FAST: Financial News and Tweet Based Time Aware Network for Stock Trading

Ramit Sawhney*
Netaji Subhas Institute of Technology
ramits.co@nsit.net.in

Arnav Wadhwa*
MIDAS, IIIT Delhi
arnavw96@gmail.com

Shivam Agarwal
Manipal Institute of Technology
shivamag99@gmail.com

Rajiv Ratn Shah
IIIT Delhi
rajivratn@iiitd.ac.in

Abstract

Designing profitable trading strategies is complex as stock movements are highly stochastic; the market is influenced by large volumes of noisy data across diverse information sources like news and social media. Prior work mostly treats stock movement prediction as a regression or classification task and is not directly optimized towards profit-making. Further, they do not model the fine-grain temporal irregularities in the release of vast volumes of text that the market responds to quickly. Building on these limitations, we propose a novel hierarchical, learning to rank approach that uses textual data to make time-aware predictions for ranking stocks based on expected profit. Our approach outperforms state-of-the-art methods by over 8% in terms of cumulative profit and risk-adjusted returns in trading simulations on two benchmarks: English tweets and Chinese financial news spanning two major stock indexes and four global markets. Through ablative and qualitative analyses, we build the case for our method as a tool for daily stock trading.

1 Introduction

The stock market, a financial ecosystem involving quantitative trading and investing, observed a market capitalization exceeding \$US 60 trillion as of 2019.[1] Stock trading presents lucrative opportunities for investors to utilize the market as a platform for investing funds and maximizing profits. However, making profitable investment decisions is challenging due to the market's volatile and rapid-changing nature (Adam et al., 2016; Foucault et al., 2016). Research at the intersection of natural language processing (NLP) and finance presents encouraging prospects in stock prediction (Jiang,
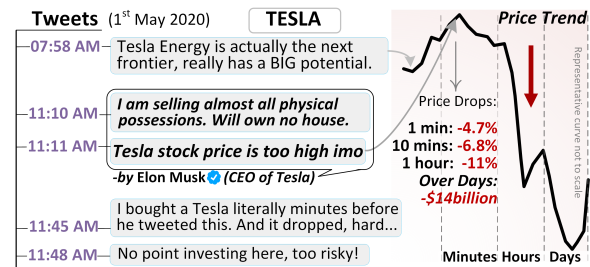

Figure 1: Here, we study how Tesla's tweets influence investors' opinions about the company and impact its stock price trend. The first tweet shows positive opinions, and we observe a rise in prices. Later, the tweets made by the CEO *rapidly* lead to drastic price drops within minutes. Further, without a sequential context, it gets challenging to understand the tweets that follow.

2020). Conventional work forecasts future trends by modeling numerical historic stock data (Lu et al., 2009; Bao et al., 2017). However, price signals alone can not capture market surprises, news, and company announcements. Such events, often reported across financial news and social media, have shown to influence market dynamics (Laakkonen, 2004). As shown in Figure 1, prices immediately react to breaking news about the related company. Such reactions conform to the Efficient Market Hypothesis (EMH), which states that financial markets are *informationally efficient* and prices reflect all available information (Malkiel, 1989).

The abundance of stock affecting information across news and Twitter helps investors analyze market trends and inspires the adoption of NLP to study the interplay between textual data and stock prices (Xu and Cohen, 2018; Oliveira et al., 2017). However, unlike structured numerical data, analyzing natural language poses various challenges. First, analyzing individual tweets or news headlines may not be informative enough. They often exhibit a sequential context-dependency, where analyzing them together can provide a greater unified context,
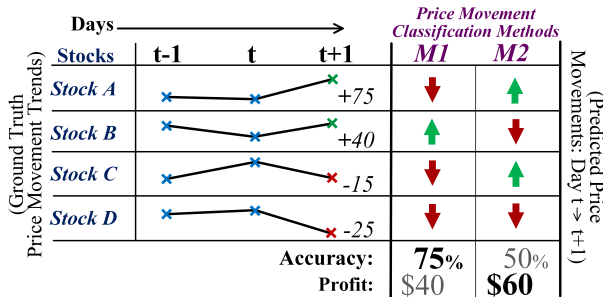
Figure 2: More accurate methods M_1 (higher accuracy) may not always be more profitable than less accurate methods M_2 (lower accuracy). Profit is gained by selling stocks having ⬆ prediction for price movement from trading day t to t+1.

as shown in Figure 1. Despite the success of recurrent neural networks (RNNs) in modeling such a sequential context (§2), a critical drawback is that they assume all text to be equally spaced in time, ignoring the inherent dynamic timing irregularities of social media and news. Timing plays a critical role as stock markets rapidly react to new information (Foucault et al., 2016), leading to significant price changes within minutes, as shown in Figure 1. Scholtus et al. (2014) show that reacting one second slower than other market participants can lead to a loss in thousands of dollars. Further, not each text holds the potential to influence stock prices, texts have a *diverse influence* on stock prices based on their content, such as breaking news or tweets from a reliable source, as opposed to noise like vague comments as shown in Figure 1. These observations mandate the need to factor in the *time-aware dependencies* and *diverse influence* in analyzing online natural language data for stock trading.

Despite profitability being the prime objective of trading, NLP methods for stock prediction (Xu and Cohen, 2018; Hu et al., 2017) are commonly framed as classification or regression tasks, and are not directly optimized towards profitable stock selection. Consider the toy example in Figure 2 that shows how methods having a higher classification accuracy may not always lead to higher overall profits. This research gap in NLP methods for stock prediction presents a new direction for stock selection, where both predictive performance and profits are jointly and directly optimized.

Contributions: We formulate stock prediction as a learning-to-rank problem (§3.1) and present FAST: **F**inancial News and Tweet based Time **A**ware Network for **S**tock **T**rading, which uses text

for maximizing profit by jointly optimizing predictive power and the optimal ranking of stocks. FAST learns *time-aware* representations of financial news and tweets, and captures *relevant* market signals using hierarchical temporal attention for ranking stocks (§3). Through experiments (§4) on English and Chinese text corresponding to the NASDAQ, Shanghai, Shenzhen, and Hong Kong markets, we show that FAST outperforms state-of-the-art methods in terms of intraday returns by over 8% and risk-adjusted returns by over 10% (§5.1, §5.2). Further, through exploratory (§5.3, §5.4) and qualitative (§6) analyses, we demonstrate the practical applicability of FAST to daily stock trading.

