ORIGINAL RESEARCH

Leveraging temporal properties of news events for stock market prediction

Akira Yoshihara*[1], Kazuhiro Seki[2], Kuniaki Uehara[1]

[1] Graduate School of System Informatics, Kobe University, Kobe, Japan
[2] Faculty of Intelligence and Informatics, Konan University, Kobe, Japan

Received: August 18, 2015 **Accepted:** October 29, 2015 **Online Published:** November 23, 2015
DOI: 10.5430/air.v5n1p103 **URL:** http://dx.doi.org/10.5430/air.v5n1p103

ABSTRACT

Investors make decisions based on various factors, including consumer price index, price-earnings ratio, and also miscellaneous events reported by newspapers. In order to assist their decisions in a timely manner, many studies have been conducted to automatically analyze those information sources in the last decades. However, the majority of the efforts was made for utilizing numerical information, partly due to the difficulty to process natural language texts and to make sense of their temporal properties. This study sheds light on this problem by using deep learning, which has been attracting much attention in various areas of research including pattern mining and machine learning for its ability to automatically construct useful features from a large amount of data. Specifically, this study proposes an approach to market trend prediction based on a recurrent deep neural network to model temporal effects of past events. The validity of the proposed approach is demonstrated on the real-world data for ten Nikkei companies.

Key Words: Recurrent neural network, Deep learning, Natural language processing, Financial text mining

1. INTRODUCTION

To assist investors' decisions, machine learning approaches have long been studied to automatically analyze vast amounts of financial information, such as past stock prices. On the other hand, investors analyze and predict market trends based not only on such numerical information but also on miscellaneous events reported by newspapers. Thus, there have been also research efforts, such as Lavrenko's[1,2] and Schumaker's[3] studies, to leverage those information expressed in natural language. To the best of our knowledge, however, all the previous works have employed bag-of-words (or bag-of-ngrams at most), ignoring the context of words. Also, most of them do not consider "temporal effects" of past significant events. Stock prices continually change partly influenced by events happening in the world. Some events may affect stock prices for a short time period, and the others may have longer-term impact. For instance, when Lehman Brothers went into bankruptcy on September 15, 2008, many stock prices intermittently declined to the late October. Such temporal difference in event life time should be considered in predicting the market trends.

This study tackles these issues by adopting deep learning, which has been attracting much attentions in machine learning and pattern recognition communities among others for its exceptional performance as compared with the existing approaches for a number of pattern recognition and classification problems.[4,5] The high standard of the deep learning models is thought to come from its ability to hierarchically

*Correspondence: Akira Yoshihara; Email: yoshihara@ai.cs.kobe-u.ac.jp; Address: Graduate School of System Informatics, Kobe University, 1-1, Rokkodaicho, Nada-ku, Kobe City, Hyogo, Japan.

learn useful features from a large amount of data, which could be beneficial to learn the context of words when applied to natural language texts. In fact, the deep learning models have been also explored in natural language processing (NLP). For example, Socher[6] reported high accuracy for predicting the sentiment of a sentence by using a model extending the autoencoder, one of the important models of deep learning.

Among the existing models of deep learning, we employ a type of recurrent models, called Recurrent Neural Networks-Restricted Boltzmann Machine (RNN-RBM)[7] for dealing with the temporal effects mentioned above and combine it with Deep Belief Network (DBN). This study is the first attempt to adopt and analyze this particular model for market trend prediction or NLP at large.

2. Related work

This section first introduces previous work for estimating the trend of stock prices. Then, we summarize the fundamental models of deep learning and more advanced models that can be applied to time-series data.

2.1 Prediction of stock price trends through text analysis

There has been a great deal of research for predicting the trend of stock prices based on numerical information in the last decade by using machine learning techniques, such as support vector machines (SVM).[8–11] For example, Tay[11] determined input variables (features) of an SVM as lagged relative difference in percentage of price (RDP) values based on five-day periods. Their experiment showed that the SVM outperformed a back-propagation neural network. However, these models would be unreliable in predicting the movement of stock prices when significant events (*e.g.*, a bankruptcy of a global financial firm) happened since such events would first appear in the news, not in the past numerical information (*e.g.*, last five days' stock prices).

