DATA PAPER

A Joint Learning Framework for the CCKS-2020 Financial Event Extraction Task

Jiawei Sheng[1,2], Qian Li[3], Yiming Hei[4], Shu Guo[5†], Bowen Yu[1,2], Lihong Wang[5†], Min He[5], Tingwen Liu[1,2] & Hongbo Xu[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

[3]School of Computer Science, Beihang University, Beijing 100191, China

[4]School of Cyber Science and Technology, Beihang University, Beijing 100191, China

[5]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China

Keywords: Event detection; Event extraction; Joint learning; Financial event

Citation: Sheng, J.W., et al: A joint learning framework for the CCKS-2020 financial event extraction task. Data Intelligence 3(3), 444-459 (2021). doi: 10.1162/dint_a_00098

Received: February 13, 2021; Revised: April 22, 2021; Accepted: April 30, 2021

ABSTRACT

This paper presents a winning solution for the CCKS-2020 financial event extraction task, where the goal is to identify event types, triggers and arguments in sentences across multiple event types. In this task, we focus on resolving two challenging problems (i.e., low resources and element overlapping) by proposing a joint learning framework, named SaltyFishes. We first formulate the event extraction task as a joint probability model. By sharing parameters in the model across different types, we can learn to adapt to low-resource events based on high-resource events. We further address the element overlapping problems by a mechanism of Conditional Layer Normalization, achieving even better extraction accuracy. The overall approach achieves an *F*1-score of 87.8% which ranks the first place in the competition.

1. INTRODUCTION

The CCKS-2020 financial event extraction task[①] aims at extracting structural events by identifying event types, triggers and arguments in sentences across multiple types. Figure 1 gives an example of event

† Corresponding authors: Shu Guo (Email: guoshu@cert.org.cn; ORCID: 0000-0002-9660-0291) and Lihong Wang (Email: wlh@isc.org.cn; ORCID: 0000-0003-0179-2364).

① https://www.biendata.xyz/competition/ccks_2020_3/

extraction for a financial news sentence. One structural event belongs to the type of 投资/investment, along with the trigger 收购/acquire and its arguments providing more complementary details. Note that this sentence contains more than one event, and the trigger and arguments have overlaps across the events.

The CCKS-2020 task provides two kinds of such event sentences. The first one contains five types of events associated with an abundant sentence corpus, called source events. The second one contains another five types of events associated with low-resource sentence corpus, called target events. Each type of event sentence is split into training (labeled data) and testing (unlabeled data) parts. Our goal is to evaluate the performance of event extraction on the test set of target events. This poses two main challenges compared to the traditional event extraction tasks [1,2,3,4,5]:

- The target events contain only 179 training sentences on average for each type. This limited supervision information cannot provide sufficient contextual information for the event extraction.
- Elements can be overlapped with each other, i.e., the same trigger or argument may belong to different events. As shown in the example, the trigger 收购/acquire and the argument 世纪华通/Shijihuatong belong to both event types of 投资/investment and 股份股权转让/share transfer. Performing event extraction by a simple sequence labeling method will cause label conflicts.

 # Sentence
世纪华通/ 作价/ 298.03亿元/ 收购/ 盛跃网络/ 100%/ 股权。
Shijihuatong/set a price of/ 29.803 billion yuan/ to acquire/ Shengyue Network's/ 100% equity.

 # Event 1
Event Type: 投资/ investment
Trigger: 收购/ acquire
Arguments
Sub-company: 世纪华通/ Shijihuatong
Obj-company: 盛跃网络/ Shengyue Network
Money: 298.03亿元/ 29.803 billion yuan

 # Event 2
Event Type: 股份股权转让/ share transfer
Trigger: 收购/ acquire
Arguments
Sub-company: 盛跃网络/ Shengyue Network
Obj-company: 世纪华通/ Shijihuatong
Money: 298.03亿元/ 29.803 billion yuan
Proportion: 100%/ 100%
Collateral: 股权/ equity

Figure 1. Example of event extraction with element overlapping problem.

