Quantitative Day Trading from Natural Language using Reinforcement Learning

Ramit Sawhney*
IIIT Delhi
ramits@iiitd.ac.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

It is challenging to design profitable and practical trading strategies, as stock price movements are highly stochastic, and the market is heavily influenced by chaotic data across sources like news and social media. Existing NLP approaches largely treat stock prediction as a classification or regression problem and are not optimized to make profitable investment decisions. Further, they do not model the temporal dynamics of large volumes of diversely influential text to which the market responds quickly. Building on these shortcomings, we propose a deep reinforcement learning approach that makes time-aware decisions to trade stocks while optimizing profit using textual data. Our method outperforms state-of-the-art in terms of risk-adjusted returns in trading simulations on two benchmarks: Tweets (English) and financial news (Chinese) pertaining to two major indexes and four global stock markets. Through extensive experiments and studies, we build the case for our method as a tool for quantitative trading.

1 Introduction

The stock market, a financial ecosystem involving quantitative trading and investing, observed a market capitalization exceeding $US 60 trillion as of the year 2019. 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, noisy, and chaotic nature (Tsay, 2005; Adam et al., 2016). Research at the intersection of Natural Language Processing (NLP) and finance presents encouraging prospects in stock prediction (Jiang, 2020). Conventional work forecasts future trends by modeling numerical historical stock data (Lu et al., 2009; Bao et al., 2017). However, price signals alone can not capture market surprises, mergers, acquisitions, and company announcements. Such events, often reported across financial news and social media, have strong influence over market dynamics (Laakkonen, 2004). For instance, prices immediately react to breaking news about the related company (Busse and Green, 2002). Such reactions conform to the Efficient Market Hypothesis (EMH), a hypothesis in finance which states that financial markets are *informationally efficient* and prices reflect all available market information at any given time (Malkiel, 1989).

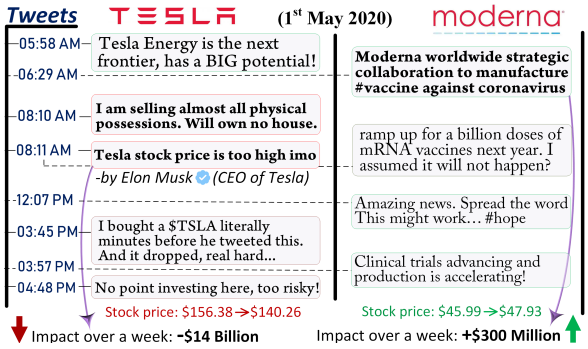

Figure 1: Here, we show how tweets about Tesla and Moderna influence investors' opinions and impact the stocks over a day and the upcoming week. The tweets by the Tesla CEO Elon Musk lead to massive price drops in Tesla's stock, and Moderna's positive news attracts investments in its stock. A profitable trading decision would entail selling off Tesla's shares (if already held) and buying Moderna's stock in such a scenario.

The abundance of stock affecting information across news and social media online inspires the adoption of natural language processing to study the interplay between textual data and stock prices (Oliveira et al., 2017; Xu and Cohen, 2018). However, unlike numerical data, the study of natural language is more challenging. Individual tweets or news headlines may not be informative enough, and analyzing them together can provide a greater

*Equal contribution.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4018–4030
June 6–11, 2021.

context, as shown in Figure 1. Moreover, the timing of their release plays a critical role as stock markets rapidly react to new information (Foucault et al., 2016). Furthermore, not each news story or tweet holds the potential to influence stock trends as texts have a diverse influence on prices (Hu et al., 2017). These observations suggest benefits in factoring in the *time-aware dependence* and *diverse influence* of text while analyzing natural language.

Despite profitability being the prime objective of quantitative trading, existing natural language processing methods for stock prediction (Hu et al., 2017; Xu and Cohen, 2018; Du and Tanaka-Ishii, 2020) are commonly formulated as classification or regression tasks, and are not directly optimized towards profit generation. Such methods face fundamental drawbacks. First, they do not innately incorporate the decision making and strategies involved in quantitative trading, in turn limiting potential profitability. Second, they have limited practical applicability as they do not factor in the monetary resources available and financial assets (stocks) held with a trader at each trading time step. This gap presents a new research direction where profit generation can be directly optimized by modeling the complex sequential decision-making process in quantitative trading as a Reinforcement Learning (RL) task. Owing to its nature, RL formulation is directly suitable to the problem of quantitative trading as it provides the potential to automatically learn the adjustment of investment budgets across stocks in portfolios while taking into account the configuration of investments made in the past.

Contributions: We formulate stock prediction as a reinforcement learning problem (**§3**) and present **PROFIT**: **P**olicy for **R**eturn **O**ptimization using **FI**nancial news and online **T**ext, a deep reinforcement learning approach that leverages financial news and tweets to model stock-affecting signals and optimize trading decisions for increasing profitability. PROFIT accounts for the monetary resources available and the existing portfolio to execute profitable trades at any given time. Through extensive experiments (**§5**) on English and Chinese text corresponding to the NASDAQ, Shanghai, Shenzhen, and Hong Kong markets, we show that PROFIT outperforms state-of-the-art methods in terms of risk adjusted returns by over 13% and minimizes extreme losses by over 16% (**§6.1**, **§6.2**). Using exploratory analyses (**§6.3**), we show PROFIT's practical and real-world applicability.