A natural approach would be to take advantage of news articles and a number of research efforts have been devoted in this direction.[1–3, 12–14] Performing linguistic analyses on such textual information may enable us to predict a sharp decline/surge of stock prices. For instance, Lavrenko[1, 2] predicted the trend of stock prices from money-market articles. First, he smoothed the time series of stock prices by piecewise linear regression[15] and defined each segment as a trend. Next, he clustered the segments based on features including the length and slope of a line segment. He regarded each article published within five hours prior to each trend as those affecting the trend and built language models from those articles. The language models were used to estimate the the likelihood of each trend (up and down) for immediate future on the evidence of the contents of an article in evaluation. A problem underlying these studies is the difficulty to identify useful features to represent natural language texts. Ding *et al.*[16] used an approach for Open Information Extraction (IE) and extracted as features various structured events from a large collection of news articles so as to predict S&P 500. They adopted a deep learning model for prediction and reported better performance than SVM. A limitation of these study is that they do not consider temporal properties, *i.e.*, the effects of past significant events.

We tackle the issue by employing a deep recurrent neural network, which would be beneficial to automatically identify useful features given a large amount of texts and to incorporate temporal properties.

2.2 Basic deep learning models

2.2.1 *Restricted Boltzmann Machines*

Restricted Boltzmann Machines (RBMs)[17] are one of the probabilistic deep learning models. An RBM defines joint distribution of visible and hidden layers, denoted as $\vec{v}$ and $\vec{h}$, respectively, through energy function E:

$$P(\vec{v},\vec{h}) = e^{-E(\vec{v},\vec{h})}/Z \tag{1}$$

$$E(\vec{v},\vec{h}) = -b_v^T\vec{v} - b_h^T\vec{h} - \vec{h}^T W\vec{v} \tag{2}$$

where W represents the weights matrix, b_v and b_h are bias terms of visible and hidden layers, respectively, and Z is a regularization term. Given $\vec{v}$, hidden units $h_i \in \{0,1\}$ are computed as follows:

$$P(h_i = 1|\vec{v}) = \sigma(b_{h_i} + W_i\vec{v}) \tag{3}$$

where $\sigma(\cdot)$ is a sigmoid function. Similarly, given $\vec{h}$, visible units $v_i \in \{0,1\}$ are computed as follows:

$$P(v_j = 1|\vec{h}) = \sigma(b_{v_j} + W_j^T\vec{h}). \tag{4}$$

Then, marginal probability $P(\vec{v})$ is computed with free-energy $F(\vec{v})$.

$$P(\vec{v}) = e^{-F(\vec{v})}/Z \tag{5}$$

$$F(\vec{v}) = -b_v^T\vec{v} - \sum_i \log(1 + e^{b_{h_i} + \vec{W}_i v}) \tag{6}$$

This model is trained by maximizing log likelihood of $P(\vec{v})$. Calculating the gradient of the log likelihood yields two terms, which are called positive and negative terms, respectively.

However, the computation of the negative term is computationally costly and and is often approximated by Gibbs sampling. This approximation method is called Contrastive Divergence.[18]

2.2.2 *Deep Belief Networks*

A DBN[19] is built up with a hierarchical stack of RBMs. Compared to an RBM, a DBN has a different representational ability owing to the connections between hidden layers.

Training of a DBN falls into two categories, pre-training and fine-tuning.[20] In pre-training, RBMs stacked hierarchically are divided individually, and each RBM conducts the aforementioned training. In fine-tuning, supervised learning is performed with parameters calculated in pre-training. First, a layer of logistic regression is added to a DBN as an output layer. This enables calculation of estimate values based on parameters trained. Second, an error between labels and the estimates are calculated and the output layer's parameters, W^o and $\vec{b}^o$, are updated so as to minimize the error. After that, the parameters of hidden layer $\vec{h}^3$, *i.e.*, W^3 and $\vec{b}^3$, are updated using W^o and $\vec{b}^o$. The parameters of the other layers are updated in the same manner.