To address these challenges, we devised a joint learning method. In our approach, the overall framework is formulated as a joint probability model, which is decomposed into submodels, i.e., the joint distribution is decomposed into a product of three conditional distributions. Each subtask will be a specific use of this distribution, including event type detection, trigger extraction and argument extraction.

For the first subtask, given a financial news sentence, we first classify the sentence into a correct event type by using a multi-class multi-label text classification paradigm. For the other two subtasks, we successively extract triggers and arguments with a pre-training/fine-tuning framework. The pre-training module is implemented by a pre-trained language model BERT [6] on all financial news sentences, and we further fine-tune the pre-trained model with respect to the trigger/argument extraction module. To deal with the overlapping element issue, we introduce conditional layer normalization, only extracting triggers according to the specific event type, and extracting arguments according to the specific trigger. This method can extract elements separately in different conditions, avoiding overlapping. In addition, by sharing parameters across different types in such a unified model, we can learn to adapt to low-resource events based on high-resource events. Our approach achieved an $F1$-score of 87.8% which ranked the first in the CCKS-2020 financial event extraction competition.

2. RELATED WORK

Traditional event extraction research is usually achieved in the high-resource setting, and assumes events appearing in sentences without overlapped elements. These studies can be roughly categorized into the following two groups:

1) Traditional joint methods [4,5,7,8,9] that perform trigger extraction and argument extraction simultaneously. They solve the task in a sequence labeling manner, and extract triggers or arguments by tagging the sentence only once. However, these methods fail in extracting overlapped elements since the overlapped elements will cause label conflicts when forced to have more than one label.
2) Pipeline methods [1,10,11,12,13,14] that perform trigger extraction and argument extraction in separate stages. Though the pipeline methods enable extracting overlapped elements in separate stages [14], they usually lack explicit dependencies between the triggers and arguments and suffer from error propagation. All the above methods require sufficient training data to learn model parameters for each event type, and few methods can extract complex overlapped elements in event extraction.

Recently, several methods were proposed to solve event extraction in several kinds of low-resource setting, such as few-shot learning setting [2], zero-shot learning setting [3,12] and incremental learning setting [15]. However, these methods cannot be directly transferred to this CCKS-2020 financial event extraction task, for the reason that this task aims at extracting low-resource target events with the help of rich-resource source events, which is a completely different setting comparing to the above low-resource event extraction settings.

3. OUR APPROACH

We will present the overview, the design of each component, and some strategies for improvements.

3.1 Overview

Given a sentence denoted as s, we proposed a joint learning approach to identify its event types C, event triggers T and event arguments A. The approach is formulated as a joint probability model, which is decomposed into three submodels with respect to the event type detection, the event trigger extraction and the event argument extraction:

$$P(C, T, A|s; \Theta) \propto P(C|s; \Theta_1)\, P(T|s, C; \Theta_2, \Theta_3)\, P(A|s, C, T; \Theta_2, \Theta_4)$$

The event type detection is modeled by a multi-class multi-label text classification paradigm, where Θ_1 is the set of type detection model parameters. The other two extraction parts are modeled by a pre-training/fine-tuning framework, where Θ_2 contains model parameters shared by both modules, while Θ_3 and Θ_4 are respective private model parameters. All parameters in $\Theta \triangleq \{\Theta_1, \Theta_2, \Theta_3, \Theta_4\}$ are used across different event types (either high-resource or low-resource), which promotes rich interactions between source events and target events. Figure 2 sketches the overall framework.

