Stock Price Analysis Using Combination of Analyst Reports and Several Documents

Masahiro Suzuki
Faculty of Engineering
The University of Tokyo
Tokyo, Japan
msuzuki@g.ecc.u-tokyo.ac.jp

Toshiya Katagi
School of Engineering
The University of Tokyo
Tokyo, Japan
m2017tkatagi@socsim.org

Hiroki Sakaji
School of Engineering
The University of Tokyo
Tokyo, Japan
sakaji@sys.t.u-tokyo.ac.jp

Kiyoshi Izumi
School of Engineering
The University of Tokyo
Tokyo, Japan
izumi@sys.t.u-tokyo.ac.jp

Yasushi Ishikawa
Nikko Asset Management Co., Ltd.
Tokyo, Japan
Yasushi.Ishikawa@nikkoam.com

***Abstract*—In this paper, we propose a methodology of forecasting the direction and extent of volatility in mid- to long-term excess returns of stock prices by applying natural language processing and neural networks in the context of analyst reports. Analyst reports are prepared by analysts in the research departments of stock brokerage firms. We examine the contents of reports for useful information on forecasting the movements of stock prices. First, our method extracts opinion sentences from the reports while the remaining parts are classified as non-opinion sentences. Second, our method predicts stock price movements by inputting the opinion and non-opinion sentences into separate neural networks.**

***Index Terms*—Text Mining, Analyst Report, Stock Prediction**

I. Introduction

The importance of equity investment in Japan is increasing. According to the Japan Exchange Group's (JPX) research report[1], the number of individual shareholders in Japan is rising. In particular, the number of individual investors in the fiscal year of 2017 increased by 1.22 million as compared to the previous year's number of 51.29 million, which exceeded 50 million for the first time. The number of individual investors is expected to continue increasing. Most recently, the stock prices of many companies are on the rise as a result of the effects of Abenomics[2] and the Olympics Games scheduled for 2020.

Investors need to examine much information in order to invest in target companies. However, the sources of information are diversified, and a process to collect information about everything is complicated. When we look at a company's website, there are numerous reports such as financial statements, financial results briefing materials, annual reports, and securities reports, on the Investor Relations page. If we use a search engine to find a company's name, then we can find many news reports. Moreover, internet message boards for financial markets include various investors' opinions related to financial information and stock price movements. Furthermore, in recent years, people's comments on social networks (SNS), such as Twitter, Facebook, and Instagram, may also reflect investor sentiment. Bollen et al. showed that mood states obtained from tweets are useful for forecasting the Dow Jones Industrial Average [1]. The progress of computation enabled us to refer to much information. On the other hand, it's getting more difficult for investors to find appropriate information for their investments.

Analyst reports are gathering more attention in this situation. An analyst report, as the name suggests, is a report written by analysts to evaluate individual companies by taking news, press releases, stock valuations, macroeconomic trends, etc into accounts. Therefore, we consider analyst reports as an upward compatibility of the information sources for each investment. In this study, we analyze the texts of the analyst reports with the aim to predict stock price trends. In particular, we aim to forecast the sign of stock price excess return to market and the extent of stock price volatility, which are particularly crucial in stock price trends.

Furthermore, we classify analyst reports by brokerage company and evaluate its effectiveness for each company since the style and content of analyst reports may depend on the company. Additionally, we use several word-embedding models created by various resources. Our secondary purpose is to analyze which combination of data would be useful for stock price forecasting. Therefore, we experiment with a variety of different data.

II. Methodology

In this section, we introduce our method for estimating the trends of stock prices. We assume that the effectiveness of

[1]https://www.jpx.co.jp/markets/statistics-equities/examination/01.html

[2]Abenomics refers to the economic policies advocated by Japanese Prime Minister Shinz Abe since the general elections of December 2012.