2 Background

Reinforcement Learning and Natural Language Processing Lately, reinforcement learning has influenced solutions for a wide variety of natural language processing tasks and applications. These include, but are not limited to information extraction (Qin et al., 2018), social media analysis (Zhou and Wang, 2018), text classification (Wu et al., 2018a), extractive (Narayan et al., 2018) and abstractive (Chen and Bansal, 2018) text summarization, neural machine translation (Wu et al., 2018b), text-based games (He et al., 2016a; Ammanabrolu and Riedl, 2019), knowledge-based question answering (Hua et al., 2020) and much more. For these tasks and applications, *deep* reinforcement learning methods have been more successful in modeling the complexities involved in natural language, such as the processing of large vocabularies and phrases that otherwise make action selection (He et al., 2016a,b) arduous for RL methods that do not exploit deep networks as function approximators. However, most existing methods for a variety of tasks face a fundamental drawback – they do not take into account the influence of the inherent dynamic temporal irregularities and the variably influential nature of text while modeling a time-series of language data over action selection and sequential decision making.

Reinforcement Learning in Finance Recent years have witnessed the adoption of reinforcement learning in the financial realm to solve tasks such as portfolio management (Filos, 2019; Almahdi and Yang, 2019), equity asset reallocation (Meng and Khushi, 2019; Katongo and Bhattacharyya, 2021), cryptocurrency trading (Jiang et al., 2017; Jiang and Liang, 2017; Lucarelli and Borrotti, 2019; Ye et al., 2020) and much more. Existing work heavily relies on factors such as technical indicators (Wang et al., 2019; Liu et al., 2020) to model price signals, or use simple numeric features like sentiment scores from text (Yang et al., 2018) to model stock affecting information reflected across news items. However, these methods experience two significant drawbacks. Firstly, despite their success, the performance of such methods depend largely on the quality of external feature representations (for instance, sentence embeddings (Ye et al., 2020)) of text. Secondly, methods that only use prices exhibit lower practical applicability to real-world trading, owing to the lack of information in prices alone.

3 Problem Description

We formulate stock trading as a *reinforcement learning* problem. Let $S = \{s_1, s_2, \ldots, s_N\}$ denote a set of N stocks. We aim to design a trading *agent* that learns to interact with the stock market *environment* by leveraging stock-affecting signals present across financial news items and tweets to trade stocks. In the context of an agent, an *interaction* comprises observing the environment state at any particular time-step to generate an action, and reach the next time-step to receive a reward along with the next state. The typical Markov Decision Process (MDP) description is widely adopted for RL tasks where environments are fully-observable. However, in the stock market, prices are influenced by numerous macro- and micro-economic factors, investor opinions about stocks formed through social media, financial news, and countless other sources. Thus, it becomes pragmatically and computationally impractical to *observe* and incorporate stock affecting information from *all* possible sources to make trading decisions. As the stock markets and the underlying factors that drive stock prices are not fully-observable, Partially Observable MDP (POMDP) provides a natural generalization of the MDP to model the stock trading environment (Jaakkola et al., 1995). Hence, the key components of the stock trading environment considered and developed in this study are as follows:

State observations: At a time-step τ, the *state* s_τ comprises a *trading-account* observation o_τ, and a *market-information* observation o_m. The trading-account observation o_τ comprises the account balance and the number of shares owned corresponding to each stock at time-step τ. The market-information observation o_m comprises stock-relevant news or tweets released during a T-day lookback period (days $\in [\tau - T + 1, \tau]$). The text input in o_m is structured such that it comprises all stock relevant text in a lookback window of length T in a hierarchical fashion within and across days. The orders made through the trading actions taken by the reinforcement learning agent would have minute impacts on the overall market trends, thus having little to no direct influence on the market-information observations.

Trading actions: The agent can buy, sell, or hold the shares for each stock at the time-step τ. We compute a vector of actions a_τ over the set of stocks S as decisions made by the agent, which result in an increase, decrease, or no change in the number of stocks shares h. One of three possible actions is taken on each stock s:

- *Buying* $k[s] \in [1, h[s]]$ shares results in $h_{\tau+1}[s] = h_\tau[s] + k[s]$, where k[s] $\in \mathbb{Z}^+$.
- *Holding* $k[s] \in [1, h[s]]$ shares results in $h_{\tau+1}[s] = h_\tau[s]$.
- *Selling* $k[s] \in [1, h[s]]$ shares lead to $h_{\tau+1}[s] = h_\tau[s] - k[s]$.

Note that the trading actions at time-step τ directly impact the trading-account observation at time-step $\tau + 1$, $o_{\tau+1}$.

Rewards: We define the reward as the change in the value when an action is taken at state s_τ to arrive at new state $s_{\tau+1}$. Corresponding to each state change, we define a return r, as:

$$r(s_\tau, a_\tau, s_{\tau+1}) = (b_{\tau+1} + p_{\tau+1}^T h_{\tau+1}) - (b_\tau + p_\tau^T h_\tau) - c_\tau \quad (1)$$

where b_τ is the account balance, p_τ is a vector that represents the stock prices, h_τ denotes the stock shares in the trading account, and c_τ denotes the transaction costs incurred at time-step τ. To maximize the earned profit, we aim to design a reinforcement learning agent that maximizes the cumulative change $r(s_\tau, a_\tau, s_{\tau+1})$.

