Proceedings of the 4th Financial Narrative Processing Workshop @ LREC 2022, pages 124–127
Marseille, 24 June 2022

iLab at FinCausal 2022: Enhancing Causality Detection with an External Cause-Effect Knowledge Graph

Ziwei Xu[1], Rungsiman Nararatwong[1], Natthawut Kertkeidkachorn[2], Ryutaro Ichise[3,1]
[1]National Institute of Advanced Science and Technology, Japan
[2]Japan Advanced Institute of Science and Technology, Japan
[3]Tokyo Institute of Technology, Japan
xuxiaowei23@hotmail.com, r.nararatwong@aist.go.jp, natt@jaist.ac.jp, ichise@iee.e.titech.ac.jp

Abstract

The application of span detection grows fast along with the increasing need of understanding the causes and effects of events, especially in the finance domain. However, once the syntactic clues are absent in the text, the model tends to reverse the cause and effect spans. To solve this problem, we introduce graph construction techniques to inject cause-effect graph knowledge for graph embedding. The graph features combining with BERT embedding, then are used to predict the cause effect spans. The results show our proposed graph builder method outperforms the other methods w/wo external knowledge.

Keywords: Graph builder, Cause effect graph, BERT

1. Introduction

Understanding cause and effect in financial documents help us to comprehend the movement of the financial market. Nevertheless, manual annotation is not feasible due to the massive volume of published financial papers. It is necessary to develop an automatic causality extraction method to facilitate financial analysis. Therefore, FinCausal (Mariko et al., 2020b) has been proposed to be the benchmark for causality extraction in the finance domain. The task description of FinCausal 2022 is a relation detection task where we need to identify a causal sentence or text block, the causal elements, and the consequential ones in a given sentence. For example, Given the sentence *"Zhao found himself 60 million yuan indebted after losing 9,000 BTC in a single day (February 10, 2014)"*, we could identify *"losing 9,000 BTC in a single day (February 10, 2014)"* as the cause while we annotate *"Zhao found himself 60 million yuan indebted)"* as the effect.

Recently, many methods have been proposed for FinCausal (Mariko et al., 2020a; Mariko et al., 2021). In FinCausal 2021, the system named DTGNN (Tan and Ng, 2021) achieved the best results in this task. DTGNN incorporates dependency relation features from a sentence through a graph neural network into BERT (Devlin et al., 2018) token classifier with Viterbi decoding (Kao et al., 2020). As a result, the system mainly focuses on adding syntactic features by the dependency features. However, the cause-effect relation of tokens is not explored yet. In this paper, we present our approach built on top of DTGNN and incorporate the cause-effect relation of tokens. We utilize external knowledge, particularly cause and effect graph in the financial domain (Li et al., 2021), to provide the cause-effect relation.

The rest of the paper is organized as follows. We presented our system in Section 2. In Section 3, we discussed our experiment and results. This paper is concluded in Section 4.

2. Proposed System

The competition task in FinCasual 2022 is to detect the cause span and effect span from a given textual span. The BIO scheme tags the **B**eginning tokens and **I**nner tokens in the objective spans and tags **O**thers in the rest of the string. The scheme defines this task as token classification, typically applied to Named Entity Recognition and Span Detection in the state-of-art methods.

2.1. Baseline

In previous competitions, we noticed the highlighted framework, DTGNN, proposed by the winner of Fincausal 2021 (Tan and Ng, 2021). This framework is composed of different functional modules attached to BERT architecture. We follow the major modules as shown in Figure 1: BERT encoder, graph builder, GNN+BiLSTM and Viterbi Decoder. Our contribution is mainly located in the graph builder module and GNN modules.

2.1.1. Graph Builder and GNN

In the baseline, the graph builder generates a subgraph for each textual span. The SAGEConv (Hamilton et al., 2017) operator embeds the subgraph into feature representations. In this way, the weights of edges in subgraph are neglected.

In our proposal, during the graph building process, we add the knowledge from Cause-Effect Graph [1] (CEG)(Li et al., 2020) in different manners. Then each subgraph would feed to the graph neural network (GNN), which contains two graph convolutional layers with GCNConv (Kipf and Welling, 2017) operator. In this way, not only do the connected nodes matter for

[1]`https://github.com/eecrazy/CausalBank`.