2 Background

Conventional Methods: Stock prediction spans various methods, commonly framed as regression or classification tasks (Jiang, 2020). Conventional methods rely on numeric features like historical prices (Kohara et al., 1997; Lin et al., 2009), technical (Shynkevich et al., 2017), and macroeconomic indicators (Hoseinzade et al., 2019). These include discrete (Bollerslev, 1986), continuous (Andersen, 2007), and neural approaches (Zhang et al., 2017; Feng et al., 2019a). Despite their success, a limitation of these methods is that they are limited to numerical features and do not factor crucial stock influencing factors such as text (Lee et al., 2014).

Contemporary Methods: Newer models based on the EMH, leverage natural language features extracted from investor sentiments (Li and Shah, 2017), financial reports (Kogan et al., 2009; Rekabsaz et al., 2017), earnings calls (Qin and Yang, 2019), online news (Peng and Jiang, 2016; Chen et al., 2019a,b; Du and Tanaka-Ishii, 2020) and social media (Si et al., 2013; Tabari et al., 2018; Sawhney et al., 2020a) for stock price regression and movement classification tasks. These methods show how NLP can complement conventional price-based methods in capturing the effect of events like market surprises and mergers over stock returns. However, these methods do not directly optimize profit, and do not factor the fine-grain irregularities in release times of stock affecting text. For stock trading, the timing of release of information across these sources plays a critical role, as price changes rapidly factor all publicly available information (Norman, 2014). Firms may exploit investors' perception of market information (Forbes, 2009), for instance, by timing the release
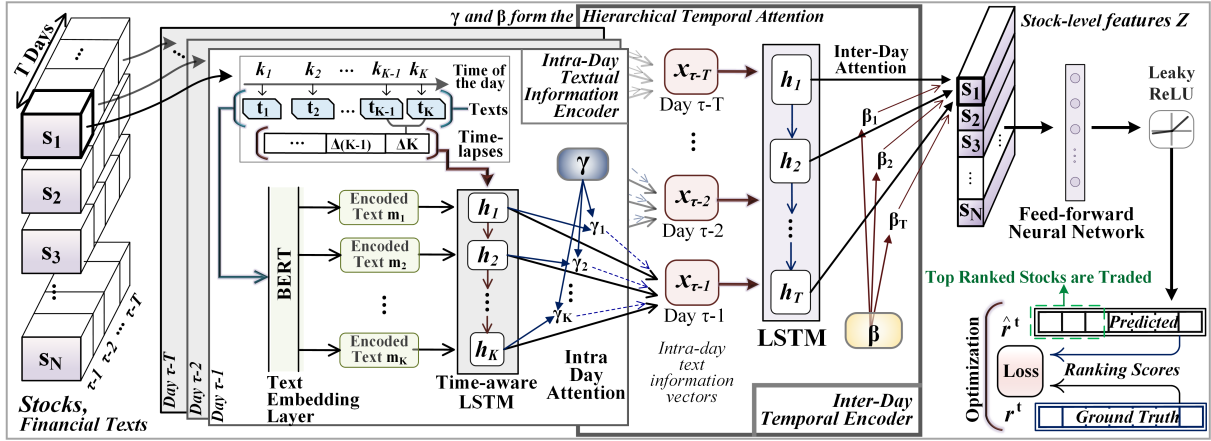
Figure 3: FAST: Time-aware modeling, hierarchical temporal attention, joint optimization for ranking.

of negative news between positive ones to minimize losses (Segal and Segal, 2016). These limitations hinder contemporary methods from modeling a time-aware progression of stock-affecting market signals to directly optimize profit generation.

Time-aware Methods: Recently, time-aware modeling of time series data has shown improvements over conventional sequential models like RNNs and LSTMs on various tasks such as patient subtyping (Baytas et al., 2017), suicide ideation and buildup detection using Twitter history (Sawhney et al., 2020b), disease progression (Gao et al., 2020), and much more. However, modeling the temporal dynamics inherent in social media and online news is complex as it involves noisy and diversely influential data across irregular time intervals. The intersection of modeling the temporal dynamics of natural language with finance presents an underexplored yet promising research avenue.

3 Proposed Approach: FAST

3.1 Problem Formulation: Stock Ranking

We adopt a learning to rank formulation for stock selection. Let $S = \{s_1, s_2, \ldots, s_N\}$ represent a set of N stocks, where for every stock $s_i \in S$, on a trading day τ, there is an associated closing price p^τ and a one-day return ratio $r_i^\tau = \frac{p^\tau - p^{\tau-1}}{p^{\tau-1}}$. On any given trading day τ, there exists an optimal *ranking* $Y^\tau = \{y_1^\tau > y_2^\tau \cdots > y_N^\tau\}$ of the stocks, such that a total order exists between the ranks $y_i^\tau > y_j^\tau$ for any two stocks $s_i, s_j \in S$, provided $r_i^\tau > r_j^\tau$. Such an ordering of stocks S on a trading day τ represents a ranking list, where stocks achieving higher ranking scores Y are expected to generate a higher investment revenue (profit) on

day τ. Formally, given stock-relevant textual data (financial news or tweets) for a lookback period of length T days (i.e., days $\in [\tau - T, \tau - 1]$), we aim to learn a ranking function that outputs a score $\hat{r}^\tau$ to rank each stock s on day τ in terms of their expected profit.

We now describe the components of FAST as shown in Figure 3, for *hierarchically* and *attentively* learning *time-aware* representations of news and tweets within (§**3.2**) and across (§**3.3**) the days in the lookback period. Lastly, we optimize FAST to rank stocks in terms of expected profitability (§**3.4**) for daily stock trading.

3.2 Intra-Day Textual Information Encoder

To model the news or tweets over a day, FAST first encodes the texts via an embedding layer.

Text Embedding Layer: Owing to the success of transfer learning and pre-training of language models in NLP, we use Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) to encode the texts. BERT has shown to capture more contextual text representations as opposed to methods like word2vec (Hu et al., 2017), GloVe (Xu and Cohen, 2018), ELMO (Mohammadi et al., 2019). We encode each text t to a higher dimensional representation $m = \text{BERT}(t) \in \mathbb{R}^d$ where $d = 768$, obtained by averaging the token level outputs from the final layer of BERT.

