize the *Binary CrossEntropy with Sigmoid* as the loss function. This function considers the K-label classification as K independent binary classification problems. Specifically, it uses the $Sigmoid$ function to map each vector component of the high-level detector's output to $(0, 1)$. We consider the mapped score of each event as its probability of existing in the input article. We then calculate the *Binary CrossEntropy* loss between the mapped score and the binary label of each event. The losses of all the events are summed as the high-level detector's loss. We sum the losses of the low-level and high-level detectors to simultaneously optimize the detectors and the text encoder.

3.1.3 Ticker Recognizer

To make the proposed strategy applicable to arbitrary news articles, besides detecting events, we also recognize the related securities (e.g., company) to trade on.

Each security listed on an exchange has a unique ticker, which is a unique arrangement of characters (e.g., Amazon's ticker at the NASDAQ exchange is AMZN). To recognize the tickers, we download company-ticker pairs (e.g., Amazon v.s. AMZN) for all the securities listed on NYSE and NASDAQ from Yahoo[3]. For a given article, we perform string matching between the article and all the company-ticker pairs. If multiple company-ticker pairs are matched, we choose the one that occurs the most times. Some tricks are employed to improve the accuracy and efficiency. For example, the company-

[3] `finance.yahoo.com`

ticker pairs that match the title's first few words are assigned higher confidence. Although a single article may include multiple securities, we only recognize the one that occurs most to simplify the setting.

3.2 Trading Policy

To minimize other factors' influences, in this paper, we trade only on stocks instead of any derivatives (e.g., options). We relate each detected event to a *long* or *short* trading signal singly based on its event type. Events that may result in a stock price rise are considered as *long* signals, while events that may lead to a fall of stock prices are considered as *short* signals.

We implement two trading policies named **Trade-At-End** and **Trade-At-Best**. Both of them long (e.g., buy) the related stocks for *long* signals and perform short-selling for *short* signals. We define a transaction as a buy(sell) and a sell(buy) of the same stock. The policies always start a transaction (e.g., perform a buy or a short-selling) at the first available time when an event is detected.

Trade-At-End (k): This policy holds a started transaction for k trading days and closes the transaction on the last day when the market closes.

Trade-At-Best (k): This policy closes a transaction at the best price (e.g., highest for sell and lowest for buy) within k trading days from the start date. It estimates the profit that a trader can gain at most within k trading days with a detected event.

4 Data

In this section, we discuss the *EDT* dataset. EDT contains data for three purposes: 1. corporate event detection (section 4.1); 2. news-based trading strategy benchmark (section 4.2); 3. financial domain adaptation (section 4.3).

4.1 Data for Event Detection

We choose 11 types of corporate events that have relatively predictable and straightforward impacts on the stock price based on financial knowledge.

4.1.1 Event Type

In this work, we only focus on non-periodic corporate events. Trading on periodic corporate events such as *Earning Call* is much trickier since investors can access their information from multiple sources in advance. We leave them for future works.

Guidance Increase (GI) Guidance is a company's public estimates of its upcoming-quarter/fiscal year earnings. This event includes the announcements of guidance increase or upgrade.

Acquisition (A) An acquisition event happens when a company announces to purchase all or a portion of another company's shares or assets.

New Contract (NC) The new contract event refers to a company announcement of being awarded a new contract.

Stock Split (SS) A stock split event refers to a company that divides the existing shares of its stock into multiple new shares.

Reverse Stock Split (RSS) This is the reverse process of the stock split, which consolidates the number of existing stock shares into fewer shares.

Positive Clinical Trial & FDA Approval (CT) This event includes (i) positive trial results from clinical studies; (ii) receiving FDA approval, clearance, or being granted by FDA to market legally in the United States.

Stock Repurchase (SR) A company's stock repurchase events include declaring, reinstatement, or increasing a stock buyback plan.

Dividend (RD) The dividend is a distribution of some of a company's profits paid to its shareholders.

Dividend Cut (DC) A dividend cut means to reduce, stop, or suspend a pre-announced dividend.