TABLE I: Example sentences of opinion and non-opinion sentences

Opinion or non-opinion	Sentences
opinion	原燃材料コストの下落も利益改善要因となった。 (Decline in raw material costs also contributed to profit improvement.)
opinion	昨年度から営業増益に転じたが、利益水準は低い見通し。 (Operating profit increased from last fiscal year, but the profit level is expected to be low.)
non-opinion	通期計画に変更はない。 (There is no change to the full-year plan.)
non-opinion	足元の経営環境を映す形で業績予想を見直した。 (The earnings forecast has been revised to reflect the current management environment.)

stock price estimation is different according to textual contents of the analyst reports. Because we believe that analyst reports are composed of analyst forecasts and objective facts, we distinguish between the opinion and non-opinion sentences in their reports.

Then, we construct an analysis of the flow of the reports as follows. First, our method distinguishes between opinion sentences and non-opinion sentences in the analyst reports. Then, our method estimates the trends of stock prices from the opinion sentences and non-opinion sentences, respectively. Before describing our method, we show how to prepare a data set of analyst reports.

A. Preparing data set for extracting of opinion sentences

In this section, we introduce the procedure for extracting opinion sentences from analyst reports. First, we extract 100 reports randomly from 10,100 analyst reports published in 2017. Then, we classify 2,213 sentences in the reports manually into opinion or non-opinion sentences. Here, opinion sentence is defined as a sentence containing an analyst's prediction of a variable, such as ratings for future stock prices, sales or predicted net earnings for next year, and backgrounds of current sales. A non-opinion sentence is defined as a sentence about facts such as past business results in this research. Examples of opinion sentences and non-opinion sentences are in Table I. After manual tagging 1,188 sentences were labeled as opinion sentences, and the other 1,025 sentences were labeled as non-opinion sentences.

B. Preparing data set for estimation of trends in stock prices

In this section, we describe the procedure for preparing stock price data for estimation. First, we acquire issued dates from 58,010 analyst reports.

Then, we acquire stock prices at the times of the publications of the analyst reports and the Tokyo Stock Price Index (TOPIX). In addition, we acquire stock prices and the TOPIX after 10 business days (about 2 weeks) from the publication dates. Using these values, excess returns are calculated. By using the price of a brand on the issue date of the analyst report P_{c0}, the price on the date after 10 business days P_{c10}, TOPIX on the issue date P_{t0} and TOPIX on the date after 10 business days P_{t10}, the excess return is calculated by Eq. (1).

$$\frac{(P_{c10} - P_{c0})}{P_{c0}} - \frac{(P_{t10} - P_{t0})}{P_{t0}} \tag{1}$$

The excess return is used because distribution of simple stock price returns can be distorted into the positive side around 2017 when Japan was in a long-term economic recovery. Moreover, for institutional investors, who are evaluated by relative performance to their benchmarks, the predictability of excess returns is important. A total of 58,010 analyst reports are used for the experiments. We use 1 to label positive excess returns and 0 to label negative excess returns mentioned in each analyst report. Each analyst report mentions at least one company. Of all the analyst reports, 29,430 reports are labeled as 1, whereas 28,580 reports are labeled as 0. We calculate the historical volatility of each brand. We obtain a stock prices, such as $P_{c0}, P_{c1}, \cdots, P_{c9}, P_{c10}$, for 10 business days after their issue dates. Volatility is the standard deviation (SD) of the array of the fractions of values of one day and the day before that, such as expressed by Eq. (2):

$$\mathbf{SD}\left(\frac{P_{c1}}{P_{c0}}, \frac{P_{c2}}{P_{c1}}, \cdots, \frac{P_{c10}}{P_{c9}}\right) \tag{2}$$

We use 1 to label that data whose absolute value of the volatility is higher than the median and 0 to label the data whose absolute value of the volatility is lower than the median. The median level is dependent on the input data. Those analyst reports are published by major brokers in Japan. The number of those reports we use was 58,010.

C. Making input vectors

We construct 200-dimensional word embeddings [2]. The embedding is done in two parts: decomposing sentence into words (the Japanese language does not have spaces between the words in a sentence) and converting each word to a vector, which is called a distributed representation. For the former part, we use MeCab[3] with the dictionary of mecab-ipadic-NEologd [3] dictionary, and for the latter part, we use Global Vectors for Word Representation (GloVe)[4]. We use five corpora to make the distributed representations in GloVe.