2.3 Deep learning models for time-series data

2.3.1 *Recurrent Temporal Restricted Boltzmann Machine*

When dealing with time-series data, Recurrent Temporal Restricted Boltzmann Machine (RTRBM) [21] is constructed recurrently. This supposes hidden units $\vec{h}_t$ on a given time t represent temporal alteration before t and makes it possible to consider only connection to a hidden layer on a former time step. Consequently, there is low calculation cost of conditional probability used in sampling. On time t, conditional joint distribution of $\vec{v}_t$ and $\vec{h}_t$ given $\vec{h}_{t-1}$ is represented as follows:

$$P(\boldsymbol{v}_t, \boldsymbol{h}_t | \boldsymbol{h}_{t-1}) = \frac{\exp\left(\boldsymbol{v}_t^T b_v + \boldsymbol{h}_t^T W \boldsymbol{v}_t + \boldsymbol{h}_t^T (b_h + W' \boldsymbol{h}_{t-1})\right)}{Z(\boldsymbol{h}_{t-1})} \tag{8}$$

where b_v, b_h, W equal to Eq.(2), and W' represents the weights connecting $\vec{h}_{t-1}$ and $\vec{h}_t$. Joint distribution on RTRBM is represented using the above equation as follows:

$$\begin{aligned} P(v_1^T, h_1^T) &= \prod_{t=1}^{T} \sum_{h_t'} P(\vec{v}_t, \vec{h}_t' | \vec{h}_{t-1}) P(\vec{h}_t | \vec{v}_t, \vec{h}_{t-1}) \\ &= \prod_{t=1}^{T} P(\vec{v}_t | \vec{h}_{t-1}) P(\vec{h}_t | \vec{v}_t, \vec{h}_{t-1}). \end{aligned} \tag{9}$$

3. Predicting stock price trends considering textual temporal properties

The aim of this study is to predict the trend of stock prices, which is influenced by miscellaneous events happening around the world. We adopt the RNN-RBM so as to deal with the temporal effect of past events with a long-term impact on stock prices. RNN-RBM is a deep learning model considering time-series data and an extension of Recurrent Temporal RTRBM summarized in Section 2. The next sections provide the overview of RNN-RBM and detail its application to market trend prediction by incorporating RNN-RBM into DBN.

3.1 Recurrent Neural Networks-Restricted Boltzmann Machine

As mentioned in the previous section, RTRBM has a lower calculation cost but imposes a restriction that a hidden layer for time t always affects another hidden layer for the succeeding time $t+1$. However, not all events have a long-term effect and the effect would not be constant for each time step. For example, articles about the bankruptcy of Lehman Brothers had a long-term impact on stock prices and, by contrast, those about the stock market typically have a short-term effect. If we use RTRBM for predicting market trends, every news article always affects stock prices in the following days, disregarding the possibly different event life time. RNN-RBM extends RTRBM to resolve such problems.

The graphical model of RNN-RBM is shown in Figure 1. RNN-RBM consists of an RBM with three parameters, $W, \vec{b}_h^{(t)}, \vec{b}_v^{(t)}$, and an RNN with five parameters, $W', W'', W_2, W_3, \vec{\hat{h}}^{(t)}$. Adding a hidden layer different from those of RBM discriminates between hidden units representing temporal alteration and visible units, which resolves the aforementioned problem.

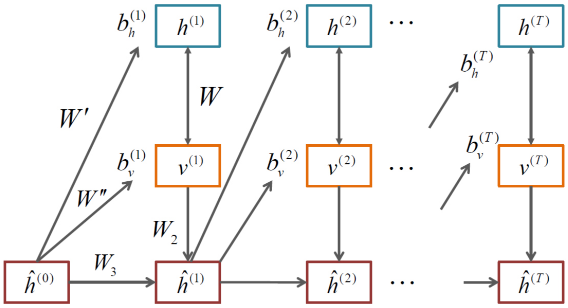

Figure 1. A graphical structure of an RNN-RBM

For simplicity, we consider a single-layer RNN-RBM. Hidden units of this model is given by

$$\vec{\hat{h}}^{(t)} = \sigma(W_2 v^{(\vec{t})} + W_3 \vec{\hat{h}}^{(t-1)} + \vec{b}_{\hat{h}}). \tag{10}$$

