Incorporating Pre-trained Model Prompting in Multimodal Stock Volume Movement Prediction

Ruibo Chen*
ruibochen@pku.edu.cn
Peking University, China

Zhiyuan Zhang*
zzy1210@pku.edu.cn
Peking University, China

Yi Liu
yliu.pku@outlook.com
Peking University, China

Ruihan Bao
ruihan.bao@mizuho-sc.com
Mizuho Securities Co., Ltd, Japan

Keiko Harimoto
keiko.harimoto@mizuho-sc.com
Mizuho Securities Co., Ltd, Japan

Xu Sun
xusun@pku.edu.cn
Peking University, China

ABSTRACT

Multimodal stock trading volume movement prediction with stock-related news is one of the fundamental problems in the financial area. Existing multimodal works that train models from scratch face the problem of lacking universal knowledge when modeling financial news. In addition, the models' ability may be limited by the lack of domain-related knowledge due to insufficient data in the datasets. To handle this issue, we propose the **Pro**mpt-based **MU**ltimodal **S**tock volum**E** prediction model (ProMUSE) to process text and time series modalities. We use pre-trained language models for better comprehension of financial news and adopt prompt learning methods to leverage their capability in universal knowledge to model textual information. Besides, simply fusing two modalities can cause harm to the unimodal representations. Thus, we propose a novel cross-modality contrastive alignment while reserving the unimodal heads beside the fusion head to mitigate this problem. Extensive experiments demonstrate that our proposed ProMUSE outperforms existing baselines. Comprehensive analyses further validate the effectiveness of our architecture compared to potential variants and learning mechanisms. Our code will be available in https://github.com/RayRuiboChen/ProMUSE.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

multimodal learning, stock movement prediction, prompt learning

1 INTRODUCTION

Stock trading volume movement prediction is one of the fundamental tasks in the financial area which has been paid much attention to [1, 4, 26], and it has various important downstream applications such as algorithmic trading [13, 39] and stock trading anomaly detection [32].

Traditional researches only use historical trading data and rely heavily on feature engineering. They employ statistical models to forecast the time series like the Autoregressive Integrated Moving Average model (ARIMA) [2, 32]. With the development of deep learning techniques, Gharehchopogh et al. [12] use linear regression for stock market trading volume prediction. More complicated models utilizing LSTM and CNN are also applied in this field [6, 30, 38]. However, these methods which only involve historical trading data suffer from the lack of basic stock information. Human traders make their decisions on multiple factors, including stock-related news. As a result, unimodal models using only historical trading data as the input may make incorrect predictions. Thus, researchers start to introduce text information like news and tweets to better model the stock movement. Typically, sentimental analysis modules are used to reflect the market sentiment [22, 31, 33, 35]. These methods usually adopt pipeline architecture and the two modalities are not integrated properly. Additional errors may get involved in the prediction of sentiment. More recent works design multimodal models to jointly process text and time series data in order to acquire a better understanding. For example, Li et al. [20] propose to use the event-driven LSTM model to leverage news data. Zou and Herremans [47] design a hybrid multimodal model using CNN and SVM as the backbone.

A significant weakness of previous works is that they tend to construct a certain architecture and train from scratch. Current high-quality financial news datasets all tend to be much smaller in size than the large-scale unlabelled corpora collected from the Internet since it is costly to write, collect and filter to get the related news. Thus, it is very difficult to train a robust large multimodal model based on them, causing inferior ability compared to pre-trained language models such as Fin-BERT [43] and ChatGPT[1]. In addition, as the topics and contents of financial news can be broad, domain knowledge and universal knowledge are both required for the textual learning process, making incorporating pre-trained language models necessary in this task.

---

*Equal contribution
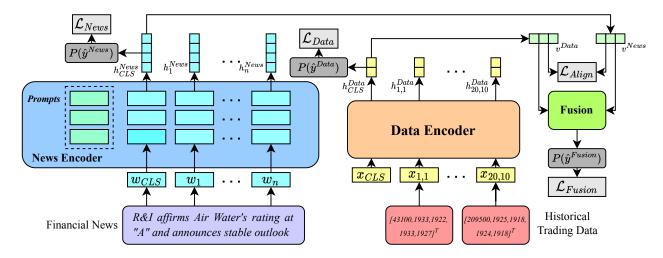[1] https://openai.com/product/chatgpt

Figure 1: An overview of our model. Financial news and historical trading data are encoded respectively. The News Encoder employs a frozen Financial-RoBERTa as the backbone and only the continuous prompts in all layers are tunable. The Data Encoder employs a pre-trained 6-layer transformer which will be fine-tuned during training. The fusion module utilizes news and trading data representations to obtain an integrated prediction. The alignment loss is designed for cross-modality contrastive alignment and prevents damage to the unimodal representations during training.

To settle the issue of lack of necessary knowledge, we propose the **Pro**mpt-based **MU**ltimodal **S**tock volum**E** movement prediction model (ProMUSE), which includes a News Encoder, a Data Encoder, and a fusion module. As illustrated in Figure 1, the News Encoder uses a pre-trained language model as a backbone, and we use prompt learning methods to efficiently exploit the knowledge within the pre-trained models. However, the direct fusion of the two encoders will severely damage the representation learning of the two modalities. Thus we propose to reserve a prediction head for each encoder to receive unimodal supervision. Specially, we add a cross-modality contrastive alignment loss to align the embedding space, alleviating the harm to their representations during the joint training. For inference, we use the algorithmic mean of the two unimodal head predictions and the multimodal fusion result through the fusion module. This makes it possible for our model to robustly generate outputs with even unimodal inputs.