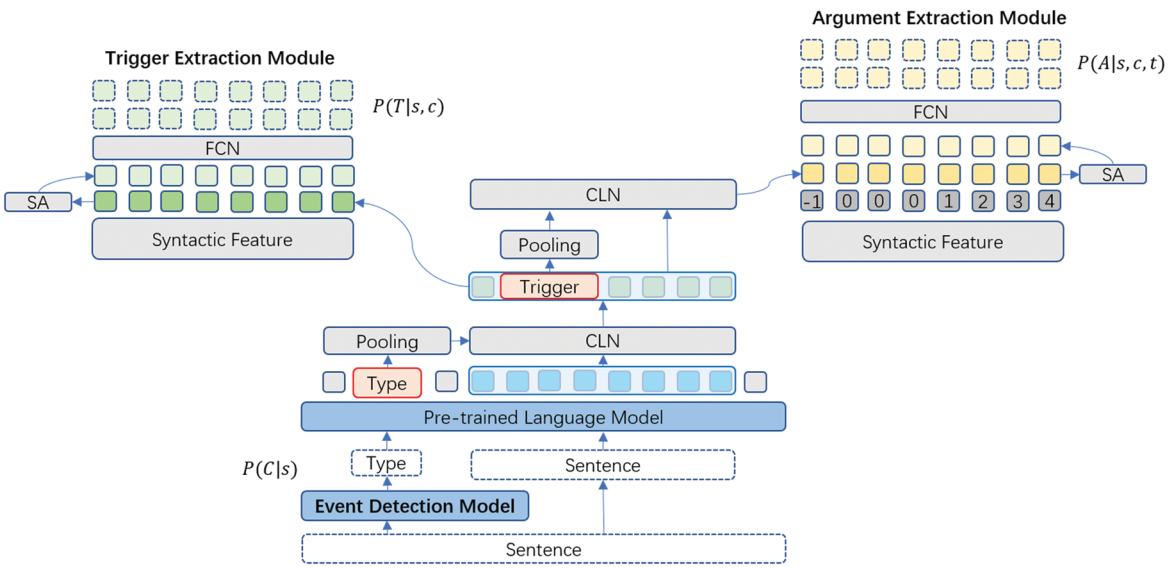

Figure 2. The overall framework of the financial event extraction approach.

3.2 Event Detection Model

In order to discover the event types occuring in the sentence, we adopted codes provided by official competition[2] as our event detection model (EDM). This model utilizes a pre-trained language model (PLM) to derive sentence representations, formulated as a multi-label multi-class text classification task. Specifically, given the sentence s, the probability of s belonging to a specific type c_m is calculated as in Equation (1):

[2] https://github.com/xyionwu/ccks-2020-finance-transfer-ee-baseline

$$p(c_m|s;\Theta_1) = \text{sigmoid}(\boldsymbol{w}_m \cdot \boldsymbol{z}_{sent}) \quad (1)$$

where $\boldsymbol{z}_{sent}$ is the hidden state corresponding to the input token <CLS> in the PLM, which encodes the entire sentence representation of s; Θ_1 includes all parameters used in the PLM. For simplicity, we denoted $p(c_m|s;\Theta_1)$ as p_m.

Then, we can update and obtain the desired sentence representation $\boldsymbol{z}_{sent}$ by minimizing following binary cross entropy loss function with Equation (2):

$$Loss(\Theta_1) = \sum_{n=1}^{N}\sum_{m=1}^{M} y_{nm} * log(p_{nm}) + (1-y_{nm}) * log(1-p_{nm}) \quad (2)$$

where N is the number of the training sentences; M is the number of the pre-defined event types; y_{nm} is the true type label, which is either 0 or 1. During prediction, we simply set a threshold δ and selected the resultant event types C where each type c_m such that $p(c_m|s;\Theta_1) > \delta$.

3.3 Event Extraction Model

This section introduces our event extraction model (EEM), achieving two subtasks by a pre-training/fine-tuning framework: trigger extraction and argument extraction. The pre-training part encodes sentence tokens as contextualized representations with the pre-trained language model, BERT [6], which contains rich language knowledge widely used for natural language processing (NLP) tasks. The fine-tuning part is divided into three modules, including a shared module to encode condition information based on conditional layer normalization, and two private modules to extract triggers and arguments. Note that both extraction modules have a similar model structure.

3.3.1 Shared Module

This section introduces a sentence representation layer shared by both trigger extraction and argument extraction, which will derive a conditional sentence representation H_{s-typ} for the specific event type c, and a syntactic feature representation H_{syn}.