4 Proposed Approach: PROFIT

We adopt reinforcement learning to optimize profitability in quantitative trading. To this end, we introduce PROFIT, a deep reinforcement learning approach for text-based stock trading, as shown in Figure 2. For this study, we make use of a custom policy network that *hierarchically* and *attentively* learns *time-aware* representations of news and tweets to trade stocks. In practice, PROFIT's proposed policy network is generalizable across various actor-critic reinforcement learning methods that exploit neural networks as function approximators. Moreover, PROFIT is compatible with any custom policy network of the same nature that can handle textual time-series data.

4.1 Deep Reinforcement Learning

We base PROFIT on the Deep Deterministic Policy Gradient (DDPG) framework (Lillicrap et al., 2015), which bridges the gap between policy gradient (Sutton et al., 2000) and value approximation methods (Watkins and Dayan, 1992) for RL. The

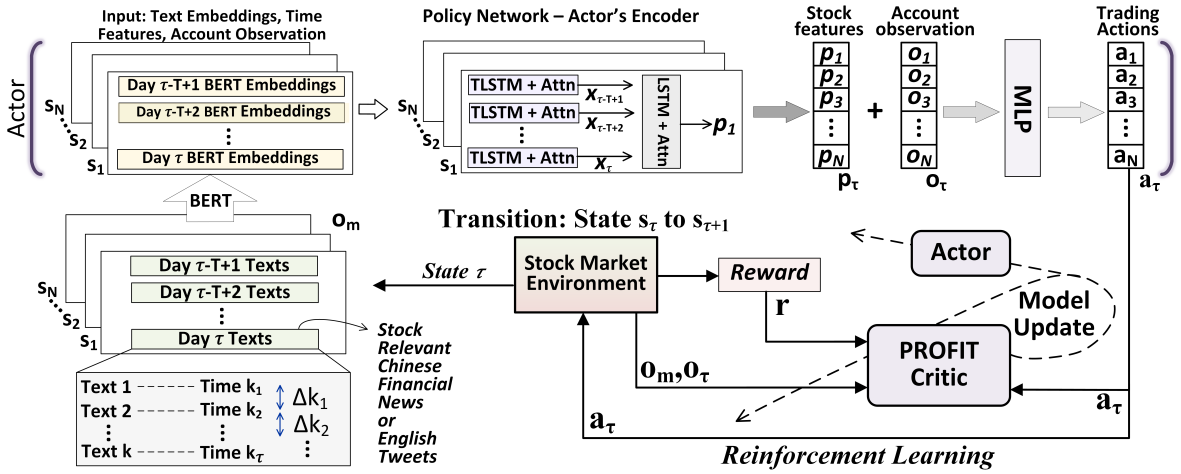

Figure 2: PROFIT: Trading policy network (top), deep reinforcement learning for stock trading (bottom).

DDPG decouples the trading *action selection* and the trading *action evaluation* processes into two separate jointly learned networks: the actor network, and the critic network. The actor-network μ, parameterized by θ, takes the observations at state s_τ as input, and outputs the trading actions a_τ. The critic-network Q, parameterized by ϕ, takes the observations at state s_τ and trading actions a_τ from the actor as input. It then outputs a scalar $\mathrm{Q}(s_\tau, a_\tau)$ to evaluate the action a_τ.

For each state s_τ, the agent performs an action a_τ, receives a reward r_τ, and reaches the next state $s_{\tau+1}$. These transitions represented as $(s_\tau, a_\tau, s_{\tau+1}, r_\tau)$ are stored in a replay buffer $\mathcal{D}$. Subsequently, a mini-batch B comprising N transitions is sampled from $\mathcal{D}$ for updating the model. For each batch B, PROFIT minimizes the following loss L with respect to ϕ to update the critic as:

$$y_\tau = r_\tau + \gamma Q^{\phi'}(s_{\tau+1}, \mu^{\theta'}(s_{\tau+1})), \quad (2)$$

$$L = \mathbb{E}[(y_\tau - Q^{\phi}(s_\tau, a_\tau))^2] \quad (3)$$

where y_τ is the updated Q-value, γ is a discount factor, θ and θ', ϕ and ϕ' are the two copy parameters of the policy μ and the value function Q, respectively. The actor is updated using the policy gradient $\nabla_\theta J$ via backpropagation through time as:

$$\nabla_\theta J = \mathbb{E}[\nabla_a Q^{\phi}(s_\tau, \mu^{\theta}(s_\tau)) \nabla_\theta \mu^{\theta}(s_\tau)] \quad (4)$$

In the above equations, θ and θ', ϕ and ϕ' are the two copy parameters of the policy μ and the value function Q, respectively. For a detailed explanation of the framework, we refer the readers to Lillicrap et al. (2015). Next, we define the trading policy network, which takes the observations at state s_τ as input to generate stock trading actions a_τ. We use the same architecture for defining the actor and the critic networks.

4.2 Trading Policy Network

To generate trading actions, we first learn representations for each stock $s \in S$ using the T-day market-information observation o_m, and the trading-account observation o_τ at the time-step τ. For this study, we derive inspiration from Hu et al. (2017); Sawhney et al. (2020, 2021) to design the policy network. However, it is important to note that PROFIT is compatible with any general deep network that is capable of handling time-series of textual data. We specifically adopt the following network as it inherently covers a breadth of components that are proved beneficial for designing language-based systems for stock trading.

First, PROFIT's policy encodes the texts t corresponding to a stock s released in a day using BERT (Devlin et al., 2019). We tokenize and truncate the input text (t) for each news item or tweet and feed it to BERT. We then aggregate the final hidden states (the final-layer transformer outputs) of the input to get the encoded representation (m, size 768) as as $m = \mathrm{BERT}(t) \in \mathbb{R}^d$, d=768. We also experiment with the $[CLS]$ token and other pooling techniques such as maximum of hidden states and concatenation of mean and maximum of hidden states but do not obtain better results.