Learning Stock Representations for One Day: For each stock s on any given day i, a variable number of texts (news or tweets) $[t_1, t_2, \ldots t_K]$ are posted at times $[k_1, k_2, \ldots k_K]$ that may discuss news or express sentiments towards the stock. We encode each of the K texts posted in a day, using

BERT as $[m_1, m_2, \ldots m_K]$. Often, analyzing a single text alone may not be informative enough to analyze a stock (Barber and Odean, 2007). Whereas, analyzing a *sequence* of texts released over the day provides a unified context to gain a more informed understanding of the performance of a stock (Hu et al., 2017).

RNNs, particularly LSTMs are a natural way to capture such *sequential context dependencies* in tweets and news over time (Akhtar et al., 2017). However, a standard LSTM assume inputs (texts) to be equally spaced sequences in time. In contrast, the time interval between news releases or tweets can vary widely, from a few seconds to many hours that can have a drastic impact on their influence on the market (Robertson et al., 2007; O'Hara, 2015). Consequently, news and opinions may change substantially over a day. Capturing the fine-grained granularities in the posting times of online text can lead to better and quicker reactions to market opportunities and increased profits.[2]

Since timing serves as a crucial factor to model the *progression* of market data (Tafti et al., 2016), we propose the use of a **time-aware LSTM** (t-LSTM) (Baytas et al., 2017), by modifying a standard LSTM. We feed the time between texts to the t-LSTM cell to model the **temporal irregularities** in news and tweets. The t-LSTM applies a decay to the short-term memory in the LSTM according to the time elapsed between the release of two successively posted texts. Formally, the t-LSTM adopts a decaying function of elapsed time, transforming the time differences into appropriate weights for each input as:

$$C_{k-1}^S = tanh(W_d C_{k-1} + b_d) \quad \text{(Short-term memory)}$$
$$\tilde{C}_{k-1}^S = C_{k-1}^S * g(\Delta k) \quad \text{(Discounted short-term memory)}$$
$$C_{k-1}^{LT} = C_{k-1} - C_{k-1}^S \quad \text{(Long-term memory)}$$
$$C_{k-1}^* = C_{k-1}^{LT} + \tilde{C}_{k-1}^S \quad \text{(Adjusted previous memory)}$$

where C_{k-1} is the previous cell memory, $\{W_d, b_d\}$ are the network parameters, Δk is the elapsed time between two financial news or tweets $[t_k, t_{k-1}]$, and $g(\cdot)$ is a heuristic decaying function. We select $g(\Delta k) = 1/\Delta k$ empirically as suggested by Baytas et al. (2017). Intuitively, the greater the elapsed time between two news or tweets, the lesser impact they should have on each other due to the

market's dynamic nature. The t-LSTM computes the current hidden state h_k for each input text t_k generated in a day, as:

$$\tilde{C} = tanh(W_c m + U_c h_{k-1} + b_c) \quad \text{(Candidate memory)}$$
$$C_k = f_k * C_{k-1}^* + i_k * \tilde{C} \quad \text{(Current memory)}$$
$$h_k = o_k * tanh(C_k) \quad \text{(Current hidden state)}$$

where $\{W_c, U_c, b_c\}$ are the network parameters of the candidate memory $\tilde{C}$, m is the embedding for text t_k and $\{i_k, f_k, o_k\}$ are input, forget and output gates. We encode the texts for each stock s on day i using the t-LSTM as:

$$h_j = \text{t-LSTM}(m_j, \Delta j, h_{j-1}); j \in [1, K] \quad (1)$$

where the hidden state h_j represents the current text j as well as the preceding texts while focusing on text j in a time-aware fashion. All news and tweets released in a day may not be equally informative, and have *diverse influence* over a stock's trend (Barber and Odean, 2007). We use an intra-day attention mechanism (Luong et al., 2015) to emphasize texts likely to have a more substantial influence on the price. As shown in Figure 3, the intra-day attention mechanism learns to adaptively aggregate the variable number of hidden states of the t-LSTM (due to a variable number of texts per day) into an *intra-day text information vector* x_i as:

$$x_i = \sum_j \gamma_j h_j, \gamma_j = \frac{\exp\left(h_j^\mathsf{T}(W \bar{h}_m)\right)}{\sum_{j=1}^K \exp\left(h_j^\mathsf{T}(W \bar{h}_m)\right)} \quad (2)$$

where $\bar{h}_m \in \mathbb{R}^{K \times d_m}$ denotes the concatenation of all the hidden states from the t-LSTM, d_m is the dimension of each hidden state, γ_j represents the learned attention weights, and W is a parameter.

3.3 Inter-Day Temporal Encoder

We now combine the representations learned from texts in each day across multiple days in a look-back period. We combine these representations in a *hierarchical* manner within and across days, using the sequence of intra-day text information vectors x. Since days are equally spaced in time, we first feed the vectors x to an LSTM layer as:

$$h_i = \text{LSTM}(x_i, h_{i-1}); \quad \tau - T \le i \le \tau - 1 \quad (3)$$

where, h_i is the hidden state representation for day i. However, as per the the adaptive market hypothesis (Lo, 2004), tweets and news published across different days have shown to have a varying impact on stock prices (Calvet and Fisher,

---

[2]Every millisecond lost results in $100m per annum in lost opportunity. Details: https://en.wikipedia.org/wiki/Low_latency_(capital_markets)

2007), due to financial phenomena such as calendar anomalies (Jacobs and Levy, 1988), the week-day effect (Berument and Kiymaz, 2001), etc. To selectively weigh critical days, we employ an inter-day attention mechanism (Luong et al., 2015). The inter-day attention aggregates representations across all days into an *overall representation* z_τ using the learned attention weights β_i for day i as:

$$z_\tau = \sum_i \beta_i h_i, \beta_i = \frac{\exp\left(h_i^\mathsf{T}(W\overline{h}_z)\right)}{\sum_{i=1}^T \exp\left(h_i^\mathsf{T}(W\overline{h}_z)\right)} \quad (4)$$

where, W is a learned linear transform, $\overline{h}_z \in \mathbb{R}^{T \times d_z}$ represents the concatenated hidden states, d_z is the size of output space of LSTM. The inter-day and the intra-day attention together comprise a **hierarchical temporal attention**. `FAST` captures time-aware dependencies in large volumes of *chaotic* text to rank stocks, as described next.