Since we have obtained event types C occuring in the given sentence s, we are going to derive sentence representations conditioned on each specific event type $c \in C$, so as to avoid element overlapping issues. To this end, we introduced a general module, named conditional layer normalization (CLN) [16,17], to integrate such conditional information into sentence representation. CLN is mostly based on the well-known layer normalization [18], but can dynamically generate gain γ and bias β based on the condition information. Given a condition representation c and a sentence representation x, CLN is formulated in Equations (3) to (5):

$$CLN(x,c) = \gamma_c \odot \left(\frac{x-\mu}{\sigma}\right) + \beta_c \quad (3)$$

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i, \sigma = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2} \tag{4}$$

$$\gamma_c = W_\gamma c + b_\gamma, \beta_c = W_\beta c + b_\beta \tag{5}$$

where x_i is the i-th dimension element in x, $\gamma_c \in R^d$ and $\beta_c \in R^d$ are the conditional gain and bias, respectively. In this way, the given condition representation is encoded into the gain and bias, and then integrated into the contextual representations.

Then we utilized CLN to integrate event type information into the sentence. Specifically, we first transformed the event type's name into textual tokens, such as the type 投资/investment which was transformed into tokens 投 and 资. Then we concatenated these type tokens together with the word tokens in the sentence s, forming a sequence as X : <CLS> + type tokens + <SEP> + word tokens + <SEP>. The sequence was input into the PLM to derive its contextualized representations, and we termed the representations corresponding to type tokens as H_c and word tokens as H_s. Then, we fused H_c with mean pooling and H_s together, to derive the conditional token representations for s with Equation (6):

$$H_{s-typ} = \mathrm{CLN}\left(H_s, MeanPooling\left(H_c\right)\right) \tag{6}$$

where H_{s-typ} is the token representations conditioned on the event type c. Such process generates type-aware token representations adaptive to various event types. As such, we can perform trigger extraction and argument extraction in the independent context of each type.

3.3.2 Private Module

The private module contains the following two sub-modules.

(1) Trigger Extraction Module (TEM)

This module extracts event triggers given the event type $c \in C$. In order to improve textual representations for trigger extraction, we adopted a self-attention (SA) layer. Thus the type-aware token representations can be enhanced as in Equations (7) and (8):

$$H_{sa-typ} = SA\left(H_{s-typ}\right) \tag{7}$$

$$H_{tri} = H_{s-typ} \oplus H_{sa-typ} \oplus H_{syn} \tag{8}$$

where $\oplus$ is the concatenation operation. H_{syn} corresponds to the representation of syntactic features, obtained by NLP tool LTP[③]. The syntactic features include B/I/O labels of word segmentation, part-of-speech tagging, named entity recognition and dependency parsing, which are initialized randomly as learnable embeddings in our model.

[③] http://ltp.ai/

In order to strengthen interactions among triggers of different event types, we predicted triggers with the same trigger extractor. For each token, we adopted a pair of fully connected networks (FCN) to predict whether it is a "begin" or "end" position of a trigger as summarized in Equations (9) and (10):

$$p\left(t^{(b)}|x_i,c;\Theta_2,\Theta_3\right)=sigmoid\left(w_{t^{(b)}}*h_{tri,i}\right) \tag{9}$$

$$p\left(t^{(e)}|x_i,c;\Theta_2,\Theta_3\right)=sigmoid\left(w_{t^{(e)}}*h_{tri,i}\right) \tag{10}$$

where $h_{tri,i}$ is the i-th element of h_{tri}, $w_{t^{(b)}}$ and $w_{t^{(e)}}$ are learnable parameters, and Θ_3 includes $w_{t^{(b)}}$, $w_{t^{(e)}}$, and parameters in SA.

Then, a binary cross entropy loss function was used for begin position prediction and end position prediction, denoted as $Loss_{t^{(b)}}$ and $Loss_{t^{(e)}}$. The final loss is defined as in Equation (11):

$$Loss_{tri}\left(\Theta_2,\Theta_3\right)=w_t*Loss_{t^{(b)}}\left(\Theta_2,\Theta_3\right)+\left(1-w_t\right)*Loss_{t^{(e)}}\left(\Theta_2,\Theta_3\right) \tag{11}$$

where $w_t \in (0,1)$ is a trade-off factor. For prediction, we simply set a threshold ϕ_{tri}, and selected positions such that their prediction scores are higher than ϕ_{tri}. We matched the begin position with the nearest end position to obtain a complete trigger. The final trigger extraction results formed the trigger set T.