Dividend Increase (DI) A dividend increase refers to an increase in the regular dividend.

Special Dividend (SD) A special dividend is an event that a company declares a non-recurring dividend paid to its shareholders.

4.1.2 Detailed information

Event	Num.	Event	Num.
A	229	CT	314
RD	290	DC	109
DI	225	GI	99
NC	518	RSS	51
SD	80	SR	385
SS	76		

Table 1: Number of articles in *EDT* for each event.

We collect 9721 English news articles, of which 2266 articles contain at least one of the above events. Table 1 shows the number of articles that corresponds to each event. The rest 7455 articles are news articles that do not contain any of the above events. We expect them to help the model better distinguish the event-related articles from the non-event ones. Among them, we deliberately include hundreds of non-event articles that are highly similar to event-related ones. An example here is "Apple announces a stock repurchase program" v.s. "Apple announces the completion of the recently announced stock repurchase program", in which we do not expect the latter one to have a significant influence on the stock price.

These news articles are downloaded from PRNewswire[4], Businesswire[5] and GlobeNewswire[6] using keywords-search. The keywords for each event are manually determined based on samples of that event. Each article's title, subtitle, and main text are concatenated after data cleaning (e.g., remove special symbols). We annotate each article with token-level labels. Two human annotators are asked to independently mark the subsequences that best describe the pre-defined corporate events. The annotations of an article are produced if the annotators give the same result. Otherwise, they discuss the best annotations.

We randomly sample 80% articles of each event and combine them with 80% of non-event articles to form the training data. The rest are considered as the validation data.

4.2 Data for Strategy Evaluation

We develop this data to benchmark news-based stock prediction models and trading strategies. To accurately account for the stock movement, the news articles should be original-sourced. To mimic the real-world situation, the news articles should be diverse enough (e.g., in different categories). Thus, we choose PRNewswire and Businesswire as the article collection sources and download all the English news from PRNewswire (Mar 1st, 2020 - Apr 30th, 2021) and Businesswire (Aug 16th, 2020 - May 6th, 2021). We remove the articles that exist in the training data (section 4.1.2). Since some pre-defined events are infrequent (e.g., stock split), to ensure that there are at least a few samples of every event, we add all the articles of the validation data (section 4.1.2) to this data. After data cleaning, there are 303893 news articles.

Each article comes with a minute-level timestamp, which allows researchers to locate the exact event happening time. Generally, news-based trading strategy evaluation involves four steps. (i) Identify trading signals (e.g., corporate events or sentiments) from news articles; (ii) For each article where trading signals are detected, recognize the related company(ticker); (iii) Get the recognized company's stock price data around the publish time of the news; (iv) Perform transactions based on trading policies.

To enable researchers without ticker recognizers and historical stock price data to benchmark their models/strategies, we assign each article with an automatically recognized ticker as well as detailed price labels of that ticker. With the detailed price labels of each news article, strategy evaluation can be as easy as "counting the price changes on the articles that are recognized as trading signals".

Specifically, an article's price labels includes: *open* / *close* prices at the first minute we can trade on, *highest* / *lowest* prices in the following 1/2/3 trading days, *close* prices in the following 1/2/3 trading days, and the minute-level timestamp corresponding to each price. If available, we take the prices in the pre-market and after-hours into consideration since many corporate events are announced in these periods, and stock prices may change greatly during these times. The price labels are empty for articles where no ticker is recognized or the historical price data is unavailable. Among all the evaluation data, 106619 articles come with non-empty price labels.[7]

4.3 Data for Domain Adaptation

The corpus for domain adaptation contains financial news articles and a financial terms encyclopedia, which is considered as unstructured domain knowledge. For encyclopedia, we download 6260 explanatory documents from Investopedia[8]. Each document explains a specific financial term and describes the role it may play in the financial market. For news articles, we directly utilize all the news articles of the training data (section 4.1.2).

