StockNet: Unifying Text, Metadata and Price Representations for Stock Movement Prediction

Yumo Xu and **Shay B. Cohen**

School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
s1617290@sms.ed.ac.uk, scohen@inf.ed.ac.uk

Abstract

We present an end-to-end hierarchical sequence learning model for stock movement prediction. While most existing research on stock movement prediction from social media data puts little emphasis on the contexts beyond the text, our model captures textual semantics, metadata information, and price signals simultaneously with deep neural networks. We conduct a comprehensive evaluation to our proposed model on the intra-day movements of 88 stocks selected from 9 different sectors. Unifying both fundamental and technical analysis, our model exceedingly enlarges the representation capacity and achieves the state-of-the-art performance on our datasets compared to previous methods for stock price movement prediction.

1 Introduction

Predicting stock price movements is of clear interest to both investors and researchers. Research in psychology and behavioural finance (Kahneman and Tversky 1979; Dolan 2002) has demonstrated the crucial role of investor moods in financial decision-making, providing the theoretical foundation for stock prediction from the sentiment side. In the natural language processing (NLP) field, sentiment analysis (Pang and Lee 2008; Liu 2012) has become a vibrant research topic with the emergence of online social networks. Twitter is one of the most representative social media platforms. Twitter users are allowed to share their opinions on specific topics, and interact through comments and retweets. Although there is a 140-characters limit imposed to the tweet post length, the aggregation of a range of tweets has been proved effective in measuring collective mood states (Bollen, Mao, and Pepe 2011).

Based upon sentiment analysis and deep learning techniques, we aim at predicting intra-day stock price movements by identifying the patterns underlying the stock-relevant tweets retrieved from the preceding days of the target transaction date. For instance, two pieces of retrieved tweets that discuss *$GOOG*, the NASDAQ ticker symbol for *Google Inc.*, along with their respective posters, are shown below,

1. *Richard*: "*$GOOG* Self-driving cars discover the limits of autonomy."
2. *Brittany*: "*$AAPL* is losing customers. Everybody is buying android phones! *$GOOG*."

The contexts of the sentiment contain its holder and its target (Liu 2012). As shown in the first instance, the sentiment holder is the Twitter user *Richard* and the target is *$GOOG*. The second tweet shows the typical circumstance that one tweet contains multiple sentiment targets where *$AAPL*, the stock symbol for *Apple Inc.*, is also a sentiment target besides *$GOOG*. Particularly, *Brittany*'s sentiment is positive on *$GOOG* while negative on *$AAPL*. It is also conceivable that the importance of the tweet content varies by its poster. For instance, when inferring the stock movements, the opinion of a renowned equity analyst is generally considered more valuable than that of an ordinary retail investor. Based upon the above evidence, we expect the semantic patterns underlying tweet texts to be better captured provided that we reasonably incorporate the information of target stocks and tweet users into the predictive model.

Most existing research on stock movement prediction with sentiment analysis techniques relies heavily on feature engineering. Some research calculates the sentiment polarity leveraging static sentiment lexicons (Oliveira, Cortez, and Areal 2013), or manually designs, extracts and aggregates feature patterns (Smailović et al. 2013). More recently, topic models are extended to jointly learn the topics and sentiments for classifiers (Si et al. 2013; Nguyen and Shirai 2015). With the prevalence of deep neural networks (Le and Mikolov 2014), event-driven approaches have also been studied with representation learning techniques. Different from previous news-driven research simply relying on shallow features, structured information is used to represent events for the predictive model. Specifically, Open IE (Fader, Soderland, and Etzioni 2011) is utilised for extracting structured events from the large-scale public news, and both linear as well as nonlinear models are employed to investigate the relationships between events and the stock market (Ding et al. 2014). Besides, event embeddings (Ding et al. 2015) are also introduced to address the sparsity issue, largely benefiting the modelling of event influences.

In this research, we introduce StockNet, an end-to-end hierarchical sequence learning model for stock movement prediction. Different from previous research, StockNet identifies discriminative features automatically from social media datasets and directly learns the representations for sequences without the pre-extraction of structured events. From the financial point of view, StockNet combines both fundamental analysis (Ritchie 1996) and technical analysis[1](Edwards, Bassetti, and Magee 2007); from the NLP perspective, StockNet unifies text, metadata and price representations with supervised deep learning approaches. Compared with the benchmarks, StockNet exceedingly enhances the representation capacity and achieves the state-of-the-art performance on the movement prediction task.

2 Problem Formulation

We regard stock movement prediction as a binary classification task where 1 denotes rise and 0 denotes fall. A stock collection indexed with $m \in \{1, 2, ..., M\}$ is pre-selected as the research target. For the mth stock at a transaction date t, by recognising the patterns underlying the stock-relevant tweet collection $\mathcal{D}_m^{[t-\Delta t_d,\, t-1]}$ and the historical price signals $\mathcal{P}_m^{[t-\Delta t_p,\, t-1]}$ where Δt_d and Δt_p denote the window sizes for retrieving tweets and prices, respectively, we predict the intra-day movement $y_m^{(t)}$,

$$y_m^{(t)} = \mathbb{1}\left(c_m^{(t)} > c_m^{(t-1)}\right) \tag{1}$$

where $c_m^{(t)}$ denotes the closing price. Following Rao and Srivastava (2012), we adopt the closing price instead of the adjusted closing price adjusted for the corporate actions affecting stock prices such as dividends and splits (Xie et al. 2013), since the impact of such events on investor emotions are also reflected in tweets and it is thus possible to predict the real stock movement trend with our datasets.

