PEN: Prediction-Explanation Network to Forecast Stock Price Movement with Better Explainability

Shuqi Li\textsuperscript{1,*}, Weiheng Liao\textsuperscript{2*}, Yuhan Chen\textsuperscript{1*}, Rui Yan\textsuperscript{1, 3†}

\textsuperscript{1}Gaoling School of Artificial Intelligence (GSAI), Renmin University of China
\textsuperscript{2}MADE by DATA \textsuperscript{‡}
\textsuperscript{3}Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education

\textsuperscript{1}Shuqi Li\textsuperscript{1,*}, Weiheng Liao\textsuperscript{2*}, Yuhan Chen\textsuperscript{1*}, Rui Yan\textsuperscript{1, 3†}
\textsuperscript{1}Gaoling School of Artificial Intelligence (GSAI), Renmin University of China
\textsuperscript{2}MADE by DATA \textsuperscript{‡}
\textsuperscript{3}Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education

Abstract
Nowadays explainability in stock price movement prediction is attracting increasing attention in banks, hedge funds and asset managers, primarily due to audit or regulatory reasons. Text data such as financial news and social media posts can be part of the reasons for stock price movement. To this end, we propose a novel framework of Prediction-Explanation Network (PEN) jointly modeling text streams and price streams with alignment. The key component of the PEN model is an aligned representation learning module that learns which texts are possibly associated with the stock price movement by modeling the interaction between the text data and stock price data with a salient vector characterizing their correlation. In this way, the PEN model is able to predict the stock price movement by identifying and utilizing abundant messages while on the other hand, the selected text messages also explain the stock price movement. Experiments on real-world datasets demonstrate that we are able to kill two birds with one stone: in terms of accuracy, the proposed PEN model outperforms the state-of-art baseline; on explainability, the PEN model is demonstrated to be far superior to attention mechanisms, capable of picking out the crucial texts with a very high confidence.

Introduction
Stock price movement prediction, a hugely challenging but equally rewarding task, has always drawn enormous interest from both academia and industry (Frankel and Frankel 1995; Bollen, Mao, and Zeng 2011; Edwards, Magee, and Bassetti 2018). Classical approaches like time series analysis have remained popular due to their simple structures and interpretability. In recent years, deep learning techniques have gained much popularity due to their ability to improve the prediction accuracy when there are large quantities of training data, as often is the case of financial markets.

For better prediction, an external network becomes a possible solution via alignment of financial news commentaries and social media posts, revealing rich information about the market, far beyond price, trading volume or financial KPIs

*These authors contributed equally.
†Corresponding author: Rui Yan (ruiyan@ruc.edu.cn)
‡https://www.madebydata.com

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related Work
Stock Price Movement Prediction. Traditional approach tends to focus on identifying patterns and correlations in

5187
trading prices and volumes to predict the direction of price movement in the future. In recent years, researchers successfully exploited text information in stock price prediction, taking advantage of the unprecedented increase of online text data such as news and social media posts. This type of models are possible because news, announcements and sometimes rumours may have a direct impact on stock price. Li et al. (2014) projected textual news onto the sentiment space and implement a generic stock price prediction framework. Khadje Nassiriroussi et al. (2015) proposed predict intraday directional-movements of a currency-pair in the foreign exchange market based on the text of breaking financial news-headlines. Hu et al. (2018) designed a Hybrid Attention Networks to predict the stock trend based on the sequence of recent related news. Xu and Cohen (2018) took a step further, their multi-layer StockNet ingests both stock price data and tweets data, uses attentions and variational autoencoders to extract latent information and makes prediction. Carta et al. (2021) propose a feature engineering process to create an extended set of features extracted using generated lexicons and news. While models with either just text data, or just stock price data have been demonstrated to yield fairly good prediction results, many believe that models that make sense of both type of datasets can be a game changer. This is because stock price is greatly influenced by both what’s happening in the market (which is captured by text data) and past patterns (which can be learnt from historical stock price).