For each stock s on a day i, a variable number (K) of tweets (t) are posted at irregular times (k). LSTMs though able to capture the *sequential con-*

text dependencies in text over time, assume inputs to be equally spaced in time. However, the intervals between release of consecutive news items or tweets can vary widely, from a few seconds to many hours, and that can have a drastic impact on their influence on the market (O'Hara, 2015). Thus, we use a time-aware LSTM (TLSTM) (Baytas et al., 2017), to capture the irregularities in the release of text, and encode them for a stock s on a day i.

All news and tweets in a day might not be equally informative, and may have *diverse influence* over a stock's trend (Barber and Odean, 2007). We use an intra-day attention mechanism (Qin et al., 2017) that allows the trading agent to emphasize texts likely to have a more substantial impact on price. The attention mechanism learns to adaptively aggregate the variable number of hidden states of the t-LSTM into an *intra-day text information vector*. We combine these representations across days in a *hierarchical* fashion using an LSTM.

We use attention again over the outputs of the LSTM to obtain a market-information vector p_τ comprising financial signals across tweets or news items released over the lookback. Lastly, we concatenate the trading-account observation o_τ at state s_τ, with the market-information vector p_τ to form an *overall stock-level representation* $z_\tau = [o_\tau, p_\tau]$.

Trading actions: We concatenate the stock-representations z_τ to form a feature vector Z across stocks for day τ. We then feed Z to a feed-forward network, followed by a $tanh$ activation function, which outputs actions a_τ to buy, hold or sell the shares of each stock $s \in S$ at the time-step τ.

5 Experimental Setup

5.1 Datasets and Stock Markets

US S&P 500[1] (Xu and Cohen, 2018): Comprises $109,915$ *English tweets* from the social media platform Twitter spanning January 2014 to December 2015, related to 88 high-trade-volume stocks from the NASDAQ Exchange forming the S&P 500 index. NASDAQ is a fairly volatile (Schwert, 2002) US exchange. The stocks are categorized into 9 industries:[2] Basic Materials, Consumer Goods, Healthcare, Services, Utilities, Conglomerates, Financial, Industrial Goods and Technology. Xu and Cohen (2018) extracted stock specific tweets using regex queries made of stock ticker symbols, for instance, $AMZN for Amazon, where $ acts as a *cashtag* on the platform Twitter).

China & Hong Kong[3] (Huang et al., 2018): Comprises $90,361$ financial *news headlines* in *Chinese*. The headlines span January 2015 to December 2015, and are originally aggregated by *Wind*[4] from major financial website like Sina[5] and Hexun.[6] The news headlines are related to 85 *China A-share* stocks from the Shanghai, Shenzhen and the Hong Kong Stock Exchanges. Huang et al. (2018) extracted news from major financial websites covering corporate news across Mainland China and Hong Kong.

Pre-processing: We pre-process English tweets using the NLTK[7] (Twitter mode), for treatment of URLs, identifiers (@) and hashtags (#). We adopt the Bert-Tokenizer for tokenization of tweets. For the English tweets, we use the pre-trained BERT-base-cased[8] model. For the Chinese news, we adopt the Chinese-BERT-base[8] model, having 12 layers and 110M parameters. We use character-based tokenization for the Chinese headlines. We collect prices from Yahoo Finance.[9] We align trading days by dropping data samples that do not possess tweets for a consecutive 7-day window, and further align the data across windows for stocks to ensure that data is available for all days in the window for the same set of stocks. We split the US S&P 500 dataset temporally based on date ranges from January 01, 2014 to July 31, 2015 for training, August 01, 2015 to September 30, 2015 for validation, and October 01, 2015 to January 01, 2016 for testing. We split the China & Hong Kong dataset temporally based on date ranges from January 01, 2015 to August 31, 2015 for training, September 01, 2015 to September 30, 2015 for validation, and October 01, 2015 to January 01, 2016 for testing all models and experiments.

5.2 PROFIT 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 (**§5.3**) for all

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

[2] https://finance.yahoo.com/industries

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

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

[5] http://finance.sina.com.cn/

[6] http://www.hexun.com/

[7] https://www.nltk.org/

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

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

models. We build the RL agent in Python programming language using PyTorch and employ OpenAI gym to implement the stock trading environment. We explore the length of the lookback period $T \in \text{range}[2, 10]$ days. Across both the datasets, we obtain that the model best performs for a week-long lookback – i.e. 7 days. We explore the hidden state dimension for both TLSTM and LSTM $d \in [32, 64, 128]$ (we achieve the best performance for: $d = 64$, both for the TLSTM and the LSTM) across both the datasets. We factor the time elapsed between the successive posting of texts at the common finest granularity available across the datasets – i.e. 1 minute intervals. We use the Xavier initialization (Glorot and Bengio, 2010) to initialize all network weights. We use an exponential learning rate scheduler (Li and Arora, 2019) with a decay rate of 0.001 and an initial learning rate of $7e-5$. For each dataset, we train PROFIT using the Adam optimizer (Kingma and Ba, 2014).

