Stock Movement Prediction That Integrates Heterogeneous Data Sources Using Dilated Causal Convolution Networks with Attention

Conference Paper · May 2020

DOI: 10.1109/ICASSP40776.2020.9053479

Divy Daiya

5 PUBLICATIONS **44** CITATIONS

SEE PROFILE

Che Lin

National Taiwan University

77 PUBLICATIONS **986** CITATIONS

SEE PROFILE

STOCK MOVEMENT PREDICTION THAT INTEGRATES HETEROGENEOUS DATA SOURCES USING DILATED CAUSAL CONVOLUTION NETWORKS WITH ATTENTION

Divyanshu Daiya*, *Min-Sheng Wu*‡ , *Che Lin*†

*Department of Computer Science & Engineering, The LNM Institute of Information Technology, India
‡Department of Electrical Engineering, National Tsing Hua University, Taiwan
† Department of Electrical Engineering & Graduate Institute of Communication Engineering,
National Taiwan University, Taiwan
*daiyadivyanshu@gmail.com, ‡akira@gapp.nthu.edu.tw, †chelin@ntu.edu.tw

ABSTRACT

The purpose of this research is to develop a high performing model for stock movement prediction utilizing financial indicators and news data. Until recently, the majority of prediction models have employed only the financial indicators, but they possess the risk of missing unconventional agitators that can be derived from other heterogeneous sources. To address this, few research studies began to explore the use of news data and other social features along with financial indicators. In this work, we propose a novel integrative approach to effectively blend views from the news and financial time series. We generate event-knowledge representations from news data by capturing direct and inverse relationships among event tuples, and then apply attention mechanism to infer inter-day relationships among the representations. To capture temporal dynamics of financial indicators, we further integrate an attention augmented dilated causal convolutional network. We report empirically that our model achieves a substantial 5% improvement from 68.81% to 74.29% in stock movement prediction for the Standard & Poor's 500 (S&P500) index and companies over existing models.

Index Terms— Deep Learning, Dilated Convolution, Stock Prediction, Event Embedding, Attention

1. INTRODUCTION

As of February 2018, the US Stock Market alone reported stock exchange by market capitalization of its listed companies at US$ 30.1 trillion; this signifies how much attractive stock trading has become. With many new investors joining every day, it remains the aspiration of every investor to be able to forecast market behaviour before making any buy or sell decision. However, it has always been challenging to do stock prediction given the highly volatile and non-stationary nature of the stock markets. Market hypothesis like random walk theory [1] purports that the stock prices movements are defined randomly and cannot be forecasted, but with the recent developments in artificial intelligence and massive data availability, it is now possible to provide better predictions than random guesses. Stock prediction involves either regression, i.e., predicting exact stock value or movement forecast, i.e., predicting whether stock value dips or increases. In this work, we focus on the prediction of stock movement. Given the time-dependent nature of stock fluctuation, recurrent networks seem to be a natural fit, and many studies have demonstrated the same, e.g., the use of long short-term memory (LSTM) and gated recurrent unit (GRU) lately have provided a substantial improvement over the conventional models [2]. Recurrent networks have an inherent ability to model sequences and realize long-term patterns in the data, which other neural network architectures like convolutional neural networks (CNN) fail to capture. However, recently a CNN architecture (WaveNet) developed by Google DeepMind [3], which applied dilated causal convolutions with residual connections, have shown to provide better or comparable results to LSTMs and GRUs [4]. We hypothesize that the use of a similar derivative, dilated causal convolutional neural network (DC-CNN), for financial time series forecasting task can provide better results over benchmarks.

Traditionally, the stock prediction models took into consideration only the financial indicators, but they are not able to capture the complicated relations in actual market movement. This could be due to the fact that many social and political factors influence the stock values, which may not be reflected by only the current or past financial factors. Given the advancements in natural language processing (NLP), these factors can be better analyzed using the data from heterogeneous sources such as news or social networking websites. The inclusion of such data would help better capture the market volatility and abrupt changes, which would not be otherwise possible. Many previous studies have employed such data to obtain substantial improvement in stock prediction. Some models have proposed the use of LSTMs to extract word embeddings [5][2]. Others have used support vector machine (SVM)