We validate our proposed ProMUSE model on the TOPIX500 trading dataset and overnight financial news [21] from Reuters. Our extensive experiments demonstrate that our proposed ProMUSE outperforms unimodal methods and multimodal baselines with significant gaps. We also conduct a series of ablation studies to analyze different modules in our model. We show that fusion-only methods perform worse as they lack unimodal supervision and therefore incur damage to unimodal representations. on the other hand, ensemble-only methods lose cross-modality contrastive alignment. Our method implements multimodal fusion and alignment reserving unimodal prediction heads to mitigate the harm to representation learning, thus achieving the best results. Besides, our proposed ProMUSE can reach higher performance than methods training from scratch consistently under varying data sizes.

Our main contributions can be summarized as follows:

- We introduce pre-trained language model prompting into multimodal stock volume movement prediction task to bring domain knowledge and universal knowledge with limited datasets.
- We propose a multimodal method ProMUSE to incorporate knowledge from texts and historical trading data. Both unimodal and multimodal supervision training are used together with cross-modality contrastive alignment to alleviate the damage to the representation learning during multimodal learning.
- Experimental results show that our method significantly outperforms competitive baselines and further analyses validate the effectiveness of our proposed modules in ProMUSE.

2 METHODOLOGY

In this section, we first introduce the task formulation of multimodal stock volume movement prediction. Then we specify the architecture of our multimodal stock trading volume movement prediction model, the contrastive loss for cross-modality alignment, and present our training objectives and inference algorithms finally.

2.1 Task Formulation

The overnight stock volume movement prediction task is a binary classification problem: label 1/0 denotes that the volume goes up/down on the next trading day. The model's classification decisions are based on the overnight news and the stock historical trading data for the past 20 days, which both serve as the model's input.

Overnight News. Following [21, 46], we only adopt news headlines for our task, as they are more informative with a suitable length for processing. The overnight news can be modeled as an input sentence with n tokens $W = \{w_1, w_2, \cdots, w_n\}$. The overnight news happened after the close of the 20th day's trading market and before the opening of the next day.

Stock Historical Trading Data. The overall stock historical trading data X includes trading volumes and prices for the past 20 days and 10 time slots for each day with a granularity of 30 minutes. The trading data x of a specific time slot includes the volume x^v, high price x^h, low price x^l, open price x^o, and close price x^c. The overall stock historical trading data X can be formulated as $X = \{x_{i,j} | i \in [1, 20] \cap \mathbb{Z}, j \in [1, 10] \cap \mathbb{Z}\}$, where $x_{i,j}$ represents the trading data for the j-th time slot in the i-th day, namely $x_{i,j} = [x_{i,j}^v, x_{i,j}^h, x_{i,j}^l, x_{i,j}^o, x_{i,j}^c]^\mathrm{T} \in \mathbb{R}^5$.

Our goal is to predict the volume movement of the first time slot for the 21st day $x_{21,1}^v$, which is the first 30 minutes after the opening of the market. Following Zhao et al. [46], we define the movement here as a comparison between $x_{21,1}^v$ and the average volume in the same time slot for the past 20 days, so the final prediction target label y can be formulated as:

$$y = \mathbb{I}\left(x_{21,1}^v > \bar{v}\right) = \begin{cases} 1, & \text{if} \quad x_{21,1}^v > \bar{v} \\ 0, & \text{if} \quad x_{21,1}^v \le \bar{v} \end{cases}, \tag{1}$$

where $\bar{v}$ denotes the average volume in the first slot for the past 20 days and is defined as: $\bar{v} = \frac{1}{20} \sum_{i=1}^{20} x_{i,1}^v$.

Then our dataset can be denoted as $\mathcal{D} = \{(X^{(i)}, W^{(i)}, y^{(i)})\}_{i=1}^N$. We adopt the accuracy (ACC) metric to evaluate the performance:

$$\text{ACC} = 100\% \times \sum_{i=1}^N \mathbb{I}\left(\arg\max_y P(y|X^{(i)}, W^{(i)}) = y^{(i)}\right), \tag{2}$$

where X denotes the input historical trading data, W denotes the stock-related news, y denotes the target label, $P(y|X^{(i)}, W^{(i)})$ denotes the predicted probabilities, and $\arg\max_y P(y|X^{(i)}, W^{(i)})$ denotes the predicted label.

2.2 Model Architecture Overview

Our proposed ProMUSE mainly includes two large-scale pre-trained modules to process the input: the News Encoder for the text modality, and the Data Encoder for the time series modality. The two encoders can effectively capture the features from the input and transform them into corresponding vectors. They also use their prediction head to produce unimodal losses and prediction to receive unimodal supervision, which will be introduced in section 2.3 and section 2.4. In section 2.5, we further utilize a fusion module for generating multimodal losses and outputs. The cross-modality contrastive alignment described in section 2.6 is designed to help improve unimodal representations during multimodal learning. Finally, we use the weighted sum of the two unimodal losses, multimodal loss, and alignment loss for the training objective in section 2.7. During inference in section 2.8, unimodal and multimodal predictions are combined. An overview of the model architecture is shown in Figure 1.

2.3 News Encoder

The News Encoder is Financial-RoBERTa[2], a 24-layer RoBERTa [27] model pre-trained on financial text, such as financial statements, news, and earnings announcements. To avoid overfitting on the

[2]https://huggingface.co/soleimanian/financial-roberta-large-sentiment

limited news data as well as to accelerate the training procedure, we adopt prompt learning methods and choose P-Tuning v2 [25] as it reaches the best performance in our preliminary experiments. This method inserts soft prompts in each layer of the Transformer-based models. We set the prompt length to 20 and enable the reparameterization. In our experiments, only the prompts are tunable, while all other parameters in the language model are frozen.

