JDD @ FinCausal 2020, Task 2: Financial Document Causality Detection

Toshiya Imoto
Japan Digital Design, Inc.
toshiya.imoto@japan-d2.com

Tomoki Ito
The University of Tokyo*
m2015titoh@socsim.org

Abstract

This paper describes the approach we built for the Financial Document Causality Detection Shared Task (FinCausal-2020) Task 2: Cause and Effect Detection. Our approach is based on a multi-class classifier using BiLSTM with Graph Convolutional Neural Network (GCN) trained by minimizing the binary cross entropy loss. In the approach, we have not used any extra data source apart from combining the trial and practice dataset. We achieve weighted F1 score to 75.61 percent and are ranked at 7-th place.

1 Introduction

Causal detection is one of the major concrete tasks on Information Extraction (IE) in NLP research. Having a causality in text can be defined as an identification of a pair of sub-strings which explain the cause-effect relationship. How a financial news (political / economical / financial events etc.) links to the other is extremely useful for analysts, investors and risk managers in bank to forecast what will happen in the (near) future in the domestic / global economy and the financial market. FinCausal-2020 Shared Task is associated with a joint workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020). It consists of the two sub-tasks:

- Task 1: Sentence Classification
- Task 2: Cause and Effect Detection

In this paper, we propose a method for solving *Task2: Cause and Effect Detection* (Mariko et al., 2020). To solve this task, we first consider this task to be the sequence labeling task, and then applied the state-of-the-art method for the definition and extraction (DE) (Veyseh et al., 2020) to this task in a reasonable manner. We then evaluate our approach using the shared task 2 dataset. Our approach outperformed most of the other approaches submitted to this shared task. This result demonstrates the effectiveness of our approach.

2 Task and Dataset

We first describe the task setting and dataset organization. The objective of the Task2 is to extract a cause-effect pair from each sample document in the following format.

ID	Text	Cause	Effect
0026.00062	It's risen 106,500% since it came on the scene a split-adjusted $1.50 a share.	it came on the scene a split-adjusted $1.50 a share.	It's risen 106,500%

The below table shows statistics of all the dataset provided by organizer for each in csv file. ”Evaluation” dataset is aimed for the bind test; therefore, it has exactly the same format but both the ”Cause” and ”Effect” columns are left empty.

*Work started during internship at Japan Digital Design, Inc.

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 50–54
Barcelona, Spain (Online), December 12, 2020.

Dataset	#sample	avg #word	avg #word in cause	avg #word in effect
Trial	641	45.20	18.65	18.27
Practice	1109	43.07	17.84	17.56
Evaluation	638	43.07	-	-

Please note that each document can consist of multiple sentences, but limited to less than or equal to five sentences. Also, multiple (different) cause-effect pairs could be found in the same document. In that case, they are distinguished in their IDs by adding an extra separator (ex. 0026.00070.1, 0026.00070.2, 0026.00070.3, and so on).

The prediction results are evaluated using the following weighted F1 score ($= F1_{weighted}$) and the number of exact matches ($= EM$):

$$\begin{aligned} F1_{weighted} &:= w_{cause} \times F1_{cause} + w_{effect} \times F1_{effect} \\ EM &:= \frac{\#\textbf{exact match}}{\#\textbf{dataset samples}} \end{aligned}$$

where
$w_{cause} = \frac{\#\textbf{words in cause golds}}{\#\textbf{words in cause golds} + \#\textbf{words in effect golds}}$ and $w_{effect} = \frac{\#(\textbf{words in effect golds})}{\#\textbf{words in cause golds} + \#\textbf{words in effect golds}}$.

3 Approach

3.1 Problem Definition

We first consider the cause and effect detection task to be the sequence labeling task as follows. Given an input sequence $W = w_1, w_2, \cdots, w_N$ where N is the number of terms in the sequence and w_i is the i-th term in the sequence, the task can be defined as to accurately assign a label l_i to each term w_i in a sequence where $l_i \in \{\mathrm{B-Cause}, \mathrm{B-Effect}, \mathrm{I-Cause}, \mathrm{I-Effect}, \mathrm{Other}\}$. Here, B-Cause and B-Effect mean the beginning of the cause and effect part, respectively, I-Cause and I-Effect mean the inside of the cause and effect part, respectively, and Other means the other parts of the sequence. These label definitions are followed by the BIO tagging scheme.