**Prediction Explainability.** Despite big advances in deep stock price prediction models, their adoption is unfortunately very limited due to their “black box” nature. So far only a tiny number of research articles have attempted to address prediction explainability. Past work includes Hu et al. (2018) analyzed attention weights of text corpora to empirically study the importance of different news articles, and Dang, Shah, and Zerfos (2019) proposed a multi-modality neural model to discover news relevant to stock prediction.

In this paper, we take a step further to propose a Shared Representation Learning (SRL) module, which jointly analyzes text embeddings and stock price data and learns their shared representation. This module generates a Vector of Salience (VoS) to explain the importance of individual text documents in the corpora, which is then further regulated in the learning objective to maximize explainability and improve prediction accuracy.

**Prediction-Explanation Network**

**Problem Description**

We formulate the stock price movement prediction problem as a classification task. For one stock, given the past text corpora $C$ and historical prices dataset $P$ [$t - L$, $t - 1$] where $L$ is the fixed lag size, the objective is to predict the movement $y$ of the stock on next day $t$. The movement $y$ can be constructed as binary:

$$y = \begin{cases} 1, & p_t > p_{t-1} \\ 0, & p_t \leq p_{t-1} \end{cases}$$

where $p_t$ denotes the adjusted close prices at day $t$ for the given stock. The adjusted close price amends a stock’s closing price to reflect that stock’s value after accounting for any corporate actions, which is often used as an important element to predict stock price movement (Yoo et al. 2021).

**Overall Architecture**

We summarize the overall architecture of our proposed stock price Prediction-Explanation Network (PEN) in Figure 1. It comprises four main components: Text Embedding Layer (TEL) captures text information and obtains low-dimensional representations; Shared Representation Learning (SRL) models the interaction between text data and stock price data and produces Vector of Salient (VoS) that indicates the importance of each text; Deep Recurrent Generation (DRG) infers the latent variable $Z$ and decodes $y$ from $Z$ and $X$; Temporal Attention Prediction (TAP) employs attention mechanism to produce the final prediction from multiple predictions at different time steps.

**Text Embedding Layer**

In order to capture the information from the past and future words as its context and obtain lower-dimensional representations, we leverage a bi-directional GRU layer for every text to obtain text embeddings. Then, all texts in a day can be represent as the matrix $e_t = [e_{t1}, e_{t2}, \ldots, e_{tm}, \ldots, e_{tM}] \in R^{M \times h_w}$, where $M$ denotes the number of texts and $e_{tm}$ denotes word embedding matrix for $m$th text with length $l$ and hidden size $h_w$. We run the bi-directional GRU layer for every text to obtain text embeddings. Then, all texts in a day can be represent as the matrix $e_t = [e_{t1}, e_{t2}, \ldots, e_{tm}, \ldots, e_{tM}] \in R^{h \times M}$, $e_{tM} \in R^{h \times 1}$ is the embedding of $m$th text in $t$th trading day with hidden size $h$. The maximum number of tokens included in a text and the maximum number of texts on a single trading day are set to 30 and 20, respectively.

**Shared Representation Learning**

For price data in $t$th trading day of a stock, the price vector $p_t = [p_{t1}, p_{t2}, p_{t3}] \in R^{3 \times 1}$ consists of adjusted close price $p_{t1}^c$, high price $p_{t2}^h$ and low price $p_{t3}^l$. After normalization with $p_t = p_t / p_{t-1} - 1$, $p_t \in R^{3 \times 1}$ as is input sent into SRL.

SRL ingests the text embeddings $e_t \in R^{h \times M}$ and price data $p_t \in R^{3 \times 1}$ and learns their shared representation. Dif-
Different SRL modules are connected at different time steps, resulting in a recurrent structure. Each SRL module consists of three units, Text Selection Unit (TSU), Text Memory Unit (TMU), Information Fusion Unit (IFU). The details of SRL are shown in Figure 2.