Additionally, bias terms, $\vec{b}_h^{(t)}, \vec{b}_v^{(t)}$, are given as follows:

$$\vec{b}_h^{(t)} = \vec{b}_h + W' \vec{\hat{h}}^{(t-1)} \tag{11}$$

$$\vec{b}_v^{(t)} = \vec{b}_v + W'' \vec{\hat{h}}^{(t-1)}. \tag{12}$$

These parameters are used for training of this model, which is performed in two steps: First, we calculate a derivative at the RBM parameters using the contrastive divergence (CD) method. Next, we derive at the RNN parameters by applying Back-propagation Through Time (BPTT) algorithm.[22]

3.2 Incorporating RNN-RBM into DBN

Figure 2 depicts the proposed approach, which uses a model extending a DBN. There are two reasons why we choose a DBN. First, a DBN has multi-layer structure. In deep learning, it is generally considered that increasing the number of layers enhances the expression power of a model. Second, RNN-RBM is a model extending an RBM. Since DBN consists of hierarchically stacked RBMs, we could easily extend an RBM of a DBN to RNN-RBM. This extension alters a DBN into a model having multi-layer structure suitable for time-series data with long-term effects.

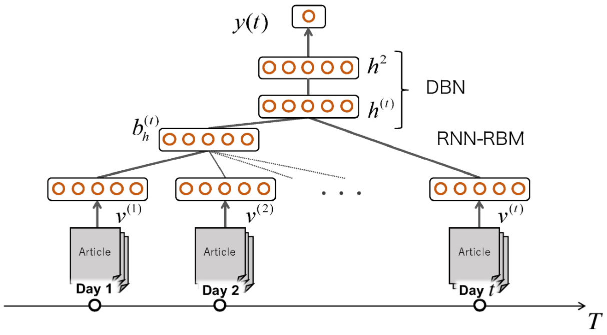

Figure 2. A graphical structure of RNN-RBM + DBN

As mentioned, this approach substitutes RNN-RBM for RBM, which is composed of an input layer $\vec{v}$ and a hidden layer $\vec{h}^1$. By this substitution, we expect that one could obtain latent depictions considering temporal alteration of input. The number of dimensions of the output layer is one because this study treats the prediction of stock prices as a binary classification problem, *i.e.*, uptrend/downtrend.

The training algorithm is similar to DBN. In pre-training, each parameter is computed by the above training algorithm of an RBM and RNN-RBM. In fine-tuning, those parameters are updated by back-propagation.

3.3 Features

In this work, it is crucial to model temporal properties of news events expressed in natural language which have short- or long-term influences on stock prices. We first group news articles per day and each article is represented by the standard bag-of-words model. The resulting word vectors are used as the input of our model. Note that an element of the word vectors is either 1 or 0, representing whether a term appeared in the document or not, respectively.

For instance, suppose that two news articles, "Lehman Brothers declared bankruptcy" and "NEW YORK–Oil prices dropped sharply to 91 dollars a barrel", were issued on the same day. We concatenate the articles and extract nouns and verbs through part-of-speech tagging, resulting in a term set, {lehman, brothers, declared, bankruptcy, new, york, oil, prices, dropped, 91, dollars, barrel}. These terms are coded as "1" in a word vector, *e.g.*, {1,1,0, . . .,0,1}, where each element of the vector represents the exsistence of a certain term. This step is applied to news articles for each day to produce input data. More details of the data processing are explained in Section 4.2.

Next, we describe how to treat the model output. As mentioned, our model predicts the trend of stock prices of the next day to be either upward or downward (*i.e.*, binary classification). To be precise, output y is defined as

$$y = f\left(\sigma(\vec{W}^o\vec{h}^2 + b^o)\right) \qquad (13)$$

$$f(x) = \begin{cases} 1 & \text{if } x > 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

where $\vec{h}^2$ represents the output of the top layer of the DBN and $\vec{W}^o$ and b^o are the parameters of the layer. The output values "1" and "0" represent uptrend and downtrend, respectively.

3.4 Overall procedure

This section provides an overall, step-by-step procedure for applying our proposed model described above so as to make market trend prediction.

(1) Grouping news articles:

All the news articles published in a day are concatenated to create a (pseudo) large document d covering all the events for the day. This step is repeated for each day.