3.2 Model Architecture

To solve this task, we utilize a neural network (NN) model that includes the following five components: Embedding Layer, bidirectional Long Short Term Memory (BiLSTM) Layer, Graph Convolutional Network1 (GCN) layer, MLP Layer, and Output Layer (Figure 1). This model is inspired by the NN model proposed in (Veyseh et al., 2020), which was proposed to address the definition extraction (DE) task.

3.2.1 Embedding Layer

This layer converts each w_i to its embedding representation $\boldsymbol{x}_i$ where $\boldsymbol{x}_i$ is the concat vector of the word embedding representation of w_i ($= \boldsymbol{e}_i$) and one-hot representation from the POS (Part-of-Speech) tag of w_i ($= \boldsymbol{p}_i$). Here, we used the open-source Wikipedia 2014 + Gigaword 5 Glove embedding[1] for the word embeddings, and analysis results by spacy[2] for the POS tags.

3.2.2 BiLSTM Layer

This layer converts each $\boldsymbol{e}_i$ to word-level contextual representation $\tilde{h}$ using BiLSTM (Schuster and Paliwal, 1997) and Multi Layer Perceptron (MLP) as follows:

$$\boldsymbol{h}_i := \mathrm{BiLSTM}(\boldsymbol{x}_i), \tilde{\boldsymbol{h}}_i := \mathrm{ReLU}(\mathrm{MLP}(\boldsymbol{h}_i)). \tag{1}$$

3.2.3 GCN Layer

This layer converts each $\boldsymbol{h}_i$ to graph embedding representation $\boldsymbol{g}_i$ using GCN (Xu et al., 2018), and MLP as follows:

$$\boldsymbol{g}_i := \mathrm{GCN}(\boldsymbol{h}_i), \tilde{\boldsymbol{g}}_i := \mathrm{ReLU}(\mathrm{MLP}(\boldsymbol{g}_i)). \tag{2}$$