5 Experiment

In this section, we first exhaustively compare Bi-level Detection with baselines under different settings. Then, we discuss how each event contributes to the overall trading results. Lastly, we analyze the profitability and practicality of the proposed strategy in real-world stock trading.

[4]https://www.prnewswire.com/
[5]https://www.businesswire.com/
[6]https://www.globenewswire.com/

[7]We acquire historical stock price data from https://polygon.io/
[8]https://www.investopedia.com/

Among the 11 corporate events in the EDT dataset, we do not trade on the **Dividend** since we do not consider it to have a significant influence on stock prices. Among the rest, we consider **Reverse Stock Split** and **Dividend Cut** to have negative influences on the stock price, while the others to have positive influences. We evaluate the performance of the proposed strategy with backtesting. Backtesting is widely used to evaluate a trading strategy's effectiveness by running the strategy against historical data. We perform trade on all the detected trading signals for each model.

Metrics For a "buy" transaction, we define its return as $\frac{P_{sell}-P_{buy}}{P_{buy}}\%$, while for a "short-selling", we define its return as $\frac{P_{sell}-P_{buy}}{P_{sell}}\%$. Here, P stands for the price. If a transaction's return is greater than or equal to 0, we call it a "win". If a transaction's return is greater than or equal to 1%, we call it a "big win". For each model, we calculate its **winning rate**, **big win rate** (rate of big wins among all the transactions) and **average return on each transaction**. We also evaluate the models' excess returns over the market, where we consider the *S&P 500* index as the benchmark of the market performance. The **market return** is estimated as the return of buying *S&P 500 index ETF* for \$10000 on Mar 1st, 2020 and sell all of them on May 6th, 2021[9]. For each model, we start with \$10000 cash and invest \$2000 to each trading signal. When available cash is less than \$2000, we invest 20% of available cash to the detected signal. We report the **excess returns** of each model, which equals to a model's **total returns** minus the **market return**. [10] We assume there is a 0.3% commission fee on each transaction.

Model Hyperparameters We employ the pre-trained BERT (Devlin et al., 2018) model as the text encoder. Specifically, we use the *bert-base-cased* checkpoint. Both the low-level and high-level detectors consist of a hidden layer and an output layer. There are 2048 hidden units in the hidden layer. We utilize AdamW optimizer with batch size 32 and learning rate 5e-5 to train the encoder and detectors together for 5 epochs. We set the maximum input length as 256 since we find almost all the events mentioned in a news article exist in its first 256 tokens. Training of the model costs 15 minutes on 4 Nvidia RTX 2080Ti GPUs. We conduct each experiment with 3 different random seeds and report the average results.

Trading Details For Trade-At-End, we execute a stop loss of 20% (e.g., sell a stock immediately when it falls 20%). In all the experiments, we only trade on the news articles where the historical price data of the detected ticker is available at the minute when the article is published. In other words, we ignore all the news articles that are not published during the market hours and articles where the historical price data is incomplete.

5.1 Baseline

Vader (Gilbert, 2014) is a rule-based sentiment analysis model that assigns *positive*, *negative*, and *neutral* scores to an article. We consider news articles with a positive score greater than 0.2 as *long* trading signals.

BERT-SST is a BERT-based (Devlin et al., 2018) sentiment analysis model trained on the Stanford sentiment treebank (SST) dataset. We respectively consider news articles with a positive score greater than 0.995 and 0.9 as *long* trading signals to reduce the threshold's influences on the final results.

Sentence (Jacobs et al., 2018) splits an article into sentences and performs sentence-level event detection based on multi-label text classification. It was original implemented with SVM and LSTM. We re-implement it with BERT to compare it fairly with our models. We split each article into sentences with the NLTK toolkit (Loper and Bird, 2002) to train and evaluate the model.

BERT-CRF (Lafferty et al., 2001) was originally proposed as a Conditional Random Fields-based sequence labeling model, which combines emission scores given by BERT and learned transition scores to find the global optimal label sequence for each input. We re-implement it to perform event detection singly on the token-level. Following the literature, we use different learning rates for the CRF(1e-3) and the BERT(3e-5) components.