3.4 Ranking and Network Optimization

To optimize `FAST` for stock ranking, we first concatenate the temporal representations z_τ obtained for each stock s to form stock-level features Z. We then feed Z to a feed-forward neural network followed by a Leaky-ReLU activation (Maas et al., 2013) which outputs the predicted return ratio $\hat{r}^\tau$ for stock ranking. We optimize `FAST` through a joint point-wise regression and pairwise rank-aware loss L, to minimize the differences between the predicted and the actual return ratios, while maintaining the relative order of the top-ranked stocks as:

$$L = \|\hat{r}^\tau - r^\tau\|^2 + \phi \sum_{i=1}^N \sum_{j=1}^N \max\left(0, -\left(\hat{r}_i^\tau - \hat{r}_j^\tau\right)\left(r_i^\tau - r_j^\tau\right)\right) \quad (5)$$

where $\hat{r}^\tau$ and r^τ are the predicted and actual scores for ranking stocks on day τ and ϕ is a loss weighing parameter.

4 Experimental Setup

4.1 Datasets and Stock Markets

US S&P 500[3] (Xu and Cohen, 2018): Comprises $109,915$ *English tweets* from social media platform Twitter spanning January 2014 to December 2015, related to 88 high-trade-volume-stocks from the *NASDAQ* Stock Exchange forming the *S&P 500* index. Xu and Cohen (2018) extract stock specific tweets using regex queries made of stock tickers (e.g., $AAPL for Apple, $ is a *cashtag* on Twitter).

China & Hong Kong[4] (Huang et al., 2018): Comprises $90,361$ financial *news headlines* in *Chinese* spanning from January to December 2015, aggregated by *Wind*[5] related to 85 top-traded *China A-share* stocks in Shanghai, Shenzhen and the Hong Kong Stock Exchange. Huang et al. (2018) extract corporate news from major financial websites.

Pre-processing: We pre-process English tweets using NLTK (Twitter mode), for treatment of URLs, identifiers (@) and hashtags (#). We adopt the Bert-Tokenizer for tokenization. For the English tweets, we use the pre-trained BERT-base-cased.[6] For the Chinese news, we adopt the Chinese-BERT.[6] We collect historical prices for all stocks from Yahoo Finance.[7] We align trading days by dropping samples that lack tweets for a consecutive 5-day trading window, and further align the data across trading windows for stocks to ensure data is available for all trading days in the window for the same set of stocks. We split the US S&P 500 temporally based on date ranges from 01/01/2014 to 31/07/2015 for training, 01/08/2015 to 30/09/2015 for validation, and 01/10/2015 to 01/01/2016 for testing. We split the China & HK dataset temporally based on date ranges from 01/01/2015 to 31/08/2015 for training, 01/09/2015 to 30/09/2015 for validation, and 01/10/2015 to 01/01/2016 for testing all models.

4.2 `FAST` Training Setup

We conduct all experiments on a Tesla P100 GPU. We use grid search to find optimal hyperparameters based on the validation Sharpe Ratio (§4.3) for all models. We explore lookback window length $T \in \text{range}[2, 10]$ (best $T = 5$), loss weighing factor $\phi \in \text{range}[1, 10]$ (best $\phi = 4$), and the hidden state dimension for both t-LSTM and LSTM $d \in [32, 64, 128]$ (best $d = 64$) for both datasets. We use Xavier initialization (Glorot and Bengio, 2010) to initialize all weights. We use an exponential learning rate scheduler (Li and Arora, 2019) with a decay rate of 0.05 and an initial learning rate of $5e-4$. We train `FAST` end-to-end using the Adam optimizer (Kingma and Ba, 2014) for 500 epochs requiring 8 hours of compute time.

---

[3]US S&P 500 dataset: www.github.com/yumoxu/stocknet-dataset

[4]China & Hong Kong dataset: https://pan.baidu.com/s/1mhCLJJi

[5]https://www.wind.com.cn/en/wft.html

[6]www.github.com/google-research/bert

[7]Prices from: https://finance.yahoo.com/

4.3 Evaluation Metrics and Trading Strategy

Returns: To assess the profit generation ability of all methods (§**4.4**), we compute the **Sharpe ratio (SR)**, a measure of the return of a portfolio compared to its risk (Sharpe, 1994), and the cumulative **investment return ratio (IRR)**. Following Feng et al. (2019b), we adopt a daily *buy-hold-sell* trading strategy, that is, when the market closes on trading day $\tau - 1$, the trader uses the method to get a ranked list of the predicted return ratio for each stock. The trader then buys the top η stocks and then sells the bought stocks on the market close of the trading day τ. The IRR on any day τ is defined as, $\text{IRR}^\tau = \sum_{i \in \mathcal{S}^{\tau-1}} \frac{p_i^\tau - p_i^{\tau-1}}{p_i^{\tau-1}}$ where, $\mathcal{S}^{\tau-1}$ denotes the set of stocks in the portfolio on day $\tau - 1$ and p_i^τ, $p_i^{\tau-1}$ are the closing price of the stock i on days τ and $\tau - 1$ respectively. We calculate SR by computing the earned return R_a in excess of the risk-free return[8] R_f, defined as: $\text{SR} = \frac{E[R_a - R_f]}{std[R_a - R_f]}$.

Ranking: We also evaluate the stock ranking ability of FAST using two widely-used ranking metrics: **Mean Reciprocal Rank (MRR)** and **Normalized Discounted Cumulative Gain (NDCG@η)**. MRR is the reciprocal rank of the first relevant stock while, NDCG@η sums the true scores for the top η stocks, ranked in the order induced by the predicted scores, after applying a logarithmic discount. For both returns and NDCG, we report results for top $\eta = 5$ stocks, and present performance variations with different values of top stocks η (§**5.4**).

4.4 Baselines

We compare FAST with baselines spanning different formulations: *regression*, *classification*, *reinforcement learning*, and *ranking*. We follow the same preprocessing protocols as proposed by the works and adopt their implementation, if available.

Regression (REG) These methods regress return ratios from past data and trade the top stocks.

- **AZFinText:** Proper noun-based text representations fed to Support Vector Regression for forecasting return ratios (Schumaker and Chen, 2009).

- **W-LSTM:** LSTMs with stacked autoencoders that encode noise-free data obtained through

wavelet transform of historic prices (Bao et al., 2017).