- Analyst reports 1
  - 2,213 sentences
  - The analyst reports used for extracting of opinion sentences
- Analyst reports 2
  - 702,315 sentences

[3]https://taku910.github.io/mecab/
[4]https://nlp.stanford.edu/projects/glove/

- 58,010 reports
- Reuters
  - Japanese articles of Reuters
  - 22,137,907 sentences
  - 2,890,515 articles
  - Period: From 1996 to 2018
  - Available for a fee
- Wikipedia
  - Japanese articles of Wikipedia
  - 19,364,683 sentences
  - 1,156,012 articles
  - Version on June 20, 2019
  - Available for free
  - Downloaded from https://dumps.wikimedia.org/jawiki/20190620/
- Nikkei
  - Articles from Nikkei news
  - 18,413,835 sentences
  - 4,959,256 articles
  - Period: From 1990 to 2017
  - Written in Japanese
  - Available for a fee

With comparison methods, we make a list of all the words in all the sentences we use. We make 0 vectors with dimensions that are the same as the length of the list for each sentence, and replace 0 at an index in the list of a word in a sentence with 1 (Fig. 1).

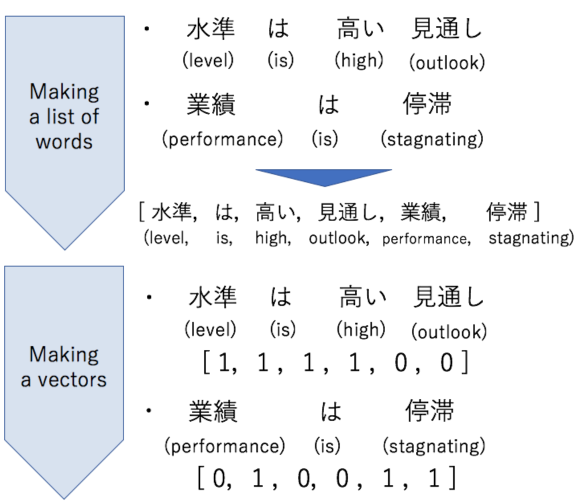

Fig. 1: Making vectors of sentences in comparison methods

D. *Opinion sentence extraction*

Here, we introduce our opinion sentence extraction model that uses neural networks. Recurrent neural networks (RNNs) have achieved superior performance in natural language processing tasks. In particular, Long Short-Term Memory (LSTM) [4] [5] and gated recurrent unit (GRU) [6], which are a type of RNN, have exhibited high performance. Therefore, in this study, we employ these models for opinion sentence extraction. In using LSTM or GRU, we use bidirectional ones. In common single-directional LSTM or GRU, only past information is used for learning. On the other hand, in bidirectional LSTM and GRU, not only the past but also future information is used for learning.

Input vectors (word embeddings) to LSTM and GRU were executed by GloVe. To align the sequence length, we pad inputs that do not have the same sequence lengths as the longest sequence with 200-dimensional 0 vectors. Between the LSTM or GRU layers and multi-layer perceptron (MLP) layers, we place a self-attention mechanism (Fig. 2), which lets us know which parts are stressed in prediction model to make more accurate predictions. Hidden state vectors that go through the LSTM or GRU are propagated to the self-attention mechanisms. The outputs of the self-attention mechanism are propagated to MLP layers. On the last MLP layers, the probabilities of 1 and 0 are output. A higher probability is adopted as a result.

We describe our method for LSTM. Here, we define LSTM processing from the beginning of a sentence as $\overrightarrow{LSTM}$ and from the end of the sentence as $\overleftarrow{LSTM}$. For each input, our method obtains $\{\overrightarrow{h_i}\}_i^n$ and $\{\overleftarrow{h_i}\}_i^n$ through LSTM($\overrightarrow{LSTM}, \overleftarrow{LSTM}$).

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(e_i), \overleftarrow{h_i} = \overleftarrow{LSTM}(e_i) \tag{3}$$

Here, n is the number of input words and e_i is the vector entered ith words.