(2) Argument Extraction Module (AEM)

This module is to extract arguments conditioned on one of the triggers T extracted from the TEM. Given a specific trigger $t \in T$ in the sentence s, we obtained trigger-aware sentence representation H_{s-tri} conditioned on t, where the process was the same as Equation (7). We also utilized self-attention layer to enhance the sentence representation, termed as H_{sa-tri}. To discern the position of trigger t, we further added its relative position embedding R, which measured the distance from current position to the trigger position. The syntactic feature H_{syn} was also taken into consideration. Thus, the enhanced sentence overall representation is defined as in Equation (12):

$$H_{arg}=H_{s-tri}\oplus H_{sa-tri}\oplus R\oplus H_{syn} \tag{12}$$

For argument extraction, we extracted all arguments with pairs of FCNs and devised them as in Equations (13) and (14):

$$p\left(a_k^{(b)}|x_i,c,t;\Theta_2,\Theta_4\right)=sigmoid\left(w_{a_k^{(b)}}*h_{arg,i}\right) \tag{13}$$

$$p\left(a_k^{(e)}|x_i,c,t;\Theta_2,\Theta_4\right)=sigmoid\left(w_{a_k^{(e)}}*h_{arg,i}\right) \tag{14}$$

where $h_{arg,i}$ is the i-th element of H_{arg}. $w_{a_k^{(b)}}$ and $w_{a_k^{(e)}}$ are learnable parameters for the k-th argument role. Θ_4 includes $w_{a_k^{(b)}}$, $w_{a_k^{(e)}}$, and all the parameters in the SA and CLN. Note that the number of the pre-defined argument roles is K, and there are $2K$ prediction sequences for all the argument roles.

The loss function is also binary cross entropy for both begin and end position prediction for each argument role and it is calculated with Equation (15):

$$Loss_{arg}(\Theta_2,\Theta_4) = \sum_{k=1}^{K} w_a * Loss_{a_k^{(b)}}(\Theta_2,\Theta_4) + (1-w_a) * Loss_{a_k^{(e)}}(\Theta_2,\Theta_4) \quad (15)$$

where $w_t \in (0,1)$ is a tradeoff factor. For prediction, we simply set a threshold ϕ_{arg}, and selected positions were prediction score is higher than ϕ_{arg}. We matched the begin position with the nearest end position to obtain a complete argument. We removed the redundant argument types for each event type based on the event schema constrain. The final results formed the argument set A.

3.3.3 Training and Prediction

To jointly learn the TEM and AEM, we combined both losses from the two modules as in Equation (16):

$$Loss(\Theta_2,\Theta_3,\Theta_4) = w_j * Loss_{trl}(\Theta_2,\Theta_3) + (1-w_j) * Loss_{arg}(\Theta_2,\Theta_4) \quad (16)$$

where $w_j \in (0,1)$ is a weight hyperparameter to balance the two modules.

We utilized ground-truth labels to train the overall model. For prediction, we first obtained trigger extraction results, and then input them into the argument extraction module. The results obtained from the two modules were returned as the final predictions.

3.4 Additional Improvement Strategy

Despite the training data sets, the unlabeled data in the testing data sets also contain rich information. In order to exploit all the data to improve the performance, we also employed the following strategies:

(1) Continuing Pre-training on PLM

PLMs are usually pre-trained on the common corpus, which may cause semantic bias on the financial corpus. Therefore, we continued pre-training the PLM on all the financial data, including training data and testing data. This strategy was applied to both EDM and EEM.

(2) Model Ensemble on Variant Data Splits

To fully exploit labeled data, we adopted K-fold validation on the labeled data, leading to K models trained on different data splits. Then, we ensembled K model predictions by the voting strategy. This model ensemble strategy was applied to EDM and EEM separately.