Classification (CLF) The following methods classify movements as $[up, down, neutral]$ and trade the stocks where prices are expected to rise.

- **TSLDA:** Topic Sentiment Latent Dirichlet Allocation, a generative model jointly exploiting topics and sentiments in text (Nguyen and Shirai, 2015).

- **CH-RNN:** An RNN with cross-modal attention on price trends and texts across days (Wu et al., 2018).

- **StockEmb:** Stock embeddings acquired using prices, and dual vector (word-level, context-level) representation of texts (Du and Tanaka-Ishii, 2020).

- **SN - HFA:** StockNet - HedgeFundAnalyst, a variational autoencoder with attention on texts and prices (Xu and Cohen, 2018).

- **HAN:** A Hierarchical Attention Network using GRU encoders with temporal attention on texts and days in the lookback period (Hu et al., 2017).

Reinforcement Learning (RL) The following approaches optimize quantitative trading through reinforcement learning.

- **iRDPG:** An imitative RDPG algorithm exploiting temporal stock price features, while optimizing the Sharpe Ratio as the reward (Liu et al., 2020).

- **S-Reward:** Inverse-RL method to model relations between sentiments and returns (Yang et al., 2018).

Ranking (RAN) The following methods rank stocks to select most profitable trading candidates.

- **R-LSTM:** Utilizes 5, 10, 20, 30-day averages and closing prices to train an LSTM (Feng et al., 2019b).

- **RankNet:** A DNN that utilizes sentiment-based shock and trend scores to optimize a probabilistic ranking function (Song et al., 2017).

---

[8]T-Bill rates: https://home.treasury.gov/

FAST: Model Components			US S&P 500				China & Hong Kong			
Intra-Day Encoder	Intra-Day Attention	Inter-Day Attention	MRR↑	IRR↑	NDCG↑	SR↑	MRR↑	IRR↑	NDCG↑	SR↑
s-LSTM	–	–	0.050	0.794	0.521	0.571	0.072	0.861	0.580	0.663
LSTM	✗	✗	0.056	0.913	0.582	0.654	0.074	0.902	0.623	0.714
LSTM	✗	✓	0.060	0.922	0.611	0.713	0.080	1.027	0.695	0.773
LSTM	✓	✗	0.063	1.097	0.630	0.865	0.090	1.148	0.728	0.901
LSTM	✓	✓	0.064	1.214*	0.725*	0.919	0.096*	1.244*	0.756*	0.965*
t-LSTM	✗	✗	0.063	1.131	0.621	0.842	0.095	1.160	0.703	0.911
t-LSTM	✗	✓	0.065*	1.224*†	0.703	0.933*	0.109*	1.295*	0.759*	1.018*
t-LSTM	✓	✗	*0.067*†	*1.267*†	*0.760*	*0.949*†	*0.114*†	*1.383*†	*0.804*†	*1.165*†
t-LSTM	✓	✓	**0.068*†**	**1.336*†**	**0.762*†**	**0.957*†**	**0.117*†**	**1.441*†**	**0.810*†**	**1.193*†**

Table 1: Ablation study over components of FAST. s-LSTM is a single LSTM layer, which models all texts within and across days. All model variations except s-LSTM use LSTM as the Temporal Encoder. Intense color indicates better results. **Bold**, *italics* indicate **best**, *second best* results. ⋆ & † imply statistically significant (p<0.001) improvements over s-LSTM & RankNet (Wilcoxon's signed rank test). ↑ indicates that higher values are better.

			US S&P500		China&HK	
	Model & Components		IRR↑	SR↑	IRR↑	SR↑
REG	AZFinText	T + P	0.42	0.40	0.49	0.50
	W-LSTM	P	0.40	0.41	0.47	0.49
CLF	TSLDA	T + P	0.40	0.39	0.50	0.51
	CH-RNN	T + P + A	0.78	0.54	0.83	0.70
	StockEmb	T + P + A	0.70	0.51	0.89	0.74
	SN - HFA	T + P + A	1.09	0.81	1.12	0.93
	HAN	T + A	1.07	0.80	1.20	1.01
RL	iRDPG	P	1.05	0.79	1.19	1.03
	S-Reward	T	0.93	0.73	1.33	1.08
RAN	R-LSTM	P	1.01	0.78	1.17	0.96
	RankNet	T	1.16*	0.87*	1.14	0.95
	FAST	T + A	1.34*†	0.96*†	1.44*†	1.19*†

Table 2: Profitability comparison with baselines (§4.4). Within Components, T = Text, P = Prices, A = Attention across modalities. Green & Blue depict Best & Second-best (SOTA) results. ⋆ & † imply statistically significant (p<0.001) improvements over S-Reward and RankNet (under Wilcoxon's signed rank test).

5 Results and Discussion

5.1 Profitability Comparison with Baselines

As the ultimate goal of stock prediction is profit, we compare the profitability of FAST against baseline methods in Table 2. FAST generates significantly ($p < 0.001$) higher cumulative and risk-adjusted returns than all methods. Overall, we observe that RL and ranking methods are more profitable as they are directly optimized towards profit generation through stock selection. This observation validates the premise of formulating stock prediction as a learning-to-rank problem, compared to conventionally adopted regression and classification tasks.

Further, we find methods that study stock affecting information from news and tweets, generate profits higher or comparable to methods that only use historical prices. These improvements revalidate the effectiveness of leveraging textual sources to capture stock affecting signals like market surprises, announcements (mergers, acquisitions) and public sentiment. We attribute the higher profitability of FAST to two major reasons. First, through its hierarchical temporal attention mechanism, FAST captures the diverse influence of different texts and days over stock movements. Second, as FAST is time-aware, it models the influence of fine-grain temporal irregularities in the release of financial news and tweets over stock movements.

The test periods of the US S&P 500 and the China & Hong Kong datasets span diverse market conditions. The China & Hong Kong test period covers the 2015-16 China Stock Market Turbulence (Liu et al., 2016), a bearish market scenario,[9] and that of US S&P 500 covers standard market conditions. We find that FAST is profitable and outperforms existing baselines over such diverse market scenarios. Next, we further probe the performance of FAST through a series of ablative experiments.