5.2 Main Results

Table 2 and 3 respectively shows the models' 1-day and 2-day trading results. The result of 3-day trading is consistent. Due to space limitations, we present it in appendix A. As shown in the tables,

[9] In accordance to the time span of data for evaluation

[10] Since Trade-At-Best always finishes the transaction at the best price, its winning rate is always 100% and its total returns is almost linearly related to the number of transactions. Thus, we only report the average return of this policy.

Model	TAE (1)			TAB (1)	
	Win Rate	Ave. Return	Exc. Returns	Ave. Return	Num. Trans.
Vader (Gilbert, 2014)	52.8||24.3%	0.06%	-$8116	1.72%	4327
BERT-SST (0.995)	54.3||26.9%	0.45%	$3743	2.94%	2378
BERT-SST (0.9)	52.9||26.1%	0.41%	$44049	3.31%	15663
Sentence (Jacobs et al., 2018)	**55.5**||30.9%	1.37%	$54064	7.21%	2881
BERT-CRF	53.7||33.8%	1.60%	$83120	8.87%	3533
Bi-level Detection (ours)	54.5||**34.2%**	**1.74%**	$**84443**	**9.11%**	3118

Table 2: This table shows the 1-day trading result, in which we start each transaction at the news article's publish time and end the transaction after 1 trading day. **Win Rate** stands for the overall winning rate (rate of transactions that have a return over 0) || big win rate (rate of transactions that have a return over 1%). **Ave. Return** stands for the average return on each transaction. **Exc. Return** stands for the total excess returns over the market when starting with $10000 and invest $2000 to each detected trading signal. **Num. Trans.** stands for the number of transactions (valid trading signals) of each model.

Model	TAE (2)			TAB (2)	
	Win Rate	Ave. Return	Exc. Returns	Ave. Return	Num. Trans.
Vader (Gilbert, 2014)	53.7||36.9%	0.38%	-$2551	3.11%	4327
BERT-SST (0.995)	**53.8**||38.4%	0.62%	$8479	4.72%	2378
BERT-SST (0.9)	52.3||37.2%	0.46%	$12802	3.93%	15663
Sentence (Jacobs et al., 2018)	52.7||39.9%	1.24%	$24673	9.49%	2881
BERT-CRF	51.3||39.9%	1.39%	$52891	11.27%	3533
Bi-level Detection (ours)	52.3||**40.8%**	**1.56%**	$**59375**	**11.53%**	3118

Table 3: This table shows the 2-day trading result, in which we start each transaction at the news article's publish time and end the transaction after 2 trading day.

our model outperforms all the baselines on average return and exceed return under all the settings.

Results of Ticker Recognizer To evaluate the performance of ticker recognizer, we manually label tickers for 1674 news articles. The proposed ticker recognizer succeeds in 1643 of them (accuracy: 98.15%). Although it obtains a satisfactory performance, its imperfect recognition may slightly impair the evaluation of trading strategies, since it may point out the incorrect securities for the strategies to trade on.

Results of Trade-At-End Even with the simple trading policy, our model achieves an average return of 1.74% and an exceed return of $84443 (844%) in 1-day trading.

Experiments on Vader and BERT-SST show that the sentiment of a news article can indicate the stock movement to some extend. For example, BERT-SST achieves great winning rates, and it successfully outperforms the market index by a considerable margin. However, these signals tend to results in small stock movements. Thus, sentiment-based models achieve poor average returns. On the other hand, event-based models obtain much higher average returns, demonstrating the superiority of corporate events over news sentiments in indicating stock movements.

Although the Sentence model highly outperforms the sentiment-based methods, compared to our models, it is less robust to ambiguous articles, and it is more likely to miss the events that are described in several continuous sentences. By utilizing global context information and detecting events from the token level, our model identifies more trading signals and avoids more potential traps.