We define h_i as the concatenation of $\overleftarrow{h_i}$ and $\overrightarrow{h_i}$.

$$h_i = \begin{bmatrix} \overleftarrow{h_i} \\ \overrightarrow{h_i} \end{bmatrix} \tag{4}$$

Then, h_i are entered into the output layer as follows:

$$s = \sum_i^n \alpha_i \cdot h_i \tag{5}$$

$$t = tanh(W_s \cdot s + b_s) \tag{6}$$

$$Y = W_t \cdot t + b_t \tag{7}$$

Here, $h \in \mathbb{R}^{2m}$, $s \in \mathbb{R}^{2m}$, and $t \in \mathbb{R}^l$. Here, W_s and W_t are weighted matrices, b_s and b_t are bias vectors, m is the number of units in the hidden layer, l is the number of units in the middle layer, and Y is an output layer comprising $Y = (y_1, y_2)$. And, α is the attention weight and calculated by the following formula.

$$u = tanh(W_h \cdot h + b_h) \tag{8}$$

$$\alpha_i = \frac{e^{u_i}}{\sum_{j=1}^n e^{u_j}} \tag{9}$$

Here, $u \in \mathbb{R}^n$ and W_h is a weighted matrix while b_h is a bias vector. Finally, our method selects the output unit having a maximum value from the output layer as output.

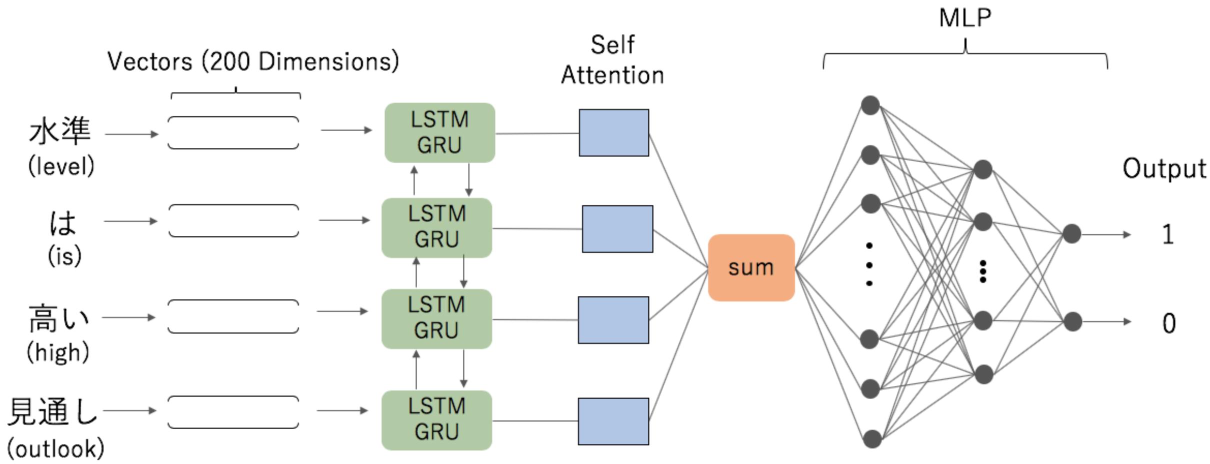

Fig. 2: Model for distinguishing between opinion and non-opinion sentences and estimation of stock prices with all sentences or only opinion sentences or only non-opinion sentences