**Text Selection Unit.** Raw text data obtained is usually highly general, and is not step-wise synchronized with stock price time series. Furthermore, large variance in texts quality means some input texts may be totally irrelevant in predicting stock price movement, hence may bring more noise to our prediction objective. Here, we construct a Text Selection Unit (TSU) to select the useful text embeddings in a day and generate a Vector of Salience (VoS) \( \omega_t \in R^{M \times 1} \), in which each single scalar represents the importance of individual text.

\[
\omega_t = \text{softmax} \left[ k_t^T \tanh \left( W_1 h_{t-1} + W_2 e_t + b_1 \right) \right]
\]

(2)

We generate initial hidden state \( h_0 \in R^{h \times 1} \) by Xavier algorithm (Glorot and Bengio 2010) and \( h_{t-1} \) is the former hidden state. \( i_t \in R^{h \times 1} \) is text embedding weighted vector. \( W_1, W_2, k_t \) are weight matrices and \( b_1 \) is bias.

**Text Memory Unit.** As it takes time for the market to digest news and announcements, the past information contained in texts is valuable and cannot simply be discarded. Taking this into consideration and inspired by the structure of a LSTM cell, we design a Text Memory Unit to preserve important information in text embeddings over time, with forget gate \( f_t \) and output gate \( o_t \) to regulate information flowing in and out of the cell.

\[
f_t = \sigma \left( W_3 [i_t, h_{t-1}] + b_3 \right) \quad o_t = \sigma \left( W_4 [i_t, h_{t-1}] + b_4 \right) \quad i_t = \tanh \left( W_5 [i_t, h_{t-1}] + b_5 \right) \quad l_t = f_t l_{t-1} + o_t i_t
\]

(3)

\( l_t \in R^{h \times 1} \) is text memory state with hidden state \( h_t \) and it is initialized by Xavier algorithm. \( W_3, W_4, W_5 \) are weight matrices and \( b_3, b_4, b_5 \) are biases.

**Information Fusion Unit.** The key idea of the SRL module is to learn the shared representation for texts and stock prices then exploit the interaction patterns to identify the important texts. We introduce an Information Fusion Unit (IFU), which ingests text embeddings and stock prices, and fuses them together.

\[
d_t = \sigma \left( W_6 [p_t, l_t, h_{t-1}] \right) \quad h_t = \tanh \left( W_7 [l_t, h_{t-1}] \right) \quad h_p = \tanh \left( W_8 [p_t, h_{t-1}] \right) \quad h_t = d_t h_p + (1 - d_t) h_t
\]

(4)

where the hidden state \( h_t \) and price information \( p_t \) is the shared representation of text memory \( l_t \) and price information \( p_t \). \( W_6, W_7, W_8 \) are weight matrices.

It is worth noting that the hidden state \( h_t \) of IFU is also an output of the SRL module, which is then used to next SRL module at the next time step \( t+1 \). We regard the last hidden state \( h_t \), i.e. the hidden state of the last day in time lag, as input of deep recurrent generation module.

**Deep Recurrent Generation**

A variational auto-encoder provides a probabilistic manner for describing an observation in latent space. Inspired by the stocknet (Xu and Cohen 2018), we use a recurrent variational auto-encoder to generate stock price movements.