Given the input news W including n tokens, we insert a CLS token at the beginning of the sentence: $\{w_\text{CLS}, w_1, w_2, \cdots, w_n\}$. Then we utilize the Financial-RoBERTa and P-Tuning v2 method to transform it into a series of vectors $\{h_\text{CLS}^\text{News}, h_1^\text{News}, h_2^\text{News}, ..., h_n^\text{News}\}$:

$$\{h_\text{CLS}^\text{News}, \cdots, h_n^\text{News}\} = \text{News-Encoder}\left(\{w_\text{CLS}, \cdots, w_n\}\right), \tag{3}$$

where $h_i^\text{News} \in \mathbb{R}^{d_\text{News}}$, d_News is the hidden size of the language model, and $d_\text{News} = 1024$ in our settings.

We further obtain a unimodal prediction $P(\hat{y}^\text{News}|h_\text{CLS}^\text{News})$ according to h_CLS^News by a linear classification head Linear-Head$^\text{News}$ to predict the scores of each categories, Linear-Head$^\text{News}\left(h_\text{CLS}^\text{News}\right) \in \mathbb{R}^2$:

$$P(\hat{y}^\text{News}|h_\text{CLS}^\text{News}) = \text{Softmax}\left(\text{Linear-Head}^\text{News}\left(h_\text{CLS}^\text{News}\right)\right). \tag{4}$$

The unimodal loss for the News Encoder head is the Cross-entropy loss:

$$\mathcal{L}_\text{News} = -\log P(\hat{y}^\text{News} = y|h_\text{CLS}^\text{News}), \tag{5}$$

2.4 Data Encoder

In order to acquire high-quality representations for historical trading data, we pre-train a 6-layer Transformer [40] model. Recall that historical trading data $X = \{x_{i,j} | i \in [1, 20] \cap \mathbb{Z}, j \in [1, 10] \cap \mathbb{Z}\}$, $x_{i,j} \in \mathbb{R}^5$, the input time series can therefore be formulated as $[x_\text{CLS}; x_{1,1}; x_{1,2}; x_{1,3}; \cdots; x_{2,1}; \cdots; x_{20,10}] \in \mathbb{R}^{201 \times 5}$ in the time order, where the time slot number is $1 + 20 \times 10 = 201$. The Data Encoder transforms the input series into a hidden vector series $[h_\text{CLS}^\text{Data}; h_{1,1}^\text{Data}; h_{1,2}^\text{Data}; h_{1,3}^\text{Data}; \cdots; h_{2,1}^\text{Data}; \cdots; h_{20,10}^\text{Data}] \in \mathbb{R}^{201 \times d_\text{Data}}$:

$$[h_\text{CLS}^\text{Data}; \cdots; h_{20,10}^\text{Data}] = \text{Data-Encoder}([x_\text{CLS}; \cdots; x_{20,10}]), \tag{6}$$

where $h_{i,j}^\text{Data} \in \mathbb{R}^{d_\text{Data}}$. Here d_Data is the hidden size of the Transformer, and we set $d_\text{Data} = 200$.

During the pre-training phase, we use a linear head after h_CLS^Data to directly predict $x_{21,1}^v$, the volume for the first time slot in the 21st day. We adopt Mean Square Error (MSE) loss for optimization:

$$\mathcal{L}_\text{MSE} = (\text{Linear-Pre}(h_\text{CLS}^\text{Data}) - x_{21,1}^v)^2. \tag{7}$$

For multimodal stock volume movement prediction, similar to the News Encoder, we get the unimodal prediction $P(\hat{y}^\text{Data}|h_\text{CLS}^\text{Data})$ and unimodal loss $\mathcal{L}_\text{Data}$ as follows:

$$P(\hat{y}^\text{Data}|h_\text{CLS}^\text{Data}) = \text{Softmax}\left(\text{Linear-Head}^\text{Data}\left(h_\text{CLS}^\text{Data}\right)\right), \tag{8}$$

$$\mathcal{L}_\text{Data} = -\log P(\hat{y}^\text{Data} = y|h_\text{CLS}^\text{Data}), \tag{9}$$

where Linear-Head$^\text{Data}$ is the linear classification head for Data Encoder and the Cross-entropy loss is utilized.

We continue to finetune all the Transformer parameters during multimodal training. Prompt methods adopted in News Encoder

are not reserved here because the 6-layer Transformer (1.7M parameters) is much smaller in size than the RoBERTa model (335M parameters).

2.5 Fusion of News Encoder and Data Encoder

News Encoder and Data Encoder provide us with the text feature $h_{\text{CLS}}^{\text{News}}$ and the time series feature $h_{\text{CLS}}^{\text{Data}}$ respectively. We build our multimodal fusion block on top of those features. We first linearly project $h_{\text{CLS}}^{\text{News}}, h_{\text{CLS}}^{\text{Data}}$ into $v^{\text{News}}, v^{\text{Data}}$ which lie in a common embedding space $\mathbb{R}^{d_{\text{Align}}}$:

$$v^{\text{News}}, v^{\text{Data}} = \text{Linear}^{\text{News}}(h_{\text{CLS}}^{\text{News}}), \text{Linear}^{\text{Data}}(h_{\text{CLS}}^{\text{Data}}). \quad (10)$$

$v^{\text{News}}, v^{\text{Data}}$ receive the supervision signal from the contrastive alignment, which will be discussed in the next section, to learn multimodal representations.

The fusion prediction $\hat{y}^{\text{Fusion}}$ and fusion loss $\mathcal{L}_{\text{Fusion}}$ is generated from $v^{\text{News}}, v^{\text{Data}}$ with a fusion process $\text{Fusion}(v^{\text{News}}, v^{\text{Data}})$. Similarly, the cross-entropy loss is also adopted:

$$P(\hat{y}^{\text{Fusion}}|v^{\text{News}}, v^{\text{Data}}) = \text{Softmax}\left(\text{Fusion}(v^{\text{News}}, v^{\text{Data}})\right), \quad (11)$$

$$\mathcal{L}_{\text{Fusion}} = -\log P(\hat{y}^{\text{Fusion}}|v^{\text{News}}, v^{\text{Data}}). \quad (12)$$

We have implemented several fusion methods in our experiments and we find that the linear fusion function achieves the best result, which is shown in our analysis. The linear fusion function is designed as:

$$\text{Fusion}_{\text{Linear}}(v^{\text{News}}, v^{\text{Data}}) = W^{\text{News}}v^{\text{News}} + W^{\text{Data}}v^{\text{Data}} + b. \quad (13)$$

where $W^{\text{News}}, W^{\text{Data}} \in \mathbb{R}^{2 \times d_{\text{Align}}}, b \in \mathbb{R}^2$.