(2) Modeling news articles:

A. The vocabulary V is determined by applying a morphological analyzer followed by statistical tests. The details of this step will be presented in Section 4.2.

B. Each document d is transformed into a vector, whose elements have 1/0 values indicating presence/absence, respectively, of a certain term in the fixed vocabulary V.

(3) Creating a dataset:

A. For each day, the trend of the stock price (up or down) is determined.

B. The corresponding market trends and the concatenated news articles d's are paired together.

C. The data created in the previous step are split into training, validation, and test data.

(4) Building RNN-RBM+DBN:

A. Using the training data from 3 (C), a DBN[19] composed of two RBMs[17] (denoted as RBM$_1$ and RBM$_2$) and an output layer (h^2 in Figure 2) is built.

B. RBM$_1$ of the DBN is replaced by RNN-RBM,[7] which in part becomes $h^{(t)}$ in Figure 2.

C. The number of the units in hidden layers ($i.e., \hat{h}; h^{(t)}; h^2$) is determined on the validation data from 3 (C).

(5) Tuning RNN-RBM+DBN:

The parameters are updated in two learning stages, namely, pre-training and fine-tuning,[20] on the training data.

4. Evaluation

We evaluated the effectiveness of our model for stock price trend prediction from two different viewpoints. One was the performance of binary classification for stock price trends. The other was the profit of trading simulation, where buying and selling decision was made based on the trend classification.

4.1 Experimental settings

We used the morning edition of the Nikkei newspapers published from 1999 to 2008 for experiments, where the training data were 834,882 articles during eight years (1999–2006), the validation data were 98,667 articles in 2007, and the test data were 99,728 in 2008. The target brands were Nikkei Stock Average and ten Nikkei companies whose names most frequently appeared in the whole data.

We computed Moving Average Convergence Divergence (MACD: a trading indicator often used in a technical analysis of stock prices) for each day and predicted the next day's MACD by our proposed model using the current day's news articles as input.

Table 1 summarizes the distribution of the uptrend/downtrend cases in the training, validation, and testing data. The number of instances in each data set is 1,894, 236, and 236, respectively. Based on the preliminary experiments on the validation data, the number of the units in hidden layers $\vec{h}^{(t)}$, $\vec{h}^2$, and $\vec{\hat{h}}$ was set to 3,750, 2,500, and 200, respectively.

Table 1. The distribution of the uptrend/downtrend cases in training, validation, and testing data

Brands	**Training (1/0)**	**Validation (1/0)**	**Test (1/0)**
Nikkei Average	0.51/0.49	0.49/0.51	0.50/0.50
Hitachi	0.39/0.61	0.37/0.63	0.37/0.63
Toshiba	0.37/0.63	0.42/0.58	0.40/0.60
Fujitsu	0.41/0.59	0.42/0.58	0.42/0.58
Sharp	0.44/0.56	0.45/0.55	0.48/0.52
Sony	0.50/0.50	0.47/0.53	0.49/0.51
Nissan Motor	0.40/0.60	0.46/0.54	0.45/0.55
Toyota Motor	0.48/0.52	0.45/0.55	0.48/0.52
Canon	0.48/0.52	0.42/0.58	0.47/0.53
Mitsui	0.40/0.60	0.48/0.52	0.48/0.52
Mitsubishi	0.43/0.57	0.44/0.56	0.49/0.51

The proposed model was implemented in Theano (`http://deeplearning.net/software/theano`) a Python library that makes it possible to define and evaluate mathematical expressions involving multi-dimensional arrays efficiently. We trained this model on an NVIDIA Tesla K40 GPU.

4.2 Preprocessing

In order to transform news articles into word vectors, we first determined the vocabularies (a term set or features) through a morphological analysis and statistical tests for independence. We utilized MeCab[23] as a morphological analyzer where Wikipedia entries and Nihon Keizai Shinbun's keywords were added to the dictionary of the analyzer to deal with possibly specialized vocabularies.