Given input \( X = [x_1; \ldots; x_T] \) for every trading day \( t \in [1, \ldots, T] \) where \( x_t = h_t \) is the output of IFU. In variational auto-encoder, we need to construct latent variable \( Z = [z_1; \ldots; z_T] \) and then predict stock movement \( y = [y_1, \ldots, y_T] \). Formally, we have to model the conditional probability distribution \( p_0(y \mid X) \) and its factorization is as follows

\[
p_0(y \mid X) = \int_Z p_0(y, Z \mid X) = \int_Z \prod_{t=1}^{T-1} p_0(y_t \mid x_{t \leq t}, z_t) p_0(z_t \mid z_{<T}, X)
\]

As it is shown in above equation, we need to infer the intractable posterior distribution \( p_0 (Z \mid X, y) \). A common
way to solve this problem is to generate a distribution \( q_\theta (Z \mid X, y) \) by variational inference (Jordan et al. 1999) to approximate \( p_\theta (Z \mid X, y) \) and then simulate \( q_\phi (Z \mid X, y) \) by using reparameterization in neural network.

Here an alternative way is used to restrict the family of distributions \( q_\phi (Z \mid X, y) \). Suppose \( Z \) can be partitioned into disjoint time classes, we then assume that the approximate distribution factorizes with respect to different time points as follows,

\[
q_\phi(Z \mid X, y) = \prod_{t=1}^{T} q_\phi(z_t \mid z_{<t}, x_{\leq t}, y_t)
\]

(6)

The likelihood of our target conditional probability distribution \( p_\theta(y \mid X) \) can be decomposed into,

\[
\begin{align*}
\log p_\theta(y \mid X) & = \log \int_Z p_\theta(y, Z \mid X) dZ \\
& + E_{q_\phi(Z \mid X, y)} \left[ \log p_\theta(y \mid X, Z) \right] \\
& - D_{KL} \left[ q_\phi(Z \mid X, y) \parallel p_\theta(Z \mid X) \right]
\end{align*}
\]

(7)

Therefore, minimizing the Kullback-Leibler divergence between \( p_\theta(Z \mid X, y) \) and its approximation \( q_\phi(Z \mid X, y) \) equals maximizing the last two terms of Eq. (7), i.e. the variational recurrent lower bound as follows,

\[
\begin{align*}
L(\theta, \phi; X, y) & = \sum_{t=1}^{T} E_{q_\phi(z_{t \mid z_{<t}, x_{\leq t}, y_t})} \{ \log p_\theta(y_t \mid x, z) \} \\
& - D_{KL} \left[ q_\phi(z_t \mid z_{<t}, x_{\leq t}, y_t) \parallel p_\theta(z_t \mid z_{<t}, x_{\leq t}) \right] \leq \log p_\theta(y \mid X)
\end{align*}
\]

(8)

We assume that when \( t < T \), \( y_t \) is independent of \( z_{\leq t} \) so that \( p_\theta(y_t \mid x, z) \) in the Eq. (8) equals to \( p_\theta(y_t \mid x_{\leq t}, z_{\leq t}) \) while \( p_\theta(y_t \mid x, z) = p_\theta(y_t \mid X, Z) \).

**Recurrent Variational Encoder & Decoder.** We use a Recurrently Variational Auto-encoder based (Li et al. 2017) framework to conduct variational inference and generation. In the encoder and decoder stage, GRU is employed as the basic recurrent model to extract information from input and decode from latent variable instead of Fully Connected Layer. We assume that both the prior and posterior of the latent variables are of Gaussian distribution. namely, \( p_\theta(z_t \mid z_{<t}, x_{\leq t}) \sim \mathcal{N}(\mu_t; \mu_\theta, \sigma_\theta^2 I) \) and \( q_\phi(z_t \mid z_{<t}, x_{\leq t}, y_t) \sim \mathcal{N}(\mu_t; \mu_\phi, \sigma_\phi^2 I) \).