3 Data Collection

Since high-trade-volume-stocks tend to be discussed more on Twitter, we select 88 stocks to target, coming from all the 8 stocks in the *Conglomerates* sector and the top 10 stocks in capital size in each of the other 8 sectors.

There are two main components in our dataset: the Twitter dataset and the historical price dataset. We access the Twitter data under the official license of Twitter, then retrieve stock-specific tweets from the corpus by querying the respective regex made up of the NASDAQ ticker symbol, for instance, "\$GOOG\b" for *Google Inc.*. Two-year Twitter data starting from 01/01/2014 to 01/01/2016 are retrieved. We preprocess tweet texts using the NLTK package (Bird, Klein, and Loper 2009) with the particular Twitter mode, including tokenisation and taking care of hyperlinks, hashtags and the "@" identifier. We extract the historical prices for the 88 selected stocks from Yahoo Finance[2] to build the historical price dataset.

Samples from 01/01/2014 to 01/10/2015 are used to build the training dataset and the last two months are for the test

[1]To a fundamentalist, stocks have their intrinsic values that can be derived from the behaviour and performance of their company. On the contrary, technical analysis considers only the trends and patterns of the stock price.

[2]http://finance.yahoo.com/

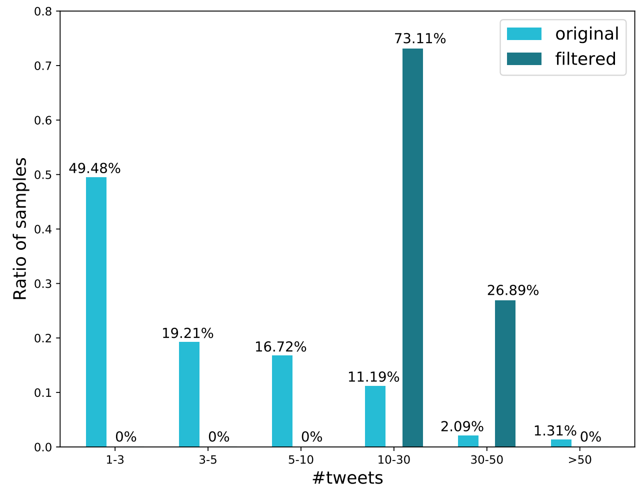

Figure 1: The distribution of samples over the number of tweets contained in each sample.

dataset. We acquire 23,657 original samples, each of which contains a non-zero number of tweets as the input. Nevertheless, most of them contain small numbers of tweets, providing exceptionally insufficient information. Conceivably, it makes little sense to discover stock movement patterns from one or two pieces of tweets, and the mapping learnt from the poor-constructed training samples will harm the generalisation capacity of our model. Hence, we set the minimal tweet sequence length contained by an eligible sample, to 10, resulting in 4,327 samples in the training dataset and 761 samples in the test dataset. Additionally, due to the finite computation resources, we also set the maximal tweet sequence length to 50, over which the excess is clipped. The ratio of samples containing over 50 tweets is relatively small (around 1.31%), and the clipping thus leads to negligible information loss. Figure 1 shows the detailed comparison between the distributions of the original and the filtered samples over the number of tweets contained in each sample.

To control the number of learning parameters, we discard rare words by setting the frequency threshold to 2, ending up with the vocabulary size of 29,866. There are 32,149 distinct users in the original training set. Since user information will be incorporated into and thus affect our prediction model, slimmed user lists with various sizes are also obtained by setting user activity thresholds, which will be elaborated in Section 5. The token "UNK" is appended to the vocabulary and the user list to handle unknown words and users.

Figure 2 shows the distribution of stocks over the number of samples that satisfy our constraints, indicating the high variance of eligible sample numbers between stocks. To be more coarse-grained, the distribution of eligible samples over the nine sectors is also shown in the below subplot of Figure 2. Noticeably, stocks in the *Financial* and *Technology* sectors are more likely to be discussed on Twitter while stocks in the *Healthcare* sector are on the contrary.[3]

[3]Our dataset is available at `http://anonymized`.

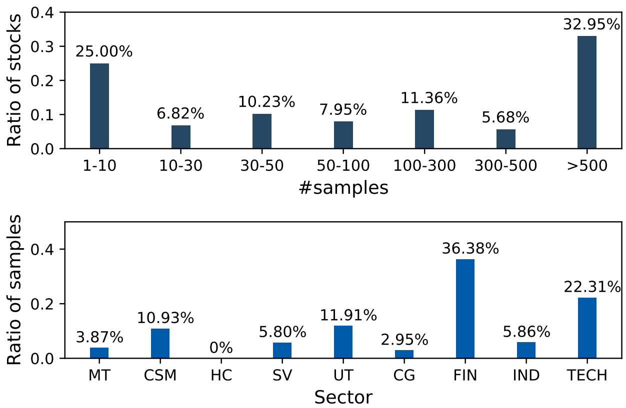

Figure 2: The distribution of stocks over the number of samples (above) and the distribution of samples over the sectors (below). MT, CSM, HC, SV, UT, CG, FIN, IND, TECH denote *Basic Materials*, *Consumer Goods*, *Healthcare*, *Services*, *Utilities*, *Conglomerates*, *Financial*, *Industrial Goods* and *Technology*, respectively.