(3) Utilizing Pseudo-Labels on Unlabeled Data

To fully exploit unlabeled data, we employed a novel strategy to label testing data with pseudo labels. Specifically, we trained models on the ground-truth data, and then predicted labels on those unlabeled data, called pseudo-labeled data. By integrating the pseudo-labeled data into the ground-truth data, we obtained a mixed training data set. We trained new models on this mixed data set. Note that this strategy was only used for EEM, where we achieved better performance.

4. EXPERIMENT

This section introduces the data set provided in the competition, and conducts experiments to evaluate the model.

4.1 Data Set

The data set provided in the competition contains source event data and target event data, including labeled data and testing data for each event type. The statistics of each event type of labeled data is shown in Table 1. The competition only evaluated on the testing target event data. For validation, we separated a part of labeled data as validation data. The details of data partition are shown in Table 2. Since the ground truth of the testing data was not available, all experiments below were conducted on the validation data.

Table 1. Statistics of each event type in the data set.

Source type	质押 pledge	投资 investment	股份股权转让 share transfer	高管减持 reduction	起诉 prosecution
Data size	815	1083	1581	670	533
Target type	收购 acquisition	判决 judgment	中标 win bid	签署合同 sign contract	担保 guarantee
Data size	200	200	200	132	163

Table 2. Data partition for training, validation and testing.

	Training	Validation	Testing
Source type	2,459	273	163,763
Target type	738	82	93,610

4.2 Implementation

We utilized an extended BERT model as our PLM, which was pre-trained on the mixed large Chinese corpus[④] (termed as BERT-ext) from model zoo[⑤]. Then we continued pre-training it on this financial data. For EDM, we set the learning rate to 2e-5. The batch size was 16. For EEM, we applied a learning rate of 2e-5 to the PLM layer and 1e-4 to other layers. The batch size was 8. The tradeoff weight w_t, w_a, and w_j was set to 0.5, 0.5, and 0.2, respectively. Each kind of syntactic embedding dimension was set to 40. The relative position embedding dimension was set to 64. We applied dropout to the SA layer and all input embeddings with the rate set to 0.3. With the model ensemble strategy, we trained five EDMs for a better event type prediction. For EEM, we trained 5 models, and ensembled the 5 results to obtain pseudo label on the testing data. Then, we trained 10 EEMs on the mixed training data, and obtained 10 predictions on the testing data. We ensembled all 15 EEM results as the final submissions.

④ https://share.weiyun.com/5G90sMJ
⑤ https://github.com/dbiir/UER-py/wiki/Modelzoo

4.3 Main Result

Since the ground truth of testing data was not available, we conducted experiments on the validation data. The *F*1-score of event detection, trigger extraction and argument extraction on validation data was 0.921, 0.970, and 0.889, respectively.

The best result (*F*1-score) of our approach on official testing data was 0.8781, which was the highest score in the competition.

4.4 Ablation Study

We conducted an ablation study on the event extraction model, where the results are shown in Table 3. As the entire decoding process is a pipelined paradigm, the performance of each submodel is affected by the previous predicted results. To avoid this impact, we adopted ground-truth results as the input of each submodel. Specifically, Line1 in Table 3 shows the complete model, which was trained on both ground-truth data and pseudo-label data with all components. Line 2 in Table 3 removed the pseudo-label data, and the results show that utilizing pseudo-label data improved performance significantly. The following experiments were ablated based on Line 2. Line 3 removed source data in training, and the result indicated that learning target events with source events was effective. Line 4 and Line 5 in Table 3 replaced the continued pre-trained PLM by BERT and the standard BERT-ext, which suggests the effectiveness of continuing pre-training for PLM. Line 6 in Table 3 replaced CLN with a simple concatenate operation, which indicates CLN can utilize condition information more effectively. Line 7 in Table 3 applied the same learning rate to all layers, which indicates utilizing different learning rates on model layers benefits the learning process. Line 8 in Table 3 removed syntactic features, which indicates syntactic features improved the extraction performance. All the results demonstrated the effectiveness of each component in the task.

Table 3. Results on validation data.