2.6 Cross-Modality Contrastive Alignment

Contrastive learning is a widely used technique in the multi-modal area. Methods like CLIP [36] and ALPRO [18] use cross-modality contrastive losses to align representations in different embedding spaces. In our model, we implement the cross-modality contrastive alignment between overnight news and historical trading data with the alignment loss.

Only matched pairs in a batch are considered to be positive pairs. Given a batch of pair $V^{\text{News}} = [v_1^{\text{News}}, v_2^{\text{News}}, \cdots, v_B^{\text{News}}], V^{\text{Data}} = [v_1^{\text{Data}}, v_2^{\text{Data}}, \cdots, v_B^{\text{Data}}] \in \mathbb{R}^{B \times d_{\text{Align}}}$ and B represents the batch size, we define the similarity using dot-product following [15, 18, 36]:

$$\text{Sim}(i, j) = v_i^{\text{News}} \cdot v_j^{\text{Data}}, \quad (14)$$

and the two symmetric News-to-Data (N2D), Data-to-News (D2N) alignment losses are subsequently defined as:

$$\mathcal{L}_{\text{Align}}^{\text{N2D}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(\text{Sim}(i,i)/\tau\right)}{\sum_{j=1}^{B}\exp\left(\text{Sim}(i,j)/\tau\right)}, \quad (15)$$

$$\mathcal{L}_{\text{Align}}^{\text{D2N}} = -\frac{1}{B}\sum_{j=1}^{B}\log\frac{\exp\left(\text{Sim}(j,j)/\tau\right)}{\sum_{i=1}^{B}\exp\left(\text{Sim}(i,j)/\tau\right)}. \quad (16)$$

Here τ is the temperature parameter. The total cross-modality contrastive alignment loss is:

$$\mathcal{L}_{\text{Align}} = \mathcal{L}_{\text{Align}}^{\text{N2D}} + \mathcal{L}_{\text{Align}}^{\text{D2N}}. \quad (17)$$

Note that previous works use the cross-modality contrastive loss to boost performance on retrieval tasks such as image classifications, and some even report harm on other tasks [18].

However, in our model, $\mathcal{L}_{\text{Align}}$ is designed to improve representation learning in different modalities, and the defined similarity Sim is not used for inference.

In addition, embedding spaces of text and historical data can be significantly different because they are not so strongly connected as typical settings in image-text or video-text scenarios. Deep connections between text-data modalities can cause models to learn incorrect relations and fall into the trap of overfitting as shown in our analysis experiments. We find that as the relatively simple cross-modality contrastive alignment loss is applied to the output of the encoders, it encourages multimodal fusion and alignment and does not harm the structures of the large models.

2.7 Training Objectives

We combine the aforementioned four different losses for our final training objectives. Two losses $\mathcal{L}_{\text{News}}, \mathcal{L}_{\text{Data}}$ are derived from unimodal encoders solely based on $h_{\text{CLS}}^{\text{News}}, h_{\text{CLS}}^{\text{Data}}$ respectively. Moreover, we leverage the fusion of $v^{\text{News}}, v^{\text{Data}}$ to construct a multimodal prediction for the target labels' distribution. The final training objective is a weighted sum:

$$\mathcal{L} = \lambda_{\text{N}}\mathcal{L}_{\text{News}} + \lambda_{\text{D}}\mathcal{L}_{\text{Data}} + \lambda_{\text{F}}\mathcal{L}_{\text{Fusion}} + \lambda_{\text{A}}\mathcal{L}_{\text{Align}}. \quad (18)$$

2.8 Inference

Our prediction results also consist of three elements as mentioned before. We use the equally weighted average to produce the ensembled predicted probabilities:

$$P(\hat{y}) = \frac{P\left(\hat{y}^{\text{News}}\right) + P\left(\hat{y}^{\text{Data}}\right) + P\left(\hat{y}^{\text{Fusion}}\right)}{3}, \quad (19)$$

and the final predicted label is generated by:

$$\hat{y} = \arg\max_{y} P(y|X, W), \quad (20)$$

where X denotes the input historical trading data and W denotes the stock-related financial news.

Note that our methods can still be functional if one of the two modal inputs is missing. This can be done by simply disabling the corresponding encoder, making our model more robust toward different input situations.

3 EXPERIMENTS

In this section, we first introduce the datasets, then we describe the baseline algorithms, detailed settings, and the experimental results.

3.1 Datasets and Data Processing

The historical trading data is extracted from TOPIX500, which is comprised of the 500 most liquid and highly market-capitalized stocks in Tokyo Stock. We split the dataset chronologically to avoid information leakage. The data from Jan. 1st, 2013 to Dec. 31st, 2017

Table 1: Dataset statistics for stock movement prediction.

Split	Train	Dev	Test
Samples	8,483	687	938

Table 2: Dataset statistics for Data Encoder pre-training.

Split	Train	Dev	Test
Samples	74,950	2,214	4,072

is used as the training set, and Jan. 1st, 2018 to Apr. 30th, 2018 as the development set, May 1st, 2018 to Sept. 30th, 2018 as the test set. During data processing, we drop the data point where there is a missing entry. The overnight news is collected from Reuters Financial News[3]. Following [5, 21, 46], the data is filtered with RIC labels provided by Reuters, which are the possible stocks that may be influenced by the news.