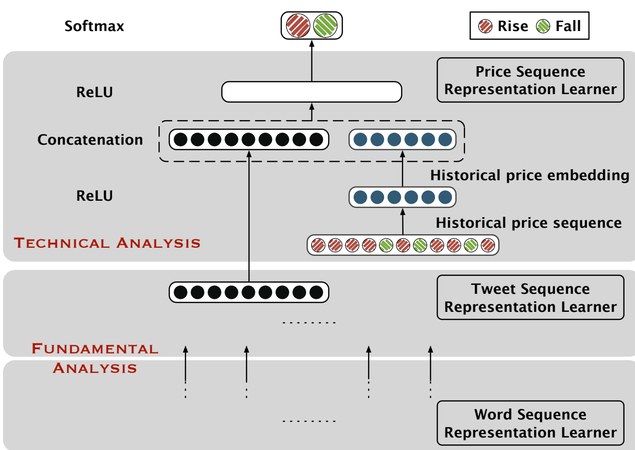

Figure 3: StockNet: hierarchical sequence learning

4 Architecture and Training

We detail the architecture used and the way we estimate it.

General Framework for StockNet

As shown in Figure 3, StockNet combines both fundamental and technical analysis for stock movement prediction following a bottom-up fashion. The fundamental analysis contains two main constituents: the word sequence representation learner (WSRL) and the tweet sequence representation learner (TSRL). As the output of TSRL, the representation for the tweet collection indicating future price trend is fed into the price sequence representation learner (PSRL) to be concatenated with the historical price embedding. PSRL integrates both the current and the historical price information, yielding the final representation that is regarded as the features for prediction by feeding into the fully-connected layer

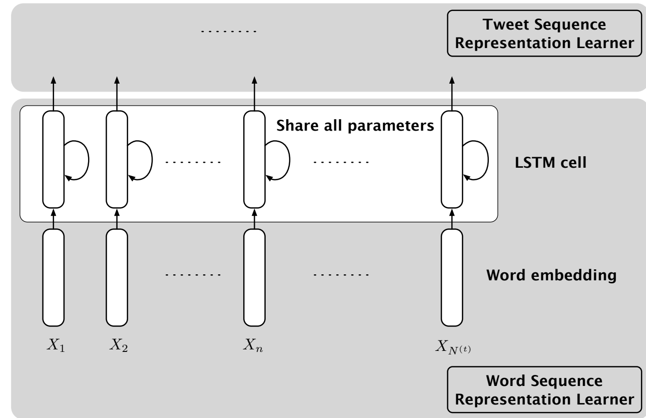

Figure 4: WSRL with Raw LSTMs (Raw-WSRL)

and then the last *softmax* layer to output the distribution of the confidence over upward and downward movements.

In PSRL, for the mth stock at the transaction date t, we regard its historical price movements of the past Δt_p days, $y_m^{[t-\Delta t_p,\, t-1]}$ as the price signals $\mathcal{P}_m^{[t-\Delta t_p,\, t-1]}$, and use it to build the historical price embedding,

$$v_m^{(t)} = \mathrm{ReLU}(W_v y_m^{[t-\Delta t_p,\, t-1]} + b_v) \tag{2}$$

One could also try more sophisticated learning strategy to mine the historical price pattern more precisely. For instance, applying RNNs to the recognition of the triangle patterns indicating an important clue to the trend of future change in stock prices (Kamijo and Tanigawa 1990). In this research, we mainly focus on the representation learning from tweets for fundamental analysis, including WSRL and TSRL. The overall strategy is to first learn the representation for the word sequence in each tweet with WSRL, then gather the representations as the inputs for TSRL to learn the collective representation of the tweet sequence in each sample. WSRL and TSRL will be discussed at length.

Word Sequence Representation Learner (WSRL)

Raw-WSRL We first propose WSRL with raw LSTMs (Raw-WSRL). As shown in Figure 4, Raw-WSRL adopts the same LSTM cell to learn the representations for the multiple tweets contained in one single sample. Specifically, for one single sample of the mth stock on the transaction date t, there are $N_m^{(t)}$ tweets whose representations are required for the prediction task. We use X_n, $n \in \left[1, N_m^{(t)}\right]$ to denote the word embedding matrix for the nth tweet, built with the corresponding word embedding sequence of each tweet. All word embedding matrices are fed as the inputs of Raw-WSRL and processed in the LSTM cell with shared parameters. As outputs, a set of representations for all tweets contained in the given sample is acquired.

Raw-WSRL utilises only the textual information in tweets. To better handle the circumstance that multiple stocks are discussed in one single tweet, we further propose Att-WSRL and Bi-WSRL to incorporate stock signals.