Table 2. Cross table for computing chi-square scores

	Uptrend (> 1%)	**Downtrend (> 1%)**	**Neutral**	**Sum**
Appear	U_{w+}	D_{w+}	N_{w+}	A_{w+}
Not appear	U_{w-}	D_{w-}	N_{w-}	A_{w-}
Sum	U	D	N	A

Considering computation time, we fixed the number of terms used in our experiments to 5,000, whose chi-square scores were the highest among the terms appeared in the training data set. Chi-square scores were computed for each term and each brand in Nikkei 225 with respect to three possible directions of trend, *i.e.*, uptrend (over 1%), downtrend (over 1%), and neutral. Table 2 shows the cross table, where U_{w+} (U_{w-}) denote the number of days when a term in question appeared (did not appear) in news articles during the days of uptrend. Similarly, N_{w+} (N_{w-}) are the number of days when a term in question appeared (did not appear) during the downtrend.

The two chi-square scores, $\chi^2_{uptrend}, \chi^2_{downtrend}$, were calculated for each term w for each brand as in Eq. (15) and Eq. (16).

$$\chi^2_{uptrend} = \frac{(|U_{w+} \cdot (D_{w-} + N_{w-}) - U_{w-} \cdot (D_{w+} + N_{w+})| - \frac{A}{2})^2 \cdot A}{A_{w+} \cdot A_{w-} \cdot U \cdot D} \tag{15}$$

$$\chi^2_{downtrend} = \frac{(|D_{w+} \cdot (U_{w-} + N_{w-}) - D_{w-} \cdot (U_{w+} + N_{w+})| - \frac{A}{2})^2 \cdot A}{A_{w+} \cdot A_{w-} \cdot U \cdot D} \tag{16}$$

Then, terms were sorted in descending order of the scores where $\chi^2_{uptrend}$ and $\chi^2_{downtrend}$ were not distinguished, and 5,000 unique terms from the top were identified as the vocabularies.

4.3 Empirical results

The performance (error rates) for predicting the market trend on the test data is presented in Table 3, where “baseline” simply chose the majority class between uptrend and downtrend, SVM used a linear kernel with parameter $C = 0.0001$, DBN is a deep learning model not considering temporal properties, and RNN-RBM+DBN is our model. For DBN, the number of units in hidden layers $\vec{h}^1$, $\vec{h}^2$ were also optimized using the validation data.

Table 3. Test error rates for stock price prediction

Brands	Baseline	SVM	DBN	RNN-RBM + DBN
Nikkei Average	49.57	48.73	45.50	**43.62**
Hitachi	35.71	37.29	32.00	**32.00**
Toshiba	39.52	41.95	**38.50**	**38.50**
Fujitsu	40.00	40.25	**32.00**	34.00
Sharp	42.00	47.88	**40.00**	**40.00**
Sony	43.00	47.46	41.43	**40.95**
Nissan Motor	40.00	45.34	39.50	**37.00**
Toyota Motor	44.29	53.39	43.81	**42.38**
Canon	43.81	53.39	43.00	**39.11**
Mitsui	46.96	47.88	**41.43**	**41.43**
Mitsubishi	43.81	49.15	43.33	**40.43**
Average	42.61	46.61	40.05	**39.04**

Comparing the baseline and SVM with DBN, the error rate reduces from 42.60% and 47.30% to 40.05% on average and the differences were found statistically significant ($p < .01$). The result indicates the effectiveness of the general deep learning model in the target problem. In addition, our proposed approach, RNN-RBM+DBN, produced the lowest error rate for most of the brands (including some ties) and consequently further reduced the error rate to 39.04%. Although this difference was not significant ($p = .076$), it may suggest the importance of considering the past events as we will discuss in the following section.

Here, it should be mentioned that the error rates of the baseline are slightly different from Table 1. This is due to that input data were divided into mini batches and some data not included in any batches were ignored in evaluation.