In this paper, we only select the data in which both the overnight news and the historical trading data are available for stock movement prediction. In addition, we filter the data where the volume movement is not significant enough[46], alleviating the effect of randomness and minor, irrelevant news:

$$\sigma^v = \sqrt{\frac{1}{20} \sum_{i=1}^{20} \left(x_{i,1}^v - \bar{v} \right)^2}, \quad s^v = \frac{x_{21,1}^v - \bar{v}}{\sigma^v}. \quad (21)$$

If $|s^v| \leq 0.5$, we consider its volume movement insignificant and remove it from the dataset. The statistical information for the final dataset can be found in Table 1.

For pre-training our Data Encoder, we use the same split as the stock movement prediction dataset. The dataset statistics are demonstrated in Table 2.

3.2 Baselines

In this section, we introduce the algorithms of the baseline models, including traditional statistical methods, unimodal models, fusion methods, and ensemble methods.

3.2.1 Statistical Methods.
- **Random:** Randomly predict the label $\hat{y} \sim B(1, 1/2)$.
- **Exponential Moving Average (EMA):** In this task, the EMA series are defined as:

$$\text{EMA}_n = \frac{1}{21} \left(2\text{EMA}_{n-1} + 19 x_{n,1}^v \right), \quad (22)$$

where EMA_1 is initialized as $x_{1,1}^v$, and we use EMA_{20} as a prediction of $x_{21,1}^v$. The prediction is $\hat{y} = \mathbb{I}(\text{EMA}_{20} > \bar{v})$.

3.2.2 Models Training from Scratch.
We try to train a six-layer Transformer from scratch with the hidden size set to 200 to replace the pre-trained Financial-RoBERTa in the News Encoder. With regard to the Data Encoder, we try multiple structures including linear, one-layer LSTM [14], and a six-layer Transformer. The detailed structure is similar to Zhang et al. [45]. We also explore different fusion or ensembling paradigms for model training from scratch.

---

[3]https://github.com/liweitj47/overnight-stock-movement-prediction

3.2.3 Unimodal Models.
For unimodal models, we use only one of the encoders. As a result, the final training objective and predicted probability distribution degrades into only $\mathcal{L}_{\text{News}}$, $P\left(\hat{y}^{\text{News}}\right)$ or $\mathcal{L}_{\text{Data}}$, $P\left(\hat{y}^{\text{Data}}\right)$.

3.2.4 Fusion Methods.
- **Attention:** Using attention mechanism to produce the weights for fusion. $w_1, w_2 \in \mathbb{R}^{d_{\text{Align}}}$ is derived linearly from v^{News} and v^{Data}:

$$w_1, w_2 = \text{Softmax}\left(\text{Linear}([v^{\text{News}}, v^{\text{Data}}])\right), \quad (23)$$

and the fusion function is:

$$\text{Fusion}_{\text{Attention}} = \text{Linear}\left(w_1 \odot v^{\text{News}} + w_2 \odot v^{\text{Data}}\right). \quad (24)$$

Here $\odot$ represents element-wise multiplication.
- **Transformer:** Stack a multimodal Transformer encoder on top of both encoders. It takes their last-layer hidden states as input after transforming to the same dimension and uses a linear classification head for prediction.

3.2.5 Ensemble Methods.
In this section, we introduce the ensemble methods based on the unimodal models' predictions, $P(\hat{y}^{\text{News}})$ and $P(\hat{y}^{\text{Data}})$.

- **Learnable Weights:** We set $w \in \mathbb{R}$ as a learnable parameter, and initialize it to 1. The formulation is $P(\hat{y}^{\text{Learnable}}) = \text{Sigmoid}(w)P(\hat{y}^{\text{News}}) + (1 - \text{Sigmoid}(w))P(\hat{y}^{\text{Data}})$.
- **Predicted Weights:** Use a linear head to predict the weights between two modalities:

$$w_1, w_2 = \text{Softmax}(\text{Linear}([h_{\text{CLS}}^{\text{News}}, h_{\text{CLS}}^{\text{Data}}])), \quad (25)$$

and then $P(\hat{y}^{\text{Predicted}}) = w_1 P(\hat{y}^{\text{News}}) + w_2 P(\hat{y}^{\text{Data}})$.
- **Normalized:** Normalize the unimodal results to standardized Gaussian distribution: $P(\hat{y}^{\text{Norm}}) = \text{Norm}(P(\hat{y}^{\text{News}})) + \text{Norm}(P(\hat{y}^{\text{Data}})) + 0.5$, where $\text{Norm}(p) = (p - \mu)/\sigma$ and hyperparameters μ and σ for corresponding probabilities $P(\hat{y}^{\text{News}})$ or $P(\hat{y}^{\text{Data}})$ are calculated on the development set.

3.3 Settings and Hyperparameters

We train every model in our experiments for 40 epochs and report the test accuracy as the result. The checkpoints with the best accuracy on the development set are selected for the report. We repeat every experiment 4 times and report the average results.

In our main experiments, we adopt the AdamW optimizer [28], using the learning rate as 1e-5 and the weight decay factor as 1e-3. The batch size B is chosen as 32. The weights for each loss are set as $\lambda_N = \lambda_D = \lambda_F = 1$ and $\lambda_{\text{Align}} = 0.1$. d_{Align} is set to 200. We use the log value of the stock historical trading data X as the input of the Data Encoder.

For the pre-training of our Data Encoder, we run 100 epochs and save the checkpoint with the lowest development loss for the main experiments. We also use the AdamW optimizer with the learning rate as 1e-5 and the weight decay factor as 1e-3. Additional Exponential decay is used with a factor of 0.95. Pre-training batch size is set for 64.

Table 3: Main results. Existing methods can mainly be classified into statistical, unimodal, fusion-only, and ensemble-only methods, while our proposed ProMUSE achieves state-of-the-art performance.