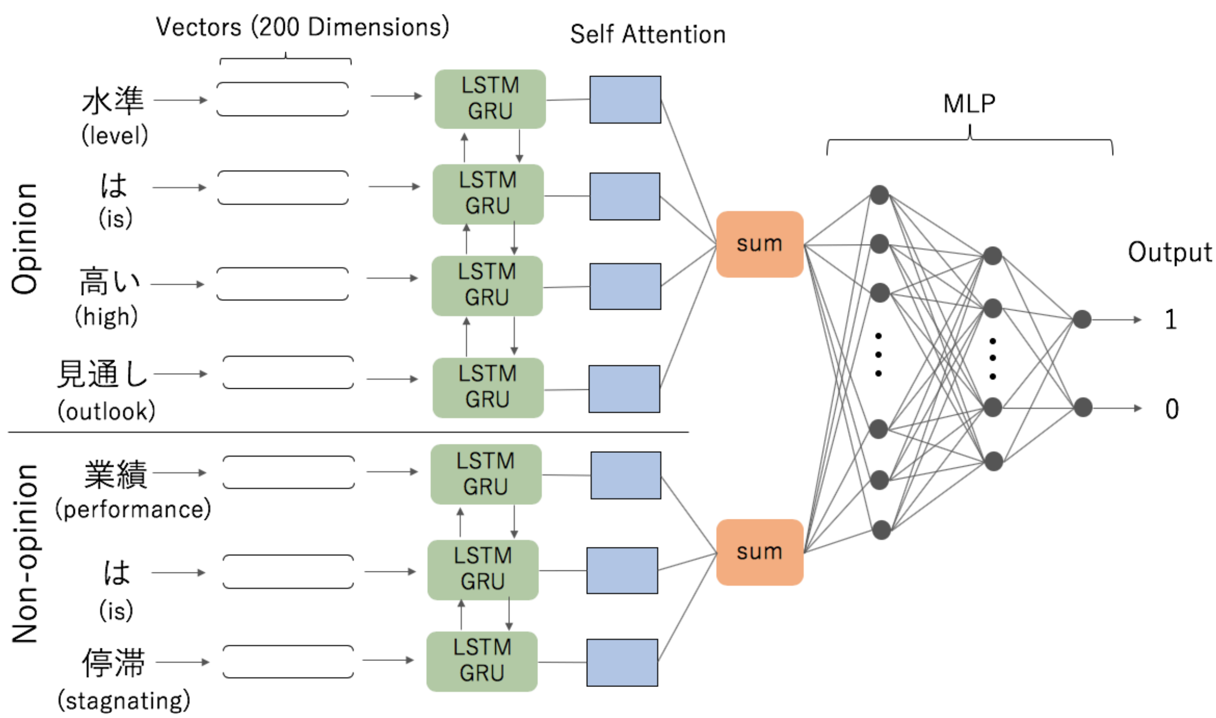

Fig. 3: Model of stock price analysis with separate opinion sentences and non-opinion sentences

E. Estimation of stock prices

In this section, we describe one of our stock price estimation neural network models. The network of the three ways of using all sentences, or using only opinion sentences or using only non-opinion sentences is illustrated in Figure 2. The network of the way of using opinion sentences and non-opinion sentences separately is illustrated in Figure 3.

Therefore, to fix the model (Fig. 3), we update the formula 5, 6, and 7 as follows.

$$s_{opinion} = \sum_{i}^{n} \alpha_i^{opinion} \cdot h_i^{opinion} \quad (10)$$

$$s_{nonopinion} = \sum_{i}^{n} \alpha_i^{nonopinion} \cdot h_i^{nonopinion} \quad (11)$$

$$t = tanh(W_s \cdot [s_{opinion}; s_{nonopinion}] + b_s) \quad (12)$$

$$Y = W_t \cdot t + b_t \quad (13)$$

Here, $h_i^{opinion}$ is a hidden layer of $LSTM_{opinion}$ that has the

TABLE II: Parameters for distinguishing between positive excess returns and negative excess returns

Parameter	Values
Epoch	30, 35, 40, 45, $\cdots$, 135, 140, 145, 150
RNN model	LSTM, GRU
Hidden layers and inner layers	(50, 25), (80, 40), (120, 60)
Mini-batch size and learning rate	(64, 1×10^{-4}), (128, 1×10^{-4}), (256, 1×10^{-4}), (128, 2×10^{-4}), (256, 2×10^{-4})
Corpus	analyst reports 1, analyst reports 2, Reuters, Wikipedia, Nikkei

TABLE III: Parameters for distinguishing between positive excess return and negative excess return