5.2 FAST Component Ablation

Table 1 shows how FAST's stock ranking ability and profitability benefits from each of its components. On just feeding all data into a single LSTM, we observe poor performance. As we adopt the intra-day (LSTM) and the temporal (LSTM) encoders, we observe higher profits that suggest benefit in modeling the sequential context of texts hierarchi-

---

[9]Bearish markets are those that experience prolonged price declines, and experience high volatility and risk on investments.
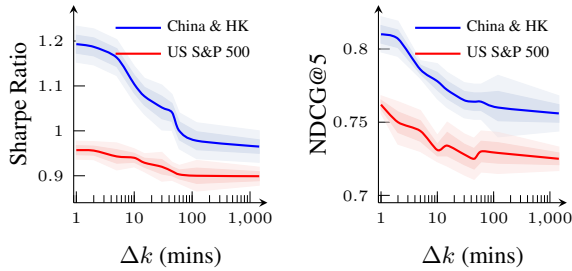
Figure 4: Profitability and ranking performance against granularities of time difference Δk adopted by t-LSTM

Figure 5: Sensitivity to parameters T and η

cally within and across days. Further, on adding intra-day attention, we note improvements in profit as FAST can better distinguish noise inducing text from relevant market signals, minimizing false evaluations and overreactions (De Long et al., 1990). The attention mechanism can likely diminish the impact of such noise (rumours, vague comments). Intuitively, complementing the intra-day with the inter-day attention lead to further improvements as FAST can better capture the *diverse influence* of texts, hierarchically within and across days.

Next, we note the biggest improvements on adding the t-LSTM instead of a standard LSTM as the intra-day encoder, suggesting that FAST benefits by factoring the fine-grain time irregularities in texts to model the flow of stock-affecting information (Kalev et al., 2004). Through this time-aware mechanism, FAST can potentially better react to online news and tweets, by discounting stale information more accurately by factoring in fine-grained elapsed time differences (seconds). We further quantify the impact of time-aware modeling on the improvements in ranking and profitability, next.

5.3 Advantages of Time-Aware Modeling

The influence of older information over the market decreases rapidly as newer data is released (Russell, 2010). As we coarsen the granularity of the elapsed time difference between two texts fed to the t-LSTM, from minutes, to hours, to a day, we observe drops in FAST's performance as shown in Figure 4. At the coarsest granularity of a day, the t-LSTM essentially degenerates to an LSTM, and attains the lowest performance. The drops show that factoring time at finer granularities benefits FAST in modeling the temporal dynamics of market response to stock affecting signals. Our findings line with financial research that shows market reactions to news complete within minutes (Smales, 2013), and the impact of news and tweets achieves
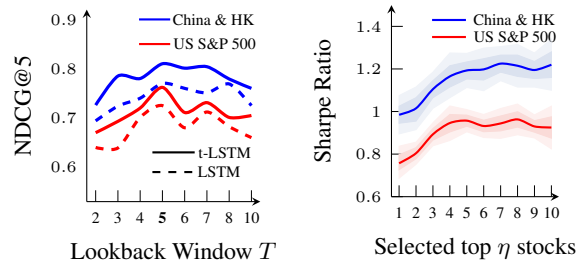
an equilibrium over time (Shen et al., 2018).

5.4 Parameter Analysis: Probing Sensitivity

Lookback window length T: Here, we study how FAST's stock ranking performance (NDCG@5) varies with the length of lookback $T \in [2, 10]$ days in Figure 5. Lower ranking performance indicates the inability of shorter lookbacks to capture stock affecting market information, likely as public information requires time to absorb into price movements (Luss and D'Aspremont, 2015). As we increase T, we find that larger lookbacks allow the inclusion of stale information from older days having relatively lower influence on prices (Bernhaedt and Miao, 2004), thus deteriorating the ranking performance. We observe the best stock ranking performance for mid-sized (approx. 5-day) lookback periods.

Selected top stocks η: We analyze FAST's profitability (SR) variation with the number of top stocks η in Figure 5. We find that FAST performs well across varying η, showing suitability to strategies having different risk taking appetites.

6 FAST Qualitative Analysis

We now conduct an extended analysis as shown in Figure 6 to elucidate on FAST's explainable predictions and practical applicability to real-world quantitative trading. Here, we study the China & Hong Kong market during 5th - 9th December 2015. We visualize token-level and hierarchical temporal attention to analyze how FAST ranks stocks on 10th December 2015, outperforming the state-of-the-art baseline methods: RankNet and SN-HFA.

Analyzing Hierarchical Attention: Within days, the intra-day attention filters less informative news and emphasizes more influential ones. For instance, we observe that the second news about BOE Technology on 7th categorizes the stock as "overweight," a rating through which equity
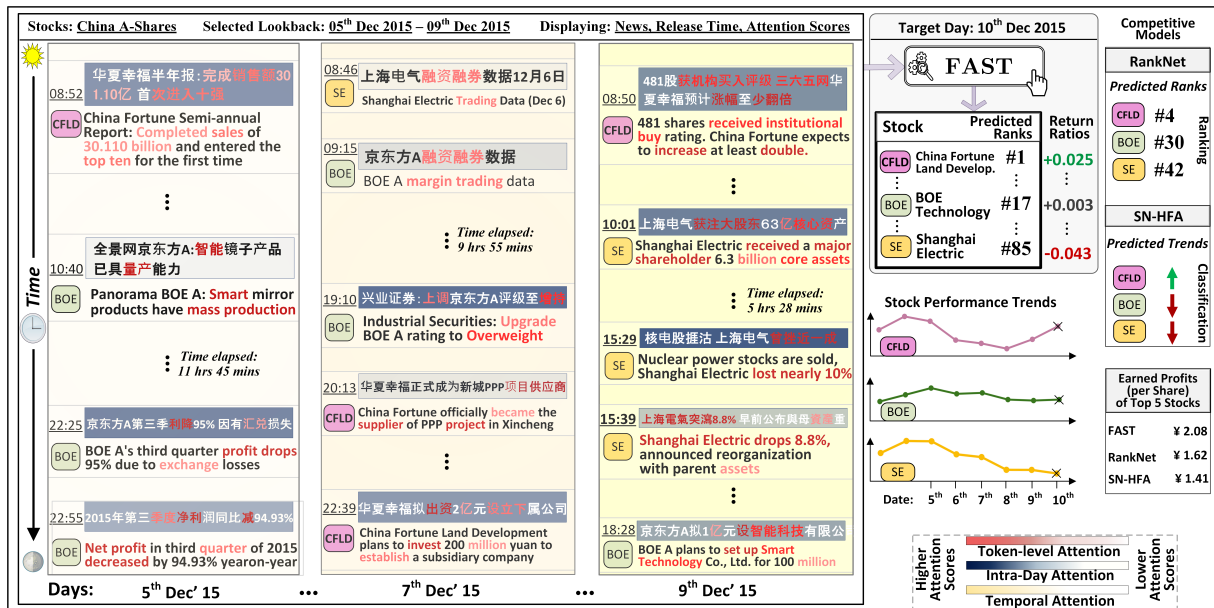
Figure 6: Financial news in Chinese (shown with corresponding English translations) during 5th-9th December 2015 for stocks in the China & Hong Kong dataset (test split), with visualization of FAST's attention mechanisms (left); Stock trading performance for 10th December 2015 across FAST and competitive baseline methods (right).