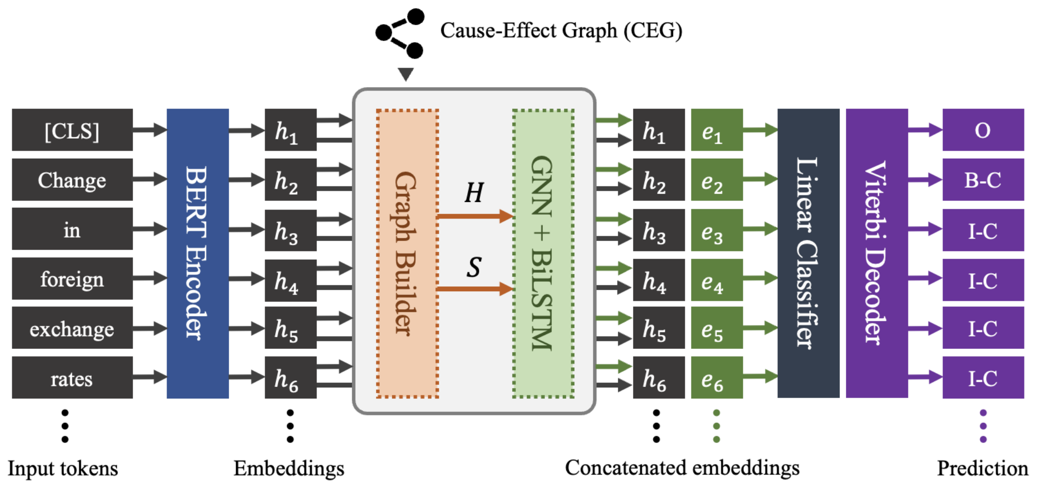


Figure 1: Our method consists of four main components: The BERT encoder, graph builder, GNN+BiLSTM, and Viterbi decoder. We proposed two approaches for the graph builder, which constructs a subgraph for GNN using the external cause-effect graph.

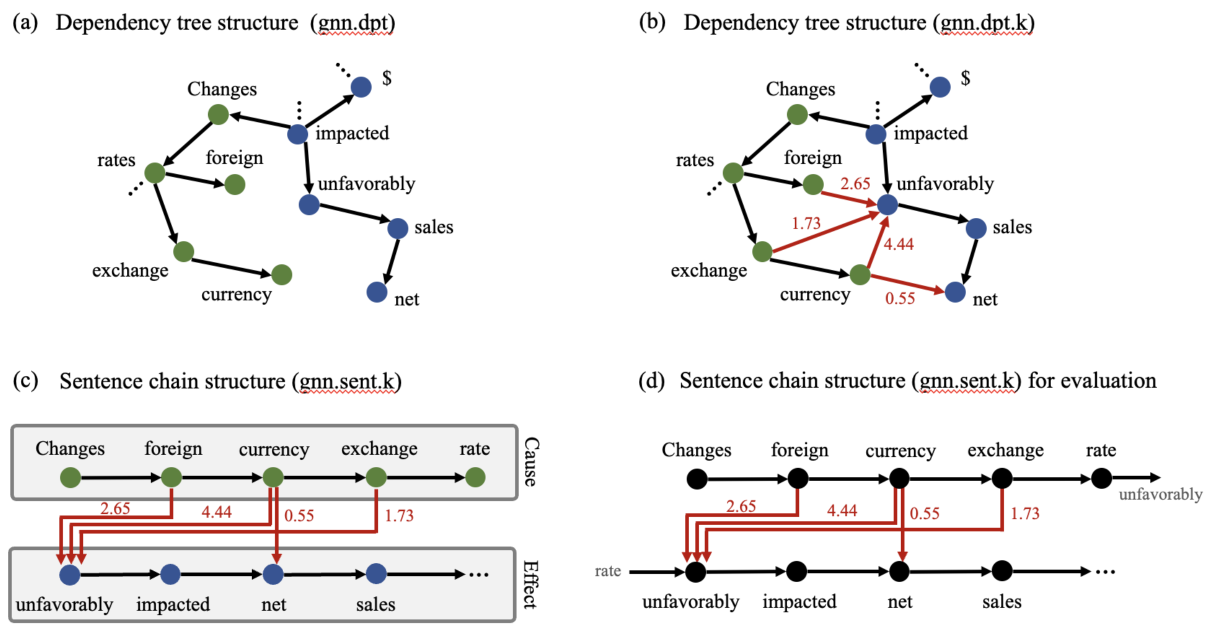

(a) Dependency tree structure (gnn.dpt)
(b) Dependency tree structure (gnn.dpt.k)
Changes
$
impacted
rates
foreign
unfavorably
sales
exchange
currency
net
2.65
1.73
4.44
0.55
(c) Sentence chain structure (gnn.sent.k)
(d) Sentence chain structure (gnn.sent.k) for evaluation
Changes
foreign
currency
exchange
rate
Cause
Effect
unfavorably
impacted
net
sales
rate

Figure 2: Our graph builder generates four types of sub-graph: (a) The original dependency tree-based subgraph in DTGNN, (b) the same subgraph with additional edges and weights (highlighted in red) for cause-effect relationships, (c) cause and effect chains with cause-effect edges for training, and (d) a single chain connecting the whole input sequence (except stopwords) for inference. All black edges in (b), (c), and (d) have the same small weight (0.1). The green nodes are tokens in the cause span, and the blue nodes are tokens in the effect span.

message passing, but the weight of connections is taken into account.