Parameter	Values
Inputs	all sentences, opinion sentences, non-opinion sentences, opinion sentences and non-opinion sentences separately
Broker	A, B, $\cdots$
Epoch	30, 35, 40, 45, $\cdots$, 135, 140, 145, 150
RNN model	LSTM, GRU
Hidden layers and inner layers	(50, 25), (80, 30), (120, 40)
Mini-batch size and learning rate	(128, 5×10^{-5}), (128, 2×10^{-5}), (256, 5×10^{-5})
Corpus	analyst reports 1, analyst reports 2, Reuters, Wikipedia, Nikkei

TABLE IV: Top 5 results of distinguishing opinion sentences

F1	Model	Corpus	Epoch	Learning Rate	Mini-batch	Hidden Layers	Inner Layers
0.836	GRU	analyst reports 2	130	1×10^{-4}	256	50	25
0.835	GRU	analyst reports 2	125	1×10^{-4}	256	50	25
0.835	LSTM	analyst reports 1	50	1×10^{-4}	64	80	40
0.835	GRU	analyst reports 2	120	1×10^{-4}	256	50	25
0.834	GRU	analyst reports 2	135	1×10^{-4}	256	50	25

TABLE V: Results of distinguishing between positive excess returns and negative excess returns with comparison methods

Model	F1 (Test)	Broker
SVC	0.506	A
Random Forest	0.553	A

opinion sentences as inputs. $h_i^{nonopinion}$ is a hidden layer of $LSTM_{nonopinion}$ that has the non-opinion sentences as inputs. Additionally, $\alpha_i^{opinion}$ is an attention weight of $LSTM_{opinion}$. $\alpha_i^{nonopinion}$ is an attention weight of $LSTM_{nonopinion}$.

F. Experiments

1) Distinguishing opinion sentences : For the task of learning to distinguish between opinion sentences and non-opinion sentences, inputs are the vectors of the words in a sentence. Of all 2,213 sentences, 70% are used for training, 10% are used for validation, and the remaining 20% are used for testing. We change hyperparameters such as type of an RNN model, the number of epochs, the number of hidden layers of the RNN, the number of the inner layers of MLP, a mini-batch size, a learning rate, and a corpus. Types of RNN models are LSTM and GRU. The parameters we choose are listed in Table II. We also perform this task with comparison methods, i.e., Linear Support Vector Classification (SVC) and Random Forest.

2) Estimation of stock prices: In this experiment, we perform two tasks. One task is the distinction between positive excess returns and negative excess returns, whereas the other is the distinction between high volatility and low volatility. We experiment with four ways to input into LSTM or GRU, i.e., using all the sentences in analyst reports, using only opinion sentences, using only non-opinion sentences, using opinion sentences and non-opinion sentences separately (Fig. 2). We use the model illustrated in Section II-F1 with the best parameters to separate all the sentences in the analyst reports into opinion sentences and non-opinion sentences (Fig. 3). The inputs are the vectors of the words in each analyst report. To reduce the influence of padding with 0 vectors, we limit the sequence length. The maximum sequence length when using all sentences in the analyst reports is 530, the maximum one when using only opinion sentences is 370, the maximum one when using only non-opinion sentences is 250. This criterion is based on the way that the sequence length of 90% of the inputs without padding is less than these numbers. We experiment with some ways to input the analyst reports, i.e., inputting the analyst reports published by Broker A, published by Broker B, $\cdots$, respectively and all the brokers. We also perform this task with the comparison methods, i.e., SVC and Random Forest. We experiment with different brokers as the input.

III. Results

A. Distinguishing opinion sentence

The top five results in the f1-score are in Table IV. F1 is the average f1 score of eightfold cross validation. The f1 score of the test data with the best parameters (at the top of Table IV) is 0.813. The test scores of SVC and Random Forest are 0.797 and 0.664, respectively.

B. Estimation of stock prices

1) Distinction between positive excess returns and negative excess returns: The top five results in the f1-score are in Table VI. F1 is the average f1 score of eightfold cross validation.