	Method	News Encoder	Data Encoder	ACC
Statistical	Random	-	-	50.00
	20-day EMA	-	-	59.38
Unimodal (News)	Training from scratch	Transformer	-	58.26
	Zero-shot	Financial-RoBERTa	-	48.82
	Fine-tuning	Financial-RoBERTa	-	62.19
	Prompt	Financial-RoBERTa	-	**63.75**
Unimodal (Data)	Training from scratch	-	Linear	58.93
		-	LSTM	54.78
		-	Transformer	66.07
	Zero-shot	-	Pre-trained Transformer	64.71
	Fine-tuning	-	Pre-trained Transformer	**68.44**
Fusion-Only	Training from scratch	Transformer	Linear	61.38
		Transformer	LSTM	57.70
		Transformer	Transformer	60.16
	Prompt	Financial-RoBERTa	Pre-trained Transformer	**72.47**
Ensemble-Only	Training from scratch	Transformer	Linear	58.48
		Transformer	LSTM	57.92
		Transformer	Transformer	60.60
		Transformer	Linear+LSTM+Transformer	61.94
	Prompt (Average)	Financial-RoBERTa	Pre-trained Transformer	**72.28**
ProMUSE	**Prompt**	Financial-RoBERTa	Pre-trained Transformer	**73.56**[★]

3.4 Experimental Results

Our main experimental results can be found in Table 3.

3.4.1 Training from Scratch. We adopt a six-layer Transformer as an alternative to the Financial-RoBERTa model in the News Encoder. In the case of the Data Encoder, we explore linear, LSTM, and Transformer models as substitutes for the pre-trained Transformer. The outcomes reveal that the available data volume is insufficient for training a News Encoder from scratch, while our method successfully utilizes the knowledge of the pre-trained Financial RoBERTa. Furthermore, the linear and LSTM structures yield inferior performance compared to Transformer. The benefits of employing a pre-trained Transformer model are also evident.

3.4.2 Unimodal Models. Unimodal models perform significantly worse than multimodal methods, as only the information of a single modality is fed to them, and both modalities are important in this task. Besides, we find that prompting the Financial-RoBERTa can more effectively leverage its knowledge and performs better than fine-tuning. Moreover, modeling historical trading data is easier than financial news, thus data-only methods can achieve better performance than new-only models.

3.4.3 Fusion-Only and Ensemble-Only Methods. Direct multimodal fusion or ensemble can achieve improvement compared with unimodal methods. However, fusion-only methods without unimodal supervision may inflict damage on unimodal representations, and ensemble-only methods lack multimodal connections. Our method settles these problems by constructing fusion and cross-modality contrastive alignment as well as retaining unimodal predictions to help representation learning.

3.4.4 Effectiveness of ProMUSE. As Table 3 shows, ProMUSE can achieve the best performance. We successfully surpass the widely-used traditional EMA baseline in the financial area and exhibit our advantages against the models training from scratch, unimodal models, fusion-only methods, and ensemble-only methods.

4 ANALYSIS

In this section, we first conduct the ablation study to prove the effectiveness of each module in our model. Then discuss the more detailed parts, the exploration of the tuning paradigm, fusion models, ensemble algorithms, and prompt learning methods. Finally, we present that our method helps the representation learning process.

4.1 Ablation Study

4.1.1 Effectiveness of Different Losses and Prediction Heads. We conduct various ablation experiments to test the effectiveness of each component in our proposed ProMUSE model shown in Table 4.

The results prove that the combination of fusion, ensemble, and cross-modality contrastive alignment reaches the best performance.

Table 4: Ablation study results.

Method	$\mathcal{L}^{\text{News}}$	$\mathcal{L}^{\text{Data}}$	$\mathcal{L}^{\text{Fusion}}$	$\mathcal{L}^{\text{Align}}$	$P(\hat{y}^{\text{News}})$	$P(\hat{y}^{\text{Data}})$	$P(\hat{y}^{\text{Fusion}})$	**ACC**
Fusion-Only	✓	✓	✓	✗	✗	✗	✓	72.47
Ensemble-Only	✓	✓	✗	✗	✓	✓	✗	72.28
w/o News Head	✗	✓	✓	✓	✗	✓	✓	72.15
w/o Data Head	✓	✗	✓	✓	✓	✗	✓	69.91
w/o News/Data	✗	✗	✓	✓	✗	✗	✓	70.55
w/o Fusion	✓	✓	✗	✓	✓	✓	✗	67.80
w/o Alignment	✓	✓	✓	✗	✓	✓	✓	70.52
w/o Ensemble	✓	✓	✓	✓	✗	✗	✓	73.35
ProMUSE	✓	✓	✓	✓	✓	✓	✓	**73.56**

Table 5: Results of different fusion models. The models below are fusion-only models without unimodal supervision. The linear fusion outperforms other sophisticated fusions.

Fusion Model	ACC
Attention	67.00
One-layer Transformer	65.43
Six-layer Transformer	69.11
Fusing data into the last layer in News Encoder	63.81
Fusing data to every layer in News Encoder	63.86
Linear	**72.47**

The ensemble-only method can cause degradation where accuracy falls from 73.56 to 72.28 as cross-modality interaction is lost. The fusion-only setting also triggers a loss in accuracy to 72.47 due to a heavier influence on unimodal representations without $\mathcal{L}^{\text{Align}}$. Dropping unimodal heads will severely damage the performance as both modalities are indispensable. The removal of the fusion or alignment process also breaks the connections between modalities.

Note that during the inference stage, predicting $P(\hat{y}^{\text{Data}})$ and $P(\hat{y}^{\text{News}})$ does not need much extra computation as the calculation of $P(\hat{y}^{\text{Fusion}})$ also needs the calculation of two heads. Predicting them for ensemble can get an improvement from 73.35 to 73.56.