For the posterior, denoting the hidden state of encoder GRU as \( h^{enc}_t \) and its shared representation with \( z_t \) as \( h^{enc}_t = \tanh \left( W^\phi_z [z_{t-1}, x_t, h^{enc}_{t-1}, y_t] + b^\phi_z \right) \), so we can calculate \( \mu_t^\phi, \log(\sigma_t^\phi)^2 \) and reparameterize the posterior \( z_t \) by,

\[
\begin{align*}
\mu_t^\phi & = W_{\phi, \mu_\phi} h^{enc}_t + b_{\mu_\phi}^\phi \\
\log(\sigma_t^\phi)^2 & = W_{\phi, \sigma_\phi} h^{enc}_t + b_{\sigma_\phi}^\phi \\
& \Rightarrow z_{post}^t \sim \mathcal{N}(\mu_t^\phi; \sigma_t^\phi) \odot \epsilon
\end{align*}
\]

(9)

where \( \epsilon \sim \mathcal{N}(0, I) \) is white Gaussian noise, \( W^\phi_z, b^\phi_z, W_{\phi, \mu_\phi}, W_{\phi, \sigma_\phi} \) are weight matrices and \( b_{\mu_\phi}^\phi, b_{\sigma_\phi}^\phi \) are bias vectors.

For the prior, the shared representation of \( z_t \) and \( h^{enc}_t \) is \( h^{enc}_t = \tanh \left( W^\phi_z [z_{t-1}, x_t, h^{enc}_{t-1}] + b^\phi_z \right) \), and \( \mu_t^\phi, \log(\sigma_t^\phi)^2 \)

and the prior \( z_t \) are calculated by,

\[
\begin{align*}
\mu_t^\phi & = W_{\phi, \mu_\phi} h^{enc}_t + b_{\mu_\phi}^\phi \\
\log(\sigma_t^\phi)^2 & = W_{\phi, \sigma_\phi} h^{enc}_t + b_{\sigma_\phi}^\phi \\
& \Rightarrow z_{prior}^t = \mu_t^\phi + \sigma_t^\phi \odot \epsilon
\end{align*}
\]

(10)

where \( W_{\phi, \mu_\phi}, W_{\phi, \sigma_\phi} \) are weight matrices and \( b_{\mu_\phi}^\phi, b_{\sigma_\phi}^\phi \) are bias vectors. Then we integrate the recurrent generative decoding component with the discriminative deterministic decoding component to predict stock price movement \( \hat{y}_t \) by,

\[
\begin{align*}
& h_t^{dec} = \tanh \left( W_{dec} [z_t, h^{enc}_t, z_{post}^t + b_{dec}] \right) \\
& \hat{y}_t = \text{softmax} \left( W_{y} h_t^{dec} + b_{y} \right), t < T
\end{align*}
\]

(11)

where \( W_{dec}, W_{y} \) are weight matrices and \( b_{dec}, b_{y} \) are bias vectors.

**Temporal Attention Prediction**

To explore the relationship between prediction target \( \hat{y}_T \) with its former information \( \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{T-1} \), we adopt a temporal attention mechanism as shown in Figure 3. The decoder hidden state \( h_t^{dec} = [h_1^{dec}, \ldots, h_{T-1}^{dec}] \) is used to calculate attention weight. The dependency score vector \( q^{dec} \), information score vector \( k^{dec} \), normalized attention weight vector \( u^{dec} \) and value vector \( v^{dec} \) are denoted as follows, respectively.

\[
\begin{align*}
q^{dec} & = \left( H_{T}^{dec} \right)^{\top} \tanh \left( W_q H_{dec} \right) \\
k^{dec} & = W_k^{dec} \tanh \left( W_{k} H_{dec} \right) \\
u^{dec} & = \text{softmax} \left( q^{dec} \odot (k^{dec})^{\top} \right) \\
v^{dec} & = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{T-1}]
\end{align*}
\]

(12)

where \( W_q, W_k \) are weight matrices. \( q^{dec} \) represents the hidden state of \( T \)th trading day, information score vector \( k^{dec} \) includes stock movements information of the past trading days and the weight vector \( u^{dec} \) measures the correlation between target day with the past trading days.