5.3 Evaluation Metrics

To assess the profitability and trading performance of all methods, we compute the **Sharpe ratio (SR)**, its variant **Sortino Ratio (StR)**, the **Cumulative Return (CR)**, and the **Maximum Drawdown (MDD)**. The Sharpe Ratio is a measure of the return of a portfolio compared to its risk (Sharpe, 1994). We calculate SR by computing the earned return R_a in excess of the risk-free return[10] R_f, defined as: $\text{SR} = \frac{E[R_a - R_f]}{std[R_a - R_f]}$. The Sortino Ratio is a variation of the Sharpe Ratio, which uses an asset's standard deviation of negative portfolio returns (downside deviation, σ_d) as: $\text{StR} = \frac{E[R_a - R_f]}{\sigma_d}$. The StR is a useful way to evaluate an investment's return for a given level of bad risk, and provides a better view of the risk-adjusted return – as positive volatility is essentially considered beneficial. The CR is the change in the investment over time and is computed using the initial (b_0) and the final (b_f) account balance as: $\text{CR} = \frac{b_f - b_0}{b_0} * 100$. The MDD measures the maximum loss from a peak r_p to a trough r_t of a portfolio, and is defined as: $\text{MDD} = \frac{r_t - r_p}{r_p} * 100$. Larger values (in magnitude) of MDD indicate higher volatility. MDD is an indicator used to assess the relative riskiness of one stock trading strategy versus another, as it focuses on capital preservation, which is a key concern for most investors. For instance, two trading strategies may have the same volatility, average outperformance, and tracking error, but their maximum drawdowns compared to the benchmark can differ drastically. Investors typically prefer the strategy with lower maximum drawdowns.

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

5.4 Practical Trading Constrains

The following assumptions and constraints reflect concerns for practical stock trading. PROFIT accounts for various elements of the trading process and the financial aspects like transaction costs, market liquidity, and risk-aversion (Yang et al., 2020).

Non-negative account balance: Ideally, the allowed trading actions should not result in a negative account balance. Based on the stock-level actions generated at time τ, the stocks are divided into sets for selling, buying, and holding, non-overlapping sets. The constraint for non-negative balance is that for any given time step τ, the sum of account balance b_τ; the money gained through selling the stocks in set S; and the money spent for acquiring the stocks in the buying set: should be positive, or at minimum zero.

Transaction costs: For each trade, various types of transaction costs such as exchange fees, execution fees, and SEC fees are incurred. Further, in practice, different brokers have different commission fees, and despite these variations, we assume our transaction costs to be 0.1% of the value of each trade (either buy or sell).

5.5 Baseline Approaches

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

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

- **W-LSTM:** LSTMs with stacked autoencoders that encode noise-free data obtained through wavelet transform of prices (Bao et al., 2017).
- **AZFinText:** Proper noun-based text representations fed to Support Vector Regression for forecasting returns (Schumaker and Chen, 2009).

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

Formulation	US S&P 500				China & Hong Kong			
	CR↑	*SR*↑	*StR*↑	*MDD*↓	*CR*↑	*SR*↑	*StR*↑	*MDD*↓
Regression	9.62 ± 2.16	0.76 ± 0.21	0.99 ± 0.40	18.98 ± 4.56	24.81 ± 11.56	1.02 ± 0.29	1.49 ± 0.37	14.34 ± 5.63
Classification	10.06 ± 2.73	0.89 ± 0.26	1.15 ± 0.51	19.06 ± 5.07	25.72 ± 13.29	1.03 ± 0.22	1.56 ± 0.43	15.88 ± 5.58
Ranking (Sawhney et al., 2021)	21.45 ± 6.78	0.95 ± 0.11	1.35 ± 0.27	16.93 ± 5.58	33.25 ± 15.12	1.19 ± 0.16	1.67 ± 0.34	10.45 ± 2.36
Reinforcement Learning	29.64 ± 8.22	1.03 ± 0.24	1.87 ± 0.65	5.01 ± 4.21	40.88 ± 13.04	1.29 ± 0.32	1.99 ± 0.61	6.78 ± 6.09

Table 1: Trading performance over different problem formulations (mean of 5 runs). All formulations use the same base architecture defined in PROFIT's policy network to model stock affecting text over the lookback period.

- **TSLDA:** Topic Sentiment Latent Dirichlet Allocation, a generative model jointly exploiting topics and sentiments in textual data (Nguyen and Shirai, 2015).
- **StockEmb:** Stock embeddings acquired using prices, and dual vector (word-level vectors and context-level vectors) 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).
- **MAN-SF (text only):** BERT based hierarchical encoder for financial text using hierarchical temporal attention (Sawhney et al., 2020).
- **Chaotic:** A Hierarchical Attention Network using GRU encoders with temporal attention applied on text within days, and the days in the lookback period (Hu et al., 2017).

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

- **R-LSTM:** Utilizes 5-day, 10-day, 20-day, and 30-day averages and closing prices of stocks to train an LSTM model (Feng et al., 2019).
- **RankNet:** A DNN that utilizes sentiment-based shock and trend scores to optimize a probabilistic ranking function (Song et al., 2017).

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

- **iRDPG:** An imitative Recurrent Deterministic Policy Gradient (RDPG) algorithm exploiting temporal stock price features, while optimizing the Sharpe Ratio as the reward (Liu et al., 2020).
- **AlphaStock:** An LSTM based network to model prices, comprising attention to model inter-stock cross relations (Wang et al., 2019).
- **S-Reward:** Inverse reinforcement learning method to model relations between sentiments and returns (Yang et al., 2018).
- **SARL:** A Deterministic Policy Gradient with augmented states, comprising stock prices and encoded news (Ye et al., 2020).