Bi-level Detection also achieves better performance than BERT-CRF under all the settings. The improvements mainly come from the high-level detector. By combining the global contextual information with token-level detected results, Bi-level Detection is more robust and more effective. When the low-level detector generates false alarms on some ambiguous tokens or fails to detect events that are not explicitly described, the high-level detector may point it out after analyzing the meaning of the entire article.

Results of Trade-At-Best This trading policy is designed to measure the ceiling of trading signals given by each model. Under this setting, Bi-level Detection obtains a 9.11% average return that dramatically exceeds all the sentiment-based models, indicating how significant the stock price changes when corporate events take place. Com-

Metric	Policy	Event Type									
		A	CT	DC	DI	GI	NC	RSS	SD	SR	SS
WR. %	TAE (1)	52.8	51.7	60.0	66.9	55.9	53.3	57.8	84.0	60.2	43.3
	TAE (3)	53.5	48.0	49.4	60.2	52.8	50.5	55.6	64.0	55.6	81.1
AR. %	TAE (1)	2.15	1.89	0.75	0.33	0.88	1.72	3.26	4.76	1.27	-0.04
	TAE (3)	1.93	2.05	0.02	1.02	1.19	1.32	4.25	5.49	1.51	4.99
	TAB (1)	9.98	12.32	5.71	1.74	4.63	9.40	8.19	8.02	5.26	4.27
	TAB (3)	13.46	17.51	10.95	4.21	7.73	13.24	19.68	13.13	8.79	17.91
Num.		960	721	39	135	282	697	26	29	225	5

Table 4: The table shows **Bi-level Detection**'s 1-day and 3-day trading results on each event, where **WR.** stands for the winning rate and **AR.** stands for the average return on each transaction. Each column respectively shows the results of Acquisition(A), Clinical Trial(CT), Dividend Cut(DC), Dividend Increase(DI), Guidance Increase(GI), New Contract(NC), Reverse Stock Split(RSS), Special Dividend(SD), Stock Repurchase(SR) and Stock Split(SS).

Model	TAE (1)			TAB (1)
	Win Rate	Ave. Return	Exc. Returns	Ave. Return
Bi-level Detection - Open	**54.5**||**34.2**%	**1.74**%	$**84443**	**9.11**%
Bi-level Detection - Close	51.4||29.5%	-0.07%	-$12026	4.50%

Table 5: This table shows **Bi-level Detection**'s 1-day trading results when start transactions respectively with the ***Open*** price and ***Close*** price at the minute that the event is detected.

pared to other event-based models, Bi-level Detection achieves much better performance in identifying corporate events.

5.3 Event Analysis

Table 4 shows Bi-level Detection's 1-day and 3-day trading results on each event. As shown in the table, the Dividend Increase has the smallest influence on the stock price, while the Reverse Stock Split and Clinical Trial significantly impact the stock prices.

Reverse Stock Split and Special Dividend are the most profitable corporate event, yet they are relatively rare. Acquisition, Clinical Trial and New Contract are ubiquitous, and they also lead to significant stock movements. However, comparisons between TAE and TAB's profits indicate that these events are relatively trickier to trade. Although the ideal policy TAB achieves dramatic profits, TAE makes much fewer profits, indicating that the stocks may oscillate significantly after these events. On the other hand, Special Dividend and Reverse Stock Split are comparatively easier to trade on.

Different events also affect the stock prices in different periods. We measure the influence by comparing TAB (1) and TAB (3)'s average return on the same event. They achieve close average returns on Acquisition, Clinical Trial, and Stock Repurchase, indicating that the stock prices do not continue to change sharply after the first day. In contrast, Stock Split impacts the stock price for a more extended period.

Thus, an ideal trading strategy should take the above factors into account. For example, it may assign different weights to each event based on their profitability and use a different policy to trade each event based on their potential price change patterns.

5.4 Profitability in Real-world

In this section, we discuss the possible profitability of the proposed strategy in real-world trading. Backtesting against historical data shows that the proposed strategy dramatically outperforms the market index. However, this result is based on two main assumptions.