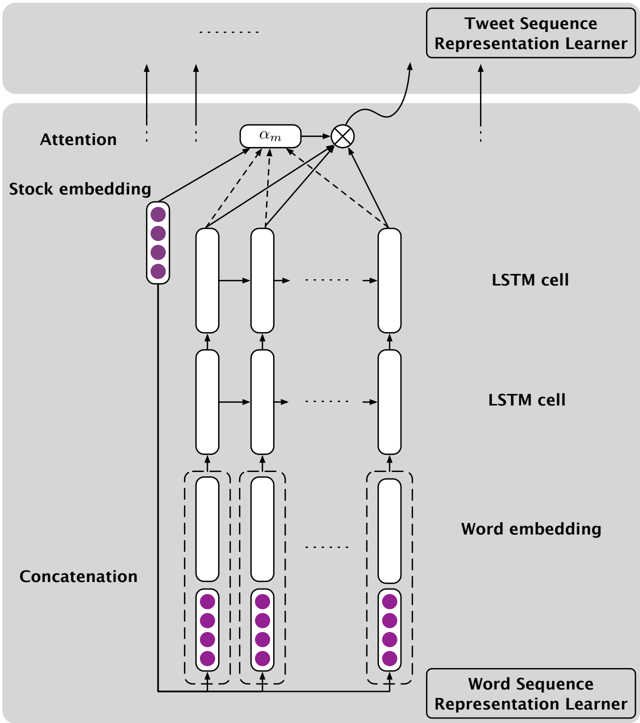

Figure 5: WSRL with Attentive LSTMs (Att-WSRL)

Att-WSRL The network, shown in Figure 5, introduces the stock embedding to selectively focus on different parts of the word sequence and learn the stock-specific representation of the word sequence with the attention mechanism (Wang et al. 2016). For better visualisation, Figure 5 presents the processing flow for only one single word sequence. Similar to Raw-WSRL, all learning parameters in the LSTM cell and attention mechanism are shared between all word sequences.

For the word sequence $W = \{w_1, w_2, ..., w_L\}$ in a tweet discussing the mth stock, we use the embedding of the mth stock v_m , and the hidden states $H = [h_1\ h_2\ ...\ h_L]$, to calculate the attention weight vector α_m, shown as follows,

$$M = \text{ReLU}(\begin{bmatrix} W_h H \\ W_v v_m \otimes e_L \end{bmatrix}) \tag{3}$$

$$\alpha_m = \text{Softmax}(w^{\intercal} M) \tag{4}$$

where $v_m \otimes e_L = [v_m; v_m; ...; v_m]$. Then α_m combines hidden states to output the stock-specific representation d_n:

$$r = H\alpha_m^{\intercal} \tag{5}$$

$$d_n = \text{ReLU}(W_r r + W_L h_L) \tag{6}$$

Bi-WSRL Since stock-relevant tweets are retrieved according to the stock symbols in the data preparation phase, the corresponding symbol is guaranteed to occur in the word sequence. Instead of focusing on different parts of the tweet by calculating the semantic similarities between word embeddings and the stock embedding in Att-WSRL, in Bi-WSRL we simply assume that the symbol position in the tweet word sequence gives the most significant information in contextual modelling, and thus model the word sequence with the bidirectional tactic.

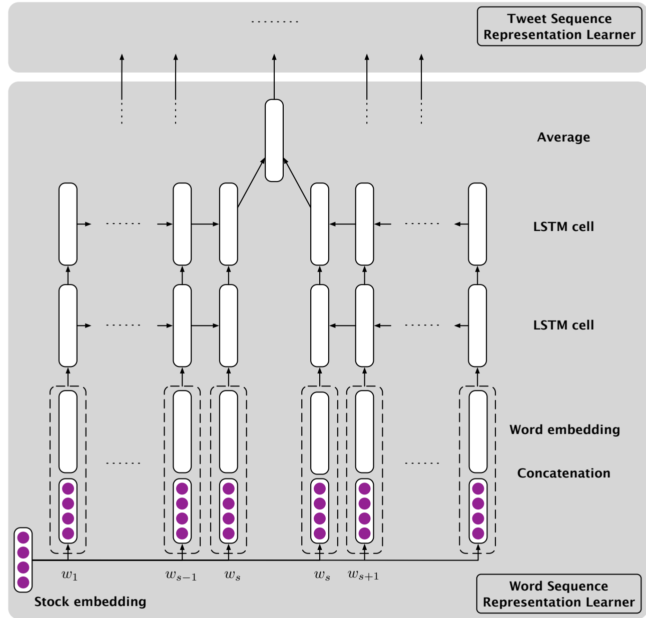

Figure 6: WSRL with Bidirectional LSTMs (Bi-WSRL)

As shown in Figure 6, Bi-WSRL consists of the forward and the backward LSTMs that model the preceding and following contexts of the stock symbol, respectively. Specifically, for the word sequence $W = \{w_1, w_2, ..., w_L\}$, and the stock symbol index s, we run the forward LSTMs on the sequence $W_f = \{w_1, w_2, ..., w_s\}$ and the backward LSTMs on $W_b = \{w_L, w_{L-1}, ..., w_s\}$. The stock symbol is regarded as the last unit in both the preceding and the following contexts, and the final outputs of the two LSTMs are composed to acquire the final representation of the word sequence. Here, we average the two outputs, but the summation and the concatenation can also be investigated as alternatives.