6 Results and Discussion

6.1 Stock Trading Problem Formulation

We experiment with four different formulations for neural stock trading in Table 1. For each formulation, we treat our custom policy trading network as the base architecture for modeling stock affecting textual information over the lookback period. We find that classification and regression formulations generate relatively low profits compared to others. This is likely as trades in such methods are not optimized for the overall profit as a reward. Moreover, another limitation of classification and regression approaches is that the trading strategy needs to be defined manually. Next, we find that reinforcement learning provides the best performance as it allows PROFIT to enjoy a more granular control over trading actions and learn to optimize the strategy directly for making profitable trades using text. Further, we also observe that trading under RL formulation experiences the lowest MDD, likely as the agent has more flexibility in selecting the trades, which leads to lower losses. Next, we study how different baseline stock trading networks across the four formulations perform compared to PROFIT.

6.2 Performance Comparison with Baselines

We now compare PROFIT's profitability (Sharpe Ratio) and risk in investment (Maximum Drawdown) against baseline approaches in Table 2. PROFIT generates higher risk-adjusted returns and experiences lower losses than all methods, as we show in Figure 3. We find methods that incorporate stock affecting information from textual sources generate profits higher or comparable to price-only methods. These results indicate that textual sources

	Models & Components		US S&P 500		China & Hong Kong	
			SR↑	*MDD*↓	*SR*↑	*MDD*↓
REG	W-LSTM (Bao et al., 2017)	P	0.41 ± 0.15	32.91 ± 7.91	0.49 ± 0.13	30.86 ± 10.98
	AZFinText (Schumaker and Chen, 2009)	T + P	0.40 ± 0.10	31.46 ± 5.91	0.50 ± 0.09	19.09 ± 1.56
CLF	TSLDA (Nguyen and Shirai, 2015)	T + P	0.39 ± 0.08	31.72 ± 6.71	0.51 ± 0.12	38.75 ± 15.92
	StockEmb (Du and Tanaka-Ishii, 2020)	T + P + A	0.51 ± 0.14	22.01 ± 10.87	0.74 ± 0.21	20.19 ± 9.39
	SN - HFA (Xu and Cohen, 2018)	T + P + A	0.81 ± 0.08	12.15 ± 2.01	0.93 ± 0.09	8.17 ± 1.97
	MAN-SF (Text only) (Sawhney et al., 2020)	T + A	0.80 ± 0.11	18.09 ± 7.24	1.01 ± 0.15	8.95 ± 6.19
	Chaotic (Hu et al., 2017)	T + A	0.86 ± 0.21	15.49 ± 5.38	0.95 ± 0.37	18.30 ± 6.44
RAN	R-LSTM (Feng et al., 2019)	P	0.78 ± 0.19	21.42 ± 3.21	0.96 ± 0.05	13.86 ± 4.74
	RankNet (Song et al., 2017)	T	0.87 ± 0.09	10.40 ± 2.90	0.95 ± 0.10	8.13 ± 1.14
RL	iRDPG (Liu et al., 2020)	P	0.79 ± 0.14	17.71 ± 9.56	1.03 ± 0.28	13.73 ± 5.62
	AlphaStock (Wang et al., 2019)	P + A	0.71 ± 0.24	11.54 ± 6.91	0.95 ± 0.24	9.96 ± 7.15
	S-Reward (Yang et al., 2018)	T	0.73 ± 0.16	10.46 ± 7.22	1.08 ± 0.39	13.27 ± 7.32
	SARL (Ye et al., 2020)	T + P	0.91 ± 0.13	8.38 ± 4.95	1.10 ± 0.19	16.67 ± 7.47
	PROFIT (Ours)	T + A	1.03 ± 0.24	5.01 ± 4.21	1.29 ± 0.32	6.78 ± 6.09

Table 2: Profitability comparison against baseline approaches (mean of 5 runs) (§5.5). Within Components, T = Text, P = Prices, A = Attention across modalities. Green and blue depict best and second-best results, respectively.

can augment neural stock prediction, as they potentially help capture classic financial anomalies such as the over- and under-reaction of asset prices to news (Bondt and Thaler, 1985; Corgnet et al., 2013). This observation also follows prior research that shows financial text are generally better indicators of market volatility, compared to price signals (Atkins et al., 2018). In general, we observe that ranking and reinforcement learning methods generate high returns as they are directly optimized towards profit generation. Further, reinforcement learning approaches are typically more profitable as the trading agents optimize every trading action for profit generation directly, unlike ranking, where the task is only to select profitable stocks to trade. These observations validate the premise of formulating quantitative trading as a reinforcement learning problem, compared to conventionally adopted regression and classification formulations.

Despite the 2015-16 Chinese Market Turbulence Recession[11] (Liu et al., 2016), the lower MDD of PROFIT indicates the trading agent's ability to respond to bearish markets[12], and its performance is attributable to the following reasons. Amongst competitive baselines, PROFIT's policy design differentiates it from others, as it captures the hierarchical dependencies in the news and attentively learns to emphasize crucial trading indicators during such turbulent economies. The attention mechanisms potentially account for financial phenomena such as the calendar (Jacobs and Levy, 1988) and