TABLE VI: Top 5 results of distinguishing between positive excess return and negative excess return

F1	Input Sentences	Broker	Model	Corpus	Epoch	Learning Rate	Mini-batch	Hidden Layers	Inner Layers
0.574	all sentences	A	LSTM	analyst reports 2	140	2×10^{-5}	128	50	20
0.573	all sentences	A	LSTM	analyst reports 1	140	2×10^{-5}	128	120	40
0.573	all sentences	A	LSTM	analyst reports 1	120	2×10^{-5}	128	120	40
0.573	all sentences	A	LSTM	analyst reports 1	135	2×10^{-5}	128	120	40
0.573	all sentences	A	LSTM	analyst reports 1	145	2×10^{-5}	128	120	40

TABLE VII: Results of distinguishing between high volatility and low volatility with comparison methods

Model	F1 (Test)	Broker
SVC	0.571	A
Random Forest	0.627	A

The f1 score of the test data with the best parameters (at the top of Table VI) is 0.558. The test scores of SVC and Random Forest are in Table V.

2) Distinction between high volatility and low volatility: The top five results in the f1-score are in Table VIII. F1 is the average f1 score of eightfold cross validation. The f1 score of the test data with the best parameters (at the top of Table VIII) is 0.635. The test scores of SVC and Random Forest are in Table VII.

IV. Discussion

We achieved more than 80% f1 score to distinguish between opinion sentences and non-opinion sentences. In the 15 results in Table IV, VI, and VIII, the corpora of 14 results were based on analyst reports. The analyst reports contain not only facts but also the opinions of the analysts, whereas articles in newspapers and Wikipedia usually contain only facts or information. For this reason, it can be said that analyst reports contain their own expressions and corpora based on analyst reports got higher f1 scores.

On the other hand, according to Table VI and Table VIII, to distinguish opinion sentences or non-opinion sentences was not effective for analyzing stock prices. The results indicate that our assumption had been mistaken. Therefore, we need to further analyze which parts of the analyst reports affect stock price movements.

However, according to Table VI and Table VIII, when creating a word embedding, we found that there were valid analyst reports. For example, analyst reports 2 showed high performance in the experiment to distinguish between opinion sentences and non-opinion sentences. Moreover, analyst reports 1 showed high performance in the experiment to distinguish between high volatility and low volatility. We believe that the results indicate that the analyst reports of brokers had different characteristics concerning stock price estimation when using the texts of analyst reports, i.e., we should use different analyst reports according to the contents of the analysis.

V. Related Works

Bollen et al. showed that tweet moods were useful for forecasting the Dow Jones Industrial Average [1]. In their research, they used self-organizing fuzzy neural networks for forecasting. As a result, they were able to predict rises and drops with an accuracy of more than 80%. Schumaker et al. proposed a machine-learning approach for predicting stock prices by analyzing financial news articles [7]. Their research predicts indicators and stock prices by using one resource. On the other hand, our method uses a combination of several documents, such as analyst reports and the Wikipedia corpus, for predicting stock price movements.

Concerning financial text mining, Koppel et al. proposed a method for classifying the news stories of a company according to their apparent impacts on the performance of the company's stock [8]. Low et al. proposed a semantic expectation-based knowledge extraction methodology (SEKE) for extracting causal relationships [9] by using WordNet as a thesaurus for extracting terms representing movement concepts. Ito et al. proposed a neural network model for visualizing online financial textual data [10] [11]. Additionally, their neural network model could acquire word sentiment and its category. Milea et al. predicted the MSCI euro index (upwards, downwards, or constant) based on fuzzy grammar fragments extracted from a report published by the European Central Bank [12].

With regards to financial text mining for the Japanese language, Sakaji et al. proposed a method to automatically extract basis expressions that indicated economic trends from newspaper articles by using a statistical method [13]. Additionally, Sakaji et al. proposed an unsupervised approach to discover rare causal knowledge from financial statement summaries [14]. Their method extracts basis expressions and causal knowledge by using syntactic patterns. Kitamori et al. proposed a method for extracting and classifying sentences indicating business performance forecasts and economic forecasts from summaries of financial statements [15]. The classification method is based on a neural network using a semi-supervised approach. Hirano et al. proposed an extended scheme for selecting related stocks for themed mutual funds [16] [17]. They used some Japanese documents such as Japanese financial summaries, news articles and web pages for their methodology.

These financial text mining studies targeted only one language only. In contrast, our method used stock price movements as the target data.