Finally, we have target prediction \( \hat{y}_t \) as a temporal attention of hidden states as follows,

\[
\hat{y}_T = \text{softmax}(W_{T}^{dec} [q^{dec} \top (W_{dec}^{dec})^{\top}, h_t^{dec}] + b_{dec})
\]

(13)

where \( W_{T}^{dec} \) is a weight matrix and \( b_{dec} \) is a bias vector.
Learning Objective

As its name suggests, the learning objective of PEN consists of two parts: to maximize the prediction accuracy and to maximize the explainability of prediction results by selecting the most relevant input texts.

Based on Eq. (8), the objective $L_t$ of $t$th day $t \in [1, \ldots, T]$ can be isolated as

$$L_t = \log p_\theta(y_t | x_{<t}, z_{<t}) - \lambda_t D_{KL}(q_\phi(z_t | z_{<t}, x_{<t}, y_t) \parallel p_\theta(z_t | z_{<t}, x_{<t}))$$

where $\lambda_t \in (0, 1]$ is a decay weight from the KL term annealing trick. We then apply temporal attentions to $L_t$ and average samples to formulate the first optimization objective:

$$L_1(\theta, \phi; X, y) = \frac{1}{N} \sum_n w^{obj}(n) L_t(n)$$

where $w^{obj} = [\lambda_t w^{dec}, 1]$ is the global temporal weight vector, which can be adjusted by the hyper parameter $\lambda_t$.

For explainability, we design a salient regulator to strengthen SRL module’s capability of information concentration. This can be achieved by maximize the Kullback-Leibler divergence between VoS $\omega_t$ and discrete uniform distribution. So formally, another optimization objective is as follows,

$$L_2(\theta, \phi; X, y) = D_{KL}(\omega_t | p_u) .$$

where $p_u \sim U(M)$ is the discrete uniform distribution, and $M$ is the number of texts in a sample.

Combining the two objectives with equal weight to avoid excessive parameter tuning, we have the overall objective:

$$L(\theta, \phi; X, y) = L_1(\theta, \phi; X, y) + L_2(\theta, \phi; X, y)$$

Experiments

Experimental Setup

Datasets. We train and evaluate our model on two datasets: ACL18 (Xu and Cohen 2018) and Daily News for Stock Price Movement Prediction Dataset (DJIA) 1. We choose these two datasets because 1) they span two distinct time periods; 2) ACL18 is for individual stocks while DJIA is for stock market indices; 3) they include two completely different types of text data: news articles and social media posts (tweets). ACL18 includes the text data and historical price for 88 highly traded US stocks between 2014-01-01 and 2016-01-01 from 9 industries. Texts are tweets retrieved from Twitter and the historical prices data are collected from Yahoo Finance. We process ACL18 in the same way as proposed in (Xu and Cohen 2018). DJIA includes news and price data on Dow Jones Industrial Average from 2008-06-08 to 2016-07-01, where news data is consisted of the top 25 headlines of Reddit WorldNews Channel every day.

Evaluation Metrics. We evaluate the accuracy of model results by two metrics: accuracy (ACC) and Matthews Correlation Coefficient (MCC)(Xu and Cohen 2018).

Parameters Setup. With our PEN model we use Tensorflow to construct the computational graph, initialize all bias zero, all weights with Xavier algorithm (Glorot and Bengio 2010), and optimize the final loss by Adam with learning rate of 1e-3. We use 32 shuffled samples in a batch and a 5-day lag window for model to learn historical context. We set the size of hidden state in SRL module and in word embedding to be 100 and 50, respectively. Besides, we use the input dropout rate of 0.4 to regularize latent variables. As the DJIA dataset is relatively small, we use the models trained on ACL18 as pre-trained models and then fine tune them on DJIA dataset for all baselines.

Baselines. We compare PEN with the following baselines.