[1] https://nlp.stanford.edu/projects/glove/
[2] https://spacy.io

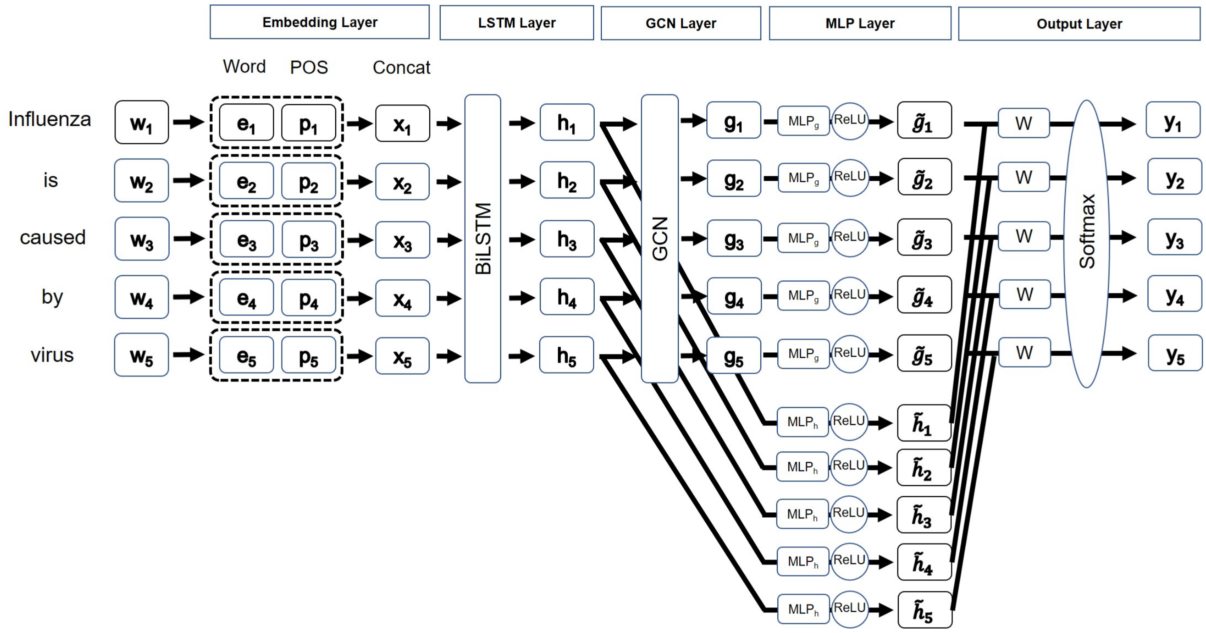

Figure 1: Architecture of our model

3.2.4 Output Layer

Finally, this layer outputs the prediction results as follows:

$$\boldsymbol{a}_i^{B-Cause} := \mathrm{Softmax}(\mathrm{W}^{\mathrm{B-Cause}}[\tilde{\mathrm{h}}_\mathrm{i}, \tilde{\mathrm{g}}_\mathrm{i}]), \mathrm{y}_\mathrm{i}^{\mathrm{B-Cause}} := \mathrm{argmax}\ \mathrm{a}_\mathrm{i}^{\mathrm{B-Cause}} \tag{3}$$

$$\boldsymbol{a}_i^{I-Cause} := \mathrm{Softmax}(\mathrm{W}^{\mathrm{I-Cause}}[\tilde{\mathrm{h}}_\mathrm{i}, \tilde{\mathrm{g}}_\mathrm{i}]), \mathrm{y}_\mathrm{i}^{\mathrm{I-Cause}} := \mathrm{argmax}\ \mathrm{a}_\mathrm{i}^{\mathrm{I-Cause}} \tag{4}$$

$$\boldsymbol{a}_i^{B-Effect} := \mathrm{Softmax}(\mathrm{W}^{\mathrm{B-Effect}}[\tilde{\mathrm{h}}_\mathrm{i}, \tilde{\mathrm{g}}_\mathrm{i}]), \mathrm{y}_\mathrm{i}^{\mathrm{B-Effect}} := \mathrm{argmax}\ \mathrm{a}_\mathrm{i}^{\mathrm{B-Effect}} \tag{5}$$

$$\boldsymbol{a}_i^{I-Effect} := \mathrm{Softmax}(\mathrm{W}^{\mathrm{I-Effect}}[\tilde{\mathrm{h}}_\mathrm{i}, \tilde{\mathrm{g}}_\mathrm{i}]), \mathrm{y}_\mathrm{i}^{\mathrm{I-Effect}} := \mathrm{argmax}\ \mathrm{a}_\mathrm{i}^{\mathrm{I-Effect}} \tag{6}$$

where $\boldsymbol{W}^{B-Cause} \in \mathbb{R}^{2\times(d_h+d_g)}$, $\boldsymbol{W}^{B-Effect} \in \mathbb{R}^{2\times(d_h+d_g)}$, $\boldsymbol{W}^{I-Cause} \in \mathbb{R}^{2\times(d_h+d_g)}$, and $\boldsymbol{W}^{I-Effect} \in \mathbb{R}^{2\times(d_h+d_g)}$. Here, d_h and d_g are the dimension sizes of $\tilde{\boldsymbol{h}}_i$ and $\tilde{\boldsymbol{g}}_i$, respectively. In the above, if $y_i^{B-Effect} = 1$ or $y_i^{I-Effect} = 1$, then, we predict that w_i is included in the effect part; whereas if $y_i^{B-Cause} = 1$ or $y_i^{I-Cause} = 1$, then, we predict that w_i is included in the cause part. Moreover, if $y_i^{B-Effect} = y_i^{I-Effect} = y_i^{B-Cause} = y_i^{I-Cause} = 0$, then, l_i is predicted as $Other$.