2.1.2. Viterbi Decoder

For token classification tasks, labels tend to be independent and discontinuous. Viterbi decoder (Kao et al., 2020) solves this problem by using the transition and emission matrices for those labels during the evaluation step, which correct predictions for continuous span labels. Thus we will applly this techniques as well in our framework.

2.2. Structure of knowledge injection

We expect this framework to use linguistic features to train the model. However, once the textual spans do not include the clear syntactic clues, e.g., *because* or *as*, the prediction of cause and effect spans can be reversed, resulting in the wrong prediction. We resort to injecting extra knowledge to distinguish cause spans and effect spans. Cause-Effect Graph (Li et al., 2020) stores the causality relations and weights between tokens pairs. In sentences, cause spans could potentially contain the tokens that have directed causality to the tokens inside effect spans. This section proposes two approaches to insert the target knowledge into embeddings using graph neural networks: sentence chain structure and dependency tree structure.

2.2.1. Dependency Tree Structure

For dependency tree structure, one sentence could be organized by the dependency tree relations, as shown in Figure 2(a). In this manner, tokens in the same sub-sentence tend to connect closer than those in the opposite. To insert causality knowledge, in Figure 2(b), we add the directed linkage between causality token pairs and assign weights for these relations. For the weights of dependency tree connections, we would equally assign them with the same low value (e.g., 0.1). Comparatively, the structure idea of the dependency tree has been implemented by previous work (Tan and Ng, 2021). In their work, the relation weights between tokens are not considered, which neglects much useful information.

2.2.2. Sentence Chain Structure

In sentence chain structure, cause span and effect span are separated into two chains. In each chain, tokens are linked to reserve their orders from beginning to end. In Figure 2(c), between two chains, the tokens holding causality relations are connected unidirectionally with the corresponding weights. As for the connection inside a chain, their weights are equally assigned with the same low value (e.g., 0.1). However, in the validation or test steps, we transform the textual span into an entire chain because of the lack of labels, the causality tokens pairs would be connected with the intra-chain links in Figure 2(d).

3. Experiments and Results

3.1. Data Preparation

We combined 2020, 2021, and 2022 versions of the FinCausal dataset for training, including both *practice* and *trial* sets. As we noticed several duplicate samples, we searched for and removed those with the exact input text and answers to ensure the reliability of our cross-validation data, resulting in 2,775 samples. We then split the reduced dataset into ten folds, nine of which were for training (2,497 samples) and one for validating (278 samples). While a sample may have multiple answer spans, the dataset format (csv) does not allow flexible multi-span labeling. As a result, a sample with multiple answers is split into multiple samples with the same input text. Therefore, we merged these samples and obtained the final 2,290 training samples and 255 samples for validation.

3.2. Replication Settings

Device and Time We used NVIDIA RTX 3090-24G, and it took 1h11min to simultaneously train three models, refer to the idea presented in sub-figure a,b,c of Figure 2. The best scores were achieved with random seed 123, 456, 123 for these models respectively.

Hyperparameter The pre-trained BERT model (bert-base-cased) is initialized by Huggingface [2]. All models were trained with ten epochs, learning rate 5e-5, and dropout 0.1. The maximum sequence length is set to 350, and the train batch size is 4. For GNN, the hidden and out graph dimension is 1024 and 512, respectively.

	F1	**Recall**	**Precision**	**EM**
gnn.dpt	93.58	93.56	93.66	82.53
gnn.dpt.k	93.41	93.37	93.50	81.99
gnn.sent.k	<u>93.90</u>	<u>93.89</u>	<u>93.95</u>	<u>82.64</u>

Table 1: The best scores achieved on blind test set.

	F1	**Recall**	**Precision**	**EM**
gnn.dpt	89.70	89.66	89.77	73.38
gnn.dpt.k	88.38	88.36	88.42	73.02
gnn.sent.k	<u>90.22</u>	<u>90.20</u>	<u>90.25</u>	<u>75.90</u>

Table 2: The best scores achieved on our validation set.