[†] Corresponding Author

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

[6] while some have used event embeddings and employed LSTM-CNN[7][8] for predictions. Compared to word and sentence embeddings, structured event embeddings are more useful because they inherently capture associations in the text by extracting subject-action-object pairs. For example, ”Apple TV Plus cost undercutting Netflix”, Apple is the subject, cost undercutting is the action, and Netflix is the object. From this, we deduce that the corresponding shares of Netflix would fall. However, we might miss ”who” is the suffering party with the conventional embeddings, but the use of events help us keep this information intact [7]. In this work, we propose an extension to such an idea and map events to embedding by assimilating inverse event relations [9]. To capture the inter-day event dependencies and to make predictions, we employ attention mechanism with neural tensor network (NTN). The attention here helps to emphasize the most appealing days in the window by automatically capturing the relation between values at different time stamps [10]. We employ a similar attention mechanism for financial indicators before feeding them into a DC-CNN architecture. In summary, we propose a novel model which extracts features from financial indicators using DC-CNN and employ attention augmented neural tensor network to extract features from the embedding generated by inverse embedding (InvED), and eventually combine the embeddings from both heterogeneous data sources. Our model obtained better performance than established models evaluated on a dataset produced by [7] and the relevant Standard & Poor’s 500 (S&P 500) stocks from the corresponding period. We demonstrate that our proposed model performed the best in terms of accuracy (Acc) and Matthews correlation coefficient (MCC) among existing models for stock prediction.

2. PRELIMINARIES

2.1. Dilated Causal Convolution

Dilated convolution was proposed to aggregate multi-scale contextual information [11], which is otherwise lost due to the down-sampling operation. The causal convolution helps keep in check that the network performs convolution operation using only the current and historical values, as required in the time-series analysis. The WaveNet architecture [3], developed by Google DeepMind, used skip connections and stacked dilated causal convolutions blocks to generate better raw waveforms than existing models.

The dilated convolution in each convolutional layer skips input values with a certain step size. Note that traditional convolution has a dilation of one. The dilation helps to extend the receptive field of the network exponentially without requiring many convolutional layers or large filter sizes [3]. The expression for the receptive field r, with the number of layers L and filter size k with dilation assumed to be 2 for each layer, is given by $r = 2^{L-1} * k$. The receptive field increases as we traverse through the layers, which means that later layers capture relations from more distant time-stamps. The skip-connections were used to remember features captured from earlier layers.

2.2. Event Embedding

Given an event tuple $E = (E_1, R, E_2)$, where E_1 entity is the actor, R is the action or relation, E_2 entity is an object on which the action is performed, we aim to generate a w dimensional embedding and use the same for stock prediction. Specifically, we extend the embedding generation framework proposed by [7], which utilized NTN. Ding’s NTN model only utilized forward relation for the event tuple, i.e., the mapping from entity E_1 to E_2 as R, but as demonstrated by Kazemi [9] in addition to forward mapping the utilization of inverse relation between entities which is from E_2 to E_1 as R_{inv} considerably improved link prediction performance in knowledge graphs. We employ Kazemi’s proposition [9] in our extension $InvED$ to Ding’s model [7].

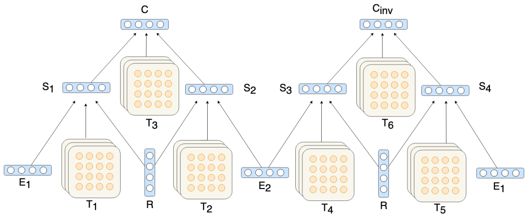

Fig. 1. $InvED$: Proposed NTN for Event Embedding Generation

As shown in Figure 1, the NTN utilizes bilinear tensors T_1 and T_2 to model the relationship between (E_1, R) and (R, E_2) while tensors T_4 and T_5 to model the relationship between (E_2, R) and (R, E_1). We denote $\mathbf{e_1}$, $\mathbf{e_2}$, and $\mathbf{r}$ as the $d-$dimensional aggregate Glove word embeddings [12] for E_1, E_2, and R respectively. C and C_{inv} model relationship between S_1, S_2 and S_3, S_4 respectively. Then, $S_1 \in \mathbb{R}^d$ is computed by:

$$S_1 = f\left(\mathbf{e_1^T} T_1^{[1:k]}\mathbf{r} + W\left[\begin{array}{c}\mathbf{e_1}\\ \mathbf{r}\end{array}\right] + b\right) \tag{1}$$

$$g(E_1, R) = g(S_1) = U^T S_1$$

where $T_1^{[1:k]} \in \mathbb{R}^{d\times d\times k}$ is a tensor, which is a set of k matrices, each with $d \times d$ dimensions. The bilinear tensor product $\mathbf{e_1^T} T_1^{[1:k]}\mathbf{r}$ is a vector $\mathbf{q} \in \mathbb{R}^k$, where each entry is computed by using one slice of the tensor ($q_i = \mathbf{e_1}^T {T_1}^{[i]}\mathbf{r}, i = 1, \cdots, k$). The other parameters are standard neural network parameters, where $W \in \mathbb{R}^{k\times 2d}$ is the weight matrix, $b \in \mathbb{R}^k$ is the bias vector, $U \in \mathbb{R}^k$ is a hyper-parameter and $f = tanh$ is a standard nonlinear activation applied element-wise. S_2, S_3, S_4 and C, C_{inv} in Figure 1 are computed similarly as S_1.