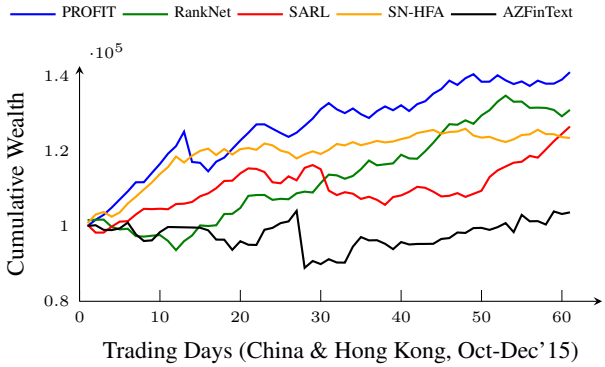

Figure 3: Capital growth (initial $\$100,000$) through PROFIT's trades compared against baseline methods.

the day-of-the-week (Halil, 2001) effects, and better distinguish noise inducing text from relevant market signals to minimize false evaluations and overreactions (De Long et al., 1989). Further, Jiao et al. (2020) show that frequent news media coverage is an indicator of a decrease in stock volatility. Through its time-aware mechanism, the agent can incorporate such frequencies and learn to trade less volatile stocks to execute low-risk and high-profit trades even in bearish market scenarios.

6.3 Parameter Analysis: Probing Sensitivity

Lookback period length T Here, we study how PROFIT's performance varies with the length of lookback period $T \in [2, 10]$ days in Figure 4. Lower performance indicates the inability of shorter lookbacks to capture stock affecting market information, as public information requires time to absorb into price movements (Luss and D'Aspremont, 2015). As we increase T, we observe a deterioration in

[11] https://www.vox.com/2015/7/8/8908765/chinas-stock-market-crash-explained

[12] Bearish markets are those that experience prolonged price declines, experience high volatility and risk on investments.

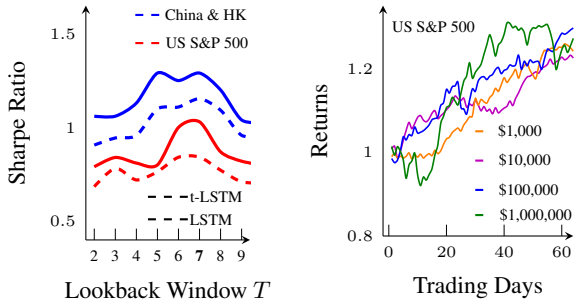

Figure 4: Sensitivity to parameters T and b_0

the trading performance. This indicates that larger lookbacks allow the inclusion of stale information from older days having relatively lower influence on prices (Bernhaedt and Miao, 2004). We observe optimal performance for mid-sized lookbacks.

Initial trading balance b_0 To further analyze PROFIT's trading performance, we simulate the cumulative returns for different initial trading amounts. Financial studies highlight that larger investments are prone to higher risk (Stout, 1995), as higher budgets allow increased risk-taking abilities. Ghysels et al. (2005) find significantly positive relations between larger risk and higher returns (risk-return tradeoff).[13] PROFIT's performance is akin to this phenomena as we observe generally high rewards even for riskier decisions taken on larger investments, as shown in Figure 4. We attribute PROFIT's versatility to its policy design that allows diverse trading choices based on resource availability. These results indicate that PROFIT holds practical applicability to investors across diverse economic milieus: from individual traders to larger firms having greater investment margins.

7 Conclusion

We propose PROFIT, a deep RL approach for quantitative trading using textual data across online news and tweets. To model the market information, PROFIT hierarchically learns temporally relevant signals from texts in a time-aware fashion, and directly optimizes trading actions towards profit generation. Through extensive analyses on English tweets and Chinese news spanning four markets, we highlight PROFIT's real-world applicability. In trading simulations on the S&P 500 and China A-shares indexes, PROFIT outperforms baselines in terms of profitability and risk in investment.

[13] https://www.investopedia.com/terms/r/riskreturntradeoff.asp

8 Ethical Considerations

There is an ethical imperative implicit in this growing influence of automation in market behavior, and it is worthy of serious study (Hurlburt et al., 2009; Cooper et al., 2020). Since financial markets are transparent (Bloomfield and O'Hara, 1999), and heavily regulated (Edwards, 1996), we discuss the ethical considerations pertaining to our work. Following (Cooper et al., 2016), we emphasize on three ethical criteria for automated trading systems and discuss PROFIT's design with respect to these criteria.

Prudent System A prudent system *"demands adherence to processes that reliably produce strategies with desirable characteristics such as minimizing risk, and generating revenue in excess of its costs over a period acceptable to its investors" (Longstreth, 1986)*. PROFIT is directly optimized towards profit-generation and minimizing investor risk by selectively investing in the less volatile stocks (**§6.2**), and generates risk-adjusted returns: Sharpe Ratio, as shown in Table 2.

Blocking Price Discovery A trading system should not block price discovery and not interfere with the ability of other market participants to add to their own information (Angel and McCabe, 2013). For example, placing an extremely large volume of orders to block competitor's messages *(Quote Stuffing)* or intentionally trading with itself to create the illusion of market activity *(Wash Trading)*. PROFIT does not block price discovery in any form.

Circumventing Price Discovery A trading system should not hide information, such as by participating in dark pools or placing hidden orders (Zhu, 2014). We evaluate PROFIT only on public data in highly regulated stock markets.

Despite these considerations, it is possible for PROFIT, just as any other automated trading system, to be exploited to hinder market fairness. We follow broad ethical guidelines to design and evaluate PROFIT, and encourage readers to follow both regulatory and ethical considerations pertaining to the stock market.

References

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

Saud Almahdi and Steve Y Yang. 2019. A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning. *Expert Systems with Applications*, 130:145–156.

Prithviraj Ammanabrolu and Mark Riedl. 2019. Playing text-adventure games with graph-based deep reinforcement learning. 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 3557–3565, Minneapolis, Minnesota. Association for Computational Linguistics.