Table 4. Profits/losses of trading simulation in Japanese yen

Brands	SVM	RNN-RBM + DBN
Hitachi	-2965	**-475**
Toshiba	-1583	**-880**
Fujitsu	-188	**-71**
Sharp	-479	**3**
Sony	-40	**29**
Nissan Motor	-150	**-38**
Toyota Motor	-787	**111**
Canon	-887	**209**
Mitsui	-658	**-87**
Mitsubishi	-1680	**-305**
Total	-9417	**-1504**

Next, we report on the results of trading simulation. The experiment was conducted on the test data, *i.e.*, for the year 2008. The trading decision was made based on the stock price trend prediction above. Specifically, if the previous day’s prediction was “downward” and the current day’s prediction was “upward”, the decision was “buy”, and *vice versa*. We looked at two consecutive days’ predictions because we were predicting not the trend of stock prices but the trend of MACD. Since MACD represents the trend of stock prices by averaging them, one could expect greater profits by focusing on its change point. The results are summarized in Table 4 comparing SVM and our model. Note that we did not consider intermediary fees for dealing stocks. Also note that since the unit of trading was fixed to one, the absolute profit/loss was relatively small.

Even though the total performance was turned out to be negative for both models, our model was better than SVM for all brands. A possible reason for the negative results is that there was an overall downward trend in 2008. Because it is difficult to gain a large profit in a stock market in such trend without share holdings, it is practically important to minimize the loss.

5. Discussion

While the error rate of RNN-RBM+DBN was lower than that of DBN, the difference was not statistically significant. We conjecture that the inconclusive result was obtained due to the small number of significant events having long-term effects for the period corresponding to the test data. In other words, if there are not many such significant events, it is natural that there is not much difference between DBN and RNN-RBM+DBN.

Therefore, we carried out another experiment focusing on a shorter period in which a known significant event actually occurred. Specifically, we focused on the bankruptcy of Lehman Brothers and predicted the market trend between September 15 (when the bankruptcy was reported) to October 28, 2008. The results of the experiment are shown in Table 5.

Table 5. Comparison of test error rates after a significant financial crisis

Brands	SVM	RNN-RBM + DBN
Nikkei Average	51.61	**38.70**
Hitachi	61.29	**32.25**
Toshiba	54.83	**38.70**
Fujitsu	45.16	**32.25**
Sharp	58.06	**45.16**
Sony	**41.93**	**41.93**
Nissan Motor	**29.03**	35.48
Toyota Motor	48.38	**45.16**
Canon	**54.83**	**54.83**
Mitsui	41.93	**38.70**
Mitsubishi	29.03	**25.80**
Average	46.92	**39.00**

Similar to the results in Table 3, our approach, RNN-RBM+DBN, showed the lowest error rates for many brands but this time the improvement of our approach over DBN is more apparent. The average error rate was found to be 39.00% as compared with 46.92% of DBN; the difference was statistically significant at the 5% significance level ($p = .025$). These results suggest the capability of RNN-RBM+DBN to consider temporal properties of significant past events which have long-term effects on stock prices.

Here, it should be noted that even though the exact event of Lehman Brothers' bankruptcy does not exist in the training or validation data, our model was capable of capturing the effect of the unknown event appearing for the first time in the test data. This is assumedly due to the feature selection process. We selected features (terms) as described in Section 4.2 and the selected features do include "Lehman", "bankruptcy", *etc*. When Lehman Brothers failed, "Lehman" and "bankruptcy" were often used together, which formed a word vector coding their co-existence and led to the correct prediction in many cases.

6. Conclusion

This paper proposed an approach to predicting the trend of stock prices by focusing on news events with long-term effects. To consider such events, we employed a deep learning model, specifically, a combination of RNN-RBM, a recurrent model typically used for time-series data, and DBN composed of hierarchically stacked RBMs. The input data of the model are news articles represented as word vectors by the bag-of-words representation.

To evaluate the effectiveness of the approach, we conducted experiments on 10 years' worth of Nikkei newspaper articles, where those published between 1999 and 2007 were used for training and tuning model parameters and those in 2008 were used for testing. We chose the Nikkei Stock Average and ten brands whose names appeared most frequently in the newspaper and predicted the trend of their stock prices to be uptrend or downtrend (*i.e.*, binary classification). The results were compared with SVM and DBN (which do not consider past events) along with a simple baseline choosing the majority class. On average, our proposed approach showed the lowest error rate and the improvement was statistically significant except when compared with DBN. We looked into the insignificant case (*i.e.*, our approach *vs.* DBN) and found that if we focused on a certain period after a known significant event, the improvement of our approach over DBN also became more apparent. In addition, we reported on the dealing simulation based on the trend prediction made by our model. Even though the total performance was negative, our model consistently showed less losses than SVM. These results indicate the effectiveness of the deep learning models in general in the financial domain and also suggest the potential of the recurrent model to capture the properties of past significant events with long-term effects on the stock market.