To train the model, we assume that the corrupted tuples should have a higher loss than the true event tuples in the training data. We generate the corrupted tuples by randomly replacing any of the two entities. Specifically, the corrupted event tuple $E^r = (E_1^r, P, E_2)$ is obtained by replacing the word in E_1 with a random word w^r from set $\mathcal{D}$ (all the words in the training data) to obtain a corrupted counterpart E_1^r. We choose randomly between E_1 and E_2 to corrupt. Note that this is different from [12], wherein only E_1 is corrupted. We calculate the *margin loss* [13] of the two event tuples as:

$$\mathcal{L}_{\mathcal{E}} = loss(E, E^r) = \max(0, 1 - G(E) + G(E^r)) + \lambda \|\Phi\|_2^2, \tag{2}$$

$$\text{with } G(E) = g(C) + g(C_{inv}), \tag{3}$$

where $\mathit{\Phi} = (T_1, T_2, T_3, T_4, T_5, T_6, W, b)$ is the set of model parameters. The loss function requires that the score generated by the actual tuple should be at least greater by unity than the one generated by a corrupted tuple, for the loss to be equal to zero. We iterate over training data 500 times. L_2 regularization is used, with weight λ set to 0.0001. For each training instance, if the loss is equal to zero, the training algorithm proceeds to process the next event tuple. Otherwise, the parameters are updated to minimize the loss using backpropagation.

2.3. Self Attention Mechanism

Given a time series matrix $X_{t,L}$ with time variable t and a time window of length L, we apply self-attention as:

$$H_t = tanh(X_{t,L}W_t + B_t),\ E_t = tanh(W_a H_t + B_a) \tag{4}$$

$$A_t = softmax(E_t),\ X'_{t,L} = A_t \odot X_{t,L}, \tag{5}$$

where $X_{t,L}, X'_{t,L} \in \mathbb{R}^{t \times L}$, $W_t \in \mathbb{R}^{L \times L}$, $W_a \in \mathbb{R}^{t \times t}$, B_t, B_a, H_t, E_t, $A_t \in \mathbb{R}^{t \times L}$, $X'_{t,L}$ is the weighted output, W_t, B_t, W_a, B_a are trainable parameters with functions $softmax$ and $tanh$ being applied element-wise.

3. ARCHITECHTURE

We have divided our proposed architecture Attention Dilated Convolution and Event network (Att-DiCE) *(Fig. 2)* into two segments, each segment dealing with feature extraction from financial indicators and news data, respectively. As discussed earlier, for extracting features, we employ DC-CNN for financial indicators and NTN for event embeddings, i.e. the news data. We generate a $p-$dimensional embedding as the output from each segment. Then, we concatenate and connect them to a fully-connected (FC) layer and finally employ $softmax$ activation for movement prediction, i.e., whether the stock price goes *up* or *down*. The self-attention mechanism are applied to both the input $x_{t,L}$ and $e_{t,L}$ to the Att-DCNN i.e attention augmented dilated causal convolutional network and Att-biNTN i.e. attention augmented bilinear neural tensor network, respectively. The attention helps highlighting dominant values across time-series by capturing their underlying relation over the complete instance. The attention is time-restricted [14], i.e., it is only applied over a time window of length k. Similar to Vaswani [10], our attention mechanism is soft, i.e. it can simultaneously attend to different points in time with different weights.