First, we assume the cost of time in acquiring the news articles and making trading decisions is almost 0. Table 5 indicates the significance of timeliness, in which **Bi-level Detection - Open** starts transactions with the ***Open*** price at the minute that the news article is published (e.g., price at 11:23:00), while **Bi-level Detection - Close** starts with the ***Close*** price at that minute (e.g., price at 11:23:59). As shown in the table, when the model trades tens of seconds after the publish time of the news article, it greatly underperforms the market index and achieves a negative average return on each transaction. These results demonstrate that the profitability of an event-based model highly depends on how "quick" one can perform tradings after a piece of news is published.

Second, we assume we can always buy/sell the desired amount of stock shares and ignore the liquidity of the stocks. When the investment scale is relatively small, this assumption doesn't have a big impact. However, as the investment scales up, the liquidity may greatly constrain the model's profitability.

6 Related Works

Text-base Stock Prediction Existing methods usually count on textual features and sentiment analysis to forecast the stock movements (Hagenau et al., 2013; Mohan et al., 2019; Xie and Jiang, 2019; Huynh et al., 2017; Liu, 2018; Mittal and Goel). Hagenau et al. (2013) utilizes N-Gram, Noun-phrases, and 2-words combinations of corporate announcements to predict the stock movement. The influence of financial news on the stock market is also widely explored (Engle and Ng, 1993; Tetlock, 2007). In recent years, researchers resort to support vector machine and deep neural network to analyze financial news articles (Liu, 2018; Xie and Jiang, 2019; Ding et al., 2015; Huynh et al., 2017). Mohan et al. (2019) combines news text (e.g., sentiment, tf-idf, and word2vec representation) with historical stock data to predict future stock prices. Event-based stock predictions are also introduced. Ding et al. (2015) extract events from news articles, calculate event embedding, and use it to predict direction of stock moves. Ben Ami and Feldman (2017) build sentiment-type trading signals with word polarities and event-type trading signals with existing information extraction platform and demonstrates the superiority of event over sentiment in making trading decisions.

Event Detection General domain event detection that aims to recognize structured schemata/frames from the text has been widely explored by data-driven supervised learning methods (Ahn, 2006; Mitamura et al.). In the economic domain, however, existing approaches (Arendarenko and Kakkonen, 2012; Hogenboom et al., 2013; Xie et al., 2013) usually exploit knowledge-based and rule-based methods, which require extensive hand-designed rules and ontology. Recent works conceptualize corporate events as sequences of text that reported company-related occurrences and introduce data-driven methods to solve financial event detection with text classification (Ein-Dor et al., 2019; Jacobs et al., 2018). Ein-Dor et al. (2019) explored a Wikipedia-based supervised method to detect the sentences that may include corporate events. Jacobs et al. (2018) propose a multi-label sequence classification model to detect specific corporate events from news articles.

7 Conclusion

This paper introduces an event-driven trading strategy based on corporate event detection from news articles. We introduce a bi-level event detection model that utilizes global and local information to identifies corporate events. Experiments on the presented dataset EDT demonstrate the proposed model's superiority over all the baselines. The results also signify the corporate event's timeliness and effectiveness in indicating stock movement.

In future work, we plan to explore more on both the event detection model and trading policy. We expect to involve external knowledge and few-shot learning methods to relieve the event detection model from the data imbalance and data-scarce scenarios. On the trading policy side, we aim to explore more types of events and customize different policies for each event based on the potential price change patterns it may lead to.

Acknowledgments

Han Liu is grateful for the support of National Science Foundation (BIGDATA 1840866, RI 1408910, CAREER 1841569, TRIPODS 1740735), Alfred P Sloan Fellowship, and DARPA QEDRML. We sincerely thank Polygon.io for their high-quality historical stock price data.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Ernest Arendarenko and Tuomo Kakkonen. 2012. Ontology-based information and event extraction for business intelligence. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 89–102. Springer.