4.1.2 Effectiveness of Fusion Models. In this section, we further explore different fusion models in Table 5. The linear fusion method which is adopted in our proposal outperforms all other variants. Here Attention is described in equation 24, One-layer/Six-layer Transformer is stacked on the two encoders as discussed in the baseline section. Fusing data into the News Encoder methods use $h_{\text{CLS}}^{\text{Data}}$ to generate the continuous prompts for the News Encoder.

The reason is that the two encoders are pre-trained and will not easily overfit into the knowledge of the small dataset, while the complicated fusion methods such as introducing another Transformer model are trained from scratch, which create additional parameters and can lead to incorrect connections and overfitting.

4.1.3 Effectiveness of Ensemble Algorithms. We test the effectiveness of different ensemble algorithms in Figure 2. In our experiments on ensemble-only models, we find that an average of news prediction and data prediction perform best. In accordance with
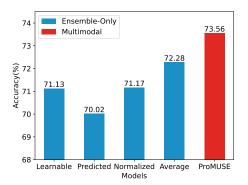

Figure 2: Results of different ensemble algorithms. The average of the two unimodal outputs exceeds other algorithms.

Table 6: Results of different prompt paradigms. P-Tuning v2 outperforms hard prompt, soft prompt, and fine-tuning.

Model	Tuning Method	ACC
News Encoder	Hard Prompt	48.82
	Soft Prompt	56.08
	Fine-tuning	62.19
	P-Tuning v2	**63.75**
Data Encoder	Zero-Shot	64.71
	Fine-tuning	**68.44**
ProMUSE	Fine-tuning	66.79
	Prompt	**73.56***

our previous analysis of fusion models, complicated ensemble algorithms also tend to deteriorate the performance, because new parameters involved in ensembling are estimated on a small dataset, which are difficult to generalize and may cause overfitting. Therefore we choose to calculate the algorithmic mean of $P(\hat{y}^{\text{News}})$, $P(\hat{y}^{\text{Data}})$ and $P(\hat{y}^{\text{Fusion}})$ for inference in our model.

4.2 Why We Utilize P-Tuning v2 Paradigm

In search of the proper paradigms for our News Encoder and Data Encoder, we conduct the experiments in Table 6. The template we use for the hard prompt in our experiments is *News: $\{w_1, w_2, \cdots, w_n\}$. The volume will go up/down.* Soft prompt here represents implementing continuous prompts in the embedding layer, which are similar to Prefix-Tuning [23] and P-Tuning [24].

We find that P-Tuning v2 best suits the News Encoder and achieve an accuracy of 63.75, which inserts continuous prompts in every layer due to more trainable parameters and deeper layers. Fine-tuning the Data Encoder can get great improvement against zero-shot setting, but fine-tuning the large Financial-RoBERTa model with limited data can easily get into the overfitting problem and cause sub-optimal, with the accuracy dropping to 66.79.

4.3 ProMUSE Works under Lower Resources

We further analyze the effect of our prompt-based multimodal model by varying the size of the training data. The development
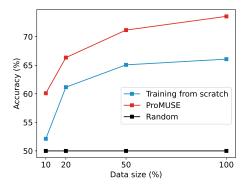
Figure 3: Results of varying data size. ProMUSE outperforms methods training from scratch under lower resources.

Table 7: Analysis of the damage of multimodal learning to unimodal representations. Existing methods corrupt unimodal representations, while ProMUSE can mitigate the damage of multimodal learning to unimodal representations.

Model	News-Only	Data-Only	Multi-Modal
News Encoder	**63.75**	-	63.75
Data Encoder	-	**68.44**	68.44
w/o Alignment	58.32	61.09	70.52
w/o Fusion	62.60	63.65	67.80
ProMUSE	62.69	67.38	**73.56**

set and test set are kept unchanged, and we train all models for 80 epochs to compensate for the reduction of data volume.

Here we compare the performance of ProMUSE with a six-layer transformer which is trained from scratch and only receives historical trading data input because it performs best in our previous experiments. We drop News Encoder as the data volume continues to shrink and a high-quality encoder cannot be obtained under these circumstances. The results are shown in Figure 3. We find that ProMUSE can achieve significant improvement over training from scratch in all settings, proving that exploiting the universal knowledge of pre-trained language models through prompt learning is essential in this task.

4.4 ProMUSE Helps Representation Learning

In this section, we discuss why our proposal can achieve improvement. As Table 7 shows, the introduction of multi-modal learning can get general gains compared to unimodal models. However, simultaneously training the two modalities can harm the representation learning of the individual encoders, as we see a decrease in accuracy for news-only or data-only input scenarios.

Our model uses the cross-modality contrastive alignment and multi-modal fusion to better alleviate the damage caused in the multi-modal learning process, providing a strong constraint and regularization for the unimodal encoders to best avoid degradation.

5 RELATED WORK

5.1 Pre-trained Language Models

Since Tranformer [40] shows great success in the natural language processing (NLP) area, various pre-trained language models are proposed and achieve state-of-the-art performance in numerous NLP tasks. Models such as and GPT-3 [3] exploit Transformer decoder structure to construct unidirectional autoregressive language models. Bidirectional BERT-like models [11, 16, 27] are basically based on Transformer encoders. T5 [37], BART [17], and Flan-T5 [10] choose to adopt the encoder-decoder framework. They are trained on corresponding pre-training tasks and large unlabelled corpora, and models become increasingly large in size [9, 44] to pursue better performance. Recently, Reinforcement Learning from Human Feedback techniques are applied to ChatGPT and GPT-4 [34], which gain outstanding performance.

5.2 Multimodal Learning

Existing multimodal learning methods mainly focus on image-text and video-text tasks. UNITER [8] learns joint contextualized representations for both text and image through pretraining. ViL-BERT [29] extends the BERT model by the co-attentional Transformer layers for learning task-agnostic representations of image content and natural language. ALIGN [15] and CLIP [36] construct the dual-encoder architecture to align visual and textual representations using image-text contrastive learning, and the cross-modality contrastive loss has become an important component in many models. BLIP-2 [19] uses a lightweight querying Transformer to learn from frozen image encoders and large language models.

