Predicting Stock Price Movement with Event-Driven Deep Learning Approach

Yuan Fang, Shelly Wang, Yitian Zeng. <yuanfy, shelly29, ytzeng1>@stanford.edu

Link to presentation: https://youtu.be/EowN_DrDjIk

Motivation and Problem

Predicting stock prices is of central interest to investment professionals as well as academic researchers. One central tenet in financial theory—the Efficient Market Hypothesis—states that, at any given point in time, stock prices reflect all available information; prices only change whenever new information becomes available. This theory thus offers theoretical support for the value of news as a lucrative source of signals, with the recent technological developments only fueling the attention towards financial news.

Traditional NLP approaches rely on simple features such as bag-of-words and noun phrases from text documents to predict stock-related quantities; such techniques fail to capture the structured relations within texts, as initiators of actions are not differentiated from their subjects. In this project, we aim to build a non-traditional NLP model that predicts the directions of stock price movements with financial news headlines.

Data

Input: Financial news headlines from Reuters and Bloomberg from Oct 2006 to Nov 2013, released by Ding et al. [2014]. The average numbers of headlines published during each business day and weekend day are around 500 and 20, respectively. Since RNN requires datasets to have same number of time steps in each batch, and there was no corresponding stock price during weekends, headlines published on weekends were not used in RNN models.

Output: Price movements of the Standard & Poor's 500 index from the same period were obtained from the quantmod R package.

Preprocessing

Embedding sentences into vectors while still preserving useful information is the core of NLP. We experiment with four embedding methods in this project.

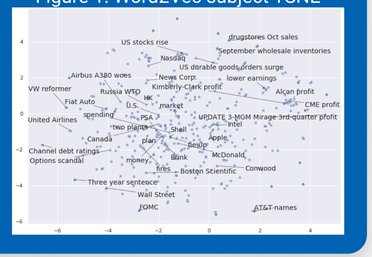
Word2Vec: We first use open information extraction to obtain structured event representations of headlines. A pre-trained 300-parameter Word2Vec model then codes event tuples into vectors. Figure 1 shows the top 1000 results of applying t-distributed Stochastic Neighbor Embedding to the subject fields of event tuples to obtain corresponding 2D vectors.

Bidirectional Encoder Representations from Transformers: Designed to pre-train deep directional representations from unlabeled text, BERT maps each headline to a 768-dimensional vector.

Smooth Inverse Frequency: SIF, mapping each headline to a 300-dimensional vector, involves computing weighted average of word vectors in the sentence and removing projections of the average vectors on their first singular vector.

Universal Sentence Encoder: An embedding method that targets transfer learning to other NLP tasks, USE maps each headline to a 512-dimensional vector.

Figure 1: Word2Vec subject TSNE



Methods

Convolutional Neural Network

The first CNN model we tried (CNN1) takes a sequence of word embeddings, consisting of long-term, mid-term, and short-term words embeddings (corresponding to the past 30 days, 7 days, and 1 day). The long-term and mid-term words embeddings are fed into a 1D convolutional layer (filter size f=3, stride s=1, padding p=same) and then a max pooling layer to obtain Vl and Vm, respectively. The model performs an average pooling on short-term words to get Vs, and it then feeds Vc= (Vl, Vm, Vs) into a hidden layer with 100 neurons, which is the second last layer. Then, given the information of the hidden layer, the output layer outputs either 1 or 0, signifying price increase or decrease, respectively. However, CNN1 performs poorly regardless of the embedding method. The best test accuracy, from SIF embeddings, was 0.46.

Embedding	Accuracy	Word2Vec	BERT	SIF	USE
CNN2	Training	0.55	0.55	0.58	0.56
	Test	0.56	0.48	0.59	0.57

Table 1: CNN2 results

Confusion matrix	Predicted: 0	Predicted: 1
Actual: 0	2	103
Actual: 1	1	145

Table 2: Confusion matrix of CNN2 with SIF embedding

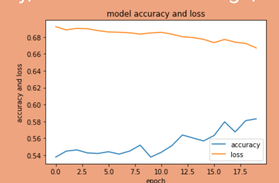


Figure 2: Training accuracy and loss for CNN2 with SIF embedding

Due to the poor performance of CNN1, we adjusted the architecture and got our second CNN model (CNN2). The input to CNN2 is a list of word embeddings of the past 20 days. CNN2 consists of a 2D convolutional layer (filter size (f1,f2)=(3,size of word embeddings), strides=1, padding p=valid, channels=64), a 2D average pooling layer with pool size=(3,1), two hidden layers with dropout=0.4 and 0.2, respectively, and an output layer which outputs the probabilities of price increase and decrease. We use different word embeddings to train and test the model, and the result is in Table 1. We got the best test accuracy of 0.59 with SIF embeddings; corresponding confusion matrix and "accuracy and loss" graph are shown in Figure 2 and Table 2. The accuracy measures of CNN1 and CNN2 show that CNN models able to learn the connection between news headlines and stock price.