Tweet Sequence Representation Learner (TSRL)

TSRL learns the representation for the tweet sequence based upon the representation collection of word sequences produced by WSRL. In the last section, we introduced the methods for incorporating stock signals in tweet understanding, and we will further present the approaches to make the representation user-dependent at the TSRL level. Initially, we have a user collection indexed with $k \in \{1, 2, ..., K\}$ and to incorporate user distinctions, for the k_nth user that posts the nth tweet in a single sample, we introduce the user importance matrix U_{k_n} to incorporate user signals.

UR-TSRL As per the nature of time series, TSRL with user-modified RNNs (UR-TSRL) takes the user-modified representation sequence sorted by the posting timestamp as the input and uses another LSTM network to generate the ultimate sentiment representation for the given tweet set.

As shown in Figure 7, UR-TSRL adopts the multiplica-

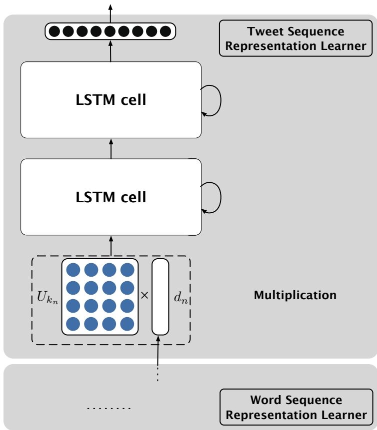

Figure 7: TSRL with User-modified RNNs (UR-TSRL)

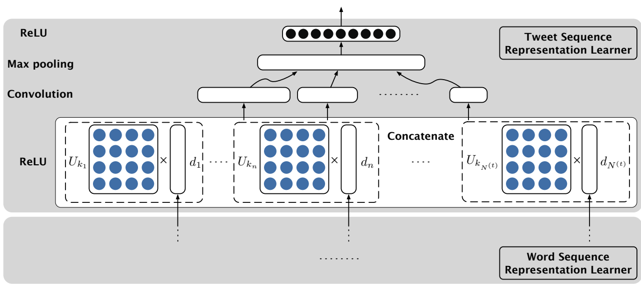

Figure 8: TSRL with User-modified CNNs (UC-TSRL)

tive composition of each word and the user. UR-TSRL first leverages U_{k_n} to conduct tensor product to project the original tweet representation d_n to the user sub-space, weighing its importance in the collective intelligence measurement. Following Tang et al. (2015), we apply a low-rank approximation to reduce the number of learning parameters instead of multiplying the user matrix to the tweet representation directly. The final nth input of the first layer LSTM cell, d_n^* , is calculated as follows,

$$d_n^* = U_{k_n} d_n \approx (U_{k_n,1} U_{k_n,2} + \mathrm{diag}(u')) d_n \tag{7}$$

UC-TSRL UR-TSRL explicitly constructs the representation for stock movement prediction by processing tweets in the strict temporal order. The modelling approach is intuitive but meanwhile, leads to a strong hypothesis on the information transparency and the tweet dependency. Nevertheless, it is not always reasonable to affirm the tweet dependency regarding time evidence. To address this problem, we further propose TSRL with user-modified CNNs (UC-TSRL) as an alternative. Different from UR-TSRL, UC-TSRL relaxes the assumption and adopts a convolution layer and a pooling layer, to learn the local feature patterns underlying the tweet representations at a more abstract level.

For comparative study, the RNNs and CNNs without user-modified inputs are denoted as RawR-TSRL and RawC-TSRL, respectively. and will also be experimented with in Section 5.

Objective Function

We first vectorise the labeled movement to the gold movement distribution using the one-hot coding scheme where in the two dimensions, the one corresponding to the ground truth class is 1 and the other is 0. After that, the cross entropy loss is adopted as the training objective function,

$$\text{loss} = -\sum_{i,j} p_j^{(i)} \log \hat{p}_j^{(i)} + \lambda \left\|\theta\right\|^2 \tag{8}$$

where $p_j^{(i)}$ and $\hat{p}_j^{(i)}$ denote the probability of the ith sample on the jth class under the gold movement distribution p and the predicted movement distribution $\hat{p}$, respectively. λ is the L_2-regularisation term and θ is the learning parameter set. We take the derivative of the loss with respect to θ through backpropagation for the update.

5 Experiments

Configurations for Model Training

We set the word embedding size and the stock embedding size to 50. Stock embeddings are initialised with their corresponding stock symbol word embeddings. For instance, the stock embedding for *$GOOG* is initialised with the "GOOG" in the word embedding table, The sequence length of historical prices is empirically set to 30 so the price information of the past one month is incorporated. As per the nature of tweets, we set the maximal length of the word sequence to 80 with the excess clipped. With regards to the CNNs component in StockNet, nine filters with three different filter sizes, 3, 4 and 5, are used. The values of the weight matrices between each layer are initialised with the fan-in trick. In the training phase, we apply the mini-batch training strategy and the Adam optimiser (Kingma and Ba 2014). We also use Dropout (Srivastava et al. 2014) to avoid overfitting and the effective batch normalisation trick (Ioffe and Szegedy 2015). Tensorflow (Abadi et al. 2016) is used to construct the computational graph of StockNet and hyperparameters are tweaked on the development set.