- Random: randomly generated movement predictions.
- HAN (Hu et al. 2018): a hybrid attention networks based on related news.
- Stocknet (Xu and Cohen 2018): a deep generative model jointly exploiting text and price signals.
- CPC (Wang et al. 2021): a copula-based contrastive predictive coding method which models relevant macroeconomic variables to improve prediction accuracy.
Results of Prediction Accuracy
First we make a comparison on ACC and MCC between PEN and Stocknet with a varied $\lambda_t$ (see Eq. (15)) which controls the impact of past loss. As is shown in Figure 4, PEN outperforms Stocknet for almost every value of parameter $\lambda_t$, and achieves the best performance on ACL18 when $\lambda_t = 0.1$, and on DJIA when $\lambda_t = 0.7$. These two $\lambda_t$ values are adopted in PEN to carry out the rest of experiments.

Table 1: Prediction results of baseline models and PEN. These results show that PEN is the best performing model on both datasets under the two accuracy criteria.

<table>
<thead>
<tr>
<th>Models</th>
<th>ACC(%)</th>
<th>MCC</th>
<th>ACC(%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>50.8900</td>
<td>-0.002266</td>
<td>50.2435</td>
<td>0.000360</td>
</tr>
<tr>
<td>RF</td>
<td>53.0800</td>
<td>0.012929</td>
<td>52.6455</td>
<td>0.050990</td>
</tr>
<tr>
<td>HAN</td>
<td>57.6400</td>
<td>0.051800</td>
<td>53.2258</td>
<td>0.060150</td>
</tr>
<tr>
<td>Stocknet</td>
<td>58.2300</td>
<td>0.080796</td>
<td>56.9231</td>
<td>0.136909</td>
</tr>
<tr>
<td>CPC</td>
<td>59.1100</td>
<td>0.181700</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PEN</td>
<td>59.8976</td>
<td>0.155652</td>
<td>60.5128</td>
<td>0.220423</td>
</tr>
</tbody>
</table>

Table 2: A Comparison of HAN and PEN on Explainability Metrics RTT, RoR and Kappa.

<table>
<thead>
<tr>
<th>Model</th>
<th>RTT (%)</th>
<th>RoR (%)</th>
<th>Kappa score</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAN</td>
<td>68.3</td>
<td>54.8</td>
<td>0.430</td>
</tr>
<tr>
<td>PEN</td>
<td>99.5</td>
<td>89.3</td>
<td>0.591</td>
</tr>
</tbody>
</table>

It is shown that our SRL modules are able to focus on just one or two salient texts out of all available text corpora for 99.5% of the test samples, significantly more concentrated than standard attention weights. Moreover, among the samples for human inspection, 89.3% of PEN’s top rated texts agree with at least one human investor with an average kappa score of 0.591. This further demonstrates the effectiveness of SRL module’s selection mechanism.

We further compare the RTT ratios of PEN and HAN over samples with different number of text documents. The results, in Figure 5, show that a large text corpora doesn’t seem to present any challenges to SRL, manifested by a consistent RTT despite increased size of texts.

As examples of qualitative evaluation, we present a couple of examples from ACL18 and DJIA datasets to illustrate how VoS generated by SRL compares with normal attention weights. Table 3 show the top texts picked out by PEN, the corresponding VoS of those texts, and attention weights from HAN. Note the columns “Ground Truth” (GT) indicates the number of individual investors who has the text as their top pick over the total number of human assessors (which is 3). The most relevant texts for Apple on 10/22/2015 are shown in the top half of Table 3. In this particular sample, SRL assigns a VoS weight of 0.994 to investors’ top pick while the second ranked only receives a weight of 0.00386. The top picks by PEN are the same picks by investors. Similarly, the bottom half of Table 3 shows the
Ablation Studies