Recurrent Neural Network

Long short-term memory (LSTM) and gated recurrent unit (GRU) are two major building blocks of recurrent neural networks. We tried a variety of combinations and our best performaing model is illustrated in Figure 3. The input shape of the model (10, 500, 300) is equivalent to 10 consecutive days, 500 news headlines each day with 300 features of each headline. Each day is fed into a bidirectional LSTM cell with increasing number of hidden units from day 1 to day 10. The output of each LSTM cell is flatten out and stacked into another 3D dataset and fed into a new LSTM layer and finally connected to a fully connected layer and output as binary classification problem.

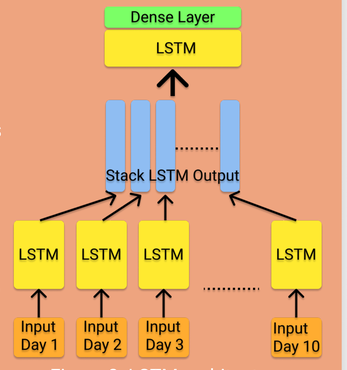


Figure 3: LSTM architecture

Baseline Method

Several baselines were established due to the breadth of models investigated in this study. A simple logistic regression achieved a training accuracy of 0.69 and test accuracy of 0.58. Test results of one layer of different RNN models on different embeddings is summerized in Table 3.

Embedding	Accuracy	LSTM	Bi-LSTM	GRU	Bi-GRU	cnn-LSTM
BERT	Training	0.67	0.55	0.71	0.75	0.52
	Test	0.59	0.46	0.59	0.38	0.59
SIF	Training	0.79	0.55	0.86	0.54	0.54
	Test	0.56	0.59	0.60	0.59	0.59
USE	Training	0.83	0.55	0.83	0.54	0.55
	Test	0.56	0.59	0.56	0.59	0.59

Table 3: Results of one layer of various RNN models

Results and Discussion

The results of our models show that there is a close relationship between financial news and stock price. The test accuracy of our logistic regression model is 0.58, and we assume the accuracy is nothigh mainly due to that it is a fairly simple model. The best test accuracy of CNN1 is 0.46 and the best word embeddings for CNN1 is SIF; we believe that CNN1 performs poorly because redundant information in the input distracts CNN1 from making good predictions. We thus came up with CNN2, which only uses the past 20 days' financial news as input, instead of using long-term, mid-term, short-term news as for CNN1. The best test accuracy of CNN2 is 0.59, given SIF word embeddings as input. As we can see from the confusion matrix and the accuracy and loss graph (Table 2; Figure 2), CNN2 does learn from the training data; however, the trained CNN2 tends to predict that the stock price as increasing, and this could be due to a lack of cleaned training data.

For RNN models, our best model achieved a training accuracy of 0.56 with regularization and a test precision of 0.64 for positive class and 0.49 for negative class. The performance of this model is on par with published work by Ding et al. [2015], and shows that RNN is able to learn the connection between financial news and stock price to some degree. However, with such large amount of trainable parameters in our model, our training data is not sufficient and we are constantly overfitting the model and yielding less satisfactory results. Accumulating a larger dataset would alleviate this issue.

Conclusion and Next Steps

The highest test accuracy we have is comparable to the performance of the model obtained by Dinget al. [2015].
There are three steps we can take to further improve performance:
1. Remove irrelevant data from and expand our dataset;
2. Experiment with more complex model archirectures, possibly combining CNN2 and GRU;
3. Further tune hyperparameters and input features to train aggressive and preservative models.

References and Links

X. Ding, Y. Zhang, T. Liu, and J. Duan. Using structured events to predict stock price movement: Anempirical investigation. InProceedings of the 2014 Conference on Empirical Methods in NaturalLanguage Processing (EMNLP), pages 1415–1425, 2014.
X. Ding, Y. Zhang, T. Liu, and J. Duan. Deep learning for event-driven stock prediction. InTwenty-fourth international joint conference on artificial intelligence, 2015.

GitHub: http://bit.ly/stockpricedl
YouTube: https://youtu.be/EowN_DrDjIk