Evaluation Metrics

Following previous work for stock prediction (Xie et al. 2013; Ding et al. 2015), we adopt the standard measure of accuracy and Matthews Correlation Coefficient (MCC) as evaluation metrics. Given the confusion matrix $\begin{pmatrix} \mathrm{TP} & \mathrm{FN} \\ \mathrm{FP} & \mathrm{TN} \end{pmatrix}$ where TP, FP, TN and FN denote True Positive, False Positive, True Negative and False Negative, respectively, MCC is calculated as,

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{9}$$

Comparison with Benchmarks

For comparative study, we construct the following four benchmarks based upon textual information,

- LEXICON: counts sentiment words based upon lexicons (Oliveira, Cortez, and Areal 2013).

Model name	-WSRL	-TSRL
TECHNICALANALYST	None	None
RECURRENTNOVICEANALYST	Raw	RawR
CONVOLUTIONALNOVICEANALYST	Raw	RawC
RECURRENTATTENTIVEANALYST	Att	RawR
CONVOLUTIONALATTENTIVEANALYST	Att	RawC
RECURRENTADVANCEDANALYST	Bi	RawR
CONVOLUTIONALADVANCEDANALYST	Bi	RawC
SOCIALRECADVANCEDANALYST	Bi	UR
SOCIALCONVADVANCEDANALYST	Bi	UC
HEDGEFUNDANALYST	Bi	UC

Table 1: Different variations of StockNet. TECHNICALANALYST uses only the PSRL component. The difference between SOCIALCONVADVANCEDANALYST and HEDGEFUNDANALYST is that the latter uses the PSRL component. All other models do not make use of the PSRL component.

Model name	Acc.	MCC
LEXICON (Oliveira et al., 2013)	48.62	-0.014288
SVMS (Smailović et al., 2013)	51.51	0.003487
TSLDA (Nguyen and Shirai, 2015)	**54.40**	0.036947
WORD2VEC (Pagolu et al., 2016)	53.88	**0.051256**
SIMPLEMA5 (Brown 2004)	49.19	-0.020519
WEIGHTEDMA5 (Stevens 2002)	50.32	0.005315
TECHNICALANALYST	54.27	0.048793
RECURRENTNOVICEANALYST	51.91	-0.051612
CONVOLUTIONALNOVICEANALYST	**53.48**	**0.017966**
RECURRENTATTENTIVEANALYST	52.56	-0.002125
CONVOLUTIONALATTENTIVEANALYST	53.75	0.046281
RECURRENTADVANCEDANALYST	54.01	0.059130
CONVOLUTIONALADVANCEDANALYST	55.19	0.055979
SOCIALRECADVANCEDANALYST	54.80	0.058041
SOCIALCONVADVANCEDANALYST	55.32	0.063466
HEDGEFUNDANALYST	**56.64**	**0.084306**

Table 2: Models performances in accuracy and MCC.

- SVMS: classifies tweets into three sentiment categories with Support Vector Machines. (Smailović et al. 2013).
- TSLDA: an unsupervised topic model jointly learning topics and sentiments (Nguyen and Shirai 2015).
- WORD2VEC: uses Word2vec to represent texts for the Random Forest classifier (Pagolu et al. 2016).

We also build two moving average benchmarks using only price signals, SIMPLEMA5 (Brown 2004) and WEIGHTEDMA5 (Stevens 2002), for pure technical analysis.

To make a detailed analysis, we construct ten variations of StockNet, shown in Table 1. As shown in Table 2, HEDGEFUNDANALYST that integrates all text, metadata (stocks and users included) and historical price information, achieves the best results and outperforms the best baseline, the best novice analyst and TECHNICALANALYST, with 2.24, 3.16 and 2.37 in accuracy, 0.033050, 0.066340 and 0.035513 in MCC, respectively.

Generally, due to the involvement of stock information, models equipped with Att-WSRL and Bi-WSRL, such as CONVOLUTIONALATTENTIVEANALYST, RECURRENTADVANCEDANALYST and CONVOLUTIONALADVANCEDANALYST, obtain better performance than the novice ones such as RECURRENTNOVICEANALYST and CONVOLUTIONALNOVICEANALYST. Additionally, based upon Bi-WSRL, SOCIALRECADVANCEDANALYST with UR-TSRL and SOCIALCONVADVANCEDANALYST with UC-TSRL prove the effectiveness of user modification.