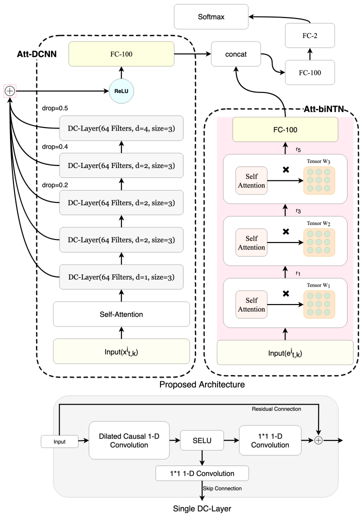

Fig. 2. Att-DiCE

3.1. Att-DCNN

The WaveNet architecture [3], which introduced dilated convolutions and residual blocks for time series forecasting, employed gated convolutions but it suffers from slow convergence to low training error for non-stationary noise [15]. Augmented WaveNet [15] proposed the use of $ReLU$ instead and provided a lower error rates. We further tested $tanh$, $LeakyReLU$ [16], $SELU$ [17], of which $LeakyReLU$, $SELU$ performed the best since the negative values are preserved in both. As a result, we use $SELU$ activations after each DC-Convolution, and the output from each residual block is passed on to the next block and to the final layer (via skip connections) after 1×1 convolution. Different dropout rates are applied for each skip connections to ensure that the latent representations developed in the dilated convolutional layers are not overly influenced by past trends that may not be relevant in the recent forecast horizon of the time series. All the skip connections are summed and connected to a 100 cell FC layer to generate the final 100-dimensional embedding.

3.2. Att-biNTN

We use attention augmented tensor layers to extract the features from time series of the event. We derive the idea from Vasavani [10], which demonstrated how the use of transformers could provide better performance than recurrent networks and other conventional models. Transformers employ stacked blocks of multi-head attention and feed-forward network, with model relying primarily on the attention mechanism for

Models	Index Acc(%)	Avg Acc(%)
Ding, 2014	58.83	61.02
EB-CNN	64.21	64.23
SI-RCNN	63.09	60.89
KGEB-CNN	66.93	64.56
DA-RNN	68.05	68.81
Att-DiCE	**73.89**	**74.29**
Att-DCNN	71.26	72.32
Att-biNTN	69.14	70.12
Att-biNTN (w/o InvED)	67.33	67.91

Table 1. Accuracy for Index Prediction and Avg. Accuracy

capturing long-range dependencies. The results have been outstanding with the architecture providing remarkable improvements for natural language understanding (NLU) and NLP [18] tasks. As realized in Figure 2, our model employs three stacked layers of attention augmented tensor network. For a given single training instance $e^i_{t,k}$,

$$e^{i'}_{t,k} = Self\ Attention(e^i_{t,k}), r_1 = e^{i'}_{t,k} \times W_1 \qquad (6)$$

$$r_2 = Self\ Attention(r_1), r_3 = r_2 \times W_2 \qquad (7)$$

$$r_4 = Self\ Attention(r_3), r_5 = r_4 \times W_3 \qquad (8)$$

$$r_6 = Fully - Connected_{100}(r_5) \qquad (9)$$

r_6 is the 100-dim embedding returned, where $e^i_{t,k}, e^{i'}_{t,k} \in \mathbb{R}^{t\times k}, W_1 \in \mathbb{R}^{k\times 100}, W_2, W_3 \in \mathbb{R}^{100\times 100}, r_1, r_2, r_3, r_4, r_5 \in \mathbb{R}^{t\times 100}$, $r_6 \in \mathbb{R}^{100}$. For input in the network, we generate C and C_{inv} for each event, concatenate them into $C_T \in \mathbb{R}^{2d}$. We aggregate events for a particular day to generate series $E_{2d,L_{TotalDays}}$, with model given $L_{TotalDays} - k + 1$ training instances $e^i_{t,k}$ each using k days, where $i \in (1, L_{TotalDays} - k + 1)$, $t = 2d$.

4. NUMERICAL RESULTS AND DISCUSSION

We use the publicly available data released by Ding [7], which contains financial news article from Bloomberg and Reuters over the period of October 2006 to November 2013. We obtain S&P 500 index and individual stock prices over the same period from Yahoo Finance, with a total 1,782 instances. We follow the same train, development, and test split as Ding [7], i.e. 80%, 10% and 10%, respectively. Structured events are extracted using ”OpenIE” [19] and parsed using Fader [20]. We use Open, High, Close, Low, Volume (OHCLV) values and derivative's relative strength index (RSI), stochastic oscillator (SO), moving average convergence divergence (MACD), rate of change (ROC), on balance value (OBV), weighted moving average (WMA) as financial indicators. For both the Att-biNTN and Att-DCNN, we use a time window of 40 days, which means closing price movement prediction for a day requires data from past 40 days. We use Adam optimizer [21] with a starting learning rate of 0.0075 and linearly decaying by 0.1 for every 10000 iteration. We train the model for 100 epochs with a batch size of 32.