	Trigger Extraction			Argument Extraction			*F*1-mean
	P	*R*	*F*1	*P*	*R*	*F*1	
1. Complete model	**.969**	**.979**	**.970**	.844	**.969**	**.889**	**.930**
2. w/o pseudo-label data	.940	.952	.939	.845	.863	.838	.888
3. w/o source data	.941	.945	.934	.818	.865	.823	.878
4. repl PLM: BERT	.901	.924	.904	.789	.825	.789	.846
5. repl PLM: BERT-ext	.931	.938	.929	.837	.886	.828	.879
6. repl CLN: concat	.931	.931	.926	.807	.860	.814	.870
7. w/o layer lr	.946	.945	.940	.799	.869	.816	.878
8. w/o syntactic feature	.921	.924	.917	**.863**	.874	.856	.887

4.5 Case Study

To demonstrate the predicted results of the model, we conducted case study for model predictions. Figure 3 depicts an example of the model results. Given an input sentence, the model sequentially conducted

event detection, trigger extraction and argument extraction. The model first predicted all event types occurring in the sentence. Then, the model extracted triggers according to the given event type, respectively. Next, the model extracted all arguments according to the given event type and the given trigger. Such a process extracted overlapped elements separately. Besides, by sharing parameters across different types in such a unified model, the model learned to extract low-resource events based on the high-resource events.

 # Sentence
世纪华通/ 作价/ 298.03亿元/ 收购/ 盛跃网络/ 100%/ 股权。
Shijihuatong/set a price of/ 29.803 billion yuan/ to acquire/ Shengyue Network's/ 100% equity.

 # Event Detection Model
-> given: the sentence
prediction: 投资/ investment
股份股权转让/ share transfer

 # Trigger Extraction Module
-> given: type: 投资/ investment
prediction: 收购/ acquire
-> given: type: 股份股权转让/ share transfer
prediction: 收购/ acquire

 # Argument Extraction Module
-> given:
type: 投资/ investment;
trigger: 收购/ acquire
prediction:
Sub-company: 世纪华通/ Shijihuatong
Obj-company: 盛跃网络/ Shengyue Network
Money: 298.03亿元/ 29.803 billion yuan

-> given:
type: 股份股权转让/ share transfer
trigger: 收购/ acquire
prediction:
Sub-company: 盛跃网络/ Shengyue Network
Obj-company: 世纪华通/ Shijihuatong
Money: 298.03亿元/ 29.803 billion yuan
Proportion: 100%/ 100%
Collateral: -

Figure 3. An example of the model predictions.

Though the model attempted to solve the low-resource issue and the element overlapping issue, there existed two main error patterns: 1) the error proporgation problem: Since the subtasks were conducted in a cascading manner, the errors of the former predictions would lead to the errors of the following predictions; 2) the argument prediction errors: Since the arguments were complicatedly associated with their roles, the model tended to predict less arguments or predict arguments with wrong roles. We would attempt further improvements in the future.

5. CONCLUSION

In this paper, we proposed a financial event extraction approach based on a joint learning framework, which fully utilizes all the data to improve the performance of low-resource event types, and effectively solves the overlapping problem of events. The experimental results show that the approach achieved significant performance, and it ranked the first place in the CCKS-2020 financial event extraction competition.

ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Program of China (No. 2016YFB1000105) and the National Natural Science Foundation of China (No. 61772151). This work's computing device is also supported by Beijing Advanced Innovation Center of Big Data and Brain Computing, Beihang University. The author Shu Guo is supported by "Zhizi Program".

AUTHOR CONTRIBUTIONS

This work was a collaboration of all the authors. J.W. Sheng (shengjiawei@iie.ac.cn) proposed the idea of the joint learning framework, and was the team leader in the CCKS-2020 competition. Q. Li (liqian@act.buaa.edu.cn) and Y.M. Hei (black@buaa.edu.cn) devised and implemented the details of the model. S. Guo (guoshu@cert.org.cn) revised and proofread the resulting manuscript. B.W. Yu (yubowen@iie.ac.cn) provided constructive solutions of model components. L.H. Wang (wlh@isc.org.cn) guided the team of the competition. M. He (heminsmile@163.com), T.W. Liu (liutingwen@iie.ac.cn) and H.B. Xu (hbxu@iie.ac.cn) provided insightful and constructive comments on this manuscript. All the authors have made meaningful and valuable contributions to this manuscript.