Acknowledgements

This work is partially supported by JSPS KAKENHI Grant Numbers 25330363 and MEXT, Japan.

References

[1] Lavrenko V, Schmill M, Lawrie D, *et al*. Language models for financial news recommendation. In: Proceedings of the ninth international conference on Information and knowledge management; 2000. p. 389-96.

[2] Lavrenko V, Schmill M, *et al*. Mining of concurrent text and time series. In: Proceedings of the KDD-2000 Workshop on Text Mining; 2000. p. 37-44.

[3] Schumaker RP, Chen H. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS). 2009; 27(2): 12-9.

[4] Dahl GE, Dong Y, Li D, *et al*. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing. 2012; 20(1): 30-42. http://dx.doi.org/10.1109/TASL.2011.2134090

[5] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proceedings of the twentyfifth international conference on Neural Information Processing Systems; 2012. p. 1106-14.

[6] Socher R, Pennington J, Huang EH, *et al*. Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the sixteenth Conference on Empirical Methods in Natural Language Processing; 2011. p. 151-61.

[7] Nicolas BL, Bengio Y, Vincent P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In: Proceedings of the twenty-ninth International Conference on Machine Learning; 2012. p. 1159-66.

[8] Huang W, Nakamori Y, Wang SY. Forecasting stock market movement direction with support vector machine. Computers & Operations Research. 2005; 32(10): 2513-22.

[9] Kim KJ. Financial time series forecasting using support vector machines. Neurocomputing. 2003; 55(1): 307-19.

[10] Shin KS, Lee TS, Kim HJ. An application of support vector machines in bankruptcy prediction model. Expert Systems with Applications. 2005; 28(1): 127-35. http://dx.doi.org/10.1016/j.eswa.2004.08.009

[11] Tay FEH, Cao LJ. Application of support vector machines in nancial time series forecasting. Omega. 2001; 29(4): 309-17. http://dx.doi.org/10.1016/S0305-0483(01)00026-3

[12] Gidofalvi G, Elkan C. Using news articles to predict stock price movements. Department of Computer Science and Engineering, University of California, San Diego, 2001.

[13] Izumi K, Goto T, Matsui T. Trading tests of long-term market forecast by text mining. In: Proceedings of the tenth IEEE International Conference on Data Mining Workshops; 2010. p. 935-42.

[14] Mittermayer MA. Forecasting intraday stock price trends with text mining techniques. In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, pages 10.pp. IEEE, 2004.

[15] Pavlidis T, Horowitz S. Segmentation of plane curves. IEEE transactions on Computers. 1974; 23(8): 860-70.

[16] Ding X, Zhang Y, Liu T, *et al*. Using structured events to predict stock price movement: An empirical investigation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. Association for Computational Linguistics; 2014. p. 1415-25.

[17] Smolensky P. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory. MIT Press; 1986. p. 194-281.

[18] Hinton GE. Training products of experts by minimizing contrastive divergence. Neural computation. 2002; 14(8): 1771-800. PMid:12180402. http://dx.doi.org/10.1162/089976602760128018

[19] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural computation. 2006; 18(7): 1527-54. PMid:16764513. http://dx.doi.org/10.1162/neco.2006.18.7.1527

[20] Bengio Y, Lamblin P, Popovici D, *et al*. Greedy layer-wise training of deep networks. In: Proceedings of the twenty-first international conference on Neural Information Processing Systems; 2007. p. 153-60.

[21] Sutskever I, Georey EH, Taylor GW. The recurrent temporal restricted boltzmann machine. In: Proceedings of the twenty-second international conference on Neural Information Processing Systems; 2008. p. 1601-8.

[22] Werbos PJ. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE. 1990; 78(10): 1550-60. http://dx.doi.org/10.1109/5.58337

[23] Kudo T. Mecab: Yet another part-of-speech and morphological analyzer. 2005. Available from: http://mecab.sourceforge.net/.