analysts forecast better future performance (Kumar, 2009). Such news would likely induce positive public sentiments and drive more investment to the stock, as opposed to the less informative news about "margin trading data." The intra-day attention accurately captures the *diverse influence* of such news headlines. Further, we observe that news released on 9th comprises relatively more crucial information than other days, and the inter-day attention accurately emphasizes its importance. These observations reiterate the diverse influence of different news and days over future stock returns, accurately captured by the hierarchical temporal attention mechanism.

Probing Time-aware Modeling: The news released for Shanghai Electric (SE) on the morning of 9th reports a positive event, likely indicating future profits. Later during that afternoon, after a few hours, two other news report negative impacts over SE due to a loss in nuclear power stocks, indicating a downtrend for the upcoming days. FAST disregards older news to emphasize on the newer ones to forecast the upcoming loss and allots a lower rank to SE. In contrast, ranking methods such as RankNet assign a higher rank to SE, potentially due to their inability to model the time-aware dependencies. Further, classification methods like SN-HFA do not correctly predict the stock return trends, as they do not capture the fine-grain temporal irreg-

ularities in texts, and are not optimized towards profit. Consequently, FAST relatively outperforms SN-HFA by a margin of 47.5%, and RankNet by 28.4% in profits on 10th December 2015.

7 Conclusion

We propose FAST, a neural approach to rank profitable stocks using stock-relevant textual data across online financial news and tweets. To model the market information, FAST hierarchically learns temporally relevant signals from texts and shows the positive effects of factoring the fine-grain temporal irregularities in textual data. Through quantitative and qualitative analyses on English tweets and Chinese financial news spanning four stock markets, we highlight the real-world applicability of FAST. In trading simulations on the S&P 500 and China A-shares indexes, FAST outperforms state-of-the-art methods across four different formulations by over 8% in terms of profit and Sharpe Ratio.

References

Klaus Adam, Albert Marcet, and Juan Pablo Nicoli. 2016. Stock market volatility and learning. *The Journal of Finance*, 71(1):33–82.

Md Shad Akhtar, Abhishek Kumar, Deepanway Ghosal, Asif Ekbal, and Pushpak Bhattacharyya. 2017. A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis.

In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 540–546, Copenhagen, Denmark. Association for Computational Linguistics.

Leif BG Andersen. 2007. Efficient simulation of the heston stochastic volatility model. *Available at SSRN 946405*.

Wei Bao, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*.

Brad M. Barber and Terrance Odean. 2007. All That Glitters: The Effect of Attention and News on the Buying Behavior of Individual and Institutional Investors. *The Review of Financial Studies*, 21(2).

Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. 2017. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 65–74, New York, NY, USA. Association for Computing Machinery.

Dan Bernhaedt and Jianjun Miao. 2004. Informed trading when information becomes stale. *The Journal of Finance*, 59(1):339–390.

Hakan Berument and Halil Kiymaz. 2001. The day of the week effect on stock market volatility. *Journal of economics and finance*, 25(2):181–193.

Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.

Laurent E. Calvet and Adlai J. Fisher. 2007. Multifrequency news and stock returns. *Journal of Financial Economics*, 86(1):178 – 212.

Deli Chen, Shuming Ma, Keiko Harimoto, Ruihan Bao, Qi Su, and Xu Sun. 2019a. Group, extract and aggregate: Summarizing a large amount of finance news for forex movement prediction. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*, pages 41–50, Hong Kong. Association for Computational Linguistics.

Deli Chen, Yanyan Zou, Keiko Harimoto, Ruihan Bao, Xuancheng Ren, and Xu Sun. 2019b. Incorporating fine-grained events in stock movement prediction. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*, pages 31–40, Hong Kong. Association for Computational Linguistics.

J Bradford De Long, Andrei Shleifer, Lawrence H Summers, and Robert J Waldmann. 1990. Noise trader risk in financial markets. *Journal of political Economy*, 98(4):703–738.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xin Du and Kumiko Tanaka-Ishii. 2020. Stock embeddings acquired from news articles and price history, and an application to portfolio optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3353–3363, Online. Association for Computational Linguistics.

Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019a. Enhancing stock movement prediction with adversarial training.

Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019b. Temporal relational ranking for stock prediction. *ACM Trans. Inf. Syst.*, 37(2).

William Forbes. 2009. *Behavioural finance*. John Wiley & Sons.

Thierry Foucault, Johan Hombert, and Ioanid Roşu. 2016. News trading and speed. *The Journal of Finance*, 71(1):335–382.

Junyi Gao, Cao Xiao, Yasha Wang, Wen Tang, Lucas M. Glass, and Jimeng Sun. 2020. Stagenet: Stage-aware neural networks for health risk prediction. In *Proceedings of The Web Conference 2020*, WWW '20, page 530–540, New York, NY, USA. Association for Computing Machinery.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Ehsan Hoseinzade, Saman Haratizadeh, and Arash Khoeini. 2019. U-cnnpred: A universal cnn-based predictor for stock markets.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2017. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the 11 international conference on web search and data mining*.

Jieyun Huang, Yunjia Zhang, Jialai Zhang, and Xi Zhang. 2018. A tensor-based sub-mode coordinate algorithm for stock prediction.

Bruce I Jacobs and Kenneth N Levy. 1988. Calendar anomalies: Abnormal returns at calendar turning points. *Financial Analysts Journal*, 44(6):28–39.

Weiwei Jiang. 2020. Applications of deep learning in stock market prediction: recent progress.