Zvi Ben Ami and Ronen Feldman. 2017. Event-based trading: Building superior trading strategies with state-of-the-art information extraction tools. *Available at SSRN 2907600*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

Liat Ein-Dor, Ariel Gera, Orith Toledo-Ronen, Alon Halfon, Benjamin Sznajder, Lena Dankin, Yonatan Bilu, Yoav Katz, and Noam Slonim. 2019. Financial event extraction using Wikipedia-based weak supervision. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*, pages 10–15, Hong Kong. Association for Computational Linguistics.

Robert F Engle and Victor K Ng. 1993. Measuring and testing the impact of news on volatility. *The journal of finance*, 48(5):1749–1778.

G Pui Cheong Fung, J Xu Yu, and Wai Lam. 2003. Stock prediction: Integrating text mining approach using real-time news. In *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.*, pages 395–402. IEEE.

CHE Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text.

Michael Hagenau, Michael Liebmann, and Dirk Neumann. 2013. Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, 55(3):685 – 697.

Alexander Hogenboom, Frederik Hogenboom, Flavius Frasincar, Kim Schouten, and Otto Van Der Meer. 2013. Semantics-based information extraction for detecting economic events. *Multimedia Tools and Applications*, 64(1):27–52.

Huy D. Huynh, L. Minh Dang, and Duc Duong. 2017. A new model for stock price movements prediction using deep neural network. In *Proceedings of the Eighth International Symposium on Information and Communication Technology*, SoICT 2017, page 57–62, New York, NY, USA. Association for Computing Machinery.

Gilles Jacobs, Els Lefever, and Véronique Hoste. 2018. Economic event detection in company-specific news text. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Joshi Kalyani, Prof Bharathi, Prof Jyothi, et al. 2016. Stock trend prediction using news sentiment analysis. *arXiv preprint arXiv:1607.01958*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Huicheng Liu. 2018. Leveraging financial news for stock trend prediction with attention-based recurrent neural network. *arXiv preprint arXiv:1811.06173*.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. Overview of tac kbp 2015 event nugget track.

Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis.

S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu. 2019. Stock price prediction using news sentiment analysis. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 205–208.

Dev Shah, Haruna Isah, and Farhana Zulkernine. 2018. Predicting the effects of news sentiments on the stock market. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4705–4708. IEEE.

Paul C Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168.

Boyi Xie, Rebecca J. Passonneau, Leon Wu, and Germán G. Creamer. 2013. Semantic frames to predict stock price movement. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–883, Sofia, Bulgaria. Association for Computational Linguistics.

Yancong Xie and Hongxun Jiang. 2019. Stock market forecasting based on text mining technology: A support vector machine method. *arXiv preprint arXiv:1909.12789*.

A Results for 3 days trading

As shown in table 6, the trading results on 3-day trading is consistent with the results of 1-day and 2-day tradings.

Model	**TAE (3)**			**TAB (3)**	
	Win Rate	**Ave. Return**	**Exc. Returns**	**Ave. Return**	**Num. Trans.**
Vader (Gilbert, 2014)	55.6||41.9%	0.69%	-$152	4.11%	4327
BERT-SST (0.995)	**56.1**||**42.8**%	0.98%	$1643	8.07%	2378
BERT-SST (0.9)	54.1||41.1%	0.73%	$3490	6.23%	15663
Sentence (Jacobs et al., 2018)	54.3||42.6%	1.59%	$40066	11.11%	2881
BERT-CRF	51.5||41.2%	1.57%	$53152	12.94%	3533
Bi-level Detection (ours)	52.0||42.0%	**1.71**%	$**55339**	**13.11**%	3118

Table 6: This table shows the 3-day trading result, in which we start each transaction at the news article's publish time and end the transaction after 3 trading day. **Win Rate** stands for the overall winning rate (rate of transactions that have a return over 0) || big win rate (rate of transactions that have a return over 1%). **Ave. Return** stands for the average return on each transaction. **Exc. Return** stands for the total excess returns over the market when starting with $10000 and invest $2000 to each detected trading signal. **Num. Trans.** stands for the number of transactions (valid trading signals) of each model.