TABLE VIII: Top 5 results of distinguishing between high volatility and low volatility

F1	Input Sentences	Broker	Model	Corpus	Epoch	Learning Rate	Mini-batch	Hidden Layers	Inner Layers
0.667	all sentences	B	GRU	analyst reports 2	135	5×10^{-5}	256	120	40
0.666	all sentences	B	GRU	analyst reports 2	145	5×10^{-5}	128	50	20
0.666	all sentences	B	GRU	Reuters	115	5×10^{-5}	128	120	40
0.665	all sentences	B	GRU	analyst reports 2	125	5×10^{-5}	256	120	40
0.665	all sentences	B	GRU	analyst reports 2	130	5×10^{-5}	128	80	30

VI. Conclusion

With the assumption that an opinion of an analyst would be effective to predict on stock prices, we split analyst reports into opinion sentences and non-opinion sentences. Under this experimental condition, the assumption was not validated. We obtained better results by inputting all the sentences. On the other hand, a combination of a source of a corpus was effective for estimation of stock price movements. It was profitable information. In the future, we will experiment with other conditions about opinion sentences and non-opinion sentences to verify the effectiveness of the way of natural language processing for fluctuations of stock prices.

References

[1] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, Vol. 2, No. 1, pp. 1–8, 2011.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, 2013.

[3] Sato Toshinori. Neologism dictionary based on the language resources on the web for mecab, 2015.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[5] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, Vol. 18, No. 5, pp. 602–610, 2005.

[6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, October 2014.

[7] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, Vol. 27, No. 2, pp. 12:1–12:19, March 2009.

[8] Moshe Koppel and Itai Shtrimberg. *Good News or Bad News? Let the Market Decide*, pp. 297–301. Springer Netherlands, Dordrecht, 2006.

[9] Boon-Toh Low, Ki Chan, Lei-Lei Choi, Man-Yee Chin, and Sin-Ling Lay. Semantic expectation-based causation knowledge extraction: A study on hong kong stock movement analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 114–123, 2001.

[10] Tomoki Ito, Hiroki Sakaji, Kota Tsubouchi, Kiyoshi Izumi, and Tatsuo Yamashita. Text-visualizing neural network model: Understanding online financial textual data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 247–259, 2018.

[11] Tomoki Ito, Hiroki Sakaji, Kiyoshi Izumi, Kota Tsubouchi, and Tatsuo Yamashita. Ginn: gradient interpretable neural networks for visualizing financial texts. *International Journal of Data Science and Analytics*, Dec 2018.

[12] Viorel Milea, Nurfadhlina Mohd Sharef, Rui Jorge Almeida, Uzay Kaymak, and Flavius Frasincar. Prediction of the msci euro index based on fuzzy grammar fragments extracted from european central bank statements. In *2010 International Conference of Soft Computing and Pattern Recognition*, pp. 231–236, Dec 2010.

[13] Hiroki Sakaji, Hiroyuki Sakai, and Shigeru Masuyama. Automatic extraction of basis expressions that indicate economic trends. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 977–984, 2008.

[14] Hiroki Sakaji, Risa Murono, Hiroyuki Sakai, Jason Bennett, and Kiyoshi Izumi. Discovery of rare causal knowledge from financial statement summaries. In *The 2017 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, pp. 602–608, 2017.

[15] Shiori Kitamori, Hiroyuki Sakai, and Hiroki Sakaji. Extraction of sentences concerning business performance forecast and economic forecast from summaries of financial statements by deep learning. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, 2017.

[16] Masanori Hirano, Hiroki Sakaji, Shoko Kimura, Kiyoshi Izumi, Hiroyasu Matsushima, Shintaro Nagao, and Atsuo Kato. Selection of related stocks using financial text mining. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 191–198, 2018.

[17] Masanori Hirano, Hiroki Sakaji, Shoko Kimura, Kiyoshi Izumi, Hiroyasu Matsushima, Shintaro Nagao, and Atsuo Kato. Related stocks selection with data collaboration using text mining. *Information*, Vol. 10, No. 3, 2019.