Petko S Kalev, Wai-Man Liu, Peter K Pham, and Elvis Jarnecic. 2004. Public information arrival and volatility of intraday stock returns. *Journal of Banking & Finance*, 28(6):1441–1467.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

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*, pages 272–280, Boulder, Colorado. Association for Computational Linguistics.

Kazuhiro Kohara, Tsutomu Ishikawa, Yoshimi Fukuhara, and Yukihiro Nakamura. 1997. Stock price prediction using prior knowledge and neural networks. *Intelligent Systems in Accounting, Finance & Management*, 6(1):11–22.

Alok Kumar. 2009. Who gambles in the stock market? *The Journal of Finance*, 64(4):1889–1933.

Helinä Laakkonen. 2004. The impact of macroeconomic news on exchange rate volatility. *Bank of Finland discussion paper*, (24).

Heeyoung Lee, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. 2014. On the importance of text analysis for stock price prediction. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1170–1175, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Quanzhi Li and Sameena Shah. 2017. Learning stock market sentiment lexicon and sentiment-oriented word vector from StockTwits. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 301–310, Vancouver, Canada. Association for Computational Linguistics.

Zhiyuan Li and Sanjeev Arora. 2019. An exponential learning rate schedule for deep learning.

Xiaowei Lin, Zehong Yang, and Yixu Song. 2009. Short-term stock price prediction based on echo state networks. *Expert systems with applications*, 36(3):7313–7317.

Dehong Liu, Hongmei Gu, and Tiancai Xing. 2016. The meltdown of the chinese equity market in the summer of 2015. *International Review of Economics & Finance*, 45:504 – 517.

Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. 2020. Adaptive quantitative trading: An imitative deep reinforcement learning approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):2128–2135.

Andrew W. Lo. 2004. The adaptive markets hypothesis. *The Journal of Portfolio Management*, 30(5):15–29.

Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. 2009. Financial time series forecasting using independent component analysis and support vector regression. *Decision support systems*, 47(2):115–125.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation.

Ronny Luss and Alexandre D'Aspremont. 2015. Predicting abnormal returns from news using text classification. *Quantitative Finance*, 15(6):999–1012.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models.

Burton G Malkiel. 1989. Efficient market hypothesis. In *Finance*, pages 127–134. Springer.

Elham Mohammadi, Hessam Amini, and Leila Kosseim. 2019. CLaC lab at SemEval-2019 task 3: Contextual emotion detection using a combination of neural networks and SVM. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 153–158, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1354–1364, Beijing, China. Association for Computational Linguistics.

George Norman. 2014. *Semi-Strong Form Efficient Market Hypothesis*. Edward Elgar Publishing.

Nuno Oliveira, Paulo Cortez, and Nelson Areal. 2017. The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144.

Maureen O'Hara. 2015. High frequency market microstructure. *Journal of Financial Economics*, 116(2):257 – 270.

Yangtuo Peng and Hui Jiang. 2016. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 374–379, San Diego, California. Association for Computational Linguistics.

Yu Qin and Yi Yang. 2019. What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 390–401, Florence, Italy. Association for Computational Linguistics.

Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Alexander Dür, Linda Andersson, and Allan Hanbury. 2017. Volatility prediction using financial disclosures sentiments with word embedding-based IR models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1712–1721, Vancouver, Canada. Association for Computational Linguistics.

Calum Robertson, Shlomo Geva, and Rodney Wolff. 2007. Predicting the short-term market reaction to asset specific news: Is time against us? In *Emerging Technologies in Knowledge Discovery and Data Mining*, pages 15–26, Berlin, Heidelberg. Springer.

Jeffrey R Russell. 2010. Market microstructure effects. *Encyclopedia of Quantitative Finance*.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn Shah. 2020a. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8415–8426, Online. Association for Computational Linguistics.

Ramit Sawhney, Harshit Joshi, Saumya Gandhi, and Rajiv Ratn Shah. 2020b. A time-aware transformer based model for suicide ideation detection on social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7685–7697, Online. Association for Computational Linguistics.

Martin Scholtus, Dick van Dijk, and Bart Frijns. 2014. Speed, algorithmic trading, and market quality around macroeconomic news announcements. *Journal of Banking and Finance*, 38:89 – 105.

Robert P. Schumaker and Hsinchun Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27(2).

Benjamin Segal and Dan Segal. 2016. Are managers strategic in reporting non-earnings news? evidence on timing and news bundling. *Review of Accounting Studies*, 21(4):1203–1244.

William F Sharpe. 1994. The sharpe ratio. *Journal of portfolio management*, 21(1):49–58.

Dehua Shen, Xiao Li, and Wei Zhang. 2018. Baidu news information flow and return volatility: Evidence for the sequential information arrival hypothesis. *Economic Modelling*, 69:127 – 133.

Yauheniya Shynkevich, T.M. McGinnity, Sonya A. Coleman, Ammar Belatreche, and Yuhua Li. 2017. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, 264:71 – 88. Machine learning in finance.

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)*, pages 24–29, Sofia, Bulgaria. Association for Computational Linguistics.

Lee A. Smales. 2013. Impact of macroeconomic announcements on interest rate futures: High-frequency evidence from australia. *Journal of Financial Research*, 36(3):371–388.

Qiang Song, Anqi Liu, and Steve Y. Yang. 2017. Stock portfolio selection using learning-to-rank algorithms with news sentiment. *Neurocomputing*, 264:20 – 28. Machine learning in finance.

Narges Tabari, Piyusha Biswas, Bhanu Praneeth, Armin Seyeditabari, Mirsad Hadzikadic, and Wlodek Zadrozny. 2018. Causality analysis of Twitter sentiments and stock market returns. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 11–19, Melbourne, Australia. Association for Computational Linguistics.

Ali Tafti, Ryan Zotti, and Wolfgang Jank. 2016. Real-time diffusion of information on twitter and the financial markets. *PloS one*, 11(8):e0159226.

Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. 2018. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1627–1630, New York, NY, USA. Association for Computing Machinery.

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)*, pages 1970–1979, Melbourne, Australia. Association for Computational Linguistics.

Steve Y. Yang, Yangyang Yu, and Saud Almahdi. 2018. An investor sentiment reward-based trading system using gaussian inverse reinforcement learning algorithm. *Expert Systems with Applications*, 114:388 – 401.

Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 2141–2149, New York, NY, USA. Association for Computing Machinery.