5.3 Stock Movement Prediction

Stock movement prediction is a key research direction in the finance area. Xu and Cohen [42] present a deep generative model to jointly learn from tweet text and price signals, and use recurrent latent variables to process stochasticity. Li et al. [21] design an LSTM-RGCN model for learning overnight news and the correlation between stocks. Chen et al. [7] set up a dual-process meta-learning method to mine general patterns and stock-specific knowledge. Xie et al. [41] analyze the zero-shot ability of ChatGPT in multimodal stock movement prediction.

6 CONCLUSION

In this paper, we present ProMUSE, a prompt-based multimodal stock volume movement prediction model, to fully exploit the textual information in financial news and the potential of pre-trained language models with limited data. We use the News Encoder and Data Encoder to process the overnight financial news and historical trading data respectively, and use a fusion model to generate multimodal output. Unimodal supervision, multimodal supervision, and cross-modality contrastive alignment are used for training, while unimodal and multimodal predictions constitute the final inference result. Extensive experiments show that our method significantly outperforms various baselines. Comprehensive analysis testifies to the effectiveness of different modules. Moreover, ProMUSE can help mitigate the harm to representation learning during joint training of textual and time series modalities.

REFERENCES

[1] Bipin B Ajinkya and Prem C Jain. 1989. The behavior of daily stock market trading volume. *Journal of accounting and economics* 11, 4 (1989), 331–359.

[2] George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association* 65, 332 (1970), 1509–1526.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Álvaro Cartea and Sebastian Jaimungal. 2016. A closed-form execution strategy to target volume weighted average price. *SIAM Journal on Financial Mathematics* 7, 1 (2016), 760–785.

[5] Deli Chen, Yanyan Zou, Keiko Harimoto, Ruihan Bao, Xuancheng Ren, and Xu Sun. 2019. Incorporating fine-grained events in stock movement prediction. *arXiv preprint arXiv:1910.05078* (2019).

[6] Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)*. IEEE, 2823–2824.

[7] Ruibo Chen, Wei Li, Zhiyuan Zhang, Ruihan Bao, Keiko Harimoto, and Xu Sun. 2022. Stock Trading Volume Prediction with Dual-Process Meta-Learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 137–153.

[8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. (2019).

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[10] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

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

[12] Farhad Soleimanian Gharehchopogh, Tahmineh Haddadi Bonab, and Seyyed Reza Khaze. 2013. A linear regression approach to prediction of stock market trading volume: a case study. *International Journal of Managing Value and Supply Chains* 4, 3 (2013), 25.

[13] Terrence Hendershott and Ryan Riordan. 2013. Algorithmic trading and the market for liquidity. *Journal of Financial and Quantitative Analysis* 48, 4 (2013), 1001–1024.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*. PMLR, 4904–4916.

[16] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[18] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. 2022. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4953–4963.

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023).

[20] Qing Li, Jinghua Tan, Jun Wang, and Hsinchun Chen. 2021. A Multimodal Event-Driven LSTM Model for Stock Prediction Using Online News. *IEEE Transactions on Knowledge and Data Engineering* 33, 10 (2021), 3323–3337. https://doi.org/10.1109/TKDE.2020.2968894

[21] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2021. Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 4541–4547.

[22] Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems* 69 (2014), 14–23.

[23] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).

[24] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, 61–68. https://doi.org/10.18653/v1/2022.acl-short.8

[25] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602* (2021).

[26] Xiaotao Liu and Kin Keung Lai. 2017. Intraday volume percentages forecasting using a dynamic SVM-based approach. *Journal of Systems Science and Complexity* 30 (2017), 421–433.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[28] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* 32 (2019).

[30] Wenjie Lu, Jiazheng Li, Jingyang Wang, and Lele Qin. 2021. A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications* 33 (2021), 4741–4753.

[31] Anshul Mittal and Arpit Goel. 2012. Stock prediction using twitter sentiment analysis. *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)* 15 (2012), 2352.

[32] Brian K Nelson. 1998. Time series analysis using autoregressive integrated moving average (ARIMA) models. *Academic emergency medicine* 5, 7 (1998), 739–744.

[33] Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications* 42, 24 (2015), 9603–9611.

[34] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[35] Venkata Sasank Pagolu, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. 2016. Sentiment analysis of Twitter data for predicting stock market movements. In *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*. IEEE, 1345–1350.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[38] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. 2017. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 1643–1647.

[39] Philip Treleaven, Michal Galas, and Vidhi Lalchand. 2013. Algorithmic trading review. *Commun. ACM* 56, 11 (2013), 76–85.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[41] Qianqian Xie, Weiguang Han, Yanzhao Lai, Min Peng, and Jimin Huang. 2023. The Wall Street Neophyte: A Zero-Shot Analysis of ChatGPT Over MultiModal Stock Movement Prediction Challenges. *arXiv preprint arXiv:2304.05351* (2023).

[42] 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)*. 1970–1979.

[43] Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097* (2020).

[44] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[45] Zhiyuan Zhang, Wei Li, Ruihan Bao, Keiko Harimoto, Yunfang Wu, and Xu Sun. 2023. ASAT: Adaptively scaled adversarial training in time series. *Neurocomputing* 522 (2023), 11–23.

[46] Liang Zhao, Wei Li, Ruihan Bao, Keiko Harimoto, Xu Sun, et al. 2021. Long-term, Short-term and Sudden Event: Trading Volume Movement Prediction with Graph-based Multi-view Modeling. *arXiv preprint arXiv:2108.11318* (2021).

[47] Yanzhao Zou and Dorien Herremans. 2022. A multimodal model with Twitter FinBERT embeddings for extreme price movement prediction of Bitcoin. *arXiv preprint arXiv:2206.00648* (2022).