DATA AVAILABILITY STATEMENT

The data sets generated and/or analyzed during the current study are not publicly available due to the fact that the data sets are produced by expert consultants of China Merchants Bank based on their own experience. The publicly released version of the data sets needs the consent of all expert consultants, and they are available from the corresponding author on reasonable request.

REFERENCES

[1] Chen, Y., et al.: Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 167–176 (2015)

[2] Deng, S., et al.: Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In: WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 151–159 (2020)

[3] Huang, L., et al.: Zero-shot transfer learning for event extraction. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 2160–2170 (2018)

[4] Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 300–309 (2016)

[5] Nguyen, T.M., Nguyen, T.H.: One for all: Neural joint modeling of entities and events. In: The 33rd AAAI Conference on Artificial Intelligence (AAAI-19), pp. 6851–6858 (2019)

[6] Devlin, J., et al.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 4171–4186 (2019)

[7] Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 73–82 (2013)
[8] Liu, X., Luo, Z., Huang, H.: Jointly multiple events extraction via attention-based graph information aggregation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1247–1256 (2018)
[9] Sha, L., et al.: Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In: The 32nd AAAI Conference on Artificial Intelligence (AAAI-18), pp. 5916–5923 (2018)
[10] Du, X., Cardie, C.: Event extraction by answering (almost) natural questions. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 671–683 (2020)
[11] Li, F., et al.: Event extraction as multi-turn question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 829–838 (2020)
[12] Liu, J., et al.: Event extraction as machine reading comprehension. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1641–1651 (2020)
[13] Wadden, D., et al.: Entity, relation, and event extraction with contextualized span representations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5783–5788 (2019)
[14] Yang, S., et al.: Exploring pre-trained language models for event extraction and generation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 5284–5294 (2019)
[15] Cao, P., et al.: Incremental event detection via knowledge consolidation networks. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 707–717 (2020)
[16] Su, J.L.: Conditional text generation based on conditional layer normalization (in Chinese). Available at: https://spaces.ac.cn/archives/7124. Accessed 21 May 2021
[17] Yu, B., et al.: Semi-open information extraction. In: Proceedings of WWW (2021). Available at: https://www2021.thewebconf.org/program/papers/. Accessed 21 May 2021
[18] Ba, L.J., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)

AUTHOR BIOGRAPHY

Jiawei Sheng is currently pursuing his PhD degree in the Institute of Information Engineering, Chinese Academy of Sciences. His current research interests include information extraction, knowledge graph embedding and knowledge acquisition.
ORCID: 0000-0002-4865-982X

Qian Li is currently pursuing her PhD degree in the School of Computer Science, Beihang University. Her current research interests include knowledge graph embedding and information extraction.
ORCID: 0000-0002-1612-4644

Yiming Hei is currently pursuing his PhD degree in the School of Cyber Science and Technology, Beihang University. His research interests include knowledge graph embedding and information extraction.
ORCID: 0000-0003-0794-9932

Shu Guo received her PhD degree from the Institute of Information Engineering, Chinese Academy of Sciences. She is currently working at the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her research interests include knowledge graph embedding, knowledge acquisition and Web mining.
ORCID: 0000-0002-9660-0291

Bowen Yu is currently pursuing his PhD degree in the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include information extraction, metric learning and unsupervised learning.
ORCID: 0000-0002-6804-1859

Lihong Wang is a Professor at the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her current research interests include big data mining and analytics, information retrieval and natural language processing.
ORCID: 0000-0003-0179-2364

Min He is currently working at the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her research interests include natural language processing, Web mining and information security.

Tingwen Liu is an Associate Researcher of the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include information extraction and knowledge fusion, text understanding, semantic computing, and heterogeneous network representation.

Hongbo Xu is a researcher in the Institute of Information Engineering, Chinese Academy of Sciences. His current research interests include knowledge graph and Web mining.