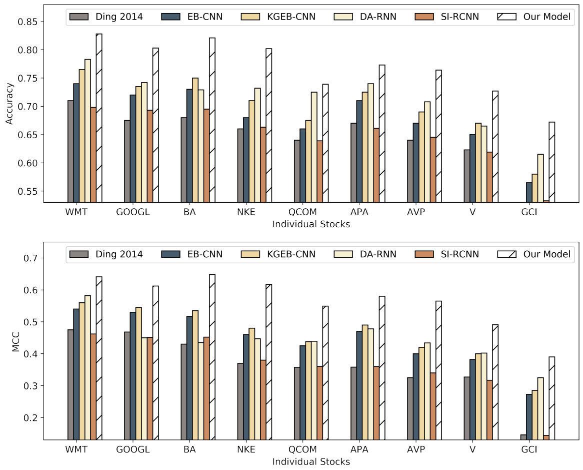

Fig. 3. Accuracy and MCC comparison

Following [22], the standard measure of Accuracy(Acc) and Matthews correlation coefficient(MCC) are used to evaluate the performances of S&P 500 index and individual stock prediction. We use Ding (2014) [23], EB-CNN [7], KGEB-CNN [8], SI-RCNN [5], DA-RNN [24] as a set of baseline models. DA-RNN was developed for stock price prediction and provided the least MAE among all existing models; we can utilize this predicted price for movement prediction. We can see that our model provides the best MCC and Acc for index prediction *(Table 1)* as well as for individual stock prediction *(Fig. 3)*. Averaging our model performance over index and Top 20 S&P 500 companies, we demonstrate that our model provides around 5% improvement from 68.81% to 74.29% over the best performing model DA-RNN. We also test the performance of the Att-DCNN and Att-biNTN; we still obtain an average model performance of 72.32% and 70.12% respectively. To examine the effectiveness of our event generation model (InvED), we test the event model standalone against the original NTN architecture proposed by Ding [7]. We showed a performance improvement from 67.91% to 70.12% validating our hypothesis that inverse embeddings indeed increase performance. We can also say the same for Att-biNTN as there is a performance increase from 64.23% to 67.91%. Our proposed Att-DCNN architecture can achieve 72.23% accuracy, confirming the idea that a more effective use of dilated convolutions can provide better performance than conventional models.

5. REFERENCES

[1] Eugene F Fama, “Random walks in stock market prices,” *Financial analysts journal*, vol. 51, no. 1, pp. 75–80, 1995.

[2] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara, “Deep learning for stock prediction using numerical and textual information,” *2016 IEEE/ACIS 15th ICIS*, pp. 1–6, 2016.

[3] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[4] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda, "Speaker-dependent WaveNet vocoder.," *Interspeech*, pp. 1118–1122, 2017.

[5] Manuel R Vargas, Beatriz SLP De Lima, and Alexandre G Evsukoff, "Deep learning for stock market prediction from financial news articles," *2017 IEEE International Conference on CIVEMSA*, pp. 60–65, 2017.

[6] Ronny Luss and Alexandre d'Aspremont, "Predicting abnormal returns from news using text classification," *Quantitative Finance*, vol. 15, no. 6, pp. 999–1012, 2015.

[7] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan, "Deep learning for event-driven stock prediction," *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[8] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan, "Knowledge-driven event embedding for stock prediction," *Proceedings of COLING 2016*, pp. 2133–2142, 2016.

[9] Seyed Mehran Kazemi and David Poole, "Simple embedding for link prediction in knowledge graphs," *Advances in Neural Information Processing Systems*, pp. 4284–4295, 2018.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[11] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," *Proceedings of the 2014 conference on EMNLP*, pp. 1532–1543, 2014.

[13] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri, "Are loss functions all the same?," *Neural Computation*, vol. 16, no. 5, pp. 1063–1076, 2004.

[14] Daniel Povey, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur, "A time-restricted self-attention layer for asr," *2018 IEEE ICASSP*, pp. 5874–5878, 2018.

[15] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, 2017.

[16] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, "Rectifier nonlinearities improve neural network acoustic models," p. 3, 2013.

[17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter, "Self-normalizing neural networks," *Advances in neural information processing systems*, pp. 971–980, 2017.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland, "Textrunner: open information extraction on the web," *Proceedings of Human Language Technologies*, pp. 25–26, 2007.

[20] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni, "Paraphrase-driven learning for open question answering," *Proceedings of the 51st Annual Meeting of the ACL (Volume 1: Long Papers)*, pp. 1608–1618, 2013.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Boyi Xie, Rebecca Passonneau, Leon Wu, and Germán G Creamer, "Semantic frames to predict stock price movement," *Proceedings of the 51st ACL*, pp. 873–883, 2013.

[23] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan, "Using structured events to predict stock price movement: An empirical investigation," *Proceedings of the 2014 Conference on EMNLP*, pp. 1415–1425, 2014.

[24] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.