3.3 Learning

Our model can be trained using the following $\mathcal{L}$ as a loss function:

$$\mathcal{L} = \sum_{i=1}^{N} BCE(\boldsymbol{a}_i^{B-Cause}, l_i^{B-Cause}) + BCE(\boldsymbol{a}_i^{I-Cause}, l_i^{I-Cause})$$
$$+BCE(\boldsymbol{a}_i^{B-Effect}, l_i^{B-Effect}) + BCE(\boldsymbol{a}_i^{I-Effect}, l_i^{I-Effect})$$

where $BCE(a, b)$ means the binary cross-entropy loss between a and b and l_i^x is defined as follows:

$$l_i^x = \begin{cases} 1 & (x\ \ is\ \ l_i) \\ 0 & (otherwise) \end{cases} \tag{7}$$

Here, it should be noted that we utilize a BCE loss instead of one of the Conditional Random Field (CRF) used in (Veyseh et al., 2020). This is because the BCE loss performed better than the one of CRF in a prior experiment. In addition, it should be noted that utilization of the BCE loss alone could not avoid pathological cases where the predicted B-Cause (Effect) comes after I-Cause (Effect) or the predicted cause phrase overlaps with the effect's one. In the first case, the algorithm gives no cause (effect) label to the document. The second case, the overlap will be labelled as effect phrase. A simple approach to ensure the one and the only one cause-effect pair in a document is to introduce special position embedding used in (Zheng et al., 2017). In Section 4, we evaluate our two models (with and without the position embedding) on the blind dataset.

4 Experimental Evaluation

This section evaluates our approach using the shared task 2 dataset.[3]

4.1 Model Development Setting

We trained our model using the trial and practice datasets, and then submitted the prediction results for the evaluation dataset using the trained model. In this training, we used zero padding where the padding size was 200. Moreover, the number of layers in the BiLSTM layer and dimension size of hidden vectors were 1 and 50, respectively.

4.2 Comparison Approach

To evaluate our approach, we compare the results of the following four approaches:

- Baseline (CRF): CRF based classifier using pycrfsuite provided by organizer [4],
- BiLSTM + GCN + CRF: state-of-the-art DE model (Veyseh et al., 2020) manually adapted to our task.
- BiLSTM + GCN + BCE Loss: our model explained in section 3, and
- BiLSTM + GCN + BCE Loss with Position Embed: our model with the position embedding explained in section 3.

4.3 Results

Table 1 shows the results, demonstrating that our approach outperformed both the Baseline (CRF) and BiLSTM + GCN + CRF at the weighted F1 score. These results indicate the effectiveness of our approach.

Model	Precision	Recall	Weighted F1	EM
Baseline (CRF)	50.99%	51.74%	51.06%	11.11%
BiLSTM+GCN+CRF	72.61%	72.12%	72.29%	0.00%
BiLSTM+GCN+BCE Loss (Ours)	**75.95%**	75.57%	**75.61%**	0.00%
BiLSTM+GCN+BCE Loss with Position Embed (Ours)	75.80%	**76.60%**	75.29%	**53.45%**

Table 1: Evaluation Result for the shared task 2

5 Conclusion

In this paper, we propose a method for extracting cause and effect parts from texts. We first consider this task to be the sequence labeling task and then, applied the method for DE to this task in a reasonable manner. Our approach was ranked at 7-th. In future, our approach can be improved by utilizing the pre-trained language models (e.g., BERT) or transfer learning approaches.

[3] For all the experiments in this section, we used the hardware with GPU cores GeForce GTX 1660 and with RAM 16.0GB.
[4] `https://github.com/yseop/YseopLab/tree/develop/FNP_2020_FinCausal/baseline/task2`

References

Dominique Mariko, Hanna Abi Akl, Estelle Labidurie, Stephane Durfort, Hugues de Mazancourt, and Mahmoud El-Haj. 2020. The Financial Document Causality Detection Shared Task (FinCausal 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020)*, Barcelona, Spain.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Amir Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. A joint model for definition extraction with syntactic connection and semantic consistency. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 9098–9105.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *Proceedings of ICML 2018*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1227–1236.