Effects of WSRL, TSRL and PSRL

As to models with the different WSRLs but the same TSRL, RECURRENTADVANCEDANALYST beats RECURRENTATTENTIVEANALYST, and CONVOLUTIONALADVANCEDANALYST beats CONVOLUTIONALATTENTIVEANALYST, indicating that Bi-WSRL yields better results than Att-WSRL. Although comparing to bidirectional LSTMs, the attention mechanism provides a more flexible approach to focus on the stock-relevant parts of word sequences, it however meanwhile brings far more learning parameters. Therefore, the result analysis is two-folded: (1) explicitly focusing on the preceding and following contexts of the stock symbols in tweets is effective enough to capture the stock-specific sentiment or semantic information precisely. The attention mechanism thus does not give extra functional credits; (2) bounded by the size of our training set, the optimisation for the learning parameters in Att-WSRL is possibly insufficient. Better results are expected when larger datasets are accessible.

Table 2 also shows the effectiveness of using the CNNs rather than the RNNs as TSRL. For instances, CONVOLUTIONALNOVICEANALYST and SOCIALCONVADVANCEDANALYST outperform RECURRENTNOVICEANALYST and SOCIALRECADVANCEDANALYST, respectively, regarding both accuracy and MCC. The results demonstrate that in both the settings with or without involving stock and user modifications, modelling tweet sequence in a temporally-strict approach that exaggerates the transparency and dependency between tweets, does not well-reflect the actual structure of social networks. Conversely, the convolution and pooling operations relax the tweet order restriction, better capturing the local patterns between adjacent tweets.

The performance of TECHNICALANALYST further confirms the positive effects from PSRL. Compared with classic technical analysis, TECHNICALANALYST learns from training data and incorporates more flexible non-linearity, outperforming both SIMPLEMA5 and WEIGHTEDMA5 .

Effects of Word Embeddings

As per previous studies, purely-random initialisation for learning parameters leads to unsatisfying performance (Kim 2014). Therefore, apart from the uniformly-random initialisation, we also experiment with two other initialisation strategies for word embeddings: initialisation with pre-trained Glove (Pennington, Socher, and Manning 2014) and with the sentiment-specific word embeddings (SSWE) including SSWE$_h$, SSWE$_r$ and SSWE$_u$ (Tang et al. 2014). As shown in Figure 9, Glove generates the best results

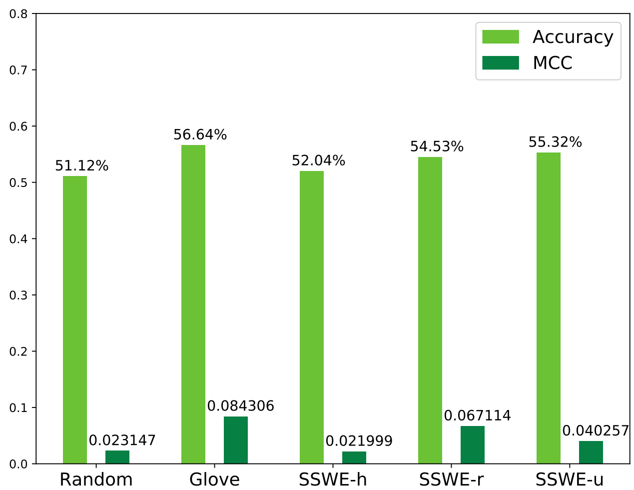

Figure 9: Model performances with different initialisation methods for word embeddings.

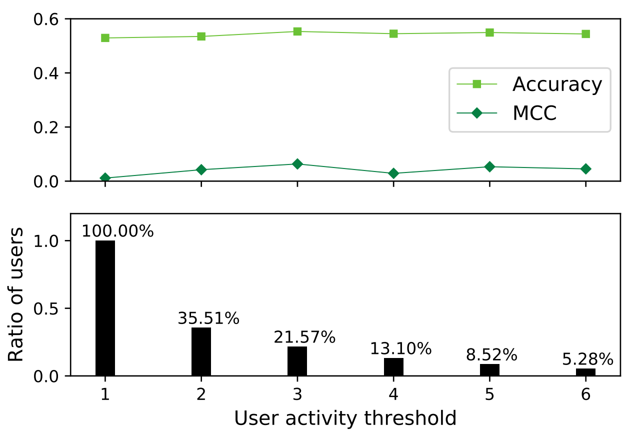

Figure 10: Model performances (above) and the ratio of eligible users (below) for different user activity threshold.

in both accuracy and MCC while SSWE$_h$ does not yield better results than random initialisation. One of the possible reasons is that the SSWE is trained with the distant-supervision method on a domain-independent corpus, thus lacks the capacity to precisely capture the sentiment underlying financial texts. Besides, the discriminative features for stock movement prediction should incorporate both objective news contents and subjective expressions, while the SSWE puts more emphasis on the later one. Instead, Glove pre-trained on 2 billion tweets leads to the most robust representation capacity in this research.

Effects of User Matrices

As mentioned in Section 4, we maintain a user list in user matrix learning. Following Tang et al. (2015), we filter users as per their activities that are the numbers of posted tweets. The main difference is that Tang et al. (2015) filter users in the test phase to select more predictable samples, while we are more interested in the filtering threshold for building the user list, which directly affects the training phase. The results of SocialConvAdvancedAnalyst with different user thresholds are shown in Figure 10. With the decreasing of the user list size by increasing the user activity threshold, both accuracy and MCC curves first ascend then descend, with the maximum achieved when the threshold equals to 3.