James J Angel and Douglas McCabe. 2013. Fairness in financial markets: The case of high frequency trading. *Journal of Business Ethics*, 112(4):585–595.

Adam Atkins, Mahesan Niranjan, and Enrico Gerding. 2018. Financial news predicts stock market volatility better than close price. *The Journal of Finance and Data Science*, 4(2):120 – 137.

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):785–818.

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.

Robert Bloomfield and Maureen O'Hara. 1999. Market transparency: who wins and who loses? *The Review of Financial Studies*, 12(1):5–35.

Werner F. M. De Bondt and Richard Thaler. 1985. Does the stock market overreact? *The Journal of Finance*, 40(3):793–805.

Jeffrey A. Busse and T. Green. 2002. Market efficiency in real time. *Journal of Financial Economics*, 65(3):415 – 437.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting.

Ricky Cooper, Michael Davis, Andrew Kumiega, and Ben Van Vliet. 2020. Ethics for automated financial markets. *Handbook on Ethics in Finance*, pages 1–18.

Ricky Cooper, Michael Davis, Ben Van Vliet, et al. 2016. The mysterious ethics of high-frequency trading. *Business Ethics Quarterly*, 26(1):1–22.

Brice Corgnet, Praveen Kujal, and David Porter. 2013. Reaction to public information in markets: how much does ambiguity matter? *The Economic Journal*, 123(569):699–737.

J. BRADFORD De Long, ANDREI SHLEIFER, LAWRENCE H. SUMMERS, and ROBERT J. WALDMANN. 1989. The size and incidence of the losses from noise trading. *The Journal of Finance*, 44(3):681–696.

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.

Franklin R Edwards. 1996. *The new finance: regulation and financial stability*. American Enterprise Institute.

Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems*, 37(2):1–30.

Angelos Filos. 2019. Reinforcement learning for portfolio management. *arXiv preprint arXiv:1909.09571*.

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

Eric Ghysels, Pedro Santa-Clara, and Rossen Valkanov. 2005. There is a risk-return trade-off after all. *Journal of Financial Economics*, 76(3):509 – 548.

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.

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

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016a. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630, Berlin, Germany. Association for Computational Linguistics.

Ji He, Mari Ostendorf, Xiaodong He, Jianshu Chen, Jianfeng Gao, Lihong Li, and Li Deng. 2016b. Deep reinforcement learning with a combinatorial action space for predicting popular Reddit threads. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1838–1848, Austin, Texas. Association for Computational Linguistics.

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*.

Yuncheng Hua, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, and Tongtong Wu. 2020. Few-shot complex knowledge base question answering via meta reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5827–5837, Online. Association for Computational Linguistics.

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

George F Hurlburt, Keith W Miller, and Jeffrey M Voas. 2009. An ethical analysis of automation, risk, and the financial crises of 2008. *IT professional*, 11(1):14–19.

Tommi Jaakkola, Satinder P Singh, and Michael I Jordan. 1995. Reinforcement learning algorithm for partially observable markov decision problems. In *Advances in neural information processing systems*, pages 345–352.

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.

Zhengyao Jiang and Jinjun Liang. 2017. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 905–913. IEEE.

Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*.

Peiran Jiao, Andre Veiga, and Ansgar Walther. 2020. Social media, news media and the stock market. *Journal of Economic Behavior & Organization*, 176:63–90.

Musonda Katongo and Ritabrata Bhattacharyya. 2021. The use of deep reinforcement learning in tactical asset allocation. *Available at SSRN 3812609*.

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

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

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

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

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.

Bevis Longstreth. 1986. *Modern investment management and the prudent man rule*. Oxford University Press on Demand.

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.

Giorgio Lucarelli and Matteo Borrotti. 2019. A deep reinforcement learning approach for automated cryptocurrency trading. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 247–258. Springer.

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

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

Terry Lingze Meng and Matloob Khushi. 2019. Reinforcement learning in financial markets. *Data*, 4(3):110.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 1 (Long Papers), pages 1747–1759, New Orleans, Louisiana. 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.

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.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2137–2147, Melbourne, Australia. Association for Computational Linguistics.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn Shah. 2020. 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, Arnav Wadhwa, Agarwal Shivam, and Rajiv Ratn Shah. 2021. Fast: Financial news and tweet based time-aware network for stock trading. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*.

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).

G William Schwert. 2002. Stock volatility in the new millennium: how wacky is nasdaq? *Journal of Monetary Economics*, 49(1):3–26.

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

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.

Lynn A. Stout. 1995. Are stock markets costly casinos? disagreement, market failure, and securities regulation. *Virginia Law Review*, 81(3):611–712.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Ruey S Tsay. 2005. *Analysis of financial time series*, volume 543. John wiley & sons.

Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. 2019. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1900–1908, New York, NY, USA. Association for Computing Machinery.

Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8(3-4):279–292.

Jiawei Wu, Lei Li, and William Yang Wang. 2018a. Reinforced co-training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1252–1262, New Orleans, Louisiana. Association for Computational Linguistics.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018b. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.

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*.

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. *Available at SSRN*.

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.

Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. 2020. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Pro-*

ceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 1112–1119.

Xianda Zhou and William Yang Wang. 2018. MojiTalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1128–1137, Melbourne, Australia. Association for Computational Linguistics.

Haoxiang Zhu. 2014. Do dark pools harm price discovery? *The Review of Financial Studies*, 27(3):747–789.