3.3. Results

Table 1 shows the best results in the test set. We notice that all models achieve similar high scores, but the sentence chain structure (*gnn.sent.k*) outperforms others by around 0.3% to 0.5% F1 score. As for knowledge injection variant *gnn.dpt.k*, it fails to improve the performance with the addition of extra knowledge compared to the original dependency tree structure (*gnn.dpt*). To sum up, the knowledge injection works well on our proposed sentence chain structure but not on the dependency tree structure. Ultimately, the inclusion of the Cause-Effect Graph in our proposed graph builder enhances the performance of span prediction tasks.

It is also worth mentioning that Table 2 shows the best scores achieved in the validation set. Surprisingly, we got lower values than the test set in general. We attribute this variance to the different evaluation metrics, in which the scikit-learn metrics that we applied in the validation set have stricter rules than used in competition.

Moreover, we experimented with 3-fold Cross-Validation (CV) and anticipated achieving higher performance. The precision on the train and validation set can be as high as 98% precision and 97% F1 score. However, on the test set, these models cannot reach the peak. They lay behind around 1% of those models without CV. Given these points, the application of CV introduces the over-fitting problem. It is not unsuitable for our models.

[2] `https://huggingface.co/`

4. Conclusion

We focus on generating better graph embedding with the proposed graph builders. Accordingly, the sentence chain structure with the injection of the Cause-Effect Graph outperforms the other structures w/wo knowledge, which helps distinguish between cause spans and effect spans. In the future, we will attempt to inject knowledge into different GNN variants to find the optimal way for knowledge embedding.

5. Bibliographical References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hamilton, W. L., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *NIPS*.

Kao, P.-W., Chen, C.-C., Huang, H.-H., and Chen, H.-H. (2020). NTUNLPL at FinCausal 2020, task 2:improving causality detection using Viterbi decoder. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 69–73, Barcelona, Spain (Online), December. COLING.

Kipf, T. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907.

Li, Z., Ding, X., Liu, T., Hu, J. E., and Van Durme, B. (2020). Guided generation of cause and effect. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3629–3636. International Joint Conferences on Artificial Intelligence Organization, 7. Main track.

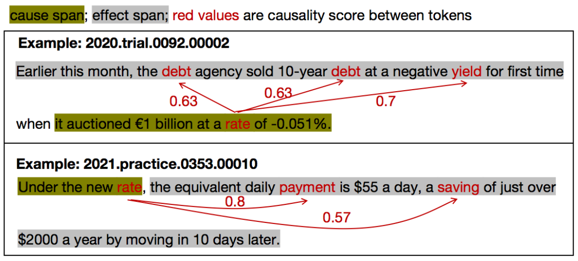
Li, Z., Ding, X., Liu, T., Hu, J. E., and Van Durme, B. (2021). Guided generation of cause and effect. *arXiv preprint arXiv:2107.09846*.

Mariko, D., Abi-Akl, H., Labidurie, E., Durfort, S., De Mazancourt, H., and El-Haj, M. (2020a). The financial document causality detection shared task (FinCausal 2020). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 23–32, Barcelona, Spain (Online), December. COLING.

Mariko, D., Labidurie, E., Ozturk, Y., Akl, H. A., and de Mazancourt, H. (2020b). Data processing and annotation schemes for fincausal shared task. *arXiv preprint arXiv:2012.02498*.

Mariko, D., Akl, H. A., Labidurie, E., Durfort, S., de Mazancourt, H., and El-Haj, M. (2021). The financial document causality detection shared task (FinCausal 2021). In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 58–60, Lancaster, United Kingdom, 15-16 September. Association for Computational Linguistics.

Tan, F. A. and Ng, S.-K. (2021). NUS-IDS at FinCausal 2021: Dependency tree in graph neural network for better cause-effect span detection. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 37–43, Lancaster, United Kingdom, 15-16 September. Association for Computational Linguistics.


cause span; effect span; red values are causality score between tokens

Example: 2020.trial.0092.00002

Earlier this month, the debt agency sold 10-year debt at a negative yield for first time when it auctioned €1 billion at a rate of -0.051%.

Example: 2021.practice.0353.00010

Under the new rate, the equivalent daily payment is $55 a day, a saving of just over $2000 a year by moving in 10 days later.

Figure 3: The example for which the model(*gnn.sent.k*) predict correctly.

6. Appendix

This section shows the typical examples when the models (*gnn.dpt* and *gnn.dpt.k*) mix up the cause span and effect span in Figure 3. In the opposite, with the addition of cause-effect knowledge, the model (*gnn.sent.k*) trained on sentence chain structure are able to predict the cause and effect span correctly.