When the threshold is smaller, the model learns specific matrices even for exceptionally rare users. The matrices have a high possibility to be ill-trained and project the original word sequence representation into a noisy space.

Conversely, when we keep increasing the threshold after 3, the effects of user information diminish since most users are regarded as “UNK” for which we learn merely one unified user matrix (two if the approximation method is applied, according to Equation 7). Although minor oscillation is detected in MCC, the overall model performance tends to converge to ConvolutionalAdvancedAnalyst where no user information is involved. Based upon above evidence, we argue that the threshold of 3 provides the best trade-off in this research between incorporating user distinctions and keeping the generalisation capacity of StockNet.

Further Correlation Analysis

One more interesting question beyond the binary stock price movement is that will the prediction be easier for StockNet when the movement margin is larger? To evaluate the correlation, we compute the probability vector of inferencing the true movement label for test samples, $p^* = \mathrm{diag}(\hat{p}p^\intercal)$, and the corresponding movement margin vector q. The Pearson Correlation Coefficient (Pearson 1895) and the Spearman Rank Correlation Coefficient (Spearman 1904) are applied to measure the correlation between p^* and q, outputting the coefficients of 0.312192 and 0.546758, respectively. Consistent with our intuition, the results show a moderate positive relationship. For instance, due to more significant movement clues contained in tweets, the predictability of a shape nose dive caused by negative company news is generally considered higher than that of a small oscillation.

6 Conclusion and Future Work

We demonstrated the effectiveness of deep neural networks for stock movement prediction from social media data by introducing StockNet. Experimental results showed that compared with the benchmarks, StockNet can better identify the discriminative patterns for the prediction task, but its performance will be affected by the components and the initialisation methods. Additionally, we confirmed the existence of the optimum of the user filtering threshold and found the concrete value performing the best trade-off. We also showed the moderate positive relationship between the predictability and the movement margin.

We used stock symbols for tweet retrieval, which is efficient but its recall and precision can both be increased. Tweets simply reporting the current stock prices were retrieved, but those commenting on the breaking news about a company without referring its stock symbol were discarded. As future work, more sophisticated learning methods can be further investigated to provide better tweet resources.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Bollen, J.; Mao, H.; and Pepe, A. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *ICWSM* 11:450–453.

Brown, R. G. 2004. *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation.

Ding, X.; Zhang, Y.; Liu, T.; and Duan, J. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1415–1425.

Ding, X.; Zhang, Y.; Liu, T.; and Duan, J. 2015. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 2327–2333.

Dolan, R. J. 2002. Emotion, cognition, and behavior. *science* 298(5596):1191–1194.

Edwards, R. D.; Bassetti, W.; and Magee, J. 2007. *Technical analysis of stock trends*. CRC press.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1535–1545. Association for Computational Linguistics.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456.

Kahneman, D., and Tversky, A. 1979. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the econometric society* 263–291.

Kamijo, K.-i., and Tanigawa, T. 1990. Stock price pattern recognition-a recurrent neural network approach. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, 215–221. IEEE.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32*, 1188–1196. Beijing, China: JMLR. org.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

Nguyen, T. H., and Shirai, K. 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, 1354–1364.

Oliveira, N.; Cortez, P.; and Areal, N. 2013. Some experiments on modeling stock market behavior using investor sentiment analysis and posting volume from twitter. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, 31. Madrid, Spain: ACM.

Pagolu, V. S.; Reddy, K. N.; Panda, G.; and Majhi, B. 2016. Sentiment analysis of twitter data for predicting stock market movements. In *Proceedings of 2016 International Conference on Signal Processing, Communication, Power and Embedded System*, 1345–1350. Rajaseetapuram, India: IEEE.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2(1-2):1–135.

Pearson, K. 1895. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London* 58:240–242.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Rao, T., and Srivastava, S. 2012. Analyzing stock market movements using twitter sentiment analysis. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, 119–123. IEEE Computer Society.

Ritchie, J. C. 1996. *Fundamental analysis: a back-to-the-basics investment guide to selecting quality stocks*. Irwin Professional Pub.

Si, J.; Mukherjee, A.; Liu, B.; Li, Q.; Li, H.; and Deng, X. 2013. Exploiting topic based twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2, 24–29.

Smailović, J.; Grčar, M.; Lavrač, N.; and Žnidaršič, M. 2013. Predictive sentiment analysis of tweets: A stock market application. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer. 77–88.

Spearman, C. 1904. The proof and measurement of association between two things. *The American journal of psychology* 15(1):72–101.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15(1):1929–1958.

Stevens, L. 2002. *Essential technical analysis: tools and techniques to spot market trends*, volume 162. John Wiley & Sons.

Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; and Qin,

B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, 1555–1565.

Tang, D.; Qin, B.; Liu, T.; and Yang, Y. 2015. User modeling with neural network for review rating prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 1340–1346. AAAI Press.

Wang, Y.; Huang, M.; Zhao, L.; and Zhu, X. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 606615.

Xie, B.; Passonneau, R. J.; Wu, L.; and Creamer, G. G. 2013. Semantic frames to predict stock price movement. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, 873–883.