Variations in PEN Architecture. To understand the contributions of different components in PEN, we conduct a number of ablation experiments where we remove TAP, DRG, SRL, KL-loss from PEN, respectively. The results, shown in Table 4, indicate that each and every part of PEN contributes to the overall model performance. Note when SRL is ablated, we use the normalized attention weights to handle text corpora as described in (Xu and Cohen 2018); this has lead to a reduction in ACC of 3.8% and 4.1% on ACL18 and DJIA, respectively, suggesting SRL modules are far superior to standard attention mechanism in measuring the relevance of text corpora for better prediction.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACL18 ACC(%)</th>
<th>MCC</th>
<th>DJIA ACC(%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o TAP</td>
<td>52.3891</td>
<td>0.027687</td>
<td>57.4359</td>
<td>0.0148</td>
</tr>
<tr>
<td>w/o DRG</td>
<td>51.7065</td>
<td>0.018908</td>
<td>56.9231</td>
<td>0.1587</td>
</tr>
<tr>
<td>w/o SRL</td>
<td>56.1433</td>
<td>0.073811</td>
<td>56.4103</td>
<td>0.1422</td>
</tr>
<tr>
<td>w/o KL</td>
<td>56.6553</td>
<td>0.058124</td>
<td>56.8462</td>
<td>0.0940</td>
</tr>
<tr>
<td>Full inputs</td>
<td>59.8976</td>
<td>0.155652</td>
<td>60.5128</td>
<td>0.2204</td>
</tr>
</tbody>
</table>

Table 4: An ablation study of architecture of PEN.

Stock Price Components. We also carry out a number of ablation experiments where we remove adjusted close prices, high prices and low prices from inputs to see how important they are. The results are summarized in Table 5. We observe that high and low prices are far more important than adjusted close price in this task. This seems to reaffirm the belief by many investors that volatility, reflected by the high and low prices, is a crucial characteristic of a stock. We also notice that full data enables PEN to perform the best, which suggests all types of price are valuable in prediction.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>ACL18 ACC(%)</th>
<th>MCC</th>
<th>DJIA ACC(%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o High</td>
<td>45.7338</td>
<td>0.0372</td>
<td>50.3413</td>
<td>0.0086</td>
</tr>
<tr>
<td>w/o Low</td>
<td>50.3413</td>
<td>0.0086</td>
<td>50.7692</td>
<td>0.0721</td>
</tr>
<tr>
<td>w/o Adj-close</td>
<td>57.6792</td>
<td>0.0890</td>
<td>55.3846</td>
<td>0.1319</td>
</tr>
<tr>
<td>Full inputs</td>
<td>59.8976</td>
<td>0.1556</td>
<td>60.5128</td>
<td>0.2204</td>
</tr>
</tbody>
</table>

Table 5: An ablation study of inputs on ACL18 and DJIA.

Conclusions
As a step to address the challenges of explainability in stock price movement prediction, we propose a Prediction-Explanation Network (PEN). The core of PEN is the Shared Representation Learning (SRL) module, which models the interaction between the text data and stock price data and outputs a Vector of Salience to explain the importance of texts related to stock prices changes. Inspired by the critical thinking process of investors, we further introduce such a regulation mechanism that SRL module focuses the smallest number of texts possible to maximize the explainability. Our experiments demonstrate that not only SRL module is capable of identifying highly relevant texts that explain the future stock price movements, but also greatly enhances the accuracy of our stock price movement prediction model PEN, which establishes new state-of-art accuracy across two benchmark datasets covering two distinct markets.
Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC Grant No. 62122089 and No. 61876196), Beijing Outstanding Young Scientist Program NO. BJJWZJHY1012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Inter-disciplinary Platform for the “Double-First Class” Initiative, Renmin University of China. We also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of RUC. Rui Yan is supported by Beijing Academy of Artificial Intelligence (BAAI) and CCF-Tencent Rhino-Bird Open Research Fund.

References


Pagolu, V. S.; Reddy, K. N.; Panda, G.; and Majhi, B. 2016. Sentiment analysis of Twitter data for predicting stock market movements. In 2016 international conference on signal processing, communication, power and embedded system (SCOPES), 